Erschienen in: A. Günter: Beiträge zum Workshop "Modelle beim Konfigurieren, LKI-M-93/3, Labor für Künstliche Intelligenz, Universität Hamburg, Bodenstedtstr. 16, D-22765 Hamburg.

IDAX – Ein Modell der strukturorientierten Konfiguration*

Jürgen Paulokat
FB Informatik, Universität Kaiserslautern
Postfach 3049
D-67653 Kaiserslautern
E-mail: paulokat@informatik.uni-kl.de

1 Einleitung

Die Lösung einer Konfigurationsaufgabe in technischen Domänen besteht aus einer Menge von Bauteilen, die miteinander verträglich sind und in ihrem Zusammenspiel die gegebenen Anforderung erfüllen. Eine gängige Vorgehensweise bei der Suche nach einer Lösung ist die schrittweise Spezialisierung einer abstrakten Aufgabenstellung oder ihre Zerlegung in Teilaufgaben. Ein Konfigurationssystem, das diese Vorgehensweise unterstützt, muß Wissen enthalten, wie eine Aufgabe spezialisiert oder in Teilaufgaben zerlegt werden kann, welche konkreten Bauteile zur Erfüllung einer ausreichend detaillierten Teilaufgabe verwendet werden können und ob alle Teile einer Lösung miteinander verträglich sind. Aufgrund dieses Wissens kann eine konsistente Lösung durch Tiefensuche hergeleitet werden.

Neben der Konsistenz spielt die Qualität der erzeugten Konfiguration eine zentrale Rolle, so daß der Suchprozeß nicht mit einer beliebigen Lösung terminieren darf, sondern eine Lösung liefern muß, die in der Nähe des Optimums liegt. Die Bereitschaft, eine suboptimale Lösunge zu akzeptieren, hat mehrere Gründe. Zum einen kann meistens keine globale Präferenzrelation zur Bewertung der Güte einer Lösung angegeben werden. Zum anderen setzt eine globale Präferenzrelation voraus, daß zunächst alle konsistenten Konfigurationen bestimmt werden, bevor aus diesen die optimale Lösung ausgewählt werden kann. In vielen praktischen Anwendungen ist es aufgrund der großen Anzahl und dem Herleitungsaufwand jedoch unmöglich, alle Lösungsalternativen zu bestimmen.

Ein realistischer Ansatz kann die Steuerung der Tiefensuche durch lokale Präferenzrelationen sein (vgl. auch [Gün91]). Hierbei wird für jeden Entscheidungspunkt im Suchprozeß eine Präferenzrelation über den Alternativen spezifiziert, mit deren Hilfe die Tiefensuche gesteuert wird. Da hierbei jedoch die lokale Optimierung im Vordergrund steht, kann bei dieser Vorgehensweise nicht unbedingt sichergestellt werden, daß der Lösungsprozeß mit einem globalen Optimum terminiert.

Im folgenden wird die objekt- und strukturorienterte Modellierung und Operationalisierung von Konfigurationswissen im IDAX-System beschrieben. Im Mittelpunkt der Wissensrepräsentation steht eine Begriffstaxonomie zur Modellierung der Konfigurationsobjekte und deren strukturellen Beziehungen. Bei der Operationalisierung werden Konfigurationsaufgaben durch Ziele beschrieben. Die Konfigurationsschritte werden durch Operatorschemata spezifiziert, mit deren Hilfe problemklassenspezifisch festgelegt wird, welche Teile des Modells zur Lösung einer bestimmten Konfigurationsaufgabe verwendet werden sollen.

*Die hier vorgestellte Arbeit wurde teilweise durch die Deutsche Forschungsgemeinschaft, SFB 314: „Künstliche Intelligenz - Wissensbasierte Systeme", Projekt X9 gefördert.

2 Repräsentation der Domäne in einer Begriffsheterarchie

Ein Modell der Domänenobjekte und ihrer Beziehungen, das im folgenden als Objektwissen bezeichnet wird, wird in der statischen Wissensbasis aufgebaut. Zur Definition der Objekte ist die Is-A-Relation vorgegeben. Mit ihrer Hilfe können neue Objekte ausgehend von bereits vorhandenen modelliert werden, indem deren Eigenschaften ererbt und ggf. eingeschränkt oder neue Eigenschaften hinzugefügt werden, so daß ein Objekt, das in der Is-A-Beziehung zu einem weiteren Objekt steht, eine Spezialisierung von diesem darstellt. In IDAX kann jedes Objekt zu beliebig vielen Objekten in der Is-A-Beziehung stehen (Multiple Inheritance). Die hieraus resultierende Struktur wird als Begriffsheterarchie bezeichnet. Bei dem Aufbau einer Heterarchie wird mit Hilfe von Konsistenzregeln sichergestellt, daß die Eigenschaftsausprägungen des erbenden Objekts in der Schnittmenge der Eigenschaftsausprägungen der Vorgängerobjekte liegt. Falls die Schnittmenge der Eigenschaftsausprägungen verschiedener Objekte leer ist, können sie nicht gleichzeitig Oberobjekte desselben Objekts sein.

Die zweite Relation, die ebenfalls ein fester Bestandteil der statischen Wissensbasis ist, ist die Has-Parts-Beziehung. Durch sie können die Objekte der Begriffsheterarchie in eine dekompositionelle Beziehung gesetzt werden. Sie beschreibt die Zerlegung einer Baugruppe in ihre Teilkomponenten. Hierbei kann für jede Teilkomponente eine untere und obere Anzahl ihres Auftretens in der Baugruppe festgelegt werden. Im folgenden bezeichnen wir ein Objekt, das Oberobjekt in einer Is-A-Beziehung ist oder das durch eine Has-Parts-Relation in Teilkomponenten zerlegt wird, als abstrakt oder komplex. Die übrigen sind atomare Objekte dar.

Daneben können zur Erstellung der Wissensbasis beliebige neue Relationen definiert und zur Spezifikation von zusätzlichen Beziehungen zwischen Domänenobjekten verwendet werden. Diese Beziehungen können bei der Definition von Operatorschemata (vgl. Abs. 3.2) benutzt werden. In den Operatorschemata wird die Verarbeitung dieser Relationen bei der Suche nach einer Lösung einer Konfigurationsaufgabe und somit ihre Bedeutung für den Konfigurationsprozeß festgelegt.

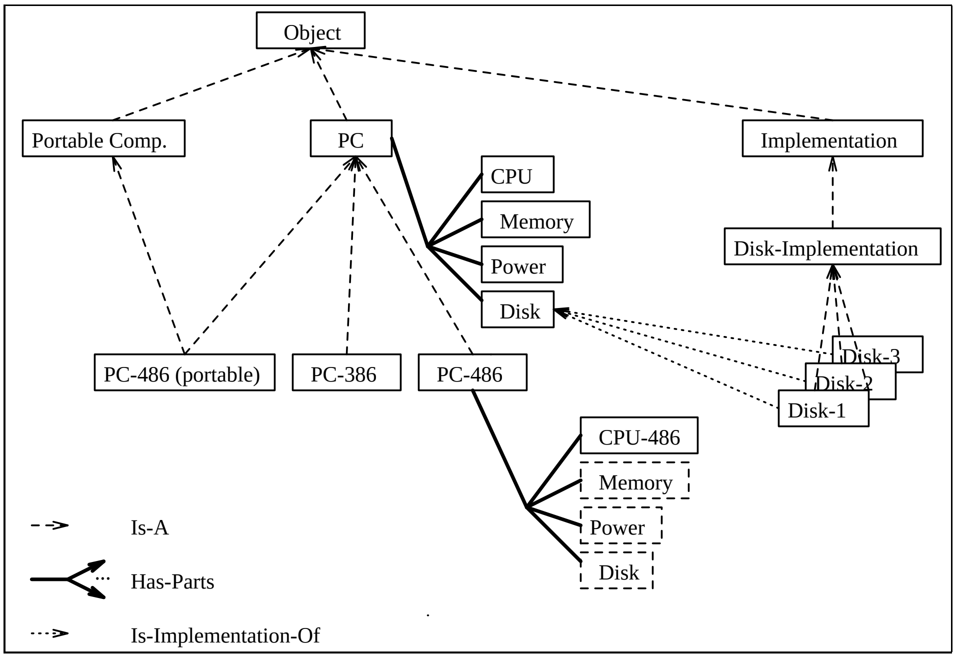

Abbildung 1: Ausschnitt aus dem Modell der PC-Konfiguration

Abb. 1 zeigt einen Ausschnitt der Begriffsheterarchie einer Rechnerdomäne, die mit Hilfe der oben beschriebenen Konzepte realisiert wurde. In diesem Beispiel repräsentiert *PC* ein abstraktes Objekt, das durch die Objekte *PC-386*, *PC-486* und *PC-486 (portable)* spezialisiert wird. Diese Objekte stehen zu *PC* in einer Is-A-Beziehung. *PC* und *PC-486* sind aber auch Baugruppen, d.h. sie können in Teilkomponenten zerlegt werden. *PC-486* erbt seine Zerlegung von *PC*, wobei die Komponente *CPU* zu *CPU-486* spezialisiert wird, die ihrerseits in einer Is-A-Beziehung zu *CPU* steht. Diese Beziehung ist jedoch aus Gründen der Übersichtlichkeit nicht dargestellt. Als zusätzliches Mittel zur Strukturierung der Domäne wird im Beispiel die Relation *Is-Implementation-Of* verwendet. Mit ihrer Hilfe werden die Festplattenspeicher *Disk-1*, *Disk-2* und *Disk-3*, die reale Bauteile repräsentieren sollen, in Beziehung zu dem abstrakten Objekt *Disk* gesetzt. Aufgrund dieser Beziehung können während des Konfigurationsprozesses die konkreten Bauteile für ein abstraktes Objekt bestimmt werden.

3 Ein Modell der strukturorientierten Konfiguration

Im IDAX-System wird ein Konfigurationsprozeß mit der Aufgabe gestartet, ein abstraktes Objekt zu konfigurieren. Im Vordergrund der strukturorientierten Vorgehensweise steht die schrittweise Konkretisierung abstrakter Objekte, bis schließlich konkrete Bauteile ausgwählte werden können, die die Anforderungen der initialen Aufgabenstellung und der gewählten Konkretisierungen erfüllen. In diesem Prozeß der schrittweisen Spezialisierung und Auwahl werden Konfigurationsaufgaben durch Ziele beschrieben, auf die ein Operator angewendet werden muß, um die damit verbundene Aufgabe zu lösen. Ein Operator kann bei seiner Anwendung neue Teilziele herleiten, die Unteraufgaben festlegen. Die gesamte Konfigurationsaufgabe ist gelöst, wenn auf jedes Ziel ein Operator angewendet wurde.

Die Modellierung eines Ziels oder Operators wird in zwei Schritten durchgeführt. Zunächst werden unabhängig von der vorliegenden Domäne Ziel- und Operatorschemata definiert, aus denen mit Hilfe des Objektwissens der statischen Wissensbasis Ziel- und Operatorinstanzen hergeleitet werden. Die Definition von Schemata und die Ableitung von Instanzen wird zu verschiedenen Zeitpunkten im Lebenszyklus eines Expertensystems durchgeführt. Operator- und Zielschemata werden im Normalfall bereits vor der Repräsentation des Objektwissens definiert und legen dessen Operationalisierung bei Lösung einer Konfigurationsaufgabe fest. Nach der Modellierung der Domäne mit Hilfe der Konzepte der statischen Wissensbasis werden Ziel- und Operatorinstanzen zur Herleitung der Lösung für eine Konfigurationsaufgabe verwendet.

3.1 Zielschemata

Durch Zielschemata werden unabhängig von einer konkreten Domäne die typischen Konfigurationsaufgaben in parametrisierter Form beschrieben. Im IDAX-System sind die Zielschemata *Configured(?Object)*, *Refined(?Object)*, *Specialized(?Object)*, *Decomposed(?Object)*, *Specified(?Object)*, *Value-Is-Specified(?Object ?Attribut)* und *Implementation-Is-Selected(?Object)* vordefiniert. Im Zusammenhang mit Ziel- und Operatorschemata wird durch einen Bezeichner, der mit ? beginnt, eine Variable eingeführt, die bei dessen Instanziierung an eine Konzeptinstanz der statischen Wissensbasis gebunden wird und dadurch eine konkrete Aufgabe festlegt, z.B. einen PC zu konfigurieren (*configured(PC)*).

Dem Modell der Konfiguration, das durch diese Zielschemata festgelegt wird, liegt die Vorstellung zugrunde, daß die Instanz eines atomaren Objekts der statischen Wissensbasis konfiguriert bzw. das Ziel *configured(?Object)* erreicht ist, sobald seine Eigenschaftsausprägungen festgelegt (*Value-Is-Specified(?Object ?Attribut)*) und eine geeignete Realisierung ausgewählt ist (*Implementation-Is-Selected(?Object)*). Der Übergang von der abstrakten Konfigurationsaufgabe zur Festlegung der Attributausprägungen und der Auswahl einer Realisierung erfolgt dabei über die abstrakte Aufgabe, das betrachtete Objekt zu spezifizieren (*Specified(?Object)*). Diese zeitliche Abfolge der Konfigurationsaufgaben bei der Bearbeitung eines atomaren Objektes ist in Abb. 2 dargestellt.

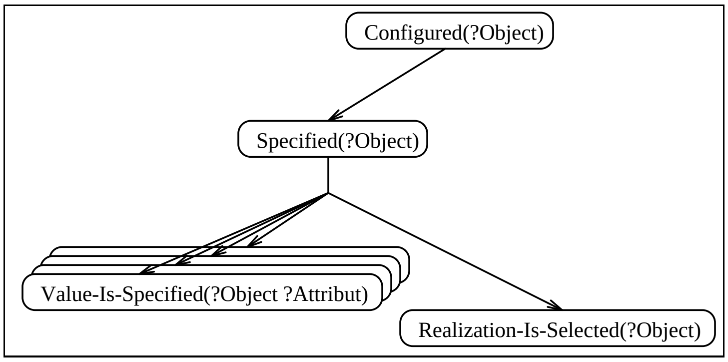

Abbildung 2: Zeitliche Abfolge der Ziele bei der Konfigurierung eines atomaren Objektes

Bezieht sich eine Konfigurationsaufgabe auf ein komplexes Objekt, so muß nach der oben beschriebenen Spezifikation eine Zerlegung entlang der Has-Parts-Beziehung (*Decomposed(?Object)*) oder eine Spezialisierung (*Specialized(?Object)*) durchgeführt werden. Wie bei der Spezifikation der Eigenschaften und der Auswahl der Realisierung wird auch hier beim Übergang ein Zwischenziel (*Refined(?Object)*) eingeführt (vgl. Abb. 3). Die Spezialisierung und die Zerlegung stellen alternative Vorgehensweisen bei der Lösung einer Konfigurationsaufgabe dar, so daß im konkreten Fall eine Auswahl zwischen beiden Verfeinerungsmöglichkeiten getroffen werden muß.

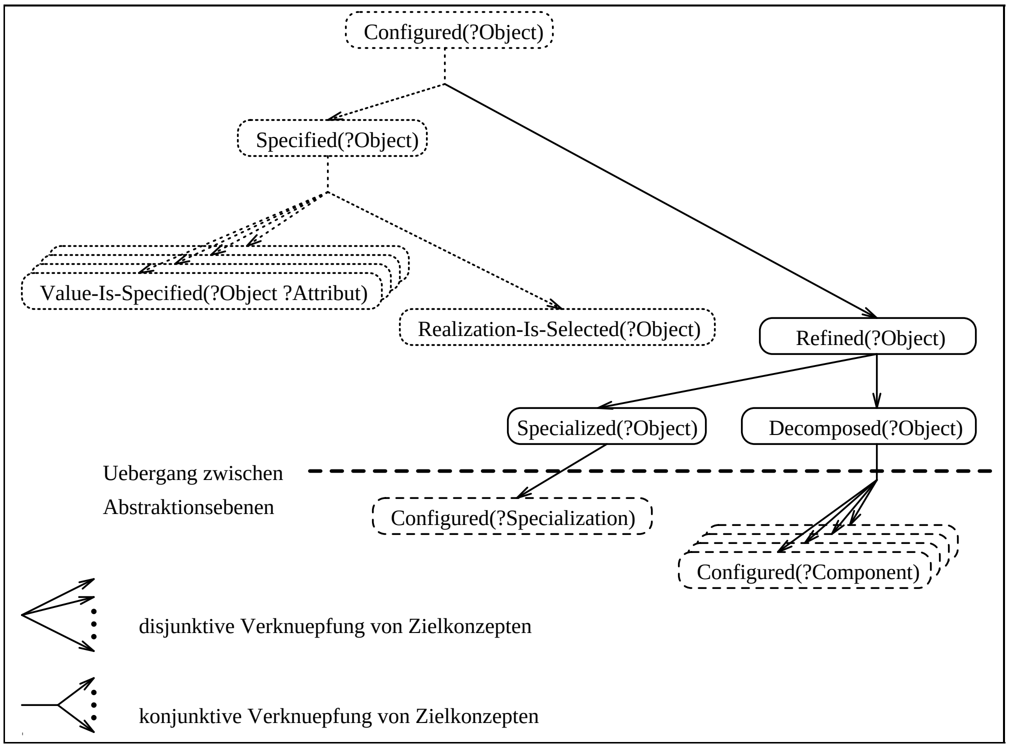

Abbildung 3: Zeitliche Abfolge der Ziele bei der Konfigurierung eines komplexen Objektes

Im Falle der Spezialisierung wird die ursprüngliche Konfigurationsaufgabe durch die Aufgabe, ein Unterobjekt bzgl. der Is-A-Beziehung zu konfigurieren (*configured(?Spezialization)*), ersetzt. Im allgemeinen gibt es bei der Spezialisierung einer Aufgabe mehrere Möglichkeiten, so daß auch hier eine Auswahl getroffen werden muß. Die gewählte Alternative muß mit den Attributausprägungen verträglich sein, daher ist es in den meisten Fällen sinnvoll, zuerst die Attributausprägungen zu bestimmen, bevor ein Verfeinerungsschritt durchgeführt wird. Da jedoch zeitliche Ordnungsbeziehungen, die auf der Ebene der Zielschemata spezifiziert werden, statischen Charakter haben und immer eingehalten werden, bleibt die Reihenfolge der Bearbeitung der Spezifikations- und Verfeinerungsaufgaben unspezifiziert. Hieraus ergeben sich bei der Lösung einer Konfigurationsaufgabe Mengen von Zielen, die in beliebiger Reihenfolge abgearbeitet werden können. Für diese Ziele können Abarbeitungsstrategien definiert werden, die dynamisch eine Reihenfolge festgelegen, so daß von der oben beschriebenen Default-Vorgehensweise abgewichen werden kann.

Bei der Spezialisierung wird neben der Festlegung von Randbedingungen eine abstrakte Aufgabe durch eine speziellere ersetzt. Bei der Zerlegung werden im allgemeinen mehrere Teilziele, die eigenständige Konfigurationsaufgaben für die Teilkomponenten beschreiben, neu in den Konfigurationsprozess aufgenommen. Diese Ziele stehen ungeordnet nebeneinader. Somit handelt es sich bei der Lösung einer Konfigurationsaufgabe um einen rekursiven Prozeß, der mit der Aufgabe, ein Objekt zu konfigurieren gestartet wird und im Falle eines komplexen Objektes diese Aufgabe auf die Konfiguration einer Spezialisierung oder der Komponenten zurückführt.

3.2 Operatorschemata

Im letzten Abschnitt wurden die Beziehungen zwischen Zielschemata dargestellt. Für die Lösung einer Konfigurationsaufgabe müssen zusätzlich die relevanten Informationen der statischen Wissensbasis in den Konfigurationsprozeß übernommen werden. Dieser Wissenstransfer wird mit Hilfe der Operatorschemata beschrieben, bei deren Definition die Menge der Zielschemata festgelegt wird, für die sie relevant sind.

Neben den vordefinierten Zielschemata stellt das IDAX-System auch eine Menge von Operatorschemata zur Verfügung, so daß für jedes Zielschema mindestens ein relevantes Operatorschema vorhanden ist. Zu dieser Menge gehören *Configure(?Object)*, *Refine-By-Specialization(?Object)*, *Refine-By-Decomposition(?Object)*, *Decompose(?Object)*, *Specialize(?Object)*, *Specify(?Object)*, *Choose-Implementation(?Object)* und *Choose-Value(?Attribut ?Value)*. Beispielsweise ist das Operatorschema *Configure(?Object)* relevant für das Zielschema *Configured(?Object)*. Daneben wird bei der Definition eines Operatorschemas auch beschrieben, welche Informationen der statischen Wissensbasis bei seiner Instanzierung verwendet werden sollen. Bei der Instanzierung von *Configure(?Object)* wird berücksichtigt, ob das zu konfigurierende Objekt komplex, abstrakt oder atomar ist und ob es Attribute besitzt. Falls es sich um ein komplexes oder abstraktes Objekt handelt, wird die Zielinstanz *Refined(...)* erzeugt. Falls das Objekt Attribute oder konkrete Realisierungen besitzt, wird die Zielinstanz *Specified(...)* angelegt. Diese Zieinstanzen werden jedoch zunächst von der Operatorinstanz eingekapselt. Sie werden erst dann in den Konfigurationsprozeß aufgenommen, wenn diese Operatorinstanz für die Bearbeitung eines Ziels ausgwählt und darauf angewendet wird. Dies muß jedoch nicht der Fall sein, da für ein Ziel mehrere anwendbare Operatorinstanzen vorliegen können, von denen immer höchstens eine ausgewählt sein kann.

Bei der Instanzierung von *Configured(?Object)* gibt es nur eine Möglichkeit, wobei die gebildete Instanz, insbesondere deren Unterziele, von der Struktur der statischen Wissensbasis abhängen. Es ist jedoch auch möglich, daß von einem Operatorschema zur Bearbeitung einer Aufgabe mehrere alternative Instanzen abgeleidet werden können. Dies ist der Fall bei *Specialize(?Object)*, wo für jede Spezialisierung des betrachteten Objekts der statischen Wissensbasis eine Instanz des Operatorschemas erzeugt wird. Durch die Auswahl einer Operatorinstanz während des Konfigurationsprozesses wird die Festlegung auf eine der Spezialisierung getroffen.

4 Repräsentation des Suchraums bei der strukturorientierten Konfiguration

Der Konfigurationsprozeß wird durch ein Zielzur Konfigurierung eines Objekts, z.B. *Configured(PC)*, das in die dynamische Wissensbasis eingetragen wird, gestartet. Nachdem eine Zielinstanz neu in die dynamische Wissensbasis eingetragen wurde, ist es offen und kann ab diesem Zeitpunkt bearbeitet werden. Zu seiner Bearbeitung werden alle Instanzen der relevanten Operatorschemata gebildet und in seine Konfliktmenge eingetragen. Aus einer Konfliktmenge muß durch eine Entscheidung eine Operatorinstanz ausgewählt werden. Der ausgewählte Operator wird auf das Ziel angewendet wodurch die von ihm erzeugten Teilziele in die dynamische Wissensbasis aufgenommen werden. Gleichzeitig werden die Teilziel als offen und das betrachtete Ziel als bearbeitet markiert.

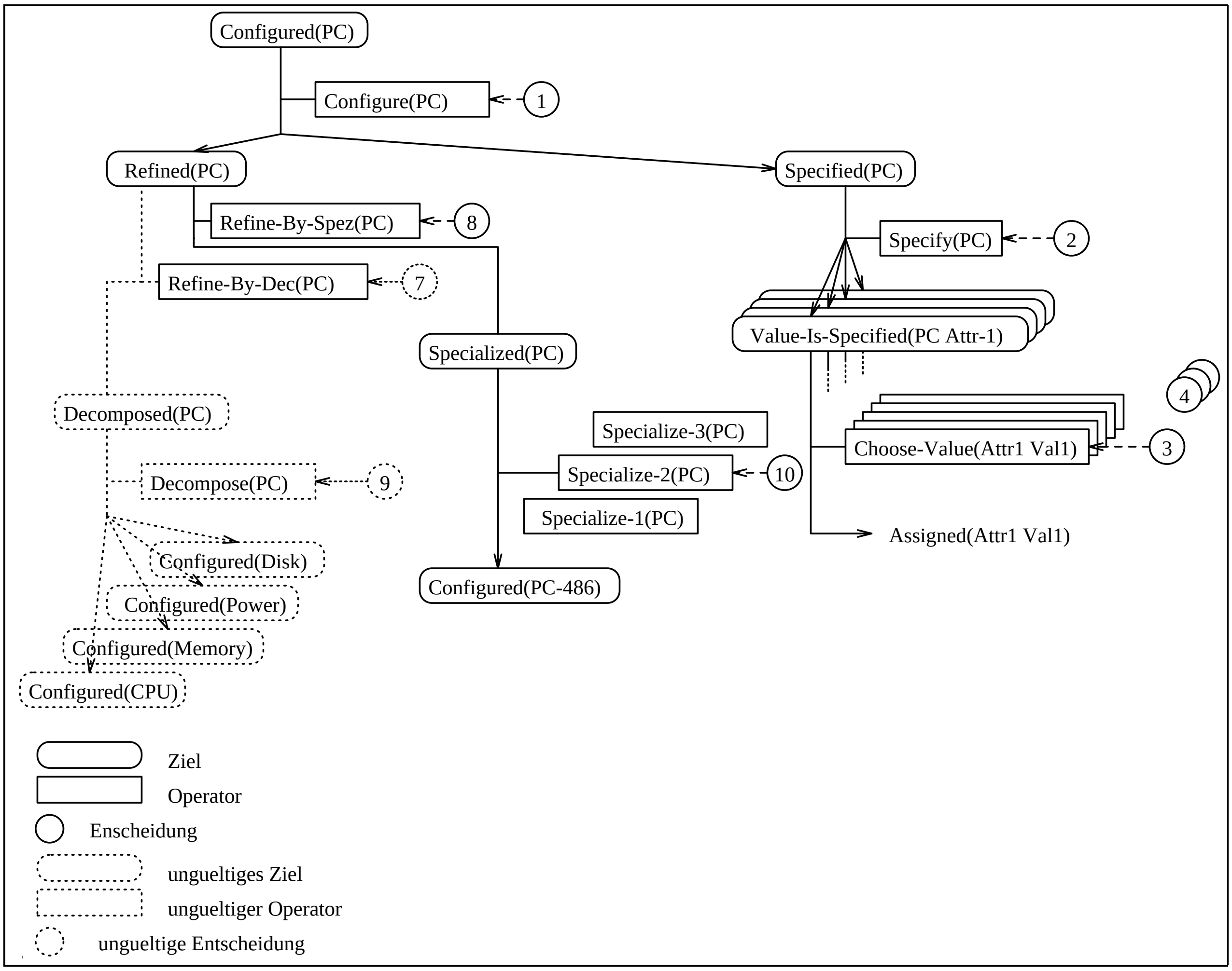

Abbildung 4: Auschnitt aus dem Suchraum bei der Konfigurierung eines PC's

Im Beispiel aus Abb. 4 besteht die Konfliktmenge des Ziels *Configured(PC)* aus der Operatorinstanz *Configure(PC)*, die mit der Entscheidung D_1 ausgewählt wurde, woraus die Zerlegung in die Teilziele *Refined(PC)* und *Specified(PC)* resultiert. Hierbei handelt es sich um eine entartete Konfliktmenge, da sie nur eine Alternative enthält. Ein Beispiel für eine Konfliktmenge mit einer Wahlmöglichkeit, ergibt sich aus der Berabeitung des Ziels *Specialized(PC)*. Hier wurde durch D_{10} eine Entscheidung zugunsten eines 486er PC's getroffen, woraus das Ziel *Configured(PC-486)* resultiert.

Bei der Bearbeitung einer Konfigurationsaufgabe können Konflikte auftreten, zu deren Behebung Entscheidungen zurückgezogen werden müssen. Ein Beispiel hierfür ist die Entscheidung D_7, nach deren Rückzug auch alle abhängigen Konfigurationsschritte zurückgenommen werden. Die Gültigkeit von Entscheidungen und ihre Abhängigkeiten werden mit Hilfe eines problemklassenspezifischen TMS's [PR93] verwaltet.

Diese Form der Organisation des Suchraums bietet die Voraussetzung für die gesteuerte Suche nach einer guten Lösung einer Konfigurationsaufgbe. Mit Hilfe von Präferenzrelationen kann ein zu bearbeitendes Ziel, eine Alternative aus der Konfliktmenge oder ein Entscheidungsrückzug, der aufgrund einer Inkonsistenz notwendig wurde, gezielt ausgewählt werden.

5 Abschlußbemerkung

Das beschriebene System ist vollständig in Smalltalk-80 implementiert. Hierbei wurde ein Konfigurationsmodell realisiert, das sich an der Begriffsheterarchie orientiert, indem die darin repräsentierten Objekte, Relationen und Attribute benutzt werden, um die möglichen Konfigurationsschritte zu bestimmen. Aktuelle Arbeiten beschäftigen sich mit der inkrementellen Akquisition von Kontrollwissen während des Einsatzes des Expertensystems. Ferner werden Ansätze zur Unterstützung der fallbasierten Vorgehensweise bei der Konfigurierung realisiert.

Literatur

[Gün91] Andreas Günter. *Flexible Kontrolle in Expertensystemen zur Planung und Konfigurierung in technischen Domänen.* DISKI - Dissertationen zur Künstlichen Intelligenz, Universität Hamburg, infix, St. Augustin, 1991.

[PR93] Jürgen Paulokat und Helmuth Ritzer. Unterstützung der Kontrolle bei der Konfigurierung druch ein problemklassenspezifisches TMS. In: *Expertensysteme 93.* Springer-Verlag, 1993.