The example program is basically to demonstrate a simple working example of functionality of 3 layers of the architecture, namely FAtiMA, CMION and SAMGAR for a Pioneer robot.

When you run the example, Robot should approach the user and greet (Printed out from CMION competency in java “Speak Competency says Hello my name is Sarah”) the user when found in the proxemic range.

In this example the robot should have the following:

1. OpenCV, ARIA and Yarp (version yarp-2.2.5 and above) correctly installed
2. Camera installed on the robot facing in the direction of forward motion. The camera ideally should be placed at a height where it can see faces of people standing and on the centre point of robot’s horizontal axis (between 2 wheels), since the program takes into account the position of face detected followed by the amount of turn required to face the user and approach the user.

Steps to start the example program: (This is an example program for Windows)

1. SAMGAR
   1. Run cmake (recommended cmake version 2.8) inside \lirecsvn\scenarios\TeamBuddy\SAMGAR, make sure that the paths in CMakeLists.txt points to your own packages for e.g. OpenCV and Yarp
   2. Compile the Example project it should generate some binary files. Binaries are also included in release folder.
   3. Start Yarp and SAMGAR as mentioned in SAMGAR tutorial (<http://adapsys.feis.herts.sc.uk/SAMGAR>/) for example
      1. Run yarp namespace /global followed by yarp server 20000 in command prompt window and in other window run
      2. yarp namespace /sarah followed yarp server 10000
   4. Start the SAMGAR GUI (SamgarKey)
   5. Start the modules (binaries complied from Example project)
      1. SamgarSendImage, SamgarFaceDetect: Connect them VideoOut(SamgarSendImage) to VideoIn(SamgarFaceDetect). You would see the result image in window of SAMGAR-FaceFinder module.
      2. SamgarMove, Please note that there is **no obstacle avoidance** in the motion module, motion commands directly invoke function from ARIA motion class, although the robot should move only if it finds a face. SamgarMove would connect to the robot platform
2. CMION:
3. Start java program CMION (\lirecsvn\libs\cmion\cmionLauncher), please refer the CMION doc (\lirecsvn\libs\cmion\cmion.doc) for configuring the java packages for your own programs. Also make sure that you include the package cmionTeamBuddy which contains the source code and competencies for TeamBuddy Example.
4. Main files for CMION configuration are
   1. ArchitectureConfiguration.xml: **Make sure you copy this file** from \lirecsvn\scenarios\TeamBuddy and paste it into \lirecsvn\libs\cmion\cmionLauncher (just make a backup copy of the original file). This is the main file which sets the connection for FAtiMA and CMION. The file also specifies the XML files for CompManagerRules and CompetencyLibrary.
   2. CompetencyLibraryTeamBuddy.xml: Contains the list of competencies
   3. CompManagerRulesTeamBuddy.xml: Contains the sequence and conditions for list of competencies to be started to carry out a particular mind action.
5. Once you start CMION, SAMGAR GUI should show CMION competencies now and then connect the CMION-FaceDetect(In) to SAMGAR-FaceFinder(out) modules. Then connect CMION-MoveRobot(Out) to SAMGAR-Move(In) Module
6. FAtiMA
7. Go to: \fatima-bin and start TeamBuddyAgent.bat file, you would see a window open for FAtiMA mind and its status
8. Inside the \fatima-bin\data\characters\minds, you need 4 main files
9. LIRECScenarios.xml: Where the properties about TeamBuddy scenario is specified
10. TeamBuddyActions.xml, TeamBuddyGoals.xml: Properties of Actions and goals are specified
11. NoCultureTeamBuddy.xml: Symbolic actions are specified
12. Inside \fatima-bin\data\characters\minds\roles\TeamBuddyAgent\ TeamBuddyAgent.xml, the goals specified, the emotional thresholds and motivation parameters may not be modified. It is important to specify the goals.

Please start the programs in the sequence mentioned above 1) SAMGAR Modules🡪 2) CMION🡪 3) FAtiMA-TeamBuddyAgent.bat