仿真环境中智能化控制语言的研究

胡圣明 李青山 陈 平 褚 华
(西安电子科技大学软件工程研究所 西安710071)

摘 要 智能化控制语言可控制仿真系统中的实体,使其表现出智能化行为。本文定义了一种智能化控制脚本语言ICSL(Intelligent Control Scripting Language),并实现了ICSL的解释器。ICSL支持基本的控制语句和核心对象,针对领域特征,对仿真领域的语义进行编码,并提供语义扩展机制,为控制策略的制定者提供一种灵活表达策略的手段。ICSL语言的解释器将智能化策略映射为仿真系统中实体的行为序列,实现了对仿真实体的智能化控制。最后,本文通过把ICSL语言用于战场仿真系统对其实用性进行了系统的实验研究,结果表明,基于ICSL语言定义的控制策略可以有效地映射到仿真系统中,从而完成对仿真实体的智能化控制。

关键词 智能化控制,仿真,脚本语言,控制策略

Research on Intelligent Control Language in Simulation System

HU Sheng-Ming LI Qing-Shan CHEN Ping CHU Hua
(Institute of Software Engineering, Xidian University, Xi'an710071)

Abstract Intelligent control languages can be used to control entities in a simulation system and make them behave intelligently. An intelligent scripting language ICSL(Intelligent Control Scripting Language) used for the controlling of simulation entities and implementation of its interpreter is presented in this paper. ICSL supports basic control statements and core objects. It encodes the semantics of simulation fields and supply mechanism of semantics extending. It is an instrument to present intelligent control strategies in different simulation fields. The interpreter of ICSL can map intelligent strategies into the behaviors of entities in a simulation system and accomplish the intelligent control. Finally, ICSL is applied to a battlefield simulation system. The result shows that using ICSL intelligent control strategies can be mapped into the simulation system efficiently, and then the aim of intelligent control on simulation entities can be achieved.

Keywords Intelligent control, Simulation, Scripting language, Control strategy

1 引言

仿真系统中存在大量的实体,这些实体应具有自动的智能化行为,例如,在一个战场仿真系统中,当我方飞机探测到另一飞机时,首先应判断此飞机是敌方还是我方,若是敌方,应采取一定的行为,或者发射武器攻击敌方,或者躲开敌方武器攻击,或者发射电子干扰武器等。要使这些实体具备上述的智能化行为,可采取人工控制的方式[1],即由人工操作来控制实体的行为,但这种方式的缺点在于控制效率低,而且仿真系统中存在大量的实体,对每一个实体都采用人工控制的方式也不现实;若将控制策略直接编码到仿真系统中,将导致仿真系统无法变更控制策略。

针对上述问题,研究一种智能化控制语言对仿真系统中的实体进行智能化控制是十分必要的。控制策略可分为策略的条件和策略的行为,策略的条件是由仿真系统中的环境计算而来,策略的行为则体现为仿真系统中实体的行为序列。本文将控制策略与仿真系统相分离,定义了一种用于描述智能化控制策略的语言ICSL,并实现了该语言的解释器。

基于ICSL语言,策略制订者可描述出智能化控制策略,ICSL解释器把这种控制策略映射到仿真实体的行为变化逻辑中,便可达到对仿真系统中实体进行智能化控制的目的。ICSL语言系统能够方便地获取仿真系统中的各种信息,即获取环境参数,并根据策略对信息数据进行计算,再把策略计算的结果映射到仿真系统中,仿真实体根据该策略控制要求来产生新的行为逻辑,即执行策略的行为。

2 ICSL语言的设计

为方便地描述出智能化控制策略,ICSL定义了基本的控制语句;为达到获取仿真系统的环境信息和控制仿真实体行为的目的,ICSL语言支持从仿真环境中抽象出核心对象,以便策略制定者使用;不同领域有各自特定的一些领域常量和处理方法,ICSL语言也提供了面向领域特征的内置常量和内置函数的支持;考虑到为了适应用户需求的不断变化,提供一种简单的用户扩充机制也是ICSL语言的关键,所以,ICSL语言还提供了多个侧面的扩充机制。图1为ICSL语言系统结构图。

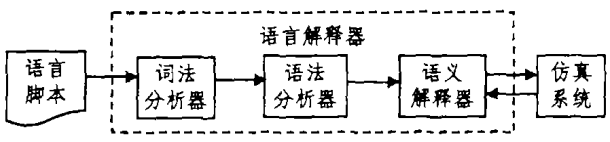

图1 ICSL语言系统结构图

胡圣明 博士研究生,主要研究领域为程序设计语言,面向对象技术和软件体系结构,逆向工程。李青山 博士,副教授,主要研究方向为逆向工程,程序理解,面向对象和软件体系结构。陈 平 博士,教授,博士生导师,主要研究领域为电子信息系统软件开发理论与技术,面向对象技术,软件工程,逆向工程。褚 华 博士研究生,主要研究领域为逆向工程,程序理解,面向对象技术和软件体系结构。

2.1 ICSL 的词法

作为一种语言，ICSL 有其自身的一套标记符号。表1给出了 ICSL 语言中部分记号的正则表达式。为方便定义，首先定义三个辅助记号以便后续的定义引用：letter＝[a-z A-Z]，digit＝[0-9]，dhex＝[0-9a-f A-F]

表1 ICSL 语言的记号定义

类型	类型标识	正则表达式
关键字与运算符	……	If，float，while，＋，｛，[，……
标识符	ID	ETTER(LETTER|DIGIT)*
字符常量	C_CHAR	"[^ \"]*"
十进制整形常量	C_DINT	(DIGIT)+
十六进制整形常量	C_HINT	0(X|X)(DHEX)+
浮点常量	C_DOUBLE	(DIGIT)+.(DIGIT)*
时间常量	C_TIME	(C_DINT:)?(C_DINT:C_DINT)…

本文在设计 ICSL 的词法时充分考虑语言的扩展性和解释效率，将不易变化的部分采用正则表达式描述，并利用词法分析工具自动生成词法分析器，以在扫描完单词后便得到记号的类型，这样提高了其分析效率；而对于需要扩展的部分，则将其归入标识符，再由识别得到的标识符串判断其记号类型，以获得较高的扩展性。

2.2 ICSL 的语法

ICSL 语言的语法定义了各种数学表达式，控制语句及对象操作。表达式包括常量和变量表达式、函数调用、算术表达式、布尔表达式、时间表达式等，常量表达式中含有 TRUE、FALSE、PI 等基本常量、Hostile(表明实体是敌方)，FixWing(表明实体是固定型机翼飞机)等实体特征常量以及各种环境因素常量；语句则包含了表达式语句、变量定义语句、赋值语句、条件语句、循环语句及时间语句等，如条件语句 if(condition) then do statement_list else statement_list endif 和时间语句 At(expression) do statement_list enddo。下面列出了 ICSL 语言文法的产生式集合中部分产生式的描述。

```
statement_list→statement;statement_list
        |statement;
statement→expression_statement|if_statement
        |declaration_statement
        |while_statement|time_statement
        |for_statement
if_statement→IF condition THEN statement_list ENDIF
        |IF condition THEN statement_list ELSE statement_list EN-
        DIF
expression→identifier DOT identifier
        |identifier DOT function
```

2.3 ICSL 的语义

ICSL 的语义可以分为两个层次：领域无关语义和领域相关语义。领域无关语义是指语言本身所固有的，不局限于任何一种或者几种特定领域的语义，这种语义主要表现为各种表达式的计算及控制语句的解释，是一种数学计算行为；领域相关语义则是和某一具体领域相关联，表示领域内各种行为的含义。

ICSL 的领域无关语义与现在各种应用广泛的语言是相同的，如 C/C++，Java 等。本文主要介绍 ICSL 的领域相关语义。ICSL 的领域相关语义集中表现为语言的核心对象，这些核心对象在不同的仿真环境中，可具有不同的含义、不同的属性以及不同的方法。

核心对象属性是为了保持对象的状态，但是核心对象属性所存储的信息在仿真系统中是一直存在的，而且随时间不断变化，为保证数据一致性，核心对象并不真正存储这些信息，而是在需要的时候从仿真系统中获得。事实上，核心对象的属性只是逻辑上的属性，它并没有真正的物理存储空间。对象的虚拟属性对语言的使用者来说是透明的，这一机制也使得动态加载对象属性成为可能。

用户操作核心对象是希望从仿真系统中获取信息，或者使仿真系统的状态发生变化。ICSL 语言将用户的操作转化为 ICSL 内部的语义编码，如使用 MSG_ENTITY_SetReqSpeed 来表示用户希望改变当前实体的速度，然后 ICSL 语言解释器将语义编码发送到仿真系统中，由仿真系统对编码作出响应。

通过语义编码，使得核心对象属性和方法的处理变得统一；不同的仿真系统可以对语义编码做出不同的解释，即一套编码可以用于各种不同仿真系统之上；仿真系统自身的演化也不会对语言造成影响；即便在一个仿真系统内部，不同类型的实体也可对同一语义编码做出不同的响应，充分实现了语义编码的可重用性。

3 ICSL 的扩充性

扩充机制是 ICSL 语言最重要的特点。除基本类型、运算和控制外，ICSL 语言的其它成分都是可扩充的，包括：内置函数和内置常量、核心对象、核心对象属性和方法以及语义编码。在 ICSL 语言系统中，扩充机制是通过元数据控制策略来实现的。元数据在程序中不是被加工的对象，而是被用来对程序的运行起控制作用，并且可以通过值的改变而改变程序的行为[2]。ICSL 元数据文件存储了语言中可扩充部分的元数据，通过对元数据的修改实现语言的扩充。图2为 ICSL 语言一个应用实例的元数据描述文件。

```
〈?xml version＝"1.0"encoding＝'GB2312"?〉
－〈ScriptContaint〉
－〈CoreObject〉
〈!--gloab 核心对象--〉
＋〈Object name＝"gloab"〉
〈!--entity 核心对象--〉
－〈Object name＝"entity"〉
 －〈DataList〉
 〈!--type：数据成员的类型，name：名称，pro：读写类型，R：只读，
 ……--〉
 〈!--实体名称--〉
 〈DATA type＝"string"name＝"name"pro＝"R"/〉
```

图2 ICSL 元数据

其中 gloab 和 entity 表示 ICSL 能够支持的核心对象，DataList 标签中描述的是 Entity 核心对象所具有的属性以及属性的类型、名称和读写标志。元数据文件还描述了内置函数、内置常量等信息。

3.1 内置函数和内置常量的扩充

在使用 ICSL 语言编写代码的过程中，可以借用大量已有的库函数，增加新的库函数只需要用户提供函数的接口描述和对应的目标代码，接口描述是为了让 ICSL 系统能够识别此函数，而目标代码则是为了在运行过程中能够正确地调用该函数。各个领域内有很多的常用数值，这些数值很少发生变化，ICSL 语言提供了内置常量来支持，内置常量的扩充只需要对相应的内置常量表加以修改即可。对内置函数的扩充，用户需要修改元数据描述文件并重新进行链接操作，而对内置常量的扩充只需修改元数据描述文件即可。

3.2 核心对象的扩充

核心对象的扩充分为两个方面：已存在核心对象的修改

和增加新核心对象。核心对象的属性和方法是基于语言内部的领域语义编码，核心对象属性和方法的操作被统一翻译成内部语义编码，所以，增加一个新的属性和新的方法需要用户给出属性和方法的元数据描述，如属性的名称、方法的原型等，然后指定属性或方法到语义编码的映射即可。

ICSL 语言允许用户自定义核心对象，所有的核心对象类都继承一个公共的基类，这个公共的基类已实现了核心对象的所有代码，所以新增一个核心对象只需要指明对象初始化时所需加载的属性表、方法表以及属性、方法到语义编码的映射表。

3.3 领域语义编码的扩充

领域语义编码是 ICSL 语言的核心，它抽象了用户使用脚本语言的意图。ICSL 语言定义了领域语义的编码，但并不提供对应的语义动作，语义动作由 ICSL 语言所控制的仿真系统完成。如果现有的语义编码无法表示新的语义动作，或者完全不符合用户要求，那么用户可在语义编码表中增加新的编码或者完全替换现有的各种编码，从而完成语义的扩充。正是这一点可让 ICSL 语言应用于各种仿真领域。

4 实验研究

ICSL 语言的词法分析器和语法分析器已利用 XDFLEX 和 XDYACC 编译工具实现，记号分类器、语言语义解释器、和领域语义映射器已采用 C++语言实现。下面给出了 ICSL 语言系统应用在一个战场仿真环境的样例，脚本的部分代码如图3所示。

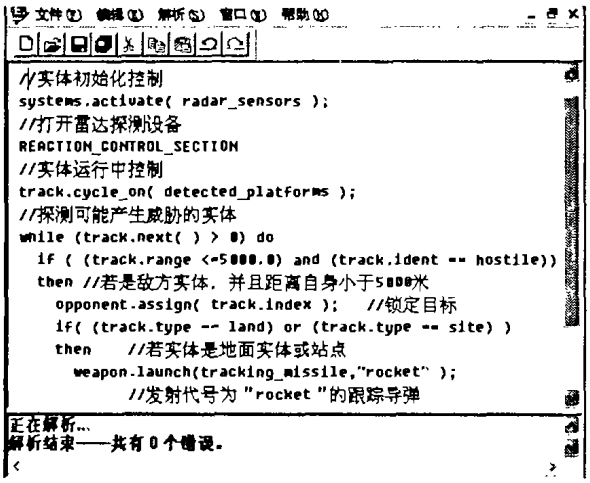

图3 ICSL 应用实例

下面是仿真系统中对语义编码响应的伪代码：

```
switch(语义编码)
        case 打开雷达设备：打开雷达设备 break;
        case 探测：探测特定类型实体 break;
        ……
        case 发射：发射指定武器 break;
        default：break;
```

这段语义响应代码可以位于仿真系统中实体的类层次树中不同层次，父类可完成子类共同的语义动作，子类可实现自身特殊的语义动作，并且子类可覆盖父类的响应行为，充分地实现了语义行为和语义编码的重用。

结束语 本文针对智能化控制，研制了 ICSL 脚本语言，利用 XDFLEX 和 XDYACC 等工具实现了 ICSL 语言的解释器。ICSL 语言系统的特点在于：1、针对仿真领域特征，对领域语义进行编码，并提供语义的扩充机制。2、通过元数据的描述，提供了 ICSL 语言的词法和语法扩充机制。3、语义的定义与实现相分离的结构实现了 ICSL 语言在仿真平台上的移植性。

下一步工作的重点在于提高 ICSL 语言的可定制性，虽然 ICSL 语言提供了很强的扩充机制，但是核心的扩充是在源代码级别，如何实现用户只要修改语言元数据而无需重新编译便可运行 ICSL 语言系统将是一个需着重解决的问题，这对用户自定制语言有着重大的意义。

参 考 文 献

1 李青山，胡圣明，李玉萍、ICSL 技术规格说明书. 西安电子科技大学，2004

2 李青山，陈平，褚华、支持柔性机制的元数据模型的研究与应用、西安电子科技大学学报，2002，29(31)：319～323

3 Bushey D E，Malloy B A、A Study of Dynamic Traffic Re-Routing in the National Airspace System、In：Proc. of the IEEE Annual Simulation Symposium，1997. 104～113

4 Keane J S，Rozenblit J W，Barnes M. The Advanced Battlefield Architecture for Tactical Information Selection (ABATIS)、In：Proc. of the 1997 Workshop on Engineering of Computer-Based Systems (ECBS '97)，March 1997、228

5 Jayaraman B、Semantics of EqL. IEEE Trans on Software Engineering，1988，14(4)：472～480

6 Kewley，Hargreaves R Jr、Computational intelligence for support of military tactical decision making：[Ph. D. Thesis]、Rensselaer Polytechnic Institute. Dec. 2000

7 Patel A J. OBSTAL：A Language with Objects，Subtyping，and Classes：[PhD. Thesis]、Stamfprd Standford University. Dec. 2001

8 Virtual Prototypes、STAGE User's Guide. June 2001

9 Subrahmanyam P A，Singh K J，Story G，et al、Quality Assurance in Scripting、IEEE Multimedia，1995，2(2)：50～59