**仿真比赛平台编译运行说明**

仿真比赛平台cserver-beta-2.02

更新日期：2013-03-09

【配置需求】

系统：Ubuntu 12.04.2 LTS

编译器：gcc 4.6.3

依赖：

Boost 1.48（或以上）[[1]](#footnote-1)

CMake 2.6.0（或以上）[[2]](#footnote-2)

iclingo 3.0.5[[3]](#footnote-3)

【编译】

假设平台源码根目录为${PLANNER\_PATH}，

在${PLANNER\_PATH}下输入命令：

mkdir build

cd build

cmake ..

make

编译成功后，在${PLANNER\_PATH}/bin下能找到平台运行程序cserver和客户端样例程序example，在${PLANNER\_PATH}/lib下能找到客户端程序开发所需的静态库libframe.a和libutility.a，以及平台评分所需的运行时库libasp.so。

【运行】

进入${PLANNER\_PATH}/bin运行runserver.sh即可启动比赛平台。如需使用更多的平台运行参数，请运行命令“./cserver -help”，查看平台运行参数。

【开发】

客户端的开发主要继承${PLANNER\_PATH}/include/cserver/plug.hpp文件中的Plug类。

主要需要实现的函数是*void Plan()*，该函数内需要补充的内容为：在限定的时间内（默认5秒），跟平台交互（执行动作、获得反馈信息），以完成测试给出的任务，在测试给出的环境描述情景下。

规划所需的环境描述可以通过函数*const std::string Plug::GetEnvDes() const*获得；

规划需要完成的任务描述可以通过函数*const std::string& GetTaskDes() const*获得；

规划执行的动作可以调用对应的动作函数，如*bool ToPlate(unsigned int a)*，表示执行动作*ToPlate*，将编号为*a*的物体放到机器人上的盘子里，返回动作是否执行成功。

${PLANNER\_PATH}/example下给出开发客户端程序的一个例子。自己开发客户端可以重写example下的例子，或独立开发。

【测试数据格式】

${PLANNER\_PATH}/tests/example目录下存放了测试集的样例。test.xml为测试描述文件的一个例子，test.list内列出该目录下所有被管理的测试描述文件。测试描述文件在test.list内的排列次序代表测试相应的编号（从编号1开始）。

测试描述文件主体包含三部分：

env环境描述部分；

instr指令式任务描述部分；

nl自然语言式任务描述部分。

环境描述部分又分三部分：

info基本场景描述信息；

mis缺失场景描述信息；

err错误场景描述信息。

env下三个属性mis，err，ans分别代表平台给客户端程序发送场景描述信息时，是否缺失信息，是否带错误信息，是否在回答客户端程序查询动作（目前为AskLoc动作）时，做出误导回答。

err下包含正确信息和错误信息，分别在r标签和w标签下。当err属性为on时，w下的信息将取代r下的信息发给客户端程序。

因此，正确完整的场景描述信息 = info + mis + err.r

1. http://www.boost.org [↑](#footnote-ref-1)
2. http://www.cmake.org [↑](#footnote-ref-2)
3. http://sourceforge.net/projects/potassco/files/iclingo/3.0.5 [↑](#footnote-ref-3)