专家系统开发环境ESDEN中多级知识库的实现

田在勤　唐雪飞

（成都电讯工程学院）

摘　要

元知识的运用对提高专家系统性能起重要作用，是新一代专家系统的主要特点之一。本文所述的多级知识库是对元知识在专家系统中的应用所进行的一次尝试。文中论述了多级知识库的设计思想，给出了领域知识与元知识的表达形式，并介绍了Turbo Prolog语言的具体实现。

一　引　言

开发专家系统最初所用的工具，可以说是Lisp、Prolog等一类符号运算语言。也有用Fortran、C这样的通用程序设计语言作为工具的。但是，很快人们就发现，用这些语言开发专家系统，一切都要从头开始，费时，费力。于是，专家系统的开发者们开始研究更高一级的开发工具。一种更好的方法是，从原有的领域专家系统中移走已有领域知识，把剩下的推理机及其辅助设施作为开发工具。基于这种策略，人们开发了大量的称为骨架系统的工具系统，如Emycin(由Mycin系统而来)，KAS(由PROSPECTOR而来)及S.1等。由于所有的设施都是事先存在的，开发时只要不加修改地照搬过来，就能构成新系统。显然，新系统的开发容易得多。但也正因为所有的设施都是事先存在、固定不变的，这又限制了它的通用性，旧的骨架系统也许对新的任务不合适（这是它的致命弱点），系统中推理策略也许不能充分地与新问题的求解方式相匹配，旧的描述规则语言也许对新任务不能描述，等等。

鉴于此，专家系统的研究者们又开始向新的目标进军，研制更加通用的开发工具，通用知识表示语言和组合开发工具。使用通用知识表示语言的典型系统有ROSIE，OPS5，HEARSAY-Ⅱ等。这种语言本身不存在任何特定的推理机制和知识库，允许用元规则描述一大类与其他知识相独立的控制知识。因此，这种语言的可应用领域相当广泛，但却是以其使用过程困难为代价。组合开发工具不像通用语言，它提供了知识库和推理控制策略，但也不像骨架系统那样有针对性和单一性。它提供的是知识库和推理机的预构件，及建立使用这些预构件的辅助设施。开发者可以根据任务的需要选择预构件的组合，从而达到多用性的目的。AGE是组合开发工具的典型代表，它的目的是提供一种环境，用户在其中可以选择并指定多种知识表示和处理方式。例如，AGE用户可以建立并运行一种类似于Emycin或HEARSAY-Ⅱ所建造的系统。值得注意的是，虽然AGE有多种处理方式，但每种方式还是事先固定好的。即使有多种方式，但对于具体任务，仍然可能碰到骨架系统所面临的问题，即系统中推理策略也许不能充分地与新的专家知识的求解方式相匹配。因为有些任务并非一种策略就能解决的，而且有的问题求解策略可能与领域本身有关，必须根据领域状态的变化不断地修改问题求解策略。显然，求解这类问题，用AGE这样的组合开发工具是无能为力的。

John McDermett在一篇文章中[4]提出，当代专家系统的主要弱点是使用一种单一的问题求解方法，而最好的解决办法是创造出在需要时会发明新的方法或修改已有办法的专家系统。但是，解决这一问题的可能途径似乎是产生这样的专家系统，尽管它不会发明新的解决问题的方法，但至少开发人

员意识到这一需要时，它会较容易地调整出新的方法。通过研究，我们发现采用元控制方法将是解决这一问题的有效途径。

二 基本思想

所谓元控制就是将用于推理控制策略方面的知识提取出来，以显式的形式组成一个元知识库，而这个元知识库本身又有自己的推理机，利用这些元知识来对推理策略进行控制就是元控制。开发人员通过加入不同的元知识就可以得到不同的问题求解策略，而当开发人员意识到求解方法不妥时，可方便地修改或增删这些元知识，从而容易地调整出新方法。

本文所述的通用型专家系统开发环境ESDEN(Expert System Development ENviroment)正是从这点出发而研制的。其知识库由多个规则集合、多个事实集合，及一个元规则集合组成。其中多个规则集合和多个事实集合构成了领域级知识库，而元规则集合则构成了元级知识库。ES DEN提供了三种基本的推理框架，每种框架都可以作用于一组或多组规则集合及相应的事实集合，从而形成多个子系统（对应于多个子任务）。元知识的表示采用了产生式规则的形式并分成三类。第一类用于各子系统的调用，完成推理控制策略的宏观控制。第二类用于事实知识的处理，完成各子系统之间的通信。第三类则用于子系统内部规则或子目标的选择，完成推理控制策略的微观控制。通过推理控制策略的宏观、微观控制及各子系统之间的通信，不难得到一个适合于某领域任务的有效的专家系统。

当然，作为开发环境，ESDEN还提供了很多辅助设施，但不属于本文的论述范围，在此不予讨论。

三 知识表示

如何表示知识库中的知识将直接关系到一个专家系统的成败。而对于专家系统开发环境，知识表达的方式很大程度上决定了该开发环境的能力和通用性。一个理想的知识表达形式应该：

- 真实地表达专家的概念和意图；
- 能够被程序正确而有效地解释(interpret)；
- 为方便人类观察者理解和评价，支持推理路径的解释(explanation)；
- 有利于查找知识库中的错误和遗漏；
- 使领域知识从解释器(interpreter)中分离出来，以便不必重新设计解释器就能扩大和修改知识库。

这些准则给系统设计者提出了相互矛盾的要求。前两条准则要求表达形式复杂，而后三条则要求表达形式单一，易于解释且一致。对于专家系统中广泛使用的产生式规则，由于其形式单一和结构上一致，所以它非常符合后三条准则。但由于没有考虑规则作用的上下文环境，因而导致低效性，由于其单一的形式，而造成表达上的非精确性，所以又使得它不能很好地满足前两条准则。作为人类专家，他们在解决问题时，通常采用产生式规则那样的关系，快速地解决日常问题。但是，当需要时，又能够根据第一条准则改用更合适的理论。从知识分级的观点来分析一下专家解决问题时所使用的知识，不难看出，尽管领域知识中也不可避免地存在一些更复杂的知识表示形式，但大量常用的形式还是产生式规则。另外，还有一种如何有效地运用这些领域知识去解决问题的知识，这就是元知识。人类专家之所以能有效地解决专门领域中的许多问题，正是因为他们具有这样多层次、多形式的知识。因此，均衡上述的五条准则，在我们研制的专家系统开发环境中，提供了多层次的知识表示方式。目前我们初步实现了采用产生式规则来表示领域级和元级知识，而其它形式待以后开发。由于系统结构上的模块化，今后的扩充是相当容易的。况且，就现已实现的系统本身而言，已经是相当强有力了。下面将进一步分别讨论ESDEN中的领域知识和元知识的表达方式。

1. 领域级知识表示 领域知识由事实（或称为数据）和规则两部分组成。关于事

实的表示，我们用巴科斯范式描述如下：

〈事实〉::=fact(〈类型〉，〈陈述语句〉，〈不确定度〉)

〈类型〉::=0|1|2

〈陈述语句〉::=〈字符串〉

〈不确定度〉::=[0，1]之间的实数

这里，当〈类型〉取0时表示该事实是由系统事先提供的；取1时表示该事实是在系统运行时由用户提供或由数据采集器提供的；取2时表示该事实由系统的推理机推导出来。〈陈述语句〉是一个字符串，完全可以用英语形式的句子来表达一件事实。〈不确定度〉是多值逻辑中的真值，当取值为0时，表示这件事实为假；取值为1时，表示这件事实完全真；取值在0和1之间，则表示这件事实部分为真。事实存储在一个所谓的综合数据库中，用来匹配规则中的条件部分，决定规则是否可用。

规则由条件和结论组成。当规则的条件被综合数据库中的事实满足时，规则可用；规则的应用又改变了综合数据库。下面用巴科斯范式来描述ESOEN中所采用的规则：

〈规则〉::=RULE(〈编号〉，〈前提条件〉，〈结论〉，〈不确定度〉)

〈编号〉::=1|2|3|……

〈前提条件〉::=[{YES(〈陈述语句〉)|NO(〈陈述语句〉)}]

〈结论〉::=〈陈述语句〉

〈不确定度〉::=[0，1]范围内的实数

〈陈述语句〉::=〈字符串〉

这里〈编号〉用来标识规则。它可以是任意实数。不同的规则具有不同的编号。〈前提条件〉由多条〈陈述语句〉的肯定或否定所组成。它们之间的关系是一种“与”的关系（与二值逻辑中的“与”关系不同，这里指的是多值逻辑中的“与”关系）。YES表示对〈陈述语句〉的肯定；NO表示对〈陈述语句〉的否定，即逻辑上的求反。〈不确定度〉表示这条规则作为多值逻辑中的蕴含式的真值。

知识表示与推理控制是不可分割的两个部分，知识表示必须考虑推理控制的方便和有效。在ESDEN中，我们提供了数据驱动、目标驱动和混合驱动三种基本的推理方式，而上面所述的规则形式都可以用这三种方式推理（如果〈结论〉部分为动作，则不能进行目标驱动的推理方式）。

数据驱动推理时，当条件部分的诸条件都同时被触发时，可导致结论部分的结论成立；并将结论部分的〈陈述语句〉和通过计算得到的〈不确定度〉作为新的事实以下述形式加入综合数据库：

fact(2，〈陈述语句〉，〈不确定度〉)

目标驱动推理时，将规则的结论部分作为目标，条件部分的诸条件作为子目标。要使目标成立，先证明子目标成立。目标成立后也以同上的形式把目标〈结论〉作为新的事实加入综合数据库中。

混合驱动则是以上两个过程的综合。

2. 元知识表示 元知识就是关于知识的知识。在专家系统中，它体现在知识获取、推理控制策略及推理解释等。本文只讨论用于推理控制策略的元知识。

在ESDEN中，我们把元知识和领域知识分开，形成两级知识库。又把元知识分为三类：

- 请求新的推理和过程；
- 对事实知识进行操作；
- 控制推理策略。

这些元知识都采用规则的形式，称为元规则。请求新的推理或过程是出于这样的考虑：为了解决一个大的任务，可能存在太多的规则，使得机器内存不够用或降低系统的效率。经过研究，我们发现，一个大的任务往往由几个子任务组成，完成各个子任务都有各自的一组规则及相应合适的推理方式。因此我们把领域规则进行了分组。规则分组是由系统开发者事先完成的。对于不能用产生式规则解决的问题，提供了过程调用。下面是这种元规则的巴科斯范式描述：

〈元规则〉::=METARULE(〈类型号〉，〈编号〉，〈前提〉，〈结果〉)

〈类型号〉::=1|2|3

〈编号〉::=1|2|3|……

〈前提〉::=〔{〈条件〉}*〕

〈条件〉::= EQUAL (〈陈述语句〉, 〈不确定度〉) |

GREATER (〈陈述语句〉, 〈不确定度〉) |

LESS (〈陈述语句〉, 〈不确定度〉) |

NO (〈条件〉)

〈结果〉::=DATADRIVEN(〈规则组名〉,〈数据库名〉, 〈处理不确定度方式〉) |

GOALDRIVEN (〈规则组名〉, 〈数据库名〉, 〈目标表〉, 〈标志〉, 〈处理不确定度方式〉) |

MIXDRIVEN (〈规则组名〉,〈数据库名〉, 〈标志〉, 〈处理不确定度方式〉) |

REQURE (〈过程名〉) |

STOP

〈陈述语句〉::=〈字符串〉

〈不确定度〉::=〔0, 1〕范围内的实数

〈规则组名〉::=〈字符串〉

〈数据库名〉::=〈字符串〉

〈标志〉::=0|1|2|3

〈过程名〉::=〈字符串〉

〈目标〉::=〈字符串〉

〈目标表〉::=〔{〈目标〉}*〕

〈处理不确定度方式〉::=C (〈证据组合方式〉, 〈不确定度传播方式〉, 〈结论组合方式〉)

〈证据组合方式〉::=1|2|3|……

〈不确定度传播方式〉::=1|2|3|……

〈结论组合方式〉::=1|2|3|……

〈类型号〉取1时，表示这条规则是请求新的推理或过程。元规则的意义是：当〈前提〉满足时，进行〈结果〉所指示的动作。条件中的EQUAL表示当综合数据库中存在事实〈陈述语句〉，且其真值（多值逻辑的真值）等于〈不确定度〉时，该条件满足。而GREATER和LESS分别表示"大于"和"小于"；NO则表示"非"。所有的条件都满足时，前提才满足。〈结果〉所描述的动作分别表示：

DATADRIVEN：用〈规则组名〉所指的规则集合，对〈数据库名〉所指的数据库进行数据驱动推理。

GOALDRIVEN：用〈规则组名〉所指的规则集合，对〈数据库名〉所指的数据库进行目标驱动推理。目标为〈目标表〉中的各个〈陈述语句〉。

MIXDRIVEN：用〈规则组名〉所指的规则集合，对〈数据库名〉所指的数据库进行混合驱动推理。

REQURE：调用〈过程名〉所指的过程。

STOP ：推理机停止工作，返回上级调用。

在目标驱动和混合驱动推理中，都有一个〈标志〉。其意义为：

〈标志〉=0，系统既不提问也不解释。

=1：系统不提问，只在达到或未达到一个目标时解释推理。

=2：系统提问，不给用户解释。

=3：系统提问，同时也给用户解释。

这里，系统提问是指：当推理过程中某子目标没有规则和事实与之匹配时，则提问用户，要求给出该子目标的真值；而解释是指：对用户的提问（Why和How）进行解释（explain）。在每种推理方式中，都要求规定处理不确定度方式。它使系统开发者具有很大的自由选择余地。

对事实知识进行操作的规则也应该作为元规则。它是在以上请求新的推理和过程的元规则所驱动的推理或过程前后，对综合数据库中的事实知识进行保留或删除的操作。它的作用是完成各种推理或过程之间的"通信"联系，如上次推理的结果可以作为下次推理的初始数据等。这类元规则被称为第二类元规则，它们的形式与第一类完全相同，只是将〈结果〉部分改为：

〈结果〉::= DELETE (〈操作时机〉,〈事实表〉) |

KEEP (〈操作时机〉,〈事实表〉) |

〈操作时机〉::=AFTER (〈元规则编号〉) |BEFORE (〈元规则编号〉)

〈事实表〉::=〔{〈陈述语句〉}*〕

〈元规则编号〉::=1|2|3|……

〈陈述语句〉::=〈字符串〉

其语义为：在〈元规则编号〉所指的第一类元规则执行之前(BEFORE)或之后(AFTER)删除(DELETE)或保留(KEEP)〈事实表〉中所有事实。

第三类元规则是控制推理策略的元规则。这些元规则只限于目标驱动和混合驱动的推理控制中，用来控制规则和目标的选择。这类元规则的形式同上，只是将〈结果〉部分改为：

〈结果〉::=FIRST(〈编号〉，〈子目标表〉，〈子目标〉)|

LAST(〈编号〉，〈子目标表〉，〈子目标〉)|

CHOOSE(〈编号〉，〈领域规则编号表〉，〈领域规则编号表〉)|

NOCHOOSE(〈编号〉，〈领域规则编号表〉，〈领域规则编号表〉)|

〈子目标表〉::=〔{〈子目标〉}*〕

〈子目标〉::=YES(〈陈述语句〉)|

NO(〈陈述语句〉)

〈编号〉::=1|2|3|……

〈领域规则编号表〉::=〔{〈领域规则编号〉}*〕

〈领域规则编号〉::=1|2|3|……

其语义为：

(1)：在〈编号〉所指某第一类元规则所驱动的推理中，当〈前提〉满足时，在〈子目标表〉中我们首先(FERST)或最后(LAST)满足〈子目标〉。注意，〈子目标〉必须是〈子目标表〉中的一个元素。

(2)：在〈编号〉所指某第一类元规则所驱动的推理中，为了满足某个子目标可能有多个领域规则可用，即第一个〈领域规则编号表〉所指的规则。当〈前提〉条件成立时，为满足该子目标只从第一个〈领域规则编号表〉中选择(CHOOSE)或不选择(NOCHOOSE)第二个〈领域规则编号表〉中的规则。要求第二个〈领域规则编号表〉是第一个的子集。

四 知识表示的Prolog实现

上面用巴科斯范式描述了知识的表示形式，如何用Prolog语言来实现这种表示呢？从形式上看，规则主要由条件和结论两部分组成，这正好与Prolog子句的体(body)和头部(head)相对应，而且Prolog自身的解释器(interpreter)可作为推理机，使规则表示成Prolog的子句是理所当然的。但是，Prolog的解释器有以下不可克服的缺点，使我们不得不放弃这种思想。这些缺点为：

·不能处理不确定性；

·只有目标驱动一种推理方式；

·无法对推理过程干预，其推理效率很可能极低。

所以，我们有必要用Prolog重新编写规则的解释器。规则和事实(数据)可以用数据库谓词来表达。下面是用Turbo Prolog语言对事实、规则和元规则的描述：

```
DOMAINS
  NUMBER=INTEGER/*(元)规则编号*/
  NUMLIST=NUMBER*/*规则编号表*/
  DECLARATION=SYMBOL/*陈述语句*/
  GOALLIST=DECLARATION*/*目标表*/
  ANTE=ATOM*      /*领域规则条件*/
  CONC=DECLARATION /*领域规则结论*/
  COND=EQUAL(DECLARATION, TV);
       GREATER(DECLARATION, TV);
       LESS(DECLARATION, TV);
       NO(COND)     /*元规则的前提条件*/
  PREM=COND*
  CONS=DATADRIVEN(SYMBOL, SYMBOL,
    C);
       GOALDRIVEN(SYMBOL, SYMBOL,
         GOALLIST, S, C);
       MIXDRIVEN(SYMBOL, SYMBOL, C);
       REQURE(SYMBOL);
       STOP;
       DELETE(TIMES, FACTLIST);
       KEEP(TIMES, FACTLIST);
       CHOOSE(NUMBER, NUMLIST, NUM-
                LIST);
       NOCHOOSE(NUMBER, NUMLIST,
                  NUMLIST)*;
       FIRST(NUMBER, ANTE, ATOM);
       LAST(NUMBER, ANTE, ATOM);
            /*元规则结论*/
  TIMES=AFTER(NUMBER); /*时机*/
    BEFORE(NUMBER)
  FACTLIST=DECLARATION*  /*事实表*/
  C=C(INTEGER, INTEGER, INTEGER)
      /*处理不确定度方式*/
  T=INTEGER/*元规则类型*/
  V=INTEGER/*事实类型*/
  S=INTEGER/*标志*/
```

```
  TV=REAL   /*真值*/
DATABASE
  FACT(V, DECLARATION, TV)/*事实*/
  RULE(NUMBER, ANTE, CONC, TV)
      /*领域规则*/
  METARULE(T, NUMBER, PREM, CONS)
      /*元规则*/
```

将知识表示定义成Turbo Prolog中的数据库谓词，我们就可以把知识存储在磁盘文件中，有效地实现了知识库与推理机等其它部分的分离，而需要时，可以用内部谓词consult把知识库调入Turbo Prolog的数据库中，推理机、编辑器等就可以对知识库进行作用。

五 讨 论

在我们开发的专家系统开发环境ESDEN中，实现了上述的多级知识库及相应的推理机制。使得ESDEN系统在一定程度上既克服了通用知识表示语言难于使用的缺点，又弥补了组合开发工具推理策略有时不能充分地与新的专家知识求解方式相匹配的不足，从而使其应用领域相当广泛。它是研制新一代专家系统开发工具的一次有益的尝试。

当然，本文所述的多级知识库只考虑了关于推理控制策略方面的元知识；而人类认知活动中起核心作用的元知识远不只这一点。另外，ESDEN中的多级知识库实际上只有两级，而知识的层次划分还可以是三级、四级等，所以，本文所述的多级知识库在深度和广度上还可以进一步发展。

参考文献

[1] Davis, R., "Meta-rule: Reasoning about control", Artificial Intelligence, Vol. 15, 1980.

[2] Davis, R. and D. Lenat, Knowledge-based Systems in Artificial Intelligence, Mc Graw-Hill, 1982.

[3] Dinebas M., "Metacontrol of logic programming in METALOG", Proc. of FGCS/84, pp361—370

[4] John McDermott, "专家系统的下一步", 计算机科学, 1987.

[5] 施鸿宝，陈健，"专家系统开发工具综述"，计算机科学，1987.

[6] 曾纪文，等，"元知识及其在专家系统中的应用"，计算机科学，1987.

书讯：《机器学习及其应用》出版

由吉林大学徐立本、姜云飞主编的《机器学习及其应用》一书由吉林大学出版社出版。

该书是根据编者在吉林大学计算机系讲授机器学习的教材改编的。适于做大学生、研究生选修课教材（可讲40学时）。也可供人工智能、专家系统、机器学习、认知心理学、思维科学的学人参考。

该书主要内容有导言，第一章，机器学习；第二章，通过采纳建议学习；第三章，通过例子学习。并附有六个附录，BACONI，BACON3，BACON4，BACON5，学习系统的模式和meta-DENDRAL。还有54篇文献目录。

这本书尤其适于计算机系及短训班讲机器学习专家系统课做教材。对于专家系统研究者了解知识获取的机器学习方法也是一本详细的参考书。

该书由北京王府井书店、北京中关村科海公司、北京黄庄科海培训中心代销。由长春市吉林大学劳动服务公司书店办理邮购。

需要该书的读者，可直接与上述单位联系，定价2.50元，邮费每册加10%。