

ATR 信息处理的软件体系结构

崔文凯, 卢再奇

(国防科学技术大学 ATR 重点实验室, 湖南 长沙 410073)

摘要:为解决雷达自动目标识别设计中的整体设计框架问题,从 ATR 系统的特点出发,采用领域工程的分析方法对大量的 ATR 系统进行了系统化分析,在此基础上构建了基于代数学的构件模型,并给出 ATR 系统的模型。该模型作为 ATR 支撑平台开发的基础,支持 ATR 系统的快速开发。理论分析表明:该模型支持插头插座式的软件体系结构,为开放式 ATR 支撑平台提供了理论基础,为后续的 ATR 系统建模提供了构件模型规范,同时为 ATR 构件库的创建提供了支撑。

关键词:ATR;领域工程;软件工程;软件体系结构;代数学

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2009)01-0046-03

Software Architecture of ATR Information Processing

CUI Wen-kai, LU Zai-qi

(State Lab. of ATR, National University of Defense Technology, Changsha 410073, China)

Abstract: To solve the problem of the integral framework for the radar ATR designs, for development of ATR data processing platform, based on domain engineering and software engineering, analyses ATR system's hierarchy. Combine with ATR self-characteristic, adopt modular design, construct component model based on algebra, list its formal semantic. Theoretical analysis shows that the model support plug and socket architecture, construct a general component specification, provides standards for future open ATR platform. Provide component model for the future ATR system modeling, meanwhile supports foundation of the ATR component library.

Key words: ATR; domain engineering; software engineering; software architecture; algebra

0 引言

针对 ATR 的研究热情方兴未艾,但是预算有限,新方法和算法也在竞争资助机会。ATR 算法开发者关注的都是一个领域,如特征提取、特征选择、识别、评估中算法的开发,不同的算法基于不同的理论。ATR 系统研制人员相互沟通困难。

ATR 系统理论尚未引起重视,国内关于 ATR 的研究大多局限于算法理论的层次,而 ATR 是一个复杂的信息感知与处理系统。ATR 评估理论尚不健全,缺乏有效的、一致的评估方法,使得在设计实用化 ATR 系统时难于判定 ATR 算法的优劣。

如何克服 ATR 技术研究中的困难,推动 ATR 技术进程,服务于日益迫切的军事需要,需要克服以上阻碍 ATR 技术应用的不利因素。必须建立一个平台以支持这些研究工作的开展。

每一个 ATR 系统的开发过程中有许多共同的知识,如需求分析、测试、系统设计、代码等。为复用过去的知识,加快开发步伐,需要制定构件标准,并建立 ATR 系统开发规范。

开放式的 ATR 信息处理支撑平台的特点:ATR 系统的快速开发,同时能够提供性能评估、算法的快速原型验证,满足日益迫切的需要。

ATR 平台的建立,首先需要对 ATR 系统进行领域分析,然后建立构件模型。但是目前工业界的构件模型都是非形式化的,而形式化方法采用明确数学定义的语言对软件的期望特性或者行为进行精确、简洁描述。采用范畴论定义构件间的关系,从宏观层次上对不同层次的各种构件关系进行跟踪,支持可视化建模。

文中针对 ATR 信息处理系统的开发,首先开展领域分析,建立 ATR 信息处理系统的层次结构,引出软件体系结构的选择,再构建构件模型,给出相应的形式化语义,提出适于 ATR 信息处理系统开发的软件体系结构,最后采用范畴论,给出 ATR 系统的基于软件体系结构的开发模型,建立统一的标准及规范。

收稿日期:2008-04-01

基金项目:国防预研基金项目

作者简介:崔文凯(1980-),男,山东济宁人,硕士研究生,主要研究领域为自动目标识别、软件工程;卢再奇,副教授,博士,主要研究领域为自动目标识别、信号处理。

1 领域分析

1.1 领域分析概念

领域分析是对特定应用领域中共同的特征、知识、需求的标识、分析和规约,如图1所示。

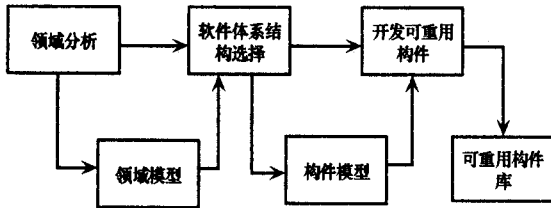


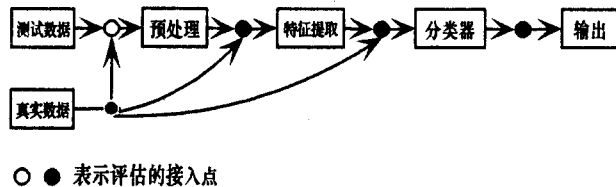
图1 领域工程

领域分析是特定领域内软件重用的基础,它的目标就是:发现和挖掘在特定领域内可以被复用的构件。

基于软件构件的软件工程强调领域工程与软件工程同时进行。领域工程创建应用领域的模型,标识、构造、分类和传播一组可重用的软件。在 ATR 信息处理平台开发中,采用领域工程进行分析,基于纵向复用建立 ATR 信息处理系统的层次结构。

1.2 ATR 信息处理系统层次结构

在对 ATR 采用领域工程进行抽象化分析时,采用纵向分析的策略。通过对 ATR 系统数据处理流程进行纵向分析,数据处理流程可以划分为:数据采集、预处理、特征提取、分类、结果输出、性能评估,可以建立 ATR 系统的层次结构^[1](如图2所示)。



○ ● 表示评估的接入点

图2 ATR 系统的层次结构视图

作为 ATR 信息处理系统开发的高层构件进行开发,支持后续的算法评估、算法验证和算法的快速原型验证,并对构件库进行功能划分。

ATR 系统的层次结构在纵向方向上可以将功能层进一步细化并将任务指派给下一层的构件^[2]。层次化的结构易于理解、管理、跟踪和测试,方便 ATR 信息处理系统的评估。

1.3 软件体系结构

体系结构是系统实现的蓝图,为构件的组装提供了基础和上下文^[3,4],及文献[5]中3种不同的体系结构类型。传统上,ATR 信息处理系统和 ATR 算法的开发通常采用自底向上的开发方法,但是自底向上的开发方法不适合软件工程实践,对于构件的集成而言,“设计重用”优于“构件重用”,一个自顶向下的软件体系结构方法可以为构件重用提供环境^[2]。

结合纵向复用和横向复用的方法自上而下地建立

有层次的 ATR 系统结构,层次化的体系结构对复杂系统进行层次化抽象,而层次化的体系结构易于理解,支持重用。

随着 ATR 理论的发展,算法会越来越复杂。采用形式化方法可避免模糊性,并支持 ATR 信息处理系统开发过程中的快速原型验证^[6]。在 ATR 信息处理系统的开发中以体系结构为基础的构件组装将加快 ATR 信息处理系统的开发,提高 ATR 信息处理系统开发的效率。

2 构件模型

ATR 信息处理系统开发过程中的共同问题已在领域分析中得到了一般认识,并将通过构件模型的建立统一进行描述和解决。构件模型的建立采用抽象的代数理论进行定义,构件的演化及构件的组合关系需要给予精确的描述。

以层次性体系结构为中心的 ATR 信息处理系统开发涉及三个方面的内容^[2]:目标分解、体系结构设计和验证。目标分解是任务分配关系,体系结构设计则是构件间的协同规则,验证是体系结构设计是否满足任务目标。建立支持插头插座式的构件模型如下:

定义1 构件模型是树状的,由不同抽象层次的体系结构模式组成集合 $\{M_0 \rightarrow M^i | i \in N | j \leq n\}$, N 为体系结构设计模式的数量,表示 M_0 可以将任务目标按某种关系分为 N 个任务子目标。

定义2 体系结构可以描述为一个二元组 (M, L) ,其中: M 是组成体系结构的构件集合, L 是构件间的关系。

定义3 软件系统的构件规范是一个三元组 $C = (R, S, P)$,其中: R 是构件描述的集合,方便构件库的分类、存储和检索; S 是构件标识的集合; P 是构件实现的集合。

定义4 构件描述是一个五元组 $R = (E, F, G, H, K, B)$,其中: E 是应用领域的集合; F 是构件功能的集合; G 是抽象层次的集合; H 是操作的集合; K 是操作对象的集合; B 是返回类型的集合。

定义5 构件标识是一个四元组 $S = (W, A, I, D)$,其中: W 是构件组成的集合; A 是组成构件的属性集合; I 是构件接口的集合; D 是构件接口连接时所影响的属性集合。

推论1:如果 $S_1 = (W_1, A_1, I_1, D_1)$, $S_2 = (W_2, A_2, I_2, D_2)$ 和 $S_3 = (W_3, A_3, I_3, D_3)$ 为构件标识,若存在 $f: S_1 \rightarrow S_2$ 和 $g: S_2 \rightarrow S_3$ 的映射,则有 $f \circ g: S_1 \rightarrow S_3$ 。

定义6 构件接口是一个二元组 $I = (O, N)$,其

中: O 是对外提供的功能、非功能集合和所实现的条件约束; N 是对外请求的功能、非功能集合和所实现的条件约束。

推论 2:若有构件 C_1 和 C_2 ,其构件接口分别为 $I_1 = (O_1, N_1)$ 和 $I_2 = (O_2, N_2)$, $I_1 \cdot O_1 = I_2 \cdot O_2$, $I_1 \cdot N_1 = I_2 \cdot N_2$, 则 C_1 和 C_2 可以实现动态替换。

定义 7 对外提供的功能、非功能集合和所实现的条件约束是一个四元组 $O = (OUT, Q, PRE, POST)$, 其中: OUT 是对外提供服务的集合; Q 是构件非功能集合; PRE 是前置条件的集合; $POST$ 是后置条件的集合。

定义 8 对外请求的功能、非功能集合和所实现的条件约束是一个四元组 $O = (IN, Q, PRE, POST)$, 其中: IN 是对外请求的服务集合; Q 是构件非功能集合; PRE 是前置条件的集合; $POST$ 是后置条件的集合。

构件的组合可以通过接口的规约耦合完成构件组装集成,构件的代数模型支持插头插座式的软件体系结构,支持 ATR 信息处理支撑平台扩展为通用开放式平台。

3 ATR 系统的模型

范畴论^[6],也称为箭头理论,是采用统一的方法描述——基于不同数学领域——各种结构概念的代数理论,支持图形化建模,以图形的形式给出直观的理解。

3.1 基础知识

引入范畴理论对构件的属性和行为特征进行抽象而提出的用于描述体系结构的构件初始化运算、分解运算和例化运算的新方法^[6]。该方法能够描述任意软件系统的体系结构和支持软件演化^[2]。

下面给出范畴论的基本性质:

(1) 态射:任意态射 $f: A \rightarrow B$, 有 $\text{dom}(f) = A$, $\text{ran}(f) = B$, 则 $\forall f: A \rightarrow B; \forall g: B \rightarrow C \cdot \exists h: A \rightarrow C \Rightarrow h = f \cdot g$

(2) 不相关的并集可以表示为: $A + B = \{a \mid a \in A\} \cup \{b \mid b \in B\}$, A 和 B 是属于不相关子系统的构件集合。

给出了代数理论的两条基本性质后,现在给出在 ATR 信息处理系统开发中涉及的基于范畴理论的关系图表,如图 3 所示。

3.2 ATR 系统开发

从 ATR 信息处理系统和 ATR 算法自身的特点出发,构件之间具有各种各样的关系,如执行、利用、扩展

等。ATR 数据处理流程:预处理、特征提取、分类、输出等为顺序关系,因此,ATR 信息处理系统的开发过程中,构件的顶层组装机制为连接组装机制。而在复合构件由原子构件的组装还包括复制组装机制。构件连接组装是采用支持插头插座式的规约层次上的匹配来完成构件的组装,连接组装的复合构件的功能行为是各个构件功能行为的并行执行。

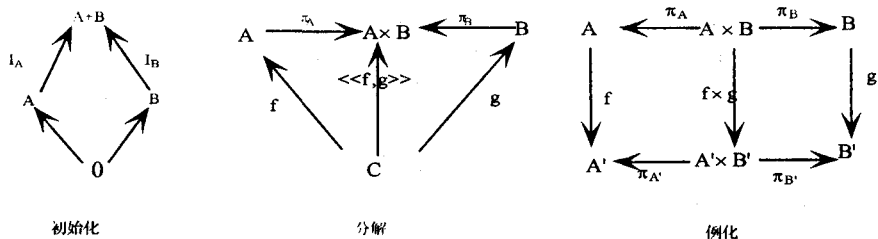


图 3 态射图

ATR 系统模型是不同抽象层次的体系结构的有机组合,层次化的结构支持自上而下的构件重用。在 ATR 的层次结构如图 4 所示。

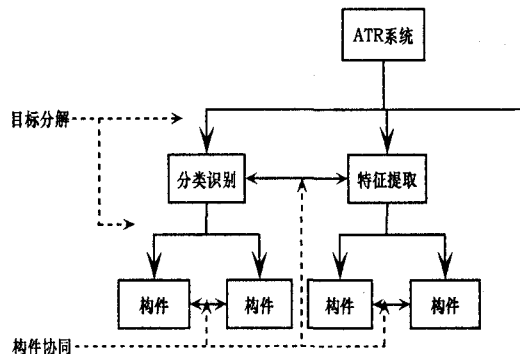


图 4 ATR 系统的层次结构

采用范畴论建立图 4 所对应的层次抽象模型,如图 5 所示。

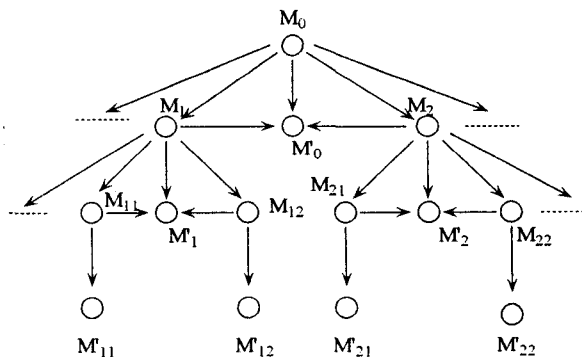


图 5 ATR 系统抽象模型

采用自顶向下的体系结构。当 ATR 信息处理系统的需求 M 与 M_1 或 M_2 相同时,可以直接 M_1 或 M_2 以下的设计模式,复用层次由构件重用上升为设计重用。更高的抽象层次复用,可以提高复用率,加快 ATR 信息处理系统的开发速度,并提高软件质量。

(下转第 52 页)

本模型开发的软件投入使用后,应用效果良好。系统实现了网络内及远程机关部门信息共享、生产计划制定、库存微机管理、成本自动核算、设计参数自动提取、报表自动生成等功能,从而提高了整体工作效率、降低了成本。

4 结束语

分别使用活动网络、基于一般 Petri 网的工作流网和层次颜色工作流网对采煤机械制造业的经营管理过程进行建模,再次证明了基于层次颜色工作流网的过程模型的优点,具备更丰富的语义表达能力、层次化表达能力和描述事务属性的能力。

参考文献:

- [1] The Workflow Reference Model[EB/OL]. 2007. <http://www.wfmc.org/standards/model.htm>.

- [2] 万和平,王明哲.层次工作流 Petri 网建模与分析[J].计算机工程与应用,2005,41(15):211-214.
- [3] van der Aalst W. Verification of Workflow nets[J]. Lecture Notes in Computer Science,1997,1248:407-426.
- [4] Desel J, Esparza J. Free Choice Petri Nets[J]. Cambridge Tracts in Theoretical Computer Science,1995,40:91-101.
- [5] CPN Tools[EB/OL]. 2007. <http://wiki.daimi.au.dk/cpn-tools/cpn-tools.wiki>.
- [6] Billington J, Gallasch G E, Han Bing. A Coloured Petri Net Approach to Protocol Verification[J]. Lecture Notes in Computer Science,2004, 3098:210-290.
- [7] 谢玉凤,杨光信,史美林.基于条件化有向图的工作流过程优化[J].计算机学报,2001,24(7):729-735.
- [8] 肖郑进.面向企业应用的工作流精简建模研究[D].杭州:浙江大学,2006:32-43.

(上接第 45 页)

参考文献:

- [1] 章毓晋.图像处理和分析[M].北京:清华大学出版社,2005.
- [2] Brown M, Lowe D. Invariant Features from Interest Point Groups[C]//In Proceedings of the 13th British Machine Vision Conference. Cardiff:[s. n.],2002:253-262.
- [3] Lowe D. Object Recognition from Local Scale-Invariant Features[C]//In Proceedings of the International Conference on Computer. Corfu, Greece:[s. n.],1999:1150-1157.
- [4] 周长发.精通 Visual C++ 图像处理编程[M].北京:电子工业出版社,2006.

- [5] 何斌,马天予,王运坚,等. Visual C++ 数字图像处理[M].北京:人民邮电出版社,2002.
- [6] 徐正光,田清,张利欣.图像拼接方法探讨[J].微计算机信息,2006,22(3):27-30.
- [7] Uyttendaele M, Eden A, Szeliski R. Eliminating ghosting and exposure artifacts in image mosaics[C]// In Proceedings of the International Conference on Computer Vision and Pattern Recognition. Kauai, Hawaii:[s. n.],2001:509-516.
- [8] Szeliski R, Kang S. Direct Methods for Visual Scene Reconstruction[C]// In IEEE Workshop on Representations of Visual Scenes. Cambridge:[s. n.],1995:26-33.

(上接第 48 页)

4 结束语

分析了 ATR 信息处理系统的数据处理流程,提出了基于领域工程的构件模型,并采用代数语言进行了形式化描述。借助于范畴理论给出了 ATR 系统抽象模型。进一步的工作就是在构件模型的基础上支持算法评估和算法的优化选择。ATR 信息处理支撑平台的建立将为 ATR 系统和 ATR 算法的开发制定标准,推动 ATR 技术实用化的进程。

参考文献:

- [1] Ruda H, Snorrason M, Shue D. Framework for automatic target recognition optimization[R]. U S: Charles River Analytics, 1999.

- [2] 楚旺,钱德沛.以体系结构为中心的构件模型的形式化语义[J].软件学报,2006,17(6):1287-1297.
- [3] Shaw M, Garlan D. Software Architecture: Perspectives on an Emerging Discipline[M]. Englewood Cliffs, NJ: Prentice Hall, Inc., 1996.
- [4] Yang Fu qing. Software reuse and relevant technology[J]. Computer Science, 1999,26(5):1-4.
- [5] Luckham D, Vera J, Meldal S. Three concepts of system architecture[R]. US: Stanford University, 1995.
- [6] Guo Jiang. Using Category Theory to Model Software Component Dependencies dependencies. [C]//In: Proc. of the 9th Annual IEEE Int'l Conf. and Workshop on the Engineering of Computer-Based Systems (ECBS 2002). [s. l.]: IEEE Computer Society, 2002:185-192.