

专家系统的软件工程建设方法*

刘群 黄厚宽

于之硕

(哈尔滨船舶工程学院, 哈尔滨 150001)

(哈尔滨电工学院, 哈尔滨 150040)

摘要 本文指出 ES 所固有的一些困难引起了 ES 的设计和维持上的困难, 提出了 ES 的软件工程方法的观点、描述认知过程的知识说明和建造 ES 过程的 PLWF 模型, 用来解决 ES 的设计阶段和维持阶段产生的问题。

关键词 专家系统、软件工程。

专家系统(ES)是一个软件系统:它拥有专家规则,以运算符号方式推理,能够进行较复杂的难题求解^[1]。ES 技术已渗透到许多领域^[2],能处理越来越复杂的问题领域,使 ES 的开发更加复杂化。现有的 ES 的建造方法如同早年的程序设计,对 ES 的开发、维持变得较为困难^[3-5]。应该探讨建造 ES 的系统方法,研究 ES 的软件工程科学以一个新的方法论作指导,进行 ES 的开发活动。

ES 开发的许多问题是由其自身特性造成的,它与一般的软件系统的开发相比具有以下特点:有效地处理领域中动态的知识,并实现知识的利用、用软件模仿专家行为,体现出正有效的推理能力和较强的解释能力,在开发过程中,逻辑主体及他们之间的相关性也比一般开发过程复杂得多。在 ES 的开发和维持活动中,有些从 ES 的问题特性继承下来的本质的(essence)问题和一些在开发过程中引起的事故(accident)问题,例如因缺少必要的文件说明造成现有 ES 维持上的困难^[6]。一些传统的 ES 开发技术,诸如设计—完成—评价(design—implement—evaluate)即不断地测试评价系统的原型使之改进、完善,和逐步求精的选择过程:变量式—迭代等,都仅适合开发小型 ES^[7],这种技术并没有提供进一步开发、完善、维持这个 ES 原型(pototype)的方法^[8]。为有效地解决确认 ES 的问题特性、描述所需知识、建立开发中逻辑主体适当的关系,为维持 ES 提供必要的信息等难题,提出针对 ES 的软件工程方法。

1 系统分析中逻辑主体之间的关系模型

ES 的开发活动可用图 1 所示的进程模型来表示。

将确认和概念化称作系统分析阶段,形式化、实现、试测与评价称为系统设计阶段,另外

* 本文 1991-12-14 收到,1992-04-21 定稿

作者刘群,37岁,副教授,主要研究领域为人工智能,知识表示。黄厚宽,54岁,教授,主要研究领域为人工智能,专家系统,现在北方交通大学工作。于之硕,53岁,副教授,主要研究领域为软件工程,专家系统。

本文通讯联系人:刘群,哈尔滨 150001,哈尔滨船舶工程学院

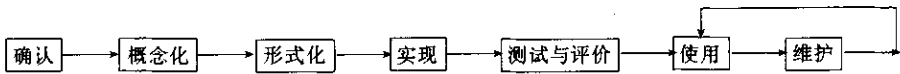


图1 ES开发的进程模型

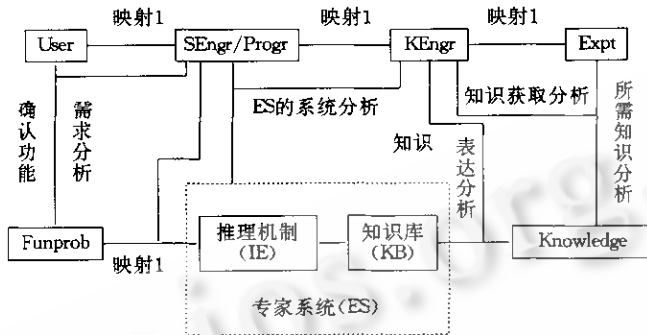


图2 系统分析阶段逻辑主体之间关系

两个为使用和维护阶段. 分析阶段是由四个逻辑主体协调一致来完成的. 即用户(User)、软件工程师/程序员(Sengr/Progr)、知识工程师(KEngr)和人类专家(Expt)完成ES的目标、功能上的确认和需求等方面的分析. 由用户提出ES的功能要求和求解难题的范围, 记作Funprob. 由知识工程师和人类专家分析满足、实现Funprob所需要的知识, 记作Knowledge. 软件工程师与知识工程师完成ES的系统分析. 在这个阶段主要确定应用域的问题特性, 对其计算模型的分析 and 说明. 逻辑主体之间的关系如图2示. 依靠四个被处理的实体(Funprob, IE, KB, Knowledge)将四个逻辑主体联系在一起, 其中的连接关系是:

(1) 映射1: 负责四个逻辑主体之间的信息传递. 记作,

$$\text{Map1}(X, Y), (X \neq Y, X, Y \in \{\text{User}, \text{SEngr/Progr}, \text{KEngr}, \text{Expt}\}).$$

Map1 是对称的、可传递的.

(2) 功能确认需求分析: 明确ES拥有的功能, 求解难题的特性范围及解决的程度, 建立ES的功能和任务特性集合Funprob. 即:

$$\text{Map1}(\text{User}, \text{SEngr}) \text{++} \text{Funprob} \text{ (“建立”记作++)}$$

$$\text{Funprob} = \{\text{Fun} \mid \text{Fun 是 ES 的功能}\} \cup \{\text{Prob} \mid \text{Prob 是 ES 能够求解的问题}\}$$

$$\text{SEngr} \text{++} \text{WFP}, \text{WFP} = (\text{Funprob}, \text{Rfp})$$

其中, $\text{Rfp} = \{\text{Rf}\} \cup \{\text{Rp}\}$, $\text{Rf} = \{\langle \text{fun}_i, \text{fun}_j \rangle \mid \text{fun}_i, \text{fun}_j \in \{\text{Fun}\}, \text{fun}_i \text{ 与 } \text{fun}_j \text{ 在软件功能上是相关联的}\}$, $\text{Rp} = \{\langle \text{prob}_i, \text{prob}_j \rangle \mid \text{prob}_i, \text{prob}_j \in \{\text{prob}\} \text{ 且有着分类上的关系}\}$. 因此说WFP是问题的计算模型.

(3) 所需知识分析: 根据{prob}确定所需要的知识, 这项工作由KEngr和Expt共同完成. 由于Map1是可传递的, 有 $\text{Map1}(\text{User}, \text{KEngr})$,

$$\text{Map1}(\text{KEngr}, \text{Expt}) \text{++} \text{Knowledge}$$

$$\text{Knowledge} = \text{text know} \cup \text{exp know}$$

其中text know是文本知识集合, exp know是经验知识集合. $\text{text know} = (\{\text{element}\})$,

Rtk). element 是文本知识的元素(最基本的处理对象). $Rtk = \{ \langle element_i, element_j \rangle \mid element_i, element_j \text{ 之间存在内在联系} \}$. KEngr++text know, 由知识工程师分析并描述出文本知识的结构模型.

(4) 知识获取分析: 按照所需知识要从专家那里获取经验知识. 其目的在于: 认真考虑获取的方法、可能性及能够达到预期结果等方面的论证, 保证求解 {prob} 的知识是完备的.

(5) 知识表示分析: 根据 text know 和 exp know, 分析出它们的特性、结构性、使用方式, 分析所采用的将知识转换成为表达的方法、技术及其可行性论证.

$$WE = (\{element\}, \{attribute\}, Re, exp\ know, Rex)$$

其中 attribute 是描述 element 的属性, Re 是 element 与 attribute 之间的确定关系:

$$Re = \{ \langle element_i, attribute_1, \dots, attribute_m \rangle \mid element_i \leftarrow attribute_1 \wedge \dots \wedge attribute_m \}$$

$$Rex = \{ \langle element_i, hypo_j \rangle \text{ or } \langle attribute_{i,k}, hypo_j \rangle \mid element_i / attribute_{i,k} \text{ 与 } hypo_j \text{ 对应} \}$$

其中 $hypo \in exp\ know$, 作为 exp know 的基本处理对象, 例如假设、求解问题的方法等. 可见 Rex 是连接 text know 和 exp know 的关系.

将知识的表达记作 WEP, 知识表示分析就是讨论怎样构造一个同态 $\Phi_p: WE \rightarrow WEP$ 和对 Φ_p 的要求.

(6) ES 的系统分析: 根据 {Fun} 和 WEP 来分析、确定建造 KB 与设计 IE 的步骤及设计要求. 要结合 Expt 思考问题的方式和 WEP 的特性提出 IE 的推理方式、推理解释方法等 IE 设计上的需求及与 IE 相适应的 KB 组织方法, 和实现上的技术.

(7) 映射 2: 在 SEngr 作用下两个被处理实体 Funprob 和 ES 的对比分析, 记作 Map2 (Funprob, ES). 即检查 ES 各组成部件的设计要求能否满足 {Fun} 和求解 {prob} 的需求.

该阶段的每项工作必须形成规范的分析说明和设计要求说明. 在设计阶段, 知识获取等工作则严格按照这些说明来完成^[9,10]. 处理好各项分析之间的介面就要依靠 Map1 来传递、交换各个方面的信息. 从方法论上讲: 对于知识获取、表示的分析, 以现有的 AI 技术为基础以 text know 为“背景”, 从 Expt 诱导 (eliciting) 出 exp know 来逐渐扩充 text know.

2 知识说明 (Knowledge Specification)

知识说明与求解问题所需要的知识的理解相关联. 它是一个认知模型的说明并非计算模型的说明. KS 是一个新的研究问题, 目前还没有现成的说明技术 (Specification technique) 用来完成这项工作.

2.1 文本知识说明

文本知识是易用自然语言描述的并具有良好的结构性. 它有:

- (1) 实体——基本元素, 例如概念, 病名等.
- (2) 属性——对实体的描述, 即实体存在的表现.
- (3) 类——具有某些相同属性的实体集合.
- (4) 关系——实体和类的相关联系.

4 个主要构成部件. 除对它们以说明书形式描述之外, 还必须完成它们的软件说明 (Software Specification). 即对知识表示的要求和在知识的表达上所进行的运算要求的说明. 有

下面内容：

- (1) 实体和属性及其使用的形式定义.
- (2) 类、关系和实体之间相关性的形式描述.
- (3) 中间结果状态的存取和使用方法.
- (4) 所要进行的模式匹配运算及运算类型.
- (5) 需要表示与问题相关的输入数据, 数据形式变换的格式, 问题描述格式.
- (6) 表示系统的输出的格式, 即推理结果或问题答案的形式.

2.2 经验知识说明

经验知识往往是无结构的. 它是由专家所使用的判断方法、问题求解的技巧、思考问题的方式等方面构成. 为了方便知识获取并保证所获取的知识是完备的, 经验知识说明显得更加重要. 它有下列几个方面的内容:

- (1) 在问题求解中使用的“假设——推断”(hypothetic——deducting)策略, 将假设公式化表示, 即如何提出假设链.
- (2) 表示出根据问题的输入和公式化表示的假设匹配、识别知识模式的方法.
- (3) 对某一个假设的提出或者所作出一个判断可能影响其它假设的提出, 即说明假设之间的相关性.
- (4) 提出假设的时间和情况方面的条件, 在某种情况下提出几个可能的假设并说明它们之间能否交换使用.

要进行软件说明是:

- (1) 建立一个能够匹配输入数据不同模式的模式匹配器.
- (2) 需要一个部件来完成选择、判断下一个要考虑的假设, 即用软件实现组织假设链.
- (3) 一个假设(或判断)适用范围和条件的形式化说明.
- (4) 对同时提出的几个可能假设进行可信度估价, 能判别哪个较其他的更为适用.

知识说明是建造 ES 中一个重要的环节, 如果忽视了它的作用会造成 ES 维护上的困难.

3 设计过程的模型

由于很难抽象出 ES 的计算模型, 只有在设计过程中不断调整其初始模型. 将瀑布(waterfall)设计过程模型与系统分析阶段的分析方法结合起来得到伪线性瀑布(pseudo-linear waterfall)设计过程模型, 记作 PLWF(如图 3).

ES 中的知识是变化的, 那么 ES 的设计过程就是围绕知识变化来讨论的. 图 3 中 Φ_1 — Φ_4 就是如下的设计工作:

Φ_1 : 由文本知识作背景诱导出专家的一些经验知识, 同样可将某些经验知识进化成文本知识.

Φ_2 : 根据经验知识说明等, 进行 exp know 的形式化及其与 KB 有关方面的设计.

Φ_3 : 根据 text know 在 IE 中作用, 进行 IE 使用 text know 的推理部分设计.

Φ_4 : 根据文本知识说明等, 进行 text know 的形式化及其与 KB 有关方面的设计.

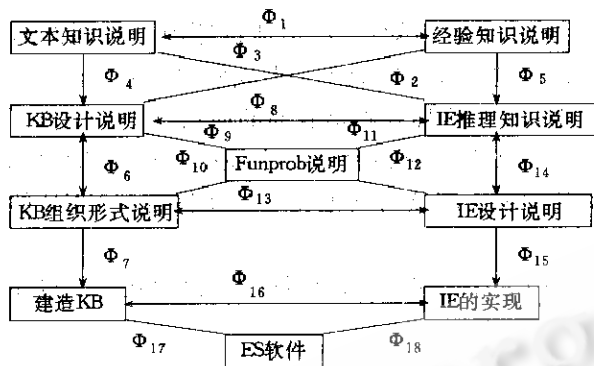


图3 PLWF设计过程模型

Φ_1 :根据 exp know 在 IE 中作用,进行 IE 使用 exp know 的推理部分设计.

Φ_2 :进行 KB 组织形式设计.

Φ_3 :根据 KB 的设计及组织形式建造 KB.

Φ_4 :验证 KB 的设计与 IE 的推理设计相适应性.

Φ_5 :检验 KB 中知识对 ES 能够求解的问题来讲是完备的.

Φ_6 :KB 的组织形式应该为实现 ES 的功能提供便利.

Φ_7 :IE 的推理设计满足 ES 的功能要求.

Φ_8 :验证 IE 的工作方式符合问题求解的思维形式.

Φ_9 :检验 KB 的组织形式适合 IE 的推理方式.

Φ_{10} :根据 IE 的推理设计完成全部的 IE 设计.

Φ_{11} :根据 IE 的设计说明实现 IE.

Φ_{12} :进行 KB 与 IE 的调试工作.

Φ_{13}, Φ_{14} :整个软件系统的调试工作.

按 PLWF 模型建造 ES,四个设计逻辑主体(Expt, KEngr, SEngr, Progr)有了分工与合作,各自的工作集合为:

Expt:E. WS = $\{\Phi_1, \Phi_2, \Phi_3\}$

KEngr:KE. WS = $\{\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6, \Phi_8, \Phi_9, \Phi_{11}\}$

SEngr:SE. WS = $\{\Phi_6, \Phi_7, \Phi_{10}, \Phi_{12}, \Phi_{13}, \Phi_{14}, \Phi_{15}, \Phi_{16}\}$

Progr:Pr. WS = $\{\Phi_7, \Phi_{14}, \Phi_{15}, \Phi_{16}, \Phi_{17}, \Phi_{18}\}$

由工作集合决定了他们的合作:

$Cop(X, Y) = X. WS \cap Y. WS, X \neq Y, X, Y \in \{Expt, KEngr, SEngr, Progr\}$. 这种合作也体现出他们之间的信息传递关系,即 $Cop(X, Y) \rightarrow Map1(X, Y)$. 在设计过程中,根据各方面的信息,可以反复地调整、改进设计说明使 ES 的设计达到最佳状态.

4 维护问题

ES 软件有 3 种维护活动:(1)校正性维护;(2)完善性维护;(3)适应性维护. 在第(2)、

(3)种维护中主要是因知识变化而引起的系统维护. 知识变化的原因有:一是更进一步地了解了系统要求.二是应用域中有了新知识.严格地讲原因一是偶然的(事故的),在建立了Funprob和采用PLWF设计过程极大地减少了形成原因一的可能.解决由原因二引起的维护却比较困难^[5].往ES的KB中增加规则是容易的而增加知识就难了,有必要建立知识的调制(modularity)规范^[12-14].知识的调制性有3个原则:(1)新知识应该与原来的知识相耦合(coupling).(2)KB中某类表达应该与其他类表达相耦合.(3)KB的组织形式与IE的推理方式相耦合.

由于PLWF模型强调了知识说明作用,根据知识说明就能够遵循调制原则修改、增加知识,使维护工作变得容易一些.

5 结 论

当在建造ES时采用ES的软件工程方法,尤其是认真进行知识说明会使ES的维护变得容易.目前对认知过程还没有形式化描述方法,借助于说明技术来描述它可能会好些.PLWF模型表示了建造ES的活动过程,由被设计实体将逻辑主体联系在一起.将他们密切合作反复磋商使许多问题在设计中得到解决.因此说,这种ES的软件工程方法能较好地解决在ES设计上和维护上出现的问题.

参 考 文 献

- 1 Hayes-Roth F *et al.* Building expert systems. Mass.; Addison-Wesley, 1983;49.
- 2 McDermott J. R1: An expert in the computer system domain. AAAI, 1980(1);269-271.
- 3 Held J P *et al.* Conceptual data modeling of an expert system. IEEE Expert, 1987,2(2);42-51.
- 4 McDermott J. 专家系统的下一步. 计算机科学,1987(2);1-7.
- 5 A van de Brug *et al.* The taming R1. IEEE Expert, 1986,1(3);33-39.
- 6 Brooks F P. No silver bullet; essence and accidents of software engineering. IEEE Computer Magazine, 1987,20(4);10-19.
- 7 Parnas D. Research problems in programming methodology. In: Wagner P *et al.* ed. Research Directions in Software Technology, Mass.; MIT Press., 1979;352-364.
- 8 Waterman D A. How do expert systems differ from conventional programs? Expert System, 1986,3(1);16-19.
- 9 Chandrasekaran B. Generic tasks in knowledge-based reasoning; high-level building blocks for expert systems design. IEEE Expert, 1986,1(4);23-30.
- 10 Clancey W J. Heuristic classification. Artificial Intelligence, 1986,27;289-350.
- 11 Boehm B W. Software engineering. IEEE Trans. on Computer, 1976,C-25;1226-1241.
- 12 Kamel R F. Effect of modularity on system evolution. IEEE Software, 1987(1);48-54.
- 13 Partridge D. Artificial intelligence; applications in the future of software engineering. Mass.; Addison-Wesley, 1986;54.
- 14 Zualkernan *et al.* Expert system and software engineering—ready for marriage? IEEE Expert, 1986,1(3);24-31.

SOFTWARE ENGINEERING METHOD OF BUILDING EXPERT SYSTEMS

Liu Qun and Huang Houkuan

(Harbin Shipbuilding Engineering Institute, Harbin 150001)

Yu Zhishou

(Harbin Institute of Electrical Technology, Harbin 150040)

Abstract This paper points out that some problems in designing and maintaining are caused with some inherent difficulties in expert system. The paper makes the suggestion of software engineering method about ES, the knowledge specification(KS) about cognitive process, and PLWF model of the process for building ES. The problems could be solved with PLWF model and KS, in the stage of building ES and in the stage of maintaining ES.

Key words Expert system, software engineering.