

基于改进BNF的作战管理语言语法形式化方法

罗 晨¹ 鲍广宇¹ 刘晓明¹ 袁黎苗²

(解放军理工大学指挥自动化学院 南京 210007)¹ (南海舰队信息化处 湛江 524001)²

摘 要 作战管理语言是一种旨在实现指控与仿真系统间无歧义通信的关键技术,有助于解决指控与仿真系统之间的互操作问题。首先阐述了作战管理语言的体系结构;然后通过比较分析形式化语法方法,并结合作战命令语言结构特点,提出了一种基于改进BNF的BML语法形式化描述方法;最后利用该方法对典型作战命令进行了语法形式化描述,并给出了具体实例。

关键词 作战管理语言,巴科斯范式,形式化

中图分类号 TP391.9 **文献标识码** A

Grammar Formal Method of Battle Management Language Based on Improved-BNF

LUO Chen¹ BAO Guang-yu¹ LIU Xiao-ming¹ YUAN Li-miao²

(Institute of Command Automation, PLA University of Science and Technology, Nanjing 210007, China)¹

(Informatization Section of South China Sea Fleet, Zhanjiang 524001, China)²

Abstract Battle management language(BML) is a critical technology which is used to realize unambiguous communication between C2 and M&S systems, and it can help to solve the problems of the interoperation of C2 and M&S systems. First the architecture of BML was discussed, and then different methods of formalization were compared. A grammar formal method of BML based on improved backus-aur form(IBNF) was proposed, finally, the grammars of typical orders were formalized with IBNF and specific examples were given.

Keywords Battle management language(BML), Backus-aur form(BNF), Formalization

1 引言

指挥控制系统正朝着栅格化和面向服务方向发展,在此过程中,作战推演、作战行动分析以及指挥控制系统试验与验证等能力的实现都必须依赖仿真技术的支撑,从而导致仿真与指控系统各要素之间的互操作需求日益凸显。

在上述背景下,对指控系统与仿真系统的互操作过程进行具体分析后发现,不能很好地实现作战命令在两者之间的标准化和自动化处理,是目前所存在的关键不足之一,这严重影响了互操作的效果^[1]。目前主要有两种方法解决上述问题:(1)以自然语言形式传递作战意图,由仿真系统采用自然语言理解技术进行翻译。但目前由于受到人工智能领域语言处理技术发展的限制,还很难实现自然语言消息的无歧义处理。(2)从作战意图中抽取无歧义数据元素,再利用简单的语法规则进行组织,传递给仿真系统。这种以格式化命令形式流动的消息,仿真系统就可以进行自动解释。因此,目前主要采用第二种方法实现作战命令的标准化和自动化处理。

BML是一种能够无歧义指挥命令部队和装备进行军事行动和提供态势感知、共享作战行动的语言^[2,3]。它将数据交换标准建立在通用指控数据参考模型的基础上,通过与

统指控系统仿真技术相结合,以及有效地运用Web技术,可从根本上解决指控与仿真系统间的互操作问题。因此,研究BML的相关理论和应用技术,对解决指控与仿真系统之间互操作性问题具有重要的理论意义和军事应用价值。

2 作战管理语言体系结构

按照美国弗吉尼亚大学Tolk教授提出的概念互操作性层次模型(Levels of Conceptual Interoperability Model, LCIM)^[4],系统间从无互操作性到完全互操作性可分为6个层次,因此,要实现指控与仿真系统之间的完全互操作性,必须从多个角度来实现各个层次的互操作要求。结合文献[2, 5]中对BML构成部分的描述,在前期工作中构建了一种基于多视图的BML体系结构,分别从条令视图、本体视图、表示视图、语法视图和协议视图等5个方面来描述BML,可使BML拥有语法、语义甚至概念层次的全面互操作性。上述5个视图的结构关系如图1所示。

其中,条令视图是BML定义的源,从各种军事条令中抽取术语及含义;本体视图抽象化条令视图中的术语及其概念,并定义表示视图中使用的实体、关系、属性和属性值,是沟通条令视图与表示视图的重要桥梁;表示视图是描述术语间关

到稿日期:2011-05-14 返修日期:2011-09-24 本文受国家自然科学基金项目(61174198)资助。

罗 晨(1981—),女,讲师,主要研究方向为仿真互操作、作战信息管理,E-mail:lc0810@126.com;鲍广宇(1974—),男,博士,教授,主要研究方向为系统工程;刘晓明(1956—),男,硕士,教授,博士生导师,主要研究方向为分布式系统;袁黎苗(1983—),男,硕士,工程师,主要研究方向为作战信息管理。

系、实体属性和属性值的通用数据模型,定义语法词典中用到的所有控制性词汇;语法视图是将无歧义术语规范生成有意义语句的语法规则,有效地关联了表示视图与协议视图;协议视图将语法正确的表达式转换成可用于网络传输的有效格式,是系统间信息交换的接口和描述这些接口的规范。

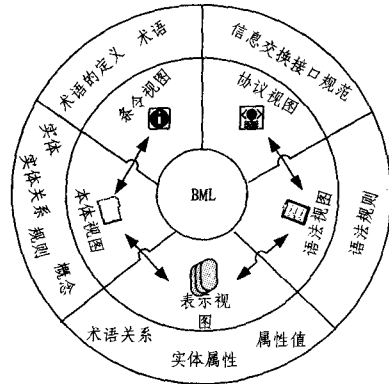


图1 基于多视图的BML体系结构

BML 体系结构中的每个视图都可实现模型中某个或多个层次的互操作性,而通过多视图的组合,可使 BML 满足各个层次的互操作性要求。统一的语法形式化描述是保证指控系统与仿真系统之间数据交换的前提^[6],本文重点针对 BML 语法视图的语法形式化描述展开研究。

3 作战管理语言语法描述方法

3.1 形式化语法分析

目前,领域内存在多种形式化语法,主要有 Z 语言^[7]、Petri 网(Petri Net, PN)^[8,9]、巴科斯范式(Backus-Naur Form, BNF)^[10]等,每一种都有自身的优势及应用领域。

Z 语言采用面向模型的方法,具有精确、简洁、无二义等优点,可保证程序的正确性,特别适用于高安全性系统开发过程中的形式化^[7]。但它也存在很多局限性,如过于抽象,理解较为困难,描述指挥控制过程中的动态行为能力不强,缺乏解析工具的支持,实用价值不高。

Petri 网与 Z 语言相比,它既有严格的数学定义,又具有可视的图形描述以及相应的工具帮助分析与实现,能够较好地刻画系统结构,充分地描述系统的动态行为^[8]。然而,它在描述作战命令时存在一定的缺陷,例如,它们在数学定义的基础上引入了大量的图形描述,使描述的简洁性和检索的简单性大大减弱,并且增加了表达、存储和修改数据的复杂性,这对于经常需要与大量数据交互的作战命令形式化描述过程来说,难以接受。

BNF 主要通过定义元符号来形式化语法,首先从一个符号开始,然后给出替换前面符号的规则,直到所有符号都不能被其它符号进一步替换。BNF 具有以下优点:①形式简单易学,便于军事人员和计算机处理;②非常精确,语法上没有歧义;③开放、易于扩展、表达能力较强,能充分表达出指挥员意图;④分析效率较高,具有多项式时间复杂度;⑤可以与 XML 等网络技术很好地配合使用,适合网络环境下的应用。

经过比较,BNF 具有简单易学、开放、易于扩展、分析效率高等优点,相对于其它形式化语法,在描述作战命令语言特性方面具有更好的适用性,可在提高指控与仿真系统之间语法互操作性方面起到理想的效果,因此使用 BNF 描述作战管

理语言语法是一种最合适的选择。

3.2 改进的 BNF 形式化方法

3.2.1 改进 BNF 的目的

在 BML 语法形式化描述过程中,必须充分考虑作战命令的结构特征。我军指控系统和仿真系统间交互的命令、报告等作战命令一般用汉语表示,受到汉语语法规则的约束,同时又遵循作战命令的格式要求,具有以下一些典型的结构特征:

(1)具有大量重复内容结构。一条命令中可能包含多个任务;一份报告中可能包含多个单元的状态信息,要求形式化方法能在分解消息时具有指代重复内容的机制。

(2)具有大量引用内容结构。包括引用其他元素和被其他元素引用两种情况。例如命令可能引用分配给某一单元的具体任务;处于同一位置的多个作战单元可能引用同一地址来说明自己的位置信息。

(3)具有大量嵌套层次结构。作战命令中元素存在上下层间的嵌套关系,因此需要不断地迭代,经历从抽象到具体的表示过程。例如计划由任务、人员等元素嵌套组合而成,而任务可进一步细分为多个更小的子元素。

(4)注重语义表达。所有的作战命令形式都必须能够完整清晰地表达指挥员的意图,避免指挥人员与机器在理解上产生歧义。

上述特征是 BNF 描述时必须充分考虑的结构,而标准 BNF 元符号较为简单,在描述作战命令的重复、引用等问题上,往往结构较为生硬、效率不高,因此需要对标准的 BNF 进行适当的改进,使其能够满足对具有上述结构的作战命令的形式化描述需求。

3.2.2 IBNF 形式化方法

为了使 BNF 更适合描述作战命令的语言特征,论文利用其开放、可扩展的特点,在平衡简洁性、兼容性和有效性的基础上,对标准 BNF 进行了适当的改进。将改进后的 BNF 称为 IBNF(Improved BNF),其元符号及说明如表 1 所列。

表1 IBNF 的主要元符号说明

符号表示	含义解释	备注
$X::=Y$	X 定义为 Y	原有
$\langle X \rangle$	X 为非终结符、必选项	原有
$[X]$	X 是可选的	原有
$X Y$	X 或 Y	原有
$*X*$	X 为终结符、必选项	改进后
$\{X\}$	X 可以重复 0 次至无数次	改进后
$\backslash X \backslash$	引用唯一标记 X 所代表的内容	改进后
$/X/$	用唯一标记 X 引用该内容	改进后
$\langle X \rangle$	括号,表示优先级顺序	改进后
$\#X\#$	X 为注释内容	改进后

IBNF 对标准 BNF 的改进主要体现在以下几个方面。

(1)定义重复多次的符号,便于表示规则需要出现多次的情况。

(2)增加引用符号,便于标识对其他规则的引用或被引用至其他规则。

(3)增加优先级表示符号,以合理组合规则中元素间的顺序。

(4)定义对注释内容的描述,可用于解释规则中的关键要素。

需要声明的是,IBNF 表达式中可能出现省略号“...”,它

并非 IBNF 元符号,而是表示省略重复的元素。此外,“\”、“{”等元符号不能与其他符号嵌套使用。

同时,为了统一规范指控系统和仿真系统之间元素名称,需要对 IBNF 基本元素的命名规则做如下约定。

(1)IBNF 中由多个词组成的元素用连接符号“-”分隔。例如,“task-time”。

(2)IBNF 中的数字元素名只可能是非负整数,不允许在数字左侧添加任何的修饰符。例如,“1223”、“0”等为合法形式,而“-1”、“063”等为非法形式。

(3)IBNF 中英文元素名的命名方式借鉴 C 语言中的变量名格式要求,规定只能由英文字母、数字组成,其中元素的开头只能是字母,并区分字母大小写。例如,“attack”、“get8s”等为合法的形式,而“9number”、“\$ invalid”、“Do’not”等为非法的形式。

4 基于 IBNF 的 BML 语法描述

作战命令是指控系统与仿真系统之间主要的信息交互形式,也是论文主要研究的 BML 消息形式。论文在 IBNF 方法的基础上,提取 BML 消息中的公共信息组件并进行形式化,深入分析典型作战命令的语言结构特点,运用 IBNF 方法进行形式化描述,使它们具有结构化形式。

论文采用四元组的结构来表示作战命令语法,记为 $G = (S, N, \Sigma, P)$ 。其中 S 代表开始符, N 代表非终结符 (IBNF 定义中用 $\langle \rangle$ 包含的内容) 的有限集, Σ 代表终结符 (IBNF 定义中 “*” 之间包含的内容) 的有限集, P 代表产生式规则的有限集。

4.1 公共信息组件的提取及 IBNF 描述

研究发现,作战命令主要由一些公共的基础元素组成,如动作、时间元素、空间元素等,论文将它们定义为公共信息组件。有针对性地提取公共信息组件并进行形式化处理,能够减少维护作战命令模板的工作量。作战命令中最重要的公共信息组件是 5W,包括 Who(谁),What(什么),When(何时),Where(何地)和 Why(为什么)^[5,11]。

下面重点分析 5Ws 公共信息组件的具体结构,并运用 IBNF 分别进行形式化描述。5Ws 公共信息组件中 Who 和 What 为简单类型,由终结符构成,无进一步扩展规则。

(1)组件 When 主要分为开始时间 Start-When 和结束时间 End-When 两类,它们的基本结构相同,都具有两种形式:一种是开始或结束的标识符、修饰符和相对时间点的组合;另一种是开始或结束的标识符、相对时间点修饰词和行动的组。以 Start-When 为例,其扩展规则为:

$$\langle \text{Start-When} \rangle ::= * \text{start} * \langle \text{Qualifier1} \rangle \langle \text{Point-in-Time} \rangle \quad (1)$$

$$\langle \text{Start-When} \rangle ::= * \text{start} * \langle \text{Qualifier2} \rangle \langle \text{Action} \rangle \quad (2)$$

其中,规则(1),(2)属于产生式规则集 P ; start 属于终结符集 Σ ,表示开始; Qualifier1 属于非终结符集 N ,为修饰词,如“不晚于”;非终结符 Point-in-Time 代表某一时间点;非终结符 Qualifier2 代表一个相对时间点的修饰词,如“在行动 Action 结束后”;非终结符 Action 代表另一个将被执行的行动。

(2)组件 Where 表示地点,分为在某地 At-Where 和经过某地 Route-Where 两种,其基本表达式为:

$$\langle \text{Where} \rangle ::= \langle \text{At-Where} \rangle | \langle \text{Route-Where} \rangle \quad (3)$$

而 At-Where 和 Route-Where 又可以分别扩展为:

$$\langle \text{At-Where} \rangle ::= * \text{at} * \langle \text{Location} \rangle \quad (4)$$

$$\langle \text{Route-Where} \rangle ::= \langle \langle \text{Source} \rangle \langle \text{Destination} \rangle \langle \text{Path} \rangle \rangle | \langle \langle \text{Source} \rangle \langle \text{Path} \rangle \rangle | \langle \langle \text{Destination} \rangle \langle \text{Path} \rangle \rangle | \dots | (* \text{along} * \langle \text{Path} \rangle) \quad (5)$$

$$\langle \text{Source} \rangle ::= * \text{from} * \langle \text{Location} \rangle \quad (6)$$

$$\langle \text{Destination} \rangle ::= * \text{to} * \langle \text{Location} \rangle \quad (7)$$

其中,“at”,“from”,“to”,“along”为终结符;非终结符 Location 来自数据模型中的元数据,代表位置信息;非终结符 Path 代表某一路线。

(3)组件 Why 用于描述任务目的,包括为了达到某个行动目标、为了实现某一状态,或是为了获得进一步的目的,具有扩展规则:

$$\langle \text{Why} \rangle ::= * \text{in-order-to} * \langle \text{Verb} \rangle [\backslash \text{Task-Label} \backslash] \quad (8)$$

$$\langle \text{Why} \rangle ::= * \text{in-order-to} * * \text{cause} * \backslash \text{EndState-Label} \backslash \quad (9)$$

$$\langle \text{Why} \rangle ::= * \text{in-order-to} * * \text{enable} * \backslash \text{Expanded-Purpose-Label} \backslash \quad (10)$$

其中,终结符“in-order-to”用于表示目的;非终结符 Verb 代表命令动作;非终结符 Task-Label 为任务引用标识符; End-State-Label 为最终状态引用标识符; $\text{Expanded-Purpose-Label}$ 为扩展目标引用标识符。

提取公共信息组件并进行形式化描述,在某一个组件变化时无需对所有作战命令模板进行更新,以大大减少维护模板的工作量。

4.2 典型作战命令的 IBNF 描述

指控系统处理的作战命令包括命令、报告、请示、计划、通报等多种形式。语法形式化方法的最终目的是为了结构化 BML 消息。通常情况下,作战命令包括头部、尾部和正文 3 个部分。其中,正文是作战命令的主体部分,各类作战命令的正文结构会有所区别。论文着重分析了命令和报告正文部分的语言结构特点,并结合具体实例,运用 IBNF 方法进行形式化描述。

4.2.1 命令的 IBNF 描述

命令用于上级对下级下达的法定性任务或决定等情况,是 BML 消息的重要形式。其基础规则可描述如下:

$$S ::= \langle \text{OB} \rangle \quad (11)$$

其中,规则(11)属于产生式规则集 P ; S 是语法的开始符,用于表示语法的开始; OB 属于非终结符集 N ,用于表示命令活动。符号“{”的使用意味着一个命令可以由任意多个基本命令表达式 OB 组成。非终结符 OB 有进一步的扩展规则。经过分析,基本表达式 OB 可以表示成子元素的有序组合,因此 OB 可扩展为:

$$\begin{aligned} \langle \text{OB} \rangle ::= & \langle \text{Verb} \rangle \langle \text{Tasker} \rangle \langle \text{Taskee} \rangle \\ & [\langle \text{Affected} \rangle | \langle \text{Action} \rangle] \langle \text{Start-When} \rangle \quad (12) \\ & [\langle \text{End-When} \rangle] \langle \text{Where} \rangle [\text{Why}] / \text{Label} / \\ & \langle \text{Mod} \rangle \end{aligned}$$

其中,非终结符 Verb 代表要执行的行动,如“进攻”、“防御”、“埋伏”等;非终结符 Tasker 代表发出命令的实体名称;非终结符 Taskee 代表接受命令的实体名称;非终结符 Affected 代表任务影响的实体名称,它的出现与否由 Verb 决定;非终结符 Action 代表另一个将被执行的行动;非终结符 Start-

When, End-When 代表时间短语, 如“不晚于某个时间点开始”, 这里的“开始”和“不晚于”都是语法终结符; 非终结符 Where 代表位置短语, 如“在何地”、“经过何地”等; 非终结符 Why 代表任务要求被执行的原因; 非终结符 Label 代表一个唯一标识符, 其它命令可以通过使用这个唯一标识符来引用该命令, 它的一般形式为“作战命令类型_引用类型_标识编号”, 如“命令_引用_标识 1”; 非终结符 Mod 代表描述特定任务所需要额外附加的信息, 如“编队”、“速度”等。

上面出现的语法符号中, Start-When, End-When, Where, Why 等属于非终结符集 N , 具有进一步的扩展规则, 在 4.1 节中已经给出; Verb, Tasker, Taskee, Affected, Action, Label, Mod 等属于非终结符集 N , 取自数据模型中的元数据; 而 start, end, at, from, to, along 等属于语法终结符集 Σ , 无需进行规则扩展。

现假设某战例中师指挥部向步兵团下达的部分命令如下: “20XX 年 3 月 16 日 18 点 00 分 00 秒出发, 进军到 XX°28′.0N, XX°08′.23E 以阻止敌突击队渗透。”

将该命令用 IBNF 描述后如下:

* 进军 * * 师指挥部 * * 步兵团 * * 开始 * * 不晚于 * * 20XX 年 3 月 16 日 18 点 00 分 00 秒 * * 到达 * * XX°28′.0N, XX°08′.23E * * 为了 * * 阻止 * \命令_引用_标识 1 / \命令_被引用_标识 2/。

其中, 引用标识“命令_引用_标识 1”表明“阻止”对象“敌突击队”的渗透需引用“命令_引用_标识 1”中的内容; 而“命令_被引用_标识 2”表明其他规则如需引用该命令结构可用标识“命令_被引用_标识 2”。

其语法树结构如图 2 所示。

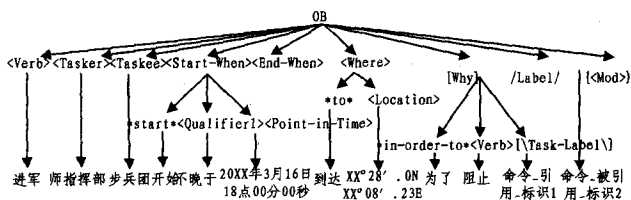


图 2 命令活动语法树

4.2.2 报告的 IBNF 描述

报告用于下级向上级反映作战情况, 回复上级的命令等, 它与命令最重要区别在于对内容的确定性具有不同的掌握。对于命令而言, 作战单元、作战行动等内容是确定的, 而对于报告而言, 这些内容通常是不确定的。因此, 需要为报告增加相应的产生式规则和非终结符, 来表示作战单元类型、装备类型、模糊度及其他重要信息。报告的形式化规则具体如下所述。

一份报告由 1 个或多个基本报告表达式 RB 组成, 基础规则可描述如下:

$$S ::= \{ \langle RB \rangle \} \quad (13)$$

根据报告内容的不同, 报告分为任务报告、事件报告和状态报告 3 种类型, 它们都可以表示成各自子元素的有序组合。它们的基本活动规则 RB 如(14)–(16)所示。

$$\begin{aligned} \langle RB \rangle ::= & * \text{Task-Report} * \langle \text{Verb} \rangle \langle \text{Executer} \rangle \\ & [\langle \text{Affected} \rangle | \langle \text{Action} \rangle] \langle \text{When} \rangle \langle \text{Where} \rangle \quad (14) \\ & [\langle \text{Why} \rangle] \langle \text{Certainty} \rangle / \langle \text{Label} \rangle / \{ \langle \text{Mod} \rangle \} \end{aligned}$$

$$\langle RB \rangle ::= * \text{Event-Report} * \langle \text{EVerb} \rangle$$

$$[\langle \text{Affected} \rangle | \langle \text{Action} \rangle] \langle \text{When} \rangle \langle \text{Where} \rangle \quad (15)$$

$$\langle \text{Certainty} \rangle / \langle \text{Label} \rangle / \{ \langle \text{Mod} \rangle \}$$

$$\langle RB \rangle ::= * \text{Status-Report} * \langle \text{Hostility} \rangle \langle \text{Regarding} \rangle$$

$$[\langle \text{Identification} \rangle \langle \text{Status-Value} \rangle] \quad (16)$$

$$\langle \text{When} \rangle \langle \text{Where} \rangle \langle \text{Certainty} \rangle / \langle \text{Label} \rangle / \{ \langle \text{Mod} \rangle \}$$

其中, 基本规则(14)–(16)属于产生式规则集 P ; Task-Report, Event-Report 和 Status-Report 属于终结符集 Σ , 分别用于表示任务报告、事件报告和状态报告 3 种类型; Verb, Affected, Action, When, Where, Why, Label, Mod, Tasker, Label 等属于非终结符集 N , 用法与命令规则中相同; Certainty 属于非终结符 N , 来自于数据模型, 用于表示确定性, 如“真实”、“可能为真”、“不确定”等。

规则(14)中用非终结符 Executer 代替非终结符 Taskee, 因为报告发送方不一定知道执行任务单元的名称。根据发送方掌握的信息, Executer 可能是 Taskee, Agent, Theme 3 种情况中的任意一种。

$$\langle \text{Executer} \rangle ::= \langle \text{Taskee} \rangle \quad (17)$$

$$\langle \text{Executer} \rangle ::= \langle \text{Agent} \rangle / [\langle \text{Label} \rangle] \quad (18)$$

$$\langle \text{Executer} \rangle ::= \langle \text{Theme} \rangle / [\langle \text{Label} \rangle] \quad (19)$$

如果报告发送方知道执行单元名称, 则按规则(17)扩展成 Taskee, 此时 Taskee 即为执行单元名称; 如果报告发送方不知道单元名称但知道单元类型时, 可按规则(18)扩展成 Agent, 然后 Agent 能够按规则(20)扩展, 包括执行单元的规模(Size)、敌对程度(Hostility)以及类型(Unit-type) 3 个子元素, 如“一个连队的友军陆战队员”; 如果报告发送方只知道单元使用的装备类型时, 则按规则(19)扩展成 Theme, Theme 又可进一步按规则(21)扩展, 包括装备数量(Count)、敌对程度(Hostility)和装备类型(Equipment-type) 3 个子元素, 如“3 艘敌军潜艇”。

$$\langle \text{Agent} \rangle ::= \langle \text{Size} \rangle \langle \text{Hostility} \rangle \langle \text{Unit-type} \rangle \quad (20)$$

$$\langle \text{Theme} \rangle ::= \langle \text{Count} \rangle \langle \text{Hostility} \rangle \langle \text{Equipment-type} \rangle \quad (21)$$

事件报告规则(15)中的非终结符 EVerb 来自于用户定义的数据模型, 用于表示特定类型事件, 如“飓风来袭”。规则(16)中的非终结符 Regarding 代表报告的主题, 用以描述报告的类型, 它往往伴随着非终结符 Hostility。Hostility 代表敌对状态, 如“友军”或者“敌军”等。Regarding 进一步扩展如下:

$$\begin{aligned} \langle \text{Regarding} \rangle ::= & [\langle \text{position-general} \rangle | \langle \text{status-general} \rangle | \\ & \langle \text{status-material} \rangle | \langle \text{status-person} \rangle] \quad (22) \end{aligned}$$

规则(22)将状态报告分为 4 种不同类型: 位置状态报告(position-general)、作战单元的整体状态报告(status-general)、物资状态报告(status-material)以及人员状态报告(status-person)。

上面出现的语法符号中, Agent, Theme, Size, Hostility, Unit-type, Count, Equipment-type, position, status-general, status-material, status-person 等属于非终结符集 N , 都取自数据模型中的元数据; 而 Task-Report, Event-Report, Status-Report 等属于语法终结符集 Σ , 无需进行规则扩展。

现假设某战例中步兵团向师指挥部发送的自己位置状态报告如下: “20XX 年 3 月 17 日 12 点 24 分 08 秒时刻我团到达 XX°48′.13N, XX°38′.42E。”

(下转第 209 页)

参考文献

- [1] Vapnik V N. The Nature of Statistical Learning Theory (seconded) [M]. New York: Springer, 2000
- [2] 文益民, 王耀南, 吕宝粮, 等. 支持向量机处理大规模问题算法综述[J]. 计算机科学, 2009, 36(7): 20-25
- [3] 顾亚祥, 丁世飞. 支持向量机研究进展[J]. 计算机科学, 2011, 38(2): 14-17
- [4] Fung G, Mangasarian O L. Proximal Support Vector Machine Classifiers[C]// Proceedings of Knowledge Discovery and Data Mining. New York, USA: ACM, 2001: 77-86
- [5] Mangasarian O L, Wild E W. Multisurface Proximal Support Vector Classification via Generalized Eigenvalues [J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 2006, 28(1): 69-74
- [6] Jayadeva, Khemchandani R, Chandra S. Fast and Robust Learning Through Fuzzy Linear Proximal Support Vector Machines [J]. Neurocomputing, 2004, 61: 401-411
- [7] Jayadeva, Khemchandani R, Chandra S. Fuzzy multicategory proximal support vector classification via generalized eigenvalues[J]. Soft Computation, 2007, 17(5): 679-685

- [8] Jayadeva, Khemchandani R, Chandra S. Twin Support Vector Machines for Pattern Classification[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2007, 29(5): 905-910
- [9] Kumar M A, Gopal M. Least squares twin support vector machines for pattern classification[J]. Expert Systems with Applications, 2009, 40(1): 7535-7543
- [10] 杨绪兵, 陈松灿, 杨益民. 局部化的广义特征值最接近支持向量机[J]. 计算机学报, 2007, 30(8): 1227-1234
- [11] 杨绪兵, 潘志松, 陈松灿. 半监督型广义特征值最接近支持向量机[J]. 模式识别与人工智能, 2009, 22(3): 349-353
- [12] Blake C L, Merz C J. UCI Repository for Machine Learning Databases[EB/OL]. <http://www.ics.uci.edu/mllearn/MLRepository.html>, May 2010
- [13] Gunn S R. MATLAB Support Vector Machine Toolbox[EB/OL]. <http://www.isis.ecs.soton.ac.uk/isystems/kernel/>, May 2010
- [14] Hsu C-W, Chang C-C, Lin C-J. A Practical Guide to Support Vector Classification [EB/OL]. <http://www.csie.ntu.edu.tw/~cjlin>, July 2010

(上接第 192 页)

将上述报告用 IBNF 描述如下:

* 状态报告 * * 我方 * * 位置状态 * * 于 * * 20XX 年 3 月 17 日 12 点 24 分 08 秒 * * 到达 * * XX°48'.13N, XX°38'.42E * * 事实情况 * / 报告_引用_标识 1/。

其语法树结构如图 3 所示。

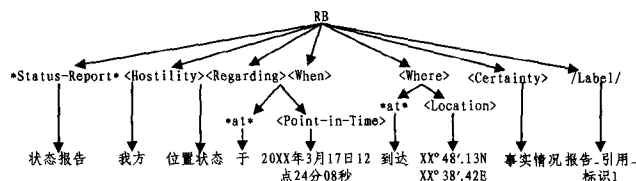


图 3 状态报告语法树

结束语 BML 是一种实现指控与仿真系统间无歧义通信的标准语言规范^[12]。针对典型作战命令的语言结构特点, 在 BNF 基础上进行了适当改进的形式化方法 IBNF, 是用来描述 BML 语法的一种合适选择。

本文提出的 IBNF 语法形式化描述方法能够使作战命令具有结构清晰、规范的统一描述, 使指控系统与仿真系统之间交换的作战命令都具有结构化形式, 从而为实现语法层次上的互操作性提供技术支撑。应用该方法可以构建作战命令模板, 按照模板配置相应参数, 即可生成指控系统和仿真系统都可理解和处理的结构化作战命令, 从而有效实现语法层次的互操作性。

下一步将围绕 BML 体系结构中协议视图和本体视图展开研究, 通过相应的转换算法, 将结构化作战命令转换成符合网络传输需求的标准网络交换形式(如 RDF/XML 文档), 从而实现指控系统与仿真系统之间的数据交换和语义共享, 使指挥控制过程更加迅速、可靠。

参考文献

- [1] Reus D, Kromp D. BML-enabling of national C2 systems for coupling to Simulation[C]// IEEE Spring Simulation Interopera-

- bility Workshop. Providence, Rhode Island, USA, 2008
- [2] 彭勇, 彭春光, 龚建兴, 等. 作战管理语言研究综述[J]. 系统仿真学报, 2009, 21(17): 5336-5339
- [3] Tolk A, Turnitsa C. Battle Management Language: A Triangle with Five Sides [C]// Fall Simulation Interoperation Workshop, 06S-SIW-016. Orlando, FL, USA, 2006
- [4] Andreas T, Muguira J A. The Levels of Conceptual Interoperability Model (LCIM) [C]// Proceedings IEEE Fall Simulation Interoperability Workshop. Orlando, FL: IEEE CS Press, September 2003
- [5] Ma Liang-jun, Chen Lin, Lu Peng, et al. A new knowledge representation method for Battle Management Language[C]// 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). Yantai, Shandong, Aug. 2010: 1761-1766
- [6] Ulrich S, Haarmann, Bastian, et al. A Grammar for Battle Management Language[C]// IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications (DS-RT). Salford, United Kingdom, Sept. 2011: 155-159
- [7] Gao Xiao-lei. The Design and Implementation of Z Language Editor[C]// Algorithms and Architectures for Parallel Processing, 9th International Conference (ICA3PP). Taipei, Taiwan, June 2009
- [8] 叶丽君. 基于 UML 描述的概念模型校验技术研究[D]. 郑州: 解放军信息工程大学, 2008
- [9] 顾妍午, 王遵彤, 吴启迪. 面向对象 Petri 网技术在系统建模中的应用[J]. 同济大学学报: 自然科学版, 2010, 38(3): 437-441
- [10] Ye C, Zhong J, Feng Y. Attribute-based access control policy specification language [J]. Journal of Southeast University, 2008, 24(3): 260-263
- [11] Schade U, Michael R. H. Formalizing Battle Management Language: A Grammar for Specifying Orders[C]// 2006 Spring Simulation Interoperability Workshop, 06S-SIW-068. Huntsville, AL, 2006
- [12] 岳秀清, 张磊, 别晓峰. 外军指挥控制建模研究现状及启示[J]. 军事运筹与系统工程, 2009, 23(1): 70-74