

VIRTUAL HUMANS FOR SERIOUS GAMING ARCHITECTURE DESIGN

Jannelie de Vries
Harmen Kroon
Jasper van Tiburg
Tom Brunner
Nando Kartoredjo



EEMC
Delft University of Technology
Delft April 29, 2016

Contents

1	Introduction	2
1.1	Design goals	2
1.1.1	Availability	2
1.1.2	Manageability	2
1.1.3	Performance	2
1.1.4	Reliability	2
1.1.5	Scalability	2
1.1.6	Securability	2
2	Software architecture views	3
2.1	Subsystem decomposition	3
2.2	Hardware/software mapping	3
2.3	Persistent data management	3
2.4	Concurrency	3

1 Introduction

This document provides a sketch of the system that is going to be built during the context project Virtual Humans. The architecture of the system is explained in the form of high level components of the system. These components are split in to sub components and sub-systems.

1.1 Design goals

The following design goals will be maintained throughout the project:

1.1.1 Availability

The system will be built by the principle of Scrum, meaning that each week a working version is available for testing by the client. By doing this we ensure that we are building what the clients wants; if (s)he sees features (s)he does not like we remove them, if they have other requirements we add them.

1.1.2 Manageability

The bot is able to be turned off and on. Its property can be managed (including its money). The goals can be altered to a higher demand.

1.1.3 Performance

The bot should run smoothly and given it should not be easily noticed that the bot is not a human due to slow functioning of the bot.

1.1.4 Reliability

The bot should work in any given scenario, although it might decide it is better to do nothing.

1.1.5 Scalability

Our bot will be able to work together will be able to work with other bots and in theory an infinite amount could be added.

1.1.6 Securability

Since our implementation will be added to the server side and not the client side of the Tygron game, we will rely on their security measures.

2 Software architecture views

2.1 Subsystem decomposition

The component diagram at **figure 1** shows the relations between the Tygron API, the connector and the GOAL agent. The GOAL agent has not been implemented yet.

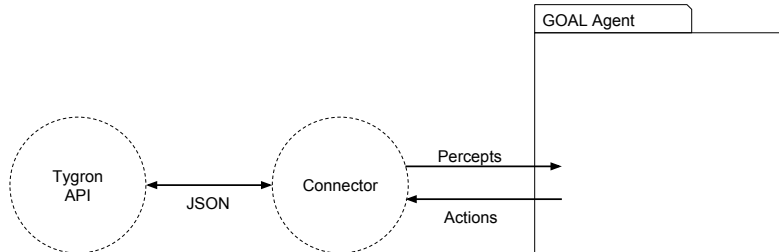


Figure 1: The component diagram

2.2 Hardware/software mapping

The software uses a different connection approach to the server than a regular user. When a session is created via the server, a user connects through its own client to that session. Our virtual human will connect to the session using a connector and the Tygron SDK. Visualisation of the actions performed by the virtual human are visible through a separate instance of the Tygron Engine.

2.3 Persistent data management

Our GOAL agent is not responsible for storing persistent data, so it does not have external files or databases. Everything is stored on the database of the Tygron server.

2.4 Concurrency

Our GOAL Agent is just one process. It shares resources with other users of the tygron environment. If a deadlock occurs, this would be on the tygron environment. Our GOAL system won't have issues with deadlocks.

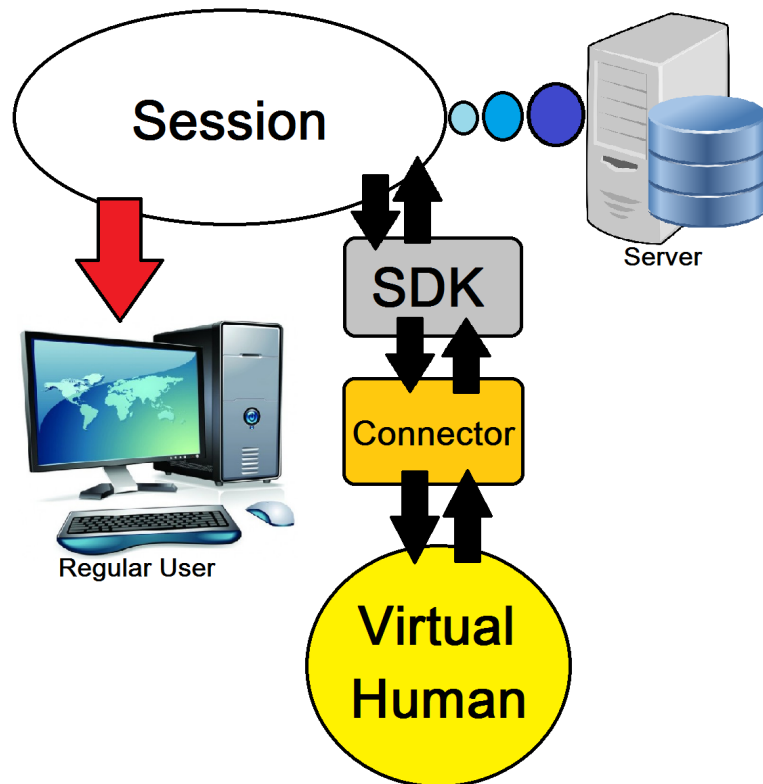


Figure 2: Hardware software mapping