

Soar 在水面舰艇 CGF 防空决策行为模型构建中的应用

吴 涛, 孙向军, 赵斯强

(海军指挥学院海军作战实验中心, 江苏 南京 210016)

摘 要: 计算机生成兵力(CGF)的行为模型研究一直是人工智能技术应用于军事仿真的一个重要研究方向。基于综合认知架构 Soar 设计了一种水面舰艇 CGF 的自动防空决策行为模型,并在海战 CGF 系统中实现了该模型,应用表明该模型可以较为真实地仿真水面舰艇 CGF 的防空决策行为,取得良好的仿真效果。

关键词: Soar; 计算机生成兵力; 行为建模

中图分类号: E917; TP391.9

文献标识码: A

DOI: 10.3969/j.issn.1673-3819.2013.02.025

The Application of Soar in Surface Warship CGF Defense Decision-Making Behavior Modeling

WU Tao, SUN Xian-jun, ZHAO Si-qiang

(Navy Command College Naval War Gaming Center, Nanjing 210016, China)

Abstract: Computer generated forces behavior modeling research is an important research about application of AI Technology military simulation. Based on the cognitive architecture of the Soar, design of surface warship CGF automatic defense decision-making behavior model in Naval Battle CGF system is implemented. Application shows that the model can correctly simulate warship CGF decision behavior and obtain good effect.

Key words: Soar; CGF(Computer Generated Force); behavior modeling

目前计算机生成兵力(CGF)的研究主要集中于 CGF 的物理模型和行为模型上。物理建模主要研究对象的物理本质及其内在的工作机理,如系统的动力学模型、传感器模型、武器系统的火控模型和毁伤评估模型等,物理模型实现技术较为成熟,现有的研究主要集中在模型的细节上,使模型更加符合真实性。行为建模是指“对在军事仿真中需要表示的人的行为或表现进行建模”,美军称之为 HBR^[1-2](Human Behavior Representation),其主要反映人类作战行为(如指挥员的个体行为、参战人员操纵武器平台和武器装备的行为),使装备实体能够不需要与人的交互而自动地对虚拟战场环境的状态和事件做出合理反应。无论是模拟训练、作战分析等仿真应用都会用到自主进行决策和行动的计算机生成兵力。可以说,CGF 行为模型的研究与应用在军事仿真领域,特别是分析仿真领域有着巨大的应用前景。

然而 CGF 行为建模一直是 CGF 研究和应用的难点部分,国内外的学者和研究机构都对此展开了广泛的研究,但对于其建模和实现方法的研究还不充分。目前常用的行为建模技术有:有限状态机^[3]、人工神经

网络^[4]、基于多智能体的方法^[5-6]等,还有一些 CGF 系统则建立在综合认知体系结构之上,例如美国空军使用的空战 CGF 系统 IFOR(Intelligent Forces)就是基于综合认知架构 Soar^[7-8]来构建的。Soar 相对于其他方法不仅提供了通用的描述规则和推理引擎,同时还提供了规划、学习等高级功能,可以帮助用户高效地实现行为建模。Soar 不仅在军事仿真领域有着成功的应用,而且在游戏的人工智能领域也有着广泛而成功的应用。

本文将综合认知架构 Soar 引入到海战 CGF 系统中,并建立了水面舰艇 CGF 的防空决策行为模型。文中首先介绍了 Soar 的基本架构、推理机制、Soar 规则等,然后以此实现了水面舰艇 CGF 的自动防空决策行为模型,并应用到了海战 CGF 系统中,取得了较好的仿真效果。

1 Soar 概述^[9-10]

1.1 Soar 简介

Soar 是由美国密歇根大学人工智能实验室开发的一种用于构造通用智能系统的体系架构。Soar 的设计思想是基于这样的假设^[11]:任何一个有确定目标的行为可以分解为针对某一状态(State)下的一系列算子(operator)选择和算子应用的过程。在此,目标可以被描述成一个所求解问题的期望结果,状态(state)则表示所求解问题的状态,算子(operator)则被用来将一

收稿日期: 2012-09-27

修回日期: 2012-10-26

作者简介: 吴 涛(1979-) 男,湖北黄冈人,讲师,研究方向为海军作战模拟。

孙向军(1972-) 男,博士,副教授。

赵斯强(1958-) 男,教授,博士生导师。

个状态转换成另外一个状态,通常可理解为一个动作。随着 Soar 的运行,Soar 重复如下过程:提出算子、选择算子和应用算子,直到达到所求解问题的目标。

1.2 Soar Agent 的框架结构

在 Soar 架构下,当 Soar 用来求解一个问题时,进行

决策推理的程序描述为 Soar Agent。Soar Agent 的框架结构如图 1 所示。该框架结构主要包含外部感知、长期知识存储器、工作内存、认知处理、行为输出等五个主要部分。

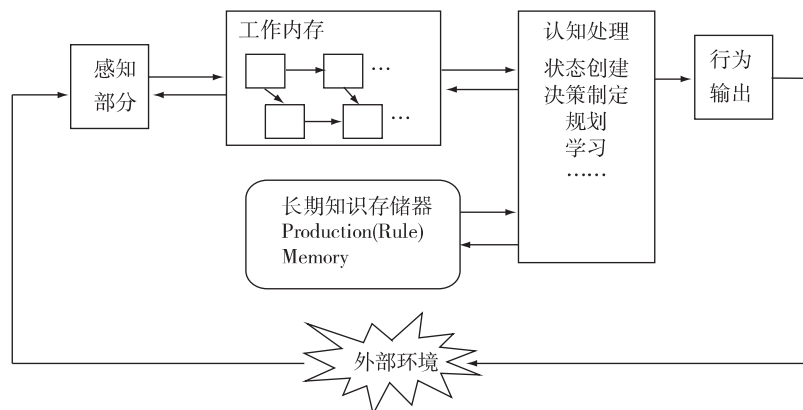


图1 Soar Agent 的认知框架

感知部分: Soar Agent 通过感知来获取外部世界的信息。

长期知识存储器(long-term Memory): 进行决策的知识可以被编码,通常这类知识被称为长期知识。在 Soar 中,长期知识被编码为产生式,知识的集合称为产生式(规则)内存 Production(Rule) Memory。产生式(规则)通常是由用户创建或者 Soar 的学习机制(chunking)创建的。

工作内存(Working Memory): 用来表示当前问题求解过程中的状态(State),工作内存是在 Soar 运行的过程中产生。工作内存包含了一个 Soar Agent 所感知到的外部世界的信息以及内部的推理,主要包括感知数据、中间推理的结果、当前的算子和所求解问题的目标。在 Soar 中,工作内存是工作内存元素(WME: Working Memory Elements)的集合,每个工作内存元素通用采用下面的三元组来描述:(identifier ^ attribute value),其中 identifier 是一个内部符号,由 Soar 运行时产生;attribute 为属性;value 为值。

WME 通常被组成一个状态图,因此每个工作内存元素(WME)直接或间接地与一个状态符号(state symbol)相关。

认知处理: 认知处理部分与工作内存、长期知识存储器交互,完成状态创建、决策制定、规划、学习等任务。认知处理部分是 Soar Agent 的推理引擎。

行为输出部分: 主要功能是输出行为并对外部环境施加影响。

1.3 Soar 进行决策推理的步骤

Soar 作为决策推理的引擎,其主要工作是将知识应用于当前的状态以便决定外部的行动。Soar 采用反向推理(向后链接)的策略,用四个阶段的循环来控制规则的扫描,这四个阶段是:

1) 匹配。由顶向下(即从知识库中第一条规则开始)依次扫描知识库中所有规则,与短期记忆中的当前事实相对比,以搜索满足条件的规则,确定所匹配的目标;

2) 选择。在发生冲突,即同时数条规则被匹配时,根据预先确定的优先准则,求出所有冲突规则的优先度,决定触发优先度最高的规则;

3) 激活规则。调用匹配所触发规则的所有子目标的行动;

4) 动作。把所触发规则的结论输出到外部环境中。

运行时,推理机重复这四个阶段的循环,根据存储部分的知识及所获得的外部环境信息,不断由已知的前提推出未知的结论记录到工作内存(Working Memory)中,作为新的前提或事实继续推理过程,直到得出最终结论。

1.4 Soar 规则的语法

在 Soar 中,知识可以被编码成 Soar 规则,这些 Soar 规则可以存储为一个文本文件。Soar 有一套固定格式和精确的语言来描述规则,以便 Soar 推理引擎来解释 Soar 规则。

每条规则以“sp”符号开头,规则的内容包含在一

个大括号中{ ... } ,主要包括规则名称、规则的文档描述(可选)、规则类型(可选)、产生式的条件、特殊符号“-->”、产生式的动作。规则模板如下:

```
sp { rule* name
    ( condition)
    ( condition)
    ... //规则执行的条件
    -->
    ( action)
    ( action) //条件符合后所产生的动作列表
    ...}
```

下面主要对产生式的条件和产生式动作的描述方法进行简要的描述。

1.4.1 产生式的条件

主要用于与工作内存中的工作内存元素(WME Working Memory Elements)进行匹配,当所有条件都匹配时,这条规则将被触发并且所有动作列表都被执行。每个条件在一对圆括号内,主要采取如下方式进行描述:

```
( identifier - test ^attribute1 - test value1 - test
    ^attribute2 - test value2 - test
    ^attribute3 - test value3 - test
    ... .. )
```

单个条件用来测试单个工作内存元素的属性,或者是一个 objects(identifier 相同的 WME's 被成为 Objects)的属性。产生式的第一个条件必须与工作内存中的一个状态相匹配。因此,第一个条件必须以特殊的“state”符号开始。所有其他条件和动作必须直接或间接与第一个条件相关。

1) 变量。产生式中的变量与工作内存元素中处于 identifier Attribute 或者 value 位置中的常量进行匹配。如果一个变量出现在一个产生式中的多个条件中,那么产生式仅与同一个 identifier 或者常量匹配。因为 identifier 是在 Soar 运行时产生的,所以在条件中不可能包含对指定 identifier 的测试。变量也提供了一种机制,这种机制将 identifier 和常量传递到一个产生式中描述动作的那一边。从语法上来说,变量由这样的形式“< variable name >”来表示,尖括号中间是变量名称。

2) 用于值(这里的值表示常量或变量)比较的谓词。对条件中的 identifier attribute,或者 value(无论是常量还是变量)进行测试时,Soar 提供了以下谓词进行值的比较。Soar 的谓词有六种,如表 1 所示。

3) 算子可接收的优先度(Acceptable preferences for operators)。如果在产生式条件中的 value-test 后面如

果紧跟一个“+”符号,则允许一个产生式对候选算子及其属性的存在性进行测试。

表 1 Soar 谓词表

| 谓词 | 谓词的语义 |
|-------|-----------------|
| < > | 与符号后的值不相等 |
| < = > | 与符号后的值相同类型 |
| < | 从数值上比小于符号后的值 |
| < = | 从数值上比小于或等于符号后的值 |
| > | 从数值上比大于符号后的值 |
| > = | 从数值上比大于或等于符号后的值 |

1.4.2 产生式的动作

产生式的动作主要包括以下 3 个方面:

1) 加入一个工作内存元素(WME)到工作内存中。从 Soar 规则的语法上看,创建工作内存元素的与前面介绍条件描述的语法类似。

2) 从工作内存中删除一个工作内存元素。从语法上看,action 的描述与条件的描述一样,需要在(action)中加一个减法符号“-”来表示删除。

3) 执行其他动作。其他动作不改变工作内存,而是执行与 soar 相关的一些动作。如停止和暂停 soar、在 soar 进行文本的输入和输出(利用 Soar 的命令)、执行简单的计算功能(如:加、减、乘、除、求最大值、三角函数计算等)、对 soar 的学习机制进行控制、用户自定义 Soar 可以执行的命令以及接口等。

2 水面舰艇 CGF 防空决策行为模型的构建

2.1 水面舰艇防空军事行动规则

要建立水面舰艇 CGF 的防空决策行为模型,首先要抽象出水面舰艇防空的军事行动规则,军事行动规则是建立 CGF 防空决策行为模型的基础。

通常,当对空警戒雷达发现远距离空中目标、判断为敌警戒飞机时,应马上发射远程无源干扰弹(箔条+烟幕),进行迷惑干扰,同时进行有源压制,破坏敌机的跟踪;对敌火控雷达进行有源干扰,施放冲淡干扰。由于编队舰距小,冲淡干扰宜选用小比例方式;如果干扰发射时机偏晚,可考虑以主动干扰破坏导弹的搜索跟踪,迫使其重新搜索。同时,使用无线电通信干扰器材干扰敌攻击机编队之间的无线电通信;对近距离来袭导弹实施质心干扰和距离拖引干扰。

1) 来袭目标成批地进入防空武器作战空域,它们构成泊松流,每个编队有目标 m 个,各编队按参数为 λ 的泊松分布随机地到达;

2) 火力单元的一次射击后观察射击结果,如目标被击毁,火力单元即刻转向其它目标;如目标未被击

毁,继续进行射击,直到该目标被击毁或飞离,射击时间服从指数为 m 的负指数分布,即: ($t > 0$)

3) 武器系统有 n 个火力单元,目标进入武器杀伤区时,若有火力单元空闲就捕捉该目标进行射击;若无火力单元空闲,目标将在杀伤区上空飞行。目标逗留时间服从指数为 v 的负指数分布,即 $t > 0$ 。

所有火力单元都在工作时,若有火力单元结束前一次射击,便转向射击其它目标;若目标飞到杀伤区的远界仍无火力单元对它射击,目标就飞越防区突破;被消灭的目标立即消失。

具备上述条件要求舰艇 CGF 具备如下参数,其参数名称及类型描述如表 2 所示。

表 2 舰艇属性表

| 字段 | 属性 | 含义 | 备注 |
|---------------|--------|----------|------|
| type | string | 舰艇类型 | |
| economicspeed | float | 经济航速 | |
| velocity | float | 当前航速 | |
| heading | float | 舰艇航向 | |
| navnumber | int | 舰空导弹数量 | 0-48 |
| missilelevel | float | 舰空导弹速度 | |
| fireavailnum | int | 可用的火力通道数 | 0-6 |
| killdistance | float | 杀伤区远界 | |
| shootinterval | float | 导弹发射间隔时间 | 12 |
| antiair | string | 反导防空开始信号 | |
| jamseven | bool | 干扰 | |

2.2 构建水面舰艇 CGF 防空行为的 Soar 规则

根据军事行动规则,下面就是根据 Soar 规则的语法,将军事行动规则转换为 Soar 推理引擎所能识别的 Soar 规则,即军事知识。下面是根据军事行动规则转换而来的 Soar 规则。

```
( state <s> ^io <io> )
-->
( <s> ^name navsh )
}
//提出目标算子
sp { propose* fireair
  ( state <s> ^name navsh
    ^io <io> )
  ( <io> ^input - link. shipair <sc>
    ^input - link. target1 <tc> )
  ( <sc> ^navnumber <x>= 1
    ^fireavailnum <y>= 1
    ^missilelevel <mv>
    ^killdistance <kd> )
  ( <tc> ^bekilnum = 0
    ^distance <de> { > 15500 < 200000 }
```

```
    ^velocity <v1> )
-->
  ( <s> ^operator <op> + )
  ( <op> ^name fire
    ^todi 90000 )
}
//应用目标算子
sp { apply* fireair
  ( state <s> ^operator <op>
    ^io. output - link <ol>
    ^io. input - link. target1 <tc> )
  ( <tc> ^bekilnum = 0
    ^distance <de> { > 15500 < 200000 } )
  ( <op> ^name fire
    ^todi > <de> )
-->
  ( <ol> ^target - output 1 )
}
//提出干扰算子
sp { propose* interfere
  ( state <s> ^name navsh
    ^io <io> )
  ( <io> ^input - link. ship <sc>
    ^input - link. target1 <tc> )
  ( <sc> ^jamseven <x>= 1 )
  ( <tc> ^bekilnum = 0
    ^distance <de> { > 22000 < = 26000 } )
-->
  ( <s> ^operator <op> + )
  ( <op> ^name interfere )
}
//应用干扰算子
sp { apply* interfere
  ( state <s> ^operator <op>
    ^io. output - link <ol>
    ^io. input - link. target1 <tc> )
  ( <tc> ^bekilnum = 0
    ^distance <de> { > 22000 < = 26000 } )
  ( <op> ^name interfere )
-->
  ( <ol> ^caninter fereon )
}
```

该规则主要包括提出目标算子、应用目标算子、提出干扰算子和应用干扰算子四个产生式(规则)。其中 $<io>$ 表示推理过程中所需要输入的参数。以提出目标算子为例,产生式中的条件部分主要包括两个输入,

shipair(条件中用变量名 sc 表示,表示舰艇参数)和 target1(条件中用变量名 tc 表示,表示目标导弹的参数),舰艇参数主要和表 2 中描述的参数来表示,其具体参数包括在条件“(< sc > …)”中,舰艇参数主要有: navnumber(舰空导弹数量,在条件中用变量名 x 表示)、fireavailnum(可用的火力通道数,在条件中用变量名 y 表示)、missilevel(舰空导弹速度,在条件中用变量名 mv 表示)以及 killdistance(杀上区远界);目标导弹参数主要有: bekilnum(目标导弹被锁定的数量)、distance(目标导弹与舰艇的距离,在条件中用变量名 de 表示)、velocity(目标导弹的速度,在条件中用变量名 v1 表示),条件中每个参数变量名如果跟有谓词符号,则表示参数需满足的条件。如:

< x > 1 表示的是舰空导弹数量需大于 1 枚;

< y > = 1 表示的是可用的火力通道至少有 1 个;

distance < de > { > 15500 < 200000 } 表示目标导弹与舰艇距离需大于 15.5km 且小于 200km。

在以上该产生式的条件列表中,“- >”符号后面的部分则表示产生式(规则)的动作列表。

其余三个产生式的表示与此类似,所表达的含义可依次类推。

2.3 水面舰艇 CGF 防空决策行为模型的执行流程

舰艇(编队) CGF 防空决策行为决策流程如下:

1) 海战仿真系统初始化,包括仿真环境初始化,舰艇 CGF 状态初始化,仿真结果初始化;

2) 加载舰艇(编队)的 Soar 规则;

3) 舰艇(编队) CGF 在仿真运行的周期中,通过相应的探测感知模型,不断感知战场环境信息,信息主要包括:舰艇编队的机动状态、有效探测范围内的敌方活动状态,以及我方兵力的精确运动状态、武器系统状况和雷达探测区域内敌方的运动状态;

4) 决策模型通过 Soar 所提供的接口将感知数据输入到 Soar 推理引擎中;

5) Soar 引擎在舰艇 CGF 的规则中进行搜索,选出条件部分与环境状态相匹配或匹配程度最高的规则,并将该规则的动作部分发送给执行器。

6) 执行器根据接收到的行动目标,产生相应的动作,作用于环境,改变环境状态。

3 结束语

本文将综合认知架构 Soar 引入海战仿真系统中,实现了舰艇 CGF 的防空决策行为模型,通过仿真实验表明该方法来构建 CGF 的行为模型是可行的。但是这

种实现仅建立在较为简单的军事规则之上,实际上军事规则比抽象出来的军事规则更为复杂,这将直接导致相应 Soar 规则的复杂性;其次,实验环境中战场态势也很简单,实际战场环境要复杂得多。这些因素都将增加 Soar 引擎推理的推理时间。在下一步的研究工作中,需要进一步抽象出更加符合实际的军事规则,而且可以研究 Soar 所提供的学习机制,并应用到 CGF 的行为模型中,从而进一步提高 CGF 行为模型的适应性。

参考文献:

- [1] Richard W. Pew, Anne S. Mavor. Modeling Human and Organization Behavior: Application to Military Simulations [R]. National Academic Press, Washington. D. C., 1998.
- [2] Richard W. Pew and Anne S. Mavor. Representation Human Behavior in Military Simulations: Iterim Report [R]. National Academic Press, Washington. D. C., 1997.
- [3] 刘小玲,潘巨辉. FSM 在海军作战仿真 CGF 中的应用 [J]. 计算机仿真, 2007, 24(8): 24-27.
- [4] 张立友,唐晓明,郭齐胜. 舰艇 CGF 系统智能推理决策研究 [J]. 计算机仿真, 2006, 23(6): 37-40.
- [5] 黄柯棣,刘宝宏,黄健,等. 作战仿真技术综述 [J]. 系统仿真学报, 2004, 16(9): 1891-1892.
- [6] 陈挺,李照顺. 海战仿真中反潜直升机 CGF 的行为建模方法 [J]. 指挥控制与仿真, 2001, 33(1): 82-84.
- [7] Rosenbloom, P. S., Johnson, et al. Intelligent Automated Agents for Tactical Air Simulation: A Progress Report [R]. Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation. Orlando, FL, 1994.
- [8] Laird, J. E. & Jones, R. M. Coordinated Behavior of Computer Generated Forces in TacAir-Soar [C]. Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation. Orlando, FL, 1994.
- [9] John E. Laird, Clare Bates Congdon. The Soar User's Manual. University of Michigan, 2006.
- [10] Laird, J. E. Newell, A. and Rosenbloom, P. S. 1987. SOAR: An Architecture for General Intelligence. Artificial Intelligence 33(1): 1(R) C64.
- [11] 张航义. 基于 Agent 的 CGF 行为建模技术研究 [J]. 计算机仿真, 2003, 20(8): 79-81.
- [12] 刘秀罗. CGF 建模相关技术及其在指挥控制建模中的应用研究 [D]. 长沙: 国防科学技术大学, 2001.