

基于知识的自动程序设计系统XAPS

唐永明 (西北大学计算机科学系, 西安710069)

摘要

This paper proposes an automatic programming system XAPS which applies the artificial intelligence techniques to software engineering. The architecture and principle of XAPS are introduced systematically. The process from initial problem supplied by user in natural language to an algorithm and program for solving the problem is described.

一、引言

自动程序设计的研究由来已久, 它的发展可以追溯到本世纪五十年代^[1]。当前所进行的自动程序设计活动主要是把人工智能技术应用到软件工程, 其最终目标是要实现软件生产的自动化^[20]。也就是说任何想利用计算机解决问题的端点用户只需给出做什么, 而不必说明如何做, 就能达到使用计算机的目的。

XAPS系统的设计正是以此为出发点。它只要求用户用其所熟悉的领域术语和概念非形式地说明要解决的问题。系统接收这一初始问题后, 利用丰富的领域知识形成正确的形式化描述。最终产生解决问题的程序。

由于自动程序设计一般分为两个活动: 形式化与程序生成^[2], 因此, 本文首先介绍XAPS系统的体系结构, 然后给出形式化与程序生成的基本思想。

二、XAPS的结构

XAPS系统由表示专家、分解专家、算法设计专家、程序设计专家、控制专家和种所需的知识库组成, 如图1所示。

显然, XAPS系统是一个专家系统的集合, 其中形式化活动由表示专家完成, 程序生成则由分解专家、算法设计专家和程序设计专家完成。控制专家负责协调各专家及整

个系统的所有活动。

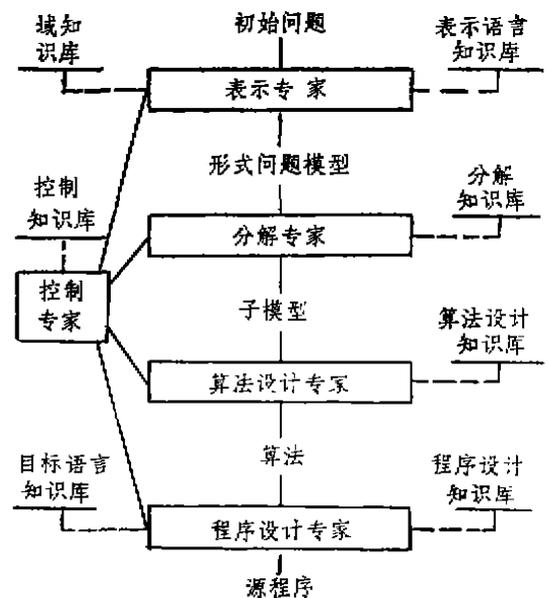


图1 XAPS的体系结构

XAPS系统的输入采用英语描述初始问题, 只要求简单地说明“做什么”。当系统接收了该输入后, 表示专家将利用领域知识和形式化表示语言产生问题的形式化模型。然后, 分解专家利用启发式分解知识, 将问题模型分解成若干子模型并在子模型间建立起合适的接口, 算法设计专家为各个子模型生成解决子问题的算法, 程序设计专家根据每个子算法先产生解决子问题的程序, 再利用

程序综合技术形成一个完整的程序。

三、形式化子系统

XAPS的形式化活动由表示专家完成，其目的是完全实现形式化的自动化，形式化表示过程如图2所示。

用户的输入仅说明“做什么”。接收输入后，利用域知识就可以确定问题的性质，明确为何种问题。接着利用这一问题的启发式知识，分别推导出问题的输入、输出以及从输入得到输出的关系。最后，问题表示系统调用表示语言形成问题的形式化模型。系统的这部分已经在银行的一般储蓄业务管理领域实现。下面列举一个形式化过程的例子。

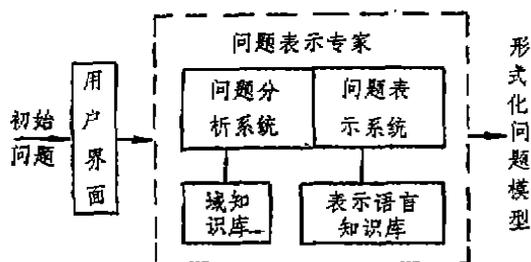


图2 形式化子系统

在银行储蓄业务管理中，常进行的处理有计算利息、存取款、开关户头等。这些工作被看成初始问题。如果以“计算利息”为例，它的用户输入为：

Compute interest of current account.

由于域知识库中包括有储蓄业务的基本知识，当系统接收了这一问题后，很容易确定其为：计算利息。

我们知道，若要计算当前储户的存款利息，下面的启发式知识是必不可少的。

- ① 如果初始问题是“计算利息”，则输入为起始日期、结算日期、利率和当前存款金额；
- ② 如果初始问题是“计算利息”，则输出为原存款额加上利息即新的存款金额；
- ③ 如果初始问题是“计算利息”，则操作为：先计算利息；再将利息加到原存款金额之上。

既然拥有这些知识，就不难确定和表示

问题的输入、输出及操作。整个问题的形式化模型如图3所示：

```

MODULE-NAME compute-interest (c-a-
    data, rate, date1, date2 → new c-a-d-
    ata)
LINK
MODULE-BODY INPUT c-a-data,
    current account data
    rate, numeric
    date1, date
    date2, date
OUTPUT new c-a-data, new
    current account data
STOP
COMPUTE interest of current account
FROM date1 TO date2
WITH-ANNUAL-RATE rate
WHERE(USING c-a-data)
END
MODIFY-FIELD (balance) INTO
    c-a-data
WITH-NEW-VALUE (c-a-data)
    balance+interest)
WHERE(OBTAINING new c-a-data)
END
MODULE-END
    
```

图3 问题模型示例

其中，c-a-data是当前储户数据库包括储户号码、储户姓名、存款金额、存款日期、存款种类等信息。在建立形式模型的过程中，只把c-a-data作为一种抽象的数据类型处理。整个模型由三部分组成，而其中的模型体MODULE-BODY部分又由三部分构成。LINK部分用于模型的分解。该语言是以E-SAP^[1]表示语言为基础建立。

显而易见，XAPS系统在完成形式化活动时，摒弃了与用户交互式对话产生问题的抽象表示的方式，无须用户干预形式模型的产生过程。

四、程序生成子系统

众所周知，软件开发是一项知识密集型活动，实现其自动化是保证质量的根本途径

之一。从目前的研究工作可以看出,软件生产自动化的方法各种各样^[2,4,7,14,17],但其目标是一样的。XAPS系统综合了现有的一些实现方法。

在XAPS系统中,分解专家首先分析表示专家给出的形式化问题模型,然后利用分解知识将一个完整的模型分解成若干个协同表示原问题的子模型。这种分解依赖于三种类型的知识:软件工程知识、特定域的启发式知识和非特定域的启发式知识。所有知识都用产生式表示包含在分解知识库中。

当问题模型被分解成一些子模型后,算法设计专家为各个子模型设计算法即回答“如何做”这一问题。从图3所给的问题模型示例不难看出,当要解决的问题的输入、输出确定之后,解决“如何做”问题的关键就是要设计出从输入得到输出的算法。XAPS系统的算法设计采用基于知识的增量转换方法。这种方法综合与算法设计有关的所有知识,将子模型中的操作部分所包含的抽象概念逐步转化成最具体的概念。例如,将集合转化成数组;将抽象符号转化成具体的变量;将计算性文字说明转化成具体的算术表达式;将抽象数据类型转化成可直接实现的数据类型等。整个转化尽可能利用所拥有的知识使任

何抽象的对象被具体内容取代,直到能够直接用高级程序设计语言表示。

程序设计专家的任务分为两个方面,其一产生每个子算法的程序即子程序;其二将所有子程序综合成一个完整的程序。由于XAPS系统的软件自动生成过程完全遵循了软件工程中的结构化设计思想,因此这里的综合过程,实际上就是将各个子模块按其相互间的接口组合起来。各个子模块间的联系由模型分解中每个子模型中的LINK部分说明。

五、结束语

实现自动程序设计的前景十分诱人。关于这方面的研究工作,也已有十多年的历史^[1]。XAPS系统旨在综合已有的方法,将人工智能技术应用到软件工程。虽以较为简单的应用领域为对象,但为探讨建立通用系统的可能性积累了经验。特别是XAPS的形式化子系统,只要求用户简单地说明要解决的问题。做到了整个过程无须用户干预。

大家知道,自动程序设计的研究涉及的方面很多,包括问题求解、专家系统、知识获取、机器学习等。我们的工作才只是一个开始,仍存在许多问题尚待进一步探讨。(参考文献共21篇略)

下期主要内容预告

认识论应当成为人工智能的主要基础

理由维护系统的研究

基于C++的良好OOP风格法则

ELF类型理论及其应用

情景逻辑与时态逻辑在知识处理中的应用

OBMS/IDKE: 模型及其无缝的面向对象编程语言

计算机学习理论及其应用(1)

解释学习的效用问题