

TP18

3. 智能系统的技能提高问题

吴信东 (合肥工业大学计算机与信息系)

摘要

The skill refinement is an important aspect of human learning. Some basic factors in human skill refinement is first analyzed with examples and then the basic problems in building intelligent systems that implement skill refinement are studied in this paper.

一. 引言

学习是人类智能的一个最基本也是最为重要的方面,所以以模拟人类智能的基本机理、开发更为“聪明”的计算机系统为目标的人工智能研究从一开始(1956年)就一直注重机器学习能力的开发,三十多年来,机器学习研究虽然经历了漫长而曲折的发展过程,但确实取得了许多令人瞩目的成就。如各种学习策略的开发,一些著名学习系统的研制和若干个实际专家系统中自动知识获取系统^[1]的应用。八十年代以来,机器学习的发展较为迅速,它已被视为解决专家系统中知识获取这个“瓶颈”问题的关键^[2]。

学习是一个外延较为抽象的概念,广义地说,一切从无到有、不断改进的人类或机器性能的实现均可理解为学习。专家系统研究中,我们常把机器学习狭义地定义为系统中“显式知识(知识库级知识)的获取(the acquisition of explicit knowledge)^[3]”。事实上,现行机器学习的研究,不光在专家系统领域,基本上都集中在显式知识的自动获取方面。如较为著名的学习系统AM、AQ11、Winston的结构知识学习系统、Eurisko,还有已投入实际应用的Meta-DENDRAL、TEIRESIAS等。显然,获取显式知识是增加或改进性能的一个重要方面。但是,人类问题求解的能力提高除了显式知识的获取之外,还有另外一个极其重要的方面,我们常称之为技能获取(skill acquisition)或技能精炼(skill refinement),这种技能获取是在大量的实践中实现的。以计算机程序编制为例,关于程序设计语言和数据结构的基本知识可以理解为显式知识,它们可以在课堂上传授给学生,但学生要想成为一名程序编制专家的话,光有显式知识还不够,他还必

须通过大量实际程序的编制实践才能提高他的编程技巧。这种技巧就是我们常称的启发式知识(heuristic)的一种。在已有和现行专家系统研究中,它们常常作为知识库级知识(显式知识)由专家提供。但是,启发式知识是否一定可以表示为显式知识呢?如何建立一个能模拟人类获取技能的问题求解系统(而不单纯是学习系统)?这样的问题在以前的研究中很少专门讨论过,其原因可能有二:一是我们对技能提高的过程还理解得甚少^[3],二是真正投入应用的学习系统还为数有限,大多数自动知识获取系统^[1]中知识的自动获取和问题求解还没有完全融合起来。

二、人类的技能提高

我们以产品设计为例,设计师的任务是根据用户提出的一组要求和特定的条件约束,设计产品的结构形式,如产品中各零部件的形状、参数以及各零部件之间的关系。有时候,设计过程还需要优化成本或其它目标函数。设计工作是一种创造性活动,设计过程中除了理论分析和数据计算外,设计者的经验、技巧往往对产品设计的工时和质量有着极为重要的影响。一项复杂的设计任务中可能遇到的主要问题有^[4]:(1)在设计的各个中间状态,设计者对各种零部件、参数等选择项的选择后果无法立即进行评价,但又必须临时地开发各种设计可能;(2)设计的约束和用户的要求来自许多方面,通常没有一个成熟的模型把它们同设计选择综合起来;(3)如果设计的产品规模较大,由于复杂性的需要,则一般先把整个产品的设计分为若干个子部分,但由于设计型问题的固有特点,这些子部分往往难于相互独立,子部分之间存在有约束和约束传播,从而加重了设计问题的难度;(4)当设计的规模很大

时,影响设计的因素往往很多,这样,一些因素就可能易于被忽视,对某些因素的修改也难以很快估计出对整个设计的影响,这就要求对设计过程中进行设计选择的合理性进行论证;(5)修改设计的时候,要重新考虑设计选择后果的各种可能性,设计和修改过程中应能以系统的观点着眼全局的考虑而不能只重视某些部分的局部优化;(6)许多设计问题还需要有立体关系的推理,如关于距离、形状、轮廓的推理,这样的推理往往靠一些定性推理方法或似然推理方法。所以,对于非专家设计师(尤其是新手)来说,设计过程充满了困难,这也是近些年来设计型专家系统在国内外引起了较为广泛重视的重要原因。

对一个系统掌握了设计理论的专业人员来说,如果他以前没有过实际产品设计或者没有过他将要设计的产品领域的实践,就开始实际设计的前期工作,肯定是耗时且费力的。但随着他设计的产品越来越多,他的经验会越来越丰富,一件产品设计的工时就越来越短,质量也会越来越可靠。这就体现了设计人员的技能提高。虽然这种提高对人类来说可能是在无意识的状态下进行的,但导致技能提高的基本因素我们可以从以下几个方面来分析:

(1)人的记忆能力。设计人员在接受一项设计任务时,如果他以前接受过同样的任务,如同样的要求和同样的约束,则他没有必要再从头设计,而可以把以前的设计版本或者以前设计时用到的知识直接搬过来,甚至不需要作多大的修改,即可很快推出设计结果。这种靠以前经历的记忆来完成的设计显然比运用所有的领域知识、涉及到前面诸多问题的重新设计要直接、高效得多。这是提高技能最简单也是最直接的一条思路。A.L.Samuel曾模拟人类下棋记棋谱的方法研制了下棋程序Checkers,这个程序在技能提高的实现方面一直被认为是所有学习系统中最为成功的一个。

(2)人的类比能力。当接受到的任务与以前设计过的零件产品有许多相似的要求和条件约束时,作为设计师来说,他会优先考虑对以前的结果进行修改以满足新的任务的不同要求或者利用以前的设计步骤作参考,在不同步骤作相应的设计选择修改。这种根据过去的经验来求解新的问题的过程是人类一种重要的思维方式,称为类比思维。

(3)人的归纳能力。两件要求和约束不完全相同的设计,可能后来一件的设计借用了前一件的经验,但也可能没有,当两件产品都推出以后,我们

可能会进一步发现两件产品中的一些相似性。这些相似性可能在设计或类比之前未曾认识到。见到这种相似性,设计师就有这种能力,他能有意识或无意识地把它同两件设计的要求和约束的差别对应起来,发现差别的某些部分并不影响或者并不实质影响产品结果的某些内容,这样他就可以进行归纳,对前一件设计的要求和约束进行强化(strengthening或generalizing)形成假设,强化的目标是把第二件设计中不影响产品结果的那些差别部分包罗进去。但其强化结果可能不仅包罗了它们两者,还包罗了尚未出现的其他差别。其他差别虽然还需要进一步验证,必要的时候还需要弱化(weakening或specializing)处理,但归纳过程一直在继续。这样,当第三种设计任务出现时,即使设计师以前从未设计过同样的产品,但他却可能运用归纳的结果很快推出产品形式。

当然,人的技能提高的因素可能还不止这三个方面,但这三个方面显然是人类专家所具备的能力。这三种能力在人类的头脑中也可能是随机、不固定的,它们在人脑中的应用是一种启发式过程,难以精确描述,因为这里常常含有一些形象思维的成份。例如,专家在可以搜索记忆的地方也可能应用了类比从而加长了设计过程,在处理一件新产品的设计时,他很快地给出设计结果,这个结果可能难以准确地说是记忆、类比还是归纳。显然,这种启发式过程是专家藉以实现技能提高的又一重要方面。对以数理逻辑为基础的计算机来说,要模拟人的技能提高必须形式化上述过程,既要让计算机具备人的记忆、类比和归纳能力,还要协调好它们的应用。

三、实现系统的技能提高

对一个问题求解的系统(如专家系统)来说,用户提出的问题可能有两类。一类是以前处理过的,另一类是从未处理过的。后一类中又有两种可能,一种是系统可以借助已经处理过的例子的求解知识进行类比或归纳求解,另一种是没有任何先前的例子可以借鉴。要建立一个能模拟人类实现上述技能提高的实际问题求解系统,首先必须让系统具备这三种可能情况下的问题求解知识。显然,三种情况下的问题求解知识可能有重叠的成份:没有可借鉴的例子时,需要具备现行设计型专家系统中所具备的知识库级知识,我们称之为第一类知识;有例子可以借鉴时,需要具备从已有例子的知识中推理出没有例子借鉴情况下推理出的同样结果的知识,这

种知识先要从已有例子的知识中生成问题求解知识,我们称之为第二类知识;当以前处理过同样的例子时,借用例子中的结论知识(我们称之为第三类知识)可以更直接地得出问题的求解结果,这种结果严格地说应该与前两种情况下得出的结果保持一致,否则重叠的知识中含有矛盾。第一类知识是系统必备的,如果它对领域的问题求解较为充分,我们可以理解为相对第三类知识而言的“深层知识”,而第三类知识相对地理解为“表层知识”。第一类知识和第三类知识均为问题求解的知识库级知识(显式知识)。第二类知识以及用于实现每次问题求解结论知识存储(记忆)的知识,是用于实现系统的知识获取的元知识。知识获取通过记忆可以获得第三类知识,通过类比和归纳可以获得第一类知识和/或第三类知识。

第一类知识和第三类知识的不断获取显然为系统问题求解的技能提高创造了条件。但我们前面分析,过如合适时,有效地运用它们,而且类比、归纳手段也很重要。所以实现技能提高的系统还需要具有一种对三类知识进行组合调度的元知识,它可能是推理机中专门设计的一个调度程序。这样的调度程

(上接第26页)

参数不超过4个;输出参数不超过4个,不少于一个。这表示了结构化设计方法中模块间参数个数的限制关系。

3.2 JSD上层活动^[5]

JSD (Jackson System Development) 的高层活动如图12所示。

首先由“JSD”frame描述JSD的高层ACTIVITY的静态组成,它们是ENT/ACT, STRUCT, MODEL, INTER, INFO, TIME, IMPL.而对每个活动利用ACTIVITY CLASS对象进行分别描述,从而,描述了JSD的高层活动。JSD详细描述请参见^[9]。

对于其它方法,如Petri网等,SOKM也应该可以描述,需要逐一研究、实验。软件工程方法请参见^[7]。

感谢:在本研究过程中,得到了软件工程实验室同事们的大力支持和陈火旺教授的指导,还得到了日本东京理科大学菅野文友教授的热情指导和帮助,日本SRA岸田孝一先生的有益的讨论,在此,一并表示感谢。

参 考 文 献

序同现有专家系统中的冲突求解策略^[2]有一些相似但有着实质上的不同。关于深层知识(如本文的第一类知识)与表层知识(如本文的第三类知识)的协调运用问题也是新一代专家系统研究的重要课题之一^[3]。

四、结 论

根据以上的分析和初步探讨,我们可以得出以下结论:

(1) 实现技能提高的系统需要具备不同层次的知识。这些知识不仅仅是知识库级的显式知识,也含有获取和调度显式知识的元知识。显式知识之间可能有重叠。

(2) 光靠显式知识的获取还不能保证系统技能的提高,但系统技能的提高又离不开显式知识的获取。

因为本文旨在探讨技能提高,关于显式知识获取的各种学习策略我们未作具体阐述。

鸣谢:本文的有关问题是上海交通大学博士研究生杨刚来合肥调研时提出的,文中内容同杨刚、我校CIMS所章振辉等一起交谈、探讨过,也得到了他们的一些启发。(转第6页)

[1] Osie^[7] Osterweil, L., "Software process Are Software Too." Proc. of the 9th ICSE, ca. USA 1987

[2] 何^[8] 软件工程中的知识和知识表示《计算机世界》No. 23, 1987

[3] SDA コンソーシアム ^[5] 第1回 SDA シンポジウム, テクニカルレポート——ソフトウェア設計環境の新しい 統合化技術 東京1987年11月13日 SDA

[4] 岸俊行, 宮下洋一^[5] つしーム用いオ ブジェクト指向モデルつニよるソフトウェア要求記述実験 STC 60技一58 A 3, 1985

[5] Lloyd G. Williams Software Process Modeling: A Behavioral Approach. SDA コンソーシアム 1988第1四半期活动报告书(I08-21) (1988年1月~1988年3月)

[6] 王^[6] 软件开发环境中一个面向对象的数据模型的设计与实现 武汉大学硕士论文 1988年7月

[7] 何^[8] 软件工程方法 武汉大学出版社 1988年12月

[8] 何、金^[9] SOKM/L的设计 武汉大学“八六三”高技术研究报告No.12 1989. 5

[9] 何、金^[9] JSD的知识表示和利用 武汉大学“八六三”高技术研究报告 No. 13, 1989.6