

5 Konsistenzprüfungs- und Wissenserwerbskomponente

Der Vorgang des **Wissenserwerbs**, d.h. der Transfer und die Transformation von Expertise in die Wissensbasis sowie deren Wartung und Einhaltung der Konsistenz wird als "Wissensakquisition" bezeichnet. Insbesondere die Instandhaltung von bereits implementiertem Wissen ist in der Literatur vielfach als Problem bzw. Engpaß bei existierenden Expertensystemen erwähnt worden [CRY-94] [BÖH-88]; daher wurde der "Wissensingenieur" eingeführt [TUR-92] [SCH-87], der die Akquisition, Formalisierung und Pflege des Wissens übernehmen soll.

Andererseits wird eine geordnete Vorgehensweise bei der Problemdarstellung und -Lösung unter dem Begriff "Expertisemodell" geführt [SCH-94]. Ein Expertisemodell beschreibt das erforderliche Problemlösungsverhalten eines wissensbasierten Systems sowie die aufgabenspezifische Wissensstruktur und wird in einem Gesamtsystem für alle jene Aufgaben erstellt, die als wissensintensiv identifiziert werden. Das Problemlösungsverhalten des menschlichen Experten ist als Eingangsinformation hierbei auszuwerten. Das Expertisemodell kann unter Nutzung von Diagrammen und ggf. einer für die Konzeptphase geeigneten Modellierungssprache [FEN-94] spezifiziert werden.

Die komfortable und fehlervermeidende Erstellung bzw. Änderung von Wissensbasen erfordert bei allen nicht-trivialen Anwendungsfällen den Einsatz von Hilfsmitteln, um die verschiedenen Wissenseinheiten und deren Zusammenhänge zu erfassen, zu formalisieren sowie die Konsistenz der Wissensbasis zu garantieren. Damit ist dann eine aufwands- und zeitgerechte Erstellung und Prüfung der Wissensbasis möglich, ohne daß der Anwender dieses Systems auf interne Wissensrepräsentations- und Implementierungsdetails eingehen muß.

Die Erstellung unterschiedlicher Wissensbasen für die Anwendung mit der hier verwendeten Inferenzkomponente (neben dem hier betrachteten Netzwiederaufbau-System, siehe [FRA-96]) wurde zuerst ohne Verwendung weiterer Hilfsmittel vollzogen. Aus den gewonnenen Erfahrungen entstanden die Anforderungen bezüglich der Schaffung zusätzlicher Komponenten, die beim Wissenserwerbsprozeß sowie der Pflege der Wissensbasis unterstützend mitwirken.

Denn trotz seiner Generizität kann beim praktischen Einsatz des hier betrachteten Netzwiederaufbau-Systems die zusätzliche Eingabe einiger **spezifischer Regeln** und/oder eine partielle Änderung der Vorgehensweise in der Wissensbasis erforderlich bzw. erwünscht sein:

- Es können **Vereinbarungen** mit Verbrauchern, unter- oder überlagerten Netzbetreibern bestehen, die nicht auf Basis physikalischer oder topologischer Gegebenheiten aus der Prozeßdatenbank des Netzes ableitbar sind, so z.B. zur Benachrichtigung von Kunden vor Wiederschaltung oder zur Bereitstellung von Leistung im Störfall.
- Andererseits kann es beabsichtigt sein, heuristische Erfahrung des Betriebspersonals bzw. betriebsinterne Richtlinien in das Vorgehen beim Netzwiederaufbau zusätzlich einzubringen (z.B. Öffnen aller Leistungsschalter vor Wiederaufbau-Beginn [SCH95]).

Allerdings müssen in solchen Fällen die erforderlichen spezifischen Regeln in **syntaktisch** korrekter Form und **logisch konsistent** [MAR-90] [LIU-95] in das bestehende Regelgefüge eingebettet werden können, wobei es wünschenswert – und hinsichtlich des Aufwandes nicht zuletzt auch notwendig – erscheint, daß dieses durch das Betriebspersonal des jeweiligen Netzes geschehen kann.

Während eine **semantische Konsistenzprüfung** der Wissensbasis übergeordnetes Sachwissen erfordert und daher im Rahmen der Verifikation auf Basis menschlicher Expertise erfolgen muß – hierbei kann sich eine leistungsfähige und flexible Erklärungskomponente als sehr hilfreich erweisen, vgl. Kap. 6.2 –, kann die **Korrektheit der Syntax** der Regeln sowie deren **formale logische Konsistenz** in ihrer potentiellen Verknüpfung untereinander durch im Rahmen der vorliegenden Arbeit geschaffene Sub-Systeme sichergestellt werden, die im Folgenden beschrieben werden; Bild 5.1 zeigt die Integration dieser Funktionen in das generische Netzwiederaufbau-System.

Die **Wissenserwerbskomponente** wird unabhängig vom Inferenzprozeß eingesetzt und bildet damit den "off line"-Teil der Mensch-Maschine-Schnittstelle mit eigener graphischer Oberfläche (siehe Bild 5.1 und Kap. 5.2). Hierbei wird durch **Visualisierung der potentiellen Regel-Konnektivitäten** ein gezieltes Navigieren und anschließendes Editieren in der Wissensbasis unterstützt. Dies erfordert einen Zugriff auf die Regeln in der Wissensbasis sowie die Abspeicherung ihrer potentiellen Konnektivitäten in einer Datei.

Die dort gespeicherten Informationen werden von der **Konsistenzprüfungskomponente** (Kap. 5.1) angelegt bzw. ebenfalls benötigt. Auch diese Komponente, die über die Dialog-Schnittstelle des Expertensystems mit dem Anwender kommuniziert, arbeitet unabhängig vom Inferenzprozeß (Bild 5.1).

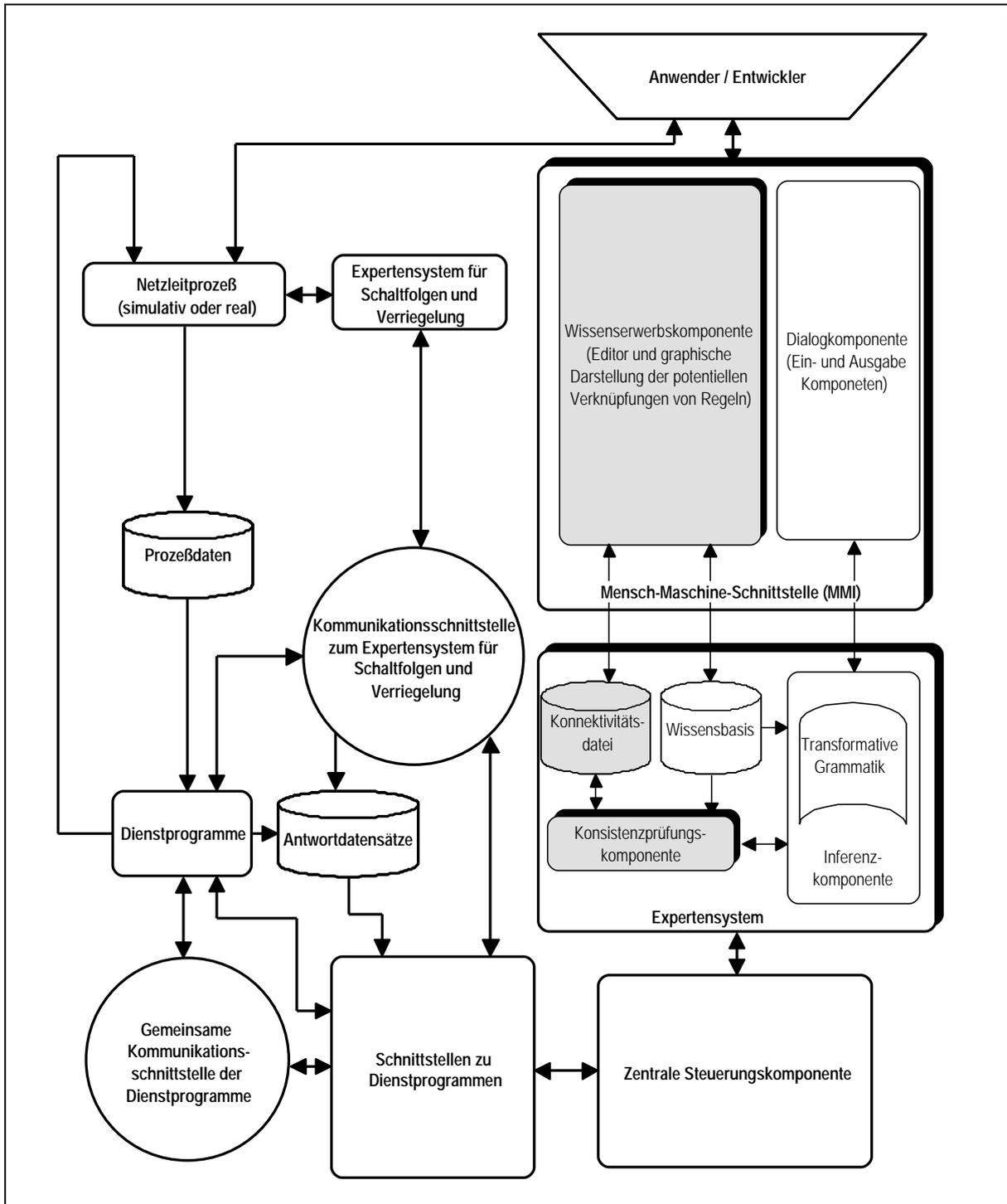


Bild 5.1: Integration der Wissenserwerbs- und Konsistenzprüfungs-komponente in das generische Netzwiederaufbau-System

Beide Funktionen sind **generisch** konzipiert, unabhängig voneinander einsetzbar und haben sich bei der Abrundung der Wissensbasis im Rahmen der Verifikation des Gesamtsystems als sehr nützlich erwiesen; sie können grundsätzlich auch unabhängig vom Einsatzgebiet des Expertensystems bei der Erstellung und Änderung weiterer Wissensbasen [HAR-95] [FRA-96] angewandt werden.

5.1 Konsistenzprüfung der Regel-Konnektivitäten

Die Konsistenzprüfungskomponente übernimmt die Analyse der Wissensbasis hinsichtlich ihrer **formalen Vollständigkeit** und untersucht selbsttätig in Rahmen verschiedener Prüfkriterien die logische Verknüpfung der Regeln in der Wissensbasis untereinander.

5.1.1 Verknüpfung der Regeln in der Wissensbasis

Die potentielle Verknüpfung der Regeln untereinander ist in der Wissensbasis festgelegt. Durch den Einsatz von Vorwärts- und Rückwärts-Regeln (Kap. 2.3.1) wird prinzipiell jede beliebige Vernetzung der Regeln in einem Lösungspfad ermöglicht (Bild 5.2). In einem aktuellen Inferenzprozeß wird ein bestimmter Pfad aus diesen kombinatorisch außerordentlich **variantenreichen** Möglichkeiten beschrieben.

Im Folgenden werden einige Erläuterungen und Definitionen für das Verständnis dieser Struktur zusammengefaßt:

- a) Der "**Eintritts-Verweis**" (Bild 5.2, Hinweis 7) ist auf das gesamte als nächstes zu bearbeitende vorwärtsverkettete Regel-Paket gerichtet; mindestens ein solcher Verweis muß für jedes Regel-Paket in der Wissensbasis vorhanden sein, um dieses überhaupt zur Anwendung bringen zu können.
- b) Der Verweis für den Fall der erfolgreichen Überprüfung einer Regel-Alternative (sog. "**Austritts-Verweis**", siehe Bild 5.2, Hinweis 8) kann prinzipiell durch entsprechenden Eintrag innerhalb der Regel-Syntax (Liste von Folgeregeln bzw. "Folgeliste", siehe Kap. 2.3.1 und Anhang A) festgelegt werden.
- c) Der erste eingetragene Name eines Regel-Paketes in der "Folgeliste" ist als ein "**direkter Verweis**" zu interpretieren. Dieses Regel-Paket wird als nächster Schritt bei der erfolgreichen Überprüfung der aktuellen Regel-Alternative in den Inferenzprozeß (d.h. in den aktuellen Lösungspfad) einbezogen.
- d) Die ggf. vorhandenen weiteren Einträge in der Liste von Folgeregeln einer erfolgreich überprüften Regel-Alternative werden auf jeden Fall innerhalb der weiteren Vorgehensweise abgearbeitet und sind als "**indirekte Verweise**" zu verstehen. Diese Einträge bestimmen damit die weiteren Regel-Pakete, die im Laufe des aktuellen Lösungspfades überprüft werden. Dabei ist zu beachten, daß ihre Bearbeitung durch aufeinanderfolgende direkte Verweise aufgeschoben wird, bis kein weiter direkter Verweis mehr vorhanden ist. Demzufolge können solche multiplen Einträge in der Zielliste

nur dann zur Anwendung kommen, wenn Regel-Alternativen in der Wissensbasis vorhanden sind, die bei ihrer erfolgreiche Überprüfung keine zusätzlichen Einträge in der Zielliste vornehmen (Kap. 2.3.1).

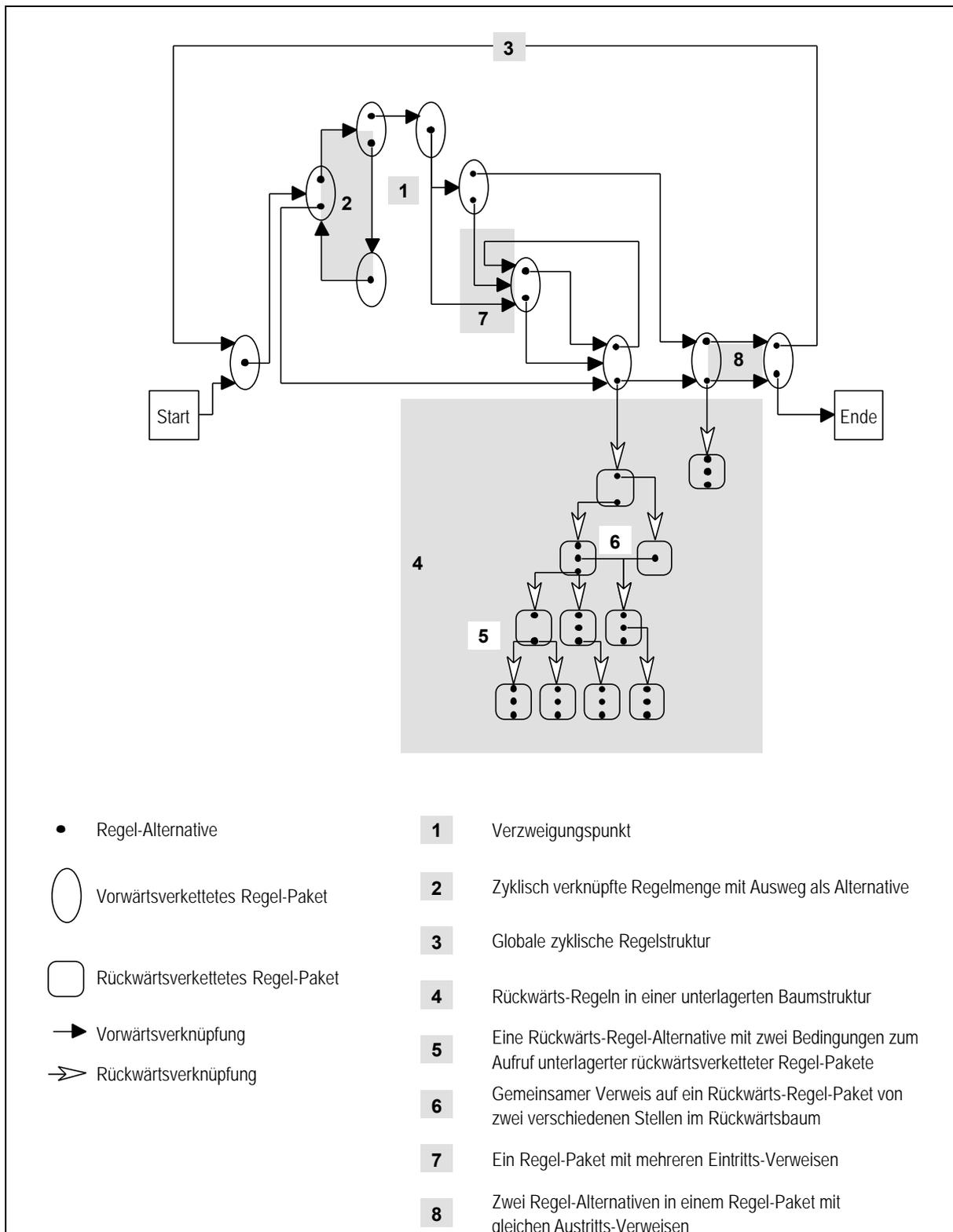


Bild 5.2: Mögliche Regel-Konnektivitäten

- e) Die Vorwärts-Regeln, die keine Einträge als Folgeregeln beinhalten, bieten eine solche Möglichkeit und geben damit einen **"dynamischen Verweis"** auf das erste bereits in vorherigen Schritten auf dem aktuellen Lösungspfad in die Zielliste eingetragene und als nächstes abzuarbeitende Regel-Paket. Diese Regel-Alternativen können dann – im Gegensatz zu anderen – mehrere unterschiedliche Austritts-Verweise zu weiteren Regel-Paketen aufweisen, deren Vorkommen von den aktuellen Einträgen in der Zielliste abhängig ist. Sie erfüllen damit innerhalb der Regelstruktur die Funktion einer **"Weiche"** mit unterschiedlichen Verzweigungsmöglichkeiten, abhängig von den vorher abgearbeiteten einzelnen Schritten (Vorgeschichte des aktuellen Lösungspfades), siehe Bild 5.2, Hinweis 1.
- f) Die in einem Regel-Paket enthaltenen Regel-Alternativen weisen gleiche Eintritts-Verweise auf. Im Gegensatz dazu sind die Austritts-Verweise jeder einzelnen Regel-Alternative innerhalb eines Regel-Paketes unterschiedlich, dürfen aber auch auf dasselbe Regel-Paket verweisen (siehe Bild 5.2, Hinweis 8).
- g) Bei der Unterlagerung rückwärtsverketteter Regel-Pakete unter beliebige Regel-Bedingungen bestehen grundsätzlich keine Einschränkungen. Im Sinne der Übersichtlichkeit wurde allerdings festgelegt, daß solche unterlagerten Strukturen keine **zyklischen** Verweise beinhalten dürfen (Baumstruktur) und daß die beteiligten rückwärtsverketteten Regel-Pakete keine eigenständigen Austritts-Verweise auf weitere überlagerte Strukturen beinhalten dürfen.
- h) **Zyklische Vorgehensweisen** (siehe Bild 5.2, Hinweise 2 und 3) lassen sich bei vorwärtsverketteten Regel-Paketen dadurch erzeugen, daß ein bereits in einem Lösungspfad beteiligtes Regel-Paket wiederholt aufgerufen wird. Dabei muß aber gewährleistet sein, daß mindestens eine Regel-Alternative existiert, die einen **Ausweg** aus diesem Zyklus ermöglicht. In Prinzip kann ein solcher Zyklus auch die gesamte Wissensbasis umfassen, z.B. als Prozeßbegleitung (siehe [KRO-99b] und Bild 5.2, Hinweis 3).
- i) Beim Start des Expertensystems wird der Eintritt in die Bearbeitung der Regeln grundsätzlich dadurch erreicht, daß zuerst ein Regel-Paket namens **"start"** bearbeitet wird, von den aus dann auf weitere Regel-Pakete verwiesen wird (Bild 5.2).
- j) Der Inferenzprozeß wird dann beendet, wenn die Zielliste nach der Bearbeitung einer sog. **"ende"-Regel** keine weiteren Einträge mehr aufweist (Bild 5.2).

Die somit implementierbaren Regelverknüpfungen sind vor allem von der **Anwendung** abhängig; die vielseitige Strukturierbarkeit der Wissensbasis hat sich neben der hier beschriebenen auch in anderen – ebenfalls prozeßgekoppelten – Anwendungen des Expertensystems im Sinne einer "Shell" als sinnvoll und vor allem flexibel erwiesen [KRO-97]. Andererseits haben sich aus den vorliegenden Erfahrungen mit dem sukzessiven Ausbau von großen Wissensbasen [KÖN-95] einige Problemfelder herausgestellt, die bei der logischen Verknüpfung der Regeln untereinander auftreten können; daher wurde das nachfolgend beschriebene Prüfprogramm implementiert.

5.1.2 Funktionen der Konsistenzprüfungskomponente

Prinzipiell wird bei der Konsistenzprüfung unter Simulation der Inferenzeigenschaften eine **vollständige** Suche durch alle kombinatorisch möglichen Lösungspfade vorgenommen. Dies geschieht "off line" und umfaßt die gesamte Wissensbasis. Als Ergebnis werden erkannte Fehler sowie warnende Hinweise oder teilweise auch Korrekturvorschläge bereitgestellt.

Zentraler Angelpunkt des Prüfprogrammes ist die von diesem initial erstellte sogenannte "Konnektivitätsdatei", die sämtliche potentiell möglichen Verweise von Regeln untereinander (d.h. jeweils von einer Regel-Alternative zum nächsten zu bearbeitenden Regel-Paket, abhängig von der jeweiligen Vorgeschichte, siehe Kap. 5.1.1 Punkt *e*) beinhaltet und in diesem Sinne die Grundstruktur der Wissensbasis wiedergibt (Kap. 5.1.1). Bei der Überprüfung dieser Konnektivitätsdatei werden folgende Inkonsistenzen erkannt:

a) Existenz einer aufrufenden Vorwärts-Regel-Alternative

Es muß gewährleistet sein, daß alle Vorwärts-Regel-Pakete aus mindestens einer anderen Regel-Alternative aufgerufen werden und damit mindestens ein Eintritts-Verweis (Kap. 5.1.1) vorhanden ist. Hierbei ist zu bemerken, daß dies eine notwendige aber nicht hinreichende Bedingung dafür darstellt, daß das entsprechende Regel-Paket bei der Abarbeitung der Wissensbasis potentiell zur Anwendung kommen könnte. Dies beruht auf der Tatsache, daß auf das zu untersuchende Regel-Paket – auch wenn es einen Eintritts-Verweis besitzt – aus einer "losen" Kette mehrerer Regeln verwiesen werden kann, auf deren erste kein Eintritts-Verweis existiert.

b) Überprüfung der Einbindung der Vorwärts-Regel-Pakete in mindestens einen Lösungspfad

Hierbei wird überprüft, ob alle Vorwärts-Regel-Pakete (Bild 5.2) prinzipiell vom Beginn der Abarbeitung ("start"-Regel siehe Kap. 5.1.1, Punkt *i*) aus erreicht werden können bzw. auf einen definierten Endzustand ("ende"-Regel siehe Kap. 5.1.1, Punkt *j*) in der Wissensbasis führen können. Es ist also damit festzustellen, ob alle in der Wissensbasis vorhandenen

Vorwärts-Regeln Mitglied mindestens eines potentiellen Lösungspfades sind, der sich von Anfang bis Ende der Wissensbasis erstreckt. Dies ist als notwendige und hinreichende Bedingung dafür zu verstehen, daß **nicht eingebundene Regel-Pakete** in der Wissensbasis erkannt werden. Hierbei ist zu bemerken, daß die Regel-Konnektivitäten auch, wie bereits erwähnt wurde, zyklische Strukturen beinhalten können. Da jede Vorwärts-Regel-Alternative als Auslöser solch eines Zyklus in Betracht kommen kann, muß bei der Suche nach allen potentiellen Lösungspfaden sichergestellt werden, daß bei zyklischen Regelsequenzen der Ausgang aus der Abarbeitungsschleife als alternativer Weg erkannt werden kann. Dafür werden alle Vorgänger bei der Überprüfung eines potentiellen Lösungspfades vermerkt, und es wird darauf geachtet, daß ein schon beschrittener Weg nicht wiederholt als unendliche Schleife durchlaufen wird.

c) Überprüfung auf unerwünschte Schleifen bei der Regel-Abarbeitung

Hierbei wird überprüft und als fehlerhaft markiert, ob sich während des Ablaufs **unerwünscht wiederholende** Sequenzen von Regelketten ergeben können. In der Wissensbasis ist, wie bereits mehrfach erwähnt, generell die zyklische Abarbeitung der Regeln zulässig. Dies kommt häufig als ein Teil des Lösungspfades vor, wenn in einem gegebenen Zustand des Systems eine bestimmte Änderung nachdrücklich erzielt werden muß (Bild 5.2, Hinweis 2). Hierbei müssen Varianten vorhanden sein, durch welche diese zyklischen Abläufe wieder verlassen werden können; anderenfalls würde eine Endlosschleife immer wieder ohne möglichen Ausweg durchlaufen. Dies ist "manuell" sehr schwer zu lokalisieren, da zwischen dem ersten Aufruf des fehlerhaften Regel-Paketes und nachfolgenden zyklischen Verweisen abhängig vom Inhalt der Zielliste mehrere Schritte liegen können. Grundsätzlich werden auch für alle möglichen Regel-Zyklen, die einen Ausweg aus dem zyklischen Ablauf beinhalten, warnende Hinweise gegeben, um ggf. unbeabsichtigte zyklische Strukturen in der Wissensbasis erkennbar zu machen.

d) Existenz eines als Folgeregel eingetragenen Vorwärts-Regel-Paketes

Hierbei wird überprüft, ob alle in den Folgelisten der einzelnen Vorwärts-Regel-Alternativen eingetragenen **Namen** von Regel-Paketten tatsächlich auch in der Wissensbasis vorkommen. Insbesondere beim manuellen Löschen eines Regel-Paketes in der Wissensbasis muß gewährleistet sein, daß dessen Name nicht als Folgeregel an anderer Stelle immer noch eingetragen ist. Auch beim Hinzufügen neuer Regel-Alternativen in der Wissensbasis muß sichergestellt werden, daß alle eingetragenen Regel-Paket-Namen jeweils auch in der Folgeliste mindestens einer Regel existieren. Ein Fehler an dieser Stelle würde bei der Abarbeitung der Regeln einen Abbruch der Vorgehensweise durch die Inferenzkomponente verursachen.

e) Überprüfung der rückwärtsverketteten Regeln

Bei rückwärtsverketteten Regeln sind die Textkonstrukte der jeweiligen überlagerten Regel-Bedingung identisch mit denen des Konklusionsteils der unterlagerten Regel-Alternative (Kap. 2.3.1). Wenn durch einen Fehler (z.B. Schreibfehler bei der manuellen Eingabe der Regeln) diese Übereinstimmung der Textkonstrukte nicht mehr gewährleistet ist, dann ist die jeweilige unterlagerte Baumstruktur bei der Abarbeitung der unterlagerten Regel-Pakete nicht mehr erreichbar, da kein Eintritts-Verweis für diese Verknüpfung existiert (Bild 5.2, Hinweis 4). Dies würde daher bei der Abarbeitung der jeweiligen Regel-Bedingungen an dieser Stelle ggf. eine nicht beabsichtigte Dialog-Anfrage veranlassen an Stelle weiterer Überprüfung der unterlagerten Baumstruktur. Aus diesem Grund kann dieser Fehler nicht unmittelbar erkannt werden. Bei der durchgeführten Überprüfung werden alle "losen" Rückwärtsketten selektiert. Außerdem wird zusätzlich in der gesamten Wissensbasis nach solchen Regel-Alternativen gesucht, die als zugeordnete überlagerte Regeln in Frage kommen könnten, weil zwei der insgesamt drei Textelemente identisch sind, und diese als Kandidaten für mögliche überlagerte Regel-Alternativen ausgegeben.

f) Überprüfung der Regel-Alternativen auf ihren Zusammenhang mit dem jeweiligen Regel-Paket

Die Regel-Alternativen in einem Regel-Paket sind durch ihre gleichen Regel-Namen gekennzeichnet (Kap. 2.3.1) und sollen der Übersichtlichkeit halber – ohne gegebene Notwendigkeit von der Inferenzkomponente her – zusammen in der Wissensbasis angeordnet sein. Um bei Erweiterungen der Wissensbasis die Verwendung schon vergebener Namen für neue Regel-Pakete auszuschließen, wird überprüft, ob die verwendeten Namen für Regel-Pakete jeweils nur zusammenhängend vorkommen und gegebenenfalls eine Warnung ausgegeben.

g) Redundante Regel-Alternativen

Dieser Fehler taucht nicht selten auf, da die Regel-Alternativen in vielen Fällen gleiche Regel-Bedingungen aufweisen, die sich nur in ihren logischen Erwartungswerten unterscheiden. Es ist zu beachten, daß diese Fehlerart keine Konnektivitäts-Probleme verursacht und die Abarbeitung nicht beeinflußt. Trotzdem ist es wichtig, diesen trivialen Fehler zu melden.

5.1.3 Zusammenfassung einiger Aspekte bei der Realisierung dieser Komponente

Die oben eingeführten Überprüfungen wurden als umfangreiches Prolog-Programm implementiert und bisher schon an **mehreren** Wissensbasen mit verschiedener Regelanzahl (bis 2700 Regeln, siehe [KRO-97]) und unterschiedlichen Strukturen (zyklische bzw. baumförmige Strukturen, siehe Bild 5.2 und

[KRO-97c]) eingesetzt. Es wurden tatsächlich diverse Fehler, die bis dahin auch beim praktischen Einsatz des jeweiligen Systems nicht bemerkt worden waren, aufgedeckt. Die einzelnen Prüfroutinen liefern teilweise Ergebnisse, die in Zusammenhang und Kombination mit anderen Fehlermeldungen einen Aufschluß über die **Fehlerquelle** angeben können.

Die erstellte Konnektivitätsdatei wird außerdem auch als Grundlage für die **Erklärungskomponente** verwendet, wenn Auskunft über eine nicht eingetretene Vermutung gegeben werden soll (Kap. 6.2).

Der grundsätzlich erhebliche **Zeitbedarf** für die Überprüfungen ist erwartungsgemäß stark von der Größe der zu überprüfenden Wissensbasis abhängig, fällt allerdings insofern nicht so stark ins Gewicht, als die Konsistenzprüfung nur bei einer Änderung der Wissensbasis relevant ist und "off line" erfolgt.

In nächsten Abschnitt wird die zusätzlich realisierte **Wissenserwerbskomponente** vorgestellt, die u.a. eine syntaktisch korrekte Eingabe und Änderung der Regeln in der Wissensbasis erleichtert und einige der weniger zeitaufwendigen Konsistenzprüfungen dabei direkt mitberücksichtigt.

5.2 Darstellung und Änderung bzw. Erstellung von Wissensbasen

Wissenserwerbskomponenten sind für verschiedene in der Literatur beschriebene Expertensysteme bereits vorgestellt worden: So ermöglicht z.B. der Einsatz von Tabellen bei weniger aufwendigen Problemen eine übersichtliche Eingabe der Wissensbasis, die dann auch selbsttätig in Regel-Syntax umgewandelt werden kann [THO-86]. Auch die Anwendung von Petrinetzen wird für umfangreichere Anwendungen vorgeschlagen [SHI-95]; dieses Modell umfaßt auch eine graphische Übersicht über die Verknüpfungen, kann aber bei einer stark "vermaschten" Wissensbasis nicht ohne Verlust der Transparenz angewandt werden [FOU-97]. Der Einsatz eines textsensitiven Editors bei einigen Expertensystem-Shells (z.B. [GEN-89]) bietet eine sehr komfortable Möglichkeit zur Wissensangabe, allerdings fehlt ein Überblick über die gesamte Wissensstruktur. Die dort realisierte Editorkomponente ermöglicht eine natürlichsprachige Formulierung der Regeln; hierbei werden anhand eines intern angelegten Diktionsars – bestehend auch aus den Attributen der am betrachteten Prozeß beteiligten Parameter – die einzelnen eingegebenen Worte auf ihre Zulässigkeit hin überprüft und entsprechende Korrektur-Vorschläge unmittelbar beim Editieren bereitstellt, so daß anschließend eine interne Ablage in der Wissensbasis in syntaktisch korrekter Form erfolgt. Grundsätzlich ist ersichtlich, daß für komplex strukturierte Wissensbasen die Anforderung nach einem einheitlichen Konzept **sowohl** für das **Editieren** von Regeln **als auch** für die **Visualisierung** ihrer Beziehung untereinander besteht.

Die für das hier betrachtete Netzwiederaufbau-System entwickelte Lösung wird im Folgenden zunächst als Konzept erläutert und anschließend in ihrer Funktionalität beschrieben.

5.2.1 Anforderungen und Konzept

Folgende **Anforderungen** sollten dabei erfüllt werden:

- a) Weitestmögliche natürlichsprachige Interaktion mit dem Benutzer, ohne daß detaillierte Kenntnisse bezüglich der Inferenzmechanismen und vereinbarter interner Konventionen (z.B. Regel-Syntax) notwendig sind.
- b) Überschaubare Darstellung aller potentiellen Lösungspfade, die sich aus den implementierten Regeln zusammen mit den zugrundeliegenden Inferenzmechanismen ergeben können; dies gilt insbesondere auch für Vermischungen und Zyklen.
- c) Transparenz durch hierarchische Abstufung verschiedener Detaillierungsgrade bei der Darstellung.
- d) Anwendung einiger Konsistenzprüfungsfunktionen (siehe auch Kap. 5.1.2) bereits während der Regel-Implementation.
- e) Verwendbarkeit über die hier betrachtete Anwendung "Netzwiederaufbau" hinaus auf beliebige Wissensbasen.

Bezüglich der Darstellungsebenen wurde ein **dreistufiges Konzept** gewählt, welches die Strukturebenen der Wissensbasis reflektiert, Bild 5.3.

1. Die oberste hierarchische Ebene umfaßt die einzelnen Regel-Module (entsprechend ihrer inhaltlichen Zusammengehörigkeit können die Regeln großer Wissensbasen zu Modulen zusammengefaßt werden [KRO-92]) und deren Verknüpfung untereinander entsprechend den Querverweisen der jeweils enthaltenen Regeln, wobei die Darstellung der globalen Verweisstruktur der Verknüpfung der Module untereinander ausreicht.
2. In einer zweiten Visualisierungsebene werden in einem jeweils vom Anwender ausgewählten Modul alle zugehörigen Regel-Pakete inklusive deren Verweisstruktur veranschaulicht. Außer den "internen Regel-Paketen" werden in dieser Darstellung auch diejenigen Regel-Pakete mit einbezogen, die zwar einem anderen Modul angehören, aber durch Verweise mit dem betrachteten Regel-Modul verknüpft sind. Diese "externen Regel-Pakete" werden farblich gekennzeichnet (siehe auch Anhang C).

3. Die dritte hierarchische Ebene betrifft die rückwärtsverketteten Regel-Pakete, die jeweils in einer untergeordneten Baumstruktur der Regel-Bedingung einer Vorwärts-Regel untergeordnet sind.

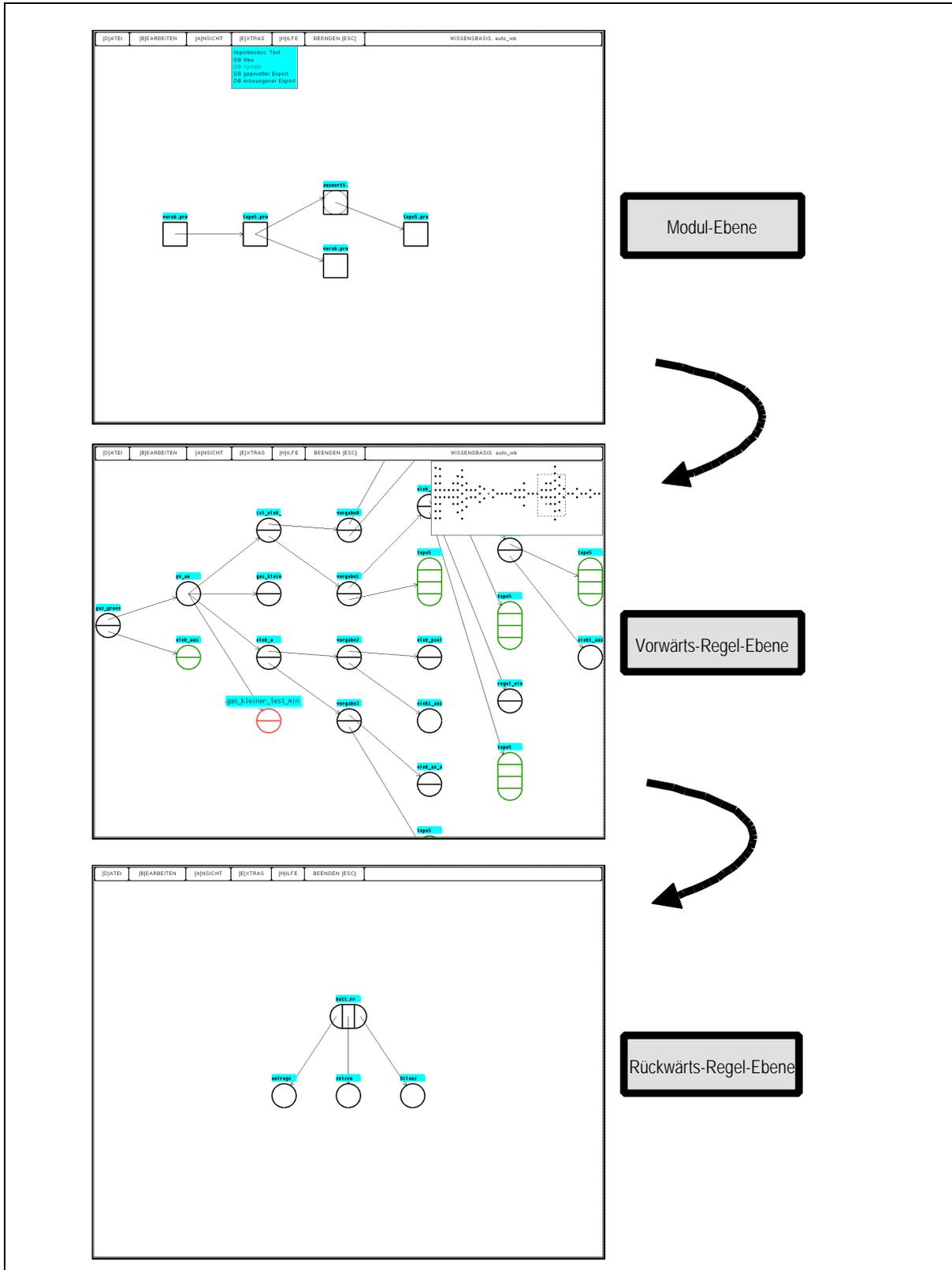


Bild 5.3: Hierarchische Ebenen der Wissensbasis und ihre graphische Darstellung

Wesentliches und innovatives Merkmal aller drei Darstellungsebenen ist, daß die Graphiken auf Basis der bekannten potentiellen Regel-Konnektivitäten unter Rückgriff auf bereits in [KEM-99] für die Darstellung elektrischer Netze und Schaltanlagen angewandte Prinzipien jeweils **automatisch** erzeugt werden; dies erspart jeglichen Eingriff in die Visualisierung bei vorgenommenen Änderungen in der Wissensbasis [SEI-97].

Angeknüpft an die einzelnen Darstellungsebenen wurden entsprechende **Editierfunktionen** implementiert, die eine beliebige Auswahl der einzelnen Elemente der Wissensbasis (Module, Regeln bzw. Bedingungen) und deren komfortable Änderung und Erweiterung ermöglichen. Die Interaktion mit dem Benutzer erfolgt in einem vorgegebenen Dialograhmen und erlaubt eine dynamisch angepaßte Auswahl der jeweils zulässigen Editorfunktionen, ohne daß dieser detaillierte Kenntnisse in Bezug auf die Regel-Syntax besitzen muß (Anhang C). Die vereinbarten Konventionen im Bezug auf die Regel-Syntax (Anhang A) sowie die Verknüpfung der Regeln (Kap. 5.1.1) werden selbstverständlich eingehalten.

Die **Konsistenz** (Kap. 5.1) wird nach jeder Änderung selbsttätig überprüft und die geänderte Wissensbasis als Ergebnis graphisch dargestellt. Dies erfordert, daß als Grundlage für die Konsistenzprüfung sowie die Darstellung der neuen bzw. geänderten Wissensbasis die entsprechende Konnektivitätsdatei jeweils neu erzeugt werden muß. Abschließend wird die geänderte Wissensbasis in syntaktisch richtiger Form zurückgespeichert.

Dieses in sich geschlossene Konzept erlaubt eine schrittweise Erweiterung bzw. auch die erstmalige Erzeugung einer Wissensbasis ohne fundiertes Wissen über die Inferenzmechanismen.

5.2.3 Funktionsbeschreibung

Die Funktionen für die Darstellung, Änderung, Überprüfung und Speicherung der jeweiligen Wissensbasis können über eine Menüsteuerung angewählt werden. Die Eingabe bzw. Änderung der Regeln selbst erfolgt nach Maus-Anwahl über ein dafür geöffnetes Fenster, in welches die erforderlichen Bestandteile – Textkonstrukte, logischer Erwartungswert, Verweis auf Dienstprogramme usw. – eingetragen werden. Ausführliche Beispiele für die auf den drei hierarchischen Ebenen implementierten Möglichkeiten finden sich im Anhang C. Die vorgenommenen Änderungen bzw. Erweiterungen der Wissensbasis werden als neue Symbole bzw. Einträge in den Regeln zuerst auf der jeweiligen Detaillierungsebene markiert und intern in einem Verarbeitungsspeicher dynamisch verwaltet. Am Ende einer Bearbeitungsphase können sie dann separat als Kette von Aktionen für die weiteren Arbeitssitzungen zwischengespeichert oder endgültig in die Wissensbasis übertragen werden [HIN-99].

Für die Einhaltung der Konsistenz der Wissensbasis trotz vorgenommener Änderungen sorgen mehrere interne Bearbeitungsschritte:

- a) Zunächst wird der Verarbeitungsspeicher, in welchem alle Änderungen protokolliert sind, auf redundante bzw. widersprüchliche Anweisungen überprüft (Plausibilitäts-Kontrolle). So wird z.B. bei mehrfacher Namensänderung eines Regel-Paketes nur der letzte eingegebene Eintrag als gültig vermerkt.
- b) In einem weiteren Schritt werden die Änderungen an den Wissensbasis-Elementen, die intern in einen relationalen Datenbestand eingeordnet sind, berücksichtigt. Diese Art der internen Darstellung gewährleistet eine geordnete Anwendung der Benutzer-Aktionen an den verschiedenen Teilen der Wissensbasis. Beispielsweise werden beim Löschen eines Moduls auch gleichzeitig alle Regel-Pakete, die sich in diesem Modul befinden und eine Relation zu diesem gelöschten Datenelement beinhalten, mit gelöscht und die Eintritts- und Austritts-Verweise entsprechend angepaßt.
- c) Die Änderungen werden dann in den bereits extrahierten dynamischen Datenbestand der Wissensbasis eingetragen; die geänderten Daten werden einer unmittelbaren Prüfung bezüglich der Eindeutigkeit der für Module und Regel-Pakete verwendeten Namen, der Gültigkeit der jeweiligen Informationsquelle bei den Regel-Bedingungen sowie der Verknüpfung der Regeln in Vorwärts- und Rückwärtsrichtung (siehe *a*, *d*, *e*, und *f* in Kap. 5.1.2) unterworfen. Eine umfassende Überprüfung kann aber zusätzlich durch Anwendung der vollständigen in Kap. 5.1 beschriebenen Konsistenzüberprüfungskomponente vorgenommen werden
- d) Die Ergebnisse dieser Überprüfungen werden dann ggf. in Form eines Fehlerprotokolls an den Anwender weitergereicht. Dieser kann daraufhin weitere Änderungen bzw. Korrekturen in der Wissensbasis vornehmen oder die noch inkonsistente Wissensbasis zunächst als Zwischenversion separat – d.h. ohne die ursprüngliche Wissensbasis zu überschreiben – abspeichern und die Bearbeitung später fortsetzen, bis eine fehlerfreie geänderte Wissensbasis vorliegt, welche dann die ursprüngliche Version ersetzt.
- e) Bei der endgültigen Übernahme der Änderungen und dem Zurückschreiben in die Wissensbasis werden alle statischen Moduldateien aktualisiert und eine Referenzdatei mit den neu eingefügten bzw. geänderten Modulnamen selbsttätig neu erstellt.
- f) Für die Visualisierung der geänderten Wissensbasis wird die aktuelle Konnektivitäts-Datei selbsttätig erstellt.

Abschließend können dann durch einen weiteren Aufruf die vorgenommenen Änderungen insgesamt graphisch dargestellt werden.

Der somit implementierte Vorgang beim Wissenserwerb begünstigt die schrittweise Erstellung bzw. Änderung der Wissensbasis durch die beschriebenen Zusatzfunktionen – insbesondere auch bei der Unterteilung eines umfangreichen Vorhabens in unterschiedliche Entwicklungsphasen – und bietet damit einen in sich geschlossenen Arbeitsprozeß [HIN-99].

Im Gegensatz zur soeben vorgestellten Wissenserwerbskomponente, die nicht in den aktuellen Ablauf des Expertensystems integriert ist, wird an die im nächsten Kapitel beschriebene **Dialogkomponente** des Expertensystems die Anforderung gestellt, **während des Inferenzprozesses** den Ablauf transparent zu machen, indem auf Wunsch situations-angepaßte Erklärungen bereitgestellt werden.

