软件复用课程设计

Client-Server 应用程序

CM构件改进意见

小组： Team10

成员： 谢志杰 1352975

计鹏玥 1352914

王思尧 1352896

谢明玥 1352937

徐锦程 1353012

在CM构件上，我们组选用的是第一组的CM构件，在此提出部分改进意见：

1. 第一组使用的配置信息格式如下：

DEBUG=true  
host=localhost  
MAX\_MESSAGE\_PER\_SECOND=5  
MAX\_MESSAGE\_PER\_SESSION=100

仅仅从配置信息上难以分辨出string类型还是int类型等等，也难以分辨配置信息是否可以被修改。建议使用JSON格式存放数据。例如：

{  
 "mutable" : {  
 "MAX\_MESSAGE\_PER\_SECOND" : 5 ,  
 "MAX\_MESSAGE\_FOR\_TOTAL" : 10 ,  
 },  
 "immutable" :{  
 "SERVER\_IP" : "127.0.0.1" ,  
 "SERVER\_PORT" : 2095 ,  
 "DBUSER" : "root" ,  
 "DBPW" : "root"  
 }  
 }

这样可以清晰地看出数据类型以及可修改与不可修改的配置信息，方便对配置信息的修改操作。

2. 第一组实现动态配置的方法是对配置文件添加一个事件监听器，当配置文件的内容发生变化时，事件监听器调用fileChanged()函数，loadConfig函数被重新调用，即程序重新载入配置信息。

DefaultFileMonitor fm = new DeafultFileMonitor(new FileListener() {  
 @Override  
 public void fileCreated(FileChangeEvent fileChangeEvent) throws Exception{  
 //Do nothing  
 }  
  
 @Override  
 public void fileDeleted(FileChangeEvent fileChangeEvent) throws Exception{  
 //Do nothing  
 }  
  
 @Override  
 public void fileChanged(FileChangeEvent fileChangeEvent) throws Exception{  
 loadConfig();  
 if(mConfigListener !=null) mConfigListener.OnConfigChanged();  
 }  
});  
fm.setRecursive(true);  
fm.addFile(listendir);  
fm.start();

但是，并不是所有的配置信息都支持动态配置。例如，SERVER\_IP与SERVER\_PORT在客户端运行时已经确定，即时程序重新载入也无法重新配置。一次载入所有配置信息会造成系统性能的下降，建议重新调用一个函数，只载入支持动态配置的配置信息。