

机器人世界杯救援模拟器手册

2021 年 9 月 27 日，RoboCup 救援
模拟团队 1.5 版

目录

1.目的	1
2.安装	1
2.1.软件需求	1
2.2.下载	1
2.3.目录	1
2.4.编译	2
2.5.例子	2
3.运行	2
3.1.预先计算模式	2
3.2.正常模式	3
3.3.参数选项	3
4.工具	3
4.1.映射编辑器	4
4.2.场景编辑器	4
4.3.随机场景	4
4.4.日志查看器	4
4.5.osm2gml	5
5.模拟器的设计	5
5.1.体系结构	5
5.2.实体	6
5.2.1.封锁	7
5.2.2.区域	7
5.2.3.建筑	8
5.2.4.道路	9
5.2.5.人类	9
5.2.6.平民	10
5.3.代理	10
5.3.1.排代理	10
5.3.2.中心代理	11
5.4.感知和命令	11
5.5.沟通	12
参考文献	12

1.目的

本手册有助于了解机器人世界杯救援模拟服务器，并帮助有兴趣参加机器人世界杯救援代理模拟比赛的人。

2.安装

本手册假设模拟器将在 Linux 机器上运行，即使可以在 Linux 机器上运行它们

微软 Windows 或苹果 macOS。我们建议使用 Linux，因为它是开源的，大多数发行版都得到了用户社区的良好支持。如果你以前从未使用过 Linux，但打算使用，我们建议从用户友好的发行版开始，比如 [Ubuntu](#) 或 [Fedora](#)。

2.1.软件需求

- [Java OpenJDK 11+](#)
- [Git](#)
- [Gradle](#)
- 实用工具如 `wget`, `bash`, `xterm`, `tar`, `gzip` 等。

注意:如果你使用的是 Ubuntu，所有这些实用程序都在默认的软件库中。

2.2.下载

您可以通过克隆 <https://github.com/roborescue/rcrs-服务器存储库>来下载模拟服务器。使用命令克隆模拟服务器

```
Git 克隆 https://github.com/roborescue/rcrs-server.git
```

2.3.目录

rcrs-server 包含多个目录。服务器的重要目录有:

- `boot`:运行模拟服务器的脚本。 `boot/config/`:默认配置文件。 `boot/logs/`:文本日志文件
 - `boot/records/`: JSON 日志文件
- `build/`:编译过的 Java 类
- `docs/`:文档
- `jars/`:模拟器 JAR 文件
- `lib/`:外部库
- `maps/`:地图和场景
- `modules/`:模拟器源代码
- `oldsims/`:旧模拟器源代码

2.4.编译

必须在您的机器上编译模拟服务器，以生成运行模拟器所需的 JAR 文件。执行下面的步骤来编译模拟服务器。

- 1.切换到目录 rcrs-server
 - a.如果是 macOS，使用命令修补 boot/functions.sh 文件

```
Sed -i -e "/readlink/s/^/#/" boot/functions.sh
```
- 2.使用 ./gradlew completeBuild 命令编译模拟器
- 3.检查安装结束时的消息。如果安装成功完成，则显示 BUILD SUCCESSFUL 信息;否则显示“BUILD FAILED”。

2.5.例子

在 Ubuntu 中，按照以下命令进行安装。*Ubuntu 上的安装*

```
$ git 克隆 https://github.com/roborescue/rcrs-server.git $
CD rcrs-server
$ ./gradlew completeBuild
```

安装成功后会显示如下信息。

安装完成

```
BUILD SUCCESSFUL in 2s 1 个
可操作的任务:1 个已执行
```

3.运行

模拟服务器有两种执行模式:预计算和正常。

3.1.预先计算模式

在预计算模式下，模拟器连接每种类型的一个代理，并允许它们写入计算结果。

在预计算模式下运行模拟服务器的命令顺序为:

在预计算模式下运行模拟服务器

```
$ CD rcrs-server
$ CD boot
```

运行模拟服务器后，运行代理。预计算完成后，按下 *Control-C*，输入 sh kill.sh，停止运行的模拟服务器。

有道文档翻译
pdf.youdao.com

在预计算模式下停止运行模拟服务器

```
Control-C
$ bash kill.sh
```

3.2.正常模式

在正常模式下，模拟器连接场景中定义的所有代理，并允许它们使用预计算输出。

在正常模式下运行模拟服务器的命令序列是：

在正常模式下运行仿真服务器

```
$ CD rcrs-server
$ CD boot
$ bash start-
```

运行模拟服务器后，运行代理。一旦模拟完成，服务器就会自动停止。

3.3.参数选项

可以使用以下参数运行模拟服务器：

- m MAPDIR 或——map MAPDIR，其中 MAPDIR 是要运行的映射所在目录的路径(默认为../maps/gml/test/map)。
- c CONFIGDIR 或——config CONFIGDIR，其中 CONFIGDIR 是包含与映射相关的配置的目录(默认为../maps/gml/test/config)。
- l LOGDIR 或——log LOGDIR，其中 LOGDIR 为日志文件存放的目录(默认为。/logs)。

这些参数可以用于运行预计算和正常的模拟。必须为 MAPDIR 和 CONFIGDIR 使用相同的参数才能在预计算和正常模式下运行模拟服务器。如何使用这些参数运行模拟服务器的示例如下：

运行带有选项的模拟服务器

```
$ bash start-precompute.sh -m ../maps/gml/kobe/map -l logs2
(完成预计算后)
control - c
$ bash kill.sh
```

4.工具

模拟服务器附带了几个工具来支持新地图和场景的开发以及模拟的重播。

4.1.映射编辑器

打开终端窗口，导航到 rcrs-server 根目录并执行

```
$ ./gradlew gmlEditor --args=<映射文件路
```

其中 --args=<地图文件路径>是可选的。
例如，

```
$ ./gradlew gmlEditor

$ ./gradlew gmlEditor --
```

4.2.场景编辑器

打开终端窗口，导航到 rcrs-server 根目录并执行

```
$ ./gradlew scenarioEditor --args=<场景路径>
```

其中 --args=<场景路径>是可选的。例如，

```
$ ./gradlew scenarioEditor

$ ./gradlew scenarioEditor --
```

4.3.随机场景

打开终端窗口，导航到 rcrs-server 根目录并执行

```
$ ./gradlew randomScenario --args=<映射路
```

其中 --args=<映射路径>是可选的。
例如，

```
$ ./gradlew randomScenario

$ ./gradlew randomScenario --args="../maps/gml/test/map -civ 10 100 -fb 1 50 -fs
0 1 -pf 1 50 -po 0 1 -at 1 50 -ac 0 1 -refuge 1 2 -fire 0 1"
```

4.4.日志查看器

打开一个终端窗口，导航到 rcrs-server 根目录并执行

有道文档翻译
pdf.youdao.com


```
$ ./gradlew logViewer --args='-c config/logViewer.cfg <日志路
```

其中 `--args='-c config/logviewer.cfg <log path>'` 是可选的，`<log path>` 定义了日志文件的路径。默认日志文件路径为 `logs/rescue.log`。

4.5. osm2gml

打开终端窗口，导航到 `rcrs-server` 根目录并执行

```
$ ./gradlew osm2gml --args='<osm 映射路径> <gml 映射路
```

`<osm 映射路径>` 是 `osm` 映射文件的路径，`<gml 映射路径>` 是目标 `gml` 映射路径。

5.模拟器的设计

本节描述模拟器体系结构及其主要特征，以及组成模拟场景的实体。

5.1.体系结构

RoboCup 救援模拟服务器由 [表 1](#) 中列出的几个模拟器组成。

表 1。模拟器列表

模 拟 器	描 述
清晰 的	管理封锁移除
崩溃	管理建筑的结构伤害和封锁创造
点火	在模拟过程中随机点燃建筑物上的火
火	控制火势在建筑物间蔓延并扑灭
交通	管理人员移动
杂项	管理人员伤害和掩埋

这些模拟器建立与 *内核模拟器* 的连接，负责协调模拟器的进程并集中它们生成的数据 (Skinner & Ramchurn, 2010)。这些连接如图 1 所示。

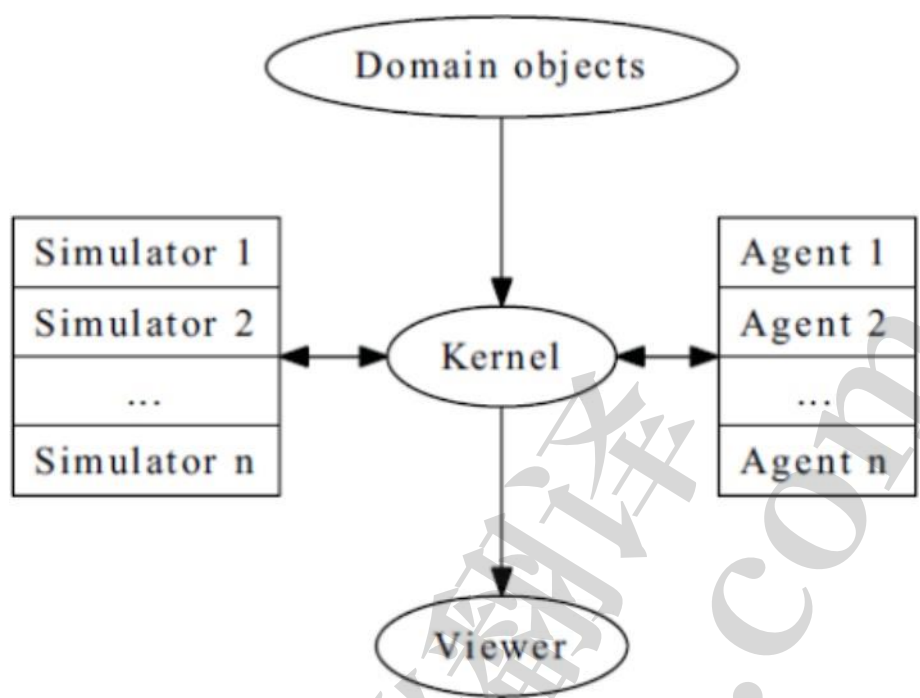


图1所示。RoboCup Rescue Agent Simulation 平台架构

RoboCup 救援模拟器旨在创建一个部分可观察的、离散时间的、动态的、随机的、多智能体的环境。换句话说，在这个环境中：

- 无法通过单个智能体的感知来了解完整的世界当前状态(即使智能体拥有无限的视野范围，它仍然无法透过建筑物的墙壁看到);
- 时间是间隔分割的，与连续时间相反;
- 有随机元素影响其状态转换;
- 存在多个代理，一个代理的行为可能会干扰其他代理的表现。

时间被划分为时间步长;在每个时间步中，智能体感知环境，并对其将要执行的动作进行推理。在每个时间步中，会发生以下情况:。内核更新所有代理的感知(视觉和通信)并等待代理的命令。。智能体更新它们的世界模型并做出决策，将它们的命令发送给内核。。内核将智能体的命令发送给模拟器。仿真器处理代理的命令，并将环境所遭受的变化发送回内核。内核将环境的变化发送给查看器。

5.2.实体

几个对象在模拟器中表示，如图2所示。

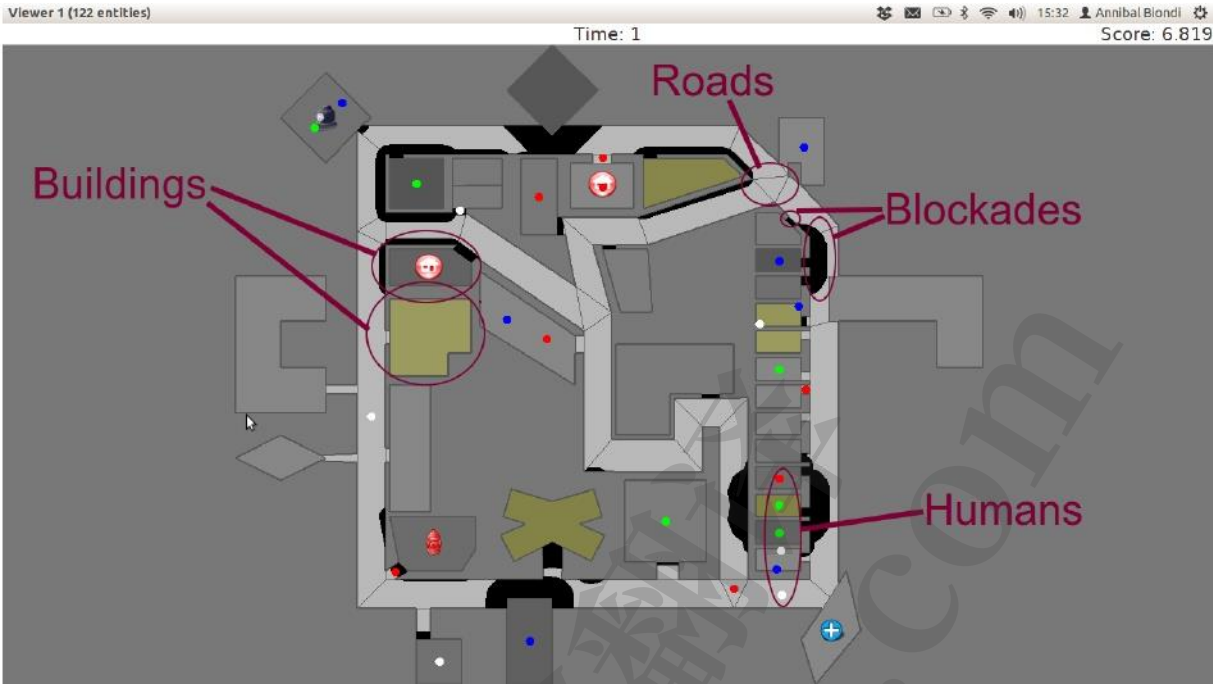


图2。模拟器的实体

5.2.1.封锁

封锁阻碍了特工和平民的前进道路;它们在道路上被表示为黑色多边形。封锁出现在模拟的开始，在这之后就不会产生。它们必须由警察部队移除。

构成封锁的财产有：

位置修复成	封锁所在道路的 ID 需要将封锁从道路上完全移除一个
本形状	矩形，该矩形围绕封锁质心的整个封锁坐标
X&Y 顶点	包含封锁线顶点的向量

5.2.2.区域

区域实体代表建筑和道路。构成区域的属性有：

封锁	一个包含该区域内封锁的列表
边缘	带有限制区域的边的列表

邻居 从这个区域可以访问的区域列表

用户 表示地图中区域的坐标

虽然**建筑**和**道路**都有封锁属性，但封锁只在**道路**中有效。

5.2.3.建筑

建筑代表模拟器中的各种建筑。有一些特殊类型的建筑，比如**避难所**、**救护中心**、**消防站**和**警察局**，它们不会着火。[表 2](#) 显示了这些特殊建筑的说明，它们将在本文件后面的部分进行描述。

表 2。特别的建筑

避 难 所	救 护 中 心
消 防 局	警 察 局

构成建筑物的物业包括：

破碎

fieryness 建筑结构受损程度如何;在模拟过程中，建筑物的火灾强度和火灾相关的破坏是否会发生变化

- unburned*-完全不被烧毁
- WATER_DAMAGE*-完全没有被烧毁，但是有水伤害
- 加热*-有点着火
- BURNING*-着火了一点
- INFERNO*-经常着火
- MINOR_DAMAGE*-熄灭但有轻微伤害
- 温和伤害*-熄灭，但中等伤害
- SEVERE_DAMAGE*-熄灭但严重伤害
- BURNT_OUT* -完全烧毁

地板

建筑的楼层数

地面面积

每层的面积

点火

表示模拟器是否点燃了该建筑物(注 :建筑物可以通过模拟器点燃或靠近燃烧的建筑物而着火;如果该建筑在模拟的某个时刻被模拟器点燃，则点火将被设置为“1”。)

有道文档翻译
pdf.youdao.com

重要性	未知函数;对所有建筑具有同等价值
温度	建筑物的温度;超过一定阈值，建筑物就会着火
总面积	建筑总面积(楼层 x 地面面积)

常规建筑用各种颜色的多边形表示，根据其状态不同，如图 3 所示;颜色越深，表示结构火灾或水损害越大。

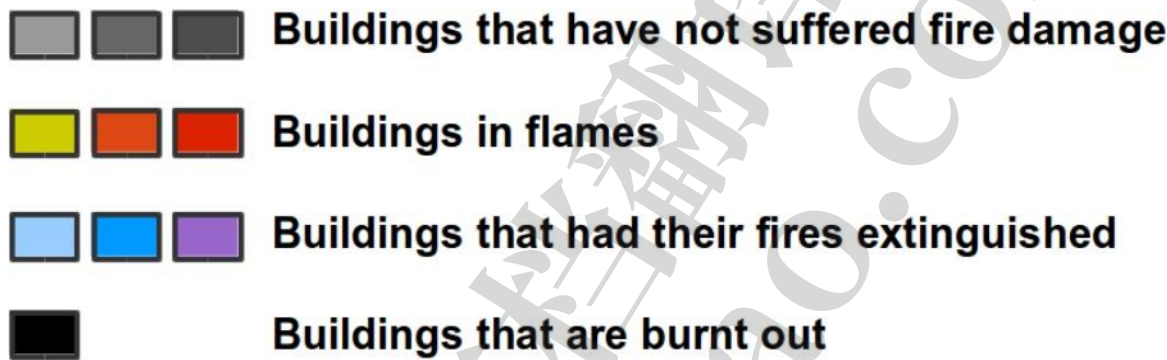


图3。常规建筑的可能状态

在模拟的开始阶段，破碎的建筑将人类困在废墟下;这些碎片必须由急救队移走，然后他们继续营救人类。

避难所是一种特殊的建筑:它代表着一个支持救援活动的地方，为伤员提供医疗服务，为消防队提供水。在模拟器中，避难所内的人类受到的伤害为零，这意味着他们在那里逗留期间不会损失生命值;然而，当人类实体离开避难所时，伤害会恢复。

此外，消防队在避难所内的每个周期都会补充一定数量的水。

5.2.4.道路

表示道路的区域实体除了区域实体的属性外没有新的属性。

5.2.5.人类

这些是代表人类的实体。在模拟器中，它们可以是平民、急救队、消防队或警察部队。他们都用不同颜色的圆圈表示，如果他们死了或被埋了，他们就不能自己移动。

构成人的属性有:

buriedness	人被埋得有多深
损害	人类每周期损失多少 HP;到达避难所时为零

方向	人类移动的方向(推断);y 轴正半为零，数值逆时针递增至 129599((360 * 60 * 60)- 1) 秒
惠普的 位置	人的生命值点;达到 0，则人类死亡 人类所在实体的 ID;可能是区域实体或人实体(如果它在救护车内)
历史地位	人类在上一个周期中经过的实体的列表，按时间顺序
耐力	没有实现;会在代理每次采取动作时减少，并在每个周期开始时部分补充 (未知)
旅行距离 X&Y	地图中代表人类的坐标

模拟器中每个人的颜色由其类型和生命值来定义:生命值越低，颜色越深。死去的人类用黑色来表示。

5.2.6.平民

平民是人的实体，他们不是救援队的一部分;他们用绿色来代表。他们的标准行为是，如果他们没有受伤或被埋，他们会走到最近的避难所;否则，他们将不得不由**急救队**运送。

5.3.代理

这些实体将组成你的救援队;换句话说，这就是你要编程的东西。agent 分为两种类型:**排代理**([第 5.3.1 节](#)) 和**中心代理**([第 5.3.2 节](#))。

5.3.1.排代理

排代理能够通过感知并在其上执行动作与模拟环境进行交互。它们还可以通过语音或无线电通信与其他智能体交换信息。他们由三个不同的类别组成:**急救队**、**消防队**和**警察部队**。

救护车的团队	负责将被救出的受害者运送到避难所。
消防队	负责挖掘(即营救)受害者和扑灭建筑物上的火灾。他们在水箱中携带一定量的水，可以在避难所补充。

警察部队 负责清除道路上的路障。接到命令后，他们会在每个周期从目标路障上清除一定数量的障碍物，这在修复成本参数中有规定。然而，与**急救队**和**消防队**不同的是，让两个**警察部队**对同一个封锁行动不会给这个过程带来任何好处:就好像只有一个**警察部队**在行动一样。

5.3.2.中心代理

中心代理是一种代理，它与世界的唯一交互是通过无线电通信。中心代理有三种类型:**救护车中心** (Ambulance Centres)、**消防站** (Fire Stations)和**警察局** (Police office)，它们被表示为建筑物。

5.4.感知和命令

模拟器有两种感知模式:**标准模式**和**视线模式**。

视线感知模拟智能体的视觉感知:定义一个视觉范围和若干射线，智能体感知这些射线到达的任何东西。

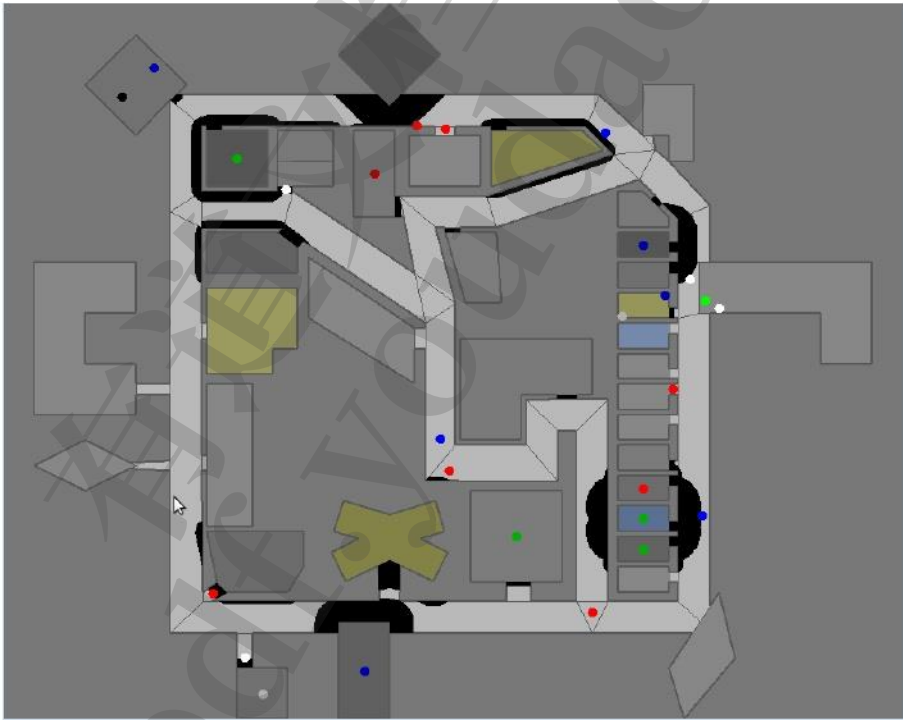


图4。No Light of Sight

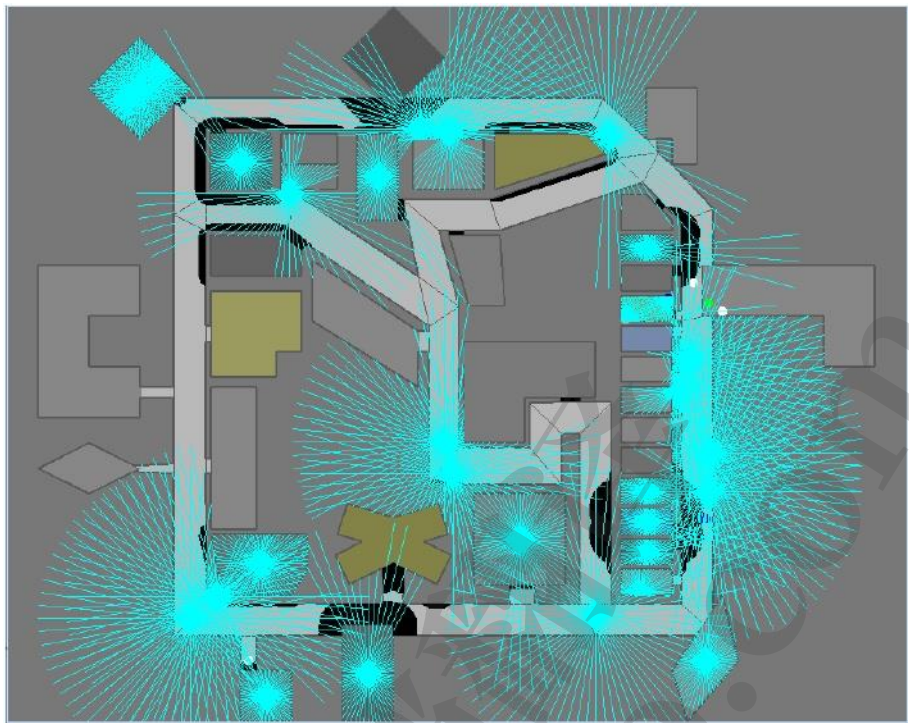


图5。包括 *Light of Sight*

代理的当前可见实体集存储在名为 *ChangeSet* 的结构中;出现在其中的实体会在其世界模型中自动更新;也就是说,如果一个智能体感知到一个它之前不知道的封锁,这个封锁就会自动添加到它的世界模型中。相反,虽然不会发生:如果智能体不再感知到封锁,它的世界模型中就不会发生任何变化,即使它之前知道那里有封锁。在这种情况下,智能体仍然会认为那条路上有封锁,即使这样的封锁已经被清除了。因此,智能体需要弄清楚这一点,并相应地修改它的世界模型。

5.5.沟通

在模拟器中有两种通信形式:*直接通信*和*无线电通信*。直接通信,用命令*说话*完成,是发射体代理半径内人类听得见的通信,就好像发射体在喊什么。

无线电通信是用命令*说*完成的,并将信息传输给所有签约到广播频道的代理。无线电通信信道的数量是有限的,每个信道的带宽都是有限的。

在这两种通信方式中,信息在发送之前都必须被编码成一串字节;受体在接收到信息后,必须对其进行解码。这两种类型的受体都可能容易受到信息缺失的影响,即信息没有被其目标接收到;无线电通信也容易出现信息失败,即接收到的信息为空。

参考文献

Skinner, C., & ramchant, S.(2010)。RoboCup 救援模拟平台。第九届自主代理和多代理系统国际会议论文集:第1卷-第1卷, 1647-1648。