文章编号:1001 - 2486(2003)06 - 0062 - 05

指挥控制的形式化描述与性质验证*

邓小妮,袁卫卫,曾 熠,罗雪山
(国防科技大学人文与管理学院,湖南 长沙 410073)

摘 要:在分析了 IDEF0 基本模型及其军用模型的基础上,结合面向对象的分析方法,提出了一个通用的指挥控制对象的概念模型,并采用形式化描述语言 LOTOS(Language of Temporal Ordering Specification) 和基于动作的时序逻辑 ACTL(Action Based Temporal Logical) 对系统进行了形式化描述和性质验证。这为 C^4ISR 系统的需求描述和验证提供了一种新的思路和方法。

关键词:指挥控制;C^4ISR;形式化描述;验证;LOTOS
中图分类号:TP301.2 文献标识码:A

The Formal Specification and Property Verification of the Command and Control

DENG Xiao-ni, YUAN Wei-wei, ZENG Yi, LUO Xue-shan
(College of Humanities and Management, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: A conceptual model of the command and control object (C^2O) is described using the formal language LOTOS(Language of Temporal Ordering Specification), and the main properties of C^2O and C^2O-based systems are defined using ACTL(Action Based Temporal Logical). Then a model-based-verification approach is provided aiming at the above key properties verification. The practical example shows that the new approach is viable.

Key words: command and control; C^4ISR; formal specification; verification; LOTOS

C^4ISR 系统的核心是指挥控制(C^2) 系统,多年来 C^4ISR 理论的研究重点一直放在 C^2 上。Joel S. Lawson 在 20 世纪 70 年代中期最早提出了指挥控制过程的概念模型[1],他用动词"监测(sense)、处理(process)、比较(compare)、决断(decide)、行动(act)"来刻画这个过程。在 Lawson 的基础上,不少人对这个模型进行进一步的细化、调整。A. H. Levis 等人在 Lawson 过程模型的基础上提出了以 Petri 网为基础的 C^2 表示工具,将 Petri 网方法推广到 C^2 系统的描述上。

20 世纪 90 年代以来,军事综合电子信息系统的理论研究有了突破性进展,其主要体现是美国国防部提出的《2010 年联合作战构想》和《C^4ISR 体系结构框架》等总体框架性文件,它们是战场信息系统理论研究成果的缩影。对 C^4ISR 系统体系结构的一体化设计和描述成为重点。IDEF0 作为一种全面而强大的功能建模语言,被用以分析和描述 C^4ISR 系统的需求和体系结构[2]。但 IDEF0 不属于严格意义上的形式化语言,其语义表达较弱。为了在系统开发的早期,包括需求分析和体系结构设计阶段,尽早发现问题,需要对模型有更严格的描述和关键性质的定义和证明。

形式化方法是需求验证和系统验证的一种有效手段[3]。文献[3] 中,把形式化描述方法(FDTs) 分为如下两类:

(1) 可构造的 FDTs。在这种方法中,对目标系统的说明是为其构造一个模型,该模型的构成成分是一些具有已知特性的数据抽象,如域、元组、集合、序列、包、映射等。这种类型的 FDTs 一般是可执行的,即允许快速地得到目标系统的一个原型。其主要缺点是不能明确地表达系统的特性,特性只能作为隐含性质被验证[4]。

* 收稿日期:2003 - 04 - 23
基金项目:国家部委预研基金资助项目
作者简介:邓小妮(1970—),女,博士生。

(2) 面向性质的 FDTs。这种方法通过直接给出目标系统的一组特性来描述目标系统,通常是一组目标系统必须满足的形式公理。在面向性质的方法中,系统的特性以明确的方式表达,允许对每个特性进行单独分析,这有助于判定每个特性的重要程度和准确程度。

通常,在基于模型的验证(Model Based Verification, MBV) 中,将二者结合使用。

本文采用时序描述语言 LOTOS(Language Of Temporal Ordering Specification) 描述指挥控制对象的概念模型。LOTOS[5] 是精确定义的语言,其静态语义基于属性文法,动态语义基于代数。它是可执行的,也是可证明的,属于可构造型 FDT。它由两部分组成,一部分是基于 CCS(通信演算系统) 和 CSP(通信顺序进程) 的进程代数,用于表示系统的时序行为;另一部分是基于代数语言的抽象代数部分,用于构造、描述和操纵数据。在 LOTOS 中,并发系统被看作是具有内部动作的一系列进程(process)。每个进程可以通过定义在事件端口(gate) 的可见动作与其他进程进行交互。通过定义进程的一系列可见动作,可以描述一个进程所具有的外部行为特征 整个系统的行为可用一个动作树来表示,树中的每一个路径代表着进程可能执行的一个可见动作序列。

另一方面,采用基于动作的时序逻辑 ACTL(Action Based Temporal Logical) 定义指挥控制系统的一般性质。ACTL 是一种基于动作的分支时序逻辑[6],它的解释域为标号转移系统 LTS(Labeled Transition System)。在这类系统中,状态间的转移带有标号,标号是用来表示引起状态转移的动作的。一般地,时序逻辑是适于并发反应式系统的性质定义的,而指挥控制是典型的并发反应式系统,因此可以用 ACTL 来表示指挥控制模型的基本性质。而且一般而言,使用 LOTOS 描述的系统的语义模型,能够由一个标号转移系统表示。这样,利用计算机辅助工具,可以对 LOTOS 描述的系统模型进行检查,验证其是否具有 ACTL 定义的性质,从而保证指挥控制系统在某些方面的正确性和可靠性。

1 指挥控制对象的概念模型

首先分析 IDEF0 基本模型及其军事应用模型。以面向对象的分析方法来看,不难发现,文献[2] 的图 3 中“旅防空指挥”活动与其下层的三个“营防空指挥”活动之间存在很大程度的功能与结构上的相似与重复。于是,结合 IDEF0 基本模型(参见文献[2] 的图 1) 与文献[2] 的图 4,另外参考 Lawson 的指挥控制过程概念模型,我们给出一个通用的指挥控制对象(C^2O) 的概念模型,如图 1 所示。可以看出,指挥控制对象 C^2O 应当是一种重用性较高、具有相对独立功能的对象构件。它具有很强的与环境(战场) 和其他对象(上级和下级指挥控制对象) 交互的能力。其结构主要由三部分组成:

(1) 指挥控制元素 Cmand&Ctrl:包含情报融合、态势评估和制定计划三个独立而相关的单元。此部分主要用于接收来自外部的消息,并作出相应的反应和动作,即制定出相应的方案和计划。这些消息可以是来自战场(包括我军和敌军) 或其他外部环境的情报,也包括来自上级的任务和命令。

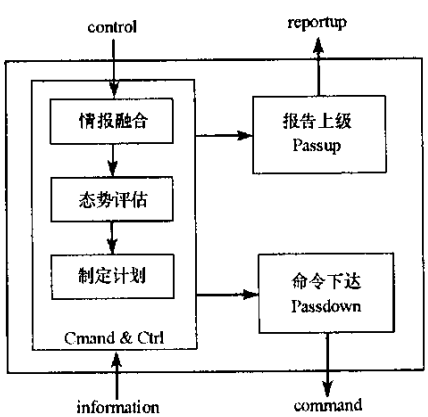

图 1 指挥控制对象(C^2O) 概念模型

Fig.1 The model of C^2O

(2) 与上级 C^2O 的接口单元 Passup。

(3) 与下级的接口元素 Passdown。

C^2O 可以通过四种端口(gate) 与外界交互。一方面,它从 information 端口接收来自外部的输入,从 command 端口输出命令到下级 C^2O,并改变本级的执行状态;另一方面,它可以从 control 端口接收来自上级的指令和计划,也可以从 reportup 端口向上级 C^2O 汇报情况。采用 LOTOS,我们对 C^2O 的交互行为严格地进行描述如下:

```
process C2O [information, command, control, reportup] : (infor _ data : InforData, control _ data :
                                                          CtrlData ) : noexit : =
(
information ?id : InforData ; reportup ! Passup(id) ;
C2O[information,command,control,reportup](id,control _ data
[] control ?cd : CtrlData ; command !Passdown(cd) ;
C2O[information, command,control,reportup](infor _ data, add(cd, control _ data))
)
endproc
```

C^2O 在接收到 information 事件后,将产生 reportup 事件,并依此递归下去。同样地,它在接收到 control 事件后,将产生 command 事件,同时将 cd 加入控制命令集合,并依此递归下去。这里使用数据代数方法定义进程参数相关的数据类型,与 reportup 事件相关的数据值由 Passup(id) 产生,而与 command 事件相关的数据值由 Passdown(cd) 产生。

2 指挥控制性质的定义

特别关注的性质有两点:

(1) 指挥控制对象 C^2O 的一般特点。这有助于发现 C^2O 的通用特征,从而完善 C^2O 模型。显然,对于 C^2O 而言,它应该在任何条件下都能够对来自外部环境或上级 C^2O 的输入事件作出相应的反应,进行内部处理,并产生相关的输出事件。这一性质可由下面两条 ACTL 公式定义:

$$AG\ (\ [information]\ E\ [true\ \ \{true\}\ U\ \ \{reportup\}\ true]\ ).$$

$$AG\ (\ [control]\ E\ [true\ \ \{true\}\ U\ \ \{command\}\ true]\ ).$$

上述公式的含义是,在任何路径(A 算子) 和状态(G 算子) 中,若 C^2O 接收到 information 或 control 事件,则其内部进程间总存在(E 算子) 一条路径,使 C^2O 产生 reportup 或 command 事件。

(2) 指挥控制系统的一般性质。这是为了检查指挥控制特性结构在设计上的逻辑完整性。在实际应用中,若系统涉及到的 C^2O 数目较多,且 C^2O 之间的协作关系较强,则系统的复杂度将显著增加。为了保证系统的可靠与稳定,以及设计的正确性,需要检查所设计的系统是否满足指挥控制系统的一般特性以及是否能够达到预期效果。这里仅讨论可达性。

可达性是指,当某一输入事件发生时,系统能否达到指定效果。这一性质允许我们验证任一 C^2O 的输入事件,能否引起隔级(可以不仅仅是直接上下级) 或同级其他 C^2O 的相应指挥控制单元部分对其产生反应。可达性分为上可达与下可达,其定义分别如下:

$$AG\ (\ [down\_infor\_action]\ E\ [true\ \ \{true\}\ U\ \ \{up\_Cmand\&Ctrl\_function\}\ true]\ ).$$

$$AG\ (\ [up\_ctrl\_action]\ E\ [true\ \ \{true\}\ U\ \ \{down\_Cmand\&Ctrl\_function\}\ true]\ ).$$

如果上式中的 down 和 up 在数值上相等,则说明同级可达。

3 模型检查方法与实例

3.1 模型检查

模型检查是检验以有限状态机为基本模型的系统的一种常用方法。对于并发反应式系统,人们常以标号转移系统作为其抽象模型,并使用与之相应的时序逻辑公式定义其性质,从而进行验证工作。

如图 2 所示,为了对指挥控制系统进行有效的验证,一方面,使用 LOTOS 对指挥控制系统进行严格的形式化描述,并将其转化为一有限状态机表示的标号转移系统 LTS。其中,LOTOS 中每一个进程对应于一个 C^2O 对象,而定义在其端口上的动作对应于一个转移事件。另一方面,提炼出所要检验的指挥控制系统的性质,并使用 ACTL 对其进行严格的逻辑定义。这样,利用基于时序逻辑的模型检查工具,就可以对指挥控制系统进行严格而有效的验证。

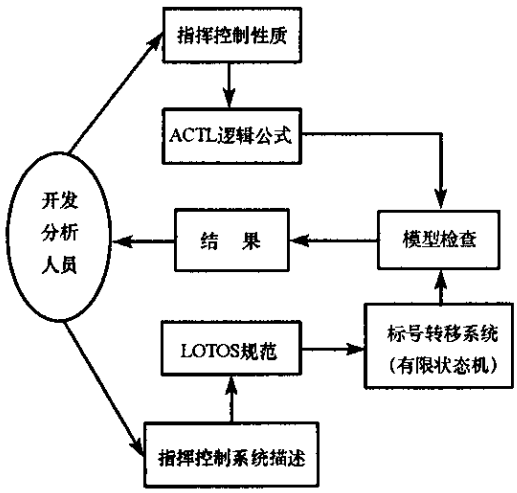

图 2 性质验证的工作原理

Fig.2 The principle of property verification

3.2 一个旅野战高炮防空指挥控制系统的验证

以一个旅野战高炮防空系统为例。验证步骤具体如下:

步骤 1 建立系统的 C^2O 模型。

由系统的需求或者从 IDEF0 描述中(参见文献[2] 的图 3),使用面向对象分析方法,将系统分解为由多个指挥控制对象 C^2O 组成的对象系统,如图 3 所示。

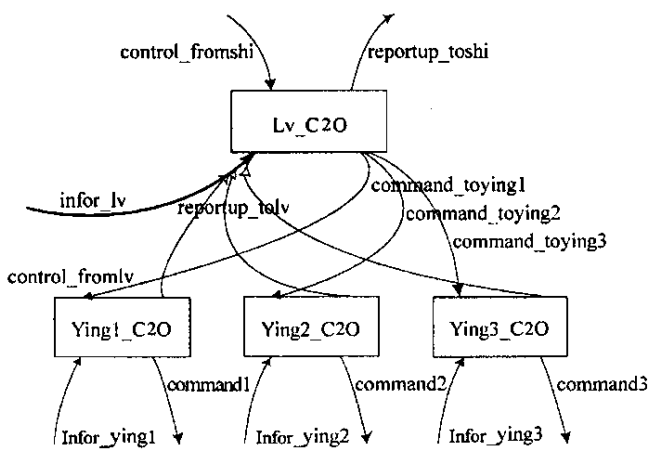

图 3 基于 C^2O 的旅防空指挥控制系统模型

Fig.3 A sample of C^2O _ based model

步骤 2 使用 LOTOS 对系统模型进行形式化描述。

需要使用 LOTOS 对每一个 C^2O 对象进行严格的形式化描述,通常这些 C^2O 对应于 LOTOS 中的进程,其交互动作可表示为事件端口。例如,旅防空指挥控制对象 Lv _ C2O 的 LOTOS 描述为:

process Lv _ C2O [infor _ lv, command _ toying, control _ fromshi, reportup _ toshi]:

(infor _ data: InforData, control _ data: CtrlData): noexit: =

```
(
infor _ lv ?id : InforData ; reportup _ toshi ! Passup(id) ;
C2O[infor _ lv, command _ toying, control _ fromshi, reportup _ toshi](id, control _ data)
[] control _ shi ?cd : CtrlData ; command _ ying ! Passdown(cd) ;
C2O[information _ lv, command _ toying, control _ fromshi, reportup _ toshi]
                                    (infor _ data, add(cd, control _ data))
)
endproc
```

步骤3 使用ACTL定义指挥控制的性质。

例如可以如下定义系统的可达性：

AG ([ying _ infor _ action] E [true {true} U {v _ Cmand&Ctrl _ function} true]).

AG ([lv _ ctrl _ action] E [true {true} U {ying _ Cmand&Ctrl _ function} true]).

步骤4 采用基于LOTOS的时序逻辑验证工具对系统模型进行验证。

这里，选用基于LOTOS的集成工具环境MiniLite[7]对该例的C^2O模型进行性质验证。这一工具的工作原理如图2所示。它将所输入的系统模型的LOTOS描述自动转化为与之对应的标号转移系统，然后在其上进行推理，检验所输入的ACTL时序逻辑公式是否成立。当公式为真时，表明模型满足该公式所表达的系统的某一性质。

步骤5 根据错误情况修改系统的C^2O模型，重复上述步骤，直至它满足所有性质。

4 结 语

本文提出了一个指挥控制对象的概念模型，同时，利用形式化描述语言LOTOS对其进行描述，并使用基于动作的时序逻辑ACTL定义了它的一些基本性质。基于这一方法，尝试对一个旅野战高炮防空指挥控制系统进行了模型验证，也对其中采用的指挥控制对象进行了基本性质验证，实践表明这一方法是可行的。这为进一步验证其他性质提供了指导，也为C^4ISR系统需求工程中的需求描述和需求验证工作提供了新的方法。进一步对C^2O及其应用模型的更细致和深入的工作正在进行中，比如C^2O模型的完善、IDEF0模型向基于C^2O模型的转换、更复杂而深入的指挥控制性质的定义和验证等。

参 考 文 献：

[1] 罗雪山，等. C^3I系统建模方法与技术[M]. 长沙：国防科技大学出版社，2000.

[2] 罗雪山，等. IDEF0方法在军事综合电子信息系统分析设计中的应用[J]. 国防科技大学学报，2001，23(3)：88 – 92.

[3] Arias J J P, Duque J G. SCTL-MUS：A Formal Methodology for Software Development of Distributed Systems. A Case Study[J]. Formal Aspects of Computing, 2001(13)：50 – 91.

[4] 周之英. 现代软件工程(中)[M]. 北京：科学出版社，2000.

[5] Turner K J. Using Formal Description Techniques. An Introduction to Estelle LOTOS and SDL[M]. Wihey, New York, 1993.

[6] DeNicola R, Fantechi A, Gnesi S. An Action Based Framework for Verifying Logical and Behavioural Properties of Concurrent Systems[J]. Computer Networks and ISDN Systems, 1993, 25(7)：761 – 778.

[7] Bolognesi T, Lagemaat J, Vissers C. LOTOShere Software Development with LOTOS[R]. Boston, London：Kluwer Academic Publishers, 1995：1 – 43.

[8] Gotzhein R. Temporal Logic and Its Applications[J]. Computer Networks and ISDN System, 1992, 24：203 – 218.

[9] Hinchey M G, Bowen J P. Applications of Formal Methods[M]. Prentice-Hall, Engewood Cliffs. NJ., 1995.