

**Entwicklung einer intelligenten  
computergestützten Lernumgebung für die  
elektrotechnische Grundlagenausbildung**

Dissertation zur Erlangung des  
akademischen Grades Doktor – Ingenieur (Dr.-Ing.)

vorgelegt der Fakultät für Elektrotechnik und Informationstechnik  
der Technischen Universität Ilmenau

von Dipl.-Ing. Vera Yakimchuk

1. Gutachter: Prof. Dr.-Ing. habil. H.-U. Seidel
2. Gutachter: Prof. Dr. C. Möbus
3. Gutachter: Dr.-Ing. H.-D. Wuttke

Tag der Einreichung: 14.02.2005

Tag der wissenschaftlichen Aussprache: 05.04.2006

## Kurzfassung

Im Rahmen der vorliegenden Arbeit werden Ansätze vorgestellt, die zur Entwicklung einer computergestützten Lernumgebung für ein selbstgesteuertes, exploratives Lernen am Fachgebiet Grundlagen der Elektrotechnik (GET) beitragen und sich als Weiterentwicklung der Forschungen des Fachgebietes in Richtungen multimedialer und intelligenter Lernumgebungen einordnen lassen. Die Motivation, Aufgabenziele und Vorgehensweise sind im ersten Kapitel dargelegt. Konzeption und Implementierung interaktiver Lernmodule für die GET-Domäne, Einsatz von Ingenieurwerkzeugen in der GET-Lehre, sowie Konzeption und prototypische Implementierung einer wissensbasierten Problemlöseumgebung für GET sind die Aufgabenstellungen dieser Arbeit.

Mit einigen Grundlagen der computerbasierten Lernumgebungen (Multimedia-Einsatz, Lerntheorien, KI-Einsatz) beschäftigt sich das zweite Kapitel.

Die "State-Of-The-Art"-Analyse basiert auf Materialien von internationalen Konferenzen und Zeitschriften: *Frontiers in Education*, *IEEE Transactions on Education*, *Artificial Intelligence in Education*, *Intelligent Tutoring Systems*, *KI*, *NITE*, *Proceedings of the IEEE* u. a. Die ausgewählten verwandten Arbeiten werden im dritten Kapitel diskutiert.

Da in virtuellen (computerbasierten) Lernumgebungen für individualisiertes Lernen das Lernen in erster Linie die kognitive Auseinandersetzung des Lernenden mit dem zu lernendem Inhalt (dem Lernobjekt) bedeutet, wird Interaktivität der Lernobjekte zu einem der entscheidenden Qualitätskriterien des Lernens und der Lernumgebungen.

Konzeption und Weiterentwicklung der interaktiven Komponenten für die Lernumgebung GETsoft und Fragen des Mathcad-Einsatzes werden im vierten Kapitel betrachtet. Die webbasierte multimediale Lernumgebung bietet den Lehrenden viele multimediale Komponenten für das Zusammenstellen ihrer Vorlesungen und Übungen. Die GETsoft-Studierenden haben Zugang zu den typischen Themen und Problemstellungen des Lehrgebietes. Interaktive Experimente und Aufgaben mit Möglichkeiten zur Selbstkontrolle ermöglichen eine aktive Verarbeitung des Lehrstoffes

Der in der Forschungsgruppe von Prof. Dr. C. Möbus (OFFIS, Oldenburg) entwickelte IPSE-Ansatz (IPSE - Intelligent Problem Solving Environment) wurde auf der neuen für den Ansatz Domäne der Grundlagen der Elektrotechnik und für die neuen Aufgabenarten (Formelanalyse, Lösungsentwurfskontrolle, Schaltungsanalyse) umgesetzt.

Die Konzeption der Problemlöseumgebung mileET aus GET-Sicht und einige Teile der implementierten wissensbasierten Komponente werden im fünften Kapitel betrachtet. MileET ist eine innovative Art der Lernerunterstützung auf dem GET-Gebiet. Das Programm ermöglicht den Lernenden aktives, selbständiges Lernen durch Aufgabenlösen mit und in dem Programm. Studierende haben die Möglichkeit, verschiedene Lösungsentwürfe zu verschiedenen Aufgaben zu den Themenfeldern Methoden der Netzwerkberechnung zusammen zu stellen und vom Programm testen zu lassen. Die implementierte wissensbasierte Komponente, eine spezielle Art des Expertensystems, besitzt die Fähigkeit die Aufgaben symbolisch zu lösen. Damit kann eine flexible und dynamische Lösungserzeugung unter Berücksichtigung der Benutzereingaben stattfinden, eine adaptive Rückmeldung über den Lösungsentwurf und eine adaptive Vervollständigung einer unvollständigen Benutzerlösung erfolgen. Den Lehrenden bietet das Programm die Möglichkeit verschiedene (neue) Aufgaben einfach zu erstellen und die Routine-Korrekturarbeit dem Programm zu überlassen.

Die Überlegungen zur Weiterentwicklung der intelligenten Lernumgebung GETsoft sind im letzten Kapitel gegeben.

## Abstract

The theses presented in the following paper describe the concept and implementation of interactive and intelligent components for computer-based learning environment in the electrical engineering department GET. The achieved results carry forward the work of GET research group on computer-based learning environments. Chapter 1 states the motivation and goals of the work presented. The main goals are the conceptual design and implementation of interactive components within the GETsoft learning environment, application of engineering tools (Mathcad) for computer-based education, and concept and prototype implementation of knowledge-based problem solving environment.

Some basics of computer-based learning (multimedia, learning science, artificial intelligence in education) are introduced in chapter 2.

The state-of-the-art analysis is based on the scientific works published by *Frontiers in Education*, *IEEE Transactions on Education*, *Intelligent Tutoring Systems*, *KI*, *NITE*, *Proceedings of the IEEE*, and *Artificial Intelligence in Education*, as well as others. The most relevant projects are discussed in chapter 3.

Learning in the virtual learning environment involves the cognitive confrontation with the subject matter to be learned (learning object). Thus, the interactivity of the learning objects becomes the crucial criteria of the quality of learning within the computer-based learning environment.

The concept and enhancements of interactive components in GETsoft as well as the aspects of using Mathcad as an engineering tool are considered in chapter 4. The web-based GETsoft offers many multimedia components that can be used by teachers to organize their own learning units. Students get access to typical subject topics and questions. The interactive components and tasks with self-control opportunities make possible the active treatment of the subject matter.

Intelligent Problem Solving Environments (IPSEs), a special type of intelligent tutoring systems, are being developed by the research group of Prof. C. Möbus (OFFIS, Oldenburg). The ideas of IPSE have been adapted for the new (for the approach) domain of the basics of electrical engineering and for new kinds of tasks (symbolic analysis, circuit analysis, solution draft control).

Chapter 5 describes from the EE-education point of view the concept of the knowledge-based problem solving environment mileET and implementation of a part of its knowledge base. MileET represents an innovative kind of learning support within the GET area. It is intended to be used by students for active, independent learning while solving tasks with and in the program. Students have the possibility to formulate different solution proposals and ask the program to check them. The implemented knowledge-based component, a special kind of expert system, contains necessary knowledge for tasks and methods of the network analysis and possesses the ability of symbolic task solving. Due to a flexible and dynamic solution production with consideration of the user's inputs, an adaptive feedback on the solution draft as well as an adaptive completion of an incomplete user's solution can take place. To the teachers, the program offers the possibility to provide different (new) tasks and to leave the routine correction work to the program.

The key results as well as some ideas concerning the future work on the intelligent learning environment GETsoft are given in the last chapter.

Allen Mitarbeitern des Fachbereiches GET an der TU Ilmenau, der Abteilung I@LLS der Universität Oldenburg und des Bereichs Sicherheitskritische Systeme des An-Instituts OFFIS (Oldenburg), die die Anfertigung dieser Arbeit ermöglichten, möchte ich sehr herzlich danken.

Für die Anregung zu dieser Arbeit und die Unterstützung bei der Anfertigung möchte ich mich besonderes bei Herrn Prof. Dr.-Ing. habil. E. Wagner, Prof. Dr. C. Möbus, Prof. Dr.-Ing. habil. H.-U. Seidel und Prof. E. V. Kuznetzov bedanken.

Mein besonders herzlicher Dank gilt Frau Dipl.-Inform. H. Garbe für die kompetente fachliche Unterstützung und wertvolle kreative Mitarbeit bei der Konzeption und Entwicklung der wissensbasierten Komponente mileET. Weiterhin möchte ich Herrn H.-J. Thole für die Beratung in Fragen der Prolog-Programmierung und den studentischen Mitarbeitern J.-P. Osterloh, L. Weber und T. Knostmann für die Implementierung der Benutzerschnittstelle für mileET herzlich danken.

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>1</b>
1.1	Motivation und Zielsetzung .....	1
1.2	Einordnung der Arbeit .....	4
1.3	Vorgehensweise .....	6
<b>2</b>	<b>Grundlagen.....</b>	<b>8</b>
2.1	Begriffsbestimmungen.....	8
2.2	Grundfragen der Lehr-/Lernprogramme .....	11
2.3	Multimedia-Einsatz in der Lehre .....	17
2.4	Einsatz der Techniken der Künstlichen Intelligenz in Lernumgebungen .....	20
<b>3</b>	<b>Stand der Technik.....</b>	<b>25</b>
3.1	(Intelligente) Lernprogramme für die Grundlagen der Elektrotechnik .....	25
3.2	Relevante Arbeiten aus anderen Domänen .....	31
3.3	Computeralgebrasysteme in der Ingenieurausbildung .....	34
3.4	Zusammenfassung .....	38
<b>4</b>	<b>Weiterentwicklung der Lernumgebung GET.....</b>	<b>40</b>
4.1	Interaktive Experimentierumgebungen in der Lernumgebung GET.....	43
4.2	Feedback-Techniken.....	48
4.3	Mathcad als Ingenieurwerkzeug .....	51
4.4	Zusammenfassung .....	54
<b>5</b>	<b>Intelligente Lernerunterstützung in der wissensbasierten Problemlöseumgebung mileET.....</b>	<b>58</b>
5.1	Konzeption einer intelligenten Lernumgebung für GET .....	58
5.1.1	Didaktische und softwaretechnische Konzeption.....	58
5.1.2	Auswahl der Themenfelder und typischer Aufgaben.....	59
5.1.3	Modelle und Repräsentation der mileET-Aufgaben und -Lösungen.....	64
5.2	mileET: prototypische Realisierung einer Problemlöseumgebung für GET.....	68
5.2.1	Benutzeraktivitäten in mileET .....	68
5.2.2	Systemarchitektur, Datenstrukturen und Wissensrepräsentation .....	74
5.2.3	Benutzeradaptive Aufgabenlösungen in mileET.....	79
5.2.3.1	Anwendung der Zweipoltheorie .....	84
5.2.3.2	Anwendung der Methoden der Kirchhoff'schen Sätze.....	93
5.2.3.3	Anwendung der Methode der Knotenspannungsanalyse.....	97
5.2.3.4	Anwendung der Methode der Maschenstromanalyse .....	102
5.3	Zusammenfassung .....	104
<b>6</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>107</b>
<b>7</b>	<b>Literaturverzeichnis .....</b>	<b>113</b>
<b>8</b>	<b>Anhang.....</b>	<b>122</b>

## Abbildungsverzeichnis

Abbildung 1. mileET als Lernumgebung für das selbständige Lernen .....	3
Abbildung 2. Beispiel einer möglichen Basisarchitektur eines Expertensystems .....	21
Abbildung 3. Verwandte Arbeiten .....	38
Abbildung 4. Computerbasierter Teil der offenen Lernumgebung GETsoft .....	41
Abbildung 5. Struktur eines GETsoft-Lernmoduls .....	41
Abbildung 6. Interaktive Komponenten der Lernumgebung GETsoft .....	42
Abbildung 7. Didaktische Interaktivität im Lernmodul Fourier-Reihe .....	43
Abbildung 8. Mischpult .....	47
Abbildung 9. Abgestufte Hilfe im Modul Fourier-Reihe .....	48
Abbildung 10. Differenzierte Feedbacktechniken in GETsoft-Lernprogramm Fourier-Reihe .....	50
Abbildung 11. Anwendungsfälle für mileET .....	59
Abbildung 12. Thematische Verteilung der GET-Aufgaben .....	60
Abbildung 13. Beispiel einer Aufgabe aus dem Themenfeld Kirchhoff'sche Gleichungen .....	61
Abbildung 14. Beispiel einer Aufgabe aus dem Themenfeld Berechnung der Widerstandskombinationen .....	62
Abbildung 15. Beispiel einer Aufgabe aus dem Themenfeld Anwendung der Zweipoltheorie .....	62
Abbildung 16. Beispiel einer Aufgabe zum Thema Anwendung der Methode der Knotenspannungsanalyse .....	62
Abbildung 17. Y-Modelle .....	64
Abbildung 18. Allgemeines Y-Modell .....	65
Abbildung 19. Übergang zu einem allgemeinem Aufgaben-Modell .....	66
Abbildung 20. mileET-Lernumgebung .....	68
Abbildung 21. Unterstützung der Dreieck-Stern-Transformation .....	70
Abbildung 22. mileET - Formeleditor .....	70
Abbildung 23. Vervollständigung einer Benutzerlösung in mileET .....	71
Abbildung 24. mileET-Dozentenmodus und Aufgabenassistent .....	73
Abbildung 25. mileET-Systemarchitektur .....	74
Abbildung 26. mileET als spezielle Art des Expertensystems .....	75
Abbildung 27. Unterstützte Aufgabenziele in mileET .....	79
Abbildung 28. Möglicher Verlauf der Arbeit mit mileET .....	80
Abbildung 29. Erzeugen einer Aufgabe in der wissensbasierten Komponente .....	81
Abbildung 30. Lösbarkeitsanalyse für die Aufgaben des Y-Typs 1 .....	82
Abbildung 31. Struktur des Aufgabenlöseprozesses .....	83
Abbildung 32. Lösen einer Aufgabe des Y-Typs 2 .....	85
Abbildung 33. Lösungsbaum beim Berechnen eines Ersatzwiderstandes .....	86
Abbildung 34. Lösungsbaum beim Ermitteln der Eingangsimpedanz .....	86
Abbildung 35. Lösungsmethode der Eingangsimpedanzberechnung aus der Sicht des Meta-Interpreters .....	87
Abbildung 36. Transformationsmöglichkeiten in mileET .....	88
Abbildung 37. Zielzerlegung und Lösungsaufbau in SPM-Mengen für Aufgaben Y-Typ 2 .....	88
Abbildung 38. mileET-Lösungsraum einer einfachen Aufgabe .....	90
Abbildung 39. Lösen einer Aufgabe des Y-Typs 1 .....	93
Abbildung 40. Methode der Kirchhoff'schen Gleichungen .....	94

Abbildung 41. Benutzerlösungsentwurf in mileET .....	95
Abbildung 42. Vervollständigung eines Lösungsentwurfes mit Kirchhoff'schen Gleichungen .....	97
Abbildung 43. Vorbereitende Analyse der Methode der Knotenspannungsanalyse .....	98
Abbildung 44. Methode der Knotenspannungsanalyse .....	99
Abbildung 45. Graphische Unterstützung der Methode der Knotenspannungsanalyse .....	102
Abbildung 46. Methode der Maschenstromanalyse .....	103
Abbildung 47. Graphische Unterstützung der Methode der Maschenstromanalyse .....	104
Abbildung 48. Lernerunterstützung in GETsoft mit mileET .....	105
Abbildung 49. Lösungskontrollmöglichkeiten in GETsoft und mileET .....	109
Abbildung 50. mileET als Teil einer multimedialen Lernumgebung .....	112

## Tabellenverzeichnis

Tabelle 1. Positionierung der Arbeit in der Struktur des Lehrgebietes GET .....	5
Tabelle 2. Zusammenfassung von Aufgabenzielen und Aufgabendaten .....	63
Tabelle 3. Berechnung des Ersatzwiderstandes in Prolog-Notation .....	85
Tabelle 4. Beispiel eines Lösungsschrittes mit GMR .....	89
Tabelle 5. Aufgabenziele zum Thema Anwendung der Zweipoltheorie .....	91
Tabelle 6. Aufbau der Methoden für die Parameterberechnung .....	92
Tabelle 7. Prolog-Notation für die Methode der Kirchhoff'schen Gleichungen .....	94
Tabelle 8. Erweiterung der intelligenten Lernumgebung auf weitere Themenfelder .....	111

## Verzeichnis der verwendeten Bezeichnungen und Abkürzungen

AIED	–	Artificial Intelligence in Education
CAI, CBT	–	computer-assisted instruction, computer-based training
FIE	–	Frontiers in Education
GET	–	Grundlagen der Elektrotechnik
gmr, GMR	–	goal-mean-relation
$I, I_q$	–	elektrische Stromstärke, Stromquelle
ILE	–	Intelligent Learning Environment
IMITS	–	Interactive Multimedia Intelligent Tutoring System
IPSE	–	Intelligent Problem Solving Environment
ISP-DL	–	Impasse -Success-Problem-Solving-Driven-Learning
ITS	–	Intelligent Tutoring System
KI, AI	–	Künstliche Intelligenz, Artificial Intelligence
$P$	–	elektrische Leistung
$R$	–	elektrischer Widerstand
SPME	–	'Schaltbild' – 'Parameter' – 'Mathematische Formeln' – 'Erklärungen'
$U, U_q$	–	elektrische Spannung, Spannungsquelle
XML	–	Extensible Markup Language

# 1 Einleitung

## 1.1 Motivation und Zielsetzung

*„The Shift from Teaching to Learning“ war einer der wichtigsten Themenbereiche auf der UNESCO-Weltkonferenz 1998 über „High Education in the XXIst Century ...Der Wechsel vom Lehren zum Lernen ... umfasst u. a. selbstgesteuertes Lernen und die Definition von Lehrenden als Förderer für dieses selbstgesteuertes Lernen... [Berendt2003].*

Dieser Paradigmawechsel betrifft sowohl organisatorische als auch technologische Aspekte des Lehrens/Lernens. Im Rahmen dieser Arbeit werden Ansätze vorgestellt, die zur Entwicklung einer Lernumgebung für ein selbstgesteuertes, explorierendes Lernen am Fachgebiet Grundlagen der Elektrotechnik beitragen.

Explorierendes Lernen am Computer erfordert eine interaktive Gestaltung der Lernwelt, die zum Erforschen, zum auf die Probe Stellen, zum Erfahrung Sammeln einlädt. Das aktive Experimentieren und die daraus resultierende konkrete Erfahrung finden nicht mit dem Objekt selbst statt, sondern an seinem modellhaften Ersatz, der Lernsoftware.

Konzeption und prototypische Realisierungen zum Einsatz des Computers zu Lernzwecken in der elektrotechnischen Grundlagenausbildung haben eine lange Tradition [Proceedings IEEE<sup>1</sup>, IEEE Transactions on Education, FIE<sup>2</sup>, NITE<sup>3</sup>]. Allerdings hat es erst zur Mitte der neunziger Jahre einen technologischen Fortschritt gegeben, der eine Entwicklung von multimedialen Systemen ermöglicht hat. Die Hauptinteressen der ersten Projekte zum Einsatz neuer Medien in der Bildung waren Konzeption und Implementierung geschlossener Lernumgebungen, die durch den Einsatz verschiedener Medien die Lehre modernisieren sollten. Mit der Entwicklung der Internet-Technologien begann eine Verschiebung der Schwerpunkte in Richtung offener webbasierter Lernumgebungen. Die Modernisierung der Lehre am Fachgebiet Grundlagen der Elektrotechnik durch neue Medien und die Konzeption einer webbasierten Lernumgebung für GET sind einige der Zielsetzungen der vorliegenden Arbeit, die im Rahmen der GETsoft<sup>4</sup>-Initiative ausgearbeitet worden sind.

Handlungsorientierung und Interaktivität der Lernobjekte sind die anerkannten Qualitätskriterien von modernen Lernumgebungen [Tulodziecki2004, Schulmeister2004, Haack2002].

---

<sup>1</sup> Institute of Electrical and Electronics Engineers, Inc. <http://www.ieee.org> (letzter Zugriff 10.02.2005)

<sup>2</sup> FIE - Frontiers in Education Homepage <http://fie.engrng.pitt.edu/> (letzter Zugriff 10.02.2005)

<sup>3</sup> NITE (rus. Novye Informationnye Technologii v Elektrotechnicheskom Obrazovanii) neue Informationstechnologien für elektrotechnische Ausbildung

<sup>4</sup> [www.getsoft.net](http://www.getsoft.net) (letzter Zugriff 10.02.2005)

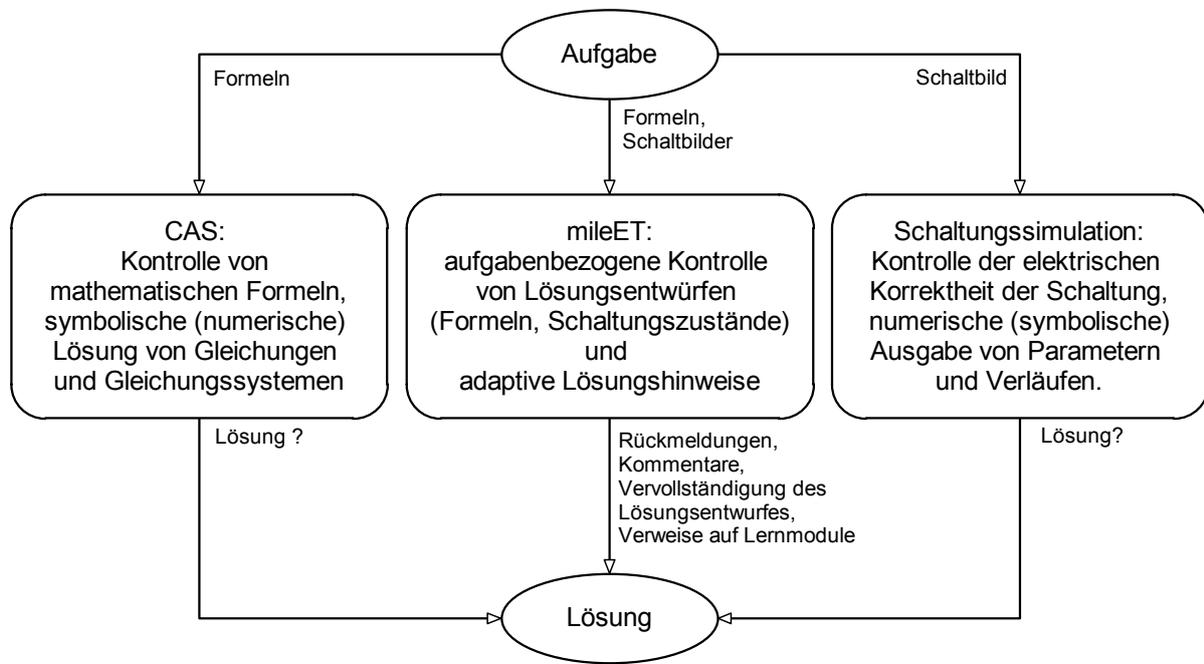
Verschiedene Möglichkeiten, interaktive Auseinandersetzung mit dem Lernstoff zu unterstützen bzw. zu ermöglichen, liegen im Interessensfeld der Autorin. Der Übergang von geschlossenen, rein instruktionsorientierten Lernprogrammen zu einer offenen Lernumgebung wird durch die Realisierung von interaktiven Modulen, die sich in verschiedenen Lernszenarien einsetzen lassen, angestrebt.

Um sich auf die spätere Berufspraxis vorzubereiten, sollten die Studierenden auch mit computerbasierten Ingenieurwerkzeugen konfrontiert werden. Computer-Algebra-Systeme (CAS) und Computer-Aided-Design-Systeme (CAD) sind die zwei wichtigen Software-Werkzeuge, die in der Ingenieurpraxis und -ausbildung eine große Rolle spielen. Die Möglichkeit, diese Systeme schon in der Grundlagenausbildung einzusetzen, soll am Beispiel einiger Themen der Grundlagen der Elektrotechnik (GET) untersucht werden. Das Erlernen eines Ingenieurwerkzeuges durch Lösen von Übungsaufgaben bzw. durch Bearbeitung der vorbereiteten Beispiele sollen ebenfalls ermöglicht werden.

Während im Rahmen des Projektes „GETsoft“ die Möglichkeiten untersucht wurden, die elektrotechnische Grundlagenausbildung multimedial zu gestalten und das selbständige Lernen durch interaktive multimediale Experimente zu unterstützen, soll mit mileET eine intelligente Begleitung des Lernens realisiert werden. Als qualitative Weiterentwicklung des eLearning-Angebotes von GETsoft soll ein Versuch unternommen werden, ein intelligentes Lehrsystem für die Grundlagen der Elektrotechnik zu entwickeln und im Studienbetrieb zu erproben.

Das Einarbeiten in die ET-Domäne wird mittels verschiedener Lehr-/Lernformen realisiert: Vorlesungen, Übungen, Laborversuche, Projektarbeiten. Das Lösen der Lehraufgaben (Übungen) spielt insbesondere in den ersten Phasen des Ausbildungsprozesses eine wichtige Rolle. Die Überprüfung von Lösungen geschieht mit Unterstützung von Lehrkräften bzw. durch den Vergleich mit vorgerechneten Beispielen bzw. angegebenen Antworten. Werden Ingenieurwerkzeuge eingesetzt, so kann für die meisten GET-Problemstellungen sicher gesagt werden, dass die mathematischen Berechnungen für die formulierten Gleichungen korrekt sind bzw. dass die errechneten numerischen Parameter die gezeichnete Schaltung tatsächlich dimensionieren. Größere Unterstützung geben die symbolischen Simulationstools: Sie ermitteln symbolische Zusammenhänge zwischen einzelnen Elementen einer Schaltung, ermöglichen damit die qualitative Analyse der Schaltung und „geben den Blick in die Schaltung hinein“ [Heulsman1996]. Aber auch sie sind nicht in der Lage, Informationen darüber zu geben, ob die vom Benutzer eingegebenen Formeln bzw. Schaltbilder zur Lösung der gestellten Aufgabe gehören. Ebenfalls können keine der Ingenieurwerkzeuge, aufgrund

ihrer Orientierung an die Domänen-Experte (Fachexperte), (unvollständige) Lösungswege beurteilen und kommentieren. Den Werkzeugen, die den modernen Lernenden zur Verfügung stehen, mangelt es an der adaptiven Unterstützung der Benutzer. Weder CAS noch Simulationstools sind in der Lage, die Eingaben des Benutzers kontextsensitiv auf Sinnhaftigkeit zu untersuchen.



**Abbildung 1. mileET als Lernumgebung für das selbständige Lernen**

Der Schwerpunkt der zu entwickelnden Umgebung soll also in der adaptiven Unterstützung der Studierenden während des Prozesses des Aufgabenlöses liegen: mileET soll in der Lage sein, auch unvollständige Benutzerlösungen zu analysieren. Mit dem System soll ein weiterer Schritt auf dem Gebiet der intelligenten Lernsysteme für GET gewagt werden.

Zusammenfassend lassen sich folgende Aufgaben der vorliegenden Arbeit formulieren:

- Konzeption und Implementierung interaktiver Lernmodule für die GET-Domäne,
- Einsatz von Ingenieurwerkzeugen in der GET-Lehre,
- Konzeption und prototypische Implementierung einer intelligenten Problemlöseumgebung für GET-Lernumgebung.

Zusätzlich sollten die Ergebnisse der Arbeit sowohl den Lernenden als auch den Lehrenden Vorteile bringen. Die zu implementierenden Elemente der GET-Lernumgebung sollen den Studierenden aktives und selbstständiges Lernen ermöglichen: der Wissenserwerb durch Experimentieren und Ausprobieren sowie durch Aufgabenlösen im System. Die Lehrenden sollen multimediale Materialien für die Vorlesungen und Übungen erhalten, mit deren Hilfe sich bestimmte Sachverhalte anschaulicher erklären lassen, und eine intelligente Lernumgebung, die sie von Routine-Korrekturarbeit bei den Übungen entlasten würde. Diese

Maßnahmen sollen den Lehrenden mehr Zeit für individuelle Beratungen bzw. für Diskussionen über den Lehrstoff geben. Auch ein Beitrag für Fernstudium und Weiterbildung wird dadurch geleistet.

## 1.2 Einordnung der Arbeit

Die vorliegende Arbeit stellt einen Beitrag zur Entwicklung einer intelligenten Lernumgebung der Grundlagen der Elektrotechnik dar und lässt sich als Weiterentwicklung der Forschungen des Fachgebietes GET [Burger/Finsterbusch1985, Rettig1997] in Richtungen multimedialer und intelligenter Lernumgebungen einordnen. Sie befasst sich mit den Möglichkeiten, die elektrotechnische Grundlagenausbildung mit Hilfe von multimedialen und intelligenten Lernmodulen zu unterstützen. Die interaktiven Komponenten der in [Rettig1997] vorgestellten Konzeption einer Lernumgebung für GET wurden im Rahmen der vorliegenden Arbeit implementiert und in der Lehre erprobt. Der in der Forschungsgruppe von Prof. Möbus entwickelte IPSE<sup>5</sup>-Ansatz wurde auf der (für den Ansatz) neuen Domäne (GET) und für neue Aufgabenarten (Formelanalyse, Lösungsentwurfkontrolle, Schaltbildtransformationsanalyse) umgesetzt.

Die "State-Of-The-Art"-Analyse basiert auf Materialien der internationalen Konferenzen und Zeitschriften: Frontiers in Education, IEEE Transactions on Education, Artificial Intelligence in Education<sup>6</sup>, Intelligent Tutoring Systems, KI<sup>7</sup>, NITE, Proceedings of the IEEE u. a.

Die Ergebnisse werden in der Lehre an der TU Ilmenau eingesetzt bzw. wurden mit Studierenden erprobt.

Die folgende Tabelle 1 zeigt die Strukturierung des Lehrgebietes Grundlagen der Elektrotechnik und die thematische Einordnung der Arbeit. Zum Thema „Fourier-Reihen“ werden einige Beispiele interaktiver multimedialer Lernmodule vorgestellt, zum Themengebiet „Methoden der Netzwerkberechnung“ wurde die wissensbasierte Lernumgebung implementiert und zu den meisten Themen sind Mathcad<sup>8</sup>-Arbeitsblätter implementiert worden, die in der Lehre eingesetzt werden.

eLearning ist zu einem Überbegriff für alle Aspekte des Lernens (Lehrens, Ausbildung) mit Anwendung der Computertechnik und des Internet geworden und stellt ein breites Forschungsfeld dar [Baumgartner2002, Nichols2003, Niegemann2004]. Viele Schwerpunkte (z.B. Lernplattformen, Learning Management Systeme, Tools für kollaboratives Lernen,

---

<sup>5</sup> Intelligent Problem Solving Environment

<sup>6</sup> AIED Society <http://aied.inf.ed.ac.uk/> (letzter Zugriff 10.02.2005)

<sup>7</sup> KI Homepage <http://www.kuenstliche-intelligenz.de/> (letzter Zugriff 10.02.2005)

<sup>8</sup> Mathcad Homepage <http://www.Mathcad.com/> (letzter Zugriff 10.02.2005)

virtuelle Labore etc.) dieses Forschungsgebietes liegen außerhalb der Betrachtung dieser Arbeit.

Die Interessen moderner Forscher und Entwickler auf dem Gebiet Lehr-/ Lernsysteme gelten den Fragen der Standards und Metadaten. Ein Überblick über aktuelle Initiativen und Projekte ist z.B. in [Seeberg2003] zu finden. Möglichkeiten, SCORM<sup>9</sup>-kompatible Lernmodule zu entwickeln, werden z.B. in [Möbus2003b, Qu2002] vorgestellt.

Die Problematik der Repräsentation, Verwaltung und Verarbeitung der mathematischen Ausdrücke spielt bei der Implementierung des eLearning für GET eine besondere Rolle [ActiveMath<sup>10</sup>, OpenMath, MathML, Medienprojekt2003]. Für die intelligente Problemlöseumgebung mileET wurden eigene Verfahren für die Verarbeitung der mathematischen Formeln entwickelt, diese Problematik ist jedoch nicht der Schwerpunkt der vorliegenden Arbeit. Die im Weiteren vorgestellten Lernmodule greifen auch auf (angepasste) fremde Tools ([Mathcad], [JEP]) zurück.

### Lehrgebiet Grundlagen der Elektrotechnik

1. Semester	2. Semester	3. Semester
Grundbegriffe, Zweipole, Grundstromkreis	Analyse von Wechselstromschaltungen	Fourier-Reihen <sup>i</sup>
Methoden zur Netzwerkberechnung <sup>ii</sup>	Frequenzselektive Schaltungen	Laplace Transformation
Energieumformungen	Brückenschaltungen	Vorgänge auf Leitungen
Elektrisches Feld	Transformator, Übertrager	i. Am Beispiel des Lernmoduls "Fourier-Reihe" werden die GETsoft-Interaktivitätsformen vorgestellt. ii. Zum Thema „Methoden zur Netzwerkberechnung“ wird ein Prototyp der intelligenten Problemlöseumgebung vorgestellt.
Technischer Magnetkreis	Drehstrom	

**Tabelle 1. Positionierung der Arbeit in der Struktur des Lehrgebietes GET**

<sup>9</sup> SCORM - Shareable Content Object Reference Model, <http://www.adlnet.org/> (letzter Zugriff 10.02.2005)

<sup>10</sup> ActiveMath - eine Lernumgebung für Mathematik, <http://www.activemath.org/indexDE.html> (letzter Zugriff 10.02.2005)

### 1.3 Vorgehensweise

Die vorliegende Arbeit entstand im Rahmen des Forschungsaufenthaltes an der TU Ilmenau, Fachgebiet Grundlagen der Elektrotechnik (FG GET), zweier Projekte an der TU Ilmenau (FG GET) und am An-Institut OFFIS<sup>11</sup> (Forschungsgruppe I@LLS, Universität Oldenburg) und einer dreisemestrigen Tätigkeit als Lehrkraft an der TU Ilmenau.

Der Forschungsaufenthalt begann im Studienjahr 1996/1997 mit einem Praktikum an der TU Ilmenau im Rahmen des Kooperationsprogramms zwischen der TU Ilmenau und der Technischen Universität MEI (Moskauer Energetisches Institut<sup>12</sup>). Die Aufgabe bestand in der Implementierung eines interaktiven Lernmoduls zum Thema „Fourier-Reihen“ mit dem Autorensystem ToolBook mit der Kopplung von Mathcad v6 als CAS für komplexe Berechnungen.

Im Rahmen des zweijährigen Stipendiums des Landes Thüringen (Okt.1998-Okt.2000) fanden die theoretische Einarbeitung in die Themengebiete Lerntheorien, Multimedia, Computerunterstütztes Lernen, Intelligente Softwaretechniken und die praktische Arbeit im Projekt „Multimediale Lernumgebungen für Weiterbildung und Fernstudium in den Ingenieurwissenschaften“ (Jan. 1998 - Dez. 2000) statt. Im Rahmen dieses Projektes und während der Tätigkeit als Lehrkraft am Fachgebiet Grundlagen der Elektrotechnik (2000 - 2001) wurde die Arbeit zu multimedialen Lernumgebungen GETsoft durchgeführt. Während dieser Zeit wurden auch mehrere Diplompraktika und studentische Hilfskräfte betreut. Die Ergebnisse dieser Tätigkeit wurden in mehreren Berichten zusammengefasst, auf Tagungen präsentiert und veröffentlicht [Hammer1998, Iakimtchouk2000a,b, Wagner1998,1999].

Der Übergang von CBT<sup>13</sup> zu WBT<sup>14</sup> am Fachgebiet wurde in der Zusammenarbeit mit Kollegen und Praktikanten durchgeführt. Die Weiterentwicklung der nun webbasierten Lernumgebung GETsoft hat im Rahmen des Projektes mile<sup>15</sup> stattgefunden [Neundorf2003]. Im Rahmen dieses Projektes entstand auch die wissensbasierte Lernumgebung mileET. Ein Forschungsaufenthalt und die Mitarbeit im Rahmen des Projektes mile am An-Institut OFFIS (Uni Oldenburg) unter wissenschaftlicher Leitung von Prof. Dr. C. Möbus (Abteilung für Lehr- und Lernsysteme<sup>16</sup>) ermöglichte eine weitere wissenschaftliche Qualifizierung auf dem Gebiet Intelligente Lehr-/ Lernsysteme. Der im Rahmen dieses Projektes entwickelte Prototyp

---

<sup>11</sup> OFFIS - Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und -Systeme, [www.offis.de](http://www.offis.de) (letzter Zugriff 10.02.2005)

<sup>12</sup> <http://www.mpei.ru/> (letzter Zugriff 10.02.2005)

<sup>13</sup> CBT - Computer Based Training

<sup>14</sup> WBT - Web Based Training

<sup>15</sup> mile - multimedia learning environment, das Projekt wurde gefördert vom Bundesministerium für Bildung und Forschung, Förderkennzeichen 08NM73, Laufzeit: 1.1.2001 bis 31.04.2004

<sup>16</sup> I@LLS <http://lls.informatik.uni-oldenburg.de> (letzter Zugriff 10.02.2005)

einer intelligenten Problemlöseumgebung mileET stellt eine Innovation für das Lehrgebiet Grundlagen der Elektrotechnik dar. Forschungsergebnisse dieser Arbeit sind in mehreren Publikationen veröffentlicht und wurden auf mehreren Tagungen, u. a. AIED 2003<sup>17</sup>, vorgestellt [Garbe2003, Yakimchuk2002,2003,2004].

Die Dissertationsschrift ist wie folgt organisiert. Das zweite Kapitel erläutert die Grundfragen der Lehr-/Lernprogramme sowie Fragen des Multimedia-Einsatzes. Im dritten Kapitel werden mehrere verwandte Arbeiten diskutiert. Die Analyse existierender Lernprogramme zu den Grundlagen der Elektrotechnik weist auf ein Defizit von handlungs- und problemorientierten Lernumgebungen hin. Somit können die Anforderungen an die zu entwickelnde Lernumgebungen formuliert werden. Das vierte Kapitel beschreibt die interaktive Lernumgebung GETsoft. Das fünfte Kapitel stellt die Konzeption und die Realisierung der intelligenten Problemlösungsumgebung mileET dar. Die erreichten Ergebnisse und die Innovation der Arbeit sowie Überlegungen zu möglichen Weiterentwicklungen sind in dem abschließenden Kapitel diskutiert.

---

<sup>17</sup> AIED2003 <http://www.cs.usyd.edu.au/~aied/> (letzter Zugriff 10.02.2005)

## 2 Grundlagen

### 2.1 Begriffsbestimmungen

Der Begriff der Lernumgebung soll äußere Bedingungen des Lernens beschreiben [Dörr&Schrittmatter2002]. Es geht um Lernmaterialien und Lernaufgaben, die so zu gestalten sind, dass verschiedene didaktische Funktionen erfüllt werden können: *„...Lernumgebungen sollen durch eine angemessene methodische Aufbereitung des Lehrstoffes und durch besondere Lehrmaßnahmen die angezielten Lernprozesse erleichtern; Die Lernumgebung soll den Lernenden Rückmeldung über den jeweiligen Lernerfolg geben; Lernumgebungen sollen selbstgesteuertes Lernen unterstützen;...Sie sollen alternative Möglichkeiten zur Problemlösung unterstützen, um auch individuellen Auffassungen Raum zu bieten.“*

Als softwaretechnisches Konzept stellt die Lernumgebung eine offene Struktur dar, die verschiedene Lehrkomponenten, wie z.B. interaktive Übungsprogramme, Hilfen und Werkzeuge, verwaltet und dem Nutzer anbietet, so dass ein flexibles erweiterbares System entsteht. Es wird nach und nach mit Lernmodulen, Aufgabensammlungen, Animationen und Simulationen gefüllt und mit Ingenieurwerkzeugen, Bibliotheken u. a. verknüpft. Die Basis solch einer Lernumgebung bilden multimediale Elemente, die feinmodulare Lerneinheiten darstellen.

Diese Beschreibung hat viel Gemeinsames mit dem Begriff „offene Lernumgebung“ nach [Keil-Slawik2000]. Lernmaterialien in einer Lernumgebung sollten *„im Sinne der Nachhaltigkeit in einer leicht modularisierbaren Form vorliegen“*. *„Die Bereitstellung von modularen kombinierbaren Lernbausteinen“*, die sich flexibel in verschiedene Lernszenarien einsetzen (Lehrer-Sicht) und sich in individuelle Lernumgebungen integrieren lassen (Lerner-Sicht) und nicht die geschlossenen komplexen Multimedia-Anwendungen ist das Wesentliche bei der Konzeption offener Lernumgebungen.

Nach [Schulmeister2004] sind offene Lernumgebungen die Lernumgebungen, in denen die Organisation des Lernens durch die Lernenden selbst erfolgt. Sie basieren auf didaktischen Modellen wie z.B. problemorientiertes Lernen, entdeckendes Lernen, fallbasiertes Lernen, Simulationen u. a. Der Begriff *„offen“ bezeichnet dabei den Grad an Freiheit, den das lernende Individuum in diesen Umgebungen einnehmen kann, um Inhalte gemäß seinen Lernvoraussetzungen zu selektieren, seinen Lernstil und seine Lernstrategien zu praktizieren und gemäß seiner Motivation vorzugehen“* [Schulmeister2004].

Die Potentiale der Informations- und Kommunikationstechnologien für das Studium sind in der mediendidaktischen Fachliteratur (Kerres1998, Issing&Klimsa2002, Niegemann2004) ausführlich dargestellt. Als einige der wichtigsten Eigenschaften der neuen Medien werden genannt: Multimodalität (Ansprechen unterschiedlicher Sinneskanäle), Multikodierung (Nutzung unterschiedlicher Symbolsysteme) und Interaktivität der Lernenden mit dem medialen Angebot und mit anderen Lernenden [Issing2002].

Die Interaktivität und das Feedback spielen in offenen Lernumgebungen eine zentrale Rolle [Schulmeister2004]. Es wird dabei viel mehr didaktische bzw. kognitive Interaktivität bzw. Interaktivität mit Lernobjekten gemeint, als physikalische bzw. Steuerungsinteraktivität [Schulmeister1996a, 2004, Haack2002].

[Haack2002] betont die Funktionen der Individualisierung und der Motivierung als besondere Funktionen der Interaktivität. Die Auswahl und die Darbietung von Lerninformationen den Interessen und Bedürfnissen des Lernenden entsprechend in angemessener Form und individuellem Tempo sowie der aktive Einbezug des Lernenden in das Lerngeschehen werden durch interaktive Techniken möglich bzw. gefördert.

Interaktivität als Ausmaß, in dem eine Lernumgebung Interaktionen ermöglicht und fördert, wird in [Niegemann2004] definiert. Dabei werden als Funktionen von Interaktivität die motivationsfördernde (z.B. Kommentare), informationsliefernde (z.B. Hinweise auf Wissenslücken), Verstehen und Behalten fördernde (z.B. Hilfen unterschiedlicher Art, Verknüpfen mit anderen Inhalten), Anwenden/Transfer fördernde (Aufgaben- bzw. Problemstellungen), Lernprozess regulierende (z.B. Inhaltsübersichten) unterschieden. Als Formen dieser Interaktionen werden u. a. die Auswahl von Lerninhalten, das Bearbeiten von Aufgaben, Lernmaterialien, der Dialog, die aktive und passive Hilfen oder das Feedback genannt.

Da in virtuellen (computerbasierten) Lernumgebungen für individualisiertes Lernen das Lernen in erster Linie die kognitive Auseinandersetzung des Lernenden mit dem zu lernendem Inhalt (dem Lernobjekt) bedeutet, wird Interaktivität der Lernobjekte zu einem der entscheidenden Qualitätskriterien des Lernens und der Lernumgebungen.

Unter "Multimedia" wird die computerbasierte Integration verschiedener medialer Formen - Text, Grafik, Bild, Ton, Video, ... - durch Digitalisierung verstanden. „*Multimedia sind demnach technische Systeme, die verschiedenartige mediale Information verarbeiten und für den interaktiven Abruf vorhalten*“ [Kerres2002]. Nach [Schulmeister1996a] ist Multimedia einfach die Zusammenführung verschiedener Technologien, die vorher für sich alleine schon lange bekannt waren (z.B. Film, Musik, ...), die aber jetzt miteinander kombiniert werden und als Kombination das neue Sachgebiet Multimedia bilden. Als Hypermedia werden

*hochgradig miteinander verbundene multimediale Systeme bezeichnet*<sup>18</sup>. Für Bildungsprozesse ist jedoch nicht diese technische Integration interessant, sondern die Integration der Funktionen der medialen Formen, nämlich Präsentation, Simulation, Interaktion und Kommunikation (mehr dazu P.2.3.). „*Multimedia [ist] eine interaktive Form des Umgangs mit symbolischem Wissen in einer computergestützten Interaktion.*“ [Schulmeister1996a]. Interaktivität spielt somit auch in den multimedialen Lernumgebungen eine entscheidende Rolle.

Die Menschen haben bestimmte Kriterien für **Intelligenz** entwickelt (die verschieden gewichtet benutzt werden): Lernfähigkeit, logisches Denkvermögen, Abstraktionsvermögen, die Fähigkeit Problemlösungsstrategien zu entwickeln, Adaptionfähigkeit, Kommunikationsfähigkeit, etc. – eine eindeutige, präzise Definition der Intelligenz gibt es aber nicht [vgl. Goertzel1993].

Es lassen sich mehrere „Definitionen“ für Künstliche Intelligenz (KI) finden, die verschiedene Schwerpunkte des Forschungsgebietes betonen [Löscher1993, Nilsson1998, Haun2000, Duden2001]. Forscher auf dem Gebiet Künstliche Intelligenz versuchen z.B. (computerbasierte) Systeme zu entwickeln, die (1) das Verhalten zeigen, welches nach allgemeinem Verständnis als intelligent bezeichnet werden kann bzw. (2) „*do things that would require intelligence if done by men*“ [Haun2000].

Quellen der KI sind u. a. Psychologie, Neurobiologie, Informatik, Mathematik [Hellmich1997]. Viele Paradigmen der KI können nach ihren Vorgehensmodellen in zwei Gruppen organisiert werden: symbolische (wissensbasierte) KI und subsymbolische (verhaltensbasierte) KI [Nilsson1998, Haun2000, Lenzen2002]. Im Falle des symbolischen Ansatzes wird versucht intelligente Phänomene auf der Begriffsebene mittels logischen Schließens zu lösen: „*member of this family of approaches is one that uses logical operations applied to declarative knowledge bases*“. Im Falle des subsymbolischen Ansatzes wird versucht intelligentes Verhalten (von Robotern) z.B. mittels Neuronaler Netze zu erzwingen [Brooks1991a,b]. Eine Kombination dieser beiden Ansätze könnte wesentliche Fortschritte der KI-Forschung bedeuten [Haun2000, Nilsson1998].

In der vorliegenden Arbeit wird die intelligente Lernumgebung im Sinne einer wissensbasierten Lernumgebung verstanden.

In verschiedenen Büchern und Artikeln lassen sich mehrere Namen und Bezeichnungen für das computerunterstützte Lernen finden. Im Weiteren verwendete Namen werden an den entsprechenden Stellen erläutert.

---

<sup>18</sup> <http://beat.doebe.li/bibliothek/> mehrere Literaturhinweise (letzter Zugriff 10.02.2005)

## 2.2 Grundfragen der Lehr-/Lernprogramme

Der Computer als multifunktionales Medium kann einzelne Phasen des sehr komplexen Lerngeschehens, der Vorstellungs-, Denk-, Erkenntnis- und Bedeutungsprozesse bei Lernenden in effizienter Weise unterstützen. Dabei soll der Computer die Bildungsaufgaben immer dort unterstützen, wo eine Beschäftigung mit dem Lerngegenstand nicht so gut oder sonst gar nicht möglich ist z.B. bei Visualisierungen und Simulationen oder zur Durchführung aufwendiger Experimente. Der Computer kann als didaktisch-methodisches Hilfsmittel dienen und die Arbeit der Lehrer in sinnvoller Weise unterstützen. Bei der Erwachsenenbildung kann der Computer den Lehrer gegebenenfalls auch ersetzen [vgl. Coleman1998].

*„In jeder Bildungssoftware schlägt sich ein theoretisches Lernmodell nieder. Egal ob dieser theoretische Ansatz nun von den AutorInnen auch tatsächlich expliziert worden ist oder nicht, spiegelt die Lernsoftware – angefangen vom behandelten Thema über den Aufbau bzw. die Struktur des Softwarepaketes bis hin zur Benutzeroberfläche des Lernprogramms – ein pädagogisches und didaktisches Modell wider, das in ihr implementiert wurde.“*  
[Baumgartner1997]

Es existieren viele Vorstellungen davon, wie der menschliche Lernprozess stattfindet. Diese Vorstellungen sind in verschiedenen Lerntheorien zusammengefasst [Lerntheorien, Strittmatter2000, Sweller1999, WK1996].

Nach [Jarz1997] lässt sich die grundlegende Unterscheidung der Lerntheorien auf die gegensätzlichen Annahmen des Objektivismus und des Subjektivismus reduzieren. Lernen als Abbilden versteht den (objektiven) Lernenden als *„unbeschriebenes Blatt, das es zu füllen gilt“*. Lernen als Konstruieren sieht den (subjektiven) Lernender *„als Kerze, die entzündet werden kann“*. Das objektivistische Paradigma geht von einem objektiven Wissen aus, mit dem sich die Welt und das Geschehen in ihr weitgehend erklären lässt. Dieses Wissen ist relativ stabil und lässt sich gut strukturieren. Personen übernehmen dieses Wissen in einem Lernprozess quasi 1:1, da es ein Abbild der Realität ist. Das subjektivistische Paradigma nimmt die gegenteilige Position ein: objektives Wissen gibt es nicht, jeder Mensch konstruiert oder interpretiert die Wirklichkeit aufgrund seiner eigenen Erfahrungen zu einer eigenständigen Bedeutung, also einem einzigartigen Wissen. Deshalb versteht jeder Mensch die Realität etwas anderes, nämlich so, wie er sie aus seinen Erfahrungen und aus dem Dialog mit der Umwelt und anderen Menschen aufbaut (konstruiert). Wissen kann deshalb nicht passiv übernommen werden. Die einzelnen Paradigmen sind eher als idealtypisch anzusehen.

Die drei einflussreichsten Theoriesysteme des letzten Jahrhunderts waren Behaviorismus (I. Pawlow, E. Thorndike, B. Skinner), Kognitivismus (mehrere Ausprägungen) und Konstruktivismus [Baumgartner1999].

Behavioristische Lehrstrategien gehen davon aus, dass Lehrende wissen, was die Lernenden zu lernen haben. Lernen wird als konditionierter Reflex gesehen, der durch Adaptation erworben wird. Die im Gehirn ablaufenden Prozesse interessieren den Behaviorismus nicht: das Gehirn wird als Box gesehen, die auf einen Input deterministisch reagiert. Dieses Theoriesystem ist nicht an kognitiven Steuerungsprozessen sondern vor allem an Verhaltenssteuerung interessiert. Der Behaviorismus erzielt Erfolge beim Trainieren von automatischen, scheinbar unbewussten Ausführungen verschiedener Fertigkeiten (insbes. körperlicher), kann aber die Komplexität menschlicher Lernprozesse nicht erklären [Baumgartner 1999].

Im Gegensatz zum Behaviorismus sieht der Kognitivismus das menschliche Hirn nicht mehr als passiven Behälter, sondern als Informationsverarbeitungssystem und versucht für die Verarbeitungsprozesse theoretische Modelle zu entwickeln [z.B. VanLehn1988, Anderson1989, Möbus1995,1996]. Daher gibt es einen engen Zusammenhang zwischen Kognitivismus und KI. Die Art des Lernens, die hier im Mittelpunkt steht, ist das Problemlösen, welches von einem Tutor begleitet wird.

Die kognitionswissenschaftlich orientierte ISP-DL-Theorie (Impasse-Success-Problem-Solving-Driven-Learning) des Wissenserwerbs beim Problemlösen dient als Basis für die Implementierung der intelligenten Problemlöseumgebung mileET und wird daher näher erläutert.

Die ISP-DL-Theorie [Möbus1995,1996] geht davon aus, dass das Wissen des Lernalers beim Lösen von Problemen ständig deduktiv und induktiv verändert wird. ISP-DL basiert auf den kognitionswissenschaftlichen Theorien von Newell [Newell1990] und VanLehn [Vanlehn1988] (impass-driven-learning), Anderson [Anderson1986, Anderson1989] (success-driven-learning), sowie der motivationsorientierten Rubikontheorie von Heckhausen [Heckhausen1989] und Gollwitzer [Gollwitzer1990] (problem solving phases). Von deduktiver Wissensveränderung bzw. -optimierung wird gesprochen, wenn nach erfolgreicher Problemlösung das Wissen so reorganisiert wird, dass der Lerner in Wiederholungen schneller und schweigsamer wird. Dagegen wird von induktiver Wissensveränderung bzw. Wissenserweiterung gesprochen, wenn dem Lerner Wissen für den weiteren Lösungsprozess fehlt, d.h. eine Stocksituation (=Impasse) auftritt und er diese mittels schwacher Heuristiken (wie Peers um Hilfe fragen, Rumprobieren, Hypothesentesten, durchgerechnete

Lösungsbeispiele aus Textbüchern selbst erklären, etc.) überwinden kann. Wichtig für den Entwurf konstruktivistischer interaktiver und adaptiver Lernumgebungen ist die Überwindung der individuellen Stocksituationen. Die Systemleistungen müssen so konzipiert werden, dass der Lerner seine schwachen (wissensarmen) Heuristiken (wie z.B. das Testen von Hypothesen) im Rahmen des eLearning-Systems einsetzen kann. Systeme, die den Heuristikeinsatz unterstützen, sind notwendigerweise wissensbasiert.

Nach der ISP-DL-Theorie wird der Wissenserwerb insbesondere durch Lerneraktivitäten wie selbständiges Formulieren von Hypothesen, autonome Prüfkativitäten und Selbsterklärungen von systemgenerierten Hilfen und Hinweisen gefördert. Diese Art des Lernens soll in den IPSEs ermöglicht werden [Möbus1995, 1996, 2002, 2003a].

Im konstruktivistischen Ansatz wird Lernen als ein aktiver Prozess gesehen, bei dem Menschen ihr Wissen aufgrund ihrer früheren Erfahrungen konstruieren. Im Gegensatz zum Kognitivismus steht im Konstruktivismus nicht das Lösen bereits existierender Probleme im Vordergrund, sondern das eigenständige Generieren von Problemen. Der Lehrer nimmt dabei die Rolle eines Moderators ein.

*„Von der Lerntheorie des Behaviorismus geprägte Lernprogramme haben den Charakter einer starren Paukmaschine...Kognitivistisch geprägte Programme bieten einen geleiteten Einstieg in ein Thema und zeigen dem Lerner Zusammenhänge und Vorgehensweisen auf... Das konstruktivistische Lernsystem hat nicht mehr führende und anleitende Funktion, sondern bietet Anlässe, Anregungen, Hilfen, es ist Berater und Begleiter.“* [Mülchen2001]

Der Strategie der reinen Instruktion liegt folgendes Modell zugrunde: Es gibt einen definierten Programmein- und ausstieg, die Lerneinheiten sind entlang vorgestellter Lernwege angeordnet, zur jeweils nächsten Lerneinheit gelangt man nur bei richtiger Lösung der vorhergehenden Lerneinheit. Ist die Lösung nur teilweise richtig oder ganz falsch, sind entsprechende Wiederholungsschleifen durchzuarbeiten oder es wird auf einen Lernweg niedrigeren Niveaus verzweigt.

Die Strategie der Konstruktion wird in hypermedialen Strukturen der Lern- und Informationseinheiten realisiert und unterstellt einen Lernenden, der in der Lage ist, sein Lernen von Anfang bis Ende selbständig zu organisieren: Programmein- und -ausstieg werden vom Lernenden ebenso selbst bestimmt wie der Weg und die Geschwindigkeit durch die Lerneinheiten und die Bewertung seines Lernerfolgs. Da Lernende damit durchaus leicht überfordert sein können, werden Navigationshilfen und verschiedene Lernwege vorgeschlagen [Haack2002].

Eine Reihe von Lern-Theorien und Modellen lassen sich als Theorien und Modelle des Instruktionsdesigns einordnen [Niegemann2004, Strittmatter2000, Sweller1999]. Die Grundidee dieser Fachdisziplin besteht in der „... Konzeption, die versucht, für unterschiedliche Kategorien von Lernaufgaben und unterschiedliche Lernvoraussetzungen und Rahmenbedingungen die jeweils (relativ) bestgeeignete Lernumgebung zu finden“ [Niegemann2004]. Aus verschiedenen Modellen des Instruktionsdesigns findet sich in computerbasierten Lernumgebungen für GET in der Ingenieurausbildung auch das Goal-Based-Scenario [Butz2000]. Im Rahmen einer Coverstory soll der Lernende verschiedene Missionen erfüllen, um bestimmte Lernziele zu erreichen. Dabei wird er durch seine Handlungen sein Wissen anwenden bzw. neues Wissen erwerben. Theoretischer Hintergrund sind die Forschungsarbeiten von R. Schank [Schank1998].

Es lassen sich viele verschiedene Klassifikationen der Lehr-/Lernprogramme in der Literatur und im Internet finden. Unter Berücksichtigung verschiedener Bewertungskriterien (z.B. Funktionen im Unterricht, Zielsetzung, Einsatzform) wurden mehrere Kategorien der Lehr-/Lernsoftware ausgearbeitet [Schulmeister1996a, Gorny1991, LfSW1995, Muelchen2001].

Hinsichtlich des Ziels des Einsatzes in der GET-Umgebung lassen sich Software-Produkte wie folgt klassifizieren: A) Vorlesungs-Präsentationen, B) (interaktive) eLernbücher, Lernprogramme (Lernmodule), C) Kalkulations- und Simulationstools, D) Entwicklungstools.

- A) Als Präsentationen dienen (kommentierte) Visualisierungen aller Art [vgl. z.B. Millard2003] und sind relativ einfach zu implementieren.
- B) Interaktive eLernbücher bieten theoretische Informationen, Aufgabenstellungen und Experimente an und sind für selbständiges Lernen am Computer gedacht.
- C) Lernprogramme sind meistens eigene Entwicklungen und ermöglichen die Bearbeitung eines begrenzten Themenkreises. Im Gegensatz zu den eLernbüchern ist der Lesestoff hier (wenn er überhaupt vorhanden ist) ein Hilfsmittel. Die Lern-Qualität solcher Lernprogramme ist sehr vom Maß an Interaktivität und Adaptivität abhängig. Einfach realisierbare Testmodule mit Multiple-Choice-Aufgaben und komplexe ITS<sup>19</sup> lassen sich unter dieser Kategorie unterbringen.
- D) Kalkulations- und Simulationstools sind: (1) Computer-Algebra-Systeme (CAS) zur Visualisierung und Analyse von komplexen mathematischen Modellen, (2) Computer-Aided-Design (CAD)-Systeme zur Analyse, Simulation und Konstruktion von Schaltbildern, (3) spezielle Lösungen, mit denen sich einige konkrete Probleme untersuchen lassen. CAS und CAD sind sehr leistungsfähig und in der Regel sehr teuer.

---

<sup>19</sup> intelligent tutoring systems

Die gratis angebotenen Studentenversionen von CAD-Systemen haben eine sehr eingegrenzte Elementenbasis, was allerdings für GET-Ausbildungszwecke ausreichend ist. Einige spezielle Lösungen sind auch in Form von Applets abrufbar [z.B. Ford2001, Rustom2004, needs]. Simulationen ermöglichen u. a. durch das Testen eines Objekts die Grenzen der Anwendbarkeit eines Modells zu erforschen. Sie können eine Grundlage für konstruktivistisch orientierte Lernumgebungen bilden [Pillage1993]. Simulatoren können numerischer oder symbolischer Art sein [Heulsman1996, Luchetta2001]. Letztere erzeugen zu einzelnen Parametern der Schaltung nicht einen „nackten“ numerischen sondern einen symbolischen Wert als Funktion von anderen Schaltungsparametern. Dieser ist wesentlich informativer für den Lernenden und kann dadurch ein besseres Verständnis für elektrische Schaltungen ermöglichen.

- E) Entwicklungsumgebungen (z.B. Java IDE, Visual C, Visual Basic) dienen zur Implementierung eigener Lernprogramme. Autorensysteme (wie z.B. Asymetrix ToolBook oder Micromedia Director) eignen sich für eine relativ schnelle Implementierung von multimedialen Lernmodulen, sind aber nicht für Implementierung berechnungsintensiver Lerneinheiten geeignet.

Die Fachqualität von Lernsystemen ist durch die Fähigkeiten und Erfahrungen der Fachautoren hinsichtlich der Auswahl der Themen und der Form für eine Lerneinheit, des Angebots der Aufgaben, Beispielen, Kommentare und Rückmeldungen bestimmt.

Da die Kategorien (C) und (D) ein breites Spektrum von Programmklassen abdecken, wird hier auf die einzelne Programmtypen eingegangen.

CBT oder CAI<sup>20</sup> beinhalten *"alle Anwendungen, die der Vermittlung, Einübung, Prüfung und Bewertung von Wissen, Kenntnissen und Fähigkeiten dienen"* [Duden2001]. Es handelt sich bei diesen Programmen um programmierte Lernkurse, wofür sich auch der Begriff Courseware etabliert hat. CBT-Programme sind sowohl für die Unterstützung herkömmlicher Ausbildungsmethoden als auch für das Selbststudium geeignet. Der zu vermittelnde Stoff wird dazu in bestimmte Lehreinheiten eingeteilt, die von jedem Lernenden in seinem individuellen Arbeitstempo durchgearbeitet werden können.

Drill-And-Practice Programme haben eine sehr starre Ablaufstruktur. Nach kurzen Informationseinheiten wird abschließend eine Frage gestellt. Das System bewertet die Antworten im einfachsten Fall nur mit richtig oder falsch und wechselt dann zum nächsten Lernabschnitt. Erläuternde Kommentare oder Lösungshinweise seitens des Systems gibt es kaum.

---

<sup>20</sup> computer-assisted instruction

Es gibt Programme, die fehlerhaft beantwortete Fragen in leicht modifizierter Form noch einmal stellen und/oder vor einer erneuten Fragestellung Lösungshilfen geben. Am Ende einer Lerneinheit erhält der Lernende einen Überblick über seinen aktuellen Leistungsstand. Hier wird das Wissensgebiet sorgfältig aufbereitet, und der Lernende wird vom System durch dieses Wissensgebiet geführt und so beim Lernen unterstützt. Ein typisches Beispiel stellen die Lernprogramme zur Sprachausbildung dar.

ITS oder ILE<sup>21</sup> und Adaptive Lernumgebungen (engl. Adaptive Learning Environment, Adaptive Hypermedia) sind die Forschungsgebiete der KI und eLearning (AIED, ITS, User Modelling u.a.). ITS sind die Expertensysteme für das Lernen (mehr dazu im Abschnitt 2.3). Charakteristisch für ein ITS sind Wissensbasis, Tutor- und Lernermodelle. Die Wissensbasis enthält domänenspezifisches Wissen, das Lernermodell sammelt benutzerspezifische Daten und das Tutormodell bestimmt aufgrund der Daten in dem Benutzermodell bzw. aufgrund der Benutzereingaben den Lernweg und die Art der Präsentation des Lernstoffes. Die Benutzermodellierung erlaubt lernerbezogene Hilfen und Rückmeldungen.

In Simulationen wird ein Modell der realen Umwelt erstellt, so dass der Benutzer Abläufe und Handlungen am Rechner wirklichkeitsgetreu beobachten bzw. trainieren kann. Bekannte Simulationsprogramme sind z.B. Flug- oder Fahrsimulatoren. Schaltungssimulationstools sind Elektrotechnik/Elektronik-spezifische Simulationstools. In modernen Versionen dieser Softwaresysteme werden den Benutzern auch Schaltungseditoren zur Verfügung gestellt. Schaltungssimulatoren können numerischer und symbolischer Art sein. Die bekanntesten numerischen Simulationstools sind PSpice<sup>22</sup>, CircuitMaker<sup>23</sup> oder Multisim Electronics Workbench<sup>24</sup>. Sie werden seit mehreren Jahren auch in der Lehre eingesetzt [FIE, NITE]. Weniger bekannt bzw. selten in der Lehre sind symbolische Simulationstools [Gielen1994, Heulsman1996, Luchetta2001]. Bei der Entwicklung symbolischer Simulatoren werden verschiedene KI-Techniken eingesetzt [Gielen1994, Kuipers1994, Romero1997].

In einer Lernumgebung können alle diese Arten von Lernsoftware interagieren bzw. in verschiedenen Formen der Zusammenarbeit eingesetzt werden (siehe Kap. 4 und 5).

Ein anderer Ansatz zur Klassifikation von Lehr-/Lernsoftware wird in [Reeves1997] vorgestellt. Dabei wird keine Zuordnung der Lernprogramme zu bestimmten Kategorien vorgesehen, sondern eine Positionierung der Ausprägung bestimmter Eigenschaften auf

---

<sup>21</sup> Intelligent Learning Environment

<sup>22</sup> <http://www.pspice.com/> (letzter Zugriff 10.02.2005)

<sup>23</sup> <http://www.altium.com/circuitmaker/> (letzter Zugriff 10.02.2005)

<sup>24</sup> <http://www.electronicsworbench.com/> (letzter Zugriff 10.02.2005)

entsprechenden Skalen. Es werden vierzehn pädagogische Dimensionen in Lernprogrammen unterschieden, welche als Grundlage für eine Klassifikation bzw. Evaluation dienen:

- 1) Epistemology (Objectivism↔Constructivism),
- 2) Pedagogical Philosophy (Instructivist↔Constructivist),
- 3) Underlying Psychology (Behavioural↔Cognitive),
- 4) Goal-Orientation (Sharply-focused↔Unfocused),
- 5) Experimental Validiy (Abstract↔Concret),
- 6) Teacher Role (Didactic↔Facilitative),
- 7) Flexibility (Teacher-Proof↔Easiliy Modifiable),
- 8) Value of Errors (Errorless Learning↔Learning from Experience),
- 9) Origin of Motivation (Extrinsic↔Intrinsic),
- 10) Accomodation of Individual Differences (Non-Exisent↔Multi-faceted),
- 11) Lerner Control (Non existent↔Unrestricted),
- 12) User Activity(Mathemagenic↔Generative),
- 13) Cooperative Learning (Unsupported↔Integral),
- 14) Cultural Sensitivity (Non-existent↔Integral).

Aufgrund einer solchen "Parametrisierung" entstandene Eigenschaften-Karten ermöglichen es, einzelne Lernprogramme in die Landschaft der Lehr-/Lernsoftware einzuordnen.

### 2.3 Multimedia-Einsatz in der Lehre

*„Hyper- und multimediale Lernprogramme sollen durch Visualisierungen und Animationen eine Reihe von erwünschten Effekten auf der Lernerseite erzielen: (1) Erhöhung der Neugiermotivation, (2) vertieftes Verständnis für komplexe, dynamische Inhalte, (3) stärkerer Praxis- und Berufsbezug in Haupt- und Nebenfachausbildung. Die webbasierten eLearning-Systeme<sup>25</sup> ermöglichen die Nutzung vorhandener Ressourcen mehrerer Hochschulen.“*  
[Yakimchuk2003]

Die Funktionen des Multimedia-Einsatzes werden im Bezug auf das Lehr-/Lernmodul „Fourier-Reihen: Netzwerke bei sinusförmiger periodischer Erregung“ diskutiert.

#### Präsentation des Lehrstoffes

Ein entscheidender Vorteil multimedialer Lernsysteme ist die Möglichkeit, dynamische Medien wie Filme, Animationen oder Töne in die Lehreinheiten einzubinden. Es ist zu

---

<sup>25</sup> thematisch relevante Beiträge sind z.B. im Programm des Workshops „eLearning: NMB-Projekte in den Ingenieurwissenschaften“ zu finden: <http://193.25.43.240/nmb-workshop> (letzter Zugriff 10.02.2005)

beachten, dass zur Erläuterung eines Konzeptes eingesetzte textuelle und visuelle Medien aufeinander angepasst werden sollten [Riehm1995]. Solche Anforderungen wie „die Bilder verdeutlichen das gelesene visuell“ oder „Kommentartexte verdeutlichen die Bilder und gesprochene und geschriebene Kommentare stimmen überein“ werden in allen kommentierten Animationen in dem Modul „Fourier-Reihen“ berücksichtigt. Das Lernprogramm bietet verschiedene Medienformen an, so dass sich der Benutzer die Form aussuchen kann, die er für sich am wirkungsvollsten hält. Auf diese Art kann er den Lernprozess so gut wie möglich an sich selbst anpassen.

Eine besondere Rolle spielen ausgearbeitete Beispiele bzw. Lösungsblätter. Die besondere Bedeutung der Beispiel-basierten Instruktion für Studierende wird in [Davidovic2003, Renkl2000] diskutiert. Die erklärten oder (besser) diskutierten Beispiele und Problemstellungen sind für Lehrgebiete wie Elektrotechnik, Physik, Mathematik (high element interactivity [Sweller1999]) von großer Bedeutung – das Verstehen und Erlernen komplexer Zusammenhänge werden dadurch unterstützt. Ausgearbeitete Beispiele und Lösungsblätter mit verschiedenen Hinweisen und Kommentaren werden in allen Lernmodulen der Lernumgebung GETsoft angeboten. Die meisten sind in Textform vorbereitet, viele aber auch als interaktive Mathcad-Arbeitsblätter realisiert, sodass für die Studierenden die Möglichkeit besteht, ihre „eigenen“ Probleme aufbauend auf den vorbereiteten Dateien zu erforschen und zu lösen.

### **Anregung und Aktivierung**

Beim Lernen mit dem Computer besteht für die Lehrenden die Gefahr des „Abschaltens“. Während im Seminar der Tutor mit den Teilnehmern ständig im Kontakt bleiben kann und sie somit „wach hält“, muss diese Funktion im Falle des Lernprozesses am Computer von dem Lernmodul übernommen werden. Das notwendige Bedienen von Tasten bzw. das monotone Drücken einer Taste zum Weiterkommen im Programm erfüllt die Forderung nach Aktivierung nicht und kann sogar ermüdend wirken. Ebenso die Vielfalt von Formen und Farben kann keine dauerhafte Anregung gewährleisten. Die Aktivierung muss sich auf den Lerngegenstand beziehen, auf das so genannte Lernobjekt. Dies wird grundsätzlich am besten durch ein möglichst großes Maß an Interaktivität im Lernsystem und Selbststeuerung bei der Programmbedienung erreicht. Eine Vielfalt an Aufgaben, Experimenten, Animationen und die Praxisvideos sowie deren Einbindung in das Kompendium tragen dazu bei, diese Anforderungen in GETsoft Lernumgebung zu erfüllen. Wissen wird nicht rein theoretisch und

abstrakt vermittelt, sondern es wird Praxisnähe angestrebt. Dies geschieht vor allem durch eine praxisbezogene Formulierung von Aufgaben und praxisnahe Beispiele.

### **Navigation**

Die einfache Navigation durch ein Lernsystem ist von großer Bedeutung, denn jedes Lernsystem ist unbrauchbar, wenn viel Zeit und Energie des Benutzers dadurch verschwendet wird, sich durch ein kompliziertes Netzsystem zu steuern.

Der Benutzer kann in dem GETsoft System frei navigieren, dabei wird auf standardmäßigen Navigationsfunktionen eines Browsers zurückgegriffen. Es wurde keine Benutzer-spezifische Verwaltung von Links bzw. Pfade des Benutzers außerhalb des im Browser vorhandenen Funktionsumfangs implementiert. Als Orientierungshilfe wird eine Markierung der Lage im Programm angeboten (Programmteil, Kapitel, Abschnitt u. ä.) und ein Zugriff auf alle Teile des Lernprogramms ist jederzeit über das Menü-(Inhaltverzeichnis)-Frame möglich. Es lässt sich allerdings einiges in den aktuellen online Versionen von GETsoft Lernmodulen kritisieren [Aslanski2002].

### **Interaktivität und Selbstkontrolle**

Viele Autoren [vgl. Riehm1995, Schulmeister1996a, Tulodziezki2004] betonen, dass der Mensch am besten durch das selbständige Bearbeiten eines Themas, durch das Lösen von Aufgaben oder das Einbringen eigener Lösungsvorschläge Sachverhalte lernt. Man spricht auch vom "Learning by Doing".

Dies ist auch der Grund, weshalb der Begriff Interaktion bzw. Interaktivität bei Lernsoftware eine sehr entscheidende Rolle spielt. Der Benutzer sollte so gut wie möglich aktiv in den Lernprozess mit einbezogen werden, denn so bleiben bei ihm die Aufmerksamkeit und damit auch die Konzentration auf den Inhalt erhalten. Der Lernende sollte ständig in eine Art Kommunikation mit dem System verwickelt sein und nicht teilnahmslos vor dem Bildschirm sitzen und „*sich mit Fakten berieselnd lassen*“. [Schulmeister1996a] spricht dabei von der sogenannten "Lernerkontrolle" des Benutzers über das System. Der Lernende kontrolliert das Geschehen und wird dadurch zu eigenen Denkkaktionen herausgefordert. Dies fördert seine Motivation und damit auch die Lernwirksamkeit.

Ein Angebot von Beispielen und Test-Fragen im theoretischen Abschnitt des Programms sollen der Erhöhung der Aktivität des Benutzers dienen und die Arbeit mit dem Lernprogramm interessanter machen.

## 2.4 Einsatz der Techniken der Künstlichen Intelligenz in Lernumgebungen

Da die Implementierung der IPSE mileET auf die Techniken der wissensbasierten KI zurückgreift (Abschnitt 2.1.), werden im Weiteren einige KI-Grundlagen dargestellt.

Zentrale Methoden der wissensbasierten KI sind: Wissensdarstellung (-repräsentation), Wissensverarbeitung, Problemlösungsverfahren, Suchverfahren, Inferenzmechanismen, Wissensakquisition, maschinelles Lernen. KI-Systeme sind Computerprogramme mit einer bestimmten Assistenzfunktion, d.h. sie haben ihre Kompetenzgrenzen. Wissensbasierte Systeme erfüllen die Aufgabe, Wissen - insbesondere das Wissen von Experten - in einem Computerprogramm zu repräsentieren (in symbolischer Form) und dieses Wissen zur Unterstützung von Anwendern zu nutzen.

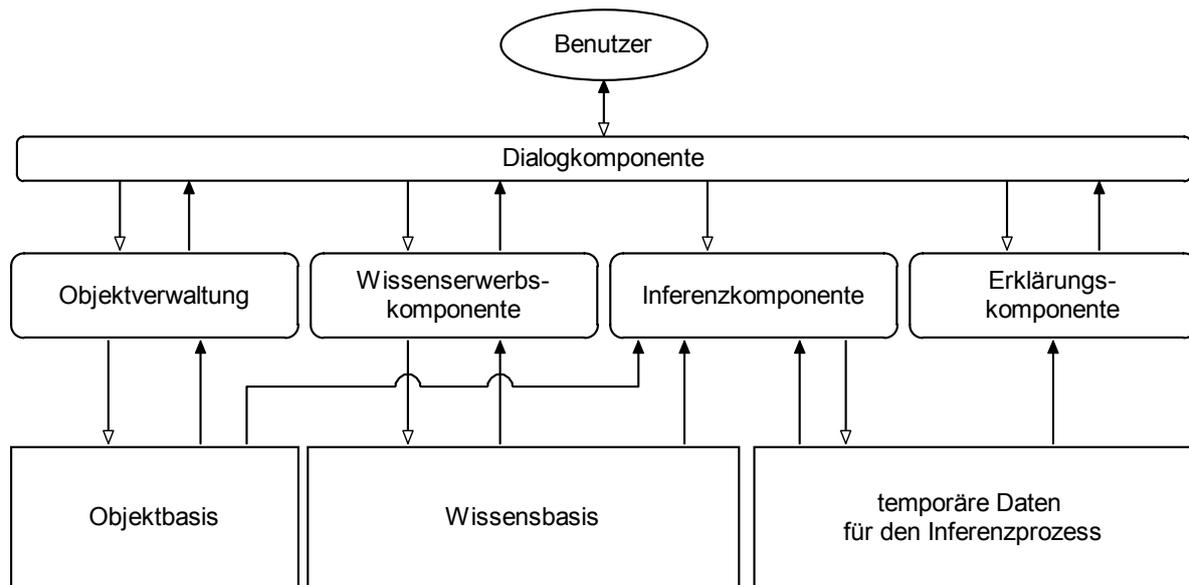
Ein wissensbasiertes System wird dadurch gekennzeichnet, dass Wissen explizit in einer Wissensbasis erfasst ist und die Verarbeitung dieses Wissens durch einen dafür konstruierten Inferenzmechanismus erfolgt.

Unter einem Expertensystem wird ein System verstanden, das *die Leistungsfähigkeiten des Rechners mit Wissen und Erfahrungen eines Experten in so einer Form vereinigt, dass das System einen Rat geben kann oder eine Aufgabe lösen kann* [Hellmich1997]. Expertensysteme sind fachspezifisch, können bei unsicheren Fakten überlegen, können Entscheidung verständlich und logisch begründen, sind erweiterbar, (basieren auf Regeln), geben als Antwort einen eindeutigen Rat, können lernfähig sein.

Die Umsetzung menschlichen Wissens in Computersystemen wird als *knowledge engineering* bezeichnet. Bei der Art des Wissenserwerbs unterscheidet man den direkten und indirekten Wissenserwerb. Beim indirekten Wissenserwerb gibt ein Wissensingenieur das Fachwissen eines Experten, den er zuvor befragt hat, am System ein. Direkter Wissenserwerb bedeutet, der Experte gibt sein Wissen selbst am System ein.

Die Wissensmodellierung ist einer der wichtigsten Aspekte eines wissensbasierten Systems. Aus den Anforderungen an ein Expertensystem ergeben sich die wesentlichen Komponenten eines solchen Systems, ein gespeichertes, umfangreiches Wissen und Mechanismen, mit diesem Wissen zu arbeiten.

Einen Überblick über die Architektur eines Expertensystems vermittelt die Abbildung 2 [Forsyth1987, Flor1996, Hellmich1997].



**Abbildung 2. Beispiel einer möglichen Basisarchitektur eines Expertensystems**

Die Dialogkomponente wird vom Benutzer dazu benutzt, gespeichertes Wissen abzurufen, bzw. mit Hilfe des Systems zu einer Problemlösung zu gelangen. Sie dient auch zur Kommunikation mit dem menschlichen Experten. Diese Komponente enthält oft mehr als die Hälfte des erforderlichen Codes des Gesamtsystems. Es ist sehr aufwendig, Regeln, Fakten und Objekte, Anfragen, Ergebnisse, Erklärungen und Inferenzabläufe auf grafisch ansprechende Weise darzustellen und die Eingabensteuerung zu verwalten.

Die Objekt-Verwaltung steuert die Eingabe und Manipulation von Klassen, von Objekten und Methoden. Über die Wissenserwerbkomponente gibt der Experte bzw. der Wissensingenieur das Wissen in das System bzw. in die Wissensbasis ein. Dabei wird das Wissen soweit wie möglich auf Konsistenz und auf logische Plausibilität geprüft.

Die Wissensbasis enthält fachspezifisches Wissen. Wie dieses Wissen in der Wissensbasis dargestellt wird, ist von der gewählten Wissensrepräsentationsform abhängig. Die Wissensbasis enthält auch Wissen über Wissen (Meta-Wissen), z.B. Heuristiken über die effektivste Strategie bei der Inferenz, und Fall-Wissen.

Die Inferenzkomponente enthält die Verfahren und Mechanismen, um das in der Wissensbasis gespeicherte Wissen zu verarbeiten. Je nach Aufgabe und Funktion des Systems können ein oder mehrere Inferenzstrategien implementiert werden.

Die dynamische Datenbasis umfasst alle Daten, welche temporär während des Inferenzprozesses anfallen.

Die Erklärungskomponente stellt nach Ablauf der Inferenzmechanismen den Lösungsweg dar. Eine weitere Aufgabe der Erklärungskomponente ist die Begründung von Systemanfragen im Verlauf der Lösungssuche. Diese Komponente soll auf folgenden Fragen antworten: wie kam

das gefundene Ergebnis zustande, welche Regeln wurden dabei benutzt bzw. nicht benutzt, gab es Fakten, die besonderen oder keinen Einfluss auf das Ergebnis hatten?

Die Wissenserwerbskomponente, Dialogkomponente und Erklärungskomponente werden auch unter dem Begriff Problemlösungskomponente zusammengefasst.

Die Entwicklung eines Expertensystems kann auf unterschiedlichen Ebenen erfolgen. Zum einen ist es möglich, ein solches System in jeder beliebigen Programmiersprache höheres Niveaus zu erstellen, zum anderen bieten sich spezielle KI-Sprachen an, wie z.B. Prolog [Bratko1990, O'Keefe1990, Sterling1988] oder Lisp. Weitere Möglichkeiten bestehen in der Verwendung angebotener Expertensystemwerkzeuge, Wissensrepräsentationssprachen oder Expertensystemshells [Murray1999].

Intelligente tutorielle Systeme (ITS) sind Expertensysteme für das Lehren. In ITS werden KI-Techniken für Wissensrepräsentation, Wissensverarbeitung und Benutzermodellierung (Studentenmodell, Tutormodell, Expertenmodell) eingesetzt [Beck1996]. Jede dieser Problemstellungen ist ein eigenes Forschungsthema (User Modeling, AIED, ITS).

*“The goal of various ITS is the use the knowledge about the domain, the student, and about teaching strategies to support flexible individualized learning and tutoring...”* und *“three core ITS technologies: curriculum sequencing, intelligent analysis of student's solutions, and interactive problem solving support”* [Brusilovsky1999].

Mit der Entwicklung der Netzwerktechnologien sind die Problematik der Verteilung intelligenter Komponente im Netz, sowie die der Entwicklung adaptiver webbasierter intelligenter Hypermedia-Systeme und Anwendungen zu neuen Forschungsschwerpunkten geworden [Devedzic2001, Brusilovsky2001].

Das Fachwissen wird in einem Experten-Modell aufgefasst, die Lehrstrategien und -techniken in einem Lehrer- bzw. Tutor-Modell und Informationen über den jeweiligen Benutzer in einem Benutzer- bzw. Studenten-Modell verwaltet.

Die Benutzermodellierung in ITS dient sowie der Organisation der Lehrmodule (*curriculum sequencing*) als auch adaptiver Lernerunterstützung beim Aufgabenlösen (*interactive problem solving support*). Für beide Ziele ist eine intelligente Analyse der Benutzertätigkeit notwendig (*intelligent analysis of student's solutions*). Die dabei gewonnene Information wird in einem Benutzermodell gespeichert. Aufgrund dieser Information und der Daten aus den Lehrer/Tutor-Modellen wird ein intelligentes Verhalten des Programms möglich. Verschiedene Techniken zur Benutzermodellierung werden in modernen Programmen eingesetzt [Holt1994].

IPSEs stellen eine spezielle Art der ITS dar. Sie unterstützen den Lernenden beim aktiven Problemlösen bezogen auf eine festgelegte Aufgabenmenge. Adaptive Hilfestellung und Lösungshinweise werden von einem in das System integrierten wissensbasierten Assistenten (spezielle Art des Expertensystems) generiert. Der Assistent analysiert und bewertet die vom Benutzer formulierten Hypothesen über die Korrektheit vollständiger Lösungsentwürfe und Einbettbarkeit von unvollständigen Lösungsentwürfen. Er hat im Allgemeinen auch eine generative Fähigkeit. Dadurch ist er in der Lage, unvollständige Lösungsentwürfe weiterzuentwickeln und korrekt zu vervollständigen. IPSEs wurden bisher für viele Domänen (funktionale Programmierung, Modellierung verteilter Systeme, pneumatische Schaltungen, elektrische Steuerungstechnik, Kunststoffherstellung, Planspiele, etc.) entwickelt und auch in eLearning erfolgreich eingesetzt [Möbus1995, 1996, 2002, 2003, Willms1997].

Der Hypothesentesten-Ansatz und die Aufgaben- bzw. Übungsbasierung der IPSEs erlauben eine adaptive Benutzerunterstützung ohne Benutzermodellierungskomponente. In mileET wird allerdings eine einfache Form des „student modelling“ implementiert – das „model tracing“ [vgl. Anderson1991, Corbett2000, VanLehn2000]. Bei dieser Methode wird die Entscheidung über weitere Programmabläufe aufgrund der Benutzereingaben getroffen. In mileET werden keine benutzerrelevanten Daten gespeichert und über mehrere Sessions verwaltet (was für Benutzermodellierung typisch wäre), ausgenommen die Funktion der Speicherung des Bearbeitungsstandes für jede Aufgabe. Aber auch in diesem Falle wird eher eine Aufgabenrelevanz als eine Benutzerrelevanz gemeint.

Die Stärke von IPSEs liegt somit in *interactive problem solving support*. Sequenzierung der Lehrinhalte (Lesestoff oder Aufgabenfolge) wird im ITS-Sinne nicht implementiert. Verweise auf die aufgabenkontextrelevanten externen Lernmodule sind möglich, die Inhalte dieser Module werden aber nicht durch mileET zusammengestellt.

Ein weiterer Forschungsaspekt in AIED sind natürlichsprachliche Dialoge in Lehrsystemen [z.B. Jordan2003].

Funktionen und Formeln sowie technische Zeichnungen bzw. Diagramme sind wichtige Elemente der Fachsprache der Elektrotechnik-Domäne (im Vergleich z.B. zu geisteswissenschaftlichen Domänen). Lösungen vieler GET-Aufgaben lassen sich nur über Formeln und Diagramme darstellen. Aufgrund dessen und aufgrund der ausgewählten GET-Themen wurde in mileET auf eine Dialogkomponente verzichtet (vgl. Kapitel 3). Viel wichtiger ist die mathematische Unterstützung der Benutzer. Eine Dialog-Komponente wäre allerdings unumgänglich beim Unterstützen der Lösungen von qualitativen Aufgaben (vgl. z.B. [Jordan2003]).

Zum Abschluss dieses Kapitels wird eine kurze Darstellung der Techniken gegeben, die in mileET eingesetzt worden sind: Objekt-Orientierte Programmierung, Prädikatenlogik, XML. Objektorientierte Programmierung (OOP) bedeutet, das zu lösende Problem durch Objekte zu modellieren, die miteinander in Beziehung stehen [Booch1998, Hellmich1997]. Das Grundprinzip der Objektorientiertheit besteht darin, dass alles Wissen, sowohl über die Struktur als auch über das Verhalten eines Objektes, beim Objekt selbst bzw. seiner Klasse liegt und nicht beim Anwender dieses Objektes. Objekte fassen Eigenschaften (Variablen) und Verhalten (Methoden) zusammen. Objekte haben Attribute, die sie genau beschreiben. Die Werte der Attribute bestimmen den Zustand des Objektes. Die Kommunikation zwischen Objekten geschieht mit Nachrichten, welche sich die Objekte gegenseitig zuschicken. Der Austausch von Nachrichten „entspricht“ dem Funktionsaufruf in der klassischen Programmierphilosophie.

Die Prädikatenlogik ist die mathematische Form der Beschreibung von logischen Zusammenhängen [Altenkrüger1992, Russel1995]. Mit ihrer Hilfe können Fakten, Regeln, logische Schlussfolgerungen ausgedrückt werden. Elementare aussagenlogische Formeln können zu komplexeren Aussagen verknüpft werden.

Die Benutzerschnittstelle in mileET ist in OO-Programmiersprache Java implementiert und die Wissensbasis mithilfe Prädikatenlogik 1. Ordnung in Prolog.

XML (Extensible Markup Language) ist eine Metasprache, ein Werkzeug, um eigene Sprachen zu definieren [xml2004, XML]. Die Erweiterbarkeit ist dabei die entscheidende Eigenschaft von XML. Eigene XML-Formate werden in mileET zur Speicherung der Informationen über Aufgabenstellungen (einschließlich Schaltungen) und Bearbeitungszustände der Benutzer verwendet.

Im nächsten Kapitel werden die State-of-the-Art Entwicklungen und die Überlegungen zur Organisation einer GET-Lernumgebung an der TU Ilmenau vorgestellt.

### 3 Stand der Technik

Die meisten Beispiele der interaktiven Lernprogramme in der elektrotechnischen Grundlagenausbildung sind Berechnungs- und Simulationstools sowie Test- bzw. Übungsprogramme. Mitte der 90-er Jahre stand eine Reihe von multimedialen Lernprogrammen aus verschiedenen Domänen den Studierenden zur Verfügung, die impulsgebend für die Entwicklung der GET-Lernumgebung an der TU Ilmenau waren. Zur Jahrtausendwende boten Universitäten und Hochschulen weltweit ihre Lehr-/Lernangebote online an: von einzelnen Lernmodulen bis zur „Bildung online“ (Teleteaching, Virtuelle Universitäten, open universities, Fernuniversitäten). Obwohl intelligente tutorielle Systeme, wissensbasierte oder adaptive Lernumgebungen teuer in der Entwicklung sind und hohe Anforderungen an die Entwickler stellen, werden sie auch für die GET-Domäne entwickelt, allerdings sind sie selten anzutreffen.

In diesem Kapitel werden sowohl intelligente als auch konventionelle Programme vorgestellt, die besonderen Stellenwert für die GET-Lehre bzw. für die Entwicklung der GETsoft Lernumgebung haben (Abbildung 3, Abschnitt 3.4)<sup>26</sup>.

#### 3.1 (Intelligente) Lernprogramme für die Grundlagen der Elektrotechnik

SOPHIE und EL werden als zwei der ersten intelligenten Systeme betrachtet, die für das Gebiet der Elektrotechnik thematisch relevant sind [Wenger1987, Stubenrauch1989].

Im **SOPHIE** (SOPHisticated Instructional Environment) [Brown1982] werden KI-Verfahren zur Dialogführung bei Hypothesen über die Fehlersuche in elektronischen Schaltungen genutzt. Der Ausgangspunkt eines Dialoges ist die Vorgabe einer fehlerhaften elektronischen Schaltung. Im kommentierten Dialog beantwortet das System die Anfragen des Benutzers über Strom-, Spannungs- und Temperaturwerte einzelner Elemente durch den Vergleich der Werte der betrachteten und der fehlerfreien Schaltung. Die Studenten werden aufgefordert, ihre Anweisungen (z.B. Bauelement ersetzen) zu kommentieren (welcher Fehler und warum wird vermutet). Zu den fehlerhaften Hypothesen werden vom System aufklärende Beispiele gegeben.

Aus der Ebene der logischen Deduktion kann jederzeit auf den Schaltungssimulator zugegriffen werden, der die Rolle des technischen Beweisers der Aussagengültigkeit darstellt. SOPHIE ermöglicht den Studierenden, mit dem System in einer Teilmenge der natürlichen

---

<sup>26</sup> Die Markierung betont die für die vorliegende Arbeit relevanten Eigenschaften der Lernsoftware.

Sprache zu kommunizieren, deren Umfang zur ingenieurmäßigen Hypothesenformulierung ausreichend ist.

**EL** ist ein Expertensystem auf dem Gebiet der elektronischen Netzwerkanalyse. Aus einer Datenbank von Netzwerken mit zufällig generierten Fehlern wählt das System eine Situation aus und lässt den Benutzer nach Fehlern suchen. Alle Analyseschritte des Systems zusammen mit den logischen Abhängigkeitsrelationen werden in einer Wissensbasis gespeichert. In einer Datenbasis sind Netzwerktopologie, komplexe Elemente (z.B. Transistoren und Operationsverstärker), Bauelementdaten, stückweise linearisierte Kennlinien von nichtlinearen Elementen usw. enthalten. Ein nach dem Vorwärtsverkettungsprinzip arbeitendes Reasoning-System bestimmt alle aus den Prämissen ableitbaren Konklusionen. Für jeden Ableitungsschritt wird die Inferenzregel gespeichert, wodurch der Benutzer auf Anforderung eine Erklärung bekommen kann. Das System ist in einer LISP-basierten Sprache implementiert.

Die beiden Programme eignen sich für die späteren Studiensemester, da sie Basisfachwissen voraussetzen. SOPHIE nimmt außerdem einen besonderen Platz in der Geschichte der intelligenten tutoriellen Dialog-Systeme ein.

Die Arbeit **Rechnergestützte Übungsplätze** von [Burger/Finsterbush1985] stellt eine Realisierung der TU Ilmenau für das lokale Netz dar. Sie ermöglichte auch eine kooperative Gruppenarbeit. In allen Phasen des Aufgaben- und Problemlösens konnten Hilfestellungen durch ständigen Zugriff zum Begriff-, Methodenspeicher und zu Lösungshilfen erfolgen. Es wurden die bekannten Analysen wie Auswahlantwort, Reihenfolgeantwort oder numerische Antwort im Toleranzbereich realisiert und eine neue Analysemöglichkeit (Lösungsweganalyse) implementiert. Die letzte Möglichkeit wird mithilfe der Speicherung der Lösung einer Aufgabe in Form eines Graphen realisiert. Die Lehrenden sollten den Lösungsgraphen eingeben. Das Programm besitzt zwar keine „Intelligenz“, was die Fähigkeit Aufgaben zu lösen oder Formelherleitung zu überprüfen betrifft, die Aufgaben des gesamten ET-Grundstudiums konnten mit dem Programm überprüft werden.

**Interaktives rechnergestütztes Übungsprogramm** von [Stubenrauch1989] ist eine Weiterentwicklung der Burger/Finsterbusch-Arbeit und stellt ein Dialogsystem zum Üben von Lehrinhalten des Lehrgebietes Grundlagen der Elektrotechnik dar. Die Antwortanalyseverfahren „Auswahlantwortkontrolle“, „Zahlenwertkontrolle im Toleranzbereich mit Kontrolle der Maßeinheit“, „Lösungswegbeschreibung“ und „Kontrolle einer Tabelle von Zahlenwerten“ konnten von Lehrenden über einen Autor-Dialog beim Erstellen der

Übungsinhalte festgelegt werden. Das System bietet u. a. den Aufruf einer „intelligenten“ Hilfestellung. Das intelligente Modul verwaltet das Formelwissen des Fachgebietes und stellt einen „*Versuch der Anwendung von Datenstrukturen und algorithmischen Vorgehensweisen der KI-Forschung auf ein eng umrissenes Anwendungsgebiet innerhalb des Dialogsystems dar*“ [Stubenrauch1989]. Der Einsatz eines minimierten semantischen ATN<sup>27</sup>-Verfahrens lässt natürlichsprachlich formulierte Anfragen zu, die durch die Anzahl von Synonymen zu Problemlöseoperatoren und durch den Inhalt einer Formeldatenbank auf ein relativ enges Gebiet begrenzt werden. Ein Parser ermittelt anhand von semantischen Wortklassen einen Operator für den Zugriff auf einen Lösungsalgorithmus. Jedem Problemlöseoperator entspricht eine semantische Wortklasse mit beliebig vielen Synonymen. Argumente (Variablenamen, Gesetzesnamen, Formelgrößen) des ermittelten Operators sind Inhalte einer Formeldatenbank. Argumente werden mittels syntaktischer Analyse der Eingabe bestimmt. Operator und Argumente stellen die Repräsentationskonstruktion der Eingabe dar und werden dem entsprechenden Such- bzw. Problemlösealgorithmus übergeben. Wort- und Formeldatenbanken können über spezielle Editoren erweitert werden. Der Datenbankgraph ist in Form von Adjazenzmatrizen und einer Entfernungsmatrix repräsentiert.

**A knowledge-based aid for DC circuit analysis** [Hanrahan1989] ist ein regel-basiertes Expertensystem, das den Benutzer bei der Auswahl einer geeigneten Lösungsmethode unterstützt. Das System erfragt eine Reihe von charakteristischen Eigenschaften einer ihm unbekanntem beliebigen Schaltung und liefert eine begründete Empfehlung, welche Analysemethode anzuwenden ist. Die Wissensbasis, implementiert im prologbasierten K-Shell-System, besteht aus sechs Sektionen (Advisers), die jeweils eine der folgenden Funktionen haben: (1) Prescreening, (2) Classification of the Problem and Circuit, (3) Simplification and Transformation Strategies, (4) Comprehensive Analysis, (5) Thevenin and Norton Equivalents, (6) Miscellaneous Methods.

Mithilfe (1) wird eine Voranalyse der Schaltung durchgeführt, die feststellt, ob z.B. die Schaltung elektrisch korrekt ist oder ob redundante Elemente vorhanden sind. In (2) wird die Komplexität der Schaltung eingeschätzt: Anzahl der Maschen und Knoten sowie das Vorhandensein aktiver oder abhängiger Quellen spielt dabei eine Rolle. (3) entscheidet ob es möglich bzw. sinnvoll ist, eine Vereinfachung bzw. Transformation der Schaltung durchzuführen. Diese Entscheidung wird aufgrund der Anzahl der unbekanntem Quellen getroffen. Bei mehreren solchen Elementen wird empfohlen, die Lösungsmethoden aus (4) zu

---

<sup>27</sup> augmented transition network

wählen: Kirchhoff'sche Sätze, Knotenspannungsanalyse, Maschenstromanalyse. Hier wird die Komplexität der Schaltung berücksichtigt, indem u. U. einige Vereinfachungen als Vorschritte empfohlen werden. (5) liefert Empfehlungen zur Ermittlung einer äquivalenten Strom-/ Spannungsquelle. In (6) sind Regeln zu Ermittlung einer Übertragungsfunktion und zu Ermittlung der Quellenparameter, wenn Ströme oder Spannungen gegeben sind.

Das System übernimmt zwar die Rolle des Tutors für die Erklärungen, der Lernende wird aber nicht bei der Anwendung der empfohlenen Analysemethoden unterstützt. Es wird nicht die Möglichkeit gegeben die Lösungen zu erstellen und überprüfen zu lassen. Unklar ist auch die Reaktion des Systems im Falle, wenn der Student die Fragen nicht beantworten kann. Auch die Richtigkeit der studentischen Antworten wird anscheinend nicht überprüft.

Mithilfe des „**ITS for Electric Circuit Exercising**“ [Yoshikawa1992] können ebenfalls die Grundkenntnisse der Schaltungsanalyse erworben werden. Zu zufällig generierten Schaltbildern formuliert das System verschiedene Fragen (z.B. zu Parameterwerten) zur Beantwortung durch den Lerner.

Die Regelbasis ist in LISP implementiert und besteht aus Regeln zur Problemgenerierung, Problemlösung und Fehleranalyse. Verschiedene Schaltungen, bestehend aus Widerständen, Kondensatoren und Spulen, sowie kombinierte Schaltungen, können generiert werden.

Die zufällig generierte Schaltung (mit der Problemstellung) wird in einem Dialogfenster angezeigt. Der Teil der Benutzeroberfläche für die Anzeige der Schaltungen ist in C implementiert, der eigentliche Dialog geschieht über das LISP-Fenster.

Das Benutzermodell besteht aus *overlay model* und *bug model*<sup>28</sup>. Das Programm erzeugt einen Lösungsbaum für das generierte Problem und vergleicht das Ergebnis des Benutzers mit dem eigenen. Stimmen die Antworten nicht überein, wird nach typischen Fehlern gesucht (*bug model*). Wird nichts gefunden, so wird dem Benutzer empfohlen den Lehrstoff zu lesen. Das System bietet eine gute Übungsmöglichkeit an, allerdings werden hier nur die numerischen Antworten überprüft. Es wird nicht die Möglichkeit gegeben, (partielle) Lösungen zu erstellen und überprüfen zu lassen. Auch für die Autoren wird die Möglichkeit, ihre eigenen Aufgaben und Aufgabensequenzen zu erstellen, nicht gegeben.

Das Übungsprogramm **Kirchhoff** [Coulon1993] ist eine Entwicklung des Swiss Federal Institute of Technology in Lausanne. Eine der im System gespeicherten Schaltungen wird in dem graphischen Modul präsentiert. Die Studierenden sollen einzelne Elemente der Schaltung

---

<sup>28</sup> State-of-the-Art zum Thema Benutzermodellierung in [Greer1994]

kennzeichnen (z.B. Knoten markieren, R-, L-, C-Elemente benennen, Zweigströme festlegen) und die Schaltung modellieren, indem sie die notwendigen Gleichungen über die entsprechenden Interface-Elemente zusammenstellen. Diese Gleichungen werden vom System analysiert. Die Autoren folgen der *“philosophy ... to keep guidance and feedback to a minimum“*, sodass nur die Kommentare Richtig/Falsch gegeben werden bzw. die Anzahl der Fehler mitgeteilt wird.

Das Programm ist ein hervorragendes Übungstool für das Thema Kirchhoff'sche Sätze. Auch die Dimensionierung der Wechselstromschaltungen ist mit dem Programm möglich. Eine Erweiterung des Programms um ein Modul zur Unterstützung symbolischer Lösungen wurde geplant. Leider konnten keine Informationen gefunden werden, ob die geplante Weiterentwicklung stattgefunden hat.

In einer weiteren Arbeit der TU Ilmenau wird eine **interaktive Lernumgebung** vorgestellt, die ein *„selbstständiges und selbstverantwortliches“* Lernen unterstützt [Rettig1997]. Die Lernumgebung (implementiert mit Asymetrix ToolBook<sup>29</sup>) ist ein Hypertext/Hypermedia-System mit tutoriellen Komponenten, wobei der Schwerpunkt der Implementierung das Wort *interaktiv* ausdrückt. Durch Bearbeitung von einer Reihe der interessant aufbereiteten Aufgaben zum Thema Grundstromkreis können sich die Lernenden spielerisch und aktiv mit den Problemstellungen auseinandersetzen. Unterschiedliche Formen von Interaktion (z.B. graphisches Arbeiten mit Kennlinien, Animationen, Taschenrechner) und Hilfe (z.B. lokales Feedback, Antwortanalyseverfahren für die Zahlenwerte mit Vorsatz und Maßeinheit, Begriffs-Wörterbuch) machen die Lernumgebung besonderes attraktiv für die Studienanfänger.

Mit dem Buch **„Elektrische Netzwerke mit PSpice mit 150 interaktiven Simulationsbeispielen“** [Justus1998] werden zwei grundsätzliche Ziele verfolgt: (1) Vermittlung von mathematisch-elektrotechnischer Basis zur Berechnung elektrischer Netzwerke und (2) soll der Leser *„ein Gespür dafür bekommen, welche Vereinfachungen, die in der praktischen Ingenieurarbeit immer notwendig sind, er bei der Beschreibung und Berechnung elektrischer Vorgänge vornehmen darf und wie weit die so erzielten Ergebnisse von den wirklichen Daten und Verläufen abweichen“*.

*„In den Anfangskapiteln des Buches werden nach der Vorstellung der aktiven und passiven Grundzweipole verschiedene Berechnungsmöglichkeiten von Netzwerken vorgestellt. Einen*

---

<sup>29</sup> <http://www.sumtotalsystems.com/> (letzter Zugriff 10.02.2005)

*breiten Raum nimmt die Berechnung und graphische Darstellung komplexer Größen ein, die Zustände in sinusförmig belasteten Netzwerken beschreiben. Auch für die Analyse mehrfrequenter Netzzustände, die mit Hilfe der Fourierzerlegung durchgeführt wird, bieten die komplexen Größen wertvolle Hilfe. PSpice ermöglicht mit seiner numerischen Berechnung die graphische Darstellung von Ausgleichsvorgängen auch in stark nichtlinearen und umfangreichen Netzwerken. Mit dem Entwurf aktiver Filter, passiver Reaktanzfilter und einer Saugkreisanlage wird eine kurze Einführung in die Netzwerksynthese gegeben“.*

**Interactive Tutorial Software for Engineering** [Wood1996] visualisiert komplexe Sachverhalte verschiedener Wissensgebiete (z.B. Ingenieurmathematik) und stellt entsprechende Informationen kompakt dar. Als Ziele des Tutorials definiert die Autorin, die Hilfe für die Studierenden, Grundkonzepte zu verstehen, zu visualisieren und die Konzepte miteinander in Verbindung zu setzen. Die Software wird als dynamische und flexible Ressource für Lehrende und Lernende definiert, für individuelles Lernen und für das Lernen in Gruppen.

Noch ein Beispiel eines Multimedia-Lernprogramms stellen die **Multimedia Moduls For Electromagnetics Education** dar [Vidal1996]. Bei dieser Arbeit handelt es sich um eine CD-ROM-basierte Sammlung der Lehreinheiten zu Themenfeldern aus dem Elektromagnetismus: Elektromagnete, Maxwellsche Gleichungen, elektromagnetische Wirkungen, Leitungen, Quiz-Spiel. In diese tutoriellen Übungen sind 2D-Simulationen integriert, u.a. ein Leitungssimulator, ein Simulator für Elektrostatik und ein Simulator für mehrschichtige dielektrische Anordnungen.

**Multimedia Courseware in Engineering Education** [Azemi1996] von Pennsylvania State University and Leonhard Engineering Center of Excellence bietet eine Reihe der in ToolBook implementierten Lerneinheiten zu einigen Themen aus den Grundlagen der Elektrotechnik (Netzwerkgrundlagen). Als Ingenieurwerkzeuge werden den Studierenden PSpice und Matlab angeboten. Diese Courseware ist ein typisches Beispiel eines in einem Autorensystem erstellten Lernprogramms.

“Guided constructivism” nennen die Autoren von **IMITS** (Interactive Multimedia Intelligent Tutoring System) [Butz2000] den im System implementierten didaktischen Ansatz. Die Studierenden werden in einem Goal-Based-Scenario zu Mitgliedern eines multinationalen Ingenieurbüros und haben verschiedene Aufgaben aus den Grundlagen der Elektrotechnik in Form von Aufträgen zu lösen. Die Aufträge werden vom virtuellen Team Leader via Emails und Televideo-Kommunikation erteilt. Der Lernende hat den Zugang zu einem virtuellen

Büro mit eigener Bibliothek und einem virtuellen Labor [vgl. Scherp2002<sup>30</sup>]. Es wird von dem Benutzer eine Antwort, die aus einem numerischen Wert besteht, erwartet und darauf in einem Dialog getestet, ob der Benutzer das für das Lösen notwendige theoretische Wissen hat. Die Benutzungsoberfläche wurde mit Macromedia Authorware und das Expertensystem mit CLIPS (Expertensystem-Shell) entwickelt. Die Aufgaben zu den Themenfeldern Netzwerke bei Gleichstrom und Wechselstrom, Übergangsvorgänge oder Laplace-Transformation werden im System angeboten.

Die Computerisierung der Ingenieurausbildung an **Russischen Hochschulen** hat in erster Linie pragmatisch-didaktischen Charakter – die Studierenden sollten gleich von Anfang ihres Studiums an den Rechner als eines der Hauptwerkzeuge eines modernen Ingenieurs bzw. Wissenschaftlers „erleben“. Forderung der selbständigen Arbeit der Studierenden, Erlernen moderner Ingenieurwerkzeuge, Entlastung der Lehrkräfte von Routine-Arbeit wie z.B. Vorbereitung der individuellen Aufgaben oder formale Überprüfung typischer Aufgaben waren einige der Hauptziele des Computereinsatzes in der Ausbildung. Mehrere interaktive Testmodule stehen den Studierenden zur Verfügung [Kuznetsov1997, NITE, VELK].

### 3.2 Relevante Arbeiten aus anderen Domänen

**LernSTATS** ist ein Programm zum Lernen der Statistik für Psychologen und Sozialwissenschaftler [Schulmeister1996b]. Es ermöglicht entdeckendes Lernen in der Statistik und kann Einsatz im Gruppenunterricht und als individuelles Selbstlernprogramm finden.

Das didaktische Konzept von LernSTATS betont vier Prinzipien:

- Herstellung eines Inhaltsbezugs der Methoden und Statistik zur Psychologie
- Gestaltung der Lernsituationen nach dem Modell des entdeckenden Lernens
- Computerunterstützte Arbeit in Gruppen
- direktes Feedback im Lern-Prozess

LernSTATS bietet eine Vielzahl interaktiver Übungen, ein Statistik-Buch, ein Glossar und eine Online-Hilfe an. LernSTATS kann unmittelbar im Unterricht, bei der Gruppenarbeit im Computerpool und ebenso beim individuellen Lernen eingesetzt werden.

Einige Übungen in LernSTATS präsentieren Puzzles und Rätsel und geben den Lernenden die Chance, die Lösung selbst zu entdecken. Verschiedene zweidimensionale Methoden der

---

<sup>30</sup> hier wird ein multimediales GEN-Labor vorgestellt.

Visualisierung veranschaulichen die erklärten Konzepte und Relationen. Diagramme und Grafiken sind ebenso interaktiv von Studierenden zu manipulieren wie Tabellen. In der Regel sind Grafiken und Tabellen miteinander so verknüpft, dass Interaktionen mit der Tabelle unmittelbar in der Grafik widergespiegelt werden und umgekehrt. Animationen illustrieren funktionale Mechanismen von Formeln durch Hervorhebung der berechneten Daten oder durch visuelles "Wandern" der Daten in die Grafik. Alle Übungen in LernSTATS bieten dem Studierenden wiederholte Möglichkeiten der Manipulation an. In der Regel kann der Studierende die Datenreihe variieren, die der Übung zugrunde liegt. Formeln werden in Teile zerlegt und sind durch direkte Manipulation zusammensetzbar.

Damit wird mit dem Lernprogramm versucht, bestimmte didaktische Prinzipien zu verwirklichen, die für die Motivation und das kognitive Lernen wichtig sind [Schulmeister1996a].

**mechANIma** [Hampel1998a,b] steht für ein offenes Konzept, komplexe Vorgänge einfach und verständlich zu visualisieren und zugleich eine Wissensdatenbank für zukünftige Lern-/Forschergenerationen zu schaffen. Ziel war es, kein Programm nur für eine spezielle Zielgruppe zu schaffen, sondern vielmehr integrativ auf die unterschiedlichsten Anforderungen aus der Laborumgebung des LTM (Laboratorium für Technische Mechanik) zu reagieren. Als besonderer Erfolg werten die Autoren, dass es gelungen ist, Studierende selbst zu Produzenten zu machen, die Multimedia konstruieren, um zu verstehen.

Animationen und Simulationen aus den Bereichen der Grundlagen der Technischen Mechanik helfen komplexe naturwissenschaftliche Sachverhalte zu illustrieren und anschaulich darzustellen. Gleichzeitig bilden diese eine umfassende Zusammenstellung von Beispielen zur kontinuierlichen Erweiterung des mechANIma Leistungsumfanges. mechANIma integriert eine ganze Anzahl multimedialer Bestrebungen. *„Der ständig wachsende Nutzungsumfang erstreckt sich über das Spektrum der Durchführung von Seminaren und Workshops, über Dokumentenverwaltung bis hin zu einer Literaturdatenbank von Studienarbeitern, Diplomanden und wissenschaftlichen Mitarbeitern“.*

**MSAFE** (ein Projekt von OFFIS, Uni Oldenburg und bfe-Oldenburg) ist ein multimediales Selbstlernsystem zur Aus- und Fortbildung von Elektrofachkräften. Das System realisiert die computerunterstützte Simulation und Verifikation elektrischer Schaltungen für den Bereich Steuerungstechnik (Schütz- und Relaischaltungen). In einer CAD-Umgebung werden Aufgaben gestellt, zu denen Lernende Schaltungen entwerfen können. In jeder Phase des

Entwurfes kann der Lernende Hypothesen zur Korrektheit seiner Entwürfe an das System formulieren. Das Programm führt mit dem Entwurf einen Funktionstest durch, um verifizieren zu können, ob das Verhalten des Entwurfs dem Soll-Verhalten entspricht. Ein Funktionstest ist durch eine Sequenz von Testanweisungen mit Realzeitanforderungen für bestimmte Bauelemente und Schaltungsfunktionen definiert. Die Testanweisungen werden durch einen Realzeit-Model-Checker geprüft [I@LLS].

**PULSE** ist ein Prototyp einer intelligenten Designumgebung für pneumatische Schaltungen. Das System stellt dem Anwender eine Sequenz von Konstruktionsaufgaben zur Verfügung. Ausgehend von einer Spezifikation, die aus einem Funktionsdiagramm und einem Beschreibungstext besteht, sollen Lernende pneumatische Schaltungen in einem speziellen Editor entwerfen. Dabei können Hypothesen über beliebige Elemente des Funktionsdiagramms vom Benutzer formuliert und vom System geprüft werden. Als Rückmeldung werden korrekt realisierte und noch zu realisierende Bereiche des Funktionsdiagramms markiert. Um beliebige Entwürfe des Benutzers überprüfen zu können, verfügt PULSE über das erforderliche Grundlagenwissen über das dynamische und statische Verhalten der Pneumatik-Bauelemente und deren Vernetzung. Die Korrektheit eines Entwurfes wird mithilfe des Model-Checking-Verfahrens überprüft. Eine visuelle Simulation betont den explorativen Charakter des Systems. Eine Hilfefunktion, basierend auf der statischen Analyse der Aufgabenstellungen, unterstützt die Lernenden mit konzeptionellen Hilfen [Willms1997, I@LLS].

**WebMath** ist ein webbasiertes ITS für die Domäne der Hochschulmathematik [Liu2002]. Das System soll die Studierenden bei den Hausaufgaben unterstützen und ist somit ein System für individuelles Lernen. Das WebMath-System wird auf der Basis von J2EE realisiert. Die Technik des Constraint Programming (C++ Implementierung) wird für das automatische Lösen von Problemen benutzt, und die Benutzermodellierung wird auf der Basis der Bayesschen Netze implementiert.

Der Lernende hat die Möglichkeit mathematische Aufgabenstellungen zu bearbeiten, indem er seine Lösungsschritte in dem WebMath-Arbeitsblatt aufschreibt. Das System beobachtet die Lösungsschritte und, sobald ein Fehler auftritt, wird der Benutzer aufgefordert, den Fehler zu korrigieren. Das System protokolliert die Fehler in dem Benutzermodell.

**Andes** ist ein ITS für Newtonsche Physik, entwickelt von Wissenschaftlern aus Learning Research and Development Center (LRDC) at the University of Pittsburgh und der United States Naval Academy (USNA) [Schulze2000, mehrere Publikationen sind unter Andes1, Andes2 erhältlich]. Das System besteht aus dem Homework Assignment Editor und dem Tutor. Mit dem Homework Assignment Editor können Autoren Hausaufgaben erstellen. Der Tutor soll die Studierenden beim individuellen Lernen unterstützen, indem er ihnen ermöglicht, verschiedene Probleme aus der klassischen Physik zu bearbeiten. Die Benutzeroberfläche besteht aus mehreren Bereichen. Im Aufgabenstellungsbereich wird die Formulierung der Aufgabe mit dem dazugehörigen Bild gegeben. Hier sollten die Studierenden auch die notwendigen Vektor-Diagramme zeichnen. Im Variablen-Bereich werden alle in der Aufgabe verwendeten Variablen definiert. In dem Lösungsformular können Lösungsformeln eingetragen werden. In Andes sind verschiedene Hilfe-Arten implementiert: Rückmeldungen, prozedurale und konzeptionelle Hilfen. Während der Entwicklung von Andes haben die Autoren verschiedene Aspekte der KI erforscht, in der Lernsoftware umgesetzt und in der Lehre erprobt. Der aktuelle Forschungsschwerpunkt ist die Entwicklung einer natürlichsprachlichen Dialogkomponente [Jordan2003]. Für die vorliegende Arbeit waren die Aspekte der Aufgabenbehandlung im System sowie die Benutzer-Interaktivitätsmöglichkeiten relevant.

### 3.3 Computeralgebrasysteme in der Ingenieurausbildung

Um Studierende auf die spätere Berufspraxis vorzubereiten, werden sie auch mit computerbasierten Ingenieurwerkzeugen konfrontiert. In der Lehre haben sich (CAD) PSpice, CircuitMaker, Electronics Workbench und (CAS) Mathematica<sup>31</sup>, Matlab<sup>32</sup>, Maple<sup>33</sup> sowie Mathcad bewährt<sup>34</sup>. CircuitMaker, Electronics Workbench und Mathcad eignen sich auch als Ingenieurwerkzeuge für Novizen. Eine Evaluation derartiger Systeme ist u. a. in [Canizares1997, Wester1999] zu finden.

**Mathematica** von Wolfram Research ist fester Bestandteil der Software-Ausstattung von Universitäten. Mathematica beherrscht die symbolischen Berechnungen und bietet eine hoch entwickelte Programmiersprache. In Form von Mathematica Notebooks lassen sich die

---

<sup>31</sup> <http://www.wolfram.com/> (letzter Zugriff 10.02.2005)

<sup>32</sup> <http://www.mathworks.com/> (letzter Zugriff 10.02.2005)

<sup>33</sup> <http://www.scientific.de/produkte/maple/index.html> bzw. <http://www.maplesoft.com/> (letzter Zugriff 10.02.2005)

<sup>34</sup> es werden auch weitere CAS benutzt [SAC]

Ergebnisse mit Text, Formeln, Diagrammen, Bildern etc. dokumentieren und in der Lehre einsetzen. Der einzige Nachteil dieses Systems ist sein zu hoher Preis.

**MuPAD**<sup>35</sup> von der Universität Paderborn ist ein mathematisches Expertensystem, mit dem sich sowohl symbolisch-algebraische Berechnungen (Formelmanipulationen, Rechnungen mit exakten Ergebnissen) als auch numerische Berechnungen mit beliebiger Genauigkeit durchführen lassen. Dieses CAS wird auch als kostengünstige Alternative zu Mathematica gesehen.

**Matlab** von The Mathworks ist eine leistungsfähige, umfassende und bedienerfreundliche Software für technische Berechnungen. Es bietet Ingenieuren, Wissenschaftlern und Technikern ein einheitliches, interaktives System, das numerische Berechnungen und wissenschaftliche Visualisierung in sich vereinigt. Matlab wird als Rechnerkomponente z.B. in einem interaktiven Lerntool für Fourier-Konzepte benutzt [Hamann2003].

**Maple** von maplesoft ist ebenfalls ein leistungsfähiges System und wurde hauptsächlich für symbolische Berechnungen entwickelt. Die Unterstützung der Numerik reicht für Ausbildungszwecke aus. Das System verfügt über eine eigene Programmiersprache und ist somit sehr flexibel. Das bringt mit sich auch das Problem hinsichtlich des Einsatzes in der Lehre - die Komplexität und somit das aufwendige Erlernen.

**Mathcad** von mathsoft ist ein leistungsfähiges System für mathematische Berechnungen, welches durch intuitive Handhabung relativ leicht zu erlernen ist. Mathcad löst eine umfangreiche Palette von Problemen aus den Bereichen Mathematik, Wissenschaft und Ingenieurwesen. Die "free form" Oberfläche ermöglicht die Integration von normaler mathematischer Notation, Grafik und Text in einem interaktiven Arbeitsblatt [CAS].

Die Vermittlung elektrotechnischer Sachverhalte ist immer in Einheit mit der Vermittlung der mathematischen und physikalischen Grundlagen zu sehen. Die Integration eines Computer-algebrasystems als Werkzeug bei der Theorieaneignung, beim Aufgaben- und Problemlösen erlaubt die Konzentration auf das Wesentliche durch die Entlastung von rechnerischen Details und das Beschreiten neuer Wege beim Erlernen inhaltlicher und methodischer Grundlagen (z.B. Simulation komplexer Sachverhalte, Darstellung dynamischer Vorgänge, allgemein also lehrinhaltsgerechte Medienwahl zum Erstellen interaktiver Elemente, in die aktive Formeln, Text, animierte Grafik und Ton eingebunden sind). Durch die Fähigkeit zur symbolischen und numerischen Berechnung sowie zur grafischen Darstellung der Ergebnisse ergeben sich neue Möglichkeiten der Interaktion zur Ergebnisbewertung, für die Lösungshinweise und für die

---

<sup>35</sup> <http://research.mupad.de> (letzter Zugriff 10.02.2005)

Beantwortung von Fragestellungen. Ingenieurwerkzeuge sind also eine unverzichtbare interaktive Komponente der multimedialen Lernumgebung.

Bei der direkten Anwendung eines CAS ist die Kontrolle über den Lehrprozess vollkommen den Lernenden überlassen: sie entscheiden welches System, für welche Aufgaben und in welcher Form sie es benutzen [vgl. Canizares1997].

Bei der Einbindung eines CAS in ein Lehrsystem handelt es sich um die Empfehlungen der Kurs- bzw. Buchautoren, ein bestimmtes System für die Bearbeitung der Lehraufgaben zu verwenden. Normalerweise werden Beispiele mit typischen Aufgaben vorbereitet, sodass die Studierenden das Problem mit verschiedenen Werten analysieren können und dabei auch in die Benutzung des Systems eingeführt werden [Iakimtchouk2000a].

Mit einigen CAS lassen sich Lehrmodule direkt in dem CAS erstellen (Notebooks in Mathematica, eBooks in Mathcad).

Eine Anbindung eines CAS an ein Lernsystem findet dann statt, wenn dem Benutzer eine zusätzliche Schnittstelle zur Verfügung gestellt wird, in der er die Eingaben macht, die dann an das CAS übergeben werden. Das CAS führt die (Teil-)Berechnungen oder Analysen durch und liefert die Ergebnisse zurück, die vom Lehrsystem beurteilt werden, um die entsprechenden Rückmeldungen an den Benutzer zu geben [z.B. Burger1997, Horn1996, Buedenbender2002].

Die Umsetzung des letzten Anwendungsszenarios ist programmtechnisch gesehen die schwierigste von allen, es gibt allerdings den Autoren die Möglichkeit, komplexere Lernsysteme mit intelligenter Lernerunterstützung zu bauen.

Mit Mathcad lassen sich Lehreinheiten in Form von **elektronischen Büchern** (eBooks) erstellen. Die in diesen interaktiven Dokumenten angebotenen Problemstellungen können direkt im Dokument von Lernenden untersucht werden. Durch die Verlinkung zwischen einzelnen Mathcad-Arbeitsblättern lassen sich interaktive Lehr- bzw. Übungsbücher zusammenstellen. Über das Mathcad-Informationszentrum wird der Zugriff auf elektronische Übungen und Problemstellungen zu verschiedensten Themen aus der Mathematik, Physik, Elektrotechnik, Mechanik usw. ermöglicht.

Die „**Kleine Formelsammlung Elektrotechnik**“ [Metz1998] ist ein Beispiel des Mathcad-basierten eBooks. Die Formelsammlung besteht aus einer Vielzahl von Mathcad-Dokumenten, ergänzt um zahlreiche Popup-Fenster mit Hinweisen und um etwa 200 Aufgaben. Die Aufgaben sind den einzelnen Abschnitten thematisch zugeordnet. Sie

enthalten konkrete Werte, die der Nutzer aber ändern kann. Ein umfangreiches Indexverzeichnis ermöglicht die Suche nach bestimmten Fachbegriffen.

**ActiveMath** ist eine online Lernumgebung für aktives Lernen der Mathematik [ActiveMath]. Die Lernumgebung bietet den Studierenden eine Vielfalt interaktiver Lerneinheiten (u. a. Übungen mit Feedback, Lösungsbeispiele, Animationen) an. In ActiveMath sind zwei CAS integriert: Maple und MuPAD [Buedenbender2002]. In den CAS-Übungen der ActiveMath-ITS werden u. a. Aufgaben gestellt, die entweder eine Antwort oder eine mehrschrittige Lösung erwarten. Die Studierenden sollten ihre Lösungen in CAS-Notation in einem dafür vorgesehenem Fenster aufschreiben und vom System überprüfen lassen. Die Rückmeldungen in Form Input-Error, Korrekt/Falsch, Elaboration-on-Error, Show-Solution unterstützen die Benutzer beim Problemlösen. Wie Andes ist ActiveMath ein großes Forschungsprojekt<sup>36</sup> über den KI-Einsatz in den Lehr-Lernsystemen. Für die vorliegende Arbeit sind die Aspekte der CAS-Einbindung relevant.

**ConsMath** ist ein Tool, mit dem die interaktiven Aufgaben erstellt werden können, die dann von Studierenden im geführten Dialog bearbeitet werden können. Die Benutzungsoberfläche ist in Java implementiert, die Manipulation der mathematischen Ausdrücke geschieht mithilfe MathML und die mathematische Verarbeitung der Formeln findet in Mathematica statt [Mora2003]. Lehrende können mit dem Tool interaktive Anwendungen und ganze Kurse entwickeln. Die Entwicklung geschieht in zwei Phasen. In der ersten Phase wird eine generalisierte Problemstellung formuliert, für die konkrete Instanzen automatisch nach Zufallsprinzip generiert werden. Während der zweiten Phase bestimmt der Lehrende den zukünftigen Dialog mit dem Lernenden - eine Sequenz aus Fragen, die für das zu bewältigende Problem von Bedeutung sind. Rückmeldungen des Systems auf diese Fragen müssen ebenfalls vom Kursautor definiert werden.

Auf die GET- Problemstellungen übertragen, kann dieser Ansatz zum aktiven Lernen der Herleitungen von Formeln umgesetzt werden.

**ToLeDe** (Teaching Linear Ordinary Differential Equation) ist ein ITS für lineare Differenzialgleichungen zweiter Ordnung mit konstanten Koeffizienten. In diesem System wird versucht, die Leistungsfähigkeit eines CAS (Maple) durch zusätzliche intelligente Komponenten, welche sowohl deklaratives Domänenwissen als auch Lehrstrategien und

---

<sup>36</sup> mehrere Publikationen unter [www.activemath.org](http://www.activemath.org) erhältlich

Benutzermodellierung enthalten, zu erweitern [Paraskakis2002]. Dadurch sollte ein CAS die Fähigkeit erreichen, die Lösungen der Studierenden kontextsensitiv beurteilen zu können.

### 3.4 Zusammenfassung

In den letzten 30 Jahren hat sich die Welt der Computer (und mit ihr die technische Qualität von Lehr-/Lernprogrammen) drastisch verändert. Die intelligenten Lernprogramme für die Grundlagen der Elektrotechnik bleiben immer noch eine Seltenheit. Auf den letzten ITS und AIED findet man kaum Beiträge zu ingenieurspezifischen Themen. Die auf FIE und in Proceedings of the IEEE vorgestellten Entwicklungen sind nun auch schon 10 Jahre alt und bis auf IMITs softwaretechnisch veraltet. Die Ideen aus den alten Projekten können allerdings in die neuen Entwicklungen übernommen werden.

Die folgende Abbildung positioniert die GETsoft und mileET zwischen den oben vorgestellten Entwicklungen und weist auf die inhaltliche bzw. mediendidaktische Verwandtschaft.

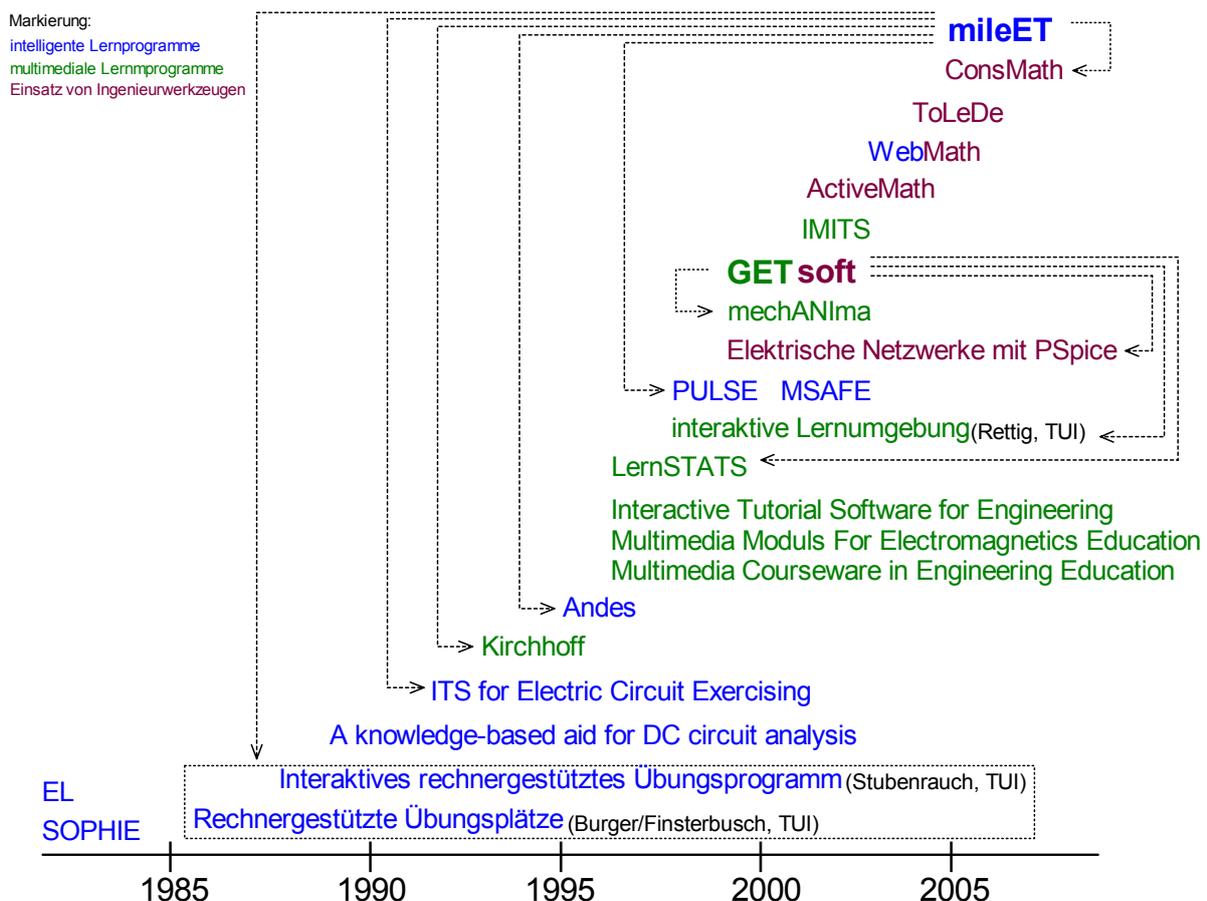


Abbildung 3. Verwandte Arbeiten

Allgemein allen Realisierungen auf dem GET-Gebiet ist eine starke Aufgabenorientierung, was auch generell für die Ingenieurausbildung charakteristisch ist. Diese Programme bieten auch im verschiedenen Maße realisierte adaptive Unterstützung, allerdings stellt kein Programm aus der GET-Domäne den Benutzern die Möglichkeit zur Verfügung, eigene Lösungsentwürfe zu verarbeiten: zu erstellen, zu editieren und unvollständige Lösungen zu überprüfen. Eine Unterstützung der symbolischen Lösungsermittlungen z.B. im Programm „Kirchhoff“ blieb vermutlich in der Konzeptionsphase. Eine mögliche Lösung solcher Probleme wird im mileET-Kapitel dieser Arbeit vorgestellt. Eine ganze Reihe von Angeboten und Ideen z.B. zum Thema virtuelle Labore oder spezielle Simulationstools werden aufgrund geringer Relevanz zum Thema der eigenen Arbeit nicht angesprochen. Diese Tools wurden in den letzten Jahren aktiv entwickelt.

„Interactive Tutorial Software for Engineering“ und „Multimedia Modul For Electromagnetics Education“ stellen Beispiele der Experimentierumgebungen aus dem englischsprachigen Raum dar.

IPSEs „MSAFE“ und „PULSE“ sind die Realisierungen in Nachbardisziplinen: Steuerungstechnik und Pneumatik. Allerdings werden hier keine mathematischen Formeln verarbeitet und die Unvollständigkeit einer Lösung kann „nur“ im Sinne des unvollständigen Schaltungsentwurfes verstanden werden.

Die ITS „Andes“ und „WebMath“ stehen, was die Interaktivitätsmöglichkeiten der Lernenden betrifft, der mileET-Lernumgebung sehr nah: sie bieten eine adaptive Unterstützung beim Herleiten einer Lösung.

Interaktive Beispiele aus den Lernprogrammen „LernStat“ und „Mechanima“ dienen als didaktische Orientierung für die Weiterentwicklung der multimedialen Lernprogramme GETsoft. Und die Webfähigkeit von „LernStat“ und „Mechanima“ sowie das Wachstum an online Lernangeboten Ende der 90-er Jahre stimulierten die Entwicklungen an der TUI von geschlossenen Lernumgebungen in Richtung Webbasierung.

## 4 Weiterentwicklung der Lernumgebung GET

Das didaktische Konzept der Lernumgebung GETsoft sowie ihre technische Realisierung basieren auf mehreren Ansätzen.

Die Lehr- und Lernmodule der multimedialen Lernumgebung (Abbildung 4<sup>37</sup>) enthalten hypertextbasierte Lehrtexte, technische Datenbanken, Aufgaben- und Übungssammlungen. Dem Nutzer werden Werkzeuge (z. B. ein Computeralgebrasystem) zur Durchführung von Experimenten und zur Lösung von Aufgaben angeboten. Die eingebundenen Videomaterialien demonstrieren Beispiele aus der elektrotechnischen Praxis. Interaktive Animationen und Simulationen illustrieren komplexe Sachverhalte. Die interaktiven Übungsprogramme mit verschiedenen Möglichkeiten zum Experimentieren und differenzierte Fehlermeldungen unterstützen die praktische Anwendung von theoretisch erworbenem Wissen. Mit dem umfassend ausgestatteten Hilfesystem wird versucht verschiedene Lernstile zu unterstützen:

- kontextsensitive Hilfen für jede Teilaufgabe;
- Hilfen, die verschiedene Methoden der symbolischen Lösung der Aufgabe illustrieren;
- Bewertung der Eingaben der Lernenden.

Somit werden sowohl instruktionsorientierte Ansätze verfolgt als auch die des entdeckenden Lernens.

Frühere Offline-Lösungen der GETsoft-Lernumgebung stehen als webbasierte Lernprogramme online<sup>38</sup> zur Verfügung. Die Sammlung von Aufgaben und multimedialen Dokumenten wird über die TaskWeb<sup>39</sup>-Datenbank angeboten. Somit stellt GETsoft eine offene webbasierte Lernumgebung dar. Die Abbildung 4 illustriert die Struktur der Lernumgebung und die Abbildung 5 zeigt den prinzipiellen Aufbau eines GETsoft-Lernmoduls.

Für wichtig wird die Integration von Texten, Beispielen, Experimenten, Werkzeugen, Aufgaben und Übungen für alle Lernmodule unter einer einheitlich gestalteten Benutzeroberfläche gehalten. Die Studierenden können frei im Lernprogramm navigieren, wobei die angegebene Reihenfolge der Themen als tutorielle Lernstoffsequenz angesehen wird. Die Führung ist übersichtlich gestaltet, die Bedienung ist einfach und für alle Lernprogramme einheitlich organisiert.

---

<sup>37</sup> In der Abbildung sind die Komponenten der Lernumgebung markiert, an deren Konzeption und/oder Entwicklung die Autorin teilgenommen hat und die in der vorliegenden Arbeit betrachtet werden.

<sup>38</sup> im Rahmen des LearnWeb unter <http://www.getsoft.net/> (letzter Zugriff 10.02.2005)

<sup>39</sup> <http://taskweb.getsoft.net/> (letzter Zugriff 10.02.2005)

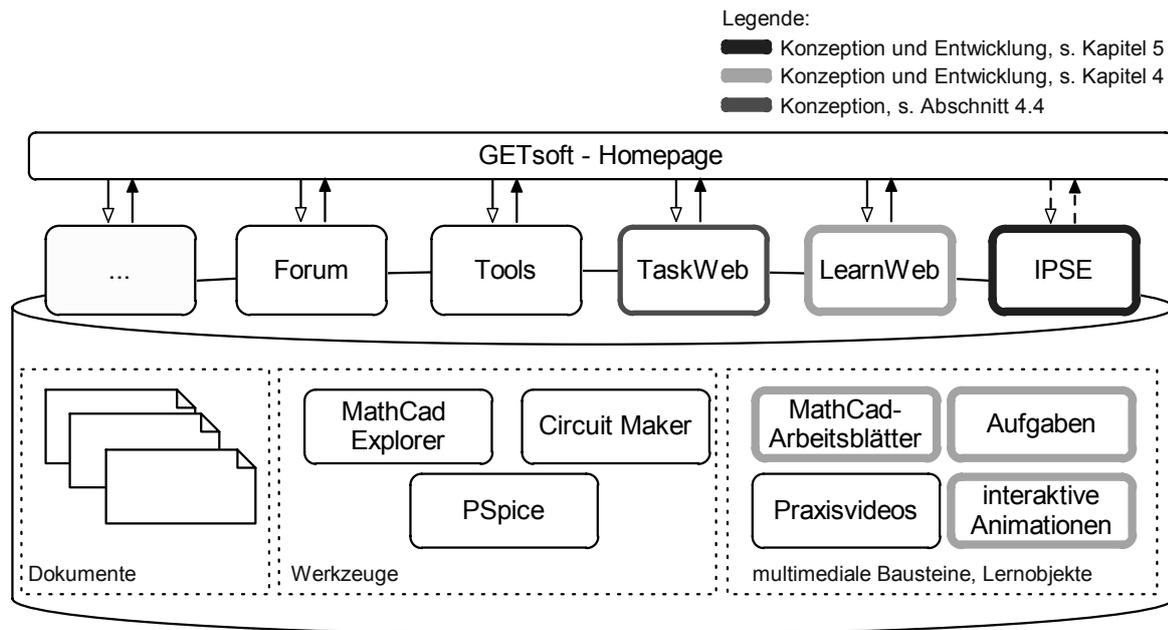


Abbildung 4. Computerbasierter Teil der offenen Lernumgebung GETsoft<sup>40</sup>

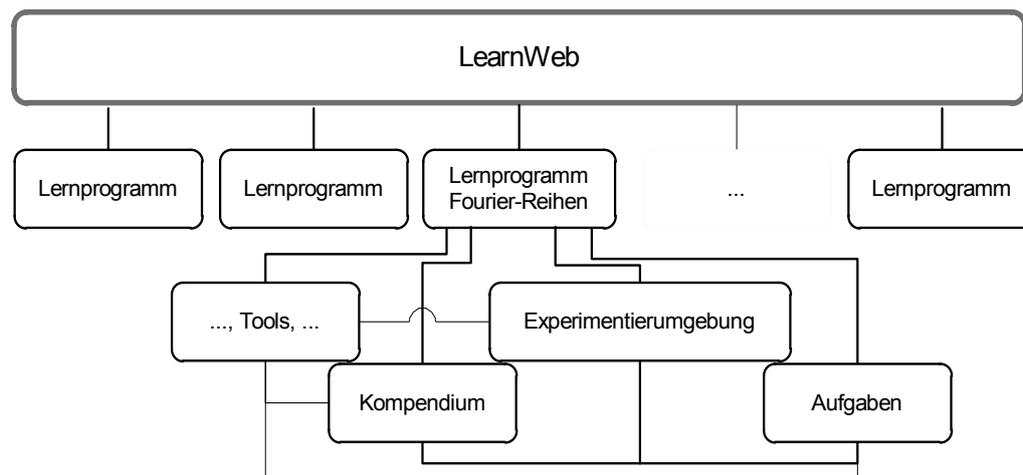
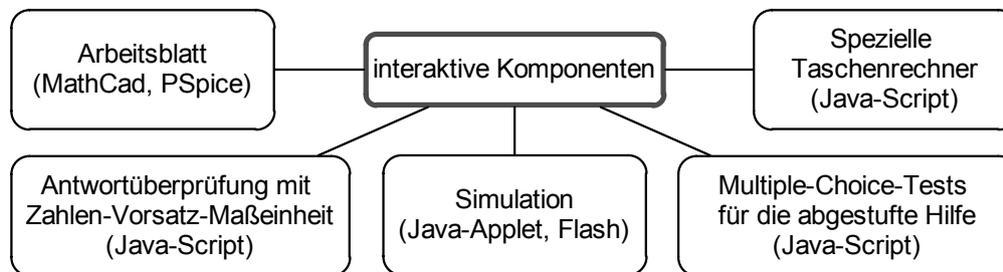


Abbildung 5. Struktur eines GETsoft-Lernmoduls

In der Einführung werden Lehrziele des jeweiligen Moduls definiert, die Bedeutung des Wissens für die Theorie und Praxis dargestellt, sowie einige interessante Beispiele aus dem Themengebiet angeboten. Im Kompodium werden die theoretischen Grundlagen behandelt, was sich zum großen Teil in Formelherleitungen widerspiegelt. Das Ziel dieses Programmteiles ist die Darstellung der theoretischen Inhalte sowohl zum Nachschlagen als auch zum Erarbeiten des Stoffes. Zum besseren Verständnis der Formelherleitungen werden Erklärungen, Videos und Animationen angeboten. Es wird auch aus dem Kompodium zu entsprechenden Aufgaben und Experimentiermöglichkeiten verwiesen.

<sup>40</sup> in Anlehnung an [Rettig1997]

Die Aufgaben in LearnWeb verfügen über eine Palette von Möglichkeiten zur Antwortüberprüfung: vom einfachen „Falsch/Richtig“ bis zu CAS-Einsatz. Dabei wird viel Wert auf verschiedene Interaktivitätsformen gelegt, die die Autoren aufgrund ihrer Erfahrung für am besten geeignet halten (Abbildung 6). Zu den Aufgaben wurde eine umfangreiche Hilfe mit abgestuften Hinweisen zusammengestellt.



**Abbildung 6. Interaktive Komponenten der Lernumgebung GETsoft**

In der Experimentierumgebung wird den Studierenden die Möglichkeit gegeben, die Grenzen der im theoretischen Teil des Lernmoduls eingeführten Beispiele zu testen, den Einfluss verschiedener Parameter auf bestimmte Funktionen bzw. Signale zu simulieren, sowie eigenes Wissen zu testen. Alle diese Aspekte spielen für das Verständnis der theoretischen Überlegungen und für die vertraute Handhabung des Lehrstoffes eine wichtige Rolle.

Eines der Ziele der Experimentierumgebung liegt darin, den Lehrstoff für die Benutzer in attraktiver und interaktiver Form darzustellen, um sie auf die Situationen und Eigenschaften aufmerksam zu machen, die sie eventuell dazu bringen, Fragen über das Gesehene zu stellen und sie so zu theoretischen Erklärungen der Phänomene zu leiten. Die Vorteile von Multimedia in der Lehre kommen damit voll zum Einsatz. Eine Experimentierumgebung ist somit ein besonderes Merkmal einer Lernumgebung.

Unter Tools werden die Teile der Lernprogramme zusammengefasst, die zur aktiven Arbeit mit der Lernumgebung beitragen: ein Computer-Algebra-System, themenspezifische interaktive Module und Module zur Antwortüberprüfung. Mit diesen Mitteln werden die Vorteile des technischen Fortschrittes und der Multimediaentwicklung für den Einsatz in der Ausbildung deutlich demonstriert.

Im Kompendium werden Sachverhalte erläutert, die nur mit Formeln, Herleitungen und Berechnungen ausgedrückt werden können. Zu dieser Problematik können schlecht Bilder oder Graphiken eingesetzt werden. Dafür werden Beispiele zur Verfügung gestellt, die es ermöglichen, die Berechnungen leichter nachzuvollziehen. In den Beispielen werden ausführliche Lösungswege angezeigt.

Die folgende Abbildung fasst alle interaktiven Komponenten, die die didaktische Interaktivität in dem Lernmodul „Fourier-Reihe“ realisieren, zusammen.

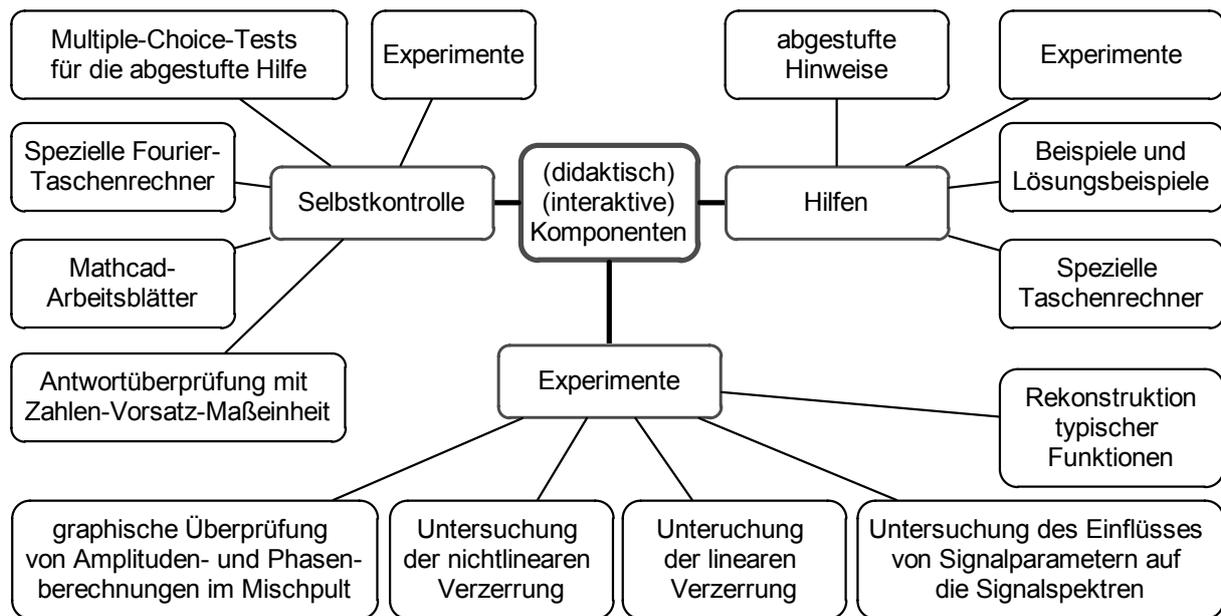


Abbildung 7. Didaktische Interaktivität im Lernmodul Fourier-Reihe

Zu betonen ist, dass didaktische Interaktivitätsformen in GETsoft in erster Linie dem Lernen durch Lösen von Aufgaben und Problemstellungen dienen. Im nächsten Abschnitt wird auf einzelne dieser Formen näher eingegangen.

#### 4.1 Interaktive Experimentierumgebungen in der Lernumgebung GET

Im Weiteren werden interaktive Elemente der Lernumgebung betrachtet, die von der Autorin implementiert worden sind bzw. deren Implementierung unter direkter Betreuung durch die Autorin stattfand.

Die ersten Versionen der multimedialen Lernprogramme zu den Grundlagen der Elektrotechnik im Fachgebiet sind mit verschiedenen ToolBook-Versionen implementiert worden [Hammer1998, Wagner1997]. Das Autorensystem besaß eine Objekt-basierte Programmiersprache OpenScript, die mächtig genug war, um offline-Versionen der Lernprogramme zu realisieren. Als Lösung für die webbasierte Lernumgebung war dieses Tool nicht geeignet trotz dem angebotenen Plugin (große Dateien, nicht alle Scripte wurden unterstützt, keine offenen Formate). Die webbasierte Version des Programms Fourier-Reihen stellt eine Weiterentwicklung des offline-Programms dar und war der erste Versuch einer webbasierten GETsoft-Lernumgebung. Die meisten vorhandenen Tools und multimedialen Elemente wurden auf die neue Technik angepasst und somit auch in die Webversion übernommen. Die mit Mathcad erstellten speicherplatzaufwendigen Animationen wurden als

Java-Applets neu implementiert, ebenso wurden alle Experimente als Java-Applets, die Antwortüberprüfungs-Routine und die Navigationsfunktionen in Java-Script neu realisiert.

Anhand eines GETsoft Lernprogramms werden die Besonderheiten des Lernens mit der Experimentierumgebung erläutert. Das Lernprogramm Fourier-Reihe kommt im Direktstudium und in der Weiterbildung zum Einsatz. Im Kompendium wird in kompakter Form das Grundwissen zu den Fourier-Reihen dargestellt.

Dieses Wissen können die Lernenden mit dem Modul Experimentierumgebung ausprobieren und in dem Modul Aufgaben anwenden und testen. Anhand von zehn wichtigen Signalfunktionen und sieben wichtigen Übertragungssystemen (siehe Anhang A1) werden die Netzwerke bei nichtsinusförmiger periodischer Erregung betrachtet.

Die Abhängigkeit der Form der Spektren von Parametern bei normierter Signalamplitude können in der Experimentierumgebung Fourier-Reihe erforscht werden. Da für die Filtersynthese in erster Linie die Form der Spektren ein wichtiges Kriterium ist, werden in den Experimentierumgebungen die Gesamtspektren dargestellt. Auch die numerischen Werte spielen eine wichtige Rolle: für das Berechnen einzelner Harmonischer ist ein spezieller Taschenrechner implementiert worden. Um den Charakter des Signals im Frequenzbereich besser vorstellen zu können, wird die Möglichkeit gegeben, die Anzahl der anzuzeigenden Harmonischen einzustellen. So kann analysiert werden, wie wichtig der Beitrag einzelner Harmonischer in dem analysierten Signal ist. Das Funktionsbild sowie die Spektren und die rekonstruierte Funktion werden entsprechend den vom Benutzer eingegebenen Daten aktualisiert und die Ergebnisse angezeigt.

In der Experimentierumgebung Fourier-Reihen<sup>41</sup> werden Animationen zu den wichtigsten Funktionen angeboten. Bei der Verschiebung der Rechteckfunktion auf der Zeitachse interessiert die Veränderung des Amplituden- und Phasenspektrums mit der Zeitverschiebung. Die Verschiebung verändert die Symmetrieeigenschaften der Funktion und somit verändert sich das Phasenspektrum.

Beim Experimentieren, z.B. mit der Rechteckimpulsfolge, kann die Veränderung des Amplituden- und Phasenspektrums beobachtet werden, wenn das Tastverhältnis, die Impulsbreite zur Periodendauer, verändert wird: bei dem schmaler werdenden Impuls wachsen die Anteile der höheren Harmonischen (der Oberwellen) im Vergleich zur 1.

---

<sup>41</sup> Alle hier angesprochene Experimente sind unter Experimentierumgebung auf [http://getsoft.net/\\_fouriertest/ger/index2.html](http://getsoft.net/_fouriertest/ger/index2.html) zu erreichen (letzter Zugriff 10.02.2005). Einige Screenshots befinden sich im Anhang A2.

Harmonischen (der Grundwelle). Bei einer Trapezfunktion werden die Anstiegs- und Abfallzeiten der Flanken verändert.

Aus dem Experiment mit der Sägezahnfunktion kann erkannt werden, dass die Werte der Fourier-Koeffizienten  $a_n$  und  $b_n$  nur vom Wechselanteil der Signalfunktion bestimmt werden.

Beim Experimentieren mit den unsymmetrischen und symmetrischen Phasenanschnitten kann die Veränderung des Amplituden- und Phasenspektrums beobachtet werden, in Abhängigkeit von der Phasenanschnitt-Zeit: bei Vergrößerung der Anschnittzeit wachsen Anteile der höheren Harmonischen (der Oberwellen) im Vergleich zur 1. Harmonischen (der Grundwelle).

Die Qualität der Signalsynthese in Abhängigkeit von der Harmonischenanzahl wird mit den kommentierten Animationen zur Fourier-Synthese verdeutlicht. Zu ausgewählten Funktionen können Animationen gestartet werden, die diese Abhängigkeit illustrieren.

Bei den Funktionen wie Rechteckfunktion, Rechteckimpulsfolge, Treppenfunktion, symmetrischer und unsymmetrischer Phasenanschnitt kann das Auftreten des Gibbsschen Phänomens an den Sprungstellen der Funktionen beobachtet werden. Bei den Dreieck- und Trapez-Funktionen, bei denen dieses Phänomen nicht auftritt, reichen schon wenige Harmonische um das Signal zu rekonstruieren.

In der Experimentierumgebung Lineare Verzerrung werden die Frequenzeigenschaften einiger der wichtigsten Netzwerke erforscht. Die Filter-Effekte dieser Netzwerke werden anhand Spektrenvergleichs verdeutlicht. Die Amplituden- und Phaseneigenschaften des Netzwerkes, welche diese Effekte erklären, werden erst nach Anfrage angezeigt, so kann der Student erst versuchen, seine eigene Erklärung für das Phänomen zu formulieren.

In dem Lernprogramm werden die Grundlagen der Signaldarstellung im Frequenzbereich erlernt. Viel Wert wird dabei auf die wichtigsten Einflussgrößen gelegt und auf die Einstellung von „unbedeutenden“ Parametern wurde bewusst verzichtet. Zu solchen Parametern gehören z.B. die Werte für die Amplitudenspannung und Basisfrequenz<sup>42</sup> bzw. die Parameter, die eine Erkennung des zu erlernenden Effektes unnötig erschweren. Für komplexere Experimente stehen den Studierenden Simulationstools, wie z.B. Electronics Workbench oder PSpice, zur Verfügung.

Aus dem Experiment mit dem Tiefpass (Abbildung im Anhang A2) kann erkannt werden, dass der Tiefpass als Integrierglied wirkt. Am Ausgang erscheint das stark gedämpfte Integral der Eingangsspannung (Rechtecksignal), eine Dreiecksspannung. Anhand von Experimenten

---

<sup>42</sup> Einfluss dieser Parameter kann mithilfe entsprechender Normierung berücksichtigt werden.

mit den Trapez- und Treppen-Funktionen an einem Tiefpass oder mit der Rechteckfunktion an einem doppelten Tiefpass kann man beobachten, dass zwischen den Eigenschaften Klirrfaktor und Dämpfung ein Kompromiss notwendig ist: Mit größer werdendem Widerstand lässt sich zwar der Klirrfaktor der Ausgangsspannung weiter verringern, nachteilig ist aber die zunehmend stärkere Bedämpfung der Grundwelle.

Die Experimente mit dem Hochpass helfen zu erkennen, dass bei bestimmten Parameter-einstellungen sich dieses Filter als Differenzierglied verhält.

Das Experiment zu dem Reihenschwingkreis hilft zu erkennen, dass wegen der geringen Bandbreite des Schwingkreises bei hoher Güte aus einem Frequenzgemisch eine beliebige Harmonische herausgefiltert werden kann.

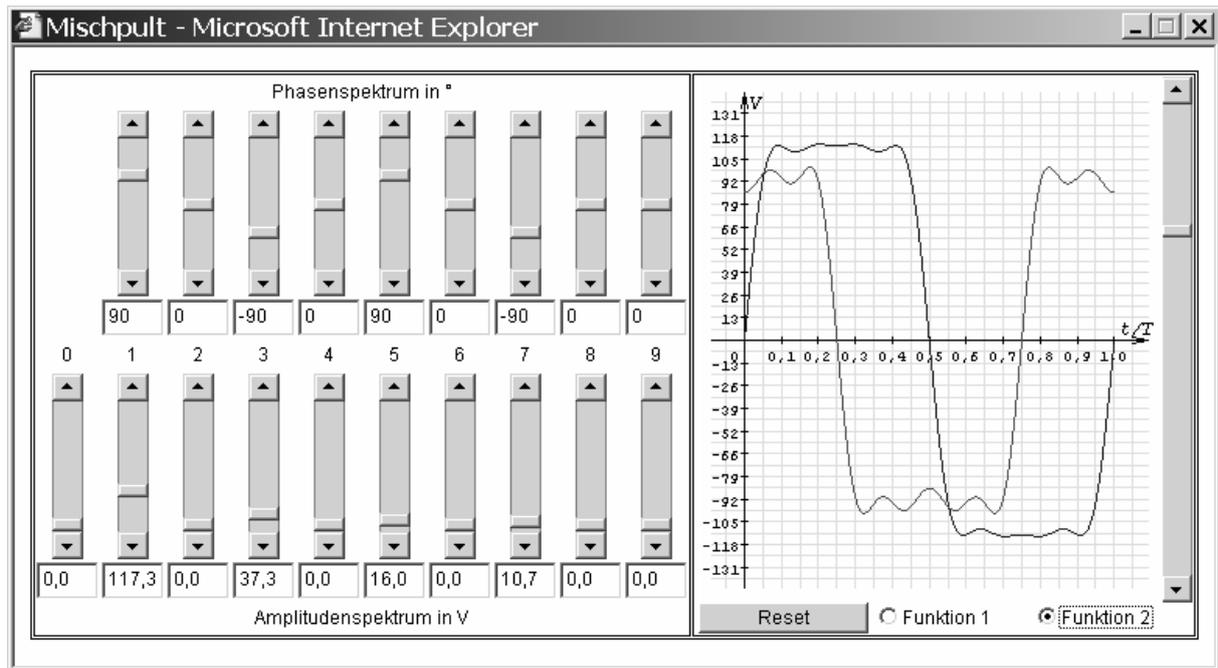
Aus dem Experiment zum RC-Bandpass wird deutlich, dass das Netzwerk eine geringe Frequenzselektivität bzw. große Bandbreite hat. Im Phasenspektrum kann man verfolgen, welche Bandmittenfrequenz vorliegt. Sie hat den Phasenwinkel Null.

Mit der Bandsperre werden über die Veränderung des Terms  $\omega CR$  einzelne Harmonische im Spektrum der Eingangsspannung gesperrt und nicht an den Ausgang übertragen.

Das Experiment mit dem phasenkompensierten Spannungsteiler erklärt, dass die Verzerrung einer Rechteckspannung am Ausgang eines RC-Spannungsteilers kompensiert werden kann.

Die implementierte Art des Lehrstoffangebotes ist der in [Belski2003] vorgestellten ähnlich. Die Lernenden haben den Überblick über das ganze System und können somit die Beziehung Schaltbild-Mathematischer Ausdruck-Graph besser nachvollziehen und sich merken.

In der Experimentierumgebung Nichtlineare Verzerrung wird die Bedeutung der nichtlinearen Übertrager mit Beispielen der magnetischen Hysterese und der Entstehung der Amplitudenmodulation vermittelt. Eins der Experimente ermöglicht das Experimentieren mit additiver Mischung zweier Signale. Der während der Animation laufende Punkt zeigt, wie die Graphen der Eingangs- und Ausgangs-Funktionen über dem Graphen der Übertragungsfunktion in Bezug zu einander stehen. Durch die Einstellung einzelnen Filtermöglichkeiten (konkrete Schaltungen werden im Rahmen des Lernprogramms nicht betrachtet) kann die Amplitudenmodulation simuliert werden. Das Bild des Signals im Zeitbereich und die Spektren verdeutlichen die Bedeutung einzelner Filterfunktionen (z.B. Ausschalten der doppelten Trägerfrequenz oder des Gleichliedes).



**Abbildung 8. Mischpult**

Mit dem Harmonischen Mischpult (Abbildung 8) können die Studierenden nachvollziehen, wie sich die Fourier-Reihe aus den einzelnen Harmonischen zusammensetzt. Das Mischpult hat großen Wert bei der Überprüfung der Aufgaben: Die vom Studenten eingegebene Werte werden an das Mischpult übergeben und der Student kann überprüfen, wie seine Fourier-Reihe aussieht. So werden z.B. Fehler in der Berechnung der Phasen besonders deutlich. Auch zum aufgabenunabhängigen Experimentieren ist das Modul geeignet und bietet eine „Entspannungsübung“ an. Die gleichzeitigen Durchführung und Vergleich der Ergebnisse mehrerer Experimente kann durch Aufruf mehrerer Fenster geschehen.

Ein Beispiel (Abbildung im Anhang A2) aus dem Programm Grundstromkreis zeigt die Möglichkeit, interaktive Experimente mit Macromedia Flash<sup>43</sup> zu realisieren<sup>44</sup>. Mit dem Flash-Modul kann die physikalische Bedeutung einer Generatorlast veranschaulicht werden. Der Lernende hat die Möglichkeit den Widerstandswert eines Schiebewiderstandes einzustellen. Gleichzeitig mit der Änderung des numerischen Wertes unter dem Widerstandsbild sieht man die Änderung der Kennlinien von beiden Elementen im Grundstromkreis. Wird die Last so klein, dass der Grenzwert des zulässigen Stromes überschritten wird, so geht der Generator kaputt (Arbeitspunkt auf dem Leistungsgraph ist außerhalb des Bildes) und lässt sich nicht durch Vergrößerung des Lastwiderstandes in den Arbeitszustand zurückbringen – man muss das Bauteil ersetzen (in der Simulation der Restart-Knopf).

<sup>43</sup> <http://www.macromedia.com/> (letzter Zugriff 10.02.2005).

<sup>44</sup> Dieses sowie weitere Beispiele (implementiert von anderen Autoren [Fredrich2003]) sind über GETsoft LearnWeb zu erreichen.

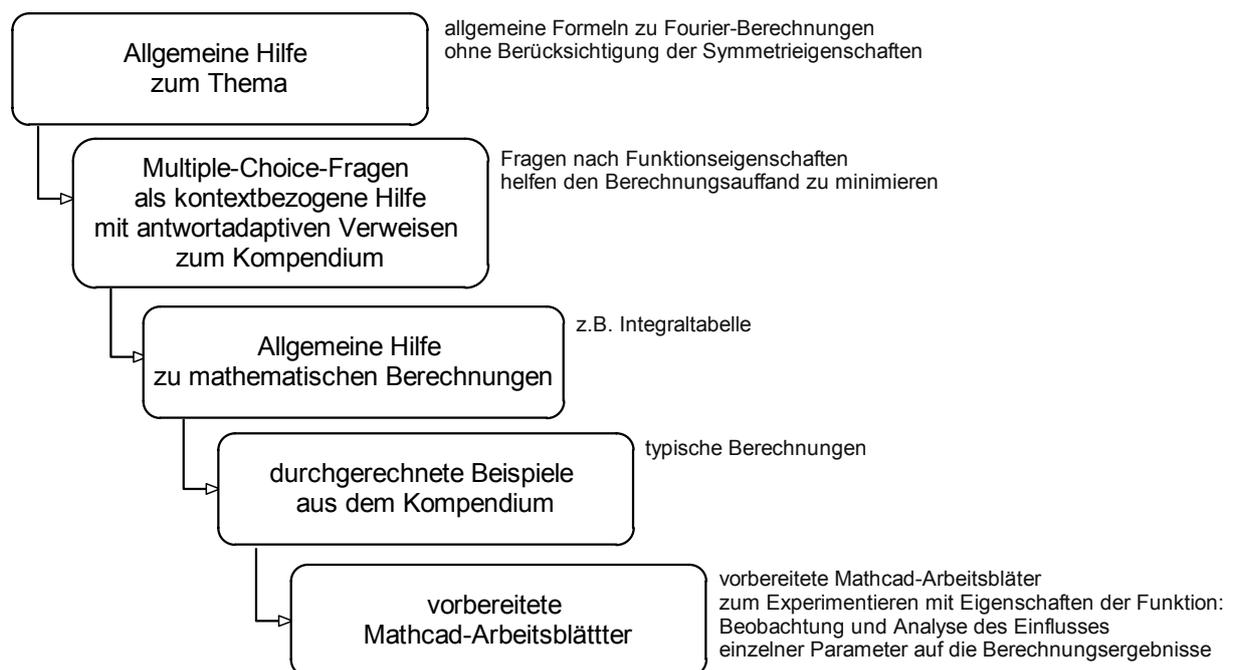
## 4.2 Feedback-Techniken

Die Feedbacktexte in den GETsoft-Lernprogrammen erfüllen zwei Bedingungen:

- Klare Rückmeldung darüber, ob die Lösung des Benutzers vom Programm als „richtig“ oder „falsch“ bewertet wird. (Nicht unbedingt nur „Ja/Nein“, wichtig ist, dass die Rückmeldung eindeutig sein soll.)
- Differenzierter Lösungskommentar zur Benutzereingabe, mit Angabe der Begründung dafür, warum eine falsche Lösung falsch ist, eine richtige richtig. Im idealen Fall werden die Lernenden dadurch genau an den Punkt geführt, an dem ihr Denkfehler oder Wissensdefizit liegt.

Es werden folgende Testformen zur Selbstkontrolle eingesetzt: Einfachauswahl, Mehrfachauswahl (Multiple-Choice), Überprüfung von Antworten in Form Zahlen-Vorsatz-Maßeinheit, graphische Form (ansatzweise).

Die nächste Abbildung zeigt einen möglichen „Dialogverlauf“ beim Lösen einer Aufgabe. Ein Ausschnitt aus dem Programm befindet sich im Anhang A2.



**Abbildung 9. Abgestufte Hilfe im Modul Fourier-Reihe**

Zu jeder Aufgabe können solche abgestuften Hilfen angefordert werden und zur Überprüfung eigener Ergebnisse kann entweder das Mischpult (das Eintragen der Benutzerwerte einzelner Harmonischen wird erwartet) benutzt werden oder ein vorgefertigtes Mathcad-Arbeitsblatt (einige Beispiele befinden sich im Anhang A3).

### Multiple-Choice

Die Multiple-Choice-Fragen werden in dem Programm Fourier-Reihen als Hilfe zu Aufgabenstellungen zum Testen der Symmetrieeigenschaften eingesetzt.

Jede Antwortmöglichkeit wird mit einem Koeffizient gewichtet:  $K_i = k \cdot 2^i$ , wobei  $k = 1$ , wenn die Antwort markiert ist, sonst  $k = 0$ ;  $i$  - die Position der Antwortmöglichkeit. Die Summe aller Koeffizienten ergibt die Nummer  $n$  der zu liefernden Rückmeldung:

$$n = \sum_{i=0}^{N-1} K_i,$$

wobei  $N$  die Anzahl der vorgesehenen Antwortmöglichkeiten ist.

Durch die Implementierung auf der Basis der binären Kodierung der Auswahlmöglichkeiten in Multiple-Choice wurden die Editierbarkeit und Kommentierbarkeit dieser Art von Testmöglichkeiten im Vergleich zur ToolBook-Implementierung erheblich verbessert. Diese Technik kann für beliebige Multiple-Choice-Aufgaben eingesetzt werden. Auch wenn Multiple-Choice-Tests allgemein als didaktisch bedenklich kritisiert werden, lassen sich mit geschickt formulierten Fragen das Wissens- bzw. das Verständnisniveau oder die Aufmerksamkeit der Lernenden schnell kontrollieren [Rose2000].

### Zahlenwerte, Vorsatz- und Maßeinheiten: Antwortüberprüfung

Die numerischen Antworten in den Aufgaben der Grundlagen der Elektrotechnik bestehen in der Regel aus einem Zahlenwert und einer Maßeinheit: 220V, 5mA, 10MHz, 0.5ms.

Zwar gehört die Umrechnung und Ableitung von verschiedenen SI-Einheiten und deren Vorsätzen zu Grundvoraussetzungen für ein Studium an einer technischen Universität, leider stellen schon die Umrechnungen von Vorsätzen für einige Studierenden ein Problem dar. Für die Webversion der Lernumgebung wurde ein Java-Script für die Überprüfung von Antworten in Form Zahlenwert-Vorsatz-Maßeinheit implementiert. Die Eingaben der Studierenden werden geparkt und auf typische Fehler hin untersucht. (z.B. ein Potenzfehler oder typische Verwechslung von Einheiten). Diese Routine wird bei der Überprüfung von Aufgabenantworten eingesetzt.

### Fourier-Taschenrechner

Für die Antwortüberprüfung beim Aufgabenlösen sowie für das Experimentieren wurde ein spezielles Tool, der Fourier-Taschenrechner, implementiert. Er steht in drei Varianten zur Verfügung: für die Berechnung der Fourier-Koeffizienten ausgewählter Funktionen, für die

Berechnung der Fourier-Koeffizienten der Übertragungsfunktionen und für die Übertragungssysteme.

Als Alternative zur textbasierten Antworteingabe wurde ansatzweise eine **graphische Form der Antwortüberprüfung** implementiert. Bei qualitativen Aufgaben werden nicht die numerischen Werte abgefragt, sondern z.B. der Charakter einer Abhängigkeit. In diesem Falle eignen sich Funktionsplotter als Eingabemöglichkeiten. In dem angegebenen Beispiel wird vom Studierenden ein Spannungsverlauf abgefragt. Der Benutzer kann einzelne Werte, die er für eine Kurve für aussagekräftig hält, per Mausklicks eingeben. Auf Anfrage wird im Hintergrund die richtige Funktion dargestellt.

Das folgende Bild fasst die verschiedenen Arten der Rückmeldungen beim Aufgabenlösen im Programm Fourier-Reihen zusammen.

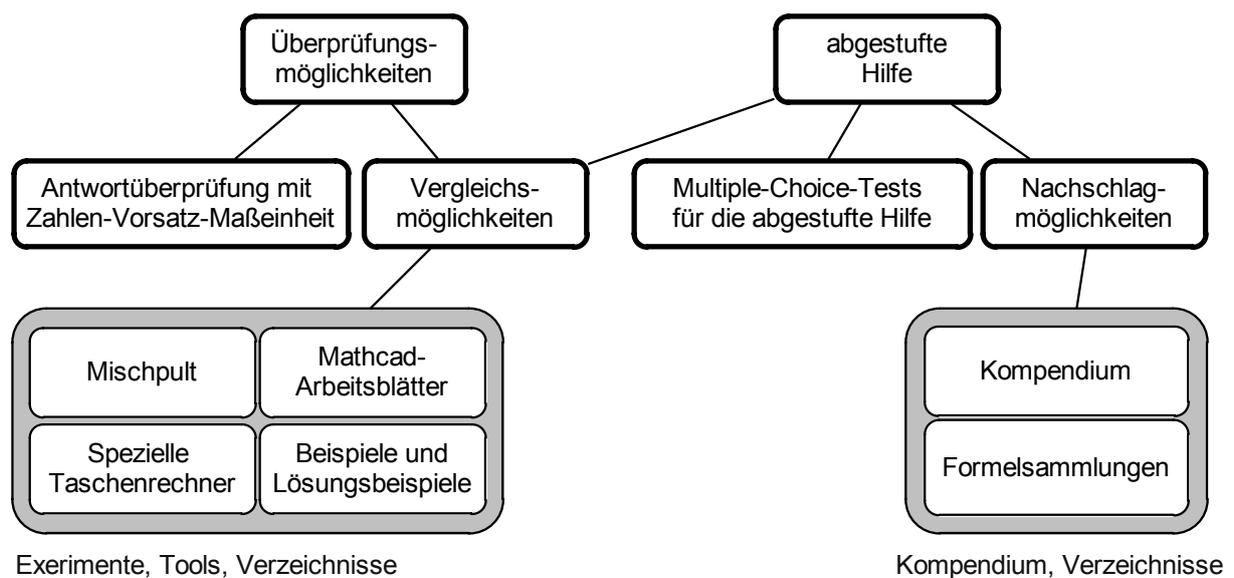


Abbildung 10. Differenzierte Feedbacktechniken in GETsoft-Lernprogramm Fourier-Reihe

Durch die Implementierung der intelligenten Komponente in GETsoft wird die Lernerunterstützung um adaptive Kommunikation beim Herleiten symbolischer Lösungen erweitert (s. Kapitel 5).

### 4.3 Mathcad als Ingenieurwerkzeug

Bei der Auswahl eines Ingenieurwerkzeuges wurden solche Kriterien wie Einsatzszenarien, Komplexität der zu bearbeitenden Aufgaben, Erlernbarkeit des Systems und Verfügbarkeit berücksichtigt. Eine Vielzahl von leistungsfähigen CAS-Systemen steht den Studierenden heutzutage zur Verfügung. Einen breiten Einsatz in der elektrotechnischen Ausbildung und in der Ingenieurpraxis finden Mathematica, Mathcad, Matlab oder Maple (siehe Abschnitt 3.3).

Vor- und Nachteile einiger CAS für die elektrotechnische Grundausbildung werden z.B. in [Ganizares1997] diskutiert. Ein Vergleich der mathematischen Fähigkeiten verschiedener CAS ist z.B. in [Wester1999] oder unter [SAC] zu finden.

Unter den vorhandenen Computer-Algebra-Systemen erfüllte Mathcad alle oben aufgeführten Kriterien optimal. Beispiele der Anwendung von Mathcad in der elektrotechnischen Ausbildung sind z.B. in [Metz1998], [DeLysser1996], [Harger1996], [Nigim2001] zu finden. Es handelt sich entweder um die Mathcad-eBooks oder um einzelne Mathcad-Arbeitsblätter. Mathsoft [Mathcad2] bietet ebenfalls eine Reihe von Mathcad-Dokumenten bzw. Links zu Mathcad-Arbeitsblättern zu verschiedenen Wissenschaftsbereichen.

Mit der Entwicklung komplexer Softwareprodukte (z.B. mileET) hat sich die Geschlossenheit des Mathcad-Systems als nachteilig erwiesen. Während Mathematica eine Schnittstelle zu modernen Programmiersprachen wie Java oder C anbietet, bietet Mathcad diese Funktionalität nicht an. Zwar können eigene Funktionen als *dll* in Mathcad eingebunden werden oder Mathcad-Arbeitsblätter als OLE-Objekte in andere Windows-Anwendungen eingefügt werden, der Zugriff auf vorhandene Mathcad-Bibliotheken lässt sich nicht ohne aufwändiges Programmieren realisieren und eine Schnittstelle direkt zu Java stand zur Zeit der Durchführung dieser Arbeit nicht zur Verfügung.

Die ersten Programmversionen in GETsoft wurden mit ToolBook 5.0 erstellt. Angebonden wurde die Version Mathcad 6.0. Die Zusammenarbeit beider Programme ist durch DDE<sup>45</sup>-Programmierung und OLE1<sup>46</sup> realisiert worden. Alle Parametereingaben wurden in ToolBook vorgenommen, kontrolliert und gegebenenfalls kommentiert zurückgewiesen. Die „guten“ Werte wurden an Mathcad übergeben. Die Ergebnisse der Berechnung (grafische Darstellungen) wurden aktualisiert in die ToolBook-Oberfläche integriert. Der Lernende brauchte nicht mit Mathcad vertraut zu sein. Unzulässige Werte werden nicht an Mathcad übergeben. Der Nutzer hat keinen direkten Zugriff auf die Quelldateien. Die kostenlose 14-Tagen-Testversion

---

<sup>45</sup> DDE - Dynamic Data Exchange

<sup>46</sup> OLE - Object Linking and Embedding

von Mathcad 6.0 konnte wegen des Zeitlimits nicht als Ingenieurwerkzeug den Studenten bei der Nutzung unserer Programme im Selbststudium angeboten werden.

Seit Mitte 1999 stand der Mathcad-Explorer (ab Version 7) kostenlos zur Verfügung. Die für GETsoft relevante Einschränkung des Mathcad-Explorers besteht darin, dass Dokumente nicht gespeichert werden können. Die GETsoft Lernprogramme wurden auf ToolBook Version 7.1 und Mathcad Version 7 umgestellt.

Mathcad ab Version 7 und ToolBook ab Version 7 unterstützen OLE2 als Technologie zum Datenaustausch. Deshalb wurden die Lernprogramme auch vom technischen Konzept des Zugriffs auf Mathcad verändert. Die Lernprogramme greifen auf Mathcad als OLE-Serveranwendung zu. Mathcad Dokumente werden als eigenständige Anwendungsfenster geöffnet.

Der Lernende kann die volle Funktionalität von Mathcad als Ingenieurwerkzeug nutzen. Komplexe elektrotechnische Sachverhalte (Ausgangsgleichungen, Auflösen numerisch oder symbolisch, Parametereinfluss u. a.) können schrittweise aufgeklärt und verständlich gemacht werden. Die Nutzung eines leistungsfähigen Ingenieurwerkzeuges zur Lösung elektrotechnischer Sachverhalte zu erlernen, ist ein dabei beabsichtigter Nebeneffekt.

Ab 2000 begann die Entwicklung der online Lernprogramme für GET. Das Konzept des Ingenieurwerkzeuges blieb in den Web-Versionen der GETsoft Lernprogramme erhalten.

Für das Lernprogramm „Grundbegriffe, Zweipole, Grundstromkreis“ wurden sechs Arbeitsblätter erstellt (siehe Beispiel im Anhang A2). Schwerpunkt ist die Darstellung linearer und nichtlinearer aktiver und passiver Kennlinien sowie die grafische und analytische Ermittlung von Arbeitspunkten beim Zusammenschalten eines aktiven und passiven Elements (Grundstromkreis). Alle Beschreibungsgleichungen sind dargelegt. Neben der Darstellung und Auswertung von Grafiken werden folgende Leistungsmerkmale genutzt: Symbolisches Lösen von Gleichungen, numerisches Lösen eines linearen Gleichungssystems, Bestimmung der Nullstellen eines Potenzpolynoms, Auflösen einer nichtlinearen Gleichung (Newtonsches Verfahren). In Mathcad-Popup-Fenstern wird die Vorgehensweise erläutert.

In den Mathcad-Arbeitsblättern wird den Lernenden empfohlen, in markierten Bereichen Parameteränderungen vorzunehmen. Der versierte Mathcad-User kann die Experimente erweitern, eigene Grafiken erstellen und selbständig Probleme lösen.

Im Mathcad-Arbeitsblatt „Grundstromkreis mit nichtlinearer Spannungsquelle“ (siehe Anhang A2) wird ein Beispiel für graphische und algebraische Ermittlung des Arbeitspunktes einer Schaltung mit dem nichtlinearen Bauelement dargestellt. Durch Ablesen der

Koordinaten des Schnittpunktes der Kennlinien der Energiequelle und der Last können die  $I$ - und  $U$ -Werte ermittelt werden. Mithilfe der Mathcad-Funktion *nullstellen(vektor)* lassen sich die Wurzeln einer quadratischen Gleichung finden. Hier sollte der Lernende den physikalischen Sinn der Lösung berücksichtigen: nur die positive Wurzel ist die Lösung der Aufgabe.

Im Lernprogramm „Fourier-Reihen“ werden 26 Mathcad-Arbeitsblätter für verschiedene Schwerpunkte des Themas „Netzwerke bei nichtsinusförmiger periodischer Erregung“ angeboten. Für zehn typische Signalfunktionen ist die Fourier-Zerlegung und Fourier-Synthese vorbereitet. Die Verzerrung des Amplituden- und Phasenspektrums für sieben wichtige Übertragungssysteme kann für unterschiedlichste Parameter untersucht werden. Selbstverständlich können die Arbeitsblätter an jeder Stelle des Lernprogramms (Experimentierumgebung, Kompendium, Aufgabenbearbeitung) aufgerufen werden. Wichtige Aussagen, wie z. B. die Veränderung des Klirrfaktors einer Signalspannung bei Filterung könnten ohne Mathcad (d.h. eines CAS Systems) nicht befriedigend getroffen werden. Ein Beispiel befindet sich im Anhang A2. Ein interaktives Tool für Fourier-Konzepte auf Matlab-Basis wird in [Hamann2003] vorgestellt.

Zu dem Themenfeld „Vorgänge auf Leitungen“ sind die Mathcad-Arbeitsblätter für die Visualisierung komplexer Sachverhalte bei der Wellenausbreitung auf Leitungen vorbereitet worden. Schwerpunkte dieser Arbeitsblätter sind z.B. Wellenausbreitung auf (verlustlosen) Leitungen beim Einschalten einer Gleichspannung, einer Pulsspannung, bei sinusförmiger Anregung oder bei Anregung mit einer Pulsfolge.

Die vielfältigen Einflussgrößen bei Vorgängen auf Leitungen erfordern eine behutsame didaktische Anleitung und Hilfestellung beim Experimentieren. Dem Lehrenden eröffnet sich ein innovatives Potential zur Visualisierung technisch höchst bedeutsamer komplexer Zusammenhänge. Eine umfangreiche Leitungssimulation ist mit PSpice möglich. Einige Simulationstools für die Vorgänge auf Leitungen sind in [Trueman2003, Khan2001] vorgestellt.

Im Lernprogramm „Laplace-Transformation“ sind die Arbeitsblätter ein Bestandteil der Unterstützung des Lösens komplexer Aufgabenstellungen. Hier zeigen sich Nachteile bei der schlüssigen Interpretation symbolischer Lösungen, weshalb die numerischen Ergebnisse mit den funktionellen Darstellungen favorisiert werden. Die Interpretation von Lösungen

komplexer Aufgabenstellungen, wie z. B. dem Anschalten von Einzelimpulsen und von periodischen Signalspannungen an Übertragungssystemen, wird erst durch die graphische Ergebnisausgabe (Ausgleichsvorgang, periodische Lösung) überhaupt möglich.

Mathcad Arbeitsblätter für die Übungen AET-I, AET-II<sup>47</sup> wurden im Laufe der WS2000/2001 – WS2001/2002 entwickelt und in den Übungen eingesetzt. Die Studierenden bekamen während der Übungen die Möglichkeit, die zu lösenden Probleme qualitativ zu diskutieren, sowie die Möglichkeit die symbolischen Lösungen numerisch einzuschätzen und zu beurteilen; die genaue Berechnung der Werte wurde allerdings auf Mathcad-Arbeitsblätter verlegt.

Die Studierenden hatten sowohl die Möglichkeit ihre eigenen Mathcad-Arbeitsblätter zu erstellen und an den Übungsbetreuer per E-Mail zu versenden, als auch die vom Übungsleiter vorgefertigten Arbeitsblätter-Vorlagen zu benutzen.

#### **4.4 Zusammenfassung**

Durch die Aufteilung der GETsoft-Lernprogramme in Module wie Kompendium, Aufgabensammlung und Experimentierumgebung sind mindestens drei Lernszenarios denkbar: Theorieorientiertes Lernen, Aufgabenorientiertes Lernen, Experimentorientiertes Lernen. Die Lernenden mit unterschiedlichen Lernstilen ([Larkin-Hein2001]) und Wissensniveaus können somit mit der Lernumgebung arbeiten. Das erste Szenario ermöglicht das geführte Lernen (instruktionsorientiert): der Studierende liest theoretische Grundlagen nach der vom Autor vorgegebenen Reihenfolge. An entsprechenden Stellen wird auf Aufgaben und Experimente verwiesen, so dass alle Teile des Programms durchgearbeitet werden. Aufgabenorientiertes (problemorientiertes) Lernen sieht vor, dass der Lernende erst in einer Stocksituation, z.B. bei einem Fehler bzw. beim Wissensmangel, während des Aufgabenlöseprozesses sich das Kompendium oder ein Experiment zu Rate zieht und diese so als Hilfsmittel verwendet. Das Experimentorientierte Lernen bedeutet eigenes Wissen durch (relativ) freies Experimentieren, durch Beobachtungen und Analyse zu konstruieren, mithilfe der Aufgaben das erworbene Wissen zu testen und ggf. notwendige Informationen und Erklärungen aus dem Kompendium bei Bedarf zu holen.

---

<sup>47</sup> AET- I, II – Allgemeine Elektrotechnik - 1. Semester, 2. Semester

Mathcad als Ingenieurwerkzeug wird im Rahmen von GETsoft zum Bearbeiten der für GET relevanten Aufgaben eingesetzt. Das System ist leicht zu erlernen und ähnelt in der Handhabung der Arbeit mit Papier und Bleistift, es erlaubt überschaubare Ergebnisse schnell zu erreichen.

Für die selbständige offline Bearbeitung der GET-Aufgaben reichen die in Mathcad zur Verfügung stehenden Funktionen völlig aus. Unter bestimmten Installationsvoraussetzungen (Mathcad-Vollversion, Browsereinstellungen<sup>48</sup>) kann auch online gearbeitet werden. Die online Bibliothek von Mathcad-Ressourcen bietet die Möglichkeit, über die vorbereiteten Aufgaben hinaus die Anwendung der Mathcad-Funktionen für verschiedene Wissensdomänen kennen zu lernen.

Die Implementierung der intelligenten Lernsysteme mit Formelverarbeitung benötigt ein System, das Schnittstellen zu mächtigen Programmiersprachen anbietet. In dieser Hinsicht bietet Mathcad wesentlich weniger Unterstützung als z.B. Mathematica. Als Lösungswegweiser könnten weiter in diesem Abschnitt vorgestellte prototypische Realisierung der Formelüberprüfung sowie die Arbeiten [Burger1997, ActiveMath] betrachtet werden.

Ein Umstieg auf ein anderes CAS als Ingenieurwerkzeug in der Lehre ist allerdings wenig sinnvoll, da es für die Lehrenden einen großen Aufwand bedeutet: eine Neuerstellung der ganzen Sammlung von Dokumenten sowie größere Einarbeitungszeiten für Lernende.

Die GETsoft-Lernumgebung wird seit 1998 in der Lehre eingesetzt. Die Überarbeitung der ToolBook-Version des Programms Fourier-Reihe zu einer netzbasierten Version hat im Jahre 2000 begonnen. Im Rahmen des mile-Projektes wurde von Kollegen eine Evaluation der Lernprogramme durchgeführt. Das Ziel dieser Evaluation war nicht nur die Handhabung der multimedialen Lernumgebung sondern vielmehr ihre Einbindung in den Studienbetrieb. Eine umfassende Evaluation in Form von Fragebogenbefragung, Unterrichtsbeobachtungen, Gruppendiskussionen und Try-Outs hat im Laufe SS2001-SS2002 stattgefunden [Aslanski2002]. Die Ergebnisse des Try-Outs haben die Verbesserungspotentiale der Lernprogramme aufgezeigt. Die meisten Probleme betreffen jedoch Detailfragen (hauptsächlich Design), die aber nicht die gute Bedienbarkeit als solche in Frage stellen. Auch eine andere projektinterne Evaluation des Lernprogramms Fourier-Reihe hat einige Schwachpunkte hinsichtlich des multimedialen Designs aufgedeckt, die in den aktuellen Versionen zu verbessern sind.

---

<sup>48</sup> Zu den neueren Lösungen gehört der Mathcad Application Server von Mathsoft

Eine Befragung der Studierenden durch die Übungsleiter ergab ebenfalls eine positive Beurteilung des Programms Fourier-Reihen. Die meisten Studierenden finden den Einsatz der Lernprogramme als Begleitung zur traditionellen Lehre sehr sinnvoll.

Eine Reihe von Vorarbeiten die in einer erweiterten oder geänderten Form in die GETsoft Umgebung eingegangen sind bzw. als Vorarbeiten für mileET anerkannt werden könnten, wurden im Rahmen von studentischen Praktika erprobt.

Im Rahmen einer Praktikumsbetreuung wurde eine **online Datenbank für Multimedia-Lerneinheiten** konzipiert und prototypisch implementiert (interner Praktikumsbericht Matschin, K., Yakimchuk, V.). Die Datenbank ist für die Verwaltung von Informationen über alle für den Lehrprozess relevante Dokumente bzw. Dateien konzipiert worden. Die Informationen über die technischen Eigenschaften einer Datei bzw. eines Moduls sowie über die didaktische Rolle der Lerneinheit sind in der Datenbank erfasst worden. Die Suche ist in verschiedenen Datenbank-Feldern möglich.

Eine Überarbeitung und Weiterentwicklung dieses Ansatzes hat im Rahmen des mile Projektes stattgefunden und steht als TaskWeb online zur Verfügung.

Der Prototyp des Programms zur symbolischen **Formelüberprüfung** wurde im Rahmen der Betreuung eines Praktikums (interner Praktikumsbericht Stasjuk, A., Yakimchuk, V.) konzipiert und für die Nutzung in einem multimedialen Lehrprogrammkomplex als Zusatzmodul implementiert. Das Modul besteht aus einer GUI Komponente (in C++ implementiert), die auf den Mathematica-Kernel zugreift. Der Datenaustausch findet über Dateien statt. Aufgrund der hohen Kosten und der Möglichkeit, den Zugriff auf Mathematica nur in den PC-Laboren der TU Ilmenau zu ermöglichen, wurde dieser Ansatz leider nicht weiterentwickelt. Die aktuelle Situation auf dem Campus der TU Ilmenau macht diese Richtung der Arbeit wieder attraktiv.

Im Rahmen eines weiteren Praktikums wurde ein Versuch unternommen, einen Schaltungseditor (Java-Applet) an das **PSpice**-Simulationsprogramm (Server) anzubinden, um die Implementierung eines online Schaltungseditors zu ermöglichen (interner Praktikumsbericht Smolin, S.). Die Arbeit blieb leider in der Konzeptions- bzw. Entwurfsphase. Ähnliche Ziele werden an der University of Central Florida verfolgt [Harb2004], so dass eine Zusammenarbeit auf diesem Gebiet perspektivisch erscheint.

Die webbasierte multimediale Lernumgebung bietet den Lehrenden viele multimediale Komponenten für das Zusammenstellen ihrer Vorlesungen und Übungen. Die Studierenden

haben den Zugang zu den typischen Themen und Problemstellungen des Lehrgebietes. Zu diesen Themen angebotene kommentierte Experimente und Aufgaben unterstützen das selbständige Lernen<sup>49</sup>. Interaktive Experimente und Aufgaben mit Möglichkeiten zur Selbstkontrolle ermöglichen eine aktive Verarbeitung des Lehrstoffes. Die Lernumgebung bietet allerdings nicht ausreichend Unterstützung beim Auflösen: partielle Lösungsentwürfe werden nicht analysiert, neue (eigene) Aufgaben lassen sich nicht erstellen und überprüfen. Den Studierenden und Lehrenden wird kein Tool zum Aufbereiten und Aufbewahren ihrer Aufgaben und Aufgabenlösungen angeboten. Ein Versuch, dieses Defizit zu beheben, stellt die wissensbasierte Lernumgebung mileET dar, die im nächsten Kapitel vorgestellt wird.

---

<sup>49</sup> Seit 2004 wird im Rahmen GETsoft auch ein online Forum in der Lehre eingesetzt.

## 5 Intelligente Lernerunterstützung in der wissensbasierten Problemlöseumgebung mileET

### 5.1 Konzeption einer intelligenten Lernumgebung für GET

#### 5.1.1 Didaktische und softwaretechnische Konzeption

Die IPSE mileET wurde nach der an der Universität Oldenburg entwickelten kognitiven Theorie des Wissenserwerbs ISP-DL (siehe Abschnitt 2.2, 2.4) konzipiert. MileET unterstützt die Lernenden beim aktiven Aufgabenlösen und erfüllt die meisten didaktischen Anforderungen an computerbasierte Lernmedien: *„(1) Lehren und Lernen sollen jeweils von einer - für die Lernenden bedeutsamen - Aufgabe ausgehen. ... (2) Lehren und Lernen sollen darauf gerichtet sein, vorhandenes Wissen oder bestehende Fertigkeiten zu einem Themengebiet zu aktivieren und – von dort ausgehend – eine Korrektur, Erweiterung, Ausdifferenzierung oder Integration von Wissen und Vorstellungen zu erreichen. (3) Lehren soll eine aktive und kooperative Auseinandersetzung der Lernenden mit einer Aufgabe ermöglichen, indem - auf der Basis geeigneter Informationen - selbstständig Lösungswege entwickelt und erprobt werden. (4) Lehren soll den Vergleich unterschiedlicher Lösungen ermöglichen sowie eine Systematisierung und Anwendung angemessener Kenntnisse und Vorgehensweisen sowie deren Weiterführung und Reflexion.“* [Tulodziecki2004]

Die mileET-Problemstellungen sind Aufgabenstellungen aus der GET-Domäne. Die Aufgabensammlungen der Universitäten Dresden, Ilmenau und Magdeburg dienten dabei als Basis für die Auswahl der Aufgaben.

MileET kann sowohl als selbständige Übungsumgebung als auch als Teil einer GET-Lernumgebung gesehen werden und kann somit in die anderen didaktischen Konzepte eingebunden werden.

Abbildung 11 zeigt die wichtigsten Funktionen, die von der Lernumgebung erwartet werden. Eine zusätzliche Markierung tragen die Funktionen, die mileET zu einer an das Aufgabenlösen orientierten intelligenten Lernumgebung machen.

Zwei Modi sind notwendig, um sowohl für die Lernenden als auch für die Lehrenden die Benutzung von mileET attraktiv zu machen. Die Stärke des Dozentenmodus liegt in der Möglichkeit neue Aufgaben zu festgelegten Themenfeldern zu erstellen. Die Möglichkeit, eine Anpassung bzw. Änderung von Lösungswegkommentaren vorzunehmen, soll für die Lehrkräfte (Aufgabenautoren) aus unterschiedlichen Ausbildungseinrichtungen von Interesse sein. Der Studentenmodus bietet den Lernenden die Möglichkeit, ihre eigenen symbolischen

Lösungsentwürfe zu erstellen, sie überprüfen oder ggf. vervollständigen zu lassen und eine adaptive Hilfe anzufordern. Diese Funktionalität unterscheidet die mileET-Lernumgebung von den meisten Tools für die GET-Lehre.

Zu spezifischen Anforderungen an eine Lernumgebung für die Grundlagen der Elektrotechnik gehören die Möglichkeiten, Schaltbilder, Formeln und Lösungsentwürfe zu erzeugen und zu bearbeiten. Spezielle maßgeschneiderte Editoren wurden konzipiert und von studentischen Mitarbeitern unter Betreuung implementiert.

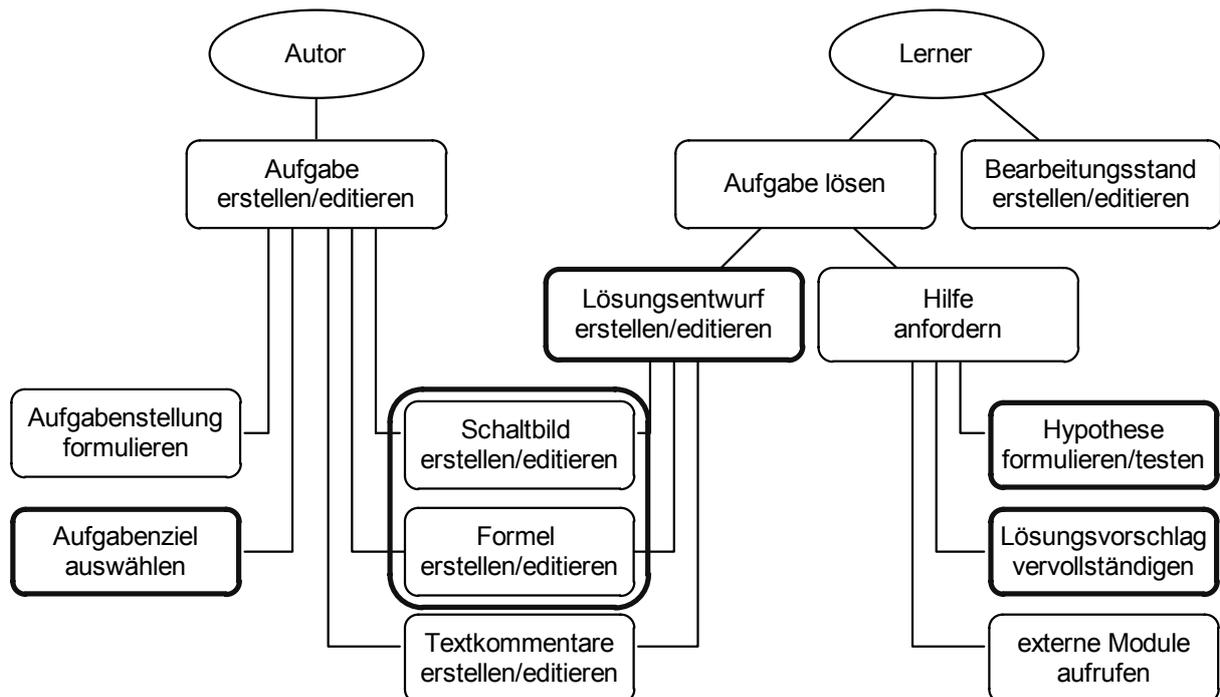


Abbildung 11. Anwendungsfälle für mileET

Die markierten Elemente in der Abbildung 11 stellen die besondere Leistung von mileET dar. Die Aufgabenautoren können neue Aufgaben erstellen, die Lernenden können diese Aufgaben in mileET lösen und ihre Bearbeitungszustände speichern. MileET kann kontextsensitive Hilfen und adaptive Lösungshinweise liefern.

Das Programm wurde so konzipiert, dass eine Übertragung in andere Sprachen möglich ist und somit eine Aufnahme in andere Lernumgebungen. Eine mögliche Erweiterung der Benutzerschnittstelle auf andere Elemente und Themengebiete wurde berücksichtigt (der Gedanke der offenen Lernumgebung). Im Weiteren werden die Auswahl der Themenfelder für die Wissensbasis und die Aufgabendarstellung in mileET vorgestellt.

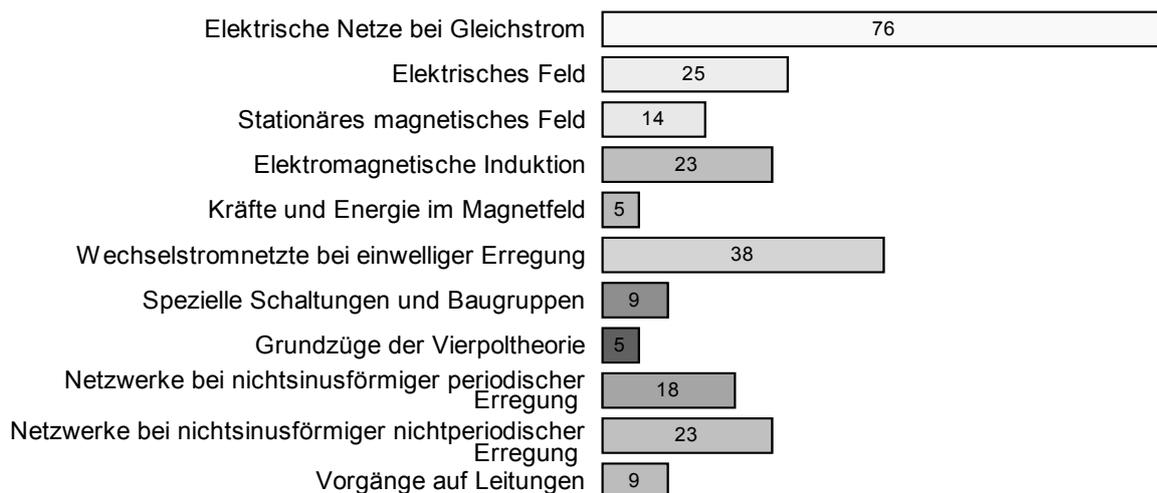
### 5.1.2 Auswahl der Themenfelder und typischer Aufgaben

Die Grundlagen der Elektrotechnik befassen sich mit einem breiten Themenspektrum: von Grundlagen der Elektrizitätslehre bis zu Vorgängen auf Leitungen [Seidel1999, 2001,

Ionkin1976, Chang1982]. Eine aktive konsequente Auseinandersetzung mit dem Fachwissen findet in Form von Aufgabenlösen und Experimenten statt.

Laut VDE-Grundpositionen zur Ingenieurausbildung, muss nicht „jede Spektrallinie der späteren Berufsarbeit auch an den Hochschulen gelehrt und vermittelt werden. Vielmehr kommt einer soliden Erstausbildung mit einem fundierten fachlichen Grundlagenwissen die größte Bedeutung zu. Die Vermittlung methodischer und wissenschaftlicher Grundlagen ist wichtiger als reines Faktenwissen“ [Wucherer2002].

Die Abbildung 12 zeigt die Verteilung der Aufgaben über die Themenkomplexe der GET-Lehre. Die Gruppe der Themen, die unter dem Überbegriff „Elektrische Netze bei Gleichstrom“ aufgefasst sind, ist die Grundlage der GET-Ausbildung und enthält eine große Anzahl von Aufgaben. Zu dieser Gruppe gehören die Methoden der Netzwerkanalyse. Sie sind als Schwerpunkt der ersten Version der Lernumgebung mileET festgelegt worden. Dieser Themenkomplex wird auf der Basis der Gleichstromnetzwerke im ersten Semester der GET-Ausbildung geübt und später bei der Analyse der Wechselstromnetzwerke angewendet.



**Abbildung 12. Thematische Verteilung der GET-Aufgaben<sup>50</sup>**

Ein weiterer Einflussfaktor ist die Anzahl der potenziellen Nutzer. Der Vergleich der Anforderungen bzw. des Lehrangebotes verschiedener Universitäten ergibt, dass das Themenfeld Grundlagen der Netzwerkanalyse weltweit zu den Grundvoraussetzungen jeder technischen Aus-, Fort- und Weiterbildung gehört [z.B. Aspnes1994]. Schüler und Studenten der Fachhochschulen und Universitäten haben die Themen wie Anwendung der Kirchhoff'schen Sätze oder äquivalente Transformationen in ihren Lehrplänen. Eine breite

<sup>50</sup> Es wurden nur die Aufgaben aus der Aufgabensammlung des Fachgebietes GET berücksichtigt. Eine Reihe von Aufgaben wird zurzeit direkt über Lernmodule angeboten und ist noch nicht in der Aufgabensammlung erfasst. Die Analyse von vorhandenen Aufgaben und die Zusammenstellung einer interaktiven Aufgabensammlung ist der Inhalt von momentan durchführenden Arbeiten.

Benutzergruppe mit dem verschiedenen Vorwissensniveau, mit unterschiedlichen Ausbildungszielen und aus verschiedenen Ausbildungseinrichtungen soll somit den Zugang zu der auf dem GET-Gebiet neuartigen Lernumgebung bekommen.

Das tiefe Verständnis des Grundlehrstoffes von Anfang an ist eine Voraussetzung für erfolgreiches Erlernen späterer sehr komplexer und komplizierter Themenfelder. Eine kompetente und geduldige Unterstützung der Studierenden besonders beim Aufbau von Grundwissen ist von großer Bedeutung. Je weiter im Studium eines Faches der Lernende sich befindet, desto selbständiger wird er und desto weniger externe tutorielle Hilfe wird er brauchen. Simulations- und Modellierungstools übernehmen im späteren Stadium die zentrale Rolle. Im Gegensatz dazu ist in den früheren Stadien eine didaktische Unterstützung sehr wichtig. Der Prozess der Aneignung von Grundmethoden und Grundtechniken des Faches findet in der Präsenzlehre statt. Innerhalb der Übungszeit bleibt kaum Zeit, verschiedene Lösungsvarianten der Studierenden zu überprüfen und auf alle Fehler im Einzelnen einzugehen. Häufig sind die Studierenden in der Lage, die Fehler in ihren Lösungen selbständig zu lokalisieren, falls sie dazu aufgefordert werden. Eine Lernumgebung, die die Studierenden während des Aufgabenlöseprozesses unterstützen würde, verschiedene Lösungswege erkennen und analysieren könnte, sowie verschiedene Aufgaben selbstständig lösen würde, ohne dass die Lehrkräfte die Aufgaben vorher durchrechnen müssen, wäre sowohl für Studierende als auch für Lehrende eine große Unterstützung in der Lehr-/ Lernpraxis.

Die Abbildungen 13 – 15 stellen beispielhaft einige typische Aufgaben aus dem Themenfeld „Methoden der Netzwerkanalyse“ dar<sup>51</sup>.

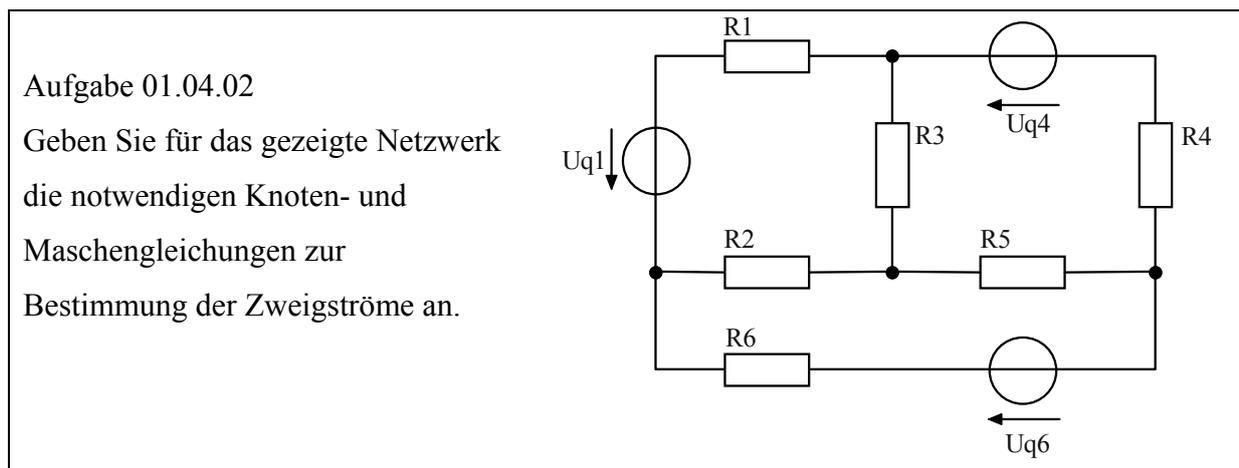


Abbildung 13. Beispiel einer Aufgabe aus dem Themenfeld Kirchhoff'sche Gleichungen

<sup>51</sup> Die Aufgabenstellungen sind der aktuellen Aufgabensammlung aus der TaskWeb entnommen. Die Formulierungen können an einigen Stellen (z.B. Bezeichnung von Elementen) von dem Original abweichen.

Aufgabe 01.06.01

Berechnen Sie den Ersatzwiderstand der angegebenen Widerstandskombinationen zwischen den Anschlussklemmen.

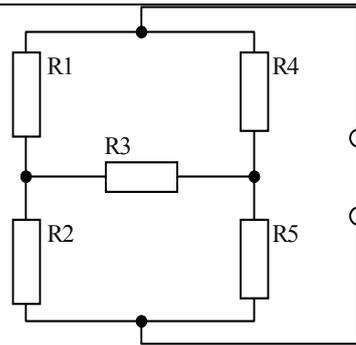


Abbildung 14. Beispiel einer Aufgabe aus dem Themenfeld Berechnung der Widerstandskombinationen

Aufgabe 01.07.08

Berechnen Sie  $I_6$  mit Hilfe der Zweipoltheorie.

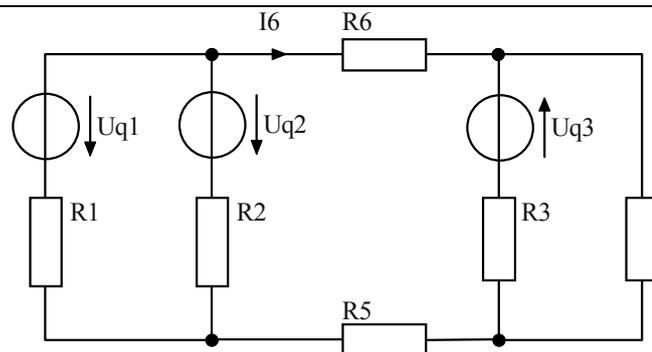


Abbildung 15. Beispiel einer Aufgabe aus dem Themenfeld Anwendung der Zweipoltheorie

Aufgabe 01.08.04

Stellen Sie das Gleichungssystem zur Berechnung der Knotenspannungen auf und geben Sie die Beziehungen zur Berechnung der Zweigströme an.

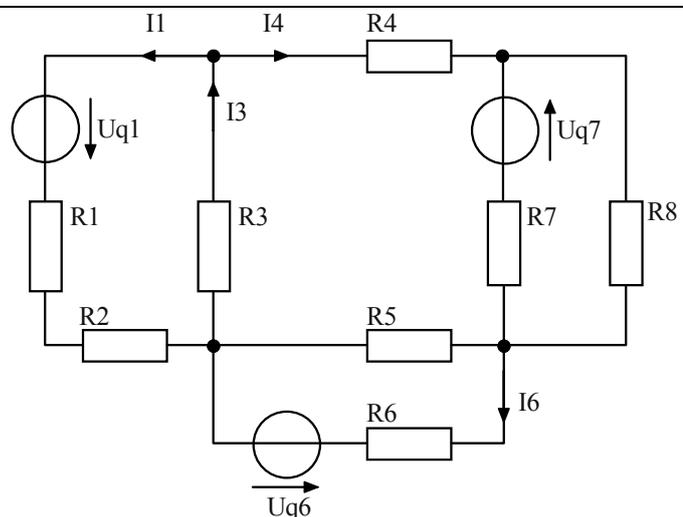


Abbildung 16. Beispiel einer Aufgabe zum Thema Anwendung der Methode der Knotenspannungsanalyse

Folgende Tabelle gibt einen Überblick über die Aufgabenstellungen zu den ausgewählten Themenfeldern aus den Aufgabensammlungen der TU Ilmenau, der Universität Magdeburg und der TU Dresden. Typisch für die Aufgabenstellungen ist die Vorgabe einer Schaltung, welche zu analysieren ist. In seltenen Fällen ist ein Schaltbild zu erstellen.

Thema	Aufgabendaten	Aufgabenziele
Allgemeines	Schalbild	Anzahl von Knoten, Zweigen bestimmen
Kirchhoff'sche Sätze	Schalbild	Gleichungssystem aufstellen
	Schalbild mit einzelnen Parametern	alle/einige Spannungen, Ströme bestimmen
Superpositionsprinzip	Schalbild	Strom bzw. Spannung in bzw. an einem Element bestimmen
Maschenstromanalyse	Schalbild	Gleichungssystem aufstellen
	Schalbild	Ströme bestimmen
Knotenspannungsanalyse	Schalbild	Gleichungssystem aufstellen
	Schalbild	Ströme bestimmen
	Schalbild	Spannungen bestimmen
Zweipoltheorie	Schalbild	Ströme bzw. Spannungen bestimmen

**Tabelle 2. Zusammenfassung von Aufgabenzielen und Aufgabendaten.**

Für die Ermittlung der gesuchten Parameter sind verschiedene Lösungsmethoden anwendbar. Abhängig davon, wie die einzelnen Komponenten der Aufgabenstellung während der Lösung behandelt werden, können die Aufgaben wie folgt klassifiziert werden:

- a) Die Aufgabenlösung enthält keine Schaltungsumformungen, die gesuchten Parameter werden durch mathematische Berechnungen ermittelt. Beispiele dafür sind z.B. die Aufgaben mit den Zielen Aufstellung eines Gleichungssystems, Berechnung eines (mehrerer) Parameters mit Hilfe der Methoden der Maschenstromanalyse oder der Knotenspannungsanalyse (Abbildung 13, Abbildung 16).
- b) Die Aufgabenlösung besteht aus einer konsequenten, explizit oder implizit durchgeführten Transformation (Vereinfachung) des Schaltbildes. Beispiele dafür sind die Aufgaben mit den Zielen Berechnung eines Eingangswiderstandes, Berechnung eines Ersatzwiderstandes, Berechnung einer äquivalenten Spannungsquelle (Abbildung 14, Abbildung 15).
- c) Die Aufgabenlösung wird erreicht durch die „parallele“ Anwendung einzelner Transformationen und die Ergebnisse jeder Transformation werden am Ende der Lösung zu einem Ergebnis zusammengeführt. Beispiele dafür sind die Aufgaben mit dem Ziel Berechnung eines (mehrerer) Parameters mit Hilfe der Methode des Superpositionsprinzips.
- d) Die Aufgabenlösung besteht in der Entwicklung einer Schaltung. Diese Art von Aufgaben ist im Grundstudium eher selten, da die Schwerpunkte im Grundstudium in der Analyse bestehender Schaltungen liegen und nicht in der Konstruktion von neuen. Die Aufgaben mit dem Ziel, eine Widerstandsmessschaltung zu erstellen, ist ein Beispiel für diese Art von Aufgaben.

Die Aufgaben a) und b) sind für mileET ausgewählt worden, für die Lösung der c)-Aufgaben werden Lösungstechniken der a) und b) kombiniert, so dass diese Aufgabentypen mit wenig Aufwand in mileET eingearbeitet werden können. Die d)-Aufgaben sind im Grundstudium sehr selten und werden in der aktuellen Version von mileET nicht wissensbasiert unterstützt. Das notwendige Wissen für die d)-Aufgaben ist in der Wissensbasis vorhanden. Die Unterstützung des Lösungsentwurfes ist im Entwurfsstadium geblieben, kann aber weiterentwickelt werden.

Zusammenfassend können die Aufgabenstellungen wie folgt formuliert werden. Für die gegebene Schaltung und die gegebenen Parameter soll ein (einige) Parameter bzw. (einige) Schaltbild(er) ermittelt werden unter Anwendung bestimmter Methoden der Netzwerkanalyse.

### 5.1.3 Modelle und Repräsentation der mileET-Aufgaben und -Lösungen

Diese für mileET ausgewählten Aufgaben und deren Lösungen wurden auf gemeinsame Merkmale hin analysiert und in einem Modell, genannt Y-Modell, graphisch visualisiert. Jede Aufgabe in mileET kann als Relation über den Mengen von Schaltbildern (S), mathematischen Gleichungen (M), Parameter (P) und Konzepten (K) aufgefasst werden.

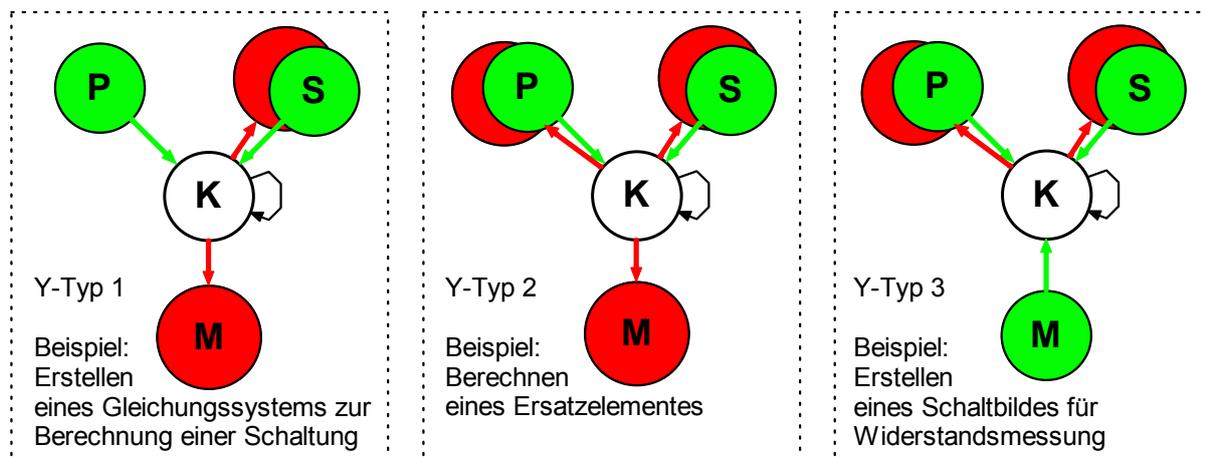


Abbildung 17. Y-Modelle

Die Information über die Schaltungstopologie wird in der Menge (S), die symbolischen und/oder numerischen Parameterwerte in der Menge (P) und Gleichungen und Formeln in der Menge (M) gespeichert. Die Mengen S, P, M lassen sich mit Hilfe von K voneinander ableiten. Dieses Ableiten repräsentiert den Prozess des Aufgabenlösen.

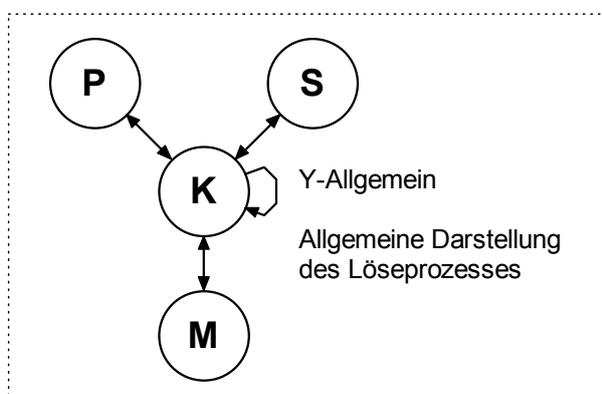
Den Aufgabenzielen und Lösungsmethoden entsprechend (Abbildungen 14 - 17) lassen sich verschiedene Instanzen des Y-Modells bilden, von denen drei Y-Typen näher betrachtet werden (Abbildung 17, Y-Typ1 bis Y-Typ3). In der Abbildung 17 sind die in der Aufgabenstellung vorgegebenen Daten als grüne Kreise dargestellt. Als Ergebnis der Aufgabe erwartete Daten sind durch rote Kreise hervorgehoben. Daten, die während eines Lösungsprozesses

erzeugt, aber nicht als Lösung erwartet werden, werden nicht dargestellt, damit das Bild überschaubar bleibt.

Y-Typ 1 sind Aufgaben mit dem Ziel, bei den gegebenen Schaltbild und Parametern mathematische Gleichungen aufzustellen (erwartete Ergebnisse liegen in M). Da allerdings bestimmte Markierungen im Schaltbild vorzunehmen sind (z.B. Zweigströme und Knotenspannungen eintragen), liegen erwartete Ergebnisse auch in der S-Menge. Die oben besprochenen Aufgaben (Abbildung 13, Abbildung 16) werden durch diesen Y-Typ modelliert. Zu Y-Typ 2 gehören die Aufgaben, in denen Parameter zu berechnen sind (ein Schaltbild und einige Parameter sind gegeben). Dabei gehören die mathematischen Gleichungen zur Lösung, wobei eventuell eine Schaltungstransformation stattfindet. Als Ergebnis wird auch die Herleitung der Formeln erwartet und somit liegen erwartete Ergebnisse in den P-, S- und M-Mengen (Abbildung 14, Abbildung 15).

Y-Typ 3 bilden die Aufgaben, in denen eine Schaltung gesucht wird, für die dann bestimmte Parameter berechnet werden müssen. Einige Parameter, mathematische Gleichungen und ein Teil der gesuchten Schaltung sind vorgegeben.

Diese Heterogenität der zu bearbeitenden Daten und die Tatsache, dass in allen drei Mengen sowohl gegebene als auch gesuchte Daten vorkommen können, stellt eine besondere Herausforderung bei der Entwicklung des wissensbasierten Assistenten dar.

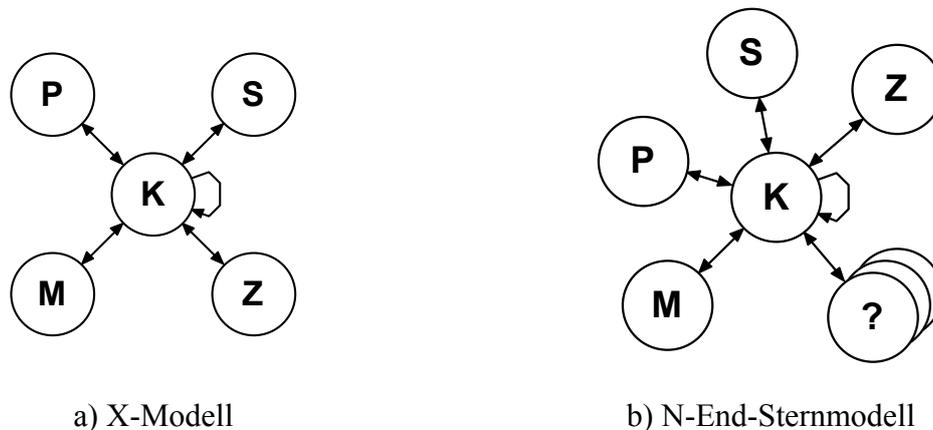


**Abbildung 18. Allgemeines Y-Modell**

Das allgemeine Y-Modell (Abbildung 18) visualisiert den Prozess des Aufgabenlöses aller in der aktuellen Version von mileET unterstützten Aufgaben. Das Y-Modell kann als graphische Darstellung des Lösungsprozesses aller Aufgaben betrachtet werden, bei denen drei Daten-Typen eine Rolle spielen: Schaltbilder, Formeln, Parameter.

Die in mileET umgesetzte Betrachtungssicht lässt sich auch auf weitere Daten-Typen anwenden, wofür eine entsprechende Anpassung des Modells vorzunehmen ist. So kann z.B. das Y-Modell zu einem X-Modell transformiert werden, falls noch ein Daten-Typ aus der

studentischen Lösung durch den wissensbasierten Assistenten analysiert werden soll (z.B. Zeigergramme, Abbildung 19, a).



**Abbildung 19. Übergang zu einem allgemeinem Aufgaben-Modell**

Mit weiteren Datentypen kommen weitere Strahlen in das Modell hinzu und das Y-Modell kann somit als ein spezieller Fall eines Stern-Modells (mit N-Enden) betrachtet werden (Abbildung 19, b).

Die zu entwickelnde wissensbasierte Komponente in mileET soll in der Lage sein, sowohl beliebige Aufgaben zu den festgelegten Themengebieten selbständig zu lösen, eigene Lösungen zu kommentieren, sowie die Lösungsentwürfe der Lernenden zu interpretieren und u. U. zu vervollständigen. Letzteres erfordert eine adaptive Reaktion auf die Eingaben des Benutzers und Flexibilität bezüglich der Anwendung verschiedener Lösungsmethoden. Damit wären die meisten Anforderungen an ein wissensbasiertes Lehrsystem GET erfüllt [Finsterbusch1990].

Alle Lösungsmethoden der mileET-Aufgaben lassen sich formal beschreiben und in einer Reihe von Algorithmen zusammenfassen<sup>52</sup>. Der Schwerpunkt der mileET-Lernumgebung liegt allerdings darin, (unvollständige) symbolische Lösungsvorschläge der Studierenden nachvollziehen und beurteilen zu können, um kommentierte Lösungsempfehlungen zu erzeugen. Diese Aufgabe erfordert eine Darstellung der Lösungsmethoden in kleinstmöglichen Schritten, die durch eine Kontrollkomponente zu einer Lösung zusammengefügt werden. Somit wird die notwendige Flexibilität gewährleistet.

Für die Kombination „Aufgabenziel-Lösungsmethode-Erklärung“ stellten die GMRs eine Basis für das mileET-Regelformat dar. GMR (engl.: goals-means-relations) bezeichnet eine Menge von Regeln im Ziel-Mittel-Relationsformat, die eine Baumgrammatik bilden. Die

<sup>52</sup> vgl. CircuitMagic, [www.circuit-magic.com](http://www.circuit-magic.com) (letzter Zugriff 10.02.2005). Das Programm löst die typischen Aufgaben mit vorgegebenen numerischen Parameter und gibt die Lösungsherleitung als Ergebnis zurück

Ziele und Mittel variieren mit der zu modellierenden Domäne, es können u.a. Implementations-Teile, chemische Elemente (inkl. Bindungen) oder Berechnungen sein. Die GMRs wurden in der Arbeitsgruppe von Prof. Möbus entwickelt und in mehreren Systemen [Thole1997, Lüdtke2002] erfolgreich eingesetzt.

Die Prolog-Repräsentation von GMR-Regeln hat die Form: `Regelkopf :- Regelkörper`. Der Kopf enthält ein Ziel-Berechnungs-Paar, der Regelkörper enthält ein Ziel-Berechnungs-Paar oder eine Konjunktion von mehreren solchen Paaren. Das Ziel ist der erste und die Berechnung der zweite Parameter einer GMR. Die Regeln repräsentieren das Wissen in einer sehr feinkörnigen Struktur. Jede Regel enthält lediglich einen Planungs- bzw. Berechnungsschritt. Durch die rekursive Aufruf-Struktur können mit diesen Mikro-Regeln auch komplexe Probleme bearbeitet werden. Mit Hilfe der GMR ist sowohl das Generieren von Lösungsvorschlägen als auch das Parsen von Hypothesen möglich. Durch das Backtracking von Prolog kann mit den GMR eine Vielzahl von Lösungen zu einer Aufgabe generiert werden.

Der durch das Y-Modell visualisierte Lösungsprozess wird in mileET in einer speziellen Form von Ziel-Mittel-Relation (GMR) abgebildet und mittels GMR-basierter Regeln in Prolog implementiert. Durch das Backtracking von Prolog und dank der Feinkörnigkeit der Regeln kann eine Vielzahl von Lösungen zu einer Aufgabe generiert werden. So können z.B. unterschiedliche Lösungswege über verschiedene elektrotechnische Gesetzmäßigkeiten vorgeschlagen, aber auch mathematisch korrekte Umformungen der Lösung erkannt werden. Da die GMR-Regeln die Lösungsmethoden abbilden und nicht die Lösungen, ist das System in der Lage, diese Regeln auf beliebige für die betrachtete Domäne relevante Schaltbilder und Parameter-Werte anzuwenden [Yakimchuk2002].

GMR-basiert sind nur die Regeln, die einen Beitrag zur Erklärungsgenerierung leisten.

Gesteuert wird der Prozess des Lösen und der Erklärungsgenerierung über einen Metainterpreter. Der Metainterpreter ist auch für das Zusammenstellen von Erklärungen (Explanations) zuständig:

```
metaInt (gmr (Goal,Mean) , Explanations, SI, SO) :-
    ...
    gmr (Goal,Mean,Explanation) ,
    ...

gmr (Goal,Mean,['#0430']) :- % '#0430' - Referenznummer der Erklärung für diese GMR
    gmr (SubGoal1,Mean1) ,
    ... ,
    gmr (SubGoalM,MeanM) .
```

## 5.2 mileET: prototypische Realisierung einer Problemlöseumgebung für GET

### 5.2.1 Benutzeraktivitäten in mileET

Entsprechend den didaktischen Forderungen an eLearning-Systeme [Tulodziecki2004] wird den Studierenden in mileET eine aktive und kooperative Auseinandersetzung mit Aufgaben verschiedener Komplexitätsniveaus aus den Themengebieten Kirchhoff'sche Gleichungen, Zweipoltheorie, Knotenspannungsanalyse, und Maschenstromanalyse ermöglicht. mileET stellt den Studierenden zur Bearbeitung dieser Aufgaben, die sowohl Schaltbilder als auch Formeln beinhalten, eine Benutzungsoberfläche<sup>53</sup> mit einem Schaltbild- und einem Formel-Editor zur Verfügung. Um solche wichtigen Anforderungen wie z.B. die Analyse partieller Lösungsentwürfe zu erfüllen, enthält mileET einen Lösungsentwurfeditor, ein elektronisches Arbeitsblatt [Yakimchuk2004].

Die Benutzungsschnittstelle besteht dementsprechend aus mehreren Bereichen (Abbildung 20), die in den folgenden Abschnitten näher betrachtet werden.

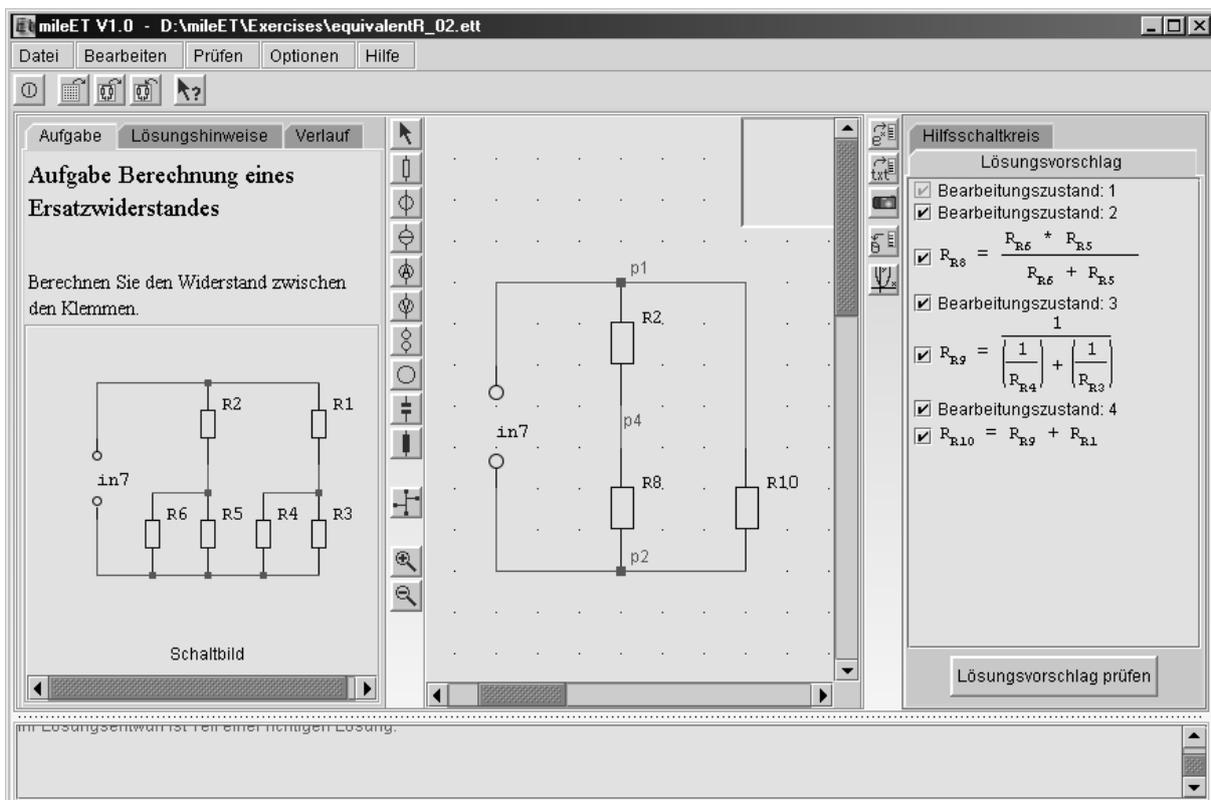


Abbildung 20. mileET-Lernumgebung

<sup>53</sup> implementiert von J.-P. Osterloh und L. Weber unter Betreuung von H. Garbe und V. Yakimchuk

Die mileET-Umgebung stellt alle für die Bearbeitung von Aufgaben notwendigen Funktionen zur Verfügung: Laden, Speichern und Drucken von Aufgaben und Bearbeitungszuständen; Erzeugen und Editieren von Schaltbildern und Parametern der Bauelemente; Erzeugen und Editieren von Formeln; Erzeugen, Editieren und Speichern von Lösungsentwürfen; Einstellungen der mileET Umgebung, Aufruf von Benutzungshilfen. Es ist trotzdem eine Optimierung und Erweiterung der Benutzungsschnittstelle notwendig (Abschnitt 5.3, S. 106).

### **Erzeugen und Editieren eigener Lösungsentwürfe**

Links im mileET-Fenster befindet sich der Aufgabenstellungs-Bereich (Abbildung 20, links). Hier wird die Aufgabenstellung mit dem zugehörigen Schaltbild dargestellt. Werden in einer Aufgabe auch Formeln vorgegeben (z.B. symbolische Werte einzelner Parameter), so erscheinen Sie direkt im Lösungsentwurf-Bereich (Abbildung 20, rechts). In dem Abschnitt 2.5.4 wird die Handhabung von Aufgabendaten in dem wissensbasierten Assistenten näher betrachtet.

Im mittleren Teil des mileET-Fensters befindet sich der mileET-Schaltungseditor. Die Implementierung des Editors geschah unter Berücksichtigung von Windows-Standard-Benutzungselementen und in Anlehnung an die analysierten Schaltungseditoren. Dort kann eine Schaltung mit verschiedenen Tools bearbeitet werden. Folgende Bauteile werden angeboten: Widerstände, Strom- und Spannungsquellen, Strom- und Spannungsmessgeräte<sup>54</sup>. Es stehen für Ingenieurwerkzeuge typische Schaltbildmanipulationsfunktionen zur Verfügung. Bauteile können platziert, gedreht, verschoben und gelöscht werden, die Parameter von Bauteilen können editiert werden. Die Eigenschaften von Bauelementen können je nach Auswahl durch den Benutzer in der Schaltung angezeigt werden. Verbindungen zwischen den Bauelementen können gezogen und gelöscht werden. Verbindungspunkte werden automatisch gezeichnet. Um die Anwendung der Kirchhoff'schen Sätze zu unterstützen, können Strom- und Spannungsrichtungen angezeigt werden (vgl. Kirchhoff im Abschnitt 3.1). Der mileET-Schaltungseditor ermöglicht es, innerhalb der festgelegten Aufgabenklassen beliebige Schaltungen zu zeichnen und diese während des Lösens den elektrotechnischen Lösungsmethoden entsprechend zu transformieren.

Darüber hinaus gibt es auch spezielle lernzielbezogene Funktionen, z.B. zur Transformation und Analyse von Schaltbildern. Die Abbildung 21 veranschaulicht den Prozess der Dreieck-Stern-Transformation.

---

<sup>54</sup> L, C – Elemente sind nur ansatzweise in der GUI vorbereitet.

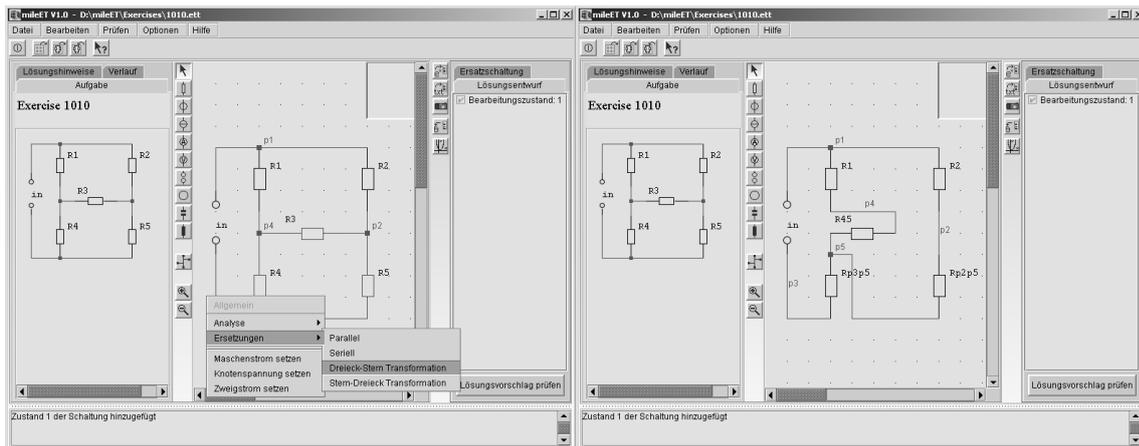


Abbildung 21. Unterstützung der Dreieck-Stern-Transformation

Zur Lösung der meisten Aufgaben gehören mathematische Ausdrücke. Sie können mithilfe eines maßgeschneiderten Formeleditors<sup>55</sup> vom Benutzer erstellt werden (Abbildung 22). Über die typischen Funktionen eines Taschenrechners hinaus wird der Zugriff auf die aufgabenspezifischen Parameter gegeben, was das Erstellen und Editieren von Formeln sowie die Änderung der Parameterwerte vereinfacht.

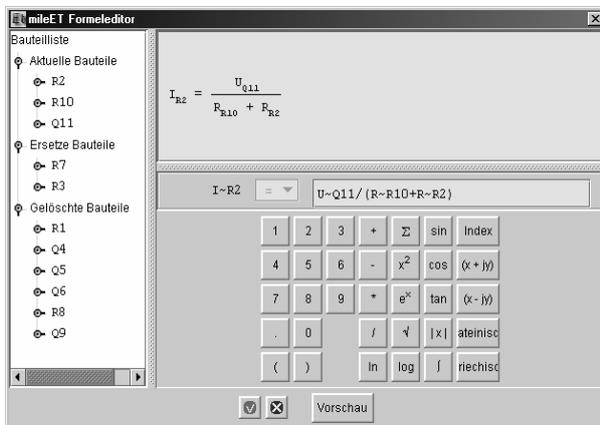


Abbildung 22. mileET - Formeleditor

Rechts im mileET-Fenster befindet sich der Lösungsentwurfseditor (Abbildung 20, rechts). Dieser Editor stellt einen besonderen Unterschied von mileET den anderen Lernprogrammen oder Schaltungseditoren gegenüber dar. Sowohl Schaltbilder, transformierte Schaltbilder, als auch Gleichungen können in den Lösungsentwurfseditor übernommen werden. Ein Lösungsentwurf besteht in mileET somit aus einer Folge von (transformierten) Schaltbildern und Formeln. Einzelne Elemente der Lösung können markiert, entfernt oder korrigiert werden.

<sup>55</sup> implementiert von J.-P. Osterloh unter Verwendung des JEP Projects (<http://www.singularsys.com/jep/>, letzter Zugriff 10.02.2005)

### Hypothesentesten und adaptive Vervollständigung der Lösungsentwürfe

Mithilfe des in mileET implementierten wissensbasierten Assistenten können die Benutzer ihre Lösungsentwürfe vom System analysieren lassen. mileET versucht, eine Lösung zu generieren, die den Lösungsentwurf des Benutzers beinhaltet. Gelingt dem System diese Einbettung, so bekommt der Benutzer die Rückmeldung, dass der Lösungsentwurf eine Lösung darstellt oder zu einer Lösung vervollständigt werden kann. Kann von mileET keine solche Lösung generiert werden, so wird der Benutzer gebeten, seine Hypothese einzuschränken. Er kann dann einzelne, von ihm als korrekt angenommene Einträge in seinem Lösungsentwurf markieren und diese vom System erneut prüfen lassen [Yakimchuk2004]. Verschiedene Arten des Hypothesentesten-Ansatzes werden in [Möbus1996, Willms1997] diskutiert.

„Durch (eventuell) wiederholtes Testen eingeschränkter Hypothesen wird der Lernende aufgefordert, aktiv nach eigenen Fehlern zu suchen. Hat der Lernende seine Hypothese auf einen korrekten Kern seines Lösungsentwurfs eingeschränkt, kann diesen jedoch nicht selbständig vervollständigen, so kann er von mileET Hilfe anfordern.

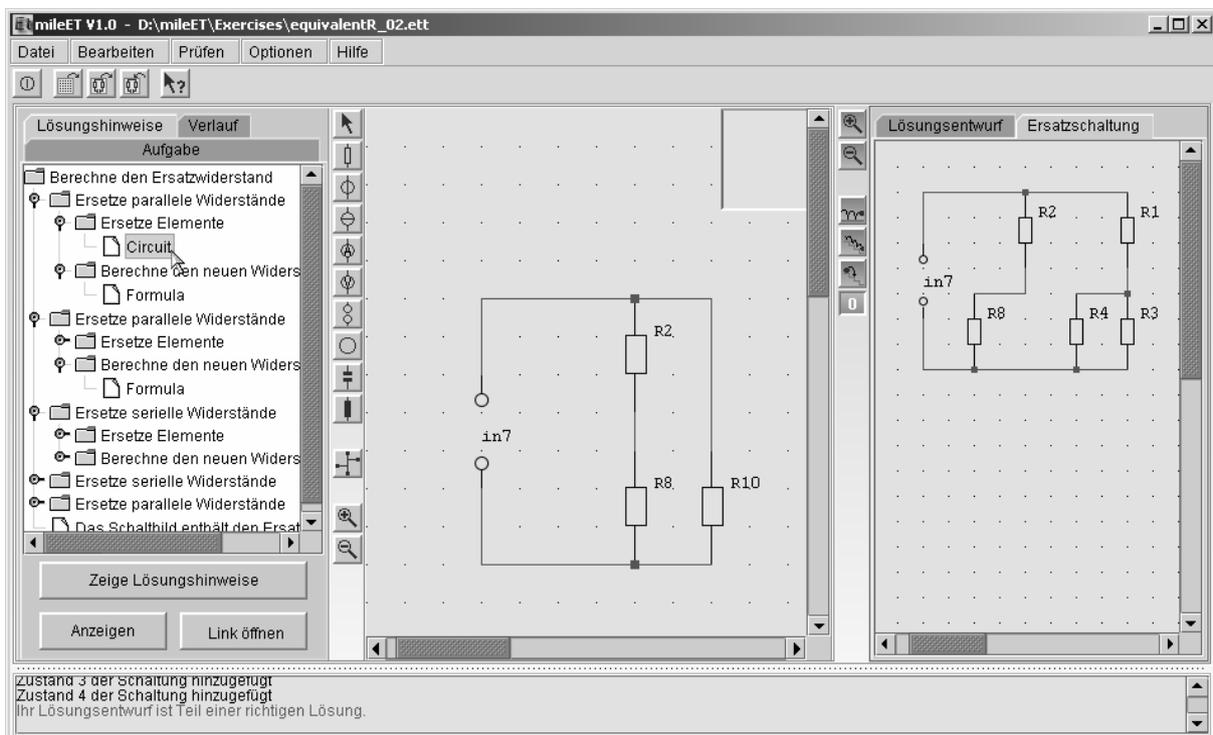


Abbildung 23. Vervollständigung einer Benutzerlösung in mileET

Ein von mileET erzeugter kommentierter Lösungsweg mit eingebetteter Lernerhypothese wird in dem Lösungshilfe-Bereich (Abbildung 23, links) in Form eines Baums angezeigt. Der Benutzer kann sukzessive detaillierte Hilfe-Informationen erhalten, indem er diesen Baum

weiter aufspannt. Die den Lösungsschritten entsprechenden Formeln und in bestimmten Fällen<sup>56</sup> auch die Schaltbilder kann sich der Benutzer anzeigen lassen. Auf diese Weise unterstützt mileET gemäß der ISP-DL Theorie das Selbsterklären und den Einsatz schwacher Heuristiken beim Problemlösen“ [Yakimchuk2004].

### **GETsoft Lernprogramme als kontextsensitive Hilfe**

Die adaptive, dynamisch generierte Hilfe besteht aus einem kommentierten Lösungstrace des Programms. Die Kommentare zu einzelnen Lösungsmethoden (oder eher zu einzelnen Schritten von Lösungsmethoden) sind in einer XML-Datei zusammengefasst und können von Aufgabenautoren editiert werden. Zu den einzelnen Schritten können Links angegeben werden, die auf die webbasierten Lernmodule verweisen, welche diese Schritte näher erläutern oder theoretische Hintergrund-Informationen geben. Die entsprechende Seite wird in einem Browser angezeigt. So können diese externen Module als Hilfesysteme verwendet werden.

### **Dozentenmodus und Aufgabenassistent**

Der Dozentenmodus<sup>57</sup> wurde zum Erstellen neuer bzw. zum Editieren vorhandener Aufgaben konzipiert. Das Erscheinungsbild ist dem Studentenmodus ähnlich.

Der Dozentenmodus ermöglicht zurzeit keine Analysen, verfügt aber über einen Prototyp des Aufgaben-Assistenten (Abbildung 24). Der Autor erstellt eine Aufgabe, indem er z.B. eine Schaltung zusammenstellt, Formeln vorgibt und beim Speichern die eigentliche Aufgabenstellung eingibt. Das Formulieren einer Aufgabenstellung bedeutet in mileET eine verbale Beschreibung der Aufgabe (für die menschlichen Benutzer) und die Auswahl eines Aufgabenziels (für die wissensbasierte Komponente). Bei einigen Aufgabenzielen sind zusätzliche Parameter auszuwählen, wie z.B. Anschlussklemmen.

Die korrekte Auswahl des Aufgabenziels ist die Voraussetzung für die vernünftige adaptive Unterstützung der Lernenden. MileET übernimmt keine Kontrolle über die Autoren, die dafür zu sorgen haben, die Aufgabe sinngemäß korrekt zu erstellen. Eine Überprüfung der Lösbarkeit wird in dem Studentenmodus zwar durchgeführt, es geht dabei aber um die Analyse des Zusammenpassens Schaltbild-Aufgabenziel (mehr dazu im Abschnitt 5.2.3.).

Zusätzlich zu dem Text und der Aufgabenzielauswahl kann aus der vorgegebenen Schaltung ein Bild erstellt werden, das im Studentenmodus im Aufgaben-Panel erscheint.

In dem Beispiel (Abbildung 24) wird die Berechnung eines Ersatzwiderstandes zwischen den Anschlussklemmen „in“ als Aufgabenstellung formuliert.

---

<sup>56</sup> Diese Art der Lernerunterstützung (graphische Vorführung einzelner Lösungsschritte) wurde prototypisch für einige Transformationen implementiert und bedarf der Weiterentwicklung.

<sup>57</sup> Eine kurze Bedienungsanleitung befindet sich im Anhang A7.



Abbildung 24. mileET-Dozentenmodus und Aufgabenassistent

### Benutzerfehler und Grenzen der Benutzerfreiheit

Es wurde versucht, eine Balance zwischen der Benutzerfreiheit und dem Aufwand, freie Entwürfe zu analysieren, zu finden. Verschiedene Arten von Fehlern werden im Programm verarbeitet: einige durch die wissensbasierte Komponente, andere direkt in der Benutzerschnittstelle. Beim Zeichnen von Schaltbildern wird darauf geachtet, dass gegen die Benennungsvereinbarungen der elektrotechnischen Zeichen nicht verstoßen wird: z.B. dürfen die Namen für Widerstände nur mit  $R$  anfangen, für Quellen mit  $I$  und  $U$ , analog sind für die Parameter der Elemente die Anfangsbuchstaben  $R$ ,  $I$ ,  $U$ ,  $P$  erforderlich. Ausgenommen diese Bedingung können neue Elemente beliebig umbenannt werden. Die in der Aufgabenstellung vorgegebenen Namen der Elemente dürfen nicht geändert werden.

Ebenfalls werden „Schaltungsfehler“, wie ein kurzgeschlossenes Element oder „hängende“ Leitungen, zurzeit im Schaltungsesitor abgefangen und verboten. Das kann sich für einige Aufgaben, z.B. mit dem Ziel das Schaltbild zu erstellen, als Nachteil erweisen.

Beim Erstellen der Lösungsentwürfe wird darauf geachtet, dass alle in einer Formel vorkommenden Operanden durch die Elemente einer der Schaltungen in dem Lösungsentwurf oder deren Parameter eindeutig definiert werden. Formeln aus dem Lösungsentwurf, die diese Bedingung nicht erfüllen, werden markiert und von der wissensbasierten Komponente nicht analysiert. Die Benutzer von mileET können textuelle Kommentare zu ihrer Lösung hinzufügen. Diese Kommentare werden allerdings nicht in der wissensbasierten Komponente analysiert. Die Benutzer können ebenfalls verschiedene Aufgaben erstellen. Kann mileET eine Aufgabe nicht lösen, kann die Umgebung als Aufgabenlösungseditor ohne intelligente Unterstützung benutzt werden.

## 5.2.2 Systemarchitektur, Datenstrukturen und Wissensrepräsentation

### Systemarchitektur

Um eine spätere Online-Benutzung des Systems zu ermöglichen, wurde die Benutzungsoberfläche in Java realisiert (Abbildung 25). Zur Speicherung der Aufgaben und des Bearbeitungsstandes wurden XML-Formate entworfen. Für die Implementierung der Diagnosekomponente wurde Prolog<sup>58</sup> als Programmiersprache gewählt. Prolog liefert einen Inferenzmechanismus der mit Prädikaten arbeitet, die sowohl zum Generieren von Ergebnissen als auch zum Parsen von Eingaben genutzt werden können. Prolog-Prädikate, die die Wissensbasis bilden, sind dynamisch erweiterbar und veränderbar. Prolog hat sich auch bei der Entwicklung von anderen IPSEs [Möbus1996, I@LLS] bewährt. Der LPA Intelligence Server [lpa] bietet eine Schnittstelle für die Kommunikation zwischen Java und Prolog. Die Unterteilung des Systems in die Blöcke Aufgaben-GUI-Wissensbasis ermöglichte die unabhängige Entwicklung der einzelnen Komponenten.

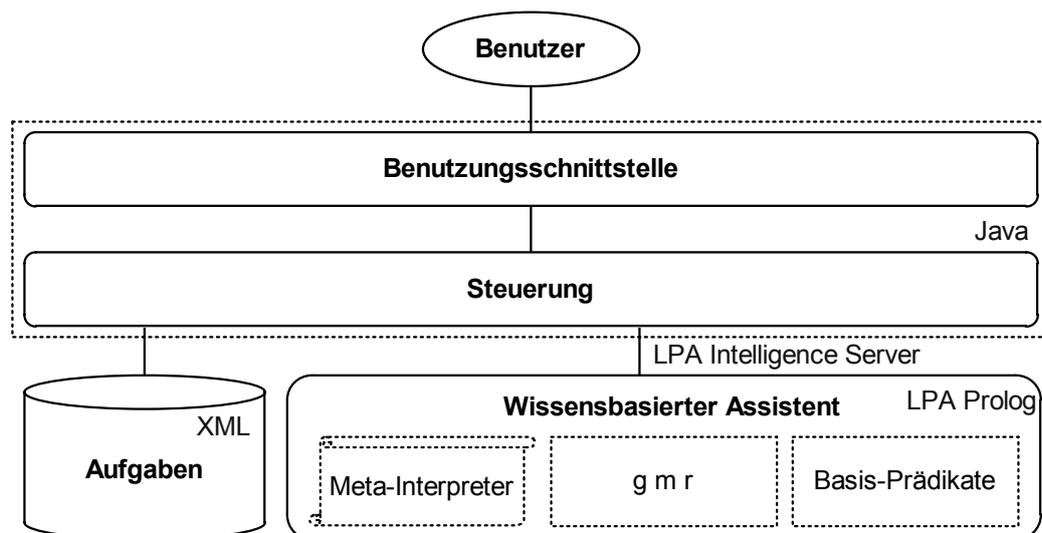


Abbildung 25. mileET-Systemarchitektur

Die Abbildung 26 zeigt die mileET-Architektur in Anlehnung an die Basisarchitektur eines Expertensystems [Hellmich1997]. Außer dem Fachwissen und den Inferenzregeln ist die wissensbasierte Komponente (auch wissensbasierter Assistent genannt) für die Verwaltung einiger Objekte der konkreten Aufgaben, wie z.B. Schaltbilder oder Elemente, zuständig.

Die Struktur und Organisation der Objektbasis, der Objektverwaltung für die Steuerungskomponente und die GUI-Komponente wird in der mileET Dokumentation<sup>59</sup> und der mileET Benutzungshilfe<sup>60</sup> beschrieben.

<sup>58</sup> <http://www.lpa.co.uk/> (letzter Zugriff 10.02.2005)

<sup>59</sup> erstellt von L. Weber und J.-P. Osterloh

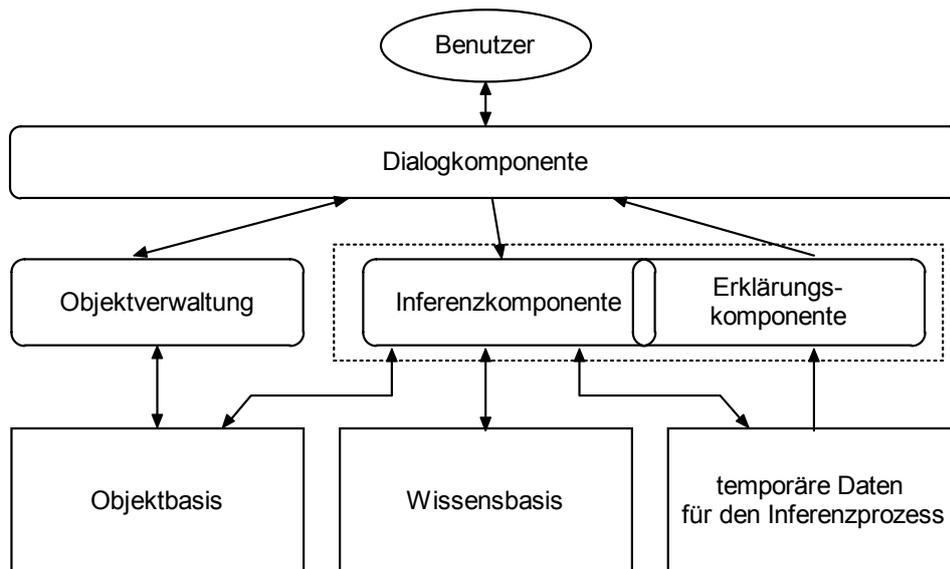


Abbildung 26. mileET als spezielle Art des Expertensystems

Im Weiteren werden die Wissensbasis, die dynamische Wissensbasis und einige Aspekte der Inferenz- und Erklärungskomponenten dargestellt.

### Datenstrukturen

Eine **GET-Aufgabe** wird durch eine verbale Beschreibung der Aufgabenstellung zusammen mit einem Schaltbild, einigen Formeln und Parameterwerten formuliert (siehe Abschnitt 5.1.3). In der Wissensbasis wird jede Aufgabe mithilfe zweier Prädikate formuliert: `task/2` und `taskInformation/2`:

```

task(TGoal, [TCircuit, TAdditions, TMath]),
taskInformation(TGoal, TInfo),

```

wobei  $\tau$  für engl. task steht.

Das Prädikat `task` beinhaltet alle in der Aufgabenstellung vorgegebenen Daten: Aufgabenziel (`TGoal`), Schaltbild (`TCircuit`) mit seinen Hilfsparametern (`TAdditions`), Formeln (`TMath`). Als Hilfsparameter werden z.B. Zweig- und Maschenströme, Knotenspannungen betrachtet. Das Prädikat `taskInformation` bestimmt die Art, wie (`TInfo`) die Lösungsentwürfe der Aufgaben zu verschiedenen Aufgabenzielen (`TGoal`) beurteilt werden: ob eine letzte Gleichung im Lösungsentwurf oder das gesamte Gleichungssystem vom Benutzer eingegeben werden muss. `TGoal` hat die Form `Aufgabenziel(Parameter des Aufgabenziels)` (siehe Überblick im Anhang A4). Das Schaltbild, Hilfsparameter und Formeln werden in Listen gespeichert.

Das Prädikat `taskInformation/2` ist in der Wissensbasis vordefiniert, das Prädikat `task/2` ist dynamisch und wird immer beim Starten einer neuen Aufgabe neu angelegt. Ein Überblick über die Aufgabenziele befindet sich im Abschnitt 5.2.3 und die Tabelle mit den

<sup>60</sup> mit Java-Help überarbeitet und ins englische übersetzt von T. Knostmann

Formulierungen der Aufgabenziele in der Notation des Aufgabeneditors (Werte für `TGoal` und deren Bedeutung) befindet sich im Anhang A4.

Beim Starten einer Aufgabe werden alle für die Bearbeitung der Aufgabe notwendigen Fakten angelegt und Listen zusammengestellt.

Das Prädikat

```
task(TGoal, [TCircuit, TAdditions, TMath])
```

dient auch zur Analyse der Lösbarkeit der Aufgabe, denn bestimmte Bedingungen müssen erfüllt werden, damit mileET eine Aufgabe lösen kann. Z.B. kann ein Ersatzwiderstand oder ein äquivalenter Zweipol nur dann ermittelt werden, wenn die Anschlussklemmen in der `TGoal` angegeben sind. Ebenfalls werden alle notwendigen Daten-Analysen (z.B. angelegte Zweigströme) und Daten-Vorbereitungen unter diesem Prädikat durchgeführt. Diese Analysen und die Ergebnisse der Datenverarbeitung werden in einem `initialState`-Fakt für den späteren Gebrauch abgelegt:

```
initialState([circuit([[TCircuit, TAdditions]]),
              parameter(ParameterList),
              math(Math),
              analyse(Analysis),
              statistic([numberOfElements(TNrOfElements),
                        numberOfPoints(TNrOfPoints)])
              ])).
```

Ein Schaltbild (`TCircuit`) wird in einer listenbasierten Form abgebildet:

```
[[Element1, ..., ElementM], Additions].
```

`Element1 – ElementM` sind die einzelnen Bauteile eines Schaltbildes, unter `Additions` werden zusätzliche Schaltbildparameter, wie z.B. Maschenstrom, Knotenspannung oder Zweigstrom, zusammengefasst.

Jedes Element wird wiederum durch eine Liste repräsentiert:

```
[Type, Name, ConnectionPins, Properties].
```

`Type` steht für den Typ des Bauelementes. Anschlussklemmen (`input`), Widerstände (`resistor`), Strom- und Spannungsquellen (`powerSourceAmp`, `powerSourceVolt`) können in den von der aktuellen mileET Version unterstützten Aufgaben verarbeitet werden. Jedes Element in dem Schaltbild hat einen eindeutigen Namen `Name` und wird zwischen den Klemmen `ConnectionPins` eingeschaltet. An dieser Stelle tragen die Pins `[Pin1, Pin2]` nur die Information über die Anschlussklemmen, die Information zu den Strom- und Spannungsrichtungen wird unter `Properties` gespeichert. Die Eigenschaften (`Properties`) eines Bauelementes beinhalten:

```
[RName, [[IPin1, IPin2], IName], [[UPin1, Upin2], UName], PName]
```

wobei

$RName$  - die Widerstandsvariable des Bauelementes,  
 $[[IPin1, IPin2], IName]$  - Stromrichtung und Stromvariable,  
 $[[UPin1, UPin2], UName]$  - Spannungsrichtung und Spannungsvariable,  
 $PName$  - die Leistungsvariable des Bauelementes sind.

Die Anschlussklemmen  $[Pin1, Pin2]$  werden mit den entsprechenden Verbindungsbezeichnungen belegt. Die Anschlussklemmen zur Strom-, Spannungsrichtung werden nur dann mit einem Wert belegt, wenn die entsprechenden Parameter (Strom durch das Element bzw. Spannung an dem Element) vom Autor oder vom Lernenden eingesetzt wurden. Die Richtung wird von  $Pin1$  zu  $Pin2$  definiert.

Diese Form der Präsentation von Schaltbildern lässt eine Anpassung des Systems auf neue zweipolige Elemente, z.B. gesteuerte Energiequellen, zu. Eine Erweiterung bzw. Änderung der Liste von Eigenschaften ist u. U. vorzunehmen. Eine Aufnahme mehrpoliger Elemente (Vierpole) ist z.B. mit Darstellung durch Ersatzzweipole möglich.

Ein Parameter wird durch eine Liste mit Parametervariablen ( $ParameterName$ ), symbolischer Wert ( $SymValue$ ), numerischer Wert ( $NumValue$ ), Maßeinheit ( $Unit$ ) und Status in der Aufgabenstellung ( $StatusSym, StatusNum$ ) repräsentiert:

$[ParameterName, SymValue, NumValue, Unit, StatusSym, StatusNum]$ .

Der Status in der Aufgabenstellung gibt Auskunft darüber, ob der entsprechende Wert gegeben, bekannt oder unbekannt ist.

Von der Benutzungsschnittstelle werden alle Parameter über Formeln übergeben. Über das Prädikat

$etNewTask(TGoal, [[TCircuit, TAdditions, TNrOfElements, TNrOfPoints]], TMath)$

werden alle durch  $TCircuit$  und  $TMath$  bestimmten Parameter erzeugt und im  $initialState$  abgelegt (siehe auch den nächsten Abschnitt).

Die Verarbeitung der Formeln<sup>61</sup> ist auf die für die mileET-Aufgaben relevanten Formeln begrenzt. Jede Formel  $LeftSide = RightSide$  wird in der Form:

$[LeftSide, RelOperator, RightSide, NullForm]$

abgebildet, wobei die  $NullForm$ :  $RightSide - LeftSide$  entspricht.

Diese Abbildung findet ebenfalls beim Zusammenstellen des  $initialState$ -Faktes statt.

Ein Lösungsentwurf aus dem Lösungsentwurfeditor in mileET besteht aus Schaltbildern und Formeln. Die Listen mit einzelnen Schaltbildern werden zu einer verschachtelten Liste zusammengefasst und Formel-Listen zu einer verschachtelten Formel-Liste. Diese

---

<sup>61</sup> Konzipiert und implementiert von H. Garbe

Information zusammen mit der Aufgabenstellung bildet den Ausgangspunkt für den wissensbasierten Assistenten in mileET. Von mileET wird ein Lösungsentwurf mithilfe von GMRs und dem Metainterpreter generiert.

Ein speziell für mileET implementierter Meta-Interpreter verwaltet die Daten während des Aufgabenlöseprozesses und steuert die Entscheidungen bezüglich des nächsten anzuwendenden Lösungsschrittes.

### **Wissensrepräsentation**

Die Wissensbasis besteht aus Prädikaten, GMR-basierten Regeln und Metainterpreter-Regeln. mileET-Prädikate bestehen aus fachspezifischen Prädikaten und Hilfsprädikaten. Die fachspezifischen Prädikate enthalten das notwendige Wissen über die elektrotechnischen Grundlagen: passives/aktives Element, Widerstand, ideale Strom-, Spannungsquelle, Amperemeter, Voltmeter, Zweig, Masche, Knoten, Schaltung, seriell geschaltete Elemente, parallel geschaltete Elemente, Stern- und Dreieck-Schaltungen, Zweipol, Anschlussklemmen. Die GMR-basierten Regeln enthalten das Wissen über die Lösungsmethoden: Parameterberechnung mithilfe des Ohm'schen Gesetzes, Erstellen von Kirchhoff'schen Gleichungen, Anwendung der Methoden der Knotenspannungsanalyse und der Maschenstromanalyse, einfache Strom- und Spannungsteiler.

Meta-Interpreter-Regeln steuern den Prozess des Aufgabenlöses, indem sie die Anwendbarkeit von GMRs überprüfen, und sind für das Zusammenstellen der Lösungen und Erklärungen verantwortlich.

Hilfsprädikate werden von fachspezifischen Prädikaten und GMR- und Meta-Interpreter-Regeln benutzt. Sie ermöglichen u. a. die Verwaltung von Schaltungselementen, Parametern, Formeln, Erklärungen und unterstützen die Analyse der Schaltungstopologie.

Detaillierte Information über einzelne Prädikate wird über die mileET-Dokumentation verfügbar sein.

### 5.2.3 Benutzeradaptive Aufgabenlösungen in mileET

Die folgende Abbildung gibt den Überblick über die zulässigen Aufgabenziele und deren Y-Zuordnung.

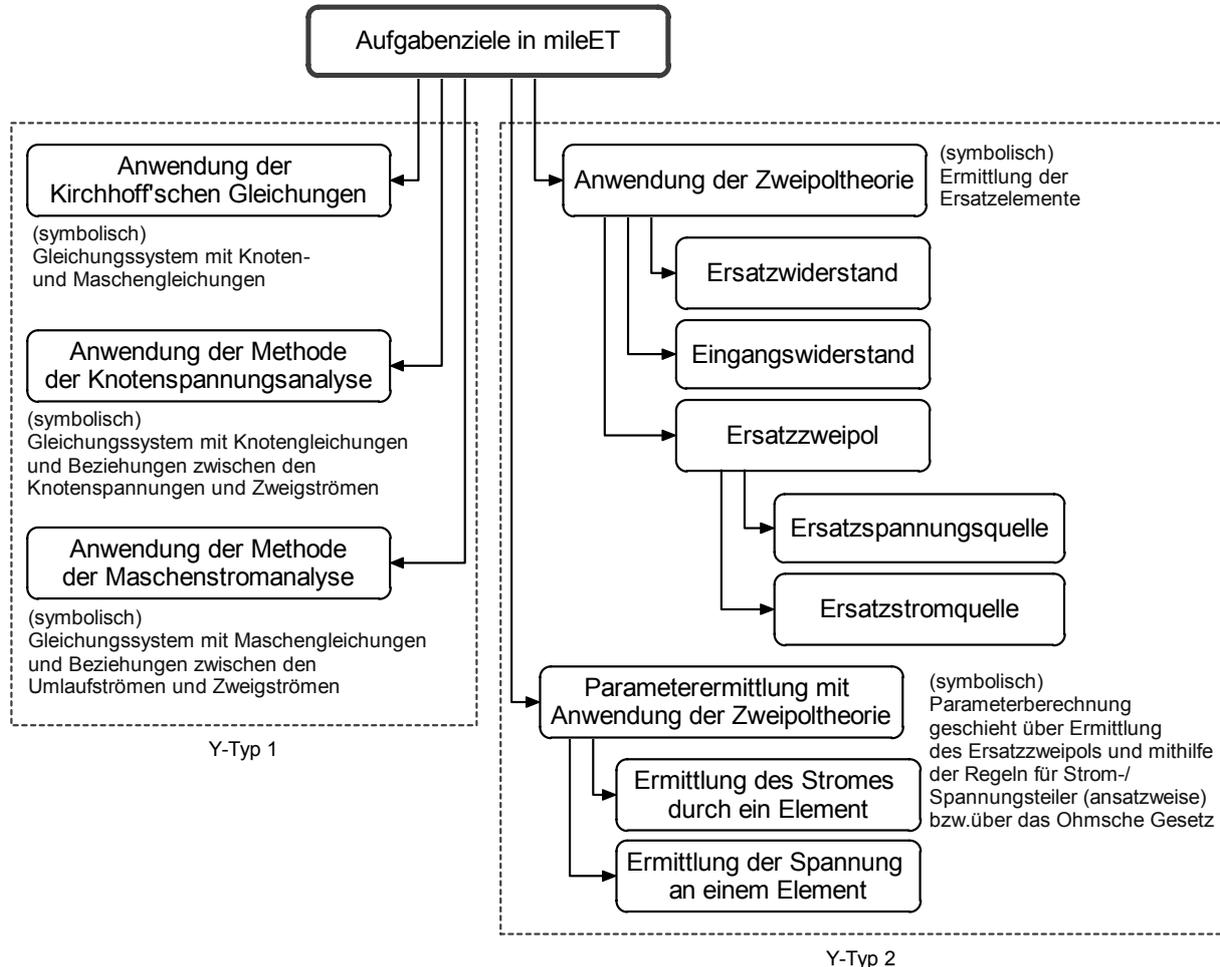
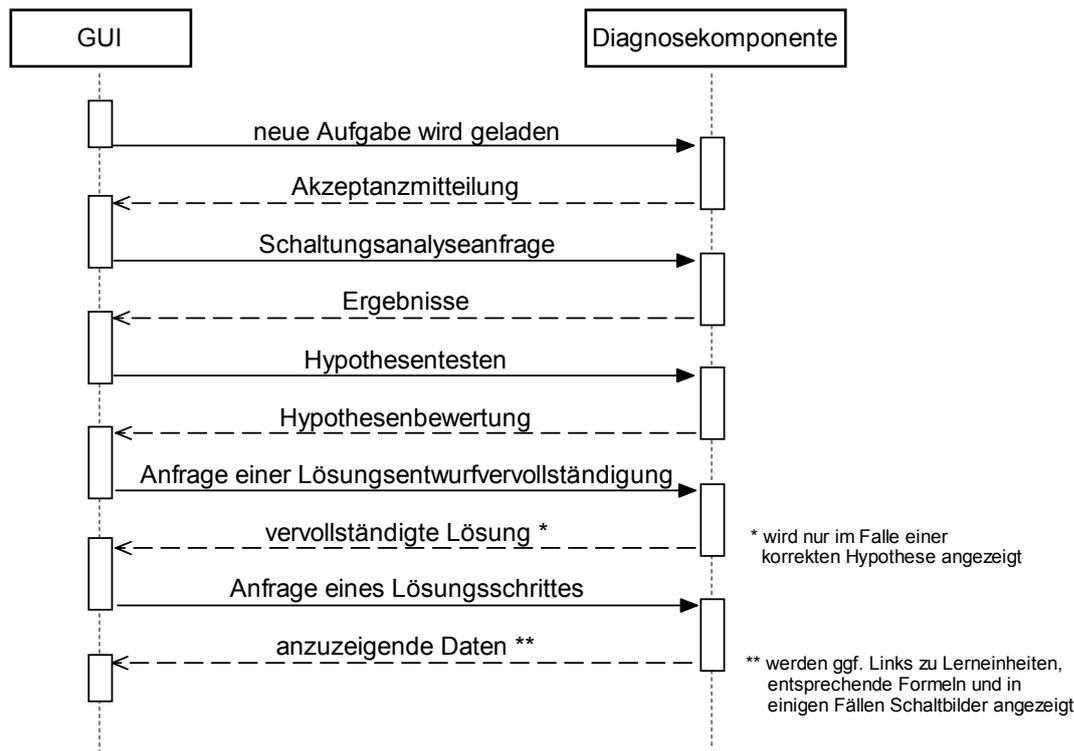


Abbildung 27. Unterstützte Aufgabenziele in mileET

Im Weiteren wird die mileET-Vorgehensweise beim Lösen und beim Erzeugen adaptiver Hinweise zu den Aufgaben der beiden Y-Typen vorgestellt. Die Verarbeitung von Aufgaben des Y-Typs 1 wird am Beispiel einer Schaltung mit allen drei Aufgabenzielen vorgestellt (Abschnitte 5.2.3.2, 5.2.3.3, 5.2.3.4). Die Verarbeitung der Aufgaben des Y-Typs 2 wird am Beispiel zweier Schaltungen vorgestellt: eine mit dem Ziel Ersatzwiderstand zu berechnen und eine mit dem Ziel den Strom durch ein Element zu berechnen (Abschnitt 5.2.3.1).

Die nächste Abbildung zeigt den Verlauf einer möglichen Sitzung mit mileET. Diese Zusammenarbeit aus der Sicht des Benutzers wurde im vorherigen Abschnitt vorgestellt. In diesem Abschnitt werden die einzelnen Schritte dieser Zusammenarbeit aus der mileET-Sicht näher dargestellt.



**Abbildung 28. Möglicher Verlauf der Arbeit mit mileET**

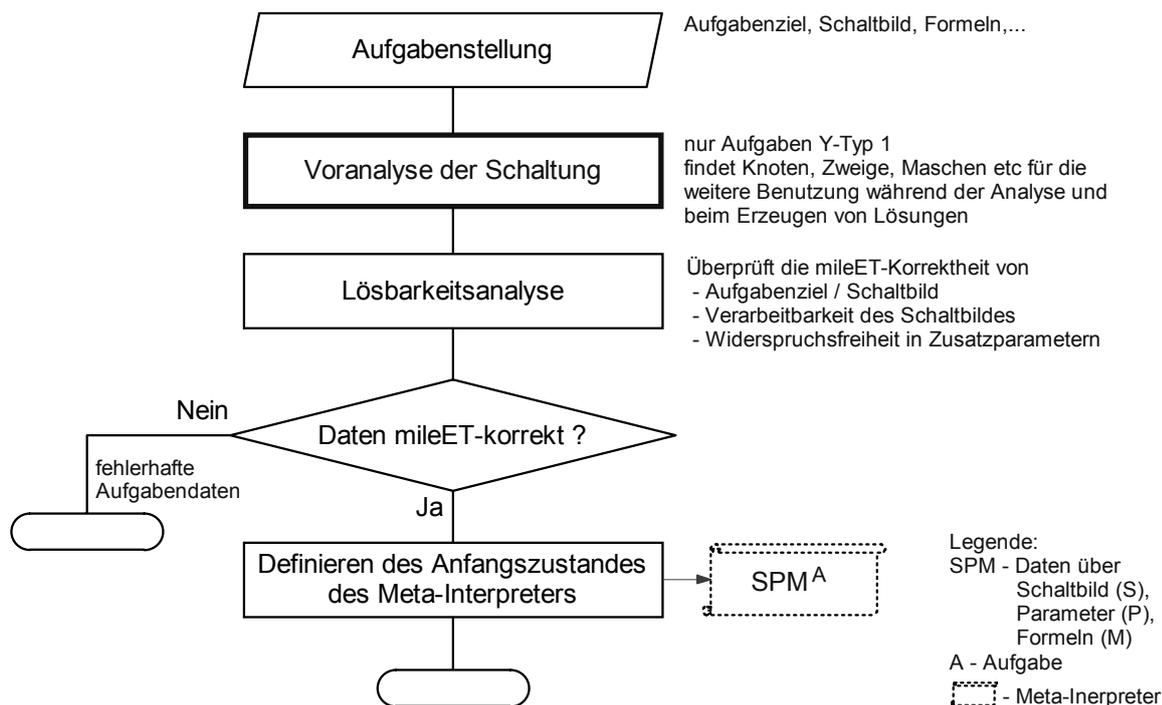
Die Arbeit beginnt mit dem Starten einer Aufgabe bzw. mit der Fortsetzung eines früher gespeicherten Bearbeitungsstandes. Da mileET keine Benutzerverwaltung durchführt, wird in beiden Fällen eine neue Aufgabe in der wissensbasierten Komponente erzeugt (Abbildung 29). Der Unterschied zwischen den Menüpunkten „Neue Aufgabe“/„Fortsetzung“ wird in der GUI bemerkbar, für die Wissensbasis stehen immer Aufgabendaten und Benutzereingaben zur Analyse. Deswegen wird in den weiteren Erklärungen von einer Aufgabe die Rede sein. Beim Zusammenstellen des Lösungsentwurfes wird die Wissensbasis immer dann angesprochen, wenn der Benutzer bestimmte Schaltungsmanipulationen durchführt (z.B. bei Schaltungstransformation). Das Lösen der Aufgabe seitens mileET findet aber in diesen Fällen nicht statt. Erst wenn der Benutzer eine Überprüfung per Knopfdruck bzw. über Menü anfordert, startet mileET den Löseprozess. Als Ergebnis kommt eine Beurteilung der Hypothese (richtige Lösung bzw. Teil einer richtigen Lösung) oder Vorschlag die Hypothese einzuschränken. Im Falle einer korrekten Hypothese kann eine Vervollständigung der Lösung erfragt werden und die kommentierten Schritte können einzeln angezeigt werden.

In mileET werden zwei Y-Typen von Aufgaben verarbeitet (siehe Abschnitt 5.2.1). Dementsprechend unterscheiden sich einige Schritte beim Umgang mit den Aufgaben-Daten für die beiden Gruppen.

Beim Laden einer Aufgabe wird bei der Aufgabe des Y-Typs 1 eine Analyse der Schaltung durchgeführt. Die Ergebnisse dieser Analyse werden später für das Hypothesentesten und die

Lösungsvervollständigung gebraucht. Für die Aufgaben dieses Typs ist solche Analyse sinnvoll, da die Zusammenschaltung von Elementen während der Lösung unverändert bleibt. Die Reaktionszeit wird dank dieser Analyse gekürzt, denn die während der Anfangsanalyse gesammelte Information, z.B. über die Maschen und Zweige, steht zum Zeitpunkt des Lösen zur Verfügung.

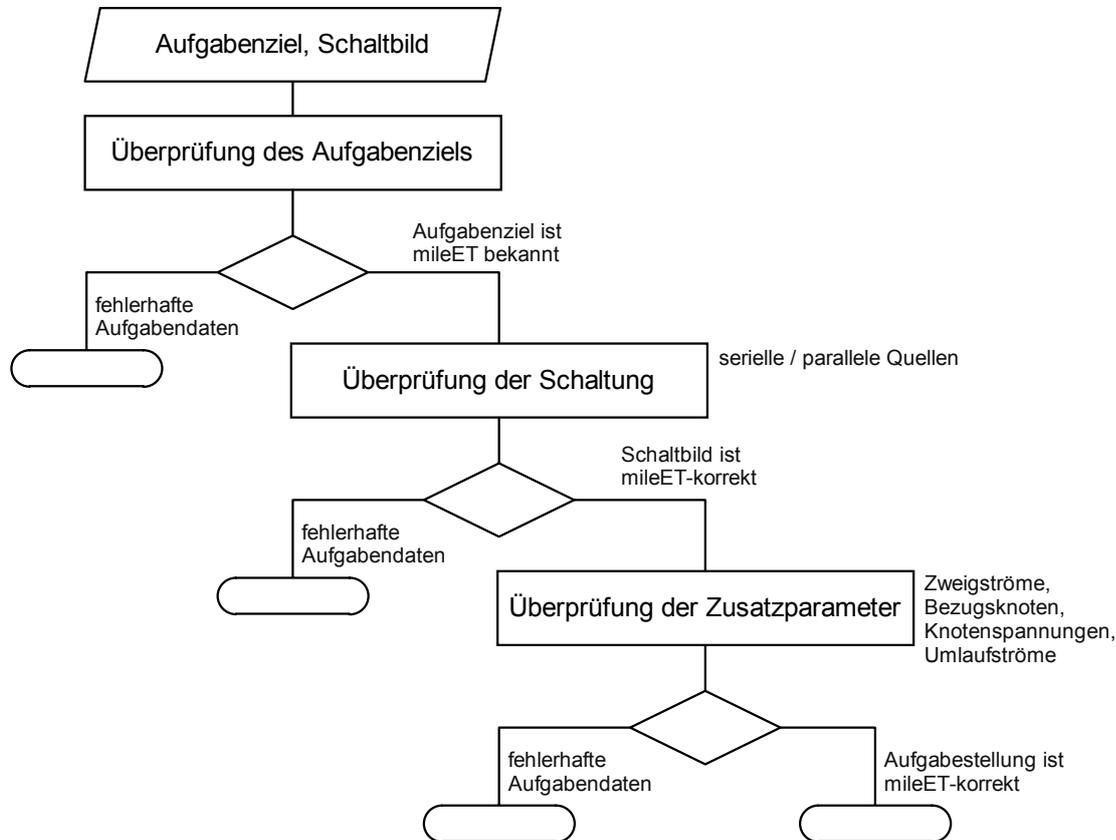
Für die Aufgaben des Y-Typs 2 würde diese Analyse nicht viel bringen, denn während der Lösung wird die Schaltung transformiert und die Zusammenschaltung von Elementen ändert sich.



**Abbildung 29. Erzeugen einer Aufgabe in der wissensbasierten Komponente**

Eine Lösbarkeitsanalyse (Abbildung 30) wird für die Aufgaben beider Gruppen durchgeführt. Es wird überprüft, ob das Aufgabenziel der mileET bekannt ist, ob mileET die Schaltung „versteh“ und ob die Zusatzparameter, wie z.B. Knotenspannungen oder Zweigströme widerspruchsfrei sind. Wurde die XML-Datei einer Aufgabe über den Aufgabenassistenten erstellt, so wird das Aufgabenziel bekannt sein. Da die Quelldateien editierbar sind, ist diese Überprüfung eher eine Sicherheitsmaßnahme für den Fall der Änderung der Aufgabendatei. Bei der Schaltungsüberprüfung handelt es sich um das Zusammenpassen des Aufgabenziels und der Schaltung sowie um die mileET-Korrektheit von Schaltungen. Das parallele Zusammenschalten zweier Spannungsquellen mit verschiedenen Spannungswerten oder serielle Zusammenschaltung zweier Stromquellen mit verschiedenen Stromwerten sind nicht zugelassen.

Außerdem dürfen die Schaltungen, z.B. für die Anwendung der Methode der Knotenspannungsanalyse, keine „input“-Elemente enthalten und die Berechnung einer Widerstandskombination kann nur für die Schaltung mit Widerständen und einem „input“-Element durchgeführt werden.



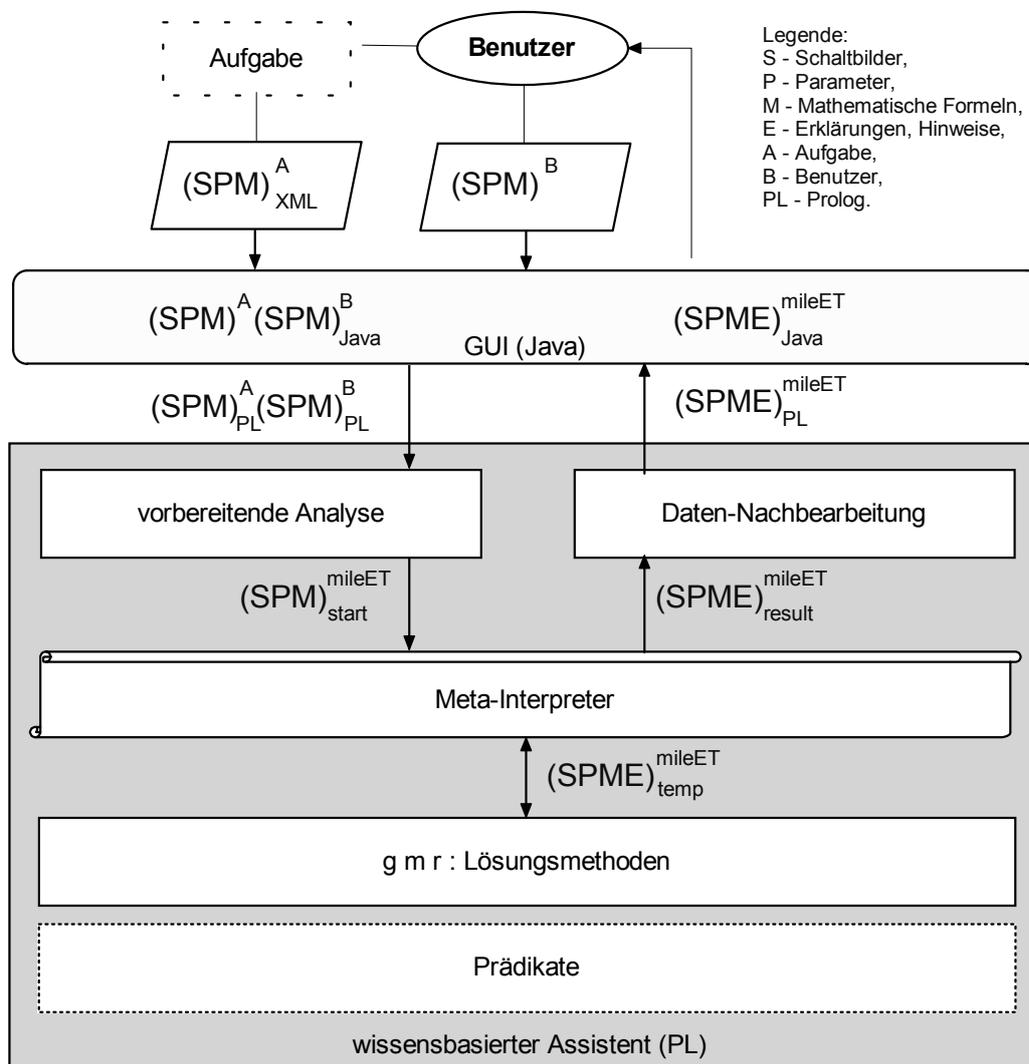
**Abbildung 30. Lösbarkeitsanalyse für die Aufgaben des Y-Typs 1**

Beim Überprüfen der Zusatzparameter wird auf die Widersprüche innerhalb der Angaben geachtet, z.B. darf nur ein Referenzknoten vorgegeben oder die Zweigströme sollen eindeutig definiert sein. Da auch die Lernenden als Aufgabenautoren im Dozentenmodus neue Aufgaben erstellen können, ist die Lösbarkeitsanalyse eine Ansatzlösung für einen intelligenten (wissensbasierten) Aufgabenassistenten.

Nachdem die Aufgabe der wissensbasierten Komponente mitgeteilt worden ist, kann eine Lösung dieser Aufgabe, eine Überprüfung bzw. eine Vervollständigung des Lösungsentwurfes angefordert werden. Folgende Abbildung illustriert die Arbeit von mileET beim Aufgabenlösen.

Die vom Benutzer ausgewählte Aufgabe ( $SPM^A$ ) zusammen mit seinem Lösungsentwurf ( $SPM^B$ ) zu dieser Aufgabe wird an den wissensbasierten Assistenten übergeben. Dafür werden die Aufgaben-Daten aus dem XML-Format ( $SPM^A_{XML}$ ) in GUI Java-Format ( $SPM^A_{Java}$ ) transformiert und danach in das Prolog-Format ( $SPM^A_{PL}$ ) der Wissensbasis.

Ebenfalls wird der Lösungsvorschlag des Benutzers in dem Prolog-Format ( $(SPM)_{PL}^B$ ) an die Diagnosekomponente überführt.



**Abbildung 31. Struktur des Aufgabenlöseprozesses**

Der mileET-Assistent wird dadurch aufgefordert, die Aufgabe zu lösen und den Lösungsentwurf des Benutzers zu beurteilen. Die Aufgaben- und Lösungsdaten werden dabei in den drei Mengen (Schaltbilder, Parameter, Mathematik -  $SPM^{mileET}$ ) verarbeitet (siehe Y-Modell im Abschnitt 5.1.3), die Anwendung von Konzepten wird über GMRs und den Meta-Interpreter realisiert. Die vorbereitende Analyse sorgt für die Effizienz des Programms, indem alle notwendigen Daten aufbereitet werden. Eine Reihe von speziellen und allgemeinen Prädikaten hilft in allen Stadien des Aufgabenlöseprozesses.

In mileET sind keine Lösungen vorprogrammiert. Eine Lösung wird aufgrund der Analyse des Lösungsentwurfes des Benutzers oder aufgrund der Reihenfolge von durchführbaren

GMRs erzeugt. Die Prozesse des Lösen und der Analyse der Benutzerlösung geschehen „parallel“. Nach jedem Lösungsschritt, wird im Meta-Interpreter der Bearbeitungsstand von mileET mit dem des Benutzers verglichen. Die während des Löseprozesses erzeugende Daten werden auf dem Bild als  $(SPME)^{mileET}_{temp}$  kennzeichnet. Die Entscheidung des mileET-Assistenten wird aufgrund dieses Vergleichs getroffen. Kann keine Lösung vom mileET-Assistenten erzeugt werden, die alle in der Benutzerhypothese vorhandenen Schritte beinhaltet, so wird eine Fehlermeldung ausgegeben. Können alle Schritte des Benutzers in eine mileET-Lösung eingebettet werden, so kommt eine Rückmeldung darüber, dass der Lösungsentwurf korrekt bzw. Teil einer korrekten Lösung ist. So entsteht die mileET-Lösung  $(SPME)^{mileET}_{result}$ .

Das Hypothesentesten und die adaptive Lösungsvervollständigung geschehen über zwei Prolog-Prädikate: `hypothesisTesting/3` und `completion/3`

Beide Prädikate benötigen Angaben zu Schaltungen (`UCircuitsIn`) und Formeln (`UMath`) aus dem Lösungsentwurf als Eingangsinformation, unterscheiden sich aber im dritten Parameter:

- `hypothesisTesting(UCircuitsIn, UMath, State)` liefert die Aussage (`State`) über die Richtigkeit des Lösungsentwurfes (Lösung ist korrekt, Lösungsentwurf ist Teil einer Lösung oder Lösungsentwurf ist fehlerhaft)
- `completion(UCircuitsIn, UMath, Explanations)` liefert eine kommentierte Vervollständigung (`Explanations`) des Lösungsentwurfs.

Verschiedene mileET-Lösungsmethoden werden im Weiteren dargestellt und diskutiert.

### 5.2.3.1 Anwendung der Zweipoltheorie

Zum Beginn des Löseprozesses liegen die Aufgaben-Daten ( $SPM^A$ ) im Meta-Interpreter vor (Abbildung 32). Die Benutzerdaten werden für die weitere Benutzung aufbereitet und ebenfalls im Meta-Interpreter aufgefasst ( $SPM^B$ ). Mithilfe von GMR-basierten Regeln wird die Aufgabe gelöst. Der Meta-Interpreter benutzt dabei die Lösungsentwurfsdaten, um einen Lösungsweg zusammen zu stellen. Die Daten-Nachbereitung sorgt für das Zusammenfassen der erzeugten Lösung für die Anzeige als Lösungsvervollständigung. Als Erklärungen werden Kommentare zu den einzelnen Schritten mit Formeln ausgegeben und die Durchführung einiger Transformationen angezeigt<sup>62</sup>. Die Erklärungen über den theoretischen Hintergrund der Lösungsschritte können über einen Link zu einer entsprechenden Internetseite erreicht

---

<sup>62</sup> graphische Unterstützung anderer Transformationen ist eine Aufgabe für die Weiterentwicklung von mileET, vgl. Kommentar zum „Hypothesentesten und adaptive Vervollständigung der Lösungsentwürfe“ aus dem Abschnitt 5.2.1

werden. Wenn keine Vervollständigung angefragt wurde, wird nur die Mitteilung über den Status des Lösungsentwurfes gegeben.

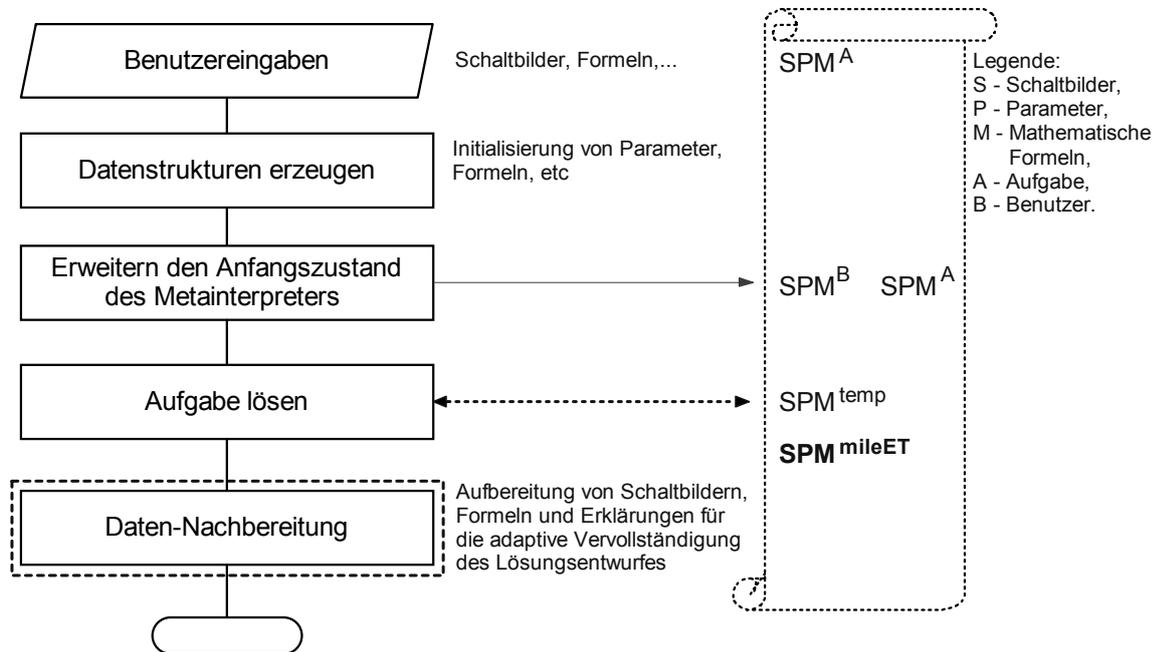


Abbildung 32. Lösen einer Aufgabe des Y-Typs 2

Um die Vorgehensweise bei der Berechnung des Ersatzwiderstandes einer Gruppe von Widerständen oder bei Berechnung der Eingangsimpedanz zu kommentieren, wird vom System eine Schaltungstransformation in kleinsten Schritten durchgeführt. Jeder Schritt führt zum Erreichen eines Ziels (GMR-Prinzip) und die Lösung geschieht somit über die Zielzerlegung.

Die oberen GMR-Regeln der Lösungsmethode in Prolog-Notation werden in der folgenden Tabelle gezeigt.

```

gmr (equivalentR (IN), _, ['#0200']) :-
    gmr (equivalentR1 (IN), _).

gmr (equivalentR1 (IN), _, ['#0201']) :-
    gmr (check (oneResistor), _).

gmr (equivalentR1 (IN), _, ['#0202']) :-
    gmr (ignoreElements (IN), _),
    gmr (equivalentR1 (IN), _).

gmr (equivalentR1 (IN), _, []) :-
    gmr (transform (IN), _),
    gmr (equivalentR1 (IN), _).

```

Tabelle 3. Berechnung des Ersatzwiderstandes in Prolog-Notation

Mithilfe dieser GMRs lassen sich verschiedene Lösungsbäume generieren, die in einer allgemeinen Form in den folgenden zwei Abbildungen dargestellt sind.

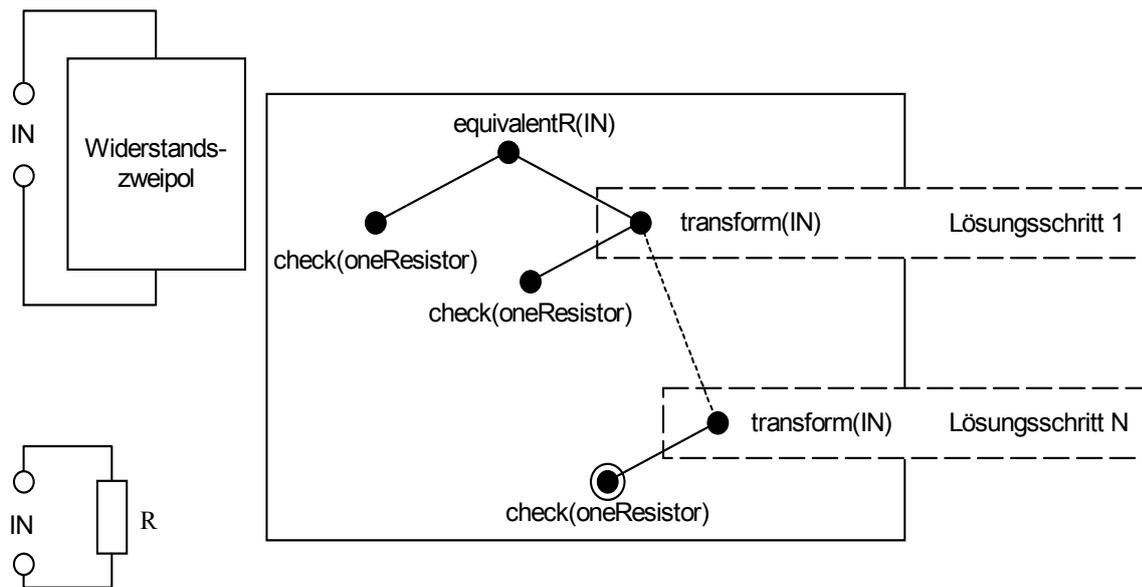


Abbildung 33. Lösungsbaum beim Berechnen eines Ersatzwiderstandes

Beim Ermitteln der Eingangsimpedanz werden zusätzliche Schritte benötigt, falls die Schaltung Strom- oder Spannungsquellen enthält. In dem Falle wird darauf geachtet, dass nur die notwendigen Elemente berücksichtigt werden: Spannungsquellen werden kurzgeschlossen, Stromquellen – unterbrochen, und die „freien“ Elemente entfernt.

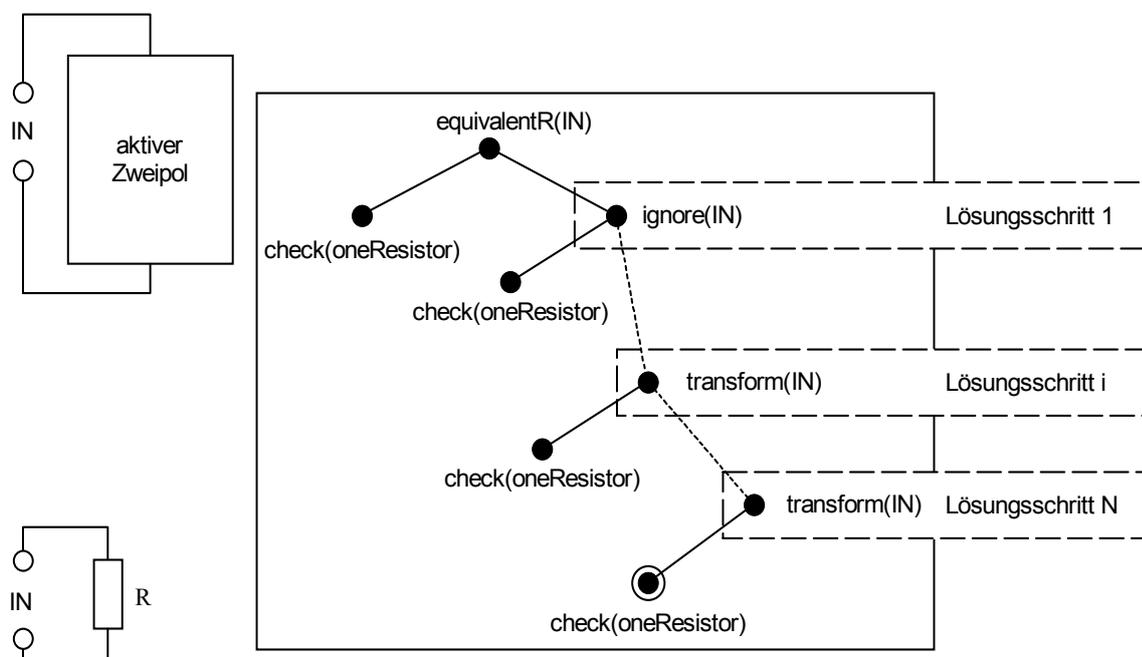


Abbildung 34. Lösungsbaum beim Ermitteln der Eingangsimpedanz

Für die resultierende Schaltung wird in beiden Fällen ein Ersatzwiderstand berechnet. Deswegen basiert die Methode der Eingangswiderstandsberechnung auf den Regeln der

Methode der Ersatzwiderstandsberechnung, die mit einigen zusätzlichen Regeln kombiniert werden.

Das nächste Diagramm bildet die allgemeine Logik der Lösungsmethode für die Anwendung der Zweipoltheorie ab, falls der Benutzer eine Lösungsvervollständigung vom System anfragt.

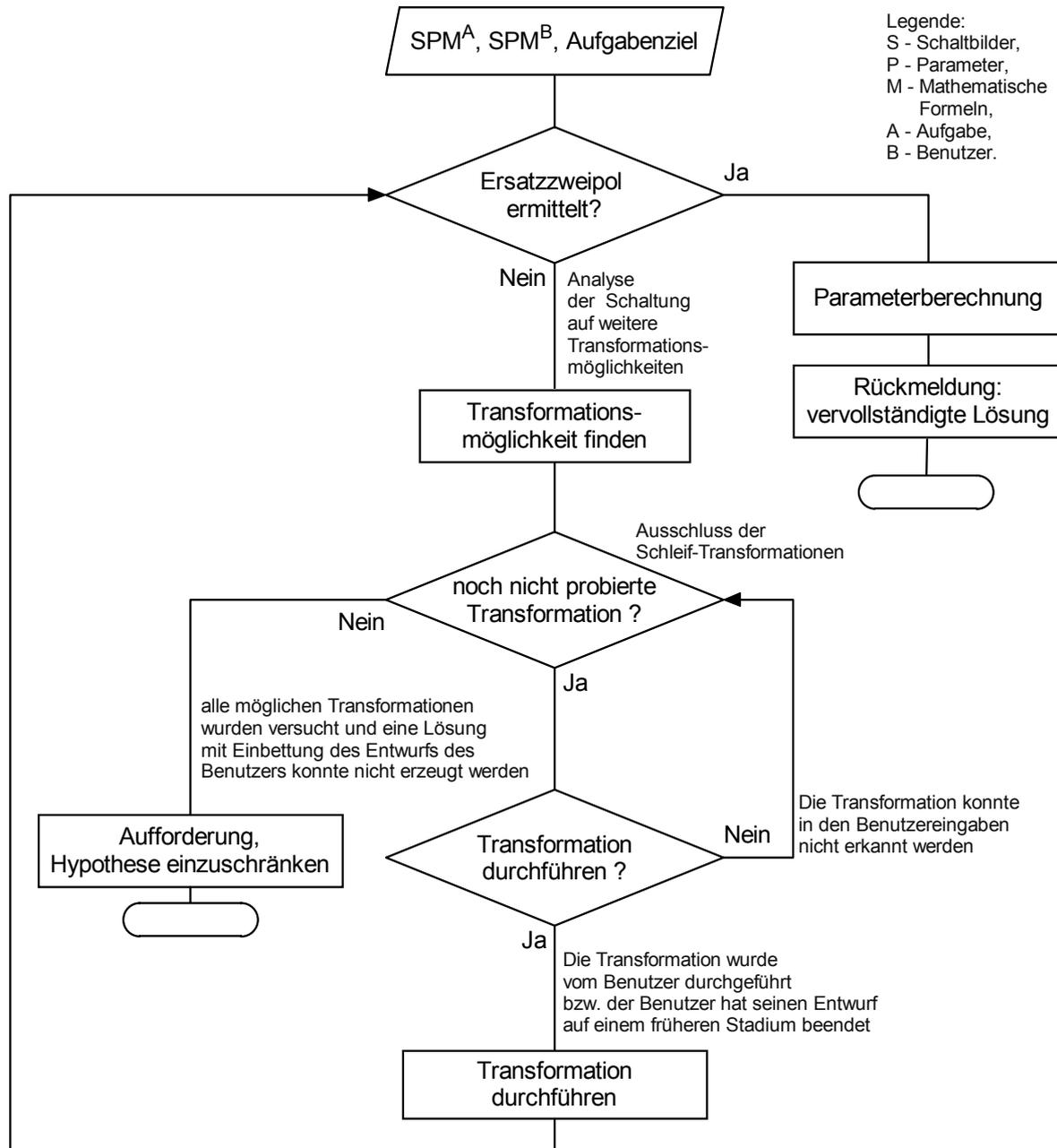


Abbildung 35. Lösungsmethode der Eingangsimpedanzberechnung aus der Sicht des Meta-Interpreters

Diese Logik wurde allerdings nicht als Prozedur programmiert, sondern wird mit GMRs abgebildet. Die Suche nach einer möglichen Transformation, die Entscheidung über ihre Durchführung und die eigentliche Durchführung geschieht über eine `transform-gmr` und wird vom Meta-Interpreter verwaltet. Die Transformationsmöglichkeiten (Abbildung 36) werden

über *backtraking* angeboten. Die Entscheidung, welche Transformation durchzuführen ist, wird aufgrund der Analyse von  $(SPM)_i$  und  $S'_i$  getroffen (Abbildung 37).

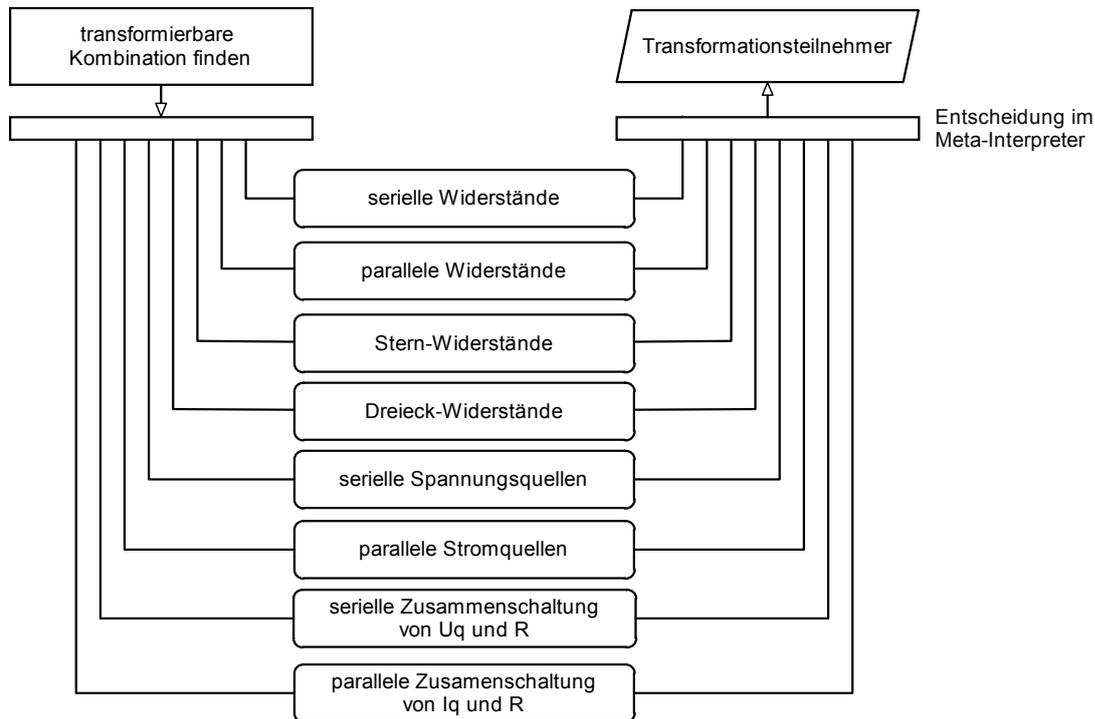


Abbildung 36. Transformationsmöglichkeiten in mileET

Die Dreieck $\leftrightarrow$ Stern-Transformationen und die  $I_q \leftrightarrow U_q$ -Transformationen gehören zu den so genannten Schleif-Transformationen. In diesen Fällen können die gleichen Elemente hin und zurück transformiert werden. Der Ausschluss solcher Wiederholungen wird über eine Vergleichsanalyse mit allen während der Lösungsgenerierung versuchten Schaltbildzuständen ermöglicht<sup>63</sup>.

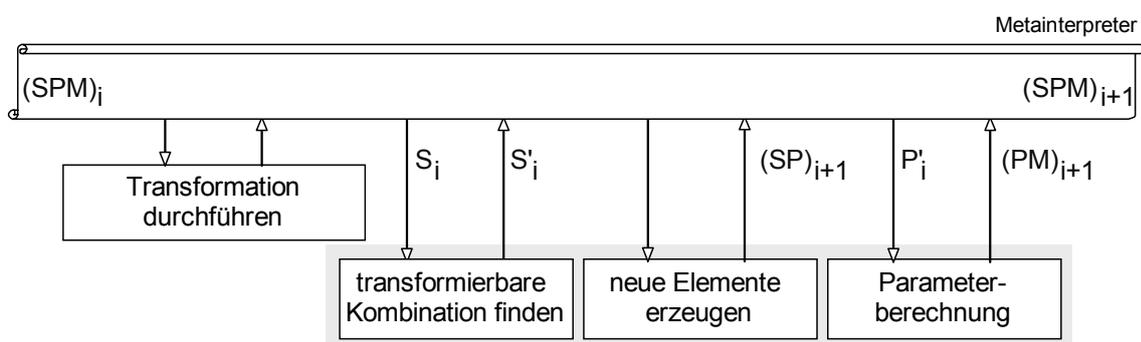


Abbildung 37. Zielzerlegung und Lösungsaufbau in SPM-Mengen für Aufgaben Y-Typ 2

Die Entscheidung, welche Formel anzuwenden ist (falls mehrere Formeln angewendet werden könnten), wird aufgrund der Analyse von  $(SPM)_i$  und  $PM_{i+1}$  getroffen.

<sup>63</sup> konzipiert in Zusammenarbeit mit und implementiert von H. Garbe

Einzelne Elemente des Diagrams (Abbildung 35) unterscheiden sich bei verschiedenen Aufgabenzielen in ihren Unterkomponenten. Von dem Aufgabenziel hängt es z.B. ab, welcher Typ des Ersatzzweipols ermittelt werden soll bzw. welche Transformationen durchgeführt werden dürfen. Die Parameterberechnung wird ebenfalls dem Aufgabenziel entsprechend durchgeführt.

Anhand des folgenden Beispiels wird erläutert, in welcher Weise die Adaptivität der Lösungen von Aufgaben des Y-Typs 2 realisiert wird, und einige Probleme der Realisierung dieser Adaptivität werden diskutiert.

Das Aufgabenziel „Berechnung des Ersatzwiderstandes“ wird dynamisch durch eine Reihe von Unterzielen ersetzt z.B. „Finde Elemente, die ersetzt werden können“, „Ersetze die Elemente“, „Berechne die Ersatzparameter“. Dabei wird „Finde Elemente, die ersetzt werden können“ nach jedem Schritt bestimmen, welches Ziel als nächstes erreicht werden soll und welcher Schritt als nächster durchgeführt wird. Tabelle 4 zeigt die „Prozedur“ „Fasse parallele Widerstände zusammen“ in Prolog-Notation.

```
gmr(transform(El), [parallel(Elements), newElements(NewEl)], ['#0210']) :-
    gmr(searchOne(El, parallel, resistor), [_], [Elements, Points], []),
    gmr(getParameter(resistance, Elements), Parameters),
    gmr(replace(parallel, [Points, Elements], []), [_], NewEl, _, _),
    gmr(getParameter(resistance, [NewEl]), [R]),
    gmr(calculate(parallelR, Parameters, R), FormulaR).
```

**Tabelle 4. Beispiel eines Lösungsschrittes mit GMR**

Die Anzahl der Lösungsmöglichkeiten zu einer Aufgabe wächst exponentiell mit der Größe des Schaltbildes. Betrachtet man für das Beispiel in Abbildung 38 die Berechnung des Ersatzwiderstandes von vier Widerständen, alle durchführbaren Ersetzungsreihenfolgen und mathematischen Umformungen, so erhält man über 130 verschiedene Lösungswege [Garbe2003, Yakimchuk2004].

mileET verwendet einen eigenen Algorithmus zur Reduzierung der Menge der äquivalenten Formeln, indem alle Formeln intern auf ein systemeigenes Standardformat abgebildet werden<sup>64</sup>. Auf diese Weise kann die Lösungsmenge in diesem Fall auf ca. 10 Lösungen reduziert werden (Abbildung 38).

<sup>64</sup> konzipiert und implementiert von H.Garbe

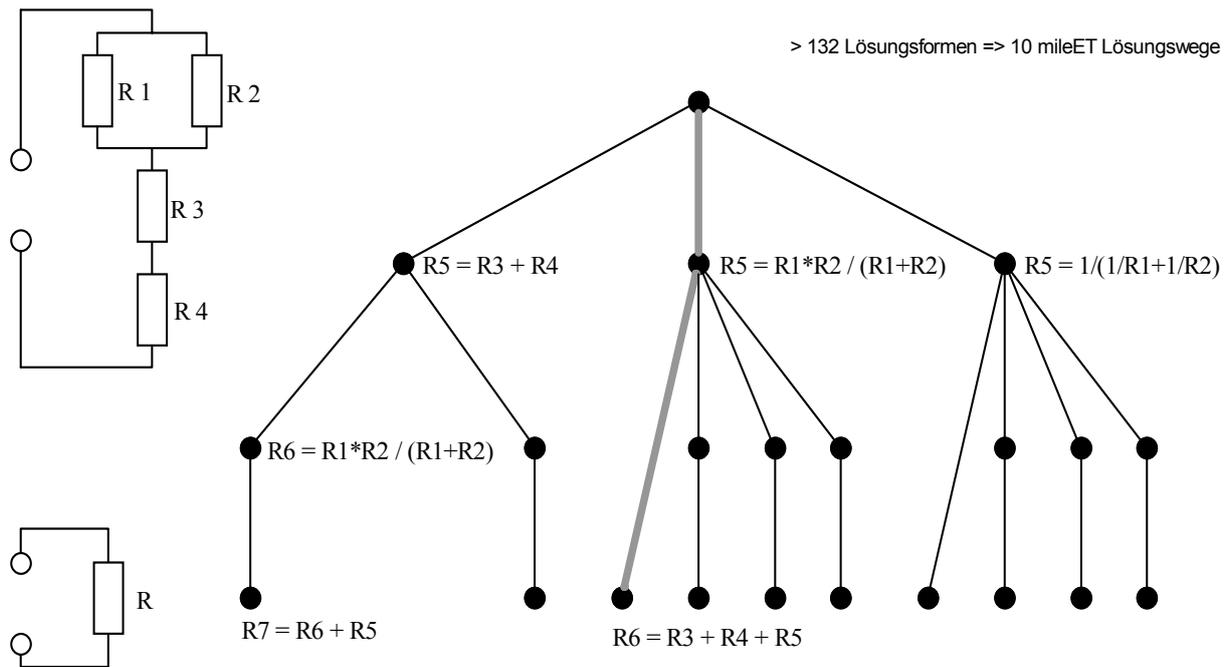


Abbildung 38. mileET-Lösungsraum einer einfachen Aufgabe

Trotzdem müssen, z.B. für die äquivalente parallele Ersetzung, zwei Formen betrachtet werden:

$$R_5 = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2}} \quad \text{und} \quad R_5 = \frac{R_1 \cdot R_2}{R_1 + R_2}.$$

Beide Formeln werden als unterschiedliche Lösungen durch GMR-Regeln erzeugt, und somit als Lösungen erkannt.

Um bei einer vorliegenden zu analysierenden Benutzerlösung nicht alle diese (in diesem Fall noch ca. 10) möglichen Lösungen vom System erzeugen lassen zu müssen, wird die Technik des „*model tracing*“ benutzt (siehe Abschnitt 2.4). Die Lösungsschritte des Benutzers (Schaltbilder und Formeln) werden den Lösungsschritten des Systems zugeordnet. Durch die Einbeziehung der Benutzerlösung in den Lösungsprozess des Systems ergeben sich für den Lerner kürzere Antwortzeiten und eine größere Adaptivität des Systems hinsichtlich seiner Lösungsvorschläge. Dadurch gewinnt mileET einen Teil seiner Adaptivität. Das System erscheint dem Benutzer als verständnisvoll und kooperativ [Garbe2003, Yakimchuk2004].

So wird eine durch searchOne-GMR-Regel (Tabelle 4) gefundene Gruppe von Bauelementen von dem mileET-Meta-Interpreter überprüft und deren Ersetzung nur dann durchgeführt, wenn eine entsprechende Ersetzung in der Benutzer-Lösung gefunden werden kann. Passend dazu entscheidet der Meta-Interpreter, welche Formeln erzeugt werden müssen.

Dank der Flexibilität des *model-tracing*-Algorithmus werden auch vom Benutzer übersprungene Lösungsschritte erkannt. Sobald das System allerdings einen Lösungsschritt des

Benutzers nicht reproduzieren kann, wird der Lösungsprozess unterbrochen, und dem Benutzer wird mitgeteilt, dass sich sein Lösungsvorschlag nicht zu einer korrekten Lösung vervollständigen lässt. Der Berechnungsaufwand wird dank *model tracing* erheblich reduziert und komplexe Aufgaben sind auf diese Weise überhaupt erst berechenbar.

Die anderen typischen Aufgabenstellungen (Abbildung 27) aus dieser Aufgabenklasse sind:

- (a) einen Ersatzzweipol (äquivalente Stromquelle bzw. Spannungsquelle) für die vorgegebene Schaltung zu ermitteln oder
- (b) einige Parameter (Strom durch ein Element bzw. Spannung an einem Element) der angegebenen Schaltung unter Verwendung der Zweipoltheorie zu berechnen.

Bei den (a)-Aufgaben werden in der vorgegebenen Schaltung die Klemmen (in-Element) markiert, bezüglich deren ein äquivalentes Zweipolelement ermittelt werden soll. Bei den (b)-Aufgaben wird der durch einen Zweipol zu ersetzende Teil der Schaltung durch das(die) Element(e) gekennzeichnet, für das(die) die Parameter gesucht werden. In beiden Fällen kann von den Aufgabenautoren durch das Formulieren des Aufgabenziels vorgegeben werden, ob der Lerner eine äquivalente Spannungsquelle oder eine äquivalente Stromquelle als Ersatzzweipol ermitteln soll, alternativ kann die Entscheidung dem Lernenden überlassen werden (Tabelle 5).

Das zu ermittelnde Ersatzelement wird vorgegeben	Die Entscheidung über das zu ermittelnde Ersatzelement wird dem Lernenden überlassen
equivalentIq(IN) equivalentUq(IN) current_equivalentUq(Element) voltage_equivalentUq(Element) current_equivalentIq(Element) voltage_equivalentIq(Element)]	equivalent2Terminal(IN) current_equivalent2Terminal(Element) voltage_equivalent2Terminal(Element)

**Tabelle 5. Aufgabenziele zum Thema Anwendung der Zweipoltheorie**

Im Unterschied zu den Aufgaben der Ersatzwiderstands- oder Eingangsimpedanzberechnung werden bei der Ermittlung von Ersatzspannungs- und Ersatzstromquellen Strom und Spannungsrichtungen berücksichtigt.

Für die Aufgaben mit dem Ziel, den Strom bzw. die Spannung für ein Element zu berechnen, wird die Lösung auf der Basis der Ersatzelementberechnung aufgebaut.

Zusätzlich müssen die Richtungen für die gesuchten Strom-/Spannungen gesetzt werden (oder aus der Aufgabenstellung übernommen werden): `gmr (setParameterDirection (iu, Res1), _) .`

Nachdem das Ersatzelement gefunden wurde, werden die gesuchten Parameter symbolisch berechnet. Dafür stehen zwei GMRs zur Verfügung:

```
gmr (calculate (simpleIDivider, [I12, R1, R2], I1), Formula)
gmr (calculate (ohmLforI, [[R1, R2], U12], I1), Formula)
```

Die erste ermittelt den gesuchten Wert über die Stromteiler-Regel, die zweite über das Ohm'sche Gesetz

```
gmr (current_equivalent2Terminal (Res1), _, ['#0401']) :-
    gmr (setParameterDirection (iu, Res1), _),
    gmr (current_equivalent2Terminal1 (Res1), _).

gmr (current_equivalent2Terminal1 (Res1), _, ['#0403']) :-
    gmr (check (simpleDivider (Type, [Input, Res1, Res2])), _),
    gmr (current (Type ([Input, Res1, Res2])), _).

gmr (current_equivalent2Terminal1 (Res1), _, ['#0404']) :-
    gmr (equivalent2Terminal (Res1), _),
    gmr (current_equivalent2Terminal1 (Res1), _).

gmr (current (simpleIDivider ([Input, Res1, Res2])), I1, ['#0430']) :-
    ...
    gmr (calculate (simpleIDivider, [I12, R1, R2], I1), Formula).

gmr (current (simpleUDivider ([Input, Res1, Res2])), Formula, ['#0431']) :- ...
    ...
    gmr (calculate (ohmLforI, [[R1, R2], U12], I1), Formula).
```

**Tabelle 6. Aufbau der Methoden für die Parameterberechnung**

### 5.2.3.2 Anwendung der Methoden der Kirchhoff'schen Sätze

Das Erstellen eines Kirchhoff'schen Gleichungssystems, die Anwendung der Methode der Knotenspannungs- und der Maschenstromanalyse für die Berechnung einzelner Parameter der vorgegebenen Schaltung sind die Aufgabenstellungen der Aufgaben Y-Typ 1 (Abbildung 27).

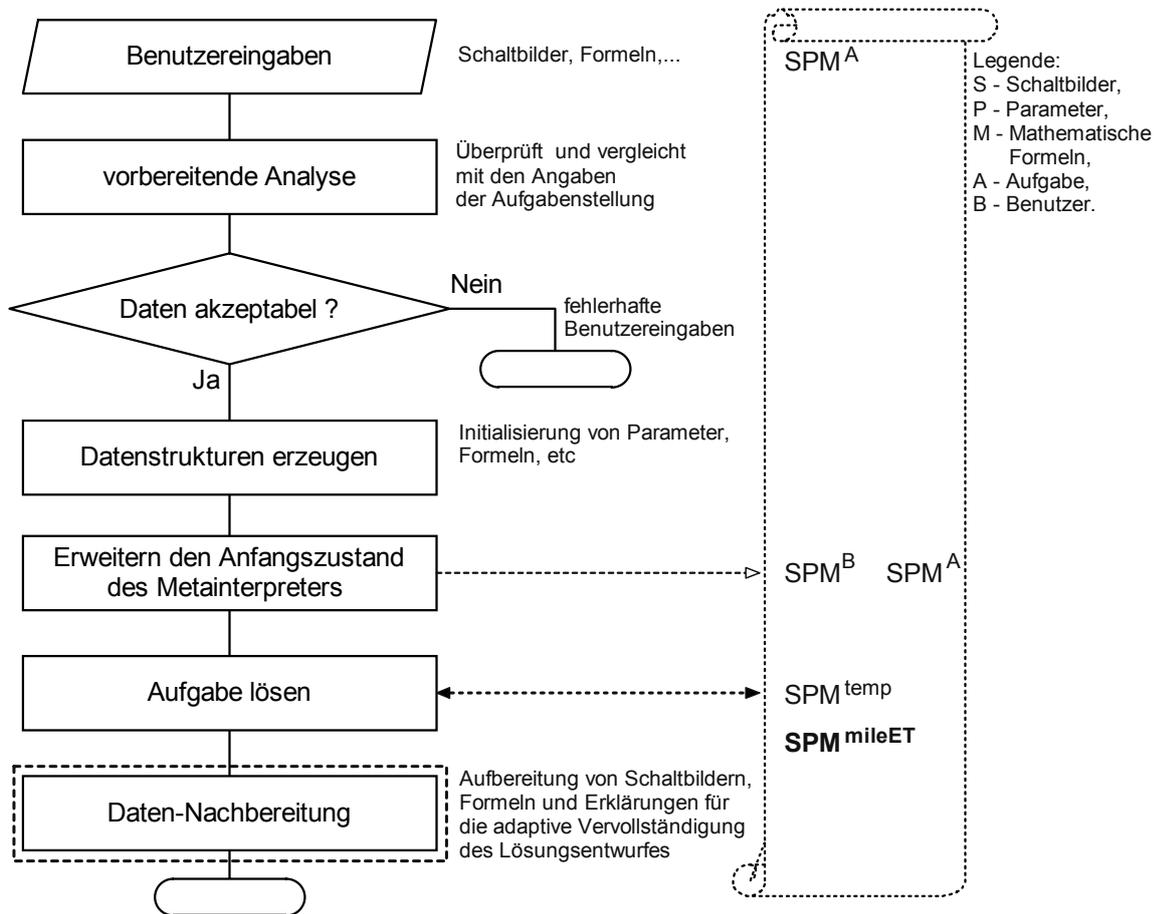


Abbildung 39. Lösen einer Aufgabe des Y-Typs 1

Das Diagramm auf der Abbildung 39 ähnelt dem Diagramm auf der Abbildung 32.

Der Unterschied besteht darin, dass bevor mileET mit dem Erzeugen der Lösung beginnt, eine vorbereitende Analyse durchgeführt wird: potenzielle Maschen für die Maschengleichungen werden ermittelt. Bei dieser Ermittlung werden die Zweige mit  $I_q$ -Elementen nicht berücksichtigt und werden erst in das Aufstellen der Knotengleichungen einbezogen.

Die Tatsache, dass die Zusammenschaltung der Elemente in den Aufgaben dieser Gruppe während des gesamten Löseprozesses gleich bleibt, erlaubt das Auslagern solcher Fehlerdiagnosen aus dem GMR-Teil („Aufgabe lösen“) in die vorbereitende Analyse. Dabei werden z.B. die Maschen oder Zweige aufgrund der Aufgaben- und Benutzerdatenanalyse außerhalb GMRs zur Benutzung in den GMRs bestimmt, was eine Reduzierung des Lösungsaufwandes zur Folge hat: die Auswahlmöglichkeiten der notwendigen Zusatzparameter

werden eingegrenzt. Zusätzlich können differenzierte Rückmeldungen über die Fehlerquelle ausgegeben werden.

Die Abbildung 40 zeigt die allgemeine Methode der Kirchhoff'schen Gleichungen, deren Prolog-Implementation befindet sich in der Tabelle 7.

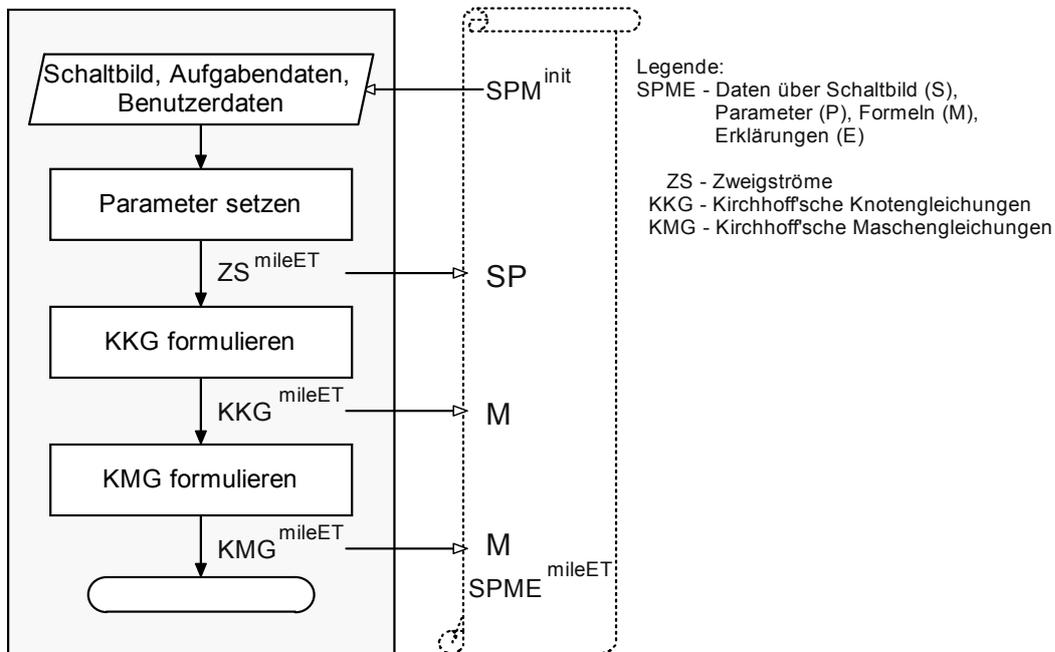


Abbildung 40. Methode der Kirchhoff'schen Gleichungen

```

gmr(kls(_,_,['#0300']) :-
    gmr(setParameters(forKLS),_),
    gmr(kcls1(_,_),
    gmr(kvls1(_,_) .

gmr(kcls1(_,_,['#0302']) :-
    gmr(search(nodes),Nodes),
    gmr(independentNodes(Nodes),IndNodes),
    gmr(equationsKCLs(IndNodes),_) .

gmr(kvls1(_,_,['#0301']) :-
    gmr(search(nrIndMeshes),NrIndependentMeshes),
    gmr(search(meshesNoIq),Meshes),
    gmr(independentMeshes(Meshes,NrIndependentMeshes),IndMeshes),
    gmr(formulate(kvlsSystem,IndMeshes),_) .

```

Tabelle 7. Prolog-Notation für die Methode der Kirchhoff'schen Gleichungen

Am Beispiel der Schaltung (Abbildung 41) wird die Arbeit des mileET-Expertensystems beim Lösen der Aufgabe zum Thema Kirchhoff'sche Gleichungen illustriert.

Beim Starten der Beispiel-Aufgabe wird dem mileET-Assistenten die Aufgabenstellung mitgeteilt.

```
etTask(kls(_), [[
  [resistor, 'R1', [p4,p1], ['R_R1', [[_,_], 'I_R1'], [[_,_], 'U_R1'], 'P_R1']],
  [resistor, 'R2', [p4,p3], ['R_R2', [[_,_], 'I_R2'], [[_,_], 'U_R2'], 'P_R2']],
  [resistor, 'R3', [p4,p2], ['R_R3', [[_,_], 'I_R3'], [[_,_], 'U_R3'], 'P_R3']],
  [powerSourceVolt, 'Q4', [p1,p2], [$, [[_,_], 'I_Q4'], [[p1,p2], 'U_Q4'], 'P_Q4']],
  [powerSourceVolt, 'Q5', [p3,p2], [$, [[_,_], 'I_Q5'], [[p2,p3], 'U_Q5'], 'P_Q5']]
], [], 5,4], []).
```

Der Benutzer erstellt seinen Lösungsentwurf (ein möglicher Bearbeitungsstand wird in der Abbildung 41 rechts dargestellt) indem er z.B. die Zweigströme setzt und die Knotengleichungen aufstellt.

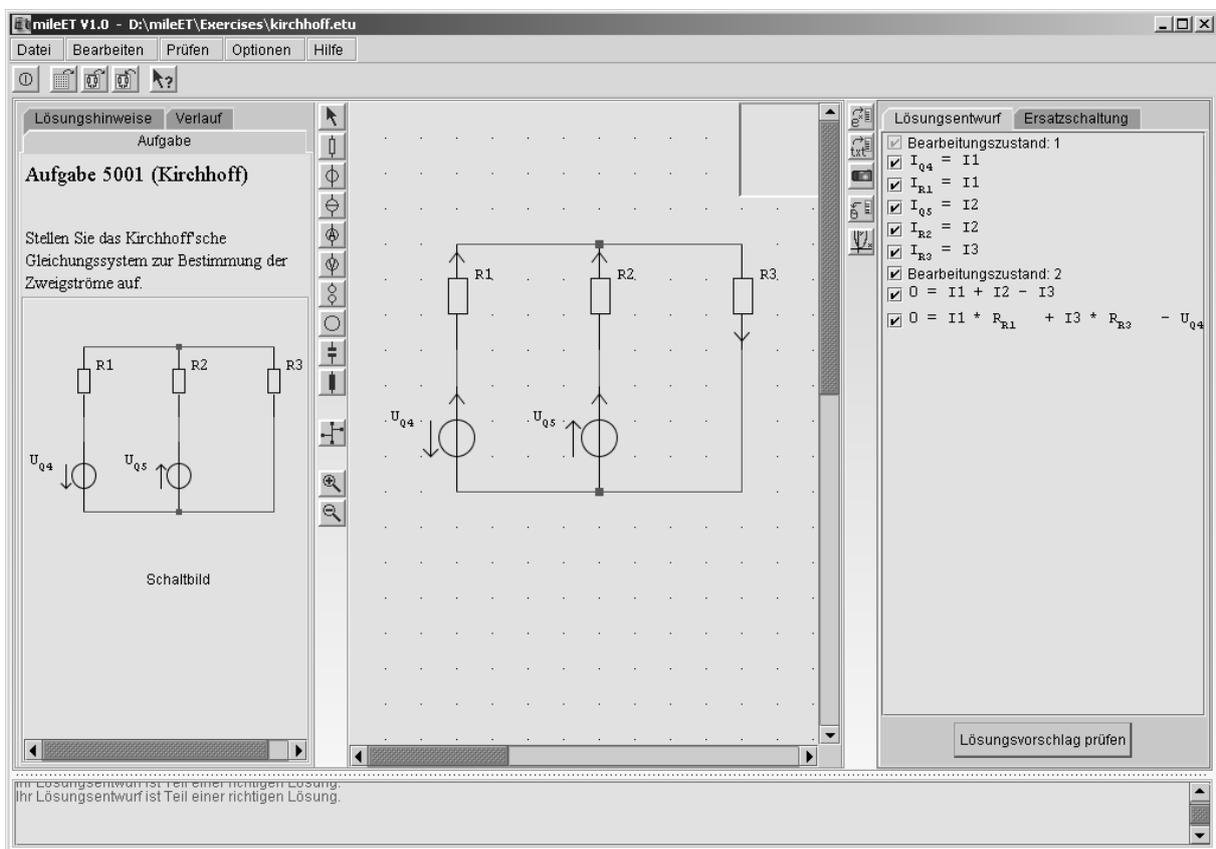


Abbildung 41. Benutzerlösungsentwurf in mileET

Die Zweigströme werden über ein Kontextmenü gesetzt, was im Schaltbild durch Strompfeile gekennzeichnet und im Lösungsentwurfeditor durch die automatisch erstellte Gleichungen der Form  $I_{R1}=I1$  zu erkennen ist. Da in der Aufgabenstellung keine Zweigströme vorgegeben worden sind, hat sich das Schaltbild durch Setzen der Zweigströme verändert. Der Benutzer übernimmt das neue Schaltbild in seinen Lösungsentwurf (Bearbeitungsstand 2), andernfalls wird das System danach fragen, sobald der Benutzer versucht, seinen Lösungsentwurf überprüfen zu lassen bzw. eine Vervollständigung anfordert.

Der Lösungsentwurf kann jederzeit überprüft werden:

```
etHypothesisTesting([
  [
    [resistor, 'R1', [p4,p1], ['R_R1', [[_,_], 'I_R1'], [[_,_], 'U_R1'], 'P_R1']],
    [resistor, 'R2', [p4,p3], ['R_R2', [[_,_], 'I_R2'], [[_,_], 'U_R2'], 'P_R2']],
    [resistor, 'R3', [p4,p2], ['R_R3', [[_,_], 'I_R3'], [[_,_], 'U_R3'], 'P_R3']],
    [powerSourceVolt, 'Q4', [p1,p2], [$, [[_,_], 'I_Q4'], [[p1,p2], 'U_Q4'], 'P_Q4']],
    [powerSourceVolt, 'Q5', [p3,p2], [$, [[_,_], 'I_Q5'], [[p2,p3], 'U_Q5'], 'P_Q5']]
  ],
  [], 5, 4], - Schaltung 1 (Bearbeitungsstand 1, in der Aufgabenstellung vorgegebene Schaltung)
  [ [Ströme in den Elementen werden angezeigt
    [resistor, 'R1', [p4,p1], ['R_R1', [[p1,p4], 'I_R1'], [[_,_], 'U_R1'], 'P_R1']],
    [resistor, 'R2', [p4,p3], ['R_R2', [[p3,p4], 'I_R2'], [[_,_], 'U_R2'], 'P_R2']],
    [resistor, 'R3', [p4,p2], ['R_R3', [[p4,p2], 'I_R3'], [[_,_], 'U_R3'], 'P_R3']],
    [powerSourceVolt, 'Q4', [p1,p2], [$, [[p2,p1], 'I_Q4'], [[p1,p2], 'U_Q4'], 'P_Q4']],
    [powerSourceVolt, 'Q5', [p3,p2], [$, [[p2,p3], 'I_Q5'], [[p2,p3], 'U_Q5'], 'P_Q5']]
  ],
  [ gesetzte Zweigströme:
    [branchCurrent, [p2,p1,p4], ['Q4', 'R1'], 'I1'],
    [branchCurrent, [p2,p3,p4], ['Q5', 'R2'], 'I2'],
    [branchCurrent, [p4,p2], ['R3'], 'I3']
  ], 5, 4], - Schaltung 2 (Bearbeitungsstand 2, Schaltung mit gesetzten Zweigströmen)
  [ [Spannungen an den Elementen werden angezeigt
    [resistor, 'R1', [p4,p1], ['R_R1', [[p1,p4], 'I_R1'], [[p1,p4], 'U_R1'], 'P_R1']],
    [resistor, 'R2', [p4,p3], ['R_R2', [[p3,p4], 'I_R2'], [[p3,p4], 'U_R2'], 'P_R2']],
    [resistor, 'R3', [p4,p2], ['R_R3', [[p4,p2], 'I_R3'], [[p4,p2], 'U_R3'], 'P_R3']],
    [powerSourceVolt, 'Q4', [p1,p2], [$, [[p2,p1], 'I_Q4'], [[p1,p2], 'U_Q4'], 'P_Q4']],
    [powerSourceVolt, 'Q5', [p3,p2], [$, [[p2,p3], 'I_Q5'], [[p2,p3], 'U_Q5'], 'P_Q5']]
  ],
  [
    [branchCurrent, [p2,p1,p4], ['Q4', 'R1'], 'I1'],
    [branchCurrent, [p2,p3,p4], ['Q5', 'R2'], 'I2'],
    [branchCurrent, [p4,p2], ['R3'], 'I3']
  ], 5, 4]],
```

Beziehungen für die Zweigströme und Kirchhoff'schen Gleichungen:

```
[[ 'I_Q5'='I2'], ['I_Q4'='I1'], ['I_R3'='I3'], [0=(( 'I1'+ 'I2')- 'I3')],
[0=(( ( 'I1'* 'R_R1')+( 'I3'* 'R_R3'))- 'U_Q4')], ['I_R2'='I2'], ['I_R1'='I1']] .
```

Die Überprüfung dieses Lösungsentwurfs über das Hypothesentesten liefert eine positive Rückmeldung.

Auf Anfrage kann eine Vervollständigung des Lösungsentwurfes geliefert werden:

```
[#0300, [[#0508, [[for41], [for42], [for43], [for44], [for45]], [cir6]]],
[[#0302, [[#0304]], [#0504, [#0306, [for46]]]],
[#0301, [[#0303]], [[[#0305, [for47]], [[#0305, [for48]]]]]]
```

Die Lösungskommentare (Format '#ZAHL') sind in der Datei prolog.xml zusammengefasst und können von den Autoren editiert werden.

Der Lösungshinweis in der Abbildung 42 stellt diese Baumstruktur in einem lesbaren Format der kommentierten Lösungsschritte dar. Alle Formeln lassen sich per Knopfdruck anzeigen. Zusätzliche Lernmodule können über Kommentar-Links erreicht werden.

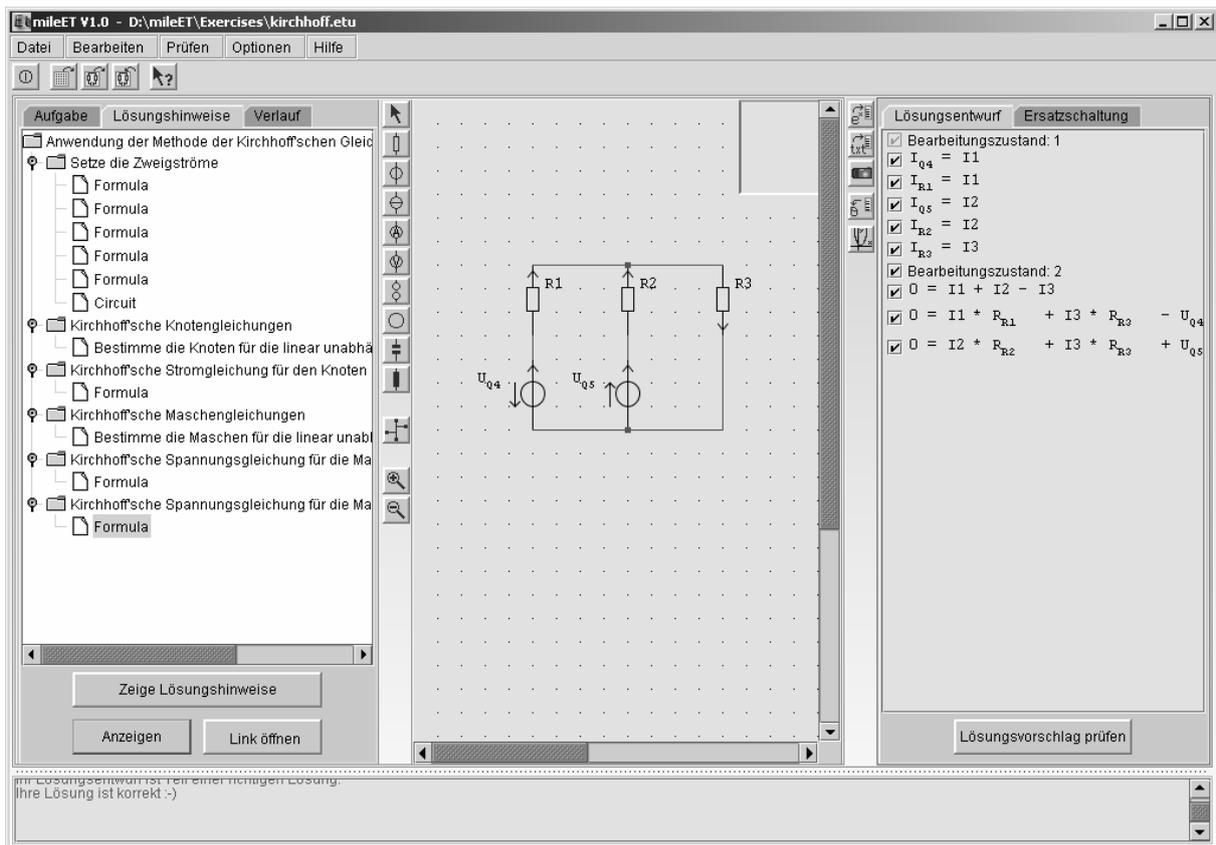


Abbildung 42. Vervollständigung eines Lösungsentwurfes mit Kirchhoff'schen Gleichungen

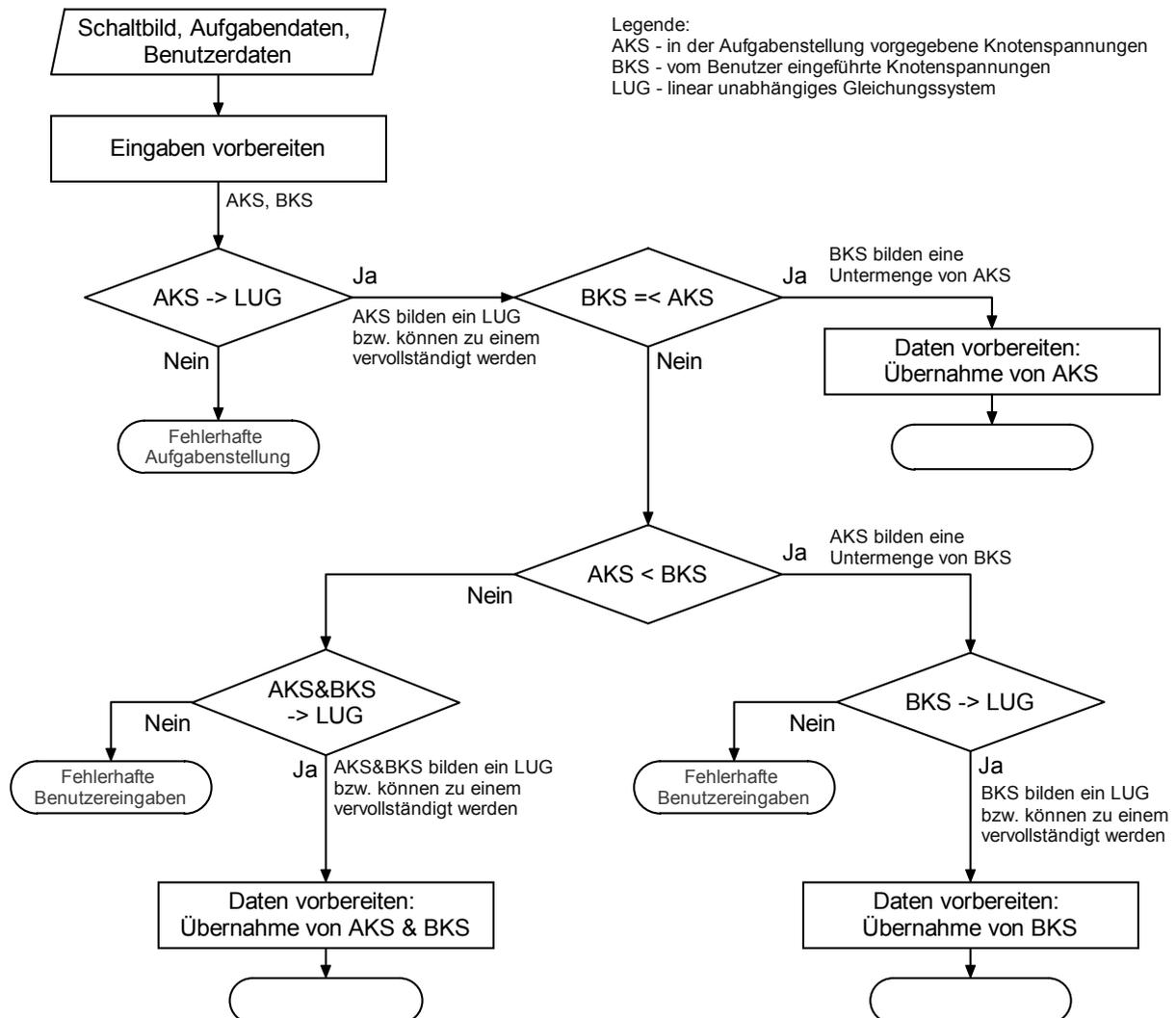
### 5.2.3.3 Anwendung der Methode der Knotenspannungsanalyse

Eine typische Aufgabenstellung zum Üben dieser Methode ist, ein Gleichungssystem mit Knotenspannungen zur Bestimmung der Zweigströme aufzustellen.

Bei dieser Methode werden die Knotenspannungen als Hilfsvariablen eingeführt. Ein Knoten wird als Referenzknoten gesetzt, die Spannungen zwischen den übrigen Knoten und dem Referenzknoten werden als linear unabhängige Knotenspannungen bezeichnet. Der Strom in jedem Zweig kann nach dem Ohm'schen Gesetz über die Knotenspannungen bestimmt werden.

Auch hier wird eine vorbereitende Analyse durchgeführt: die in der Aufgabe vorgegebenen und die vom Benutzer eingegebenen Knotenspannungen werden miteinander verglichen und

auf die lineare Unabhängigkeit hin überprüft. Das letzte geschieht über die Ermittlung der Anzahl der notwendigen Knoten.



**Abbildung 43. Vorbereitende Analyse der Methode der Knotenspannungsanalyse**

Die Abbildung 43 illustriert die Logik dieses Vergleichs. Die gleiche Logik wird auch in der Methode der Maschenstromanalyse verwendet: anstelle der BKS (Knotenspannungen des Benutzers) werden die Maschenströme (BMS) eingesetzt.

Beim Erzeugen einer Lösung mit der Methode der Knotenspannungsanalyse geht das Programm wie folgt vor (Abbildung 44).

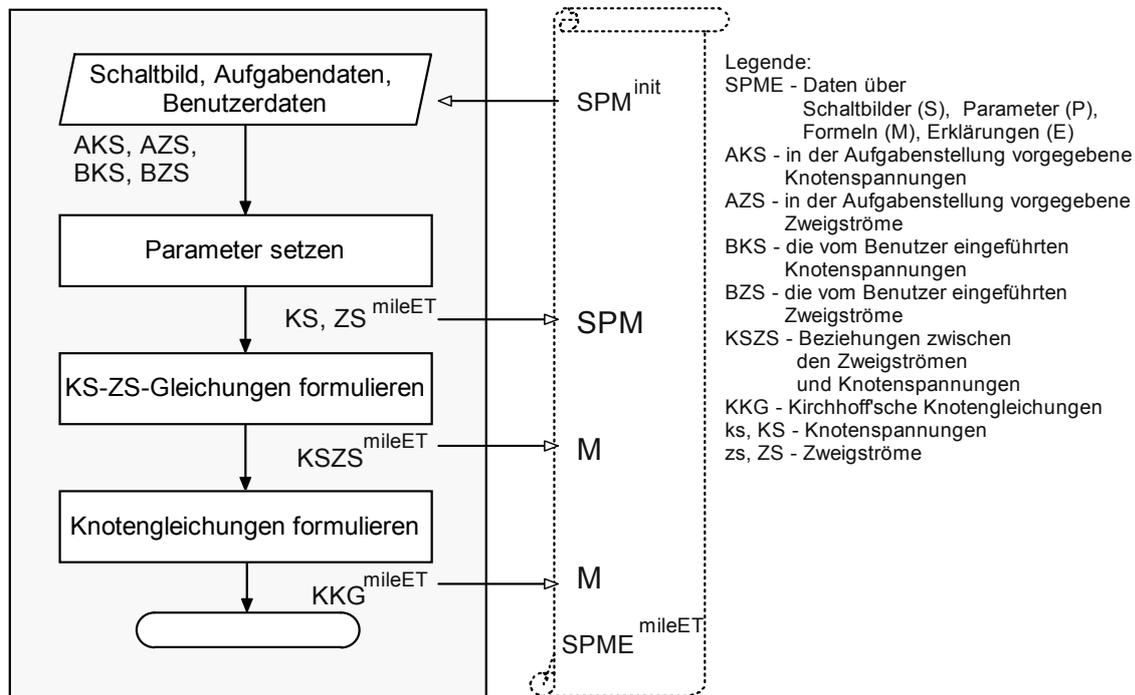
Zuerst werden alle notwendigen symbolischen Parameter gesetzt:

- ein Referenzknoten wird gesetzt,
- Knotenspannungen werden erzeugt,
- alle Zweigströme werden gesetzt,

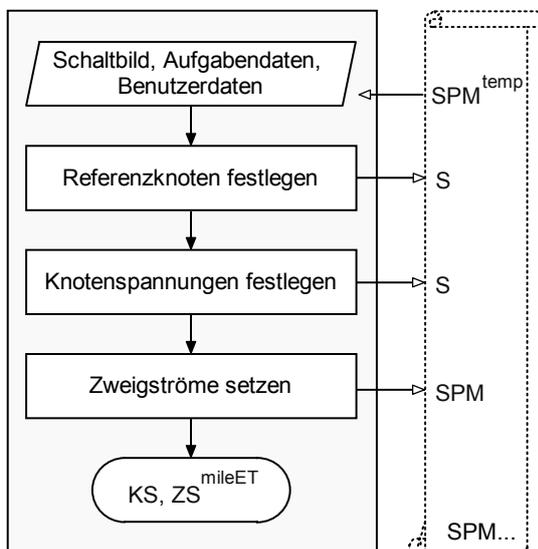
und dann werden die erforderlichen Gleichungen aufgestellt:

- Beziehungen zwischen den Zweigströmen und den Knotenspannungen,

- Kirchhoff'sche Knotengleichungen für alle Knoten mit Ausnahme des Referenzknotens.



#### Parameter setzen



#### KS-ZS-Gleichung formulieren

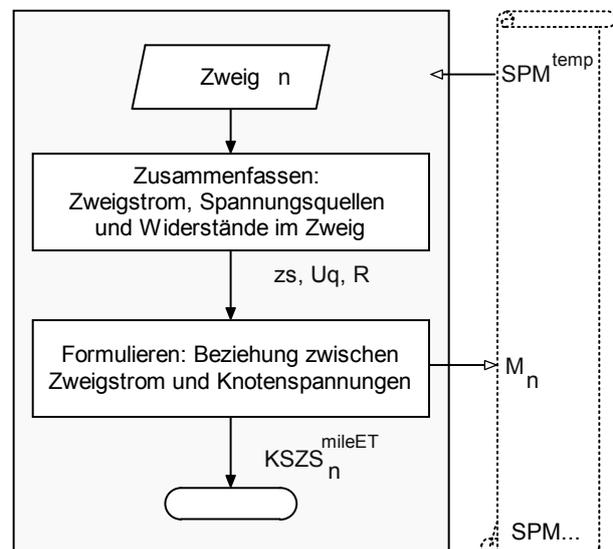


Abbildung 44. Methode der Knotenspannungsanalyse

In Prolog-Notation sieht diese Lösungsmethode wie folgt aus:

```
gmr (nvs (_, _), ['#0501']) :-
    gmr (setParameters (forNVS), KCLsNodes),
    gmr (search (branchesNoIq), BranchesNoIq),
    gmr (equationsNVs (BranchesNoIq), _),
    gmr (equationsKCLs (KCLsNodes), _).
```

Der Schwerpunkt liegt bei dieser Aufgabe in der Erkennung der einzuführenden Hilfsvariablen und der Formulierung der entsprechenden Beziehungen zwischen vorgegebenen und eingesetzten Schaltungsparametern. In der aktuellen Version wird keine Lösung des erstellten linearen Gleichungssystems durchgeführt oder kontrolliert<sup>65</sup>.

Betrachten wir die einzelnen Schritte genauer, beginnend mit dem Setzen der Hilfsparameter. Dieser Schritt besteht aus drei Unterschritten: (1) die Auswahl des Referenzknotens und das Setzen der (2) Knotenspannungen und (3) Zweigströme.

(1) In dem Fall, dass der Referenzknoten in der Aufgabenstellung vorgegeben oder von dem Benutzer angegeben wird, wird die Auswahl übernommen (Vorgaben aus der Aufgabenstellung haben Vorrang), sonst wird einer der Knoten als Referenzknoten gesetzt.

Nach der Auswahl des Referenzknotens werden alle notwendigen Knotenspannungen erzeugt:

```
[nodeVoltage, [pin1,pin0], 'U10']
[nodeVoltage, [pin2,pin0], 'U20 ']
...
[nodeVoltage, [pinN,pin0], 'UN0']
```

Beim Setzen der Zweigströme analysiert das System die Schaltung und sucht alle Zweige aus, die nicht aus einer Stromquelle bestehen.

In dem Stromquelle-Zweig ist der Strom schon durch diese Quelle definiert und wird daher in dem Gleichungssystem als eine bekannte Größe erscheinen. Liegt die Information über die Zweigströme vor (Aufgabenstellung, Benutzereingaben), übernimmt mileET sie und erzeugt die Lösung weiter. Auf die Konsistenz der Benutzereingaben und der vorrangigen Vorgaben der Aufgabenstellung wurde schon während der vorbereitenden Analyse geachtet (Abbildung 43).

Sind die Zweigströme nicht durch den Benutzer (Lehrender oder Lernender) gesetzt, werden sie von mileET gesetzt. Die Richtung wird durch die Pins-Ergebnisse der Zweigsuche bestimmt: `findBranches(Circuit, Branches)`, wobei `Branches` eine Liste mit Zweigen `[[Pins1,Elements1],[Pins2,Elements2],...,[PinsN,ElementsN]]` ist.

Für die Beziehungen Zweigstrom-Knotenspannungen werden die Zweig-Parameter (Zweigstrom, Zweigwiderstände, Zweigspannungsquelle) und die entsprechenden Knotenspannungen benötigt.

Dabei sind folgende Fälle<sup>66</sup> zu unterscheiden:

---

<sup>65</sup> eine Analyse des Lösungsentwurfes (mit den Schritten zum Lösen eines Gleichungssystems) nach IPSE-Prinzipien wäre allerdings auch im Falle des CAS-Einsatzes nicht möglich, denn die analysierte CAS liefern nur die (symbolischen) Endlösungen und liefern keine Erklärungen für ihre Methoden.

<sup>66</sup> Komplexere Fälle mit mehreren Quellen in einem Zweig werden hier nicht betrachtet.

- a) Ein Zweig besteht aus einer Stromquelle
- b) Einer der Zweig-Knoten ist der Referenzknoten
- c) Keiner der Zweig-Knoten ist der Referenzknoten

Eine Stromquelle (ein Zweig mit einer Stromquelle) spielt beim Erstellen der Kirchhoff'schen Knotengleichungen eine Rolle, beim Aufstellen der Gleichungen mit Knotenspannungen werden solche Zweige nicht beachtet (Fall a).

Für den Fall, dass die Zweigspannung eines Zweiges mit Widerständen durch eine Knotenspannung gekennzeichnet wird (Fall b), wird die Gleichung unter Berücksichtigung des Vorzeichens des Zweigstromes und der Knotenspannung erzeugt. Haben der Zweigstrom und die Knotenspannung die gleiche Richtung, dann gilt:

$$I_{\text{Zweig}} = \frac{U_{N0}}{R_{\text{Zweig}}},$$

andernfalls:

$$I_{\text{Zweig}} = -\frac{U_{N0}}{R_{\text{Zweig}}}.$$

Im Falle c) werden für die Beschreibung der Zweigspannung eines Zweiges mit Widerständen zwei Knotenspannungen benötigt.

Sei die Richtung des Zweigstroms  $[p1, p2]$  durch die Punkte  $p1, p2$  gegeben und sei der Referenzknoten  $p0$ , so ist die erste Wahl des Programms die Zweigspannung  $U_{12} = U_{10} - U_{20}$  und der Zweigstrom = Zweigspannung/Zweigwiderstand, als zweite Möglichkeit wird  $U_{21} = U_{20} - U_{10}$  versucht.

Die Betrachtungen der Zweige mit den Spannungsquellen unterscheiden sich durch die Berücksichtigung der Spannungsrichtung der Quellen in der Gleichung:

$$I_{\text{Zweig}} = \frac{U_{N0} \pm U_{q\text{Zweig}}}{R_{\text{Zweig}}}$$

bzw.

$$I_{\text{Zweig}} = \frac{U_{NM} \pm U_{q\text{Zweig}}}{R_{\text{Zweig}}},$$

$$U_{NM} = U_{N0} - U_{M0}.$$

Der mileET-Schaltungseditor bietet eine graphische Unterstützung für diese Methode (Abbildung 45): der Referenzknoten und die Knotenspannungen können in dem Schaltbild gekennzeichnet werden. Für das Setzen der Zweigströme steht ein Kontextmenü zur Verfügung, die Knotenspannungen werden über spezielle Pins gesetzt.

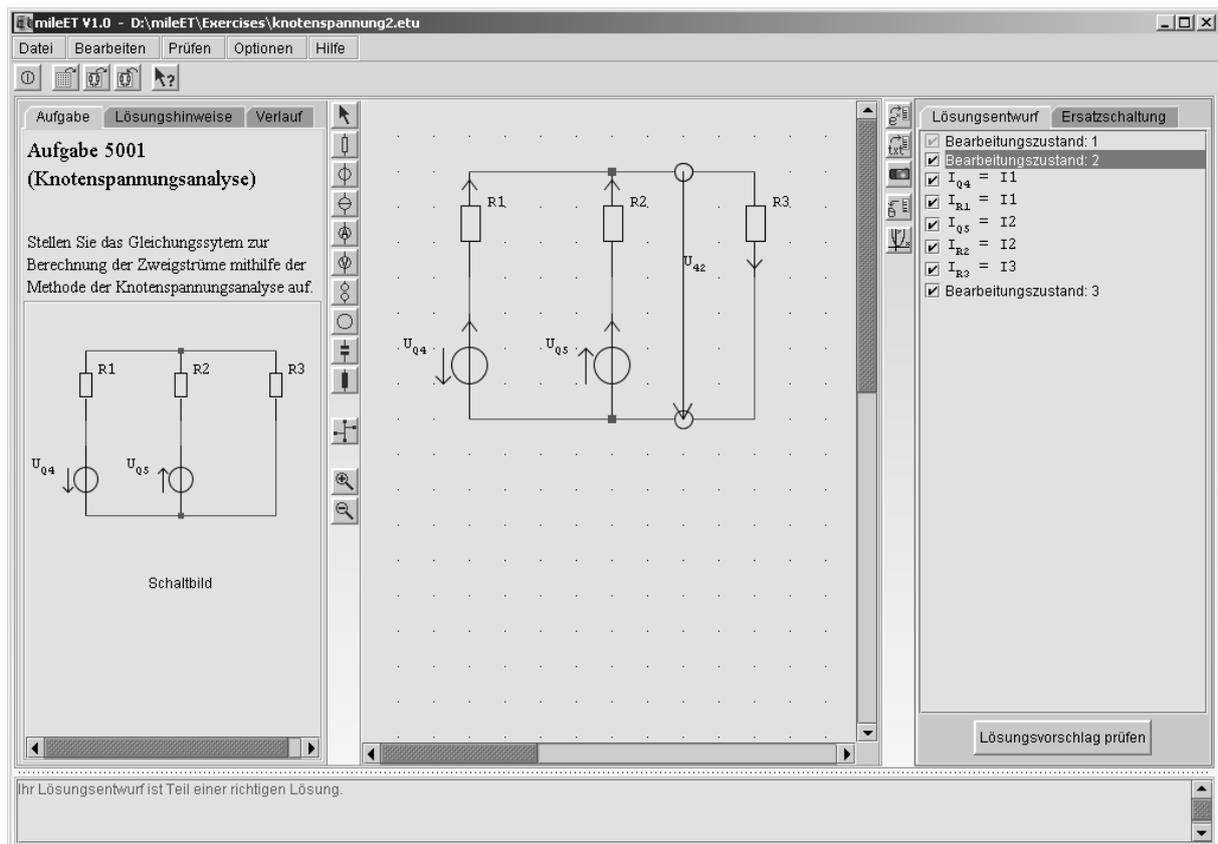


Abbildung 45. Graphische Unterstützung der Methode der Knotenspannungsanalyse

Eine Verbesserung der Handhabung der Zweigströme sowie eine Anpassung des Formeleditors sind wünschenswert: zurzeit werden die Strombezeichnungen nicht angezeigt und im Formeleditor gibt es keinen Zugriff auf die Zusatzparameter (Zweigströme, Knotenspannung, Maschenstrom) der Schaltung.

#### 5.2.3.4 Anwendung der Methode der Maschenstromanalyse

Bei dieser Methode werden Maschenströme als Hilfsvariablen eingesetzt. Dadurch lässt sich der Rang des Gleichungssystems der Kirchhoff'schen Gleichungen einer Schaltung reduzieren, was die Ermittlung der Lösung vereinfacht. Man wähle linear unabhängige Maschen in der Schaltung und setze für diese Maschen Maschenströme ein. Für diese Maschen erstellt man Kirchhoff'sche Maschengleichungen unter Verwendung der neu gesetzten Maschenstromvariablen. Die Lösung des Systems bilden die eingeführten Maschenströme. Da sie allerdings in eindeutigen Beziehungen zu den Zweigströmen stehen, ermöglichen sie die Herleitung der eigentlich gesuchten Zweigströme.

Der Student soll also bei diesen Aufgaben die Zweigströme und die Maschenströme setzen, falls sie nicht vorgegeben sind, die Beziehungen zwischen Maschen- und Zweigströmen aufschreiben und das reduzierte Kirchhoff'sche Gleichungssystem für die Maschenströme aufstellen (Abbildung 46).

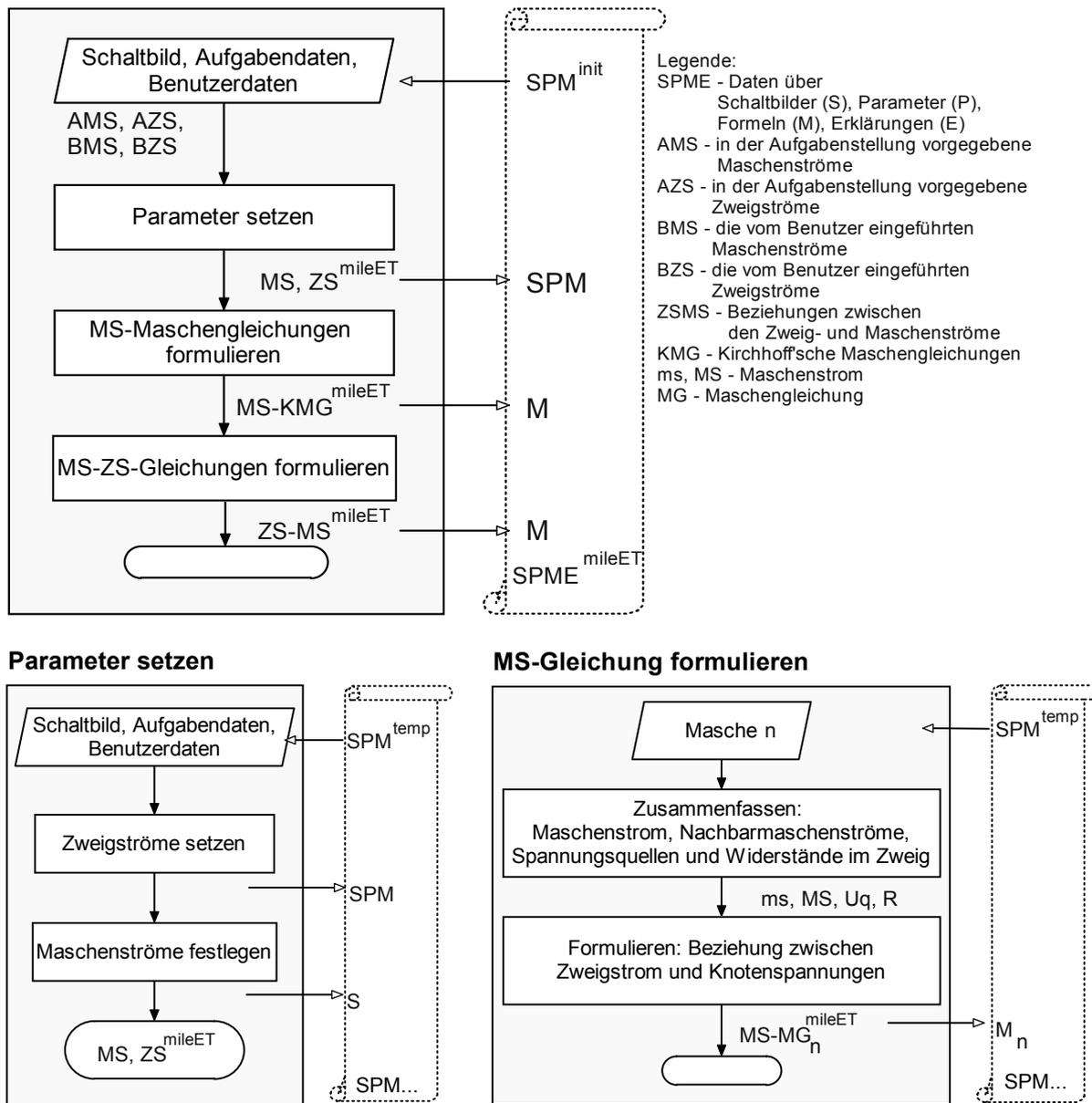


Abbildung 46. Methode der Maschenstromanalyse

Im Wesentlichen ist die Verarbeitung der Aufgaben zu dem Thema Anwendung der Maschenstromanalyse der Verarbeitung der Aufgaben mit den Knotenspannungen ähnlich. Es wird eine vorbereitende Analyse durchgeführt und im Falle der widerspruch- und fehlerfreien Eingaben eine Vervollständigung des Lösungsentwurfes erzeugt.

Eine Gleichung wird in Form

$$I_M \cdot (\sum R) \pm I_N \cdot (\sum R') = \sum_{vzb} U_q$$

erzeugt, wobei  $I_M$  – der Umlaufstrom der betrachteten Masche ist,  $I_N$  – ein Nachbar-Umlaufstrom,  $R$  – Widerstände mit dem Umlaufstrom  $I_M$ ,  $R'$  – Widerstände mit dem Nachbar-Umlaufstrom.

Ein Zweigstrom wird über die Formel  $I = I_M \pm I_N$  bestimmt.

MileET bietet für diese Methode die entsprechende graphische Unterstützung (Abbildung 47).

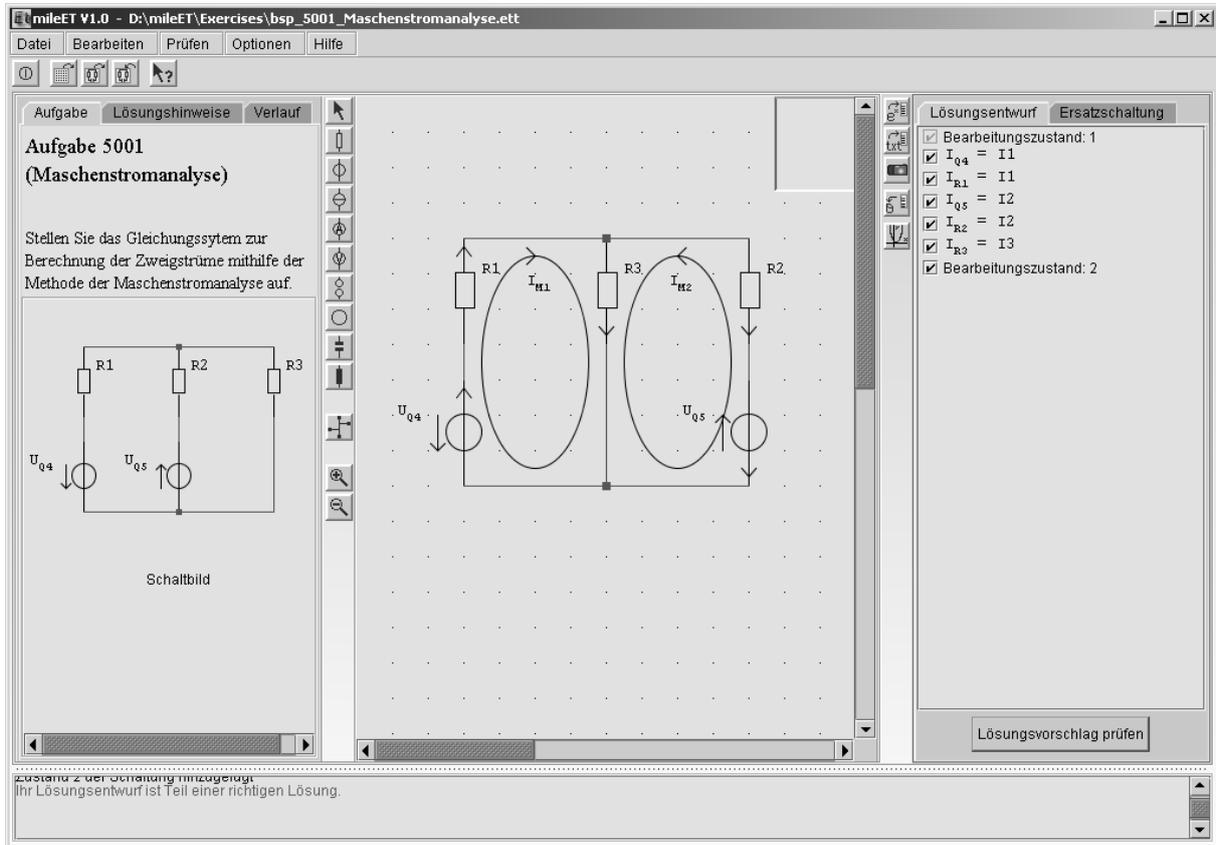


Abbildung 47. Graphische Unterstützung der Methode der Maschenstromanalyse

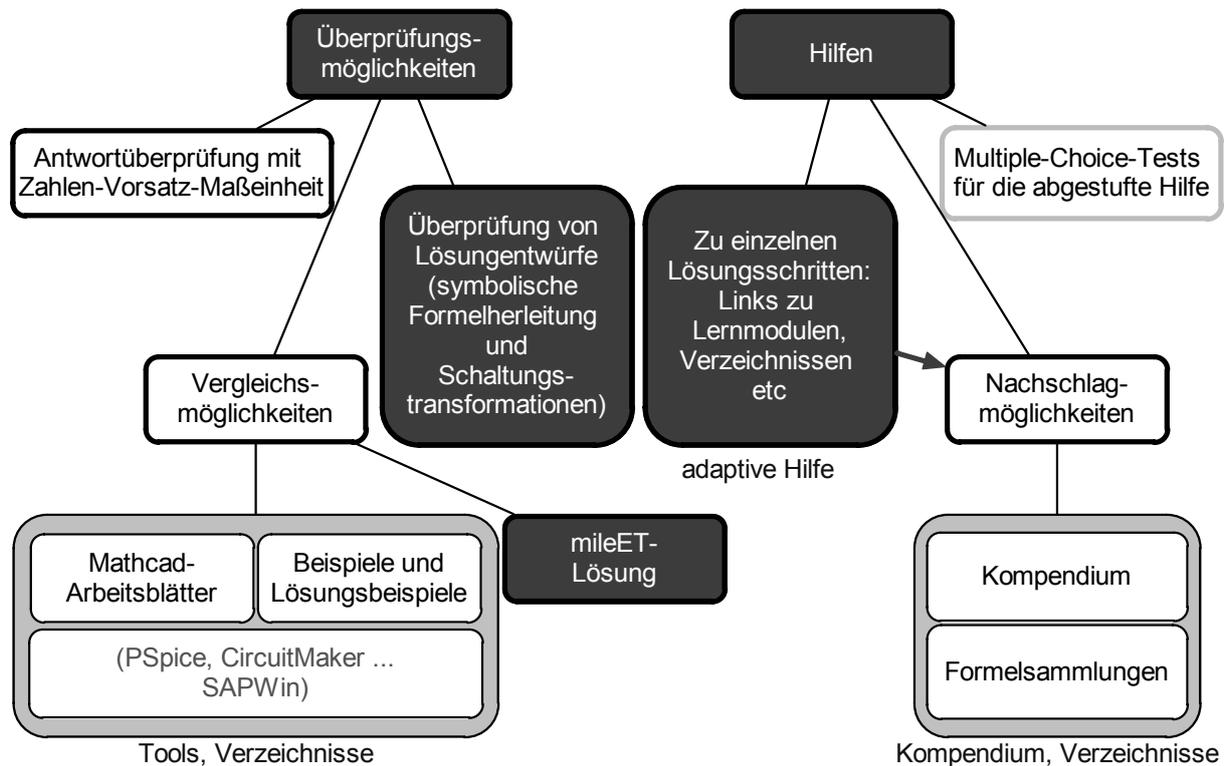
Die Zweigströme können wie bei den anderen Methoden gesetzt werden. Für die Markierung der Maschenströme steht ein Kontextmenü „Maschenstrom setzen“ zur Verfügung. Der Name des Maschenstroms lässt sich in der aktuellen Version von mileET nicht ändern.

### 5.3 Zusammenfassung

Durch den Einsatz von mileET können die GETsoft Feedback-Techniken und die Lernerunterstützung erweitert und qualitativ verbessert werden (vgl. Abbildung 10, S.50).

Den Lehrenden wird die Erstellung von Aufgaben verschiedenen Komplexitätsniveaus ermöglicht, mit denen sich die Lernenden aktiv und kooperativ in mileET auseinandersetzen können. Die Lernenden können vorhandenes Wissen und bestehende Fertigkeiten zum selbständigen Lösen dieser Aufgaben einsetzen. Der mileET-Benutzer bekommt die Möglichkeit, symbolische Lösungsentwürfe zu erstellen und sie vom System überprüfen zu lassen. Zu einem herausragenden Merkmal der mileET Lernumgebung gehört die Fähigkeit, partielle symbolische Lösungsentwürfe zu analysieren und zu vervollständigen. Keine andere

GET-Lernumgebung überprüft die Folge von transformierten Schaltbildern und entsprechenden Formeln.



**Abbildung 48. Lernerunterstützung in GETsoft mit mileET**

mileET ist auch als Aufgaben-Editor einsetzbar. Für die Aufgaben, zu denen keine intelligente Unterstützung seitens mileET gegeben werden kann, kann mileET als Lösungsentwurfeditor verwendet werden.

Ein Probeinsatz von mileET hat im Oktober 2003 an der TU Ilmenau in drei Übungen mit insgesamt 15 Studierenden stattgefunden. Die Studierenden empfanden z.B. die Funktionen zur sukzessiven Schaltbild-Transformation (z.B. Anwendung der Stern-Dreieck-Transformation) als sehr hilfreich. Besonders positiv wurde die Möglichkeit bewertet, die erstellten Formeln und Schaltbilder in einer freien Reihenfolge angeben und überprüfen lassen zu können. Die Möglichkeit, selbständig mittels Einschränkung der Hypothese nach eigenen Fehlern zu suchen, war den Studierenden aus gängigen Lernsystemen nicht bekannt. Diese Leistung von mileET bedarf einer besonderen Erläuterung, wurde jedoch von den Studierenden sehr gelobt. Einige Anregungen zu Usability der Lernumgebung wurden in der nachfolgenden Version von mileET berücksichtigt.

Eine umfassende Evaluation durch Studierende und Fachkollegen steht bevor.

Zu den Schwachpunkten von mileET gehört aus GET-Sicht die relativ geringe Anzahl von unterstützenden Themen (vgl. Abbildung 12). Einige Überlegungen bezüglich der möglichen Erweiterung des mileET-Ansatzes auf andere Themen sind im nächsten Kapitel gegeben.

Eine Reihe von Verfeinerungen der Benutzerschnittstelle sollen vorgenommen werden. Es fehlt z.B. der Zugriff auf die Zusatzparameter über den Formeleditor. Einige Parameter, wie z.B. Stromstärke, werden nicht angezeigt. Die Handhabung des Schaltungseeditors soll optimiert werden. Damit die mileET-Umgebung in die Lehre am Fachgebiet GET eingebettet werden kann, sollen zusätzlich zu den Verbesserungen der Benutzerschnittstelle das webbasierte Lernprogramm zum Thema „Methoden der Netzwerkberechnung“ erstellt und an die mileET-Umgebung angebunden werden.

## 6 Zusammenfassung und Ausblick

*Das Paradigma für die Gestaltung von hypermedialen Lernumgebungen hat sich gewandelt. Statt eine Anpassung des Systems an den Lerner anzustreben, bevorzugen Lernprogramm-Autoren offene Umgebungen, in denen der Lernende die nötigen Anpassungen selbst vornehmen kann...Optimal wären die Lernumgebungen, die es gestatten, dass Lernende im Bewusstsein ihrer individuellen Eigenarten sich an die Umgebung anpassen oder umgekehrt die Umgebung an sich und ihre Bedürfnisse anpassen können [Schulmeister2004].*

Die im Rahmen der vorliegenden Arbeit erreichten Ergebnisse (interaktive multimediale Lerneinheiten, CAS-Einsatz in GETsoft und in der Lehre, sowie wissensbasierte Problemlöseumgebung mileET) spiegeln diesen Paradigmawechsel sowie den Paradigmawechsel vom Lehren zum Lernen (Abschnitt 1.1) wieder.

Die Lernumgebung GETsoft stellt eine offene hypermediale Lernumgebung dar, die das Lernen nach konstruktivistischen Prinzipien ermöglicht. Auch der Wunsch nach einem Optimum hinsichtlich der Anpassung des Lernweges des Benutzers ist erfüllt: der Lernende kann den Vorgaben der Lernumgebung folgen und somit den Lernweg bestreiten, der von Lehrenden empfohlen wird, oder er kann seinen eigenen Weg verfolgen. Verschiedene Möglichkeiten, anspruchsvolle mathematische und technische Zusammenhänge interaktiv so zu gestalten, dass die Studierenden sich mit dem Lehrstoff aktiv auseinander setzen können, wurden implementiert und laut den Ergebnissen umfassender Evaluationen positiv bewertet. Interaktive Animationen und interaktive Module zur Antwortüberprüfung (Multiple-Choice, Zahlenwert-Vorsatz-Maßeinheit, spezielle Taschenrechner) sind implementiert worden und werden in mehreren Lernprogrammen der Lernumgebung GET benutzt.

Der Einsatz des Computeralgebrasystems Mathcad findet in Form eines Ingenieurwerkzeuges für Studierende und Lehrende statt.

Die vorgefertigten Mathcad-Arbeitsblätter sind analog den ausgearbeiteten Beispielen nach [Renkl2000]. Die Einführung in die Mathcad-Benutzung, die innerhalb dieser Beispiele gegeben wird, bereitet die Lernende auf die selbständige Benutzung des leistungsfähigen Ingenieurwerkzeuges vor. Vorbereitete Arbeitsblätter für AET-Übungen sind Beispiele einer Begleitung der mehrsemestrigen GET-Präsenzlehre und zugleich Bestandteile des umfangreichen eLearning-Angebotes des Fachgebietes.

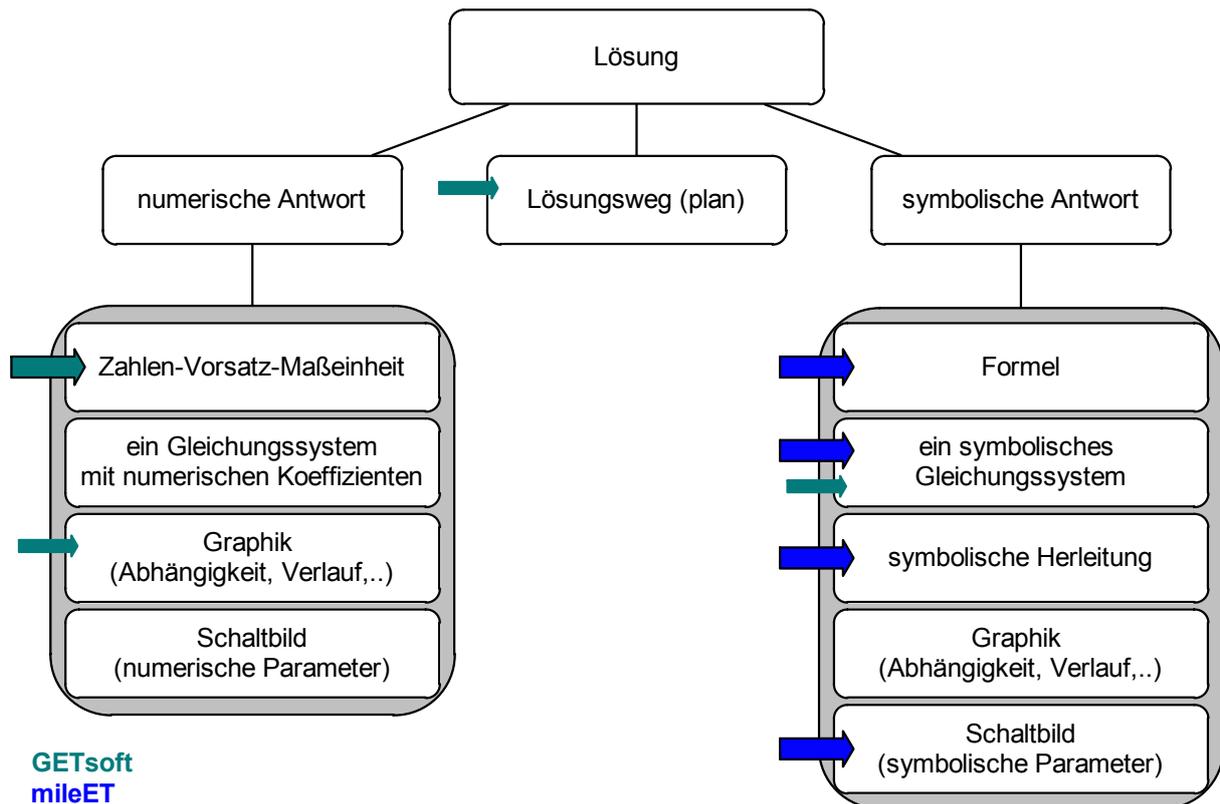
Die angestrebte Anpassungsfähigkeit und Adaptivität bzw. Adaptierbarkeit des Lernprogramms sind am weitesten in der intelligenten Problemlöseumgebung mileET realisiert.

Den Lehrenden wird die Erstellung von Aufgaben verschiedenen Komplexitätsniveaus zu festgelegten Themen der Grundlagen der Netzwerkanalyse ermöglicht. Die wissensbasierte Komponente entscheidet aufgrund ihres Meta-Wissens über Aufgabenziele und über GET-Wissen in mileET, ob die neu erstellten Aufgaben von ihr gelöst werden können.

Die Lernenden können aktiv und kooperativ mit und in mileET lernen. Sie können ihr vorhandenes Wissen und bestehende Fertigkeiten zum selbständigen Lösen dieser Aufgaben anwenden und in Stocksituationen verschiedene Heuristiken (u. a. das Hypothesentesten) einsetzen. Dank den speziell an die Lösungsmethoden orientierten Regeln besitzt die implementierte wissensbasierte Komponente in mileET die Fähigkeit, die Aufgaben so zu lösen, wie es den Studenten im Lehrbetrieb beigebracht wird. Somit kann das Programm die Rolle des Tutors übernehmen und Lösungen mit kommentierten Lösungsschritten erzeugen sowie partielle Lösungsentwürfe analysieren und adaptiv vervollständigen. Umfang und Flexibilität der Wissensbasis ermöglichen den Lernern außerdem, verschiedene und partielle Lösungswege zu erstellen und sie selbständig zu vergleichen.

Während die Nutzung der interaktiven Komponenten der multimedialen Lernumgebung GETsoft zusammen mit dem Mathcad-Einsatz zu einem bedeutenden Teil der Lehre am Fachgebiet geworden sind, wurde mileET in der ersten Version vorbereitet und muss noch einer kritischen Analyse durch Lehrende unterzogen werden. Die als Ergebnisse des umfassenden Testens durch Fachkollegen und durch Studierende zu erwartenden Anmerkungen und Empfehlungen müssen analysiert und berücksichtigt werden, bevor die Umgebung in die Präsenzlehre eingeführt werden kann.

Die folgende Abbildung fasst die Lösungskontrollmöglichkeiten der durch mileET erweiterten GETsoft-Lernumgebung zusammen. Die in GETsoft zur Verfügung stehenden Überprüfungsmöglichkeiten orientieren sich hauptsächlich auf die Überprüfung der numerischen Antworten (Zahlen-Vorsatz-Maßeinheit). Bei einzelnen Aufgaben werden z.B. ein symbolisches Gleichungssystem oder ein Lösungsplan kontrolliert. Es handelt sich dabei um so genannte Zuordnungstests – die richtige Reihenfolge von Parametern bzw. Variablen ist von den Autoren vorgegeben und gilt somit nur für diese konkrete Aufgabenstellung. Wichtig und interessant wäre, diese Überprüfungstechniken soweit zu verallgemeinern, dass keine Vorgabe des Lösungsweges durch den Autor notwendig wäre, damit verschiedene Aufgaben generiert und kontrolliert werden könnten.



**Abbildung 49. Lösungskontrollmöglichkeiten in GETsoft und mileET**

Einige Schritte in diese Richtung wurden mit mileET unternommen. Die wissensbasierte Komponente der Lernumgebung besitzt die Fähigkeit, symbolische Formeln, symbolische Gleichungssysteme und Schaltbilder mit symbolischen Parametern zu überprüfen. Die Lernumgebung braucht dafür keine Vorgabe der Lösungswege durch die Autoren. Sie ist in der Lage, die Aufgaben vom Aufgabenziel ausgehend selbstständig zu lösen. Das Schaltbild und das Aufgabenziel sollen in einem entsprechenden Format vorgegeben werden.

Die Entwicklung der Lernsoftware in letzten Jahren zeigt eine eindeutige Tendenz in Richtung webbasiertes Lernens [mehrere Publikationen auf der FIE von 1996 bis 2003, Batarseh2002, Salaverria2002]. Das Angebot reicht von PDF-Dokumenten und Präsentationsunterlagen (auch im Video-Format) bis zu interaktiven Simulationstools [Bayard2002, Rustom2004]. Im Rahmen der GETsoft-Lernumgebung wird das gesamte Themengebiet der Grundlagen der Elektrotechnik in interaktiven webbasierten Programmen umgesetzt. Diese Arbeit beinhaltet Unterstützung weiterer Themengebiete durch multimediale Experimentierumgebungen und Tools und die Erprobung neuer Technologien in der Erstellung neuer Lernprogramme. Die Problematik der Standardisierung der Lerneinheiten wird mit dem zunehmenden Wachstum der Lernangebote immer schärfer. Auch auf diesem Gebiet sind

bestimmte Maßnahmen vorzunehmen [Qu2002, Möbus2003], um auf dem internationalen Ausbildungsmarkt präsent zu sein.

Die im Rahmen des Projektes mile entstandene GETsoft-Struktur (TaskWeb, LearnWeb, BookWeb) soll durch eine LabWeb-Umgebung erweitert werden, in welcher einzelne Praktikumsversuche aus den Grundlagen der Elektrotechnik mit multimedialer Einleitung zur Bedienung der Messgeräte von Studierenden interaktiv durchgeführt werden könnten. Beispiele ähnlicher Realisierungen sind in [Svajger2003, Hodge2000] zu finden. Es ist für diesen Zweck zu untersuchen, welche Experimente und in welcher Form am sinnvollsten vorzubereiten sind [z.B. Nedic2003].

Ein weiterer Aspekt, welcher für Studierenden eine wichtige Rolle spielt, ist die Möglichkeit die Prüfungsfragen (Prüfungsaufgaben) selbstständig berechnen und Ergebnisse überprüfen lassen zu können. Für diesen Zweck bedarf es einer weiteren Umgebung im Rahmen von GETsoft – das TestWeb. Zwei Ideen eines Prüfungssystems für GET sind in [McGough2001] und in [Zahorian2001] vorgestellt. In beiden Fällen werden aus einer Menge von vorgefertigten Fragen- bzw. Aufgabenstellungen webbasierte Tests dynamisch generiert. Die Testfragen sind dabei Multiple-Choice Fragen oder Berechnungsaufgaben. Der Schwerpunkt der in [McGough2001] vorgestellten Arbeit liegt in der Verarbeitung von mathematischen Formeln online und in der Anwendung von WTML (Web Test Markup Language). Die Fragen-Sammlung liegt in Text-Dateien vor. Die Benutzerdaten und Testergebnisse werden mithilfe einer MySQL Datenbank verwaltet. Das Prüfungssystem (*Questioning System*) von [Zahorian2001] sieht eine intelligente tutorielle Unterstützung der Benutzer vor. Daher werden die Fragen nach mehreren Kriterien klassifiziert, wie Themenfeld und Position in dem Vorlesungsplan (*Content Area*), Schwierigkeitsgrad (*Difficulty Level*), Aufgabenart (*Category*), Anwendungsbereich (*Scope*), und gewichtet. Das System benutzt die Gewichtung der Aufgabenstellungen, um Fragen dem Bearbeitungszustand des Benutzers entsprechend zu kombinieren. Die Information über den Wissensstand des Benutzers wird in einem sogenannten *Knowledge Map* erfasst. Dank Klassifikation kann jede Aufgabenstellung im Falle der falschen Antwort mit den Lernmaterialien unterstützt werden. Als Berechnungstool benutzt das *Questioning System* die Ingenieurwerkzeuge wie Mathcad, MatLab und PSpice und die Fragen sind in einer MySQL Datenbank gespeichert.

Die Webbasierung der wissensbasierten Komponente mileET ist zurzeit wenig sinnvoll. Wünschenswert wäre es von der Domäne her die Entwurfsprinzipien von mileET auf andere Wissensgebiete der GET zu übertragen.

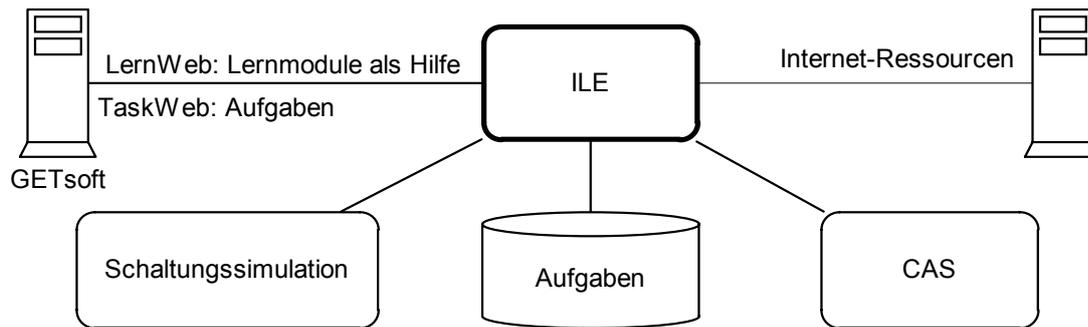
Die Tabelle 8 (vgl. Tabelle 1) illustriert Überlegungen zu Erweiterung der intelligenten Komponente der Lernumgebung auf weitere Themengebiete. Mit blauen Kreisen sind die Themen gekennzeichnet, deren wissenschaftliche Unterstützung eine Erweiterung der mileET (Wissensbasis) erfordert. Die roten Vierecke markieren Themengebiete zu denen andere als in mileET Benutzungsschnittstellen erforderlich wären, da Schaltungen in den Aufgabenstellungen (Aufgabenlösungen) zu diesen Themengebieten nicht vorhanden sind bzw. eine sekundäre Rolle spielen (z.B. Berechnung der magnetischen Größen, elektrisches Feld). Die Themen, deren einige Aufgaben sich über Analogien zu Schaltkreisen lösen lassen (z.B. thermische Vorgänge der Energieumformung, Ersatzschaltungen für technische Magnetkreise) sind mit weiß-blauen Kreisen markiert.

<b>Lehrgebiet Grundlagen der Elektrotechnik</b>		
1. Semester	2. Semester	3. Semester
  Grundbegriffe, Zweipole, Grundstromkreis	 Analyse von Wechselstromschaltungen	 Fourier-Reihen
  Methoden zur Netzwerkberechnung	 Frequenzselektive Schaltungen	 Laplace Transformation
  Energieumformungen	 Brückenschaltungen	 Vorgänge auf Leitungen
 Elektrisches Feld	  Transformator, Übertrager	 Erweiterung von mileET-WB sinnvoll  Anpassung der mileET-WB möglich
  Technischer Magnetkreis	  Drehstrom	 Erweiterung der GUI erforderlich  Entwicklung zu aufwändig

**Tabelle 8. Erweiterung der intelligenten Lernumgebung auf weitere Themenfelder**

Themengebiete im 3. Semester haben sehr komplexen mathematischen Hintergrund (die Lösungen bestehen größtenteils aus mathematischen Herleitungen), so dass aus heutiger Sicht eine intelligente Unterstützung der Lösungswege sehr aufwändig erscheint. Eine Wissensdiagnose-Komponente zu theoretischen Grundlagen dieser Domäne könnte der erste Schritt auf diesem Themenfeld sein.

Die erweiterte Version von mileET bräuchte eine Anbindung an ein CAS und ein Simulationstool. Solche CAS wie MuPAD, Maple, Mathematica oder Open-Source Lösungen sollen auf die Anbindungsmöglichkeit hin analysiert werden, als Simulationstool würde sich PSpice eignen (die Möglichkeit, das Programm als Server-Anwendung zu betreiben, wurde im Rahmen eines Praktikums erprobt).



**Abbildung 50. mileET als Teil einer multimedialen Lernumgebung**

Die Zusammenarbeit einer erweiterten (mileET-ähnlichen) intelligenten Komponente mit anderen Komponenten einer Lernumgebung illustriert die Abbildung 50.

Eine andere Arbeitsrichtung wäre die Mitarbeit an dem in [Harb2004] beschriebenen Projekt zur Entwicklung einer XML-basierten Sprache zur Beschreibung der elektrischen und elektronischen Schaltungen. Ein anderer Teil dieses Vorhabens war die Implementierung eines online Schaltungseditors mit Anbindung des PSpices als Server-Anwendung zur Schaltungsanalyse. Es scheint allerdings, dass die Arbeit an diesem Thema nicht durchgeführt wurde. Von der Arbeitsgruppe entstand eine Reihe von Java-Applets zur Analyse der Schaltungen der Leistungselektronik [Rustom2004].

Die Spice-basierte Sprache zusammen mit ansatzweise implementierten Daten-Strukturen innerhalb mileET bietet einen Ausgangspunkt für die „CirML“ oder „TaskML“-Richtung. Im Rahmen des mile Projektes entstandene mileET-Editoren bilden eine Basis für die netzbasierte Version einer Lernumgebung mit PSpice-Unterstützung.

Bei der inhaltlichen Vertiefung aus KI-Sicht wären andere Leistungen wünschenswert. Um z.B. dem Kriterium (4) von Tulodziecki<sup>67</sup> in einem weiteren Sinne zu entsprechen, müsste das System um eine Komponente zur Qualitätsbeurteilung von Lösungen erweitert werden. Den Studierenden könnte somit ein systemunterstützter Vergleich verschiedener Lösungen ermöglicht, sowie Vorschläge zur Optimierung ihrer Lösungen unterbreitet werden [Yakimchuk2004].

Eine bedeutende Rolle in der GET-Ausbildung spielen auch die Aufgaben, bei deren Lösungen die verbalen Beschreibungen und Erklärungen erwartet werden. Die computerbasierte Bearbeitung von solchen qualitativen Aufgaben mit tutoriellen Unterstützung erfordert eine natürlichsprachlichen Dialogkomponente.

<sup>67</sup> „...Lehren soll den Vergleich unterschiedlicher Lösungen ermöglichen sowie eine Systematisierung und Anwendung angemessener Kenntnisse und Vorgehensweisen sowie deren Weiterführung und Reflexion.“ [Tulodziecki2004]

## 7 Literaturverzeichnis

- [Altenkrüger1992] Altenkrüger, D., Büttner W., Wissensbasierte Systeme: Architektur, Entwicklung, Echtzeitanwendungen; eine praxisgerechte Einführung, in Reihe Künstliche Intelligenz, Vieweg, 1992, ISBN 3-528-05244-9
- [Anderson1986] Anderson, J.R., Knowledge Compilation: The General Learning Mechanism. In: R.S. Michalski, J.G. Carbonell, T.M. Mitchell, Machine Learning II. Kaufman, 1986, S. 289-310
- [Anderson1989] Anderson, J.R., A Theory of the Origins of Human Knowledge, Artificial Intelligence, 40, 1989, S. 313-351
- [Anderson1991] Anderson, J. R., Pelletier, R., A development system for model-tracing tutors, In Proceedings of the International Conference of the Learning Sciences, 1-8. Evanston, IL, 1991
- [Aslanski2002] Aslanski, K., Döring, N., Hessel, S., Try Outs von multimedialen Lernprogrammen am Beispiel der elektrotechnischen Grundlagenausbildung, 6. Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 26.-27.09.2002, ISSN 1436-4492, S. 85-88
- [Aspnes1994] Aspnes, J., A Summary and Analysis of Bachelor of Science Degree Requirements Reported by 125 Electrical Engineering Departments, IEEE Transactions on Education Vol. 37., No 2, May 1994, ISSN: 0018-9359
- [Atanov2002] Atanov, G.A., Pustynnikova, I.N, Education and Artificial Intelligence or Foundations of Modern Didactics of University Education (in Russian), Donezk, 2002, ISBN 966-8117-00-X
- [Azemi1996] Azemi, A., Using Multimedia Courseware in Engineering Education, 26th ASEE/IEEE Frontiers in Education Conference November 6-9, 1996, Salt Lake City, Utah, USA
- [Batarseh2002] Batarseh, I., Web-Based Delivery of the First Course in Power Electronics, 6. Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 26./27.09. 2002, ISSN 1436-4492, S.74-82
- [Baumgartner1997] Baumgartner, P., Didaktische Anforderungen an (multimediale) Lernsoftware, In: Information und Lernen mit Multimedia, Hg. von L. J. Issing und P. Klimsa. 2. überarbeitete. Aufl. Weinheim: Psychologie-Verl.-Union. 1997. S. 241-252.
- [Baumgartner1999] Baumgartner, P., Entwicklung interaktiver Lernumgebungen – Lerntheorie – Design – Didaktik, 3. Workshop Multimedia für Bildung und Wirtschaft im Rahmen 44th International Scientific Colloquium, TU Ilmenau, September 20.-23.1999, ISSN 1436-4492, S.5-18
- [Baumgartner2002] Baumgartner, P. et.al., e-Learning, CD Austria, Sonderheft des bm:bwk, 5/2002
- [Bayard2002] Bayard, J.-P.R., Shmarak, A.D., Nodal And Mesh Analyses For Dc Circuits: Interactive Problem-Solving Exercises And Tutorial Lectures, 32nd ASEE/IEEE Frontiers in Education Conference, November 6 - 9, 2002, Boston, MA
- [Beck1996] Beck, J., Stern, M., Haugsjaa, E., Applications of AI in Education, in Crossroads - The ACM Student Magazine, 1996, [www.acm.org/crossroads/xrds3-1/aied.html](http://www.acm.org/crossroads/xrds3-1/aied.html), letzter Zugriff 10.02.2005
- [Belski2003] Belski, I., Gray, C., Four-Screen Representation Of Electronic Systems, 33rd ASEE/IEEE Frontiers in Education Conference, November 5-8, 2003, Boulder, CO, USA
- [Booch1998] Booch, G., Object-oriented analysis and design: with applications, 1994, ISBN 0-8053-5340-2, 1998 (in russian) ISBN 5-7989-0067-3, 5-7940-0017-1
- [Berendt2003] Berendt, B., Werkstattseminar “Vom Lehren zum aktiven Lernen – Forschungs-orientierter Beitrag zur Praxis effektiver Hochschullehre”, In: WELBERS, U.: Hochschuldidaktische Aus- und Weiterbildung. W. Bertelsmann Verlag, Bielefeld 2003
- [Bratko1990] Bratko, Ivan: Prolog Programming for Artificial Intelligence, 2nd Edition, Addison-Wesley, 1990, ISBN 0-201-41606-9
- [Brooks1991a] Brooks, R. Intelligence Without Reason, IJCAI 91
- [Brooks1991b] Brooks, R.A., Intelligence without representation, AI 47 (1991)
- [Brown1982] Brown, J.S., Burton, D., Brown, J.S., Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III., In: Intelligent Tutoring Systems, Sleeman, D. and Brown, J.S. (eds), Academic Press, 1982, <http://www2.parc.com/spl/members/dekleer/>, letzter Zugriff 10.02.2005

- [Brusilovsky1998] Brusilovsky, P.: Adaptive educational systems on the World Wide Web. In: Ayala, G. (ed.), Proc. of Workshop "Current Trends and Applications of Artificial Intelligence in Education" at the 4th World Congress on Expert Systems, Mexico City, Mexico, ITESM (1998), pp. 9-16
- [Brusilovsky1999] Brusilovsky, P., Adaptive and Intelligent Technologies for Web-based Education, In C. Rollinger and C. Peylo (eds.), Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz, 4, 1999, pp. 19-25
- [Brusilovsky2001] Adaptive Hypermedia, in User Modeling and User-Adapted Interaction 11, 2001, pp.87-110
- [Burger/Finsterbusch1985] Burger, P.; Finsterbusch, H.: Ein Beitrag zur Einführung rechnerunterstützter Übungsplätze in der Ausbildung technischer Lehrgebiete. Dissertation, TH Ilmenau, 1985
- [Burger1997] Burger, P., Horn, W., Paradies, Th., Interaktive Lerndokumente im World Wide Web, 42. IWK, Technische Universität Ilmenau, 22.-25.09.1997
- [Butz2000] Butz, B.P., The Learning Mechanism Of The Interactive Multimedia Intelligent Tutoring System (IMITS), 30th ASEE/IEEE Frontiers in Education Conference, October 18 - 21, 2000 Kansas City, MO
- [Buedenbender2002] Buedenbender, J. et al., Using Computer Algebra Systems as Cognitive Tools, 6th. International Conference, ITS 2002, Biarritz, France and San Sebastian, Spain, June 2002, pp. 802 – 810
- [Canizares1997] Canizares, C.A., Faur, Z.T., Advantages and Disadvantages of Using Various Computer Tools in Electrical Engineering Courses, IEEE Transactions on Education, August 1997, Vol. 40 No.3, ISSN: 0018-9359
- [Chang1982] Chang, Sheldon S. L. [Hrsg.], Fundamentals handbook of electrical and computer engineering, Wiley, 1982 - 1983, B. 1 – 3, ISBN 0-471-86215-0, 0-471-86213-4, 0-471-86214-2
- [Coleman1998] Coleman, J.N., Effectiveness of Computer-Aided Learning as a Direct Replacement for Lecturing in Degree-Level Electronics, in IEEE Transactions on Education, August 1998, Vol. 41 No.3, ISSN: 0018-9359
- [Corbett2000] Corbett, A., Mclaughlin, M., Scarpinato, K.C., Modeling Student Knowledge: Cognitive Tutors in High School and College, User Modeling and User-Adapted Interaction 10, Kluwer Academic Publishers, 2000, pp.81-108
- [Coulon1993] Coulon, F. de, Forte, E., Rivera, J.M., KIRCHHOFF: An Educational Software for Learning the Basic Principles and Methodology in Electrical Circuits Modeling, IEEE Transactions on Education, February 1993, Vol. 36 No.1, ISSN: 0018-9359
- [Davidovic2003] Davidovic, A., Warren, J., Trichina, E., Learning Benefits of Structural Example-Based Adaptive Tutoring Systems, IEEE Transaction on Education, Vol. 46, No. 2, May 2003
- [Davis1993] Davis, R., Shrobe, H., Szolovitz, P., What Is a Knowledge Representation?, AI Magazine, Spring 1993
- [DeLysser1996] DeLysser, R.R., Using Mathcad in Electromagnetic Education, IEEE Transactions On Education, Vol. 39, No. 2, 1996, p. 198 – 210
- [DeKleer1984] Kleer, J. De, How Circuits Work, Artificial Intelligence, 1984, pp. 205-280
- [Devedzic2001] Devedzic, V., Knowledge Modeling - State of the Art, Integrated Computer-Aided Engineering, Vol.8, No.3, 2001, pp. 257-281 (<http://fon.fon.bg.ac.yu/~devedzic/>, letzter Zugriff 10.02.2005)
- [Dörr&Schrittmatter2002] Dörr, G., Schrittmatter, P., Multimedia aus pädagogischer Sicht, in Information und Lernen mit Multimedia und Internet, L.Issing, P.Klimsa (Hrsg.), Beltz Verlag, Weinheim, 2002, ISBN 3-621-27449-9, S. 29-42
- [Duden2001] Duden Informatik, Ein Fachlexikon für Studium und Praxis, 3. Aufl., 2001, ISBN 3-411-05233-3
- [Forsyth1987] Экспертная система: принципы работы и примеры. (Expert Systems: Principles and Case Studies. Edited by Richard Forsyth), 1987
- [Finsterbusch1990] Finsterbusch, H., Rettig, K., Wagner, E., Ansätze für ein wissensbasiertes Lehrsystem GET, 35.IWK, TH Ilmenau, 1990, S.119-122
- [Flor1996] Flor, T., Multi-Paradigmensysteme OOXSDAR VISIS, Ein Beitrag zur Entwicklung von Expertensystemen der zweiten Generation, Disertation, TU Ilmenau, 1996
- [Ford2001] Ford, R.M., Bondzie, J., Kitcho, P., Java Applets for Microelectronics Education, IEEE Transactions on Education, 2001

- [Fredrich2003] Fredrich, H., Knauf, U., Nitsch, J., Wollenberg, G., Webbasierte Präsentation elektrotechnischer Phänomene, 7. Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 25./26.09.2003, ISSN 1436-4492, S. 105-108
- [Garbe2003] Garbe, H., Yakimchuk, V., Möbus, C., Osterloh, J.-P., Thole, H.-J., Weber, L., Wagner, E., mileET – Knowledge Based Assistance for Electrical Engineering Education, 7.Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 25./26.09.2003, ISSN 1436-4492, S. 109-114
- [Gielen1994] Gielen, G., Wambacq, P., Sansen, W.M., Symbolic Analysis Methods and Applications for Analog Circuits: A Tutorial Overview, in Proceedings of the IEEE, Vol.82, No.2, February 1994
- [Goertzel1993] Goertzel, B., The structure of intelligence: a new mathematical model of mind, Springer 1993, ISBN 0-387-94004-9
- [Gollwitzer1990] Gollwitzer, P.M., Action Phases and Mind-Sets, in: E.T. Higgins & R.M. Sorrentino (eds), Handbook of Motivation and Cognition: Foundations of Social Behavior, 1990, Vol. 2, pp. 53-92
- [Gorny1991] Gorny, P., Software-ergonomische Umsetzung didaktischer Anforderungen an interaktive Lernsysteme. In: Audio Visuell Nr. 6/7: Neue Technologien - Gelungene und / oder mißlungene Medialisierung von Lernen?, Frankfurt, 1991, S.147-157
- [Greer1994] Student Modelling: The Key to Individualized Knowledge-Based Instruction, Ed. Jim E. Greer, Gordon I. McCalla, ISBN 3-540-57510-3, 0-387-57510-3
- [Haack2002] Interaktivität als Kennzeichen von Multimedia und Hypermedia. In: Issing, L.J., P. Klimsa (Hrsg.), Information und Lernen mit Multimedia und Internet, Weinheim, 2. Aufl., 2002, ISBN 3-621-27449-9, S.127-136
- [Hamann2003] Hamann, J., Kubichek, R., Pierre, J., Interactive Fourier Concepts: Computing Tools For Building Intuition, 33rd ASEE/IEEE Frontiers in Education Conference, November 5-8, 2003, Boulder, CO
- [Hammer1998] Hammer, S., Iakimtchouk, V., Wagner, E., Härle, F., Neundorf, V., Lehrprogramm: Fourier-Reihen, 2. Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 25.09.1998, ISSN 1436-4492, S. 55-60
- [Hampel1998a] Hampel, T., Keil-Slawik, Ferber, F., Multimedia darf nicht aus der Konserve kommen – neue Wege des Lernens verlangen neue Werkzeuge des Denkens und aktiven Gestaltens, in 2.Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 25.09.1998, ISSN 1436-4492, S. 55-60
- [Hampel1998b] Hampel, T., Ferber, F., MechANIma – ganzheitliche Sicht auf neue Medien in der Mechaniklehre, in Tagungsband GI Tagung Informatik und Ausbildung, 30. März-1. April 1998, Universität Stuttgart
- [Hanrahan1989] Hanrahan, H.E. Caetano, S.S., A knowledge-based aid for DC circuit analysis, IEEE Transactions on Education, Volume: 32, Issue: 4, Aug. 1989, ISSN: 0018-9359
- [Harb2004] Haarb, S., Batarseh, I., Teaching Electrical Circuit Analysis Using Web-Based simulation, in 49. Internationales Wissenschaftliches Kolloquium, Technische Universität Ilmenau, 27.-30.September 2004, ISBN 3-8322-2824-1, S. 463 - 467
- [Harger1996] Harger, R.O., Teaching in a Computer Classroom with a Hyperlinked, Interactive Book, IEEE Transactions On Education, Vol. 39, No. 3, 1996
- [Haun2000] Haun, M., Wissensbasierte Systeme, expert Verlag, 2000, ISBN 3-8169-1677-5
- [Heckhausen1989] Heckhausen, H., Motivation und Handeln, Heidelberg: Springer, 1989 (2nd ed.)
- [Hellmich1997] Hellmich, R., Einführung in die intelligenten Softwaretechniken. 1997, ISBN 3-8272-9546-7
- [Heulsman1996] Heulsman, L.P., Symbolic Analysis – A Tool For Teaching Undergraduate Circuit Theory, IEEE Transactions On Education, Vol. 39, No. 2, 1996, pp. 243-250
- [Hodge2000] Hodge, H., H. Scott Hinton, Lightner, M., Virtual Circuit Laboratory, 30th ASEE/IEEE Frontiers in Education Conference, October 18 - 21, 2000 Kansas City, MO
- [Holt1994] Holt, P., Dubs, S., Jones, M., Greer, J., The State Of Student Modeling, in Student Modeling: The Key to Individualized Knowledge-Based Instruction, Eds. J.E.Greer, G.I.McCalla, 1994, pp. 3-35
- [Horn1996] Horn, W., Burger, P., Interaktive und dynamische WWW-Lerndokumente mit VisualWave, 2. Workshop Multimediale Informations- und Kommunikationssysteme, Technische Universität Ilmenau, 17.-18.10.1996

- [Humar2003] Humar, I., Pustisek, M., Bester, J., Developing Dynamic Educational Material With Integrated Mathematical Notation For Web-Based E-Learning System, 33rd ASEE/IEEE Frontiers in Education Conference, 5.-8. November, 2003 Boulder, Co, ISBN 0-7803-7961-6, ISSN 0190-5848, p. T3F-19
- [Iakimtchouk2000a] Iakimtchouk, V., Wagner, E., Mathcad als Ingenieurwerkzeug in der multimedialen Lernumgebung, 4.Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 28./29.09.2000, ISSN 1436-4492, S. 63-70
- [Iakimtchouk2000b] Iakimtchouk, V., Wagner, E., Aslanski, K., Hammer, S., Multimedijnaja obuchajuschaja sreda “Osnovy Elektrotechniki” in Novye Informazionnyje Technologii v Elektrotechnicheskom Obrazovanii, NITE 2000, Astrakhan, Izd-vo ZNTEP, 2000 (Multimediale Lernumgebung Grundlagen der Elektrotechnik, in Neue Informationstechnologien in der elektrotechnischen Ausbildung, Astrakhan, Russland, 2000), ISBN 5-89388-035-8
- [Issing2002] Studieren mit Multimedia und Internet – Ende der traditionellen Hochschule oder Chance zur Innovation?, 6. Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 26./27.09. 2002, ISSN 1436-4492, S.1-18
- [Issing&Klimsa2002] Issing, L.J., P. Klimsa (Hrsg.), Information und Lernen mit Multimedia und Internet, Weinheim, 2. Aufl., 2002, ISBN 3-621-27449-9
- [Ionkin1976] Теоретические основы электротехники. Т.1, 2. Под ред. П.А. Ионкина. Учебник для электротехнических вузов. Москва «Высшая школа», 1976 (Theoretische Grundlagen der Elektrotechnik, Bände 1,2, Ionkin, P.A., et al. Lehrbuch für elektrotechnische Hochschulausbildung, Moskau, 1976)
- [Jarz1997] Jarz, E., Entwicklung multimedialer Systeme, Deutscher Universitäts- Verlag, 1997, ISBN 3-8244-6523-X
- [Jordan2003] Jordan, P. W., Makatchev, M., Venlehn, K., Abductive Theorem Proving for Analyzing Student Explanations, in Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies, Eds: U. Hoppe, F. Verdejo, and J. Kay, IOS Press Ohmsha, 2003, ISSN 0922-6389
- [Justus1998] Justus, O., Elektrische Netzwerke mit PSpice, Fachbuchverlag Leipzig im Carl Hanser Verlag München Wien, 1998, ISBN 3-446-19387-1
- [Keil-Slawik2000] Keil-Slawik, R., Novaczyk, O., Von der geschlossenen Multimediaproduktion zur offenen Lernumgebungen, in Campus 2000 : Lernen in neuen Organisationsformen, F. Scheuermann (Hrsg.), Waxmann Verlag, Münster, 2000, ISBN 3-89325-925-2, S. 265-277
- [Kerres1998] Kerres, M., Multimediale und telemediale Lernumgebungen: Konzeption und Entwicklungen, Oldenbourg Verlag, 1998, ISBN 3-486-24539-2
- [Kerres2002] Kerres, M., Technische Aspekte multi- und telemedialer Lernangebote, in: Issing, L.J., P. Klimsa (Hrsg.), Information und Lernen mit Multimedia und Internet, Weinheim, 2. Aufl., 2002, ISBN 3-621-27449-9, S.19-27
- [Khan2001] Khan, M.S.Z., Smith,C.E., Kajfez, D., NASWAVE—A Program for Display of Standing Waves on Network Analyzers, IEEE Transactions On Education, Vol. 44, No. 2, May 2001, pp.151-157
- [Kuipers1994] Kuipers, B., Qualitative Reasoning: modeling and simulation with incomplete knowledge, MIT, 1994, ISBN 0-262-11190-X
- [Kuznetzov1997] Kuznetzov, E.V., Iakimtchouk, V., Lernsoftware für Grundlagen der Elektrotechnik am MEI, Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 28.11.1997, S. 21-23
- [Larkin-Hein2001] Larkin-Hein, T., Budny, D., Research on Learning Style: Applications in the Physics and Engineering Classrooms, IEEE Transactions On Education, Vol. 44, No. 3, August 2001, pp.276-281
- [Lenzen2002] Lenzen, M., Natürliche und Künstliche Intelligenz: Einführung in die Kognitionswissenschaft, Campus Verlag, 2002, IABN 3-593-37033-6
- [LfSW1995] Landesinstitut für Schule und Weiterbildung - Beratungsstelle für neue Technologien: Zur Prüfung neuer Medien für den Unterricht in Nordrhein-Westfalen, Werkstattbericht Nr.7, Soest 1995.
- [Liu2002] Liu, C. et al., WebMath: A Web-Based ITS System, Advances in Web-Based Learning, First International Conference, ICWL 2002, Hong Kong, China, August 2002, pp.119 – 129
- [Löscher1993] Löscher, J., Menzel, U., Künstliche Intelligenz: ein Handwörterbuch für Ingenieure, VDI-Verlag, Düsseldorf 1993, ISBN 3-18-401309-X

- [Luchetta2001] Luchetta, A., Manetti, S., Reatti, A., SAPWIN—A Symbolic Simulator as a Support in Electrical Engineering Education, *IEEE Transactions On Education*, Vol. 44, No. 2, May 2001.
- [Lüdtke2002] Lüdtke, A., Möbus, C., Thole, H.-J., Cognitive Modelling Approach to Diagnose Over-Simplification in Simulation-Based Training, in: St.A. Cerri, G. Gouarderes & F. Paraguacu (eds), *Intelligent Tutoring Systems, Proceedings of the 6th International Conference, ITS2002, Biarritz, France and San Sebastian, Spain, June, 2002*, S. 496 - 506, Berlin: Springer, Lecture Notes in Computer Science, LNCS 2363, ISBN 3-540-43750-9
- [McGough2001] McGough, J., Mortensen, J., Johnson, J., Fadali, S., A web-based testing system with dynamic question generation, *31st ASEE/IEEE Frontiers in Education Conference*, October 10–13, 2001 Reno, NV
- [Medienprojekt2003] Haubold, T., Dippel, H.C., Entwicklung eines webbasierten Forums zum Austausch von Formeln auf Basis der natürlichen Schreibweise, *Medienprojekt MP2181/03/13*, TU Ilmenau 2003
- [Metz1998] Metz, D., Naundorf, U., Schlabbach, J., *Kleine Formelsammlung Elektrotechnik*, 1998, ISBN 3-446-19392-9
- [Millard2003] Millard, D., Burnham, G., Increasing Interactivity in Electrical Engineering, *33rd ASEE/IEEE Frontiers in Education Conference*, November 5-8, 2003, Boulder, CO, USA
- [Möbus1995] Möbus, C., Towards an Epistemology on Intelligent Problem Solving Environments: The Hypothesis Testing Approach, in J. GREER (ed), *Proceedings of AI-ED 95, World Conference on Artificial Intelligence and Education*, Washington, DC, Association for the Advancement of Computing in Education (AACE), 1995, pp.138 – 145.
- [Möbus1996] Möbus, C., Towards an Epistemology on Intelligent Problem Solving Environments: The Hypothesis Testing Approach, in P. BRNA, A. PAIVA, J. SELF (eds), *Proceedings of EuroAIED96, European Conference on Artificial Intelligence in Education*, Lisbon, Portugal, 1996.
- [Möbus2002] Möbus, C. et al., Towards a Specification of Distributed and Intelligent Web Based Training Systems, in: St. A. CERRI, G., GOUARDÈRES, F. PARAGUACU (eds), *Intelligent Tutoring Systems*, 291 – 300, Berlin: Springer, 2002, ISBN 3-540-43750-9
- [Möbus2003a] Möbus, C. et al., Towards an AI-Specification of Intelligent Distributed Learning Environments, *KI - Zeitschrift Künstliche Intelligenz Heft 1/03 "Schwerpunkt: Lernen: Modellierung und Kommunikation"*, Bremen: arendtap Verlag, 2003, ISSN 0933-1875, S. 19-24
- [Möbus2003b] Möbus, C. et al., Zur SCORM-Fähigkeit konventioneller XML-unterstützender eLearning Developer Tools im Projekt I-can-EIB, *7. Workshop „Multimedia für Bildung und Wirtschaft“*, TU Ilmenau, 25./26.09.2003, ISSN 1436-4492
- [Mora2003] Mora, M.A., Moriyon, R., Saiz, F., Building Mathematics Learning Applications By Means Of ConsMath, *33rd ASEE/IEEE Frontiers in Education Conference*, 5.-8. November, 2003 Boulder, Co, ISBN 0-7803-7961-6, ISSN 0190-5848
- [Murray1999] Murray, T., *Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art*, *Int.J. of AI*, 1999, Vol.10, pp. 98-129
- [Mülchen2001] Mülchen R., *Kategorienbildung im Bereich Lernsoftware*, Diplomarbeit, Universität Oldenburg, 2001
- [Nedic2003] Nedic, Z., Machotka, J., Nafalski, A., Remote Laboratories Versus Virtual And Real Laboratories, *33rd ASEE/IEEE Frontiers in Education Conference*, November 5-8, 2003, Boulder, CO
- [Neundorf2003] Neundorf, V., Hammer, S., Wagner, E., GET.netz – e-Learning Netzwerk für die Ausbildung in Grundlagen der Elektrotechnik, *7. Workshop Multimedia für Bildung und Wirtschaft*, Technische Universität Ilmenau, 25.-25. September 2003, ISSN 1436-4492.
- [Newell1982] Newell, A., *The Knowledge Level*, *Artificial Intelligence* 18, 1982, pp. 87-127
- [Newell1990] Newell, A., *Unified Theories of Cognition*, Cambridge, Mass.: Harvard University Press, 1990
- [Nichols2003] Nichols, M., A theory for eLearning, *Educational Technology & Society*, 6(2), 2003, pp.1-10, available at <http://ifets.ieee.org/periodical/6-2/1.html>, letzter Zugriff 10.02.2005
- [Niegemann2004] Niegemann, H. et. al., *Kompendium e-Learning*, Springer, 2004, ISBN 3-540-43816-5
- [Nigim2001] Nigim, K.A., DeLyser, R.R., Using Mathcad in Understanding the Induction Motor Characteristics, *IEEE Transactions On Education*, Vol. 44, No. 2, May 2001

- [Nilsson1998] Nilsson, N.J., *Artificial Intelligence A New Synthesis*, Morgan Kaufmann Publishers, Inc., San Francisco Californien, 1998, ISBN 1-55860-467-7
- [NITE98] Materialy 4. Mezhdunarodnoj Nauchno-Metodicheskoy Konferenzii Novyje Informationnyje Technologii v Prepodavanii Elektrotechnicheskikh Disziplin (NITE-98), 21.-26.09.1998 Astrakhan, Russland
- [O'Keefe1990] O'Keefe, Richard A: *The Craft of Prolog*, 1990 Massachusetts Institute of Technology, ISBN 0-262-15039-5
- [Paraskakis2002] Paraskakis, I., *ToLeDe: Towards Creating an Intelligent Computer Algebra System*, 6th. International Conference, ITS 2002, Biarritz, France and San Sebastian, Spain, June 2002, pp. 410– 420
- [Pillage1993] Pillage, L.T., *An Early introduction to Circuit Simulations Techniques*, IEEE Transactions on Education, February 1993, Vol. 36, No.1, ISSN: 0018-9359
- [Qu2002] Qu, C., Nejdil, W., *Towards Open Standards: The Evolution of an XML/JSP/WebDAV Based Collaborative Courseware Generating System*, Advances in Web-Based Learning, First International Conference, ICWL 2002, Hong Kong, China, August 2002, p.167 – 179
- [Reeves1997] Reeves, T., *Evaluating what really Matter in PC-based Education*, available at <http://www.educationau.edu.au/archives/cp/reeves.htm>, letzter Zugriff 10.02.2005
- [Renkl2000] Renkl, A., Atkinson, R. K., & Maier, U. H. (2000). *From studying examples to solving problems: Fading worked-out solution steps helps learning*. In L. Gleitman & A. K. Joshi (Eds.), *Proceeding of the 22nd Annual Conference of the Cognitive Science Society*, pp. 393-398, Mahwah, NJ: Erlbaum.
- [Rettig1997], Rettig, K.: *Ansätze zur Realisierung intelligenter Lehrsysteme in der universitären Ausbildung. Eine interaktive Lernumgebung im Fachgebiet „Grundlagen der Elektrotechnik“ der TU Ilmenau*. Dissertation, TU Ilmenau, 1997
- [Riehm1995] Riehm, U., Wingert, B., *Multimedia: Mythen, Chancen und Herausforderungen*, Bollmann Verlag, Mannheim, 1996, S.145 - 195
- [Romero1997] Romero J.J.F, *Reasoning about linear circuits in sinusoidal steady state*, Dissertation, Department of Computer and Information Science, University of Oregon, August 1997
- [Rose2000] Rose, C. P., Moore, J.D., VanLehn, K., Allbritton, D., *A Comparative Evaluation of Socratic versus Didactic Tutoring*, 2000
- [Russel1995] Russell, S. J., Norvig, P., *Artificial intelligence: a modern approach*, Upper Saddle River [u. a.] : Prentice Hall, 1995, ISBN: 0-13-103805-2, 0-13-360124-2
- [Rustom2004] Rustom, K., Batarseh, I., Kalaldiah, K., *Interactive Power Electronic Applets for Educational Delivery*, in 49. Internationales Wissenschaftliches Kolloquium, Technische Universität Ilmenau, 27.-30.September 2004, ISBN 3-8322-2824-1, S. 419 - 426
- [Salaverria2002] Salaverria, A., Moure, M.J., Valdes, G., Mandado, E., *SHIVA: A Hypermedia System For Electronics Learning*, 32nd ASEE/IEEE Frontiers in Education Conference, November 6-9, 2002, Boston, MA, USA
- [Salmon2004] Salmon, D., Jones, M., *Higher Education Staff Experience of Using Web-Based Learning Technologies*, *Educational Technology & Society*, 7(1), pp. 107-114
- [Shank1998] Shank, R. C., *Inside Multi-Media Case Based Instruction*, 1998, ISBN 0-8058-2537-1
- [Scherp2002] Scherp, A., Schlattman, W., Heuten, W., Kuszewski, R., *Virtuelle Labore für das E-Learning in Naturwissenschaftlichen Studienfächern am Beispiel der Gentechnik*, 6.Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 26./27.09. 2002, ISSN 1436-4492, S. 41-46
- [Schröder1992] Schröder, O., Möbus, C., *Zur Modellierung des hilfegeleiteten Wissenserwerbs beim Problemlösen*, in K. REISS, M. REISS, H. SPANDL (Hrsg), *Maschinelles Lernen - Modellierung von Lernen mit Maschinen*, Berlin, Heidelberg: Springer, 1992, S. 23-62
- [Schulmeister1996a] Schulmeister, R., *"Grundlagen hypermedialer Lernsysteme - Theorie, Didaktik, Design"*, Addison-Wesley, 1996. ISBN 3-486-24419-1
- [Schulmeister1996b] Schulmeister, R., *Keine Angst vor Statistik! Computerunterstütztes Lernen der Statistik mit einem Programm für Psychologen und Sozialwissenschaftler*, in *Handbuch Hochschullehre Highlights; Band 2, Neue Medien in der Hochschullehre: Perspektiven, Modelle, Methoden*, Raabe Fachverlag, 1996, ISBN 3-88649-345-8, B1.11

- [Schulmeister2004] Schulmeister, R., Didaktisches Design aus hochschuldidaktischer Sicht – Ein Plädoyer für offene Lernsituationen, in: U. Rinn, D.M. Meister (Hrsg.), Didaktik und Neue Medien: Konzepte und Anwendungen in der Hochschule, Münster: Waxmann, 2004, ISBN 3-8309-1216-1, S.19-49
- [Schulze2000] Schulze, K.G. et.al., ANDES: An Intelligent Tutor for Classical Physics, The Journal of Electronic Publishing, September, 2000, Vol. 6, Issue 1, ISSN 1080-2711, siehe auch [Andes1]
- [Seidel1999] Seidel, H.-U., Wagner, E., Allgemeine Elektrotechnik, Band 1, Carl Hanser Verlag 1999, ISBN 3-446-21339-2
- [Seidel2001] Seidel, H.-U., Wagner, E., Allgemeine Elektrotechnik, Band 2, Carl Hanser Verlag 2001, ISBN 3-446-21513-1
- [Seeberg2003] Seeberg, C., Life Long Learning: modulare Wissensbasen für elektronische Lernumgebungen, Springer 2003, ISBN 3-540-43618-9
- [Sterling1988] Sterling, L.; Shapiro, E.: Prolog – Fortgeschrittene Programmieretechniken, Addison-Wesley Verlag GmbH, Bonn, 1988, ISBN 3-925118-84-5
- [Strittmatter2000] Strittmatter, P., Niegemann, H., Lehren und Lernen mit Medien, 2000, ISBN 3-534-12346-8
- [Stubenrauch1989] Stubenrauch, R.: Interaktives rechnergestütztes Übungsprogramm für den Einsatz im Lehrgebiet GET an der TH Ilmenau. Dissertation, TH Ilmenau, 1989
- [Svajger2003] Svajger, J., Valencic, V., Discovering Electricity by Computer-Based Experiments, IEEE Transactions On Education, Vol. 46, No. 4, November 2003, p.502-507
- [Sweller1999] Sweller, J., Instructional design in technical areas, ACER Press, 1999, ISBN 0-86431-312-8
- [Thole1997] Thole, H.-J., Möbus, C., Schröder, O., Domain Knowledge Structure, Knowledge Representation and Hypotheses Testing, in: B. Boulay, R. Mizoguchi (eds.): Artificial Intelligence in Education, Amsterdam: IOS Press, 1997.
- [Trueman2003] Trueman, C.W., Animating Transmission-Line Transients With BOUNCE, IEEE Transactions On Education, Vol. 46, No. 1, February 2003, pp.115-123
- [Tulodziecki2004] Tulodziecki, G., Herzig, B., Allgemeine Didaktik und computerbasierte Medien, in: U. RINN, D.M. MEISTER (Hrsg.), Didaktik und Neue Medien: Konzepte und Anwendungen in der Hochschule, Münster: Waxmann, 2004, ISBN 3-8309-1216-1, S.50-71
- [Tuttas2000] Tuttas, J. Elemente einer Didaktik netzbasierter Lernumgebungen für die ingenieurwissenschaftliche Ausbildung, in Campus 2000 : Lernen in neuen Organisationsformen, F. Scheuermann (Hrsg.), Waxmann Verlag, Münster, 2000, ISBN 3-89325-925-2, S. 381-389
- [Vanlehn1988] Vanlehn, K., Toward a Theory of Impasse-Driven Learning, in H. Mandl, A. Lesgold (eds), Learning Issues for Intelligent Tutoring Systems, Berlin: Springer, 1988, 19-41
- [Vanlehn2000] VanLehn, K. et al., Fading and deepening: The next steps for Andes and other model-tracing tutors, Intelligent Tutoring Systems: 5th International Conference, Montreal, Canada, Gauthier, Frasson, VanLehn (eds), Springer (Lecture Notes in Computer Science, Vol. 1839), 2000, pp. 474-483
- [VELK99] Materialy Vserossijskogo Elektrotechnicheskogo Kongressa s Mezhdunarodnym Uchastiem VELK-99, 28.06-03.07.1999 Astrakhan, Rußland
- [Vidal1996] Vidal, O. de la S. et al., Interaction and Simulation-Based Modules For Electromagnetic Education, 26th ASEE/IEEE Frontiers in Education Conference November 6-9, 1996, Salt Lake City, Utah, USA
- [Virvou2001] Virvou, M., Moundridou, M., Student and Instructor Models: Two Kinds of User Model and their Interaction in an ITS Authoring Tool, in M.Bauer, P.J., Gmytrasiewicz, and J.Vassileva (Eds.) UM2001, pp. 158-167
- [Wagner1997] Wagner, E., Höhne, G., Multimediale Lernumgebungen für die Elektrotechnik und Konstruktionstechnik, Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 28.11.1997, S. 4-20
- [Wagner 1998] Wagner, E., Hammer, S., Iakimtchouk, V., Multimedia Lernumgebung Grundlagen der Elektrotechnik – Konzeption, Probleme, Beispiele, 2. Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 25.09.1998, ISSN 1436-4492, S. 41-44
- [Wagner 1999] Wagner, E., Iakimtchouk, V., Aslanski, K., Hammer, S., Seidel, H.-U., Wykowski, S., Multimedia Lernumgebung Grundlagen der Elektrotechnik – Fortschritte bei Realisierung und Einsatz, 3. Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 23./24.09.1999, ISSN 1436-4492, S. 81-86

- [Wenger1987] Wenger, E, Artificial Intelligence and Tutoring Systems, 1987, ISBN 0-934613-26-5
- [Wester1999] Wester, M., A Critique of the Mathematical Abilities of CA Systems, in Computer Algebra Systems: a Practical Guide, UK, 1999, ISBN 0-471-98353-5,  
([http://www.math.unm.edu/~wester/cas\\_review.html](http://www.math.unm.edu/~wester/cas_review.html), letzter Zugriff 10.02.2005)
- [Willms1997] Willms, J., Göhler, H., Möbus, C., Testing Hypotheses in an Engineering Domains: Combining Static and Dynamic Analysis of Pneumatic Circuits, AIED 1997
- [Willms2002] Willms, J., Konzeption einer intelligenten Problemlöseumgebung für die Patentanmeldung und –prüfung, Dissertation (Dr. rer. nat.), Fachbereich Informatik, Carl von Ossietzky Universität Oldenburg, 2002
- [WK1996] Wörterbuch der Kognitionswissenschaft, hrsg. von Gerhard Strube, Stuttgart : Klett-Cotta, 1996, ISBN 3-608-91705-5
- [Wood1996] Wood, S.L., A new Approach to Interactive Tutorial Software for Engineering Education, IEEE Transaction on Education, Vol.39, No. 3, August 1996
- [Wucherer2002] VDE-Grundpositionen zur Ingenieurausbildung, von Prof. Dr.-Ing. Klaus Wucherer, VDE-Präsident und Mitglied des Zentralvorstandes der Siemens AG, Vortrag beim Kolloquium des Hochschullehrerverbandes am 13. Juni 2002 ([www.vde.de](http://www.vde.de))
- [xml2004] Das XML-Kompodium in iX spezial, 1/2004
- [Yakimchuk2002] Yakimchuk, V., Garbe, H., Möbus, C., Thole, H.-J., Eine intelligente Problemlöseumgebung für die Grundlagen der Elektrotechnik, 6.Workshop „Multimedia für Bildung und Wirtschaft“, TU Ilmenau, 26./27.09. 2002, ISSN 1436-4492, S. 67.-72., S. 92-93
- [Yakimchuk2003] Yakimchuk, V., Garbe, H., Möbus, C., Thole, H.-J., Wagner, E., An Intelligent Problem Solving Environment in the Domain of Electrical Engineering., in Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies, Eds: U. Hoppe, F. Verdejo, and J. Kay, IOS Press Ohmsha, 2003, ISSN 0922-6389
- [Yakimchuk2004] Yakimchuk, V., Garbe, H., Thole, H.J., Möbus, C., Wagner, E., mileET: Problemorientiertes Lernen in einer wissensbasierten und adaptiven Lernumgebung für die Grundlagen der Elektrotechnik, in: Klaus Rebenburg (Hrsgb), Grundfragen Multimedialen Lehrens und Lernens, 2. Workshop GML2 2004, Alcatel SEL Stiftung für Kommunikationsforschung, Norderstedt: Books on Demand, 2004, ISBN 3-8334-1573-8, S. 73 - 84
- [Yoshikawa1992] Yoshikawa, A., Shintani, M., and OHBA, Y., Intelligent Tutoring Systems for Electric Circuit Exercising, IEEE Transactions on Education, August 1992, Vol. 35, No.3, ISSN: 0018-9359
- [Zahorian2001] Zahorian, S. A. et. al., Question model for intelligent questioning systems in engineering education, 31<sup>st</sup> ASEE/IEEE Frontiers in Education Conference, October 10 - 13, 2001 Reno, NV

**WWW-Adressen** (letzter Zugriff 10.02.2005)

- [ActiveMath] <http://www.ActiveMath.org>
- [Anderson] <http://act-r.psy.cmu.edu/people/ja/>
- [AIED] <http://aied.inf.ed.ac.uk/>
- [AIED2003] <http://www.cs.usyd.edu.au/~aied/>
- [Andes1] <http://www.press.umich.edu/jep/06-01/schulze.html>
- [Andes2] <http://www.andes.pitt.edu/>
- [Beats Biblionetz] <http://beat.doebe.li/bibliothek/>
- [Brusilovsky] <http://www2.sis.pitt.edu/~peterb/>
- [CAS] <http://www.scientificweb.de/science.html>
- [CircuitMagic] [www.circuit-magic.com](http://www.circuit-magic.com)
- [CircuitMaker] <http://www.microcode.com/>, <http://www.altium.com/circuitmaker/>
- [CiteSeer] <http://citeseer.ist.psu.edu/>
- [DeKleer] <http://www2.parc.com/spl/members/dekleer/>

- [eLearning NMB] <http://193.25.43.240/nmb-workshop>
- [Electronics Workbench] <http://www.electronicworkbench.com/>
- [FIE] Frontiers in Education Homepage, <http://fie.engrng.pitt.edu/>
- [GETsoft] <http://www.getsoft.net/>
- [IJAIED] International Journal of Artificial Intelligence in Education, <http://aied.inf.ed.ac.uk/>
- [I@LLS] <http://lls.informatik.uni-oldenburg.de>
- [IEEE] <http://www.ieee.org>
- [JEP] <http://www.singularsys.com/jep/>
- [KI] <http://www.kuenstliche-intelligenz.de/>
- [Lerntheorien] <http://tip.psychology.org/>
- [lpa] <http://www.lpa.co.uk>
- [Macromedia] <http://www.macromedia.com/>
- [maple] <http://www.maplesoft.com>
- [Mathcad] [www.mathsoft.com](http://www.mathsoft.com), [www.softline.de](http://www.softline.de)
- [Mathematica] <http://www.wolfram.com/>
- [MathML] <http://www.w3.org/math>
- [MatLab] <http://www.mathworks.com/>
- [MEI] <http://www.mpei.ru/>
- [Multisim] <http://www.electronicworkbench.com/>
- [MuPAD] <http://www.mupad.de>, [www.mathpad.de](http://www.mathpad.de)
- [needs] <http://www.needs.org>
- [OFFIS] [www.offis.de](http://www.offis.de)
- [OpenMath] <http://www.openmath.org>
- [PSpice] <http://www.orcad.com/>
- [SAC] <http://www.symbolicnet.org/>
- [SCORM] <http://www.adlnet.org/>
- [ToolBook] <http://www.sumtotalsystems.com/index.html>
- [VanLehn] <http://www.pitt.edu/~vanlehn>
- [VDE] <http://www.vde.de>
- [XML] <http://www.w3.org/XML>

## **8 Anhang**

A1. Signalfunktionen und Netzwerke der Experimentierumgebung Fourier-Reihe

A2. Beispiele von Experimentierumgebungen und Hilfestellungen in GETsoft

A3. Beispiele von Mathcad-Arbeitsblättern

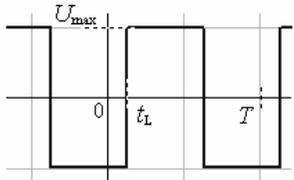
A4. Aufgabenziele in mileET

A5. Datenstrukturen für Objekte in der mileET-Wissensbasis

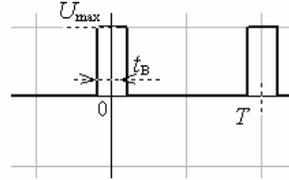
A6. Lernumgebung mileET: Screenshots

A7. Kurze Bedienungsanleitung zu mileET

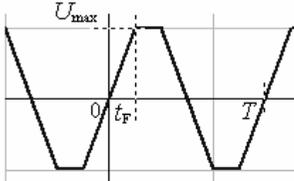
Rechteckfunktion



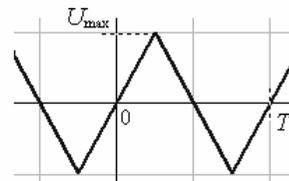
Rechteckimpulsfolge



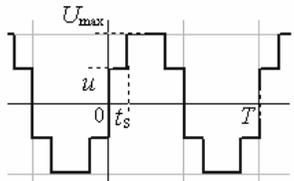
Trapezfunktion



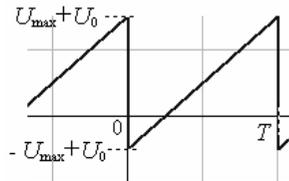
Dreieckfunktion



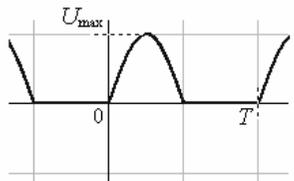
Treppenfunktion



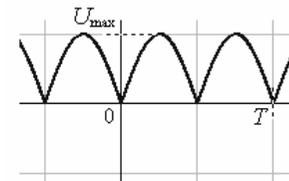
Sägezahnfunktion



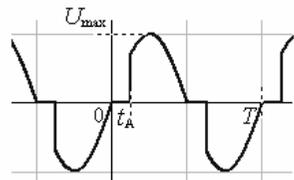
Einweggleichrichtung



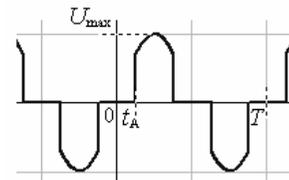
Zweiweggleichrichtung



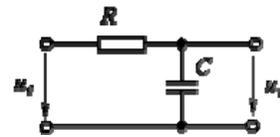
Unsymmetrischer Phasenanschnitt



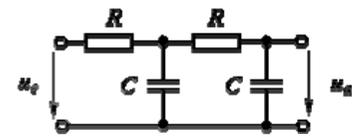
Symmetrischer Phasenanschnitt



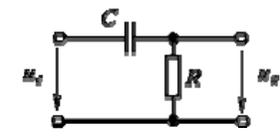
RC-Tiefpass



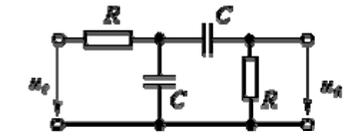
Doppelter RC-Tiefpass



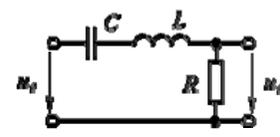
RC-Hochpass



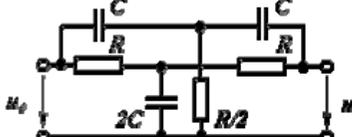
RC-Bandpass



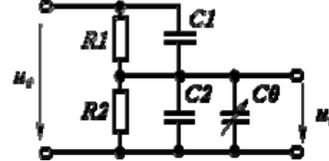
Reihenschwingkreis

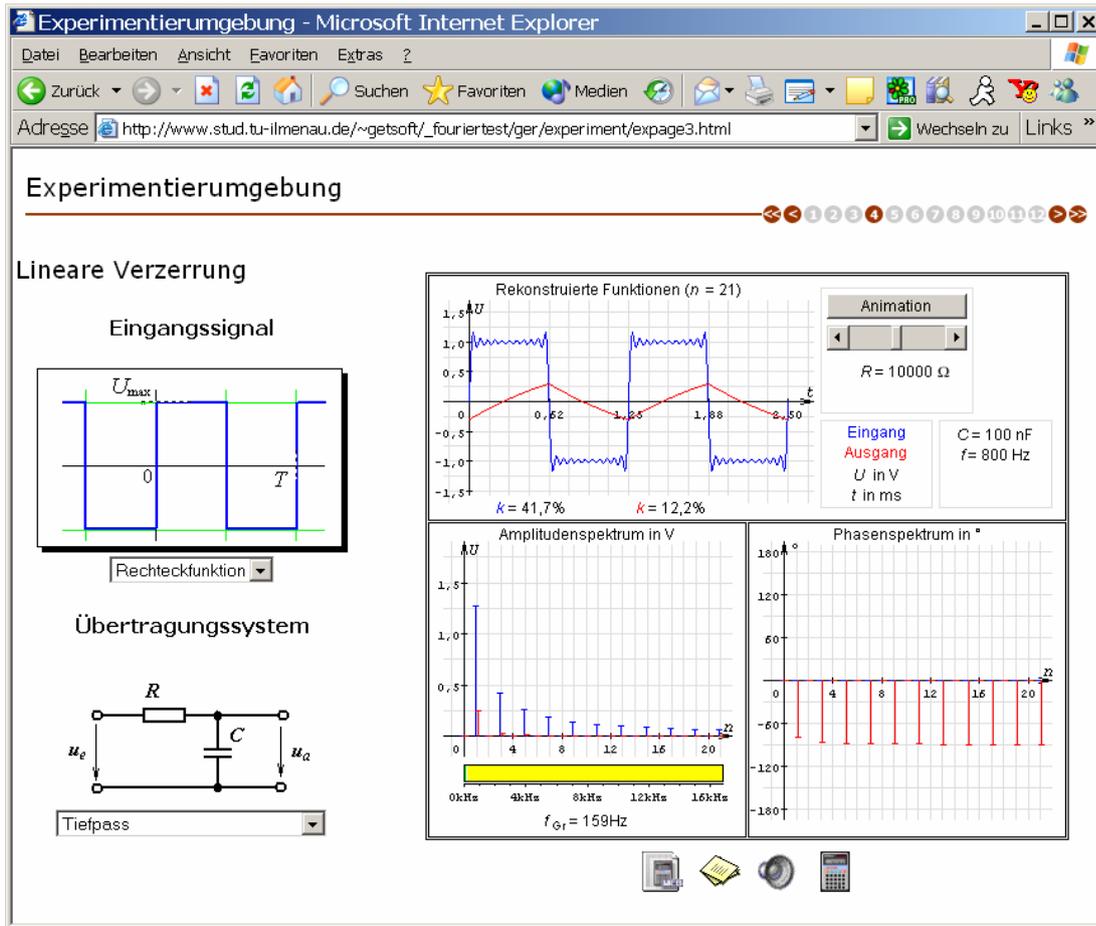


RC-Bandsperre

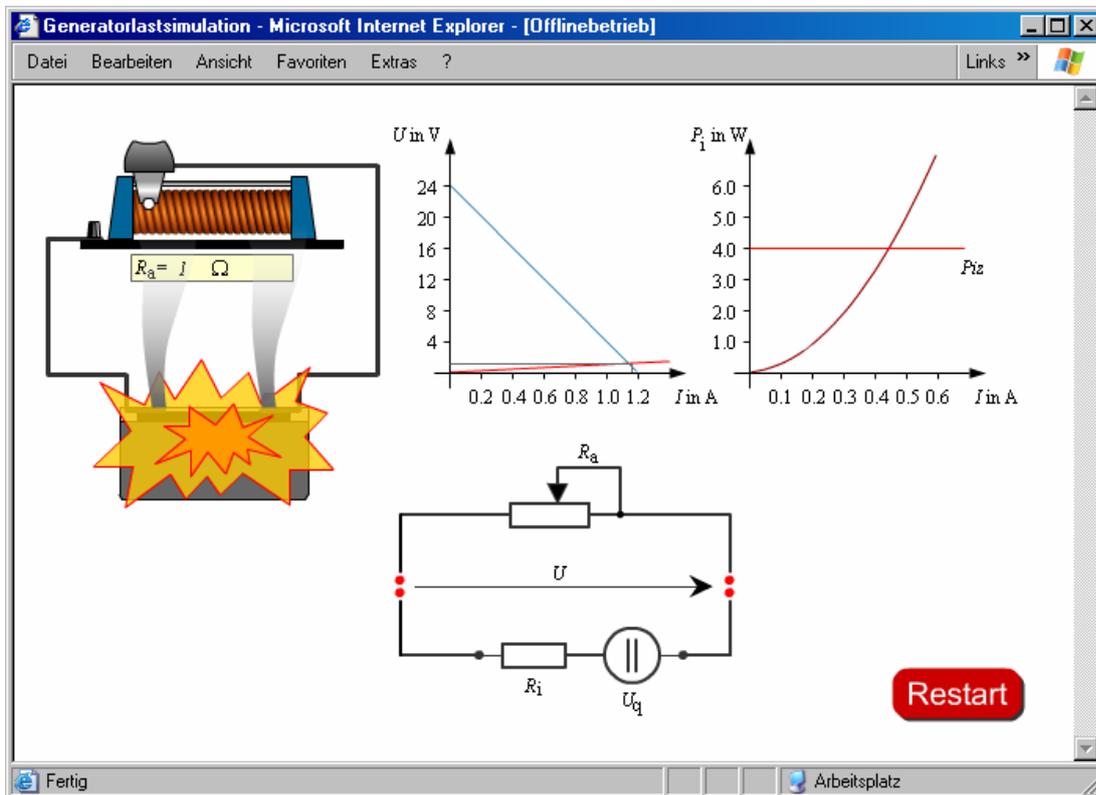


Phasenkompensierter Spannungsteiler





Experimentierumgebung Lineare Verzerrung: Rechteck an einem Tiefpass



Flash-Simulation eines belasteten Generators (angelehnt an das Beispiel aus [Rettig1997])

**Hilfe zur Aufgabe 1.14 - Microsoft Internet Explorer**

**Aufgabe 1.14**

Die Gleichungen zur Berechnung der Fourier-Koeffizienten

Hat diese Funktion Symmetrieeigenschaften?

- mittelwertfrei
- achsensymmetrisch
- nullpunktsymmetrisch
- halbwellensymmetrisch

Die Integrale finden Sie in der Integraltabelle

Mit dem Mathcad-Arbeitsblatt können Sie die Spektren berechnen und anzeigen.

**Antwort - Microsoft Internet Explorer**

Sie haben die Symmetrieeigenschaften nicht erkannt.  
Informieren Sie sich im Kompendium!

**Hilfe zur Aufgabe 1.14 - Microsoft Internet Explorer**

**Aufgabe 1.14**

Die Gleichungen zur Berechnung der Fourier-Koeffizienten

Hat diese Funktion Symmetrieeigenschaften?

- mittelwertfrei
- achsensymmetrisch
- nullpunktsymmetrisch
- halbwellensymmetrisch

Die Integrale finden Sie in der Integraltabelle

Mit dem Mathcad-Arbeitsblatt können Sie die Spektren berechnen und anzeigen.

**Antwort - Microsoft Internet Explorer**

Richtig!  
Die Funktion weist keine der angeführten Symmetrieeigenschaften auf.  
Effektiv ist die Berechnung des komplexen Fourier-Koeffizienten:

$$\underline{C}_n = \frac{1}{T} \int_{t_0}^{t_0+T} f(t) e^{-jn\omega t} dt.$$

**Überprüfung Ihrer Lösung**

**Aufgabe 1.14**

Vergleich Ihrer Ergebnisse mit den Ergebnissen und Darstellungen im Mathcad-Arbeitsblatt

Tragen Sie Ihre auf  $U_{\max}$  normierten Ergebnisse in das Mischpult ein und bewerten Sie die Summenkennlinie.

**MathCad-Arbeitsblatt: Grundstromkreis mit nichtlinearer Spannungsquelle**

*U-I*-Kennlinie der Spannungsquelle:  $U_Q(I) = U_q - K \cdot I^2$

Parameter der Spannungsquelle  $U_q := 10 \text{ V}$   $K := 1 \cdot \frac{\text{V}}{\text{A}^2}$

Leerlaufspannung  $U_L := U_q$

Berechnung des Kurzschlußstromes

Bestimmung der Nullstellen des Polynoms  $-K \cdot I^2 + U_q = 0$

$K := \frac{K}{\text{V}} \cdot \text{A}^2$   $U_q := \frac{U_q}{\text{V}}$   
 $v := \begin{pmatrix} U_q \\ 0 \\ -K \end{pmatrix}$   $w := \text{nullstellen}(v)$   $w = \begin{pmatrix} -3.162 \\ 3.162 \end{pmatrix}$

$K := K \cdot \frac{\text{V}}{\text{A}^2}$   $U_q := U_q \cdot \text{V}$   
 $I_K := w_1 \cdot \text{A}$   $I_K = 3.162 \text{ A}$

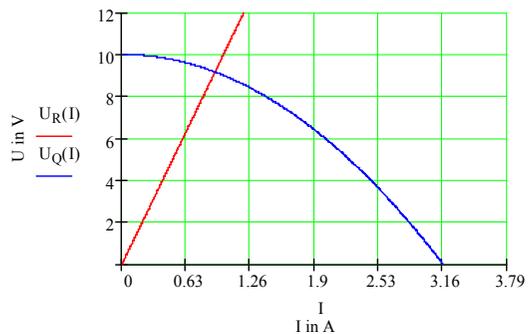
*U-I*-Kennlinie des Widerstandes:  $U_R(I) = R_a \cdot I$

Parameter des Widerstandes  $R_a := 10 \cdot \Omega$

*U-I*-Kennlinie der Spannungsquelle und des Widerstandes:

$I := 0, \frac{I_K}{1000} .. I_K$   $U_Q(I) := U_q - K \cdot I^2$   $U_R(I) := R_a \cdot I$

Darstellung der Kennlinien im Diagramm

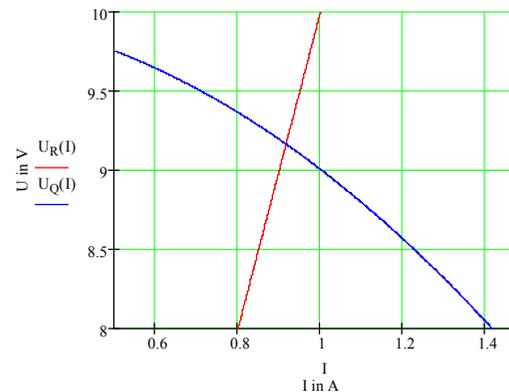


**Ablese** des Arbeitspunktes

Ändern Sie die Anzeigemaßstäbe, um die Koordinaten des Arbeitspunktes genauer ablesen zu können.

I anzeigen von / bis  $I_a := 0.5 \text{ A}$   $I_e := 1.5 \text{ A}$

U anzeigen von / bis  $U_a := 8 \text{ V}$   $U_e := 10 \text{ V}$



Berechnung des Arbeitspunktes

$U_Q(I) = U_R(I)$

$U_q - K \cdot I^2 = R_a \cdot I$   $-K \cdot I^2 - R_a \cdot I + U_q = 0$

Bestimmung der Nullstellen des Polynoms

$R_a := \frac{R_a}{\Omega}$   $K := \frac{K}{\text{V}} \cdot \text{A}^2$   $U_q := \frac{U_q}{\text{V}}$   
 $v := \begin{pmatrix} U_q \\ -R_a \\ -K \end{pmatrix}$   $w := \text{nullstellen}(v)$   $w = \begin{pmatrix} -10.916 \\ 0.916 \end{pmatrix}$

$K := K \cdot \frac{\text{V}}{\text{A}^2}$   $U_q := U_q \cdot \text{V}$   $R_a := R_a \cdot \Omega$

$I := w_1 \cdot \text{A}$   $I = 0.916 \text{ A}$

$U := K \cdot I^2$   $U = 0.839 \text{ V}$

Leistungsumsätze

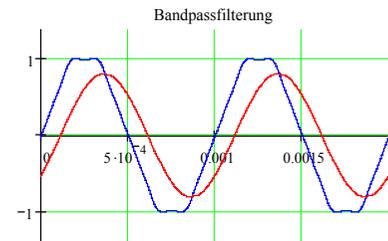
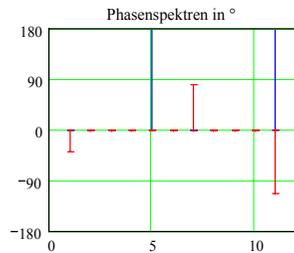
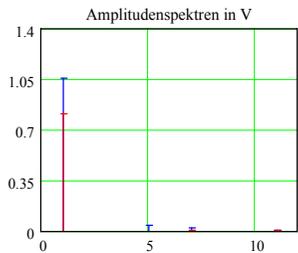
$P_a := R_a \cdot I^2$   $P_Q := U_q \cdot I$   $P_1 := P_Q - P_a$

$P_a = 8.392 \text{ W}$   $P_Q = 9.161 \text{ W}$   $P_1 = 0.769 \text{ W}$

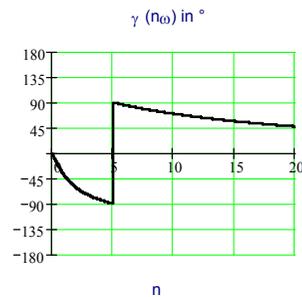
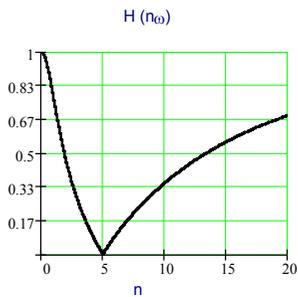
**Trapezfunktion am Eingang einer RC-Bandsperr**

$U_{max} := 1 \cdot V$   
 $T := 0.001 \cdot s$   
 $t_F := \frac{1}{6} \cdot T \quad \left( 0 < t_F \leq \frac{T}{4} \right)$   
 $f := \frac{1}{T} \quad f = 1000 \text{ Hz}$   
 $\omega := 2 \cdot \pi \cdot f$   
 Anzahl der Harmonischen  $m := 11$   
 $R := 900 \cdot 10^3 \cdot \Omega$   
 $f_{Mitte} := 5000 \text{ Hz}$   
 $C := \frac{1}{2 \cdot \pi \cdot f_{Mitte} \cdot R}$   
 $C = 3.5368 \times 10^{-11} \text{ F}$   
 $\omega \cdot R \cdot C = 0.2$

Eingang / Ausgang



Amplituden- und Phasenübertragungsfunktion



**Eingangsspannung: Trapezfunktion**

Gleichanteil  $Ae_0 := 0 \cdot V$   
 Koeffizienten  $n := 1, 3, \dots, m \quad b_n := \frac{4 \cdot U_{max}}{n \cdot \pi} \cdot \left( \frac{\sin\left(2 \cdot n \cdot \pi \cdot \frac{t_F}{T}\right)}{2 \cdot n \cdot \pi \cdot \frac{t_F}{T}} \right)$   $n := 2, 4, \dots, m \quad b_n := 0 \cdot V$   
 $n := 1, 2, \dots, m \quad a_n := 0 \cdot V$

Komplexe Eingangsamplitude  $Ae_n := b_n + j \cdot a_n$   
 Berechnung der Phasenwinkel  $\phi_n := \begin{cases} 0 & \text{if } Ae_n = 0 \\ \left[ \left( |Ae_n| \geq 0.001 \cdot U_{max} \right) \cdot \arg(Ae_n) \cdot \frac{180}{\pi} \right] & \text{otherwise} \end{cases}$

Rekonstruierte Funktion  $s_{PhaseIn}(t) := Ae_0 + \sum_{n=1}^m |Ae_n| \cdot \sin\left(n \cdot \omega \cdot t + \frac{\phi_n \cdot \pi}{180}\right)$

**RC-Bandsperr**

Übertragungsfunktion  $H_n := \frac{1 - (n \cdot \omega \cdot R \cdot C)^2}{1 - (n \cdot \omega \cdot R \cdot C)^2 + j \cdot 4 \cdot n \cdot \omega \cdot R \cdot C}$   
 $\gamma_n := \left( |H_n| \geq 0.001 \right) \cdot \arg(H_n) \cdot \frac{180}{\pi}$

**Ausgangsspannung**

Koeffizienten  $Aa_0 := 0 \cdot V \quad Aa_n := Ae_n \cdot H_n \quad \phi_{An} := \begin{cases} 0 & \text{if } Aa_n = 0 \\ \left[ \left( |Aa_n| \geq 0.001 \cdot U_{max} \right) \cdot \arg(Aa_n) \cdot \frac{180}{\pi} \right] & \text{otherwise} \end{cases}$   
 rekonstruierte Ausgangsfunktion  $s_{Phaseout}(t) := Aa_0 + \sum_{n=1}^m |Aa_n| \cdot \sin\left(n \cdot \omega \cdot t + \frac{\phi_{An} \cdot \pi}{180}\right)$

**Klirrfaktor**

$U_e := U_{max} \sqrt{1 - \frac{8}{3} \cdot \frac{t_F}{T}} \quad g_e := \frac{Ae_1}{\sqrt{2} \cdot U_e} \quad g_e = 0.9989 \quad k_e := \sqrt{1 - g_e^2} \quad k_e = 0.0463$

$U_{em} := \sqrt{\frac{1}{2} \cdot \sum_{n=1}^m (|Ae_n|)^2} \quad g_{em} := \frac{|Ae_1|}{\sqrt{2} \cdot U_{em}} \quad k_{em} := \sqrt{1 - g_{em}^2} \quad k_{em} = 0.0456$

$U_{am} := \sqrt{\frac{1}{2} \cdot \sum_{n=1}^m (|Aa_n|)^2} \quad g_{am} := \frac{|Aa_1|}{\sqrt{2} \cdot U_{am}} \quad k_{am} := \sqrt{1 - g_{am}^2} \quad k_{am} = 0.0062$

## Vorformulierte Aufgabenziele für mileET Aufgaben Y-Type 1

Beschreibung des Aufgabenziels	Formulierung des Aufgabenziels in dem Aufgabenassistent (editierbar)	Formulierung des Aufgabenziels in der Wissensbasis: TaskGoal
Erstellen eines Gleichungssystems mit Kirchhoff'schen Maschengleichungen	Erstellen der Maschengleichungen	kvls (_)
Erstellen eines Gleichungssystems mit Kirchhoff'schen Knotengleichungen	Erstellen der Knotengleichungen	kcls (_)
Erstellen eines Gleichungssystems für die Anwendung der Kirchhoff'schen Gleichungen	Erstellen der Kirchhoff'schen Gleichungen	kls (_)
Erstellen eines Gleichungssystems für die Methode der Knotenspannungsanalyse	Anwendung der Methode der Knotenspannungsanalyse	nvs (_)
Erstellen eines Gleichungssystems für die Methode der Maschenstromanalyse	Anwendung der Methode der Maschenstromanalyse	mcs (_)

## Vorformulierte Aufgabenziele für mileET Aufgaben Y-Type 2

Verbale Beschreibung des Aufgabenziels	Formulierung des Aufgabenziels in dem Aufgabenassistent	Formulierung des Aufgabenziels in der Wissensbasis
Berechnung des Ersatzwiderstandes, Anschlussklemmen IN	Ersatzwiderstand	equivalentR(IN)
Berechnung des Eingangswiderstandes, Anschlussklemmen IN	Eingangswiderstand	inputR(IN)
Ermittlung eines Ersatzzweipols, EL: Anschlussklemmen bzw. R-Element	Ersatzzweipol	equivalent2Terminal(EL)
Ermittlung einer Ersatzstromquelle, EL: Anschlussklemmen bzw. R-Element	Äquivalente Stromquelle	equivalentIq(EL)
Ermittlung einer Ersatzspannungsquelle, EL: Anschlussklemmen bzw. R-Element	Äquivalente Spannungsquelle	equivalentUq(EL)
Berechnung des Stromes im Element R unter Anwendung der Zweipoltheorie	Strom mithilfe eines äquivalenten Zweipols	current_equivalent2Terminal(R)
Berechnung der Spannung am Element R unter Anwendung der Zweipoltheorie	Spannung mithilfe eines äquivalenten Zweipols	voltage_equivalent2Terminal(R)
Berechnung des Stromes im Element R unter Anwendung einer Ersatzspannungsquelle	Strom mithilfe einer äquivalenten Spannungsquelle	current_equivalentUq(R)
Berechnung der Spannung am Element R unter Anwendung einer Ersatzspannungsquelle	Spannung mithilfe einer äquivalenten Spannungsquelle	voltage_equivalentUq(R)
Berechnung des Stromes im Element R unter Anwendung einer Ersatzstromquelle	Strom mithilfe einer äquivalenten Stromquelle	current_equivalentIq(R)
Berechnung der Spannung am Element R unter Anwendung einer Ersatzstromquelle	Spannung mithilfe einer äquivalenten Stromquelle	voltage_equivalentIq(R)

## Vorformulierte Aufgabenziele für die Aufgaben Y-Type 3 (in der Entwicklung):

```
taskType(3, [createAVMCcorrectI(_), createAVMCcorrectU(_), createAVMC(_)]).
```

**Schaltbilder**

Circuits: [Circuit1,..., CircuitN]

CircuitI: [[Element1,..., ElementM], Additions]

ElementI: [Type, Name, ConnectionPins, Properties].

ConnectionPins: [Pin1,Pin2]

Properties: [RName, [[IPin1,IPin2],IName], [[UPin1,Upin2], UName], PName]

Type = resistor, powerSourceVolt, powerSourceAmp, aMeter, vMeter, input

**Resistors**

Resistors: [Type, Name, Pins, Values]

Type: resistor

Name: 'R#'

Pins: [p#,p#]

Values: [R\_R, I\_R, U\_R, P\_R]

R\_R: 'R\_R#'

I\_R: [Direction:[p#,p#], Wert: 'I\_R#']

U\_R: [Direction:[p#,p#], Wert: 'U\_R#']

P\_R: 'P\_R#'

Beispiel: E11 = [resistor,'R1',[p4,p1],[R\_R1],[p1,p4],I\_R1],[p1,p4],U\_R1',P\_R1']]

**Ideale Spannungsquelle**

Ideale Spannungsquelle: [Type, Name, Pins, Values]

Type: powerSourceVolt

Name: 'uQ#'

Pins: [p#,p#]

Values: [R\_uQ, I\_uQ, U\_uQ, P\_uQ]

R\_uQ: '\$'

I\_uQ: [Direction:[p#,p#], Wert: 'I\_uQ#']

U\_uQ: [Direction:[p#,p#], Wert: 'U\_uQ#']

P\_uQ: 'P#'

Beispiel: E15 = powerSourceVolt,'Q4',[p1,p2],[\$,[[p2,p1],I\_Q4],[[p1,p2],U\_Q4'],'P\_Q4']]

**Anschlussklemmen**

Connecting Pins: [Type, Name, Pins, Values]

Type: input

Name: 'in#'

Pins: [p#,p#]

Values: [R\_in, I\_in, U\_in, P\_in]

R\_in: 'R\_in#'

I\_in: [Direction:[p#,p#], Wert: 'I\_in#']

U\_in: [Direction:[p#,p#], Wert: 'U\_in#']

P\_in: '\$'

Beispiel: E15 = input, 'in4', [p1,p2], [R\_in4,[[\_,\_],I\_in4],[[\_,\_],U\_in4'],\$]

**Parameter**

Parameter: [ParameterName, SymValue, NumValue, Unit, StatusSym, StatusNum].

ParameterName: Rname, Iname, Uname, PName.

SymValue in ".

NumValue: Zahlenwert (Vorsatz berücksichtigt).

Unit: in ", Einheit ohne Vorsatz (SI).

StatusSym, StatusNum: given, known, unknown, target.

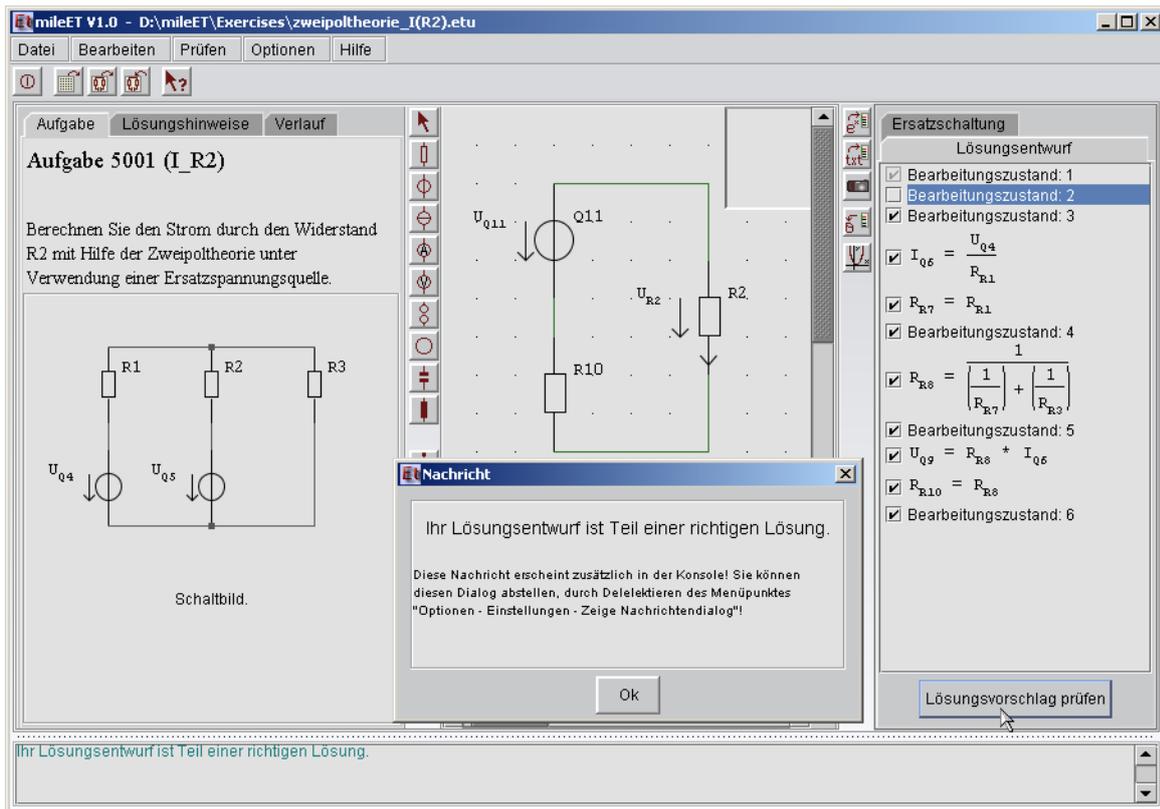
Beispiele für Parameter :

[r1, 'R1', \_, 'Ohm', given, \_].

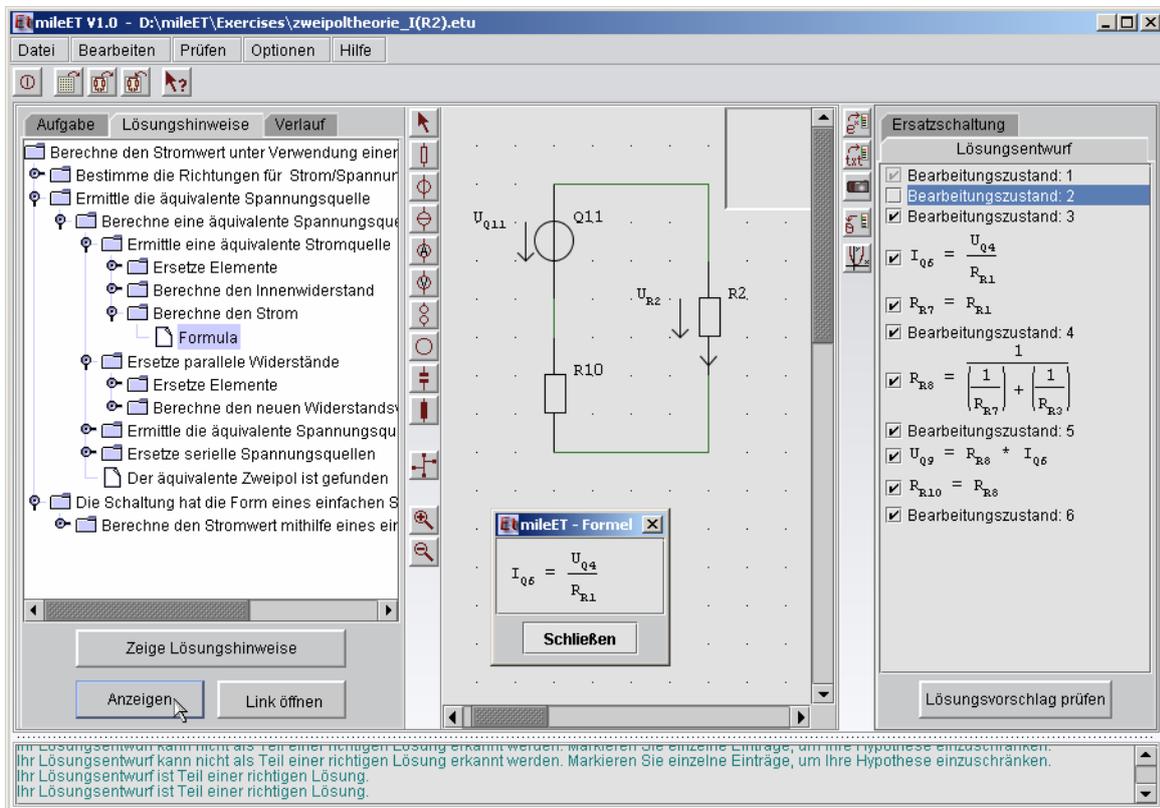
[r3, 'R1'+R2', 15, 'Ohm', known, known]

**Formeln**

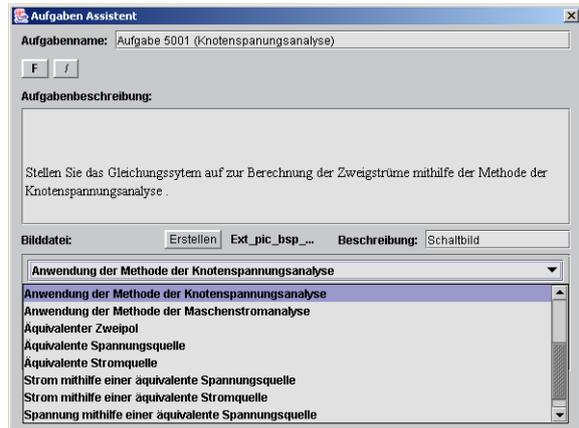
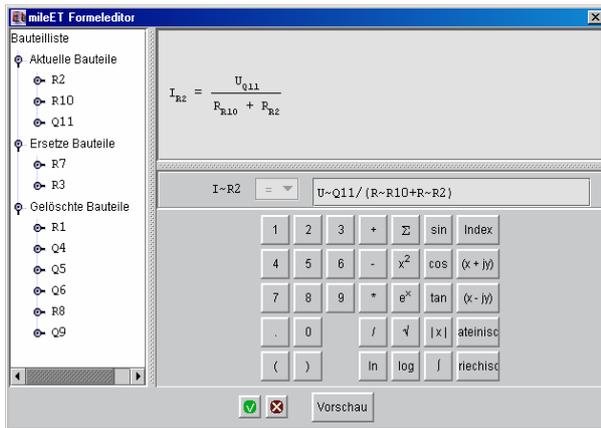
Formula: [LeftSide, RelOperator, RightSide, NullForm].



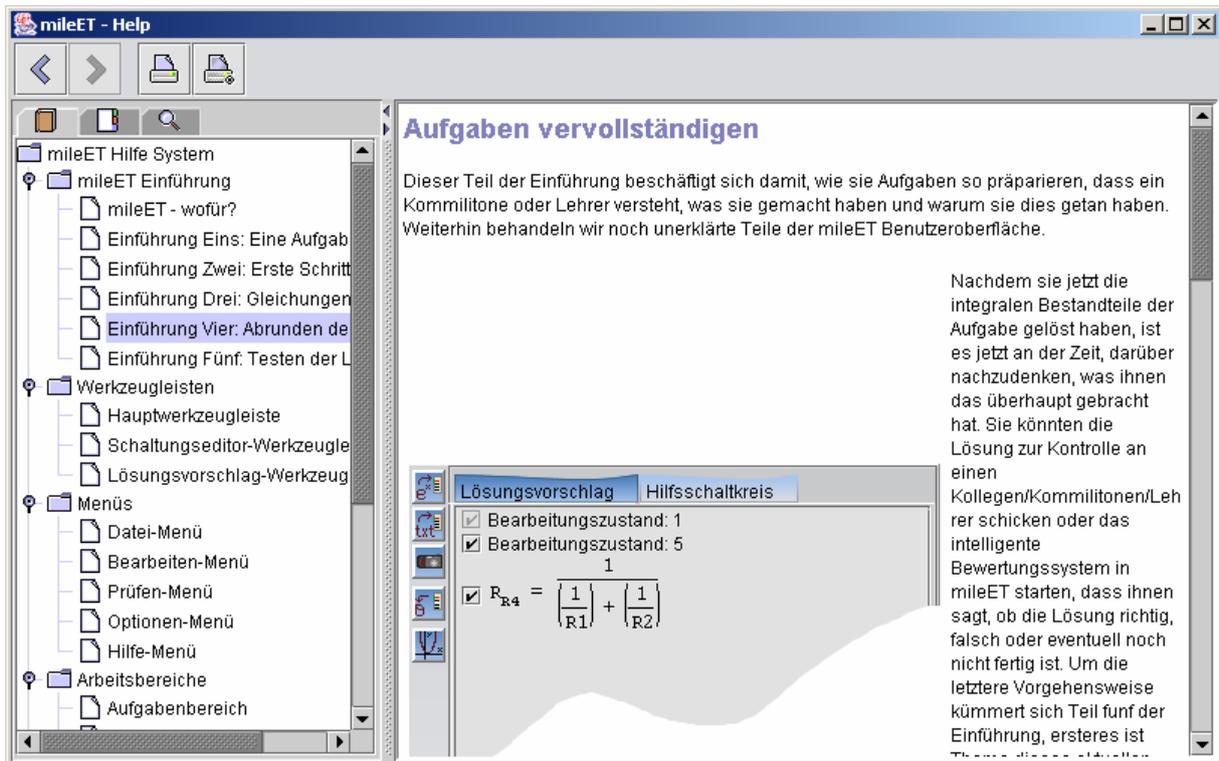
**Hypothesentesten beim Lösen der Aufgabe des Y-Typs 2**



**Situationsbezogene Hilfe für eine Aufgabe des Y-Typs 2**



**Formel-Editor und Aufgaben-Assistent**



**Bedienungsanleitung**

## 1. Allgemeines

### Systemanforderungen

Betriebssystem: MS Windows (getestet auf Windows 2000, XP); Speicherplatz: 6,50 MB bzw. 23MB; JRE 1.3.1 oder JRE 1.4.; Internet-Zugang für Benutzung der online Lernmodule.

### Installation

Java(TM) 2 Runtime Environment auf Ihrem Computer vorhanden: Das auf der CD vorhandene Verzeichnis *ET-System-TEST* auf die Festplatte kopieren (benötigter Speicherplatz ca. 6,50 MB).

Keine Java(TM) 2 Runtime Environment auf Ihrem Computer: Das auf der CD vorhandene Verzeichnis *ET-System-TEST(JRE)* auf die Festplatte kopieren (benötigter Speicherplatz ca. 23 MB).

### Programmaufbau

MileET hat zwei Modi: Dozenten- und Studenten-Modus. Der Dozentenmodus ermöglicht es, vorhandene Aufgaben zu ändern und neue Aufgaben zu erstellen. Der Studentenmodus ermöglicht es, zu den vorhandenen Aufgaben Lösungsentwürfe zu erstellen, diese auf Korrektheit überprüfen zu lassen und adaptive Hilfe anzufordern. In der aktuellen Version stellt die IPSE mileET dem Benutzer einen Schaltungsektor, einen Formeleditor und ein Lösungsentwurfeditor zur Verfügung.

## 2. Starten

Starten Sie die Datei *START-Dozent.bat* (Doppelklick) im Hauptverzeichnis ET-System, wenn Sie im Dozentenmodus arbeiten möchten.

Starten Sie die Datei *START-Student.bat* (Doppelklick) im Hauptverzeichnis ET-System, wenn Sie im Studentenmodus arbeiten möchten.

### Start-Fenster im Dozentenmodus

Zwei Möglichkeiten stehen zur Verfügung:

- Neue Aufgabe erstellen
- Vorhandene Aufgabe editieren – *Aufgabe Laden*

### Start-Fenster im Studentenmodus

Vier Möglichkeiten stehen zur Verfügung:

- Neue Aufgabe beginnen
- Vorhandenen Lösungsentwurf editieren – *Aufgabe fortsetzen*
- Eigene Schaltung testen (keine Kontrolle des Lösungsentwurfes) – *Leeren Schaltkreis öffnen*
- *Aufgabe aus der TaskWeb-Datenbank laden*

## 3. Arbeitsumgebung

### Bereiche

Bereiche, deren Inhalte vom System editiert werden: Lösungshinweise (links), Hilfsschaltkreis (rechts), Verlauf (links), Systemrückmeldungen (unten).

Bereiche, deren Inhalte vom Benutzer editiert werden können: Lösungsentwurf (rechts), Schaltungsektor (Mitte), Formeleditor (per Knopfdruck aufrufbar).

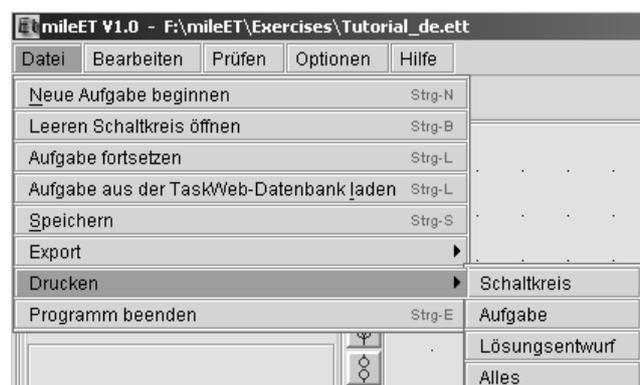
Bereich Aufgabe dient zur Anzeige der Aufgabenstellung.

### Menüleiste

Menü **Datei** im Dozentenmodus



Menü **Datei** im Studentenmodus



Menü **Bearbeiten** (Dozenten- und Studentenmodus)



*Rückgängig* – macht die letzte Aktion des Benutzers in dem Schaltkreiseditor und dem Lösungsvorschlag rückgängig.

*Kopieren/Einfügen* – kopiert markierte Schaltkreiselemente in die/aus der Zwischenablage.

*Zeige Bearbeitungszustände* – öffnet das Fenster mit den Schaltbildern aus dem Lösungsentwurfeditor.

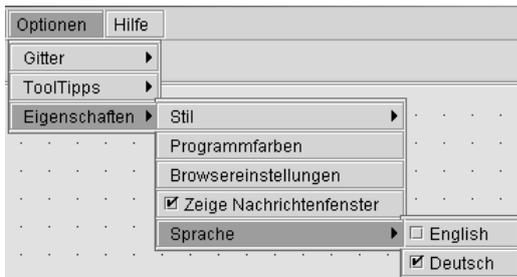
Menü **Prüfen** (Dozenten- und Studentenmodus)



*Lösungsvorschlag prüfen* – mit Haken markierte Einträge im Lösungsentwurfeditor werden überprüft, Textkommentare und Formeln mit undefinierten Variablen werden nicht kontrolliert.

*Zeige Lösungshinweise* – Vervollständigungsvorschlag zu einem korrekten Lösungs-(Teil)-Entwurf anfordern oder eine mileET-Lösung anfragen. Es gilt nicht im Falle *Leeren Schaltkreis öffnen!*

Menü **Optionen** (Dozenten- und Studentenmodus)



*ToolTips* – ToolTips-Einstellungen: Zeit, bis ToolTipp erscheint, kann konfiguriert werden.

*Stil* – Windows/mileET.

*Programmfarben* – Farben der einzelnen Komponente (Hintergrund, Menüs, Buttons) der Umgebung können gesetzt werden.

*Browsereinstellungen* – Festlegung des WEB-Browsers für die Anzeige der Hilfe.

Menü **Hilfe** (Dozenten- und Studentenmodus)



*Benutzerhandbuch* – öffnet die mileET-Hilfen.

*Direkthilfe* – Kontextsensitive Hilfe.

*Über mileET* – Zeigt das Fenster mit der Information über mileET.

**Buttonleisten**

**mileET-Buttonleiste**

- Das Programm verlassen
- Einen leeren mileET-Editor starten
- gespeicherten Lösungsentwurf öffnen
- Bearbeitungsstand speichern

**Schaltungseditor-Buttonleiste**

- Marker Tool (?)
- neuen Widerstand erstellen
- neue Spannungsquelle erstellen
- neue Stromquelle erstellen
- Amperemeter erstellen
- Voltmeter erstellen
- Eingangs/Ausgangsklemmen markieren
- Elemente verbinden
- Vergrößern
- Verkleinern

**Lösungsentwurfeditor-Buttonleiste**

- Erstellen/Bearbeiten von Formeln (mileET-Formeleditor wird geöffnet)
- Hinzufügen von Textkommentaren
- Übernahme des aktuellen Schaltbildes
- Entfernen des markierten Eintrages
- Aufruf der Anzeige von Funktionen (im Entwicklungsstadium)

## 4. Schaltungeditor

### Elemente markieren und verschieben

Schalten Sie den Editor in den Marker-Zustand um – Knopf , halten Sie die Steuerung-Taste gedrückt und klicken Sie mit der Maus (linke Taste) auf einzelne Elemente, um diese zu markieren. Die Gruppe von markierten Elementen kann verschoben werden.

### Verbindungen

Neue Verbindung erstellen:

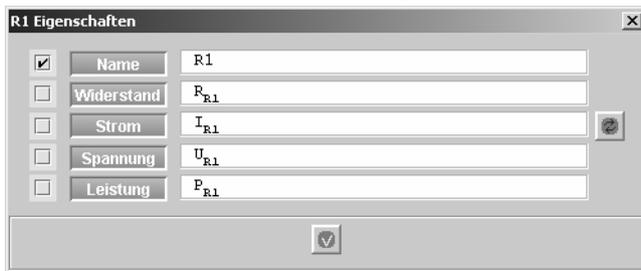
Klicken Sie auf den Knopf  - der Cursor wird zu . Klicken Sie auf einen freien Pin des Elementes. Ein Teil der neuen Verbindung erscheint, lassen Sie jetzt die Umschalt-Taste los und klicken Sie auf einen freien Pin des zweiten Elementes, die neue Verbindung wird erstellt.

Vorhandene Verbindung erweitern:

Klicken Sie auf den Button  - der Cursor wird zu . Klicken Sie auf eine vorhandene Verbindung (sie wird rot markiert) und klicken Sie auf einen freien Pin des zu verbindenden Elementes oder auf eine bestehende Verbindung: Elemente werden verbunden.

### Eigenschaften von Elementen setzen

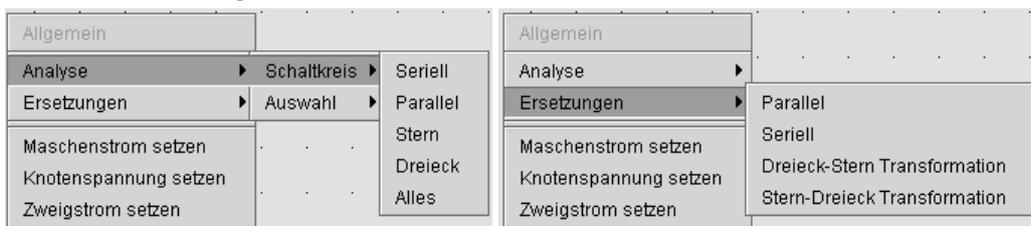
Die Eigenschaften der einzelnen Elemente können in dem Eigenschaften-Editor gesetzt werden (rechte Maustaste über Element ruft das entsprechende Kontextmenü auf).



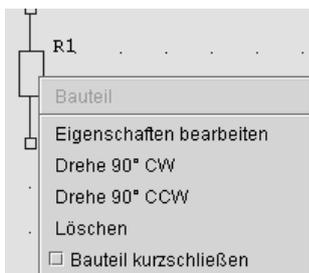
Über den Button  können Sie die Strom- bzw. Spannungsrichtung ändern. Klicken Sie auf die entsprechende Zeile und geben Sie die Eigenschaften in dem Fenster an. Den Namen können Sie über den Name-Editor eingeben. Über den Formeleditor können Sie Widerstands-, Strom-, Spannungs- und Leistungswerte eingeben.

### Kontextmenüs

Kontextmenü: *Hintergrund*

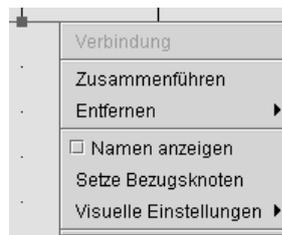


Kontextmenü *Bauteil*



Über *Eigenschaften bearbeiten* können einzelne Eigenschaften (Name, R, I, U, P, Strom- und Spannungsrichtungen) des Elementes angeben.

Kontextmenü: *Verbindung*



*Entfernen* – Sie können entweder die ganze Verbindung löschen oder einzelne Elemente trennen.

*Namen anzeigen* – Sie können die Namen der Knoten anzeigen lassen.

## 5. Anwendungsbeispiel

Erstellen einer Aufgabe im Dozentenmodus.

Starten Sie mileET im Dozentenmodus und wählen Sie **Neue Aufgabe erstellen**. Ein leerer mileET-Editor erscheint.

### Erstellen Sie die Elemente:

Schalten Sie den Editor in R-Zustand um (Toolbar ) – der Cursor wird zu , platzieren Sie das Element auf der Editorfläche. Analog können weitere Elemente erstellt werden.

Falls Sie die Elemente in einer anderen Reihenfolge erstellt haben, sind die automatischen ID Nummern anderes verteilt. Sie können Elemente umbenennen, indem Sie mit der rechten Maustaste auf das entsprechende Element klicken und das Eigenschaften-Fenster öffnen

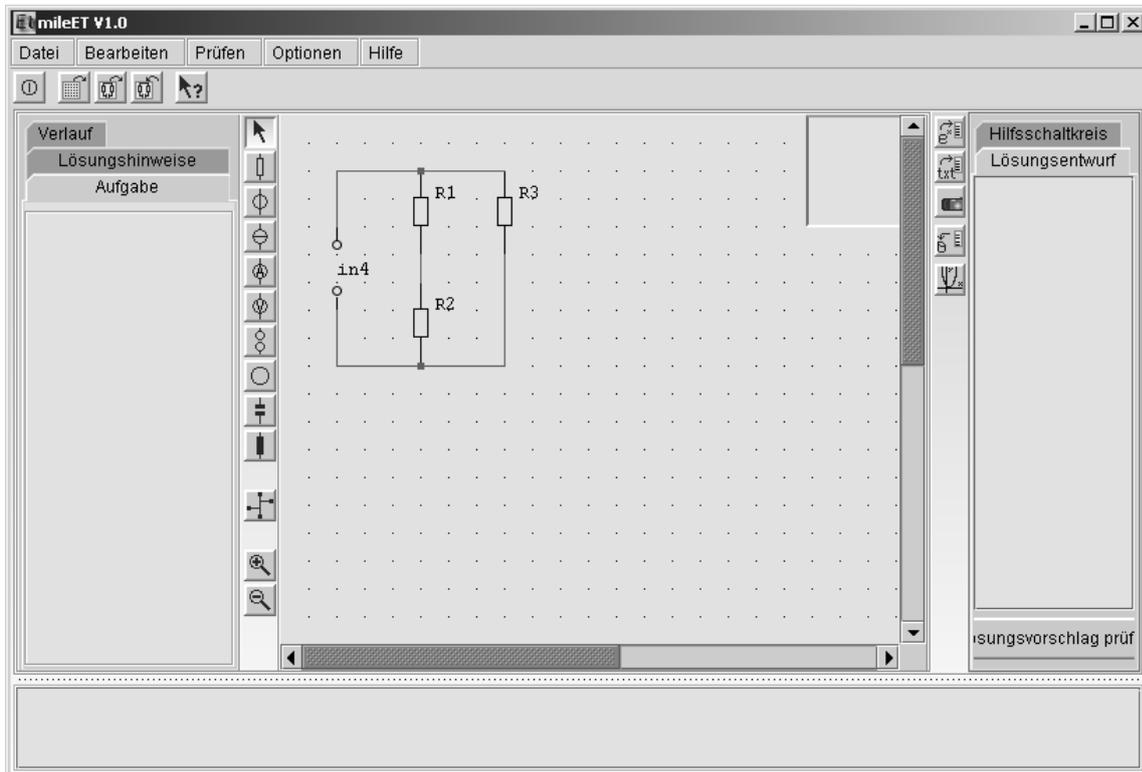
Über einen Formeleditor (Klick auf die entsprechende Zeile) können Sie die Widerstands-, Strom-, Spannungs- und Leistungswerte eingeben.

Erstellen Sie eine Pin-Markierung. mileET- Editor bietet eine Pin-Markierung der Klemmen an, so wie es bei den herkömmlichen Aufgabenstellungen üblich ist.

### Verbinden Sie Elemente zu einer Schaltung:

Schalten Sie den Editor in den Connection-Zustand (Toolbar ) um – der Cursor wird zu . Verbinden Sie die Pins der einzelnen Elemente wie im Bild.

Klicken Sie auf **Verkleinern** , damit die Schaltung in der kompakten Form gespeichert wird.



### Speichern Sie die Aufgabe

Wählen Sie den Menü-Punkt **Datei/Speichern**, es erscheint das Fenster:



Schreiben Sie den Name der zu erstellenden Datei: **test**.

Es erscheint das Fenster:

Schreiben Sie den Aufgabennamen und die Aufgabenstellung in die dafür vorgesehenen Stellen:

Für diese Aufgabe braucht der Lernende ein Schaltbild. Klicken Sie auf **Erstellen** und bestimmen Sie die Bildunterschrift (**Beschreibung**) (,Schaltbild' im Beispiel).

Jetzt müssen Sie das Aufgabenziel definieren. Die Menge der Aufgabenziele ist durch das mileET-Experten-system festgelegt. Die Formulierungen sind den Aufgabensammlungen der Partneruniversitäten entnommen. Wählen Sie das Ziel **Ersatzwiderstand** und bestimmen Sie als Bezugsklemmen **in4**.

Falls Sie die Aufgabe zum späteren Einfügen in die TaskWeb-Datenbank planen, klicken Sie **Für TaskWeb vorbereiten**. In dem Verzeichnis <mileET>/Exercise/Deploy/<Dateiname>/ werden entsprechende Dateien vorbereitet, die Sie später über FTP in die Datenbank einfügen können.

Klicken Sie **OK**, die Aufgabe ist gespeichert.

Diese Aufgabe kann im Studenten-Modus gelöst werden.

## Thesen

1. Die Lernsoftware soll sowohl den Lernenden als auch den Lehrenden Vorteile bringen.
2. Konzeption und prototypische Realisierungen zum Einsatz des Computers zu Lernzwecken in der elektrotechnischen Grundlagenausbildung haben eine lange Tradition. Während die Hauptinteressen der ersten Projekte zum Einsatz neuer Medien in der Bildung die Konzeption und Implementierung geschlossener Lernumgebungen waren, begann mit der Entwicklung der Netztechnologien eine Verschiebung der Schwerpunkte in Richtung offener, webbasierter Lernumgebungen.
3. *„Lernumgebungen sollen durch eine angemessene methodische Aufbereitung des Lehrstoffes und durch besondere Lehrmaßnahmen die angezielten Lernprozesse erleichtern.“*
4. Als softwaretechnisches Konzept stellt die Lernumgebung eine offene Struktur dar, die verschiedene Lehrkomponenten, wie z.B. interaktive Übungsprogramme, Hilfen und Werkzeuge, verwaltet und dem Nutzer anbietet, so dass ein flexibles erweiterbares System entsteht. Die Basis solch einer Lernumgebung bilden multimediale Elemente, die feinmodulare Lerneinheiten darstellen.
5. *„Die Bereitstellung von modularen kombinierbaren Lernbausteinen“*, die sich flexibel in verschiedene Lernszenarien einsetzen (Lehrer-Sicht) und sich in individuelle Lernumgebungen integrieren lassen (Lerner-Sicht) ist das Wesentliche bei der Konzeption offener Lernumgebungen.
6. Lehr-/Lernprogrammen lassen sich nach mehreren Kriterien klassifizieren. In einer Lernumgebung können verschiedene Arten von Lernsoftware interagieren bzw. in verschiedenen Formen der Zusammenarbeit eingesetzt werden.
7. Für Bildungsprozesse ist nicht die technische Integration verschiedener Medien von Bedeutung, sondern die Integration der Funktionen der medialen Formen wie z.B. Präsentation des Lernstoffes und Interaktivität.
8. Da in virtuellen (computerbasierten) Lernumgebungen für individualisiertes Lernen das Lernen in erster Linie die kognitive Auseinandersetzung des Lernenden mit dem zu lernendem Inhalt (dem Lernobjekt) bedeutet, wird Interaktivität der Lernobjekte zu einem der entscheidenden Qualitätskriterien des Lernens und der Lernumgebungen.
9. Die meisten Beispiele der interaktiven Lernprogramme in der elektrotechnischen Grundlagenausbildung sind Berechnungs- und Simulationstools sowie Test- bzw. Übungsprogramme. Die intelligenten Lernprogramme für die Grundlagen der Elektrotechnik bleiben immer noch eine Seltenheit.
10. *„In jeder Bildungssoftware schlägt sich ein theoretisches Lernmodell nieder.“*
11. Die GETsoft Lernumgebung stellt eine offene hypermediale Lernumgebung dar, die das Lernen nach konstruktivistischen Prinzipien ermöglicht. Die einzelnen Module sind nach den Prinzipien des Instruktionsdesigns implementiert.

12. Durch die Aufteilung der GETsoft-Lernprogramme in Module wie Kompendium, Aufgabensammlung und Experimentierumgebung sind mindestens drei Lernszenarios möglich: Theorieorientiertes, Aufgabenorientiertes und Experimentorientiertes Lernen. Die Lernenden mit unterschiedlichen Lernstilen und Wissensniveaus können somit mit der Lernumgebung arbeiten.
13. Eines der Ziele der Experimentierumgebung liegt darin, den Lehrstoff für die Benutzer in attraktiver und interaktiver Form darzustellen, um sie auf die Situationen und Eigenschaften aufmerksam zu machen, die sie eventuell dazu bringen, Fragen über das Gesehene zu stellen, und sie so zu theoretischen Erklärungen der Phänomene zu leiten.
14. Um sich auf die spätere Berufspraxis vorzubereiten, sollen die Studierenden mit computerbasierten Ingenieurwerkzeugen konfrontiert werden.
15. Bei der Auswahl eines Ingenieurwerkzeuges sollen solche Kriterien wie Einsatzszenarien, Komplexität der zu bearbeitenden Aufgaben, Erlernbarkeit des Systems und Verfügbarkeit berücksichtigt werden. Als Ingenieurwerkzeug für die elektrotechnische Grundlagenausbildung eignet sich das Computer-Algebra-System Mathcad.
16. Die webbasierte multimediale Lernumgebung bietet den Lehrenden viele multimediale Komponenten für das Zusammenstellen ihrer Vorlesungen und Übungen.
17. Die GETsoft-Studierenden haben Zugang zu den typischen Themen und Problemstellungen des Lehrgebietes. Interaktive Experimente und Aufgaben mit Möglichkeiten zur Selbstkontrolle ermöglichen eine aktive Verarbeitung des Lehrstoffes.
18. Kein Programm aus der GET-Domäne stellt den Benutzern die Möglichkeit zur Verfügung, eigene Lösungsentwürfe in Form von Schaltbildern und Formeln zu verarbeiten: zu erstellen, zu editieren und unvollständige Lösungen zu überprüfen. Ein Versuch, dieses Defizit zu beheben, stellt die wissensbasierte Lernumgebung mileET dar.
19. Die Entwicklung der wissensbasierten Problemlöseumgebung mileET basiert auf der kognitionswissenschaftlichen ISP-DL-Theorie (Impasse-Success-Problem-Solving-Driven-Learning), die besagt, dass *„der Wissenserwerb insbesondere durch Lerneraktivitäten wie selbständiges Formulieren von Hypothesen und Selbsterklärungen von systemgenerierten Hilfen und Hinweisen gefördert wird“*. Diese Art des Lernens soll in den IPSEs (Intelligent Problem Solving Environments) ermöglicht werden.
20. Die Stärke von IPSEs liegt in *„interactive problem solving support“*.
21. *„Die Vermittlung methodischer und wissenschaftlicher Grundlagen ist wichtiger als reines Faktenwissen“*. Die wissensbasierte Komponente in mileET ermöglicht den Erwerb von methodischen Grundlagen zu den festgelegten GET-Themengebieten.
22. Die Problemlöseumgebung mileET ist in der Lage, sowohl beliebige Aufgaben zu diesen Themen selbständig zu lösen, eigene Lösungen zu kommentieren, sowie (partielle) Lösungsentwürfe der Lernenden zu interpretieren und u. U. zu vervollständigen.
23. Die Studierenden haben in mileET die Möglichkeit, selbständig mittels Einschränkung der Hypothese nach eigenen Fehlern zu suchen.

24. Dank den speziell an die Lösungsmethoden orientierten Regeln besitzt mileET die Fähigkeit, die Aufgaben so zu lösen, wie es den Studenten im Lehrbetrieb beigebracht wird. Somit kann das Programm die Rolle des Tutors übernehmen. Umfang und Flexibilität der Wissensbasis ermöglichen den Lernenden außerdem, verschiedene und partielle Lösungswege zu erstellen und sie selbständig zu vergleichen.
25. Die Lernenden können aktiv und kooperativ mit und in mileET lernen. Die Lehrenden können neue Aufgaben erstellen und dabei von der Routinen-Korrekturarbeit entlastet werden.
26. Mit dem mileET-Ansatz wird den Studierenden und Lehrenden ein Tool zum Aufbereiten, Aufbewahren, Verarbeiten und Überprüfen ihrer Aufgaben und Aufgabenlösungen angeboten.
27. Um didaktische Anforderungen an die eLearning-Systeme in einem weiteren Sinne zu erfüllen, muss das System um eine Komponente zur Qualitätsbeurteilung von Lösungen erweitert werden. Den Studierenden könnte somit ein systemunterstützter Vergleich verschiedener Lösungen ermöglicht, sowie Vorschläge zur Optimierung ihrer Lösungen unterbreitet werden.
28. Eine Erweiterung der adaptiven Unterstützung in mileET auf Nachbarthemen, wie z.B. Analyse der Netzwerke bei sinusförmiger Erregung oder Berechnung technischer Magnetkreise, erfordert eine Erweiterung der Wissensbasis und der Benutzerschnittstelle.
29. Themengebiete späterer Semester haben sehr komplexen mathematischen Hintergrund, so dass aus heutiger Sicht eine intelligente Unterstützung der Lösungswege sehr aufwändig erscheint.
30. Eine Wissensdiagnose-Komponente zu theoretischen Grundlagen mathematisch komplexer Domäne könnte der erste Schritt auf diesem Themenfeld sein.
31. Um die gesamte GET-Lehre durch eine computerbasierte Lernumgebung zu unterstützen, soll die im Rahmen des Projektes mile entstandene GETsoft-Struktur (TaskWeb, LearnWeb, BookWeb) durch weitere Komponenten wie z.B. LabWeb- und TestWeb-Umgebungen erweitert werden.
32. PSpice-Sprache und mileET-Datenstrukturen bieten die Grundlage für die Entwicklung von den GET-spezifischen CirML oder TaskML.
33. Die Struktur und Inhalte der TaskWeb-Datenbank reichen aus, um ein webbasiertes Testsystem ohne intelligente Unterstützung zu implementieren. Die Entwicklung einer im KI-Sinne adaptiven Klausurumgebung für GET bedarf einer weiteren Forschung.