

# How to Develop A Client

Ke Shi

Multi-Agent Systems Lab.  
Department of Computer Science and Technology  
University of Science and Technology of China



Oct. 11, 2008

## 1 Base Client

- 连接server
- 从server接收来的信息
- 向server发送的信息
- Client结构

## 2 UvA Source

- SoccerServer 是一个允许竞赛者使用各种编程语言设计球员，进行仿真足球比赛的系统。Client与Sever之间仅通过UDP/IP协议通讯，这样只要编程语言提供了对UDP 连接的支持，就可以用来开发球队。

- SoccerServer 是一个允许竞赛者使用各种编程语言设计球员，进行仿真足球比赛的系统。Client与Server之间仅通过UDP/IP协议通讯，这样只要编程语言提供了对UDP 连接的支持，就可以用来开发球队。
- Server提供一个虚拟场地，对比赛中足球和所有球员的运动进行仿真，根据比赛规则以离散的方式控制比赛的进行。

- SoccerServer 是一个允许竞赛者使用各种编程语言设计球员，进行仿真足球比赛的系统。Client与Server之间仅通过UDP/IP协议通讯，这样只要编程语言提供了对UDP 连接的支持，就可以用来开发球队。
- Server提供一个虚拟场地，对比赛中足球和所有球员的运动进行仿真，根据比赛规则以离散的方式控制比赛的进行。
- Client相当于球员的大脑，通过UDP协议向Server发送命令来控制球员的行为。

- 发送: (init TeamName [(version VerNum)] [(goalie)])

- 发送: (init TeamName [(version VerNum)] [(goalie)])
- 接收: (init Side UniformNumber PlayMode)

- 发送: (init TeamName [(version VerNum)] [(goalie)])
- 接收: (init Side UniformNumber PlayMode)
- 出错提示: (error no\_more\_team\_or\_player\_or\_goalie)



- 发送: (init TeamName [(version VerNum)] [(goalie)])
- 接收: (init Side UniformNumber PlayMode)
- 出错提示: (error no\_more\_team\_or\_player\_or\_goalie)
- 断开连接: (bye)

- 球场中，一个球员只有了解周围的环境，才能据此作出各种决策，实现有效的比赛。在仿真平台中，球场上所有物体的运动都是由server进行仿真的，因此，在比赛过程中，为了获得场上情况，球员需要从server接收各种信息。

- 球场中，一个球员只有了解周围的环境，才能据此作出各种决策，实现有效的比赛。在仿真平台中，球场上所有物体的运动都是由server进行仿真的，因此，在比赛过程中，为了获得场上情况，球员需要从server接收各种信息。
- 球员从server获得的信息主要有两种，一种是对球员命令的响应信息，通知球员命令是否正确执行等，这类消息与球员命令结合在一起介绍；另一种是球员的感知信息，分为视觉信息，身体感知信息和听觉信息，所有感知信息都包含一个时间标签，用来表示server发送信息的时间（仿真周期数）。

- 视觉信息是球员最重要的感知信息，它包含了球员可以看到的物体的信息。视觉信息的主要特点有：

- 视觉信息是球员最重要的感知信息，它包含了球员可以看到的物体的信息。视觉信息的主要特点有：
  - 视觉信息中的数据都是相对于观察球员的，而不是场上的绝对信息，具有相对性。

- 视觉信息是球员最重要的感知信息，它包含了球员可以看到的物体的信息。视觉信息的主要特点有：
  - 视觉信息中的数据都是相对于观察球员的，而不是场上的绝对信息，具有相对性。
  - 视觉信息获得的不是场上所有物体的信息，而只是在球员视野中的物体的信息，具有局部性。

- 视觉信息是球员最重要的感知信息，它包含了球员可以看到的物体的信息。视觉信息的主要特点有：
  - 视觉信息中的数据都是相对于观察球员的，而不是场上的绝对信息，具有相对性。
  - 视觉信息获得的不是场上所有物体的信息，而只是在球员视野中的物体的信息，具有局部性。
  - 视觉信息中物体的信息量受到与球员距离等因素的影响，具有不完整性。

- 视觉信息是球员最重要的感知信息，它包含了球员可以看到的物体的信息。视觉信息的主要特点有：
  - 视觉信息中的数据都是相对于观察球员的，而不是场上的绝对信息，具有相对性。
  - 视觉信息获得的不是场上所有物体的信息，而只是在球员视野中的物体的信息，具有局部性。
  - 视觉信息中物体的信息量受到与球员距离等因素的影响，具有不完整性。
  - 视觉信息中的数据存在误差，具有不精确性。



- 视觉信息是球员最重要的感知信息，它包含了球员可以看到的物体的信息。视觉信息的主要特点有：
  - 视觉信息中的数据都是相对于观察球员的，而不是场上的绝对信息，具有相对性。
  - 视觉信息获得的不是场上所有物体的信息，而只是在球员视野中的物体的信息，具有局部性。
  - 视觉信息中物体的信息量受到与球员距离等因素的影响，具有不完整性。
  - 视觉信息中的数据存在误差，具有不精确性。
- 视觉信息的格式为：(see Time ObjInfo+)

- 听觉感知用来接收其他球员或者教练通过say命令发送的消息，裁判广播的信息也作为球员的听觉信息接收。仿真平台中的声音信息仿真的一个拥挤的低带宽环境，双方所有球员共用的是一个不可靠的信道。

- 听觉感知用来接收其他球员或者教练通过say命令发送的消息，裁判广播的信息也作为球员的听觉信息接收。仿真平台中的声音信息仿真的一个拥挤的低带宽环境，双方所有球员共用的是一个不可靠的信道。
- 听觉感知信息的格式为：(hear Time Sender Message)

- 身体感知信息包含了球员自身的各种状态信息。

- 身体感知信息包含了球员自身的各种状态信息。
- (sens\_body (view\_mode high—low narrow—normal—wide) (stamina Stamina Effort)(speed Speed Angle)(head\_angel Angle)(kick Count)(dash Count)(turn Count)(say Count)(turn\_neck Count)(catch Count)(move Count)(change\_view Count)(arm (movable <MOVABLE>) (expires <EXPIRES>)) (target <DIST><DIR>) (count <COUNT>))(focus (target l—r <UNUM>) (count <COUNT>))(tackle (expires <EXPIRES>) (count <COUNT>))(collision none—[(ball)][(player)][(post)]))

- (turn Moment)

- (turn Moment)
- (catch Direction)

- (turn Moment)
- (catch Direction)
- (dash Power)



- (turn Moment)
- (catch Direction)
- (dash Power)
- (kick Power Direction)

- (turn Moment)
- (catch Direction)
- (dash Power)
- (kick Power Direction)
- (move X Y)

- (turn Moment)
- (catch Direction)
- (dash Power)
- (kick Power Direction)
- (move X Y)
- (tackle Power\_or\_Dir)

- (turn Moment)
- (catch Direction)
- (dash Power)
- (kick Power Direction)
- (move X Y)
- (tackle Power\_or\_Dir)
- (pointto Distance Direction) & (pointto off)

- (turn Moment)
- (catch Direction)
- (dash Power)
- (kick Power Direction)
- (move X Y)
- (tackle Power\_or\_Dir)
- (pointto Distance Direction) & (pointto off)
- (attentionto Team Unum) & (attentionto off)

- (turn Moment)
- (catch Direction)
- (dash Power)
- (kick Power Direction)
- (move X Y)
- (tackle Power\_or\_Dir)
- (pointto Distance Direction) & (pointto off)
- (attentionto Team Unum) & (attentionto off)
- (turn\_neck Angle)

- (turn Moment)
- (catch Direction)
- (dash Power)
- (kick Power Direction)
- (move X Y)
- (tackle Power\_or\_Dir)
- (pointto Distance Direction) & (pointto off)
- (attentionto Team Unum) & (attentionto off)
- (turn\_neck Angle)
- (change\_view Width Quality)

- (turn Moment)
- (catch Direction)
- (dash Power)
- (kick Power Direction)
- (move X Y)
- (tackle Power\_or\_Dir)
- (pointto Distance Direction) & (pointto off)
- (attentionto Team Unum) & (attentionto off)
- (turn\_neck Angle)
- (change\_view Width Quality)
- (say Message)



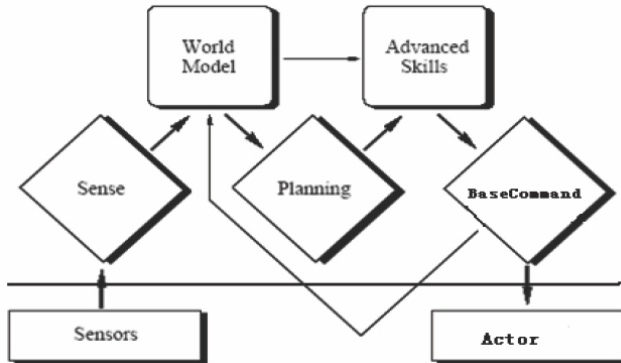
- 仿真平台提供给球员的是一个动态实时的对抗环境，一支队伍所有球员努力完成共同的目标，因此要求球员之间可以分工协作。

- 仿真平台提供给球员的是一个动态实时的对抗环境，一支队伍所有球员努力完成共同的目标，因此要求球员之间可以分工协作。
- 在每个仿真周期，球员要从Server接收感知信息，对这些信息进行处理分析，并决策产生要执行的行为，发送相应命令到Server。

- 仿真平台提供给球员的是一个动态实时的对抗环境，一支队伍所有球员努力完成共同的目标，因此要求球员之间可以分工协作。
- 在每个仿真周期，球员要从Server接收感知信息，对这些信息进行处理分析，并决策产生要执行的行为，发送相应命令到Server。
- 球员主要通过从server 获得的视觉感知信息来了解场上的状况。

- 球员在场上的运动和比赛行为都是由几个最基本的身体控制命令组合而成的，如果要求每次决策只使用这些基本命令，会大大降低决策的性能，因此，要求球员能够为决策提供相对复杂的行为支持。

- 球员在场上的运动和比赛行为都是由几个最基本的身体控制命令组合而成的，如果要求每次决策只使用这些基本命令，会大大降低决策的性能，因此，要求球员能够为决策提供相对复杂的行为支持。
- 当前使用的决策搜索算法，通常随着搜索时间的增加，获得最优决策的可能性也会增加。因此，为了提高球员决策的性能，在保证同server周期同步的前提下，应使用尽可能多的时间来进行决策。



- 整个流程的接收、决策和发送这3个基本任务由3个线程分别负责。Sense线程接收Server发送的信息并对其进行解析，Think 线程处理这些信息并进行决策得出相应的行为，Act线程则负责生成相应的命令及参数，并发送到Server。

- 整个流程的接收、决策和发送这3个基本任务由3个线程分别负责。Sense线程接收Server发送的信息并对其进行解析，Think 线程处理这些信息并进行决策得出相应的行为，Act线程则负责生成相应的命令及参数，并发送到Server。
- 多线程的主要优点是它提高程序运行效率和响应时间，保证了信息的发送和接收，同时减少了球员用来处理server 网络信息获取和发送的时间，使球员可以使用大部分时间来进行行为决策；但是多线程也会使球员实现更为复杂，需要正确解决线程间的同步问题。



```
{Sense Thread}  
while true do  
    block until server message arrives  
    determine send time t for Act thread  
    set SEND signal to go off at time t  
    WAIT(lock)  
    Parse server message and send it to world model  
    SIGNAL(lock)  
    Send READY signal to Think thread  
End while
```

```
{Think Thread}  
Create Sense thread  
Create Act thread  
While server_is_alive do  
    Block until READY signal arrives or 3 seconds have passed  
    If 3 seconds have passed then  
        Server_is_alive = false  
    Else  
        WAIT(lock)  
        Update world model  
        Determine next action  
        Send commands to Act thread  
        SIGNAL(lock)  
    End if  
End while
```

```
{Act Thread}  
while true do  
    block until SEND signal arrives  
    convert commands to string messages  
    send messages to server  
end while
```

## 1 Base Client

- 连接server
- 从server接收来的信息
- 向server发送的信息
- Client结构

## 2 UvA Source

参考资料: [Trilearn\\_Source\\_2003\\_doc.tar.gz](#)

Thank you for your attention!  
Q & A