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]

CMake 2.6.0（或以上）[2]

iclingo 3.0.5[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

[1] http://www.boost.org

[2] http://www.cmake.org

[3] http://sourceforge.net/projects/potassco/files/iclingo/3.0.5

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