计算机科学 2003Vol. 30№. 1

化学抽象机的分析与应用研究*)

赵 恒 王振宇 曹万华 叶俊民
(武汉数字工程研究所 武汉 430074)

Analysis and Application Research on Chemical Abstract Machine

ZHAO Heng WANG Zhen-Yu CAO Wan-Hua YE Jun-Ming
(Wuhan Digital Engineering Institute, Wuhan, Hubei, 430074)

Abstract This paper analyzes and studies the form and the ability of the Chemical Abstract Machine, or CHAM, on describing the system software architecture. After some expanding, the CHAM is applied to describe formally the software architecture of command and control system. It is expected that the specification of the system requirements and the software test plan would be automatically generated from the formal software architecture description in the level of software architecture.

Keywords Chemical abstract machine (CHAM), Software architecture, Command and control system

1 引言

化学抽象机形式化语言(CHAM : the Chemical Abstract Machine)最早是由法国科学家 Gerard Berry 和 Gerard Boudol 提出来的,它主要用于异步并行计算模型的建模,通过把化学反应和抽象机的概念有机地结合来描述系统状态的变化。它把一个系统的状态看成化学溶液,溶液由分子组成,分子根据一定的反应规则相互反应又引起新的系统状态变化。溶液中不同的分子可以按照反应规则平行地进行反应,只要各自反应的分子集不重叠。由于 CHAM 在描述系统动态性、并行性方面的优良特性,因而可以较好地描述异步并行计算模型。

随着软件体系结构技术研究的不断发展和深入,研究人员和软件开发人员越来越充分地认识到研究大型复杂软件系统的软件体系结构的重要性,因为软件体系结构的研究可以大幅度提高软件的重用度,降低开发费用,并提高软件质量。然而要通过软件体系结构的研究实现这一目标,首先必须用某种方式描述软件体系结构。因此先后出现了十几种软件体系结构描述语言(ADL: Architecture Descripe Language)。但这些语言大多注重软件系统结构静态特性的描述,而对其动态特性描述不足。意大利学者 Paola Inverardi 和美国学者 Alexxander L. Wolf 首先把 CHAM 应用于描述和分析软件体系结构,他们充分利用 CHAM 擅长于描述系统动态性和并行性的优点,用 CHAM 形式化方法描述和分析了软件体系结构动态操作性语义,在软件体系结构动态特性描述方面进行了有益的扩展。

通过分析和研究 CHAM,我们发现用 CHAM 描述软件体系结构不仅可以清楚地反映系统的静态和动态特性,而且它具有简单易懂,易于描述,易于扩展的特点,可以适合较大范围的领域软件体系结构的描述。为此,我们将 CHAM 应用于指挥控制系统的研究与设计,对指挥控制系统的软件体系结构进行形式化、规范化描述,并以期进一步导出形式化的系统需求规格说明,并在软件设计的早期就能自动完成软件测试设计等,从而在软件体系结构的层次上为软件的质量作相应保证。

本文首先简要说明了 CHAM 的形式化语义,然后给出用 CHAM 形式化描述典型的指挥控制系统软件结构的一个实例,最后通过对 CHAM 进行分析和研究,给出一些应有的结论。

2 CHAM 形式化语义

一个化学抽象机由一组分子 m,m',…、溶液 s,s',…和变换规则 T,T',…组成。分子是 CHAM 的基本元素,由一个常数集和操作符集派生而成的句法代数定义;溶液是有限多个分子的集合,它反映了系统的某种状态;溶液中的分子根据变换规则进行反应,以指示溶液演化的方式 S→S'。

变换规则从应用范围可分为:通用规则,即在整个 CHAM 中通用的规则;专用规则,适用于某些特定分子的规则。从反应作用可分为:加热规则,把大分子分解成小分子的规则;冷却规则,小分子合成大分子的规则;以及反应规则。一个变换规则通常表示为:

$M_1, M_2, \cdots, M_k \rightarrow M_1', M_2', \cdots, M_k'$

如果 $m_1, m_2, \cdots, m_k$ 和 $m_1', m_2', \cdots, m_k'$ 是分子的实例,则可以应用此规则而得到以下反应规则:

$m_1, m_2, \cdots m_k \rightarrow m_1', m_2', \cdots, m_k'$

如果某溶液中包含反应规则中的分子,则溶液也随之发生演化。即,

$$\frac{S \longrightarrow S'}{S \cup S'' \longrightarrow S' \cup S''}$$

在 CHAM 中,具有一定相互关系的多个分子还可以用膜(membrane)进行封装,表示成一个大分子。记作$\{|m_1, m_2, \cdots m_k|\}$。其反应方式与溶液是一致的。即

*)本文工作得到十五国防预研项目资助(项目编号 10104010202 和 41310601)。赵 恒 博士研究生,主要研究方向:软件工程。王振宇 研究员,博士生导师,主要研究方向:软件工程,软件理论与工具开发。曹万华 研究员,主要研究方向:软件工程。

$$\frac{S \longrightarrow S'}{\{|S|\} \longrightarrow \{|S'|\}}$$

由此，用CHAM描述的软件体系结构规格说明包括四个部分：

· 体系结构组成构件(即分子)的语法描述，构件被分为三类：数据元素、处理元素和连接元素；

· 系统的初始状态，用溶液 S_0 表示。初始溶液是所有按照分子语法可能产生的分子的一个子集，它实际上是系统的一个初始的、静态的配置；

· 一套反应规则，通过描述构件间的交互活动以实现系统动态行为的描述；

· 一组表示系统状态变化的溶液集。

下面我们通过一个典型的应用实例来说明CHAM在指挥控制系统中的应用方法。

3 指挥控制系统的CHAM规格说明

3.1 指挥控制系统描述

指挥控制系统是作战系统的核心，是一个网络化的分布式综合处理、显示和控制系统，它收集来自雷达、数据链、电子战、导航等传感设备送来的各种信息，以及武器系统送来的各种状态信息，显示各种态势图形及各种参数和状态页面，并对所有的信息进行综合处理，在指挥人员的干预下，完成辅助战术计算和辅助战术决策，实施武器的分配和指挥控制，完成整个作战指挥任务。

典型的指挥控制系统软件主要由以下几个部件组成：

· 报文发送处理：负责向外发送各种处理后的信息；

· 报文接收处理：对各种信息进行接收并分类；

· 综合情报处理：对各种传感器信息进行检测、相关、融合等多级综合处理，以得到精确的战场态势数据和目标的威胁估计，构成综合战场态势的基础；

· 综合显示：显示目标综合态势图形、目标参数表页和各种战术决策结果；

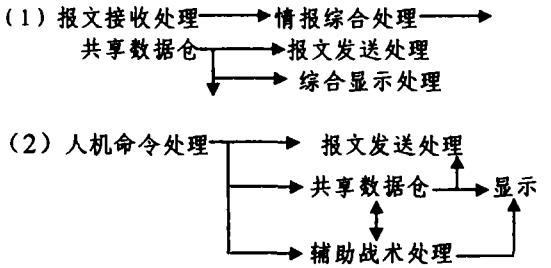
· 辅助战术处理：完成目标威胁判断和各种战术计算，提供辅助决策和武器分配；以及各种作战方案和作战预案的生成等；

· 人机命令处理：处理各种人机交互命令；

· 数据库管理：负责各类数据的存储、检索与维护。

这些部件适合于大多数指挥控制系统，具体应用时可根据特定的环境和需求进行相应的扩充或剪裁，因此都是可重用部件。它们之间的关系及系统功能结构如图1所示。

从图1可以看到，在指控系统中有两个并行运行的流程：



这两个流程分别独立运行，通过共享数据仓发生关联。

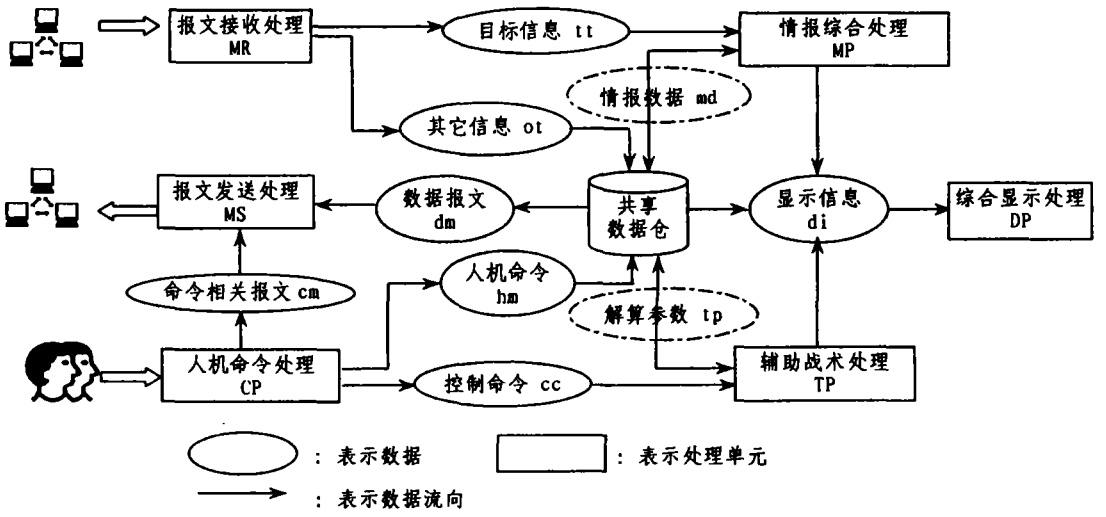


图1 指挥控制系统功能结构简化图

3.2 指挥控制系统软件体系结构的CHAM描述

(1)定义 为了描述方便，我们对所使用的符号和标识符作一定义。对于处理元素我们一律用大写字母，而对于数据元素用小写字母。

处理元素包括：报文接收处理：MR；报文发送处理：MS；人机命令处理：CP；情报综合处理：MP；辅助战术处理：TP；综合显示处理：DP。

数据元素包括：目标信息：tt；数据报文：dm；人机命令：hm；其它信息：ot；解算参数：tp；情报数据：md；命令相关报文：cm；显示信息：di；情报显示信息：mdi；解算显示信息：tdi；控制命令：cc。

另外，共享数据仓：repository。

(2)描述方式的扩展 ①用CHAM描述系统软件体系结构中通常是把整个系统的初始状态作为初始溶液，即系统是由一个初始溶液起始的单一流程[1]。这对指控系统来说是不够的，为此，我们在初始溶液中引入“//”符号，表示两个平行的分子或溶液。这两个平行的分子或溶液可同时根据反应式平行进行反应，只要两个反应集不相交。

②在文[1]中对分子的定义是：

M::=P|C|M◇M

C::=i(D)|o(D)|ir(D)|or (D)|repository

P::=处理元素

D::=数据元素

定义中 M 表示分子;C 是连接元素,但它并不是体系结构中所定义的连接件,它只是定义了数据元素的移入/移出(import/export)。按照这种定义方法,单独一个的连接元素也作为一个独立的分子参加反应,这在概念上是不完全的。因为这里的连接元素反映的是处理元素数据的移入/移出,所以必须与一个处理元素连接才能参加反应,也就是说,每个分子都必须包含至少一个处理元素,仅仅是单个的或多个连接元素不能构成分子直接参加反应。

所以,我们对分子进行重新定义如下:

M::=P|C◇M|M◇C|M◇M|M//M 。

③软件体系结构研究的重点是系统构件以及构件间的连接关系,不同的连接方式反映了不同的体系结构风格。因此,连接件同处理构件的一样是第一类构件[2]。而文[1]中对连接元素的定义没有反映分子间如何连接。为此,我们根据应用的需要对连接元素的定义进行了扩充,以反映分子间的连接方式。指控系统中所用到的连接方式归纳为三种:消息队列 Msg;管道机制 Pip;共享数据区机制 Shm。

④在对共享数据区机制的处理中,我们认为:共享数据仓(repository)作为存放数据的地方,它对数据的处理与消息队列和管道不同。消息队列和管道是什么样的数据进来就什么样的数据出去。而共享数据仓中的数据可以重新组织,也就是说可以把共享数据仓看成有自己的输入/输出。因此,在本文中把共享数据仓作为一种处理元素,而不是象文[1]中把共享数据仓作为一种连接元素。

因此,连接元素的定义如下:

C::=msg_r(D)|msg_w(D)|pip_r(D)|pip_w(D)|shm_r(D)|shm_w(D)|ir (D)|or (D)

其中,msg_r(D)表示处理元素从消息队列中读入数据元素,而 msg_w(D)表示处理元素向消息队列中写数据元素。pip 和 shm 类似。ir 是共享数据仓等待输入数据,or 则是共享数据仓准备输出数据。

综上所述,分子的完整定义如下:

M::= P|C◇M|M◇C|M◇M|M//M

P::=处理元素|repository

D::=数据元素

C::=msg_r(D)|msg_w(D)|pip_r(D)|pip_w (D)|shm_r (D)|shm_w(D)|ir (D)|or (D)

这样对分子的定义就比较完整、准确,而且在后面定义反应式时不会出现矛盾。

(3)具体描述 说明指挥控制系统软件体系结构的第一步是定义其分子 M 集合 Σ_0 的语法结构。

M::=P|C◇M|M◇C|M◇M |M//M

P::=MR|MS|CP|MP|TP|DP|repository

D::=tt|ot|dm|hm|tp|mg|cm|di|mdi|tdi|cc

C::=msg_r (D)|msg_w (D)|pip_r (D)|pip_w (D)|shm_r (D)|shm_w (D)|ir (D)| or (D)

其中,M 表示分子,它或是 P 或是由中缀符"◇"或"//"与 C 复合而成;P 是处理元素;D 是数据元素;C 是连接元素。

中缀符◇用来表达处理元素关于其输入输出行为的状态。比如,msg_r(tt)◇pip_w(di)◇MP,它表示综合情报处理器 MP 准备从消息队列中读入数据 tt,向管道输出数据 di。MP 的状态的变化可以理解为"从左到右"。再比如,pip_w(di)◇MP◇msg_r(tt),表示 MP 已经读入完毕,开始输出 di;而 MP◇msg_r (tt)◇pip_w (di),则表示 MP 完成输入输出操作处于等待激活状态。

从以上定义的分子集中可以看出系统的组成构件、连接件和数据。

为了描述系统的动态变化,接下来是定义指挥控制系统软件体系结构的初始溶液 S_0。S_0 实际上系统体系结构的一种初始静态配置,是系统动态变化的起点。由于在指挥控制系统中同时有两个输入点报文接收处理和人机命令处理,两者平行运行。因此,我们分别定义两个子溶液 S_{01}—报文接收,S_{02}—人机命令,而整个系统的初始溶液 $S_0=S_{01}//S_{02}$,其中:

S_{01}={|MR◇msg_w(tt)//MR◇shm_w (ot),msg_r(tt)◇shm_r(md)◇pip_w(mdi)◇shm_w(md)◇MP,pip_r(mdi)◇DP,ir(ot)◇repository, ir(md)◇msg_w(dm)◇repository,msg_r(dm)◇MS,repository◇or(md)|}

S_{02}={|CP◇pip_w (cm)//CP◇shm_w (hm)//CP◇msg_w (cc),msg_r (cc)◇shm_r (tp)◇pip_w(tdi)◇shm_w (tp)◇TP,pip_r(cm)◇MS//msg_r(dm)◇MS,ir(hm)◇or(dm)◇or(di)◇repository,i(tdi)◇DP//i(di)◇DP,ir(tp)◇repository,repository◇or(tp)|}

其中,MR◇msg_w(tt)//MR◇shm_w (ot)表示 MR 或向队列中输出 tt 或向共享数据仓中写入 ot 。而且 MR◇msg_w(tt)//MR◇shm_w (ot)这种状态表示上一次操作已完成,等待下一次激活。另外,为了简化起见,我们报文传输进来和发出去的过程不作为该体系结构的一部分,所以 MR 只有输出部分,MS 只有输入。CP◇pip_w(cm)//CP◇shm_w (hm)//CP◇msg_w (cc)同理。

S_{01}表示报文接收子流程的初始状态,S_{02}表示人机命令子流程的初始状态。以 S_{01}为例,其物理意义是:报文接收处理收到外部送来的报文信息后,对其进行分类处理,把目标信息送到消息队列中,准备由综合情报处理构件进行处理。而对于非目标信息则送到共享数据仓中。子流程中的其它构件则处于准备处理数据的状态。S_{02}同理。

定义了初始溶液后,最后一步是定义反应规则。反应规则通过作用于初始溶液上来定义系统如何从初始配置进行动态演化。

$T_1 \equiv m_1 // m_2 // \cdots // m_n \rightarrow m_1, m_2, \cdots, m_n$

$T_2 \equiv m◇o(d) \rightarrow o(d)◇m$

$T_3 \equiv msg\_r(d)◇m_1, msg\_w(d)◇m_2 \rightarrow m_1◇msg\_r(d), m_2◇msg\_w(d)$

$T_4 \equiv pip\_r(d)◇m_1, pip\_w(d)◇m_2 \rightarrow m_1◇pip\_r(d), m_2◇pip\_w(d)$

$T_5 \equiv shm\_r(d)◇m_1, shm\_w(d)◇m_2 \rightarrow m_1◇shm\_r(d), m_2◇shm\_w(d)$

$T_6 \equiv shm\_r(d)◇m, or(d)◇repository \rightarrow m◇shm\_r(d), repository◇or(d)$

$T_7 \equiv shm\_w(d)◇m, ir(d)◇repository \rightarrow m◇shm\_w(d), repository◇ir(d)$

$T_8 \equiv msg\_w(tt)◇MR, msg\_r(tt)◇shm\_r(md)◇pip\_w(mdi)◇shm\_w(d)◇MP, repository◇or(md) \rightarrow MR◇msg\_w(tt), shm\_r(md)◇pip\_w(mdi)◇shm\_w(d)◇MP◇msg\_r(tt)or(dm)◇repository$

$T_9 \equiv msg\_w(cc)◇CP, msg\_r(cc)◇shm\_r(tp)◇pip\_w(tdi)◇shm\_w(tp)◇TP, repository◇or(tp) \rightarrow CP◇msg\_w(cc), shm\_r(tp)◇pip\_w(tdi)◇shm\_w(tp)◇TP◇msg\_r(cc)or(tp)◇repository$

T_{10}≡MR◇msg_w(tt),MR◇shm_w (ot),MP◇msg_r (tt)◇shm_r(md)◇pip_w(mdi)◇shm_w(md), DP◇pip_r(mdi),repository◇ir(ot),repository◇ ir(md)◇msg_w (dm)MS◇msg_r(dm),reposito-ry◇or(md)→S_{01}

T_{11}≡CP◇pip_w (cm),CP◇shm_w(hm),CP◇msg_w (cc),TP◇msg_r (cc)◇shm_r (tp)◇pip_w(tdi) ◇ shm_w (tp),MS◇pip_r(cm), MS◇msg_r (dm),repository◇ir(hm)◇or(dm)◇or(di),DP◇ i(tdi),DP◇i(di),repository◇ir(tp),repository◇ or(tp)→S_{02}

其中,m_1,m_2∈M,d∈D。规则1是把大分子分解成平行的小分子,以便于他们可以平行地参加反应。规则2是激活一个处于等待状态的处理元素。这里的o(d)是一个通则式,它代表处理元素的所有输出操作,如:msg_w,shm_w,pip_w。规则3～规则5是处理元素的输入/输出反应规则。而规则6和规则7则是处理元素与共享数据仓之间的反应规则。它们都属于通用规则。规则8至规则9属于专用规则,即适用于特定分子的规则。规则10和规则11是返回到初始状态。

3.3 指挥控制系统软件体系结构CHAM描述分析

下面我们以S_{01}为例应用反应规则来跟踪系统状态的变化。

$$S_{01} \xrightarrow{T1,T2} S_{11}$$

S_{11}=msg_w (tt)◇MR,shm_w (ot)◇MR, msg_r(tt) ◇shm_r(md)◇pip_w(mdi)◇shm_w(md)◇MP, pip_r(mdi)◇DP,ir(ot)◇repository,ir(md)◇ msg_w (dm)◇repository, msg_r(dm)◇MS, repository◇or(md)

$$S_{11} \xrightarrow{T7,T8} S_{21}$$

S_{21}=MR◇msg_w(tt),MR◇shm_w(ot),shm_r(md) ◇pip_w(mdi)◇shm_w(md)◇MP◇msg_r(tt), pip_r(mdi)◇DP,repository◇ir(ot),ir(md)◇ msg_w (dm)◇repository,msg_r(dm)◇MS,or (md)◇repository

$$S_{21} \xrightarrow{T6} S_{31}$$

S_{31}=MR◇msg_w(tt),MR◇shm_w (ot),pip_w(mdi) ◇shm_w(md)◇MP◇msg_r(tt)◇shm_r(md), pip_r(mdi)◇DP,repository◇ir(ot),ir(md)◇ msg_w (dm)◇repository, msg_r(dm)◇MS, repository◇or(md)

$$S_{31} \xrightarrow{T4} S_{41}$$

S_{41}=MR◇msg_w(tt),MR◇shm_w (ot),shm_w(md) ◇MP◇msg_r(tt)◇shm_r(md)◇pip_w(mdi), DP◇pip_r(mdi),repository ◇ir(ot),ir(md)◇ msg_w (dm)◇repository, msg_r(dm)◇MS, repository ◇or(md)

$$S_{41} \xrightarrow{T7} S_{51}$$

S_{51}=MR◇msg_w(tt), MR◇shm_w (ot),MP◇msg_r (tt)◇shm_r(md)◇pip_w(mdi)◇shm_w(md), DP ◇pip_r(mdi),repository◇ir(ot),msg_w (dm) ◇repository◇ ir(md),msg_r(dm)◇MS,reposito-ry◇or(md)

$$S_{41} \xrightarrow{T3} S_{51}$$

S_{51}=MR◇msg_w(tt),MR◇shm_w (ot),MP◇msg_r (tt)◇shm_r(md)◇pip_w(mdi)◇shm_w(md), DP◇pip_r(mdi),repository◇ir(ot),repository◇ir (md)◇msg_w (dm)MS◇msg_r(dm),repository ◇or(md)

至此一次操作完成,再根据规则10又回到初始溶液状态,又可开始新的反应。

从上面系统状态变化我们可以看到,CHAM规格说明是一个基于操作的系统框架,这种框架不会把所描述的系统曲解为某种特定的计算模型[1]。在文[3]中,我们用Garlen和Shaw关于体系结构的定义对指挥控制系统的软件体系结构做了另一种风格的描述。两者相比较,可以发现Garlen和Shaw着重从构件连接的角度来对体系结构进行描述,这是一种系统静态特征的描述。而扩展后的CHAM描述不仅可以描述系统的静态特征,而且还能从系统操作的动态性方面来进行描述,因而可使软件开发人员很快地了解系统功能和行为,而且简单易懂,适用于多种层次的用户。同时CHAM是一种形式化表示方法,而形式化方法是提高软件质量的重要途径,在从高层规范到最终实现的过程中,选用适当的、以形式化方法为基础的工具进行辅助设计和验证对于提高诸如指挥控制系统这样的安全性、可靠性要求较高的系统是非常有帮助的。

结论与下一步工作 本文的目的是为了探索CHAM对于描述和分析指挥控制系统的适用性,希望能从体系结构的层次对指挥控制系统进行形式化的描述和分析,从而进一步为能自动生成需求规格说明、软件测试计划等提供形式化基础。从上面我们可以看到CHAM是一个非常有用的、表达能力强的体系结构描述工具。虽然同其它体系结构描述语言一样,它不可能适应软件体系结构描述和分析的所有方面,但我们认为它对于指挥控制系统领域体系结构的描述是合适的。

下一步我们将在CHAM形式化描述的基础上研究专用领域应用软件集成测试计划的自动生成。因为CHAM这种对系统状态变化的描述特别适合于测试系统的行为和功能,其形式化描述从软件体系结构级别上为系统的集成测试提供了一个恰当的抽象级别。

参考文献

1 Inverardi P,Wolf A L. Formal Specification and Analysis of Software Architectures Using the Chemical Abstract Machine Model. IEEE Transaction on Software Engineering, 1995, 21(4):373～386

2 Shaw M,Garlen D. Software Architecture: Perspectives on an Emerging Discipline. Englewood Cliffs, NJ: Prentice Hall, Inc., 1996

3 赵恒,曹万华,王振宇. 软件体系结构在舰载指挥控制系统中的应用研究. 舰船科学技术,2000.06

4 Berry G,Boudol G. The Chemical Abstract Machine. Theoretical Computer Science, 1992, (96):217～248

5 Inverardi P, Yankelevich D. Relating CHAM Descriptions of Software Architectures. In: Proc. of the 8th International Workshop on software Specification and Design,IEEE Computer Society Press,1996.66～74