

# 面向对象的指挥控制流程建模与应用<sup>①</sup>



李光明<sup>1</sup>, 修保新<sup>2</sup>, 刘康<sup>1</sup>

<sup>1</sup>(中国人民解放军 78118 部队, 成都 610051)

<sup>2</sup>(中山大学 系统科学与工程学院, 广州 510006)

通信作者: 修保新, E-mail: xiubx@mail.sysu.edu.cn

**摘要:** 流程建模是分析指挥控制流程, 取得作战胜利的关键步骤. 当前的建模方法不能很好地应对联合作战中指挥控制流程多域联合、松耦合关联等特征. 为了解决上述问题, 本文提出一种面向对象的流程建模语言, 构建的模型描述了指挥控制流程的数据视角、行为视角以及二者间的复杂交互关系, 并基于该模型采用流程挖掘技术进行指挥控制流程还原, 为优化指挥控制流程、提高流程效能提供了一种有效手段.

**关键词:** 指挥控制; 流程建模; 面向对象; 流程挖掘; 多实例

引用格式: 李光明, 修保新, 刘康. 面向对象的指挥控制流程建模与应用. 计算机系统应用, 2023, 32(10): 255-264. <http://www.c-s-a.org.cn/1003-3254/9096.html>

## Object-centric Command and Control Process Modeling and Its Application

LI Guang-Ming<sup>1</sup>, XIU Bao-Xin<sup>2</sup>, LIU Kang<sup>1</sup>

<sup>1</sup>(PLA No. 78118, Chengdu 610051, China)

<sup>2</sup>(School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China)

**Abstract:** Process modeling is the key step to analyzing the command and control process and winning the battle. The current modeling methods cannot well deal with the characteristics of multi-domain association and loose-coupled correlation of command and control processes in the joint battle. In view of the above problems, this study proposes an object-oriented process modeling language. The built model describes the data perspective, behavior perspective, and the complex interaction between them in the command and control process. Based on this model, process mining technology is applied to restore the command and control process, which provides an effective means to optimize the command and control process and improve process efficiency.

**Key words:** command and control; process modeling; object-centric; process mining; multiple cases

指挥控制流程描述了指挥控制命令在行动空间中的流动过程. 它是战争的关键环节, 决定了军队作战的效能, 影响着战争的胜负结局. 联合作战中的指挥控制流程具有多域联合、松耦合关联、面向对象等特点. 研究指挥控制流程的上述特点, 分析优化指挥控制流程是取得联合作战胜利的关键.

模型经常被用来对指挥控制流程进行可视化, 它在描述和理解指挥控制流程上起着重要作用. 流程建模是分析流程进而提升流程效能的一种有效手段, 目

前存在大量的业务过程建模语言, 它们也可以引入到军事领域, 对指挥控制流程进行建模.

由于指挥控制流程本身的复杂性, 其任务类型的多样性、实体和关系的复杂性难以建立一种普适性的流程模型并适用于所有场合. 为了能反映指挥控制流程活动组织的内在逻辑与运作机理, 决策者需要基于具体场景和实际情况, 选择适合的语言对其进行建模.

BPMN、Petri 网、Declare 等是当前广泛采用的流程建模语言. 指挥控制流程的多域联合特点使得流程

① 收稿时间: 2022-09-28; 修改时间: 2022-10-27; 采用时间: 2022-12-23; csa 在线出版时间: 2023-06-30

CNKI 网络首发时间: 2023-07-03

中存在多种实体的实例,松耦合关联特点使得实体之间的交互关系非常灵活.上述的过程建模语言一次只能描述基于一种案例概念的流程实例的生命周期,不能很好地描述实体的多实例及其之间灵活的交互关系.这些语言常常迫使分析人员或建模人员将面向对象的流程中多维度、立体的交互强压到扁平或者分离的模型中去,导致流程的基本特性不能够被捕获并呈现出来.在使用业务流程语言描述指挥控制流程时,建模语言与实际流程之间的不匹配以及产生的问题便凸显

出来.

为了解决上述问题,本文提出一种面向对象的流程建模语言来描述联合作战中的指挥控制流程,它结合了声明式、基于约束的语言(如 Declare)和数据/对象建模技术(如 UML)的思想,提出指挥控制流程的混合模型构建方法,同时表示流程的控制流(即行为)方面和数据方面,建立混合模型约束统一机制描述模型中控制流方面和数据方面以及它们交互中的约束,如图 1 所示.

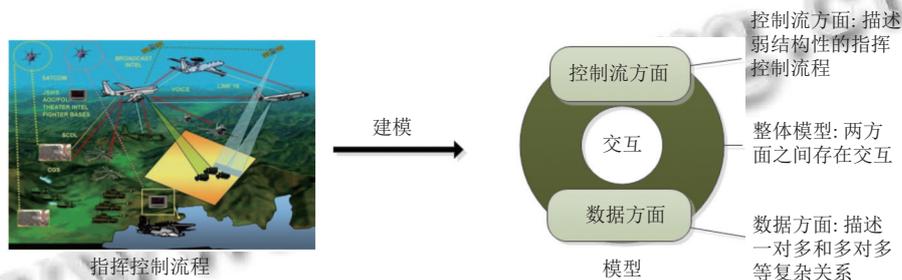


图 1 面向对象的指挥控制流程建模思路

由面向对象的流程建模语言生成的模型支持进一步的应用,即以该模型作为输出,采用流程挖掘技术对指挥控制流程进行还原,通过模型展示了指挥控制流程的真实运行过程,为优化指挥控制流程、寻找流程瓶颈、提高流程效能提供了一种有效手段.

## 1 指挥控制流程

随着军事技术和装备的发展,军事对抗领域呈现出“从单域到多域、从平面到立体、从有形到无形”的拓展变化规律.物理域是我们所熟悉的传统作战领域,它覆盖陆、海、空、天、网、电等多维空间.多域联合的特征导致指挥控制流程由多个模块组成,不存在一个统一的全局案例概念.因为每个模块有它自己的案例概念,贯穿与它相关的子流程的整个生命周期.系统支撑的指挥控制流程跨越多个子模块,因此整个流程有多个案例概念.此外,指挥控制流程中存在一对多和多对多关系.例如,由指挥控制系统支持的防空反导过程涉及多个模块,例如预警、决策、拦截等.在此过程中,一次预警发现的目标可能被分成多个批次拦截,一次拦截也可能包含连续几次发现的目标.因此,在预警和拦截之间存在多对多关系<sup>[1]</sup>.

在信息时代的联合作战中,指挥控制流程需要多个参与方的协作,尤其是有不同观点、目的、任务的

参与方.其中的关键是跨职能领域、单元和组织的一体化集成.指挥控制流程的目的不是精确控制战争,而是对战场施加一种松散形式的影响.一个典型案例是模块化指挥中心的概念,这些模块之间是松耦合的关系.为了应对动态的战争环境,战争的终端参与者被赋予更大的自主权力,具有更大的敏捷性,导致指挥控制流程具有更大的柔韧性和灵活性.在当前军种主建,战区主战的模式下,各个军种部队进行相对独立的建设,每个部队可以抽象为能够提供某种特定能力的模块,在战争或演习期间,由战区统一指挥,以一种弱关联的方式融合在一起.松耦合关联的特征导致在指挥控制流程中,提取的事件只能形成弱事件视角,需要采用描述性、开放性的语言对事件视角进行建模,且对行为的分析需要其他视角(例如数据/对象视角)的支撑.

面向对象的思想是人类认知世界、抽象物体的一种主流思想.在联合作战中,也可以采用面向对象的思想,即通过划分作战域(陆地、海洋、空中)、划分不同作战阶段、划分不同专业功能等方式,使得参与到指挥控制流程的元素抽象成易于控制管理的组织、团体或活动节点,统称为对象.基于面向对象的思想,可以将具有多域联合和松耦合关联特征的指挥控制流程看做是不同类型的对象及其之间宽松行为约束的结合.对象类型确定了指挥控制流程的边界,对象行为决定

了指挥控制流程的功能. 综合分析, 指挥控制流程主要包括以下 4 方面的内容: (1) 对象, 即流程中的物理或抽象元素 (节点、实体等); (2) 对象结构, 即元素之间的组织结构关系; (3) 功能, 即元素具有的能力行为; (4) 功能运行关系, 即行为之间的约束关系. 在联合作战的指挥控制流程中, 对象是指挥控制流程的核心, 我们称之为面向对象 (或以对象为中心) 的指挥控制流程.

## 2 研究现状

目前存在大量的过程建模语言, 他们通过不同的视角, 侧重描述流程的不同方面, 通常包括控制流 (即行为方面)、数据方面 (即数据结构方面)、资源 (即活动执行者)、时间、成本等. 传统的传统业务流程建模语言主要包括 BPMN、Petri 网、EPCs、ER、UML、Declare 等, 本节将对其进行讨论.

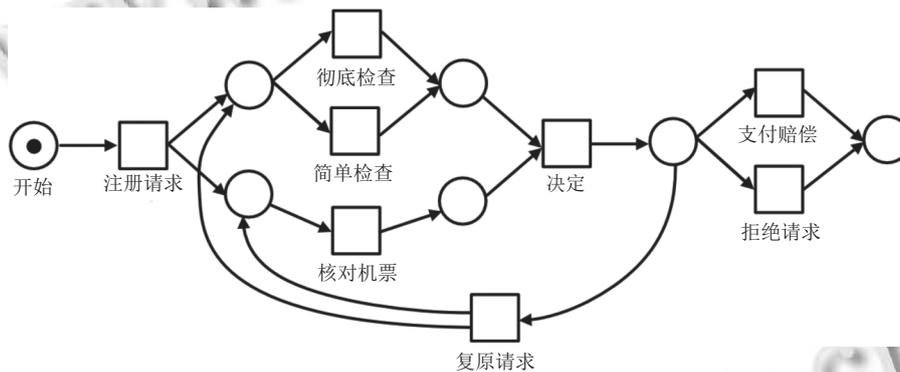


图2 航空公司处理赔偿请求的流程

Chen 在文献 [7] 中提出 ER (entity-relationship) 图, 使其迅速成为一种被广泛采用的数据建模语言. 它常被用来描述数据库系统的数据结构, 即数据中涉及的实体之间的关系. UML (unified modeling language) 类图<sup>[8]</sup>跟 ER 图很相似, 它被 OMG (object management group) 开发并作为一个标准, 对软件系统中不同模块进行描述.

此外, 文献 [9,10] 提出了一种声明式语言 Declare, 被用来对灵活的工作流程进行建模. 它创建了一些列的模型元素符号, 并基于线性时序逻辑 (linear temporal logic) 定义了符号的语义. 本质上, 一个 Declare 模型可以看做一组线性时序逻辑规则, 每一个规则描述了两个活动之间的流程依赖关系. 换句话说, 一个模型包含的所有规则必须被这个模型所描述的流程在执行过程中所遵守.

BPMN (business process modeling notation)<sup>[2-4]</sup> 是在商业领域应用最普遍的流程建模语言, 它提供了大量的图像元素来支持对流程进行可视化, 被很多公司采用 (例如 Activiti、Bonita BPM、Camunda、IBM Rational System Architect 和 Signavio). Petri 网<sup>[5]</sup> 是在学术领域尤其是过程挖掘领域的主流建模语言. 图 2 用 Petri 网描述了航空公司处理顾客因为航班延误或者取消, 提出的赔偿请求的流程. EPCs (event-driven process chains)<sup>[6]</sup> 提供了经典的模型元素来支持流程建模, 它被 ARIS 和 SAP R/3 等公司采用. 上述语言经常被用来对流程的控制流方面进行建模, 虽然它们通过添加新元素符号等方式 (例如 BPMN 2.0 在 BPMN 1.0 的基础引入“数据对象”等元素) 来增加对数据方面的建模能力, 但在描述数据中涉及的实体间的复杂交互关系方面能力依然很弱.

面向对象的指挥控制流程由多个交互的实体组成, 涉及多个模块, 多个实体和多个子流程 (每个模块或实体有一个对应的子流程), 他们之间的交互存在一对多和多对多等复杂的对应关系, 不同实体采用不同的案例概念, 导致整个指挥控制流程缺少一个全局统一的案例概念. 上述的过程建模语言一次只能描述基于一种案例概念的流程实例的生命周期, 且不能很好地应对实体之间复杂的交互关系和多实例之间的交互关系. 这些语言常常迫使分析人员或建模人员将面向对象的流程中多维度、立体的交互强压到扁平或者分离的模型中去, 导致流程的基本特性不能够被捕获并呈现出来. 在使用业务流程语言描述面向对象的指控流程, 建模语言与实际流程之间的不匹配以及产生的问题便凸显出来. 总的来说, 传统业务流程建模语言往往会遇到以下问题.

(1) 很难为整个过程确定一个案例概念. 由于业务流程中实体之间存在一对多和多对多的关系, 因此缺少一个全局唯一的案例概念. 例如, 在“侦查”和“判断(指挥机构)”部门之间存在多对多关系. 在第1个部门眼中, 案例概念是“侦查任务”, 而在第2个部门中, 案例概念是“命令”. 因此, 很难为这两个部门找到一个通用的案例概念.

(2) 很难对流程实例之间的交互建模. 主流的业务流程建模语言一次只能描述一种类型的流程实例的生命周期, 这样实例之间的交互就会丢失. 以 BPMN 为例, 尽管 BPMN 使用通道、泳池和消息流等概念来解决这个问题, 但仍然在每个(子)流程中对单个实例进行独立建模.

(3) 数据视角被轻视. 可以对数据元素进行建模, 但是 ER 模型和 UML 类模型中采用的更强大的描述符号不能在传统的过程模型中很好地表达.

(4) 很难把对端到端流程的描述统一到一张集成图中. 由于存在多个案例概念, 整个过程通常分布在多个图上. 此外, 数据视角和行为(控制流)视角的交互在当前的过程模型中没有得到很好的呈现. 例如, 数据模型中的基数约束会影响行为, 但这在 BPMN 图和数据感知 Petri 网等模型中没有反映出来.

(5) 不同视角没有采用统一的约束方式. 不同视角的约束会相互影响, 例如, 数据模型中的基数约束会影响行为视角的约束. 因此, 需要一种统一的约束方式来准确描述不同视角之间的相互作用.

### 3 面向对象的建模方法

指挥控制流程从时间维度上将指挥控制活动划分为不同的环节, 每个环节对应不同的使命任务, 各环节之间存在着因果、前提、逻辑、业务等关联. 指挥控制流程建模尝试针对指挥控制流程建立合理的概念模型以分析其运作机制和指导其优化设计. 为了解决当前的建模方法在描述联合作战指挥控制流程中遇到的问题, 我们提出了面向对象的建模方法.

#### 3.1 整体思路

在复杂战场环境下的联合作战背景下, 战场情况瞬息万变. 为了应对动态的战争环境, 战争的终端参与者被赋予更大的自主权力, 具有更大的敏捷性, 导致指挥控制流程具有更大的柔韧性和灵活性, 指挥控制流程具有多域联合、松耦合关联、面向对象等特征, 可

以总结为面向对象的流程, 使得当前的大多数建模语言都面临着一些困难. 更准确地说, 当前的过程建模语言常常用于描述稳定环境中标准化、结构化的流程. 因为可预测性和低复杂性, 这些流程被用一种“闭合世界”的方式来描述, 显式地描述所有允许的行为, 不能很好地描述活动之间更加灵活、松耦合的约束关系<sup>[11]</sup>.

为了解决这些问题, 我们提出了一种新的建模语言, 即以对象为中心的行为约束(OCBC)建模语言. 它结合了声明式、基于约束的语言(如 Declare)和数据/对象建模技术(如 UML)的思想, 提出指挥控制流程的混合模型构建方法, 同时表示流程的控制流方面和数据方面, 建立混合模型约束统一机制描述模型中控制流方面和数据方面以及它们交互中的约束<sup>[12,13]</sup>.

由这种语言构建的 OCBC 模型能够描述流程的数据视角和行为视角, 以及两者之间的交互作用. 基数约束被用作一种统一的机制来处理数据和行为依赖, 以及它们之间的相互作用. 从本质上讲, OCBC 模型是在类模型上添加了行为视角演化而来, 类模型(引用 UML 类模型)是 OCBC 模型的支撑, 用于描述业务流程的数据视角. 因为类模型可以很容易地处理一对多和多对多关系, 基于这种能力创建的流程模型, 能够描述不同类型实例之间的复杂交互(例如多对多关系). 经典的多实例问题也可以通过使用数据模型进行事件关联的处理. 添加的行为视角(引用 Declare 模型)具有声明式建模语言的特性, 支持采用数据模型中的基数约束来描述活动间的行为依赖. 总之, OCBC 模型能够描述包含交互实例和复杂数据依赖关系的业务过程.

#### 3.2 数据视角建模方法

由于指挥控制流程具有面向对象、即“以数据对象为中心”的特征, 我们首先描述数据对象(即流程中的实体)视角, 即对数据视角进行建模, 并把数据视角作为整个指挥控制流程的骨架.

数据约束用于在数据方面描述实体之间的关系. 在数据建模语言中, 基数通常用来形成数据约束, 例如, ER 模型和 UML 类模型就采用了基数约束. 基数是一组整数, 例如, “1..\*”表示所有正整数的集合. 两个实体之间的关系所指示的约束可以由两个基数指定. 例如, 考虑实体“拦截命令”和“拦截行动”之间的关系. 每个命令应该至少对应一次行动(“1..\*”), 每次行动应该恰好有一个对应的命令(“1”).

基数在某些情况下不足以精确地描述约束. 例如,

在图3中,每个拦截命令都应该在某个时间点(例如在命令下达后的5 min之内)被执行,即产生对应的拦截行动.在这种情况下,对于命令来说,不需要“总是”有一个相应的行动(即在前5 min之内没有相应的行动),但它应该“最终”(即在5 min之后)有一个相应的行动.显然,边 $r_1$ 的基数无法精确描述这种情况.因此,我们给基数添加了两种类型的符号( $\square$ 和 $\diamond$ ),如边 $r_2$ 所示,以加强它们的表达能力.

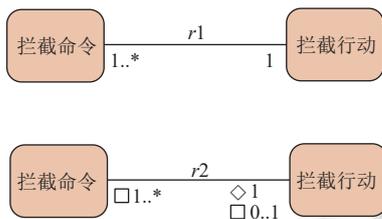


图3 使用 $\square$ 和 $\diamond$ 符号加强基数表达能力

更准确地说,具有正方形符号 $\square$ 的基数称为“总是”基数,而具有菱形符号 $\diamond$ 的基数称为“最终”基数。“总是”基数要求基数总是得到满足。“总是”基数可以和“最终”基数组合在一起,表示两种基数约束同时被满足.例如,一个组合基数“ $\square 0..1 \diamond 1$ ”可以表示一个拦截命令在某一时刻之后要有一个对应的拦截行动(“ $\diamond 1$ ”),且在任何时刻最多只能有一个对应的拦截行动(“ $\square 0..1$ ”).

定义1. 类模型. 令为 $U_{Card}$ 所有基数的集合. 符号 $\leq$ 和 $\diamond$ 附加在基数之前表示基数的时间属性. 一个类模型是一个元组 $ClM = (C, R, \pi_1, \pi_2, \#_{src}^{\square}, \#_{src}^{\diamond}, \#_{tar}^{\square}, \#_{tar}^{\diamond})$ , 其中:

- $C \in U_C$ 是类的集合.
- $R \in U_R$ 是关系的集合.
- $\pi_1 \in R \rightarrow C$ 指定一个关系的源类.
- $\pi_2 \in R \rightarrow C$ 指定一个关系的靶类.
- $\#_{src}^{\square} \in R \rightarrow U_{Card}$ 指定一个关系的来源约束(约束应该在任何时间点都被满足,由符号 $\leq$ 表示).
- $\#_{src}^{\diamond} \in R \rightarrow U_{Card}$ 指定一个关系的来源约束(约束应该在某个时间点之后都被满足,由符号 $\diamond$ 表示).
- $\#_{tar}^{\square} \in R \rightarrow U_{Card}$ 指定一个关系的的目标约束(约束应该在任何时间点都被满足,由符号 $\square$ 表示).
- $\#_{tar}^{\diamond} \in R \rightarrow U_{Card}$ 指定一个关系的的目标约束(约束应该在某个时间点之后都被满足,由符号 $\diamond$ 表示).

数据约束添加了“总是”( $\square$ )或者“最终”( $\diamond$ )类型标记. 基于数据约束及其符号,我们定义了一个类模型

来描述业务流程的数据视角.

图4展示了一个类模型 $ClM = (C, R, \pi_1, \pi_2, \#_{src}^{\square}, \#_{src}^{\diamond}, \#_{tar}^{\square}, \#_{tar}^{\diamond})$ . 它有5个对象类 $C = \{\text{指挥机构, 预警任务, 目标, 拦截命令, 拦截行动}\}$ 和5个关系 $R = \{r_1, r_2, r_3, r_4, r_5\}$ . 函数 $\pi_1$ 和 $\pi_2$ 分别指定关系的源类和靶类. 例如, $\pi_1(r_1) = \text{“指挥机构”}$ ,  $\pi_2(r_1) = \text{“拦截命令”}$ , 表示关系 $r_1$ 连接源类“指挥机构”和靶类“拦截命令”.

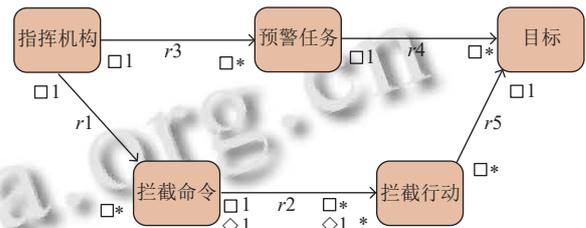


图4 类模型示例

组合基数用来描述类关系约束. 函数 $\#_{src}^{\square}$ 、 $\#_{src}^{\diamond}$ 、 $\#_{tar}^{\square}$ 和 $\#_{tar}^{\diamond}$ 用来指定这些基数. 例如, $\#_{src}^{\square}(r_2) = \{1\}$ 表示类关系 $r_2$ 源端(源类“拦截命令”一侧)的“总是”基数为 $\{1\}$ ,即要求每一个“拦截行动”对象总是对应一个“拦截命令”对象. $\#_{tar}^{\square}(r_2) = *$ 表示类关系 $r_2$ 靶端(靶类“拦截行动”一侧)的“总是”基数为 $\{0, 1, 2, \dots\}$ ,即要求每一个“拦截命令”对象总是对应任意个“拦截行动”对象. $\#_{src}^{\diamond}(r_2) = \{1\}$ 表示类关系 $r_2$ 源端的“最终”基数为 $\{1\}$ ,即要求每一个“拦截行动”对象最终对应一个“拦截命令”对象. $\#_{tar}^{\diamond}(r_2) = 1..*$ 表示类关系 $r_2$ 靶端的“最终”基数为 $\{1, 2, \dots\}$ ,即要求每一个“拦截命令”对象最终对应至少一个“拦截行动”对象.

根据定义,每个类关系都包含4个基数.为了简洁,我们在图中可以省略冗余或不关心的基数.

### 3.3 行为视角建模方法

在指挥控制流程中,行为活动之间的约束是非常松散和非结构化的,这使得指挥者能够灵活地处理动态的突发情况.例如,在发现敌方目标时,可以进行火炮拦截,也可以进行导弹拦截.与传统的业务流程相比,指挥控制流程提供了一个“开放”的环境,可以根据需求更自由地进行操作.

从行为视角更深层次地来看,指挥控制流程仅是事件的集合,事件之间的发生顺序满足流程中的规则,没有显式的案例或流程实例概念.指挥控制流程可以被看作由活动集和活动之间的约束集构成,很难形式化为一个清晰、结构化、端到端的流程模型.约束可

能是对活动在时序方面的限制,例如,拦截行动只能在拦截命令下达之后发生.约束也可能与基数相关,例如,每个拦截命令之后必须有至少一次拦截行动.

在本文中,我们使用声明式语言对行为约束进行建模,因为声明式语言是“开放”的语言,它允许没有被约束明确禁止的任何行为.声明式语言更灵活,更适合描述指挥控制流程,并根据 Declare 语言的启发,采用图形符号来描述约束.具体来说,一个声明式约束由一个参考活动、一个目标活动和一个约束类型组成,约束类型规定了参考活动和目标活动实例(即事件)之间的限制.例如,给定一个参考事件(参考活动的实例),约束类型要求在参考事件之前和之后的目标事件的个数在某个特定范围之内.基于这种思想,约束类型被定义为整数对的一个集合.

定义 2. 约束类型.  $U_{CT} = \{X \subseteq \mathbb{N} \times \mathbb{N} \mid X \neq \emptyset\}$  定义了

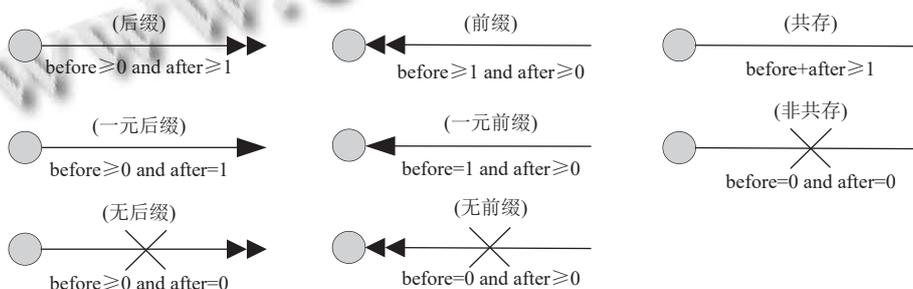


图 5 约束类型的图形化表示

定义 3. 活动模型. 一个活动模型是一个元组  $ActM = (A, Con, \pi_{ref}, \pi_{tar}, type)$ , 其中:

- $A \subseteq U_A$  是一个活动的集合(由矩形表示).
- $Con \subseteq U_{Con}$  是一个约束的集合 ( $A \cap Con = \emptyset$ , 由各种类型的边表示).
- $\pi_{ref} \in Con \rightarrow A$  定义约束的参考活动(用连接约束和活动的黑点表示).
- $\pi_{tar} \in Con \rightarrow A$  定义约束的目标活动(边的另一侧).
- $type \in Con \rightarrow U_{CT}$  指定每个约束的类型(由边的类型表示).

一个活动模型  $ActM = (A, Con, \pi_{ref}, \pi_{tar}, type)$  包含

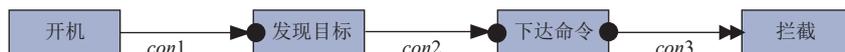


图 6 一个活动模型的示例

### 3.4 数据和行为视角交互建模方法

数据约束关联了类模型中的类,行为约束关联了

包含所有约束类型的集合.  $U_{CT}$  中的一个元素表示一个非空整数对集合,整数对中第 1 个整数表示参考事件之前的目标事件个数,第 2 个整数表示参考事件之后的目标事件个数.

定义 2 给出了约束类型的一般概念,除了图 5 展示了约束类型的 8 个示例外,还可以通过规定参考事件前后的目标事件个数来定义任何约束类型.由于声明式约束用于描述灵活业务流程中的松散规则,定义 2 中的“之前”和“之后”并不严格要求目标事件紧接着在参考事件之前或之后发生.换句话说,声明式约束有一个影响范围(例如一组事件),它要求在这个范围内,目标事件在参考事件前后的数量满足约束.在传统的流程建模语言中,范围对应于一个案例(流程实例).由于在我们的方法中没有假设案例概念,OCBC 模型中声明式约束的范围是根据数据视角确定的.

活动集  $A$  和约束集  $Con$ . 每个约束对应一个参考活动、一个目标活动和一个约束类型,可以分别通过函数  $\pi_{ref}$ 、 $\pi_{tar}$  和  $type$  获取.例如,图 6 展示了一个活动模型,它包括 4 个活动(即  $A = \{\text{开机, 发现目标, 下达命令, 拦截}\}$ )和 3 个约束(即  $Con = \{\text{con1, con2, con3}\}$ ).下面以  $con1$  为例来解释这些函数.  $\pi_{ref}(con1) = \text{“发现目标”}$ , 表明“发现目标”是参考活动;  $\pi_{tar}(con1) = \text{“开机”}$ , 表明“开机”是目标活动;  $type(con1) = \{(before, after) \in \mathbb{N} \times \mathbb{N} \mid before = 1\}$ , 表明约束类型为“一元前缀”(通过指向黑点的单箭头可知).约束  $con1$  的含义为在每个“发现目标”事件之前必须有一个“开机”事件,即只有预警设备开机后,才能够发现目标.

活动模型中的活动,这两种约束作为桥梁,把每个单视角(模型)的元素关联起来.基于同样的思路,为了将

数据视角和行为视角相结合,我们在活动和类之间定义了一种新的约束。

定义 4. *AOC* 关系. 令  $A \in U_A$  是一个活动集合,  $C \in U_C$  是一个类集合.  $AOC \subseteq A \times C$  是关联活动和类的 *AOC* 关系的集合. 为了方便,我们定义 3 个函数来指定关系上的基数.

•  $\#_A^{\square} \in AOC \rightarrow U_{Card}$  指定 *AOC* 关系来源端 (活动一侧) “总是”基数.

•  $\#_A^{\diamond} \in AOC \rightarrow U_{Card}$  指定 *AOC* 关系来源端 (活动一侧) “最终”基数.

•  $\#_{OC} \in AOC \rightarrow U_{Card}$  指定 *AOC* 关系目标端 (类一侧) 基数.

活动和类之间的约束关系在定义 4 中被定义为 *AOC* 关系,以区别于其他类型的约束关系.此外,还定义了 3 个函数来获取 *AOC* 关系中的 3 个基数.

定义 5. 面向对象的流程 (OCBC) 模型. 一个面向对象的流程模型是一个元组  $OCBCM=(Clam, ActM, AOC, \#_A^{\square}, \#_A^{\diamond}, \#_{OC}, crel)$ , 其中:

•  $Clam=(C, R, \pi_1, \pi_2, \#_{src}^{\square}, \#_{src}^{\diamond}, \#_{tar}^{\square}, \#_{tar}^{\diamond})$  是一个类模型.

•  $ActM=(A, Con, \pi_{ref}, \pi_{tar}, type)$  是一个活动模型.

•  $C, R, A$  和  $Con$  两两互不相交 (没有名字冲突).

•  $AOC \subseteq A \times C$  是一个 *AOC* 关系集合, 函数  $\#_A^{\square}$ 、 $\#_A^{\diamond}$  和  $\#_{OC}$  指定 *AOC* 关系的 3 个约束.

•  $crel \in Con \rightarrow C \cup R$  指定每一个行为约束的事件关联模式, 且每一个  $con \in Con$  满足以下条件:

a) 如果  $crel(con)=c \in C, \{(\pi_{ref}(con), c), (\pi_{tar}(con), c)\} \subseteq AOC$ .

b) 如果  $crel(con)=r \in R, \{(\pi_{ref}(con), \pi_1(r)), (\pi_{tar}(con), \pi_2(r))\} \subseteq AOC$  或者  $\{(\pi_{ref}(con), \pi_2(r)), (\pi_{tar}(con), \pi_1(r))\} \subseteq AOC$ .

一个面向对象的流程模型  $OCBCM=(Clam, ActM, AOC, \#_A^{\square}, \#_A^{\diamond}, \#_{OC}, crel)$  包括一个类模型 *Clam* (描述对象/数据) 和一个活动模型 *ActM* (描述行为). 这两个模型通过 *AOC* 关系、 $\#_A^{\square}$ 、 $\#_A^{\diamond}$ 、 $\#_{OC}$  和 *crel* 函数相互关联. 我们在图 7 中构建一个 OCBC 模型示例来描述防空反导指挥控制流程中的一个场景.

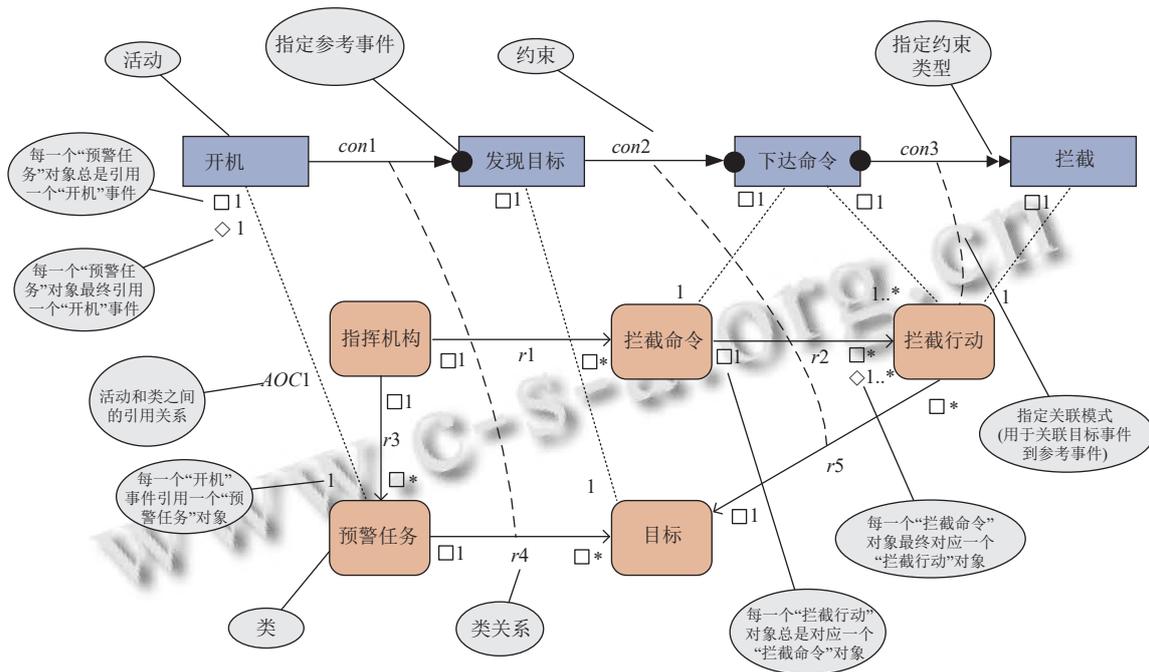


图 7 通过一个 OCBC 示例模型解释模型中的主要元素

在图 7 中, 类模型 *Clam* 包括 5 个类 (即  $C=\{\text{指挥机构, 预警任务, 目标, 拦截命令, 拦截行动}\}$ ) 和 5 个类关系 (即  $R=\{r1, r2, r3, r4, r5\}$ ).  $r2$  表明了“拦截命令”和“拦截行动”之间一对多的关系, 即每个“拦截命令”最终都有至少一个对应的“拦截行动”, 并且每个“拦截行

动”总是对应一个“拦截命令”. 活动模型 *ActM* 包括 4 个活动 (即  $A=\{\text{开机, 发现目标, 下达命令, 拦截}\}$ ). 行为约束 *con1* 表示每个“发现目标”事件之前应该有一个“开机”事件, *con3* 表示每个“下达命令”事件后至少应该跟随一个“拦截”事件.

AOC关系在模型层面将活动和类关联起来,在实例层面规定了事件和对象之间的对应关系.例如,图7中的  $AOC = \{(\text{开机}, \text{预警任务}), (\text{发现目标}, \text{目标}), (\text{下达命令}, \text{拦截命令}), (\text{下达命令}, \text{拦截行动}), (\text{拦截}, \text{拦截行动})\}$ . 通过AOC关系可知,“下达命令”事件可以对应“拦截命令”和“拦截行动”对象,但不能对应“目标”对象,因为  $(\text{下达命令}, \text{目标}) \notin AOC$ . 一个活动可以对应多个类,例如“下达命令”同时对应“拦截命令”和“拦截行动”;一个类也可以被多个活动对应,例如“拦截行动”同时被“下达命令”和“拦截”对应.

#### 4 基于流程挖掘技术的模型应用

流程挖掘技术是最近兴起的一门跨学科技术,它通过分析流程执行过程中记录的事件日志信息,从中挖掘出一个流程模型,来反映流程执行的真实情况<sup>[14-16]</sup>. 本文提出的面向对象的建模语言设计的OCBC模型支持流程挖掘技术,可以基于流程挖掘技术从事件数据中挖掘出OCBC模型来反映指挥控制流程.

##### 4.1 指挥控制流程挖掘

随着信息战时代的来历,通过网络探针截取信息等方式可以获得大量关于敌方作战体系的事件数据,这些数据中包含着丰富的能反映敌方指挥控制流程的

有用信息. 当前普遍采用人工分析的方法来推测敌方指挥控制流程. 这种方法带有强烈的主观性,不能客观准确、客观地获取真实运行的流程. 此外,因为数据量巨大,使得人工的分析方式效率低下、耗时较长,不能在战场态势不断迅速变化和演进的背景下,快速有效地获取敌方指挥控制流程,为我军决策提供支撑.

指挥控制流程挖掘能够从获取的敌方作战体系的事件大数据中自动挖掘出作战流程,分析不断变化和演进的体系行为的现实,全面还原敌方指挥控制流程,为网络空间对抗和联合作战行动提供决策依据,能大大推进我军作战智能化的发展.

##### 4.2 应用案例

为了验证OCBC模型在指挥控制流程挖掘中的应用,下面基于一个真实的案例场景进行分析. 防空反导发展成为当前武器装备和作战理念发展的新动向,中国的防空反导能力也正在走向世界一流,研究防空反导场景下的指挥控制流程具有重要意义. 火炮指控系统是防空反导中一个重要的场景,具有复杂性、时敏性、高对抗性、不确定性等特点. 如图8所示,主要是在指挥机构的组织下,利用警戒雷达来感知敌方目标,对具有威胁的目标利用拦截武器进行摧毁和驱赶,是指挥控制系统支持的最典型的流程之一.

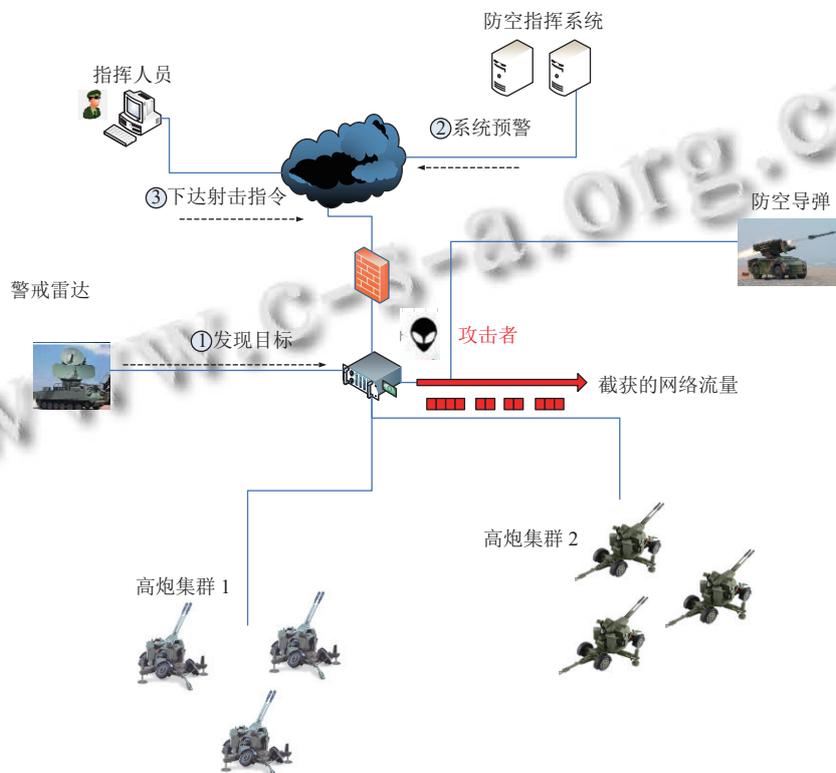


图8 火炮指控系统场景

火炮指控系统中包括多种实体,例如警戒雷达、指挥人员、高炮集群等;每种实体具有某种特定的功能活动,即能够产生特定的行为,例如雷达能够发现目标,指挥人员能够下达射击指令.实体之间有依赖关系,例如指挥人员的判断需要雷达提供预警信息;活动之间有逻辑时序关系,例如指令的下达需要在发现目标之后.在火炮指控系统中,多种实体产生不同的活动,实体之间存在依赖关系,且活动之间存在行为约束下,上述所有元素融合在一起,构成了一个有序的流程.

基于从上述场景生成的事件数据<sup>[17-19]</sup>,采用挖掘算法可以进行模型挖掘<sup>[20]</sup>.利用文献[21]中的工具挖掘得到如图9所示的模型.这个模型反映的是流程真实的运行状况,为深入分析流程、提高流程效率提供了有力支撑.根据模型可知,在雷达开机后才能发现目标;雷达发现目标要在雷达关机之前;雷达发现目标之后防空指挥系统才能发布系统预警;雷达发现目标之后指挥员才能下达射击指令;一个指令可以发给多个高炮集群,导弹拦截和高炮射击必须要在射击指令下达之后.

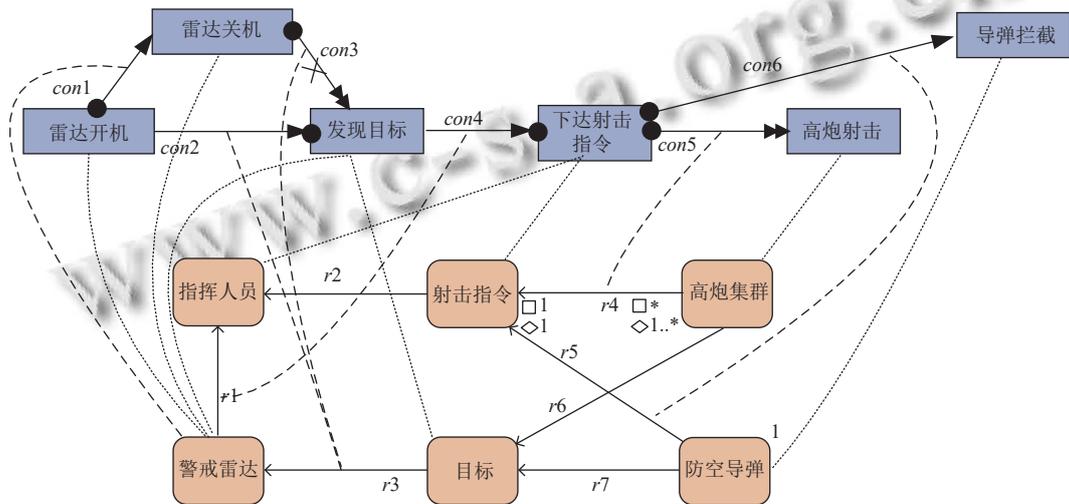


图9 挖掘出的指挥控制流程模型

传统的基于 BPMN、Petri 网的挖掘方法只侧重描述流程的行为视角,无法有效还原流程的数据视角,导致挖掘出的流程与真实的场景之间存在偏差.相比之下,图9中的 OCBC 模型在一个图中描述了数据视角、行为视角以及它们之间的相互作用,清楚地揭示了防空反导场景中所涉及的类、活动和约束.

## 5 总结

为了解决当前建模语言不能很好地应对联合作战中指挥控制流程多域联合、松耦合关联等特点的问题,本文提出一种面向对象的流程建模语言,它结合了基于约束的语言和数据建模技术的思想,建立了混合模型约束统一机制描述指挥控制流程中的复杂交互关系.

采用这种语言生成的 OCBC 模型包括 3 部分,其中类模型表示流程的数据方面,活动模型表示流程的行为方面,AOC 关系表示两个方面之间的交互.OCBC 模型能够描述包含交互实体和复杂数据依赖关系的指

挥控制过程.

OCBC 模型支持流程挖掘技术,本文通过一个基于火炮指控系统的案例对实现了对指挥控制流程还原,验证了模型的有效性,表明了本文提出的建模语言为优化指挥控制流程、提高流程效能提供了一种有效手段.

## 参考文献

- 张维明,朱承,黄松平,等.指挥与控制原理.北京:电子工业出版社,2021.
- The Object Management Group. Business Process Model and Notation (BPMN): Version 2.0. Milford: The Object Management Group, 2010.
- Marin-Castro HM, Tello-Leal E. An end-to-end approach and tool for BPMN process discovery. Expert Systems with Applications, 2021, 174: 114662. [doi: 10.1016/j.eswa.2021.114662]
- Bertoli P, Corcoglioniti F, Di Francescomarino C, et al.

- Semantic modeling and analysis of complex data-aware processes and their executions. *Expert Systems with Applications*, 2022, 198: 116702. [doi: [10.1016/j.eswa.2022.116702](https://doi.org/10.1016/j.eswa.2022.116702)]
- 5 Chenou J, Hsieh G, Williams A. Theoretical foundation of colored Petri net through an analysis of their markings as multi-classification. *arXiv:2203.01194*, 2022.
  - 6 Gottschalk F, van der Aalst WMP, Jansen-Vullers MH. Merging event-driven process chains. *Proceedings of the 2008 OTM Confederated International Conferences on the Move to Meaningful Internet Systems*. Monterrey: Springer, 2008. 418–426.
  - 7 Chen PPS. The entity-relationship model-toward a unified view of data. *ACM Transactions on Database Systems*, 1976, 1(1): 9–36. [doi: [10.1145/320434.320440](https://doi.org/10.1145/320434.320440)]
  - 8 Ahmed S, Ahmed A, Eisty NU. Automatic transformation of natural to unified modeling language: A systematic review. *Proceedings of the 20th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications*. Las Vegas: IEEE, 2022. 112–119.
  - 9 Maggi FM, Di Ciccio C, Di Francescomarino C, *et al.* Parallel algorithms for the automated discovery of declarative process models. *Information Systems*, 2018, 74: 136–152. [doi: [10.1016/j.is.2017.12.002](https://doi.org/10.1016/j.is.2017.12.002)]
  - 10 Leno V, Dumas M, Maggi FM, *et al.* Automated discovery of declarative process models with correlated data conditions. *Information Systems*, 2020, 89: 101482. [doi: [10.1016/j.is.2019.101482](https://doi.org/10.1016/j.is.2019.101482)]
  - 11 Simović AP, Babarogić S, Pantelić O. A domain-specific language for supporting event log extraction from ERP systems. *Proceedings of the 7th International Conference on Computers Communications and Control (ICCCC)*. Oradea: IEEE, 2018. 12–16.
  - 12 Li GM, de Carvalho RM, van der Aalst WMP. Object-centric behavioral constraint models: A hybrid model for behavioral and data perspectives. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. Limassol: ACM, 2019. 48–56.
  - 13 Li GM, de Carvalho RM. Dealing with artifact-centric systems: A process mining approach. *Proceedings of the 9th International Workshop on Enterprise Modeling and Information Systems Architectures*. Rostock: CEUR-WS.org, 2018. 80–84.
  - 14 van der Aalst W. *Process Mining: Data Science in Action*. 2nd ed., Berlin, Heidelberg: Springer, 2016.
  - 15 Augusto A, Conforti R, Dumas M, *et al.* Split miner: Automated discovery of accurate and simple business process models from event logs. *Knowledge and Information Systems*, 2019, 59(2): 251–284. [doi: [10.1007/s10115-018-1214-x](https://doi.org/10.1007/s10115-018-1214-x)]
  - 16 van der Aalst W, Weijters T, Maruster L. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 2004, 16(9): 1128–1142. [doi: [10.1109/TKDE.2004.47](https://doi.org/10.1109/TKDE.2004.47)]
  - 17 Li GM, de Murillas EGL, de Carvalho RM, *et al.* Extracting object-centric event logs to support process mining on databases. *Information Systems in the Big Data Era*. Tallinn: Springer, 2018. 182–199.
  - 18 van der Aalst WMP. Extracting event data from databases to unleash process mining. In: vom Brocke J, Schmiedel T, eds. *BPM-driving Innovation in a Digital World*. Cham: Springer, 2015. 105–128.
  - 19 Calvanese D, Montali M, Syamsiyah A, *et al.* Ontology-driven extraction of event logs from relational databases. *Proceedings of the 13th International Workshops on Business Process Management Workshops*. Innsbruck: Springer, 2016. 140–153.
  - 20 刘聪, 程龙, 曾庆田, 等. 基于 Petri 网的分层业务过程挖掘方法. *计算机集成制造系统*, 2020, 26(6): 1525–1537. [doi: [10.13196/j.cims.2020.06.009](https://doi.org/10.13196/j.cims.2020.06.009)]
  - 21 Li GM, de Carvalho RM, van der Aalst WMP. Automatic discovery of object-centric behavioral constraint model. *Proceedings of the 20th International Conference on Business Information Systems*. Poznan: Springer, 2017. 43–58.

(校对责编: 孙君艳)