

Architecture Design

27/05/2016



CodeFox

Marco Houtman

Ronald van Driel

Joshua Slik

Matthijs Halvemaan

Lisette Veldkamp

Contents

1. Introduction

1.1. Availability

1.2. Manageability

1.3. Performance

1.4. Reliability

1.5. Scalability

2. Software architecture views

2.1. Subsystem decomposition

2.2. Web2Eis Connector

2.2.1. Connector Overview

2.2.2. Working with the Connector

2.3. GOAL agent

2.4. Hardware/software mapping

2.5. Persistent data management

1. Introduction

This Document provides a sketch of the system that is going to be built during the context project Virtual Humans for Serious gaming. It is used to represent the current state of the design of the system. The systems architecture is described in high-level components.

Design goals

The following design goals will be maintained throughout the project:

1.1. Availability

After each sprint the system should be in a working condition. This is important because it will allow the user to utilise the systems features and provide feedback based on the experience. The final version of the product should always be available and working in the same environment as it has been developed for.

1.2. Manageability

The code will be made publicly available for other programmers who are interested in our progress. All code will be well documented and commented for a clear overview of the system and will enable additional developers to extend or modify the agent.

1.3. Performance

Our product must be at least runnable on mid end personal computer systems. The agent should be able to act and react on real time environmental changes without delays exceeding more than a minute.

1.4. Reliability

The agent can be run with help of SimpleIDE or the GOAL environment in eclipse when the Tygron environment is available to the client and enough hardware resources are available.

1.5. Scalability

Our agent should keep working and interacting when additional agents are added to the environment and when the environment changes. It does not guarantee however that additional features in the environment or in additional agents will be utilized.

2. Software architecture views

In this software architecture section we describe a high level overview of different subsections of the system. In the first section we describe the programming aspect of the system, in the second section we provide a high level overview of the full system and the third section gives a high level overview of the most important subsection of the system.

2.1 Subsystem decomposition

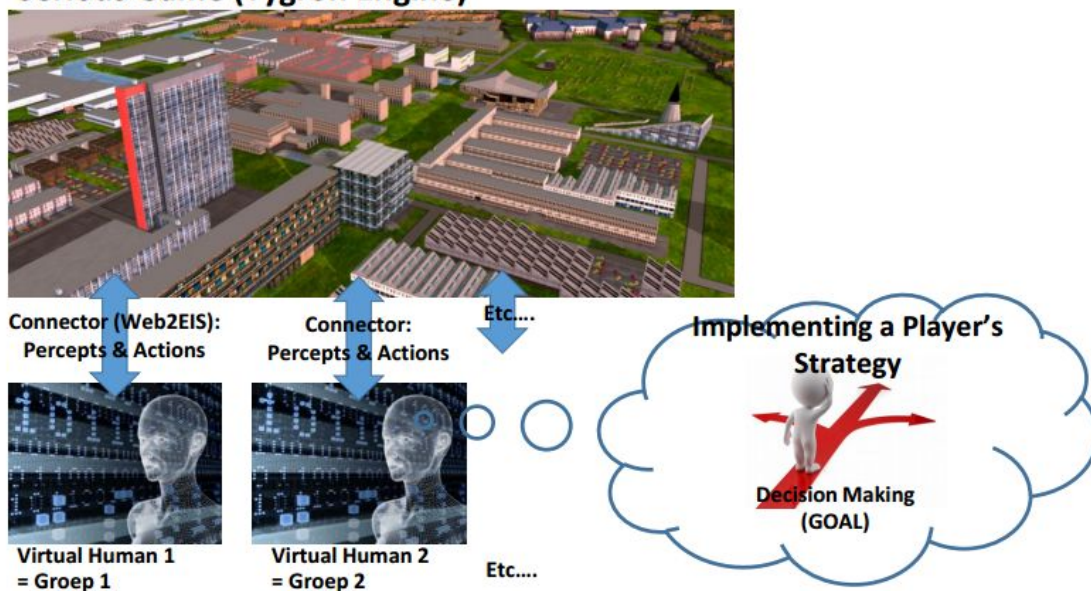
For this project both Java and GOAL, a high-level programming language for programming decision logic of cognitive agents that derive their choice of action from their beliefs and goals, will be used. Java is primarily used for the interaction between the agent and the environment. GOAL will be used to program the decision making of the agent using Artificial Intelligence.

The Tygron environment allows multiple agents to connect and interact. Each agent requires a connector which acts as an interface between Tygron engine and the agent (as seen in the image below). This interface is required for all agent interactions with the environment.

All agents have separate logic but are able to interact with each other through the Tygron Environment.

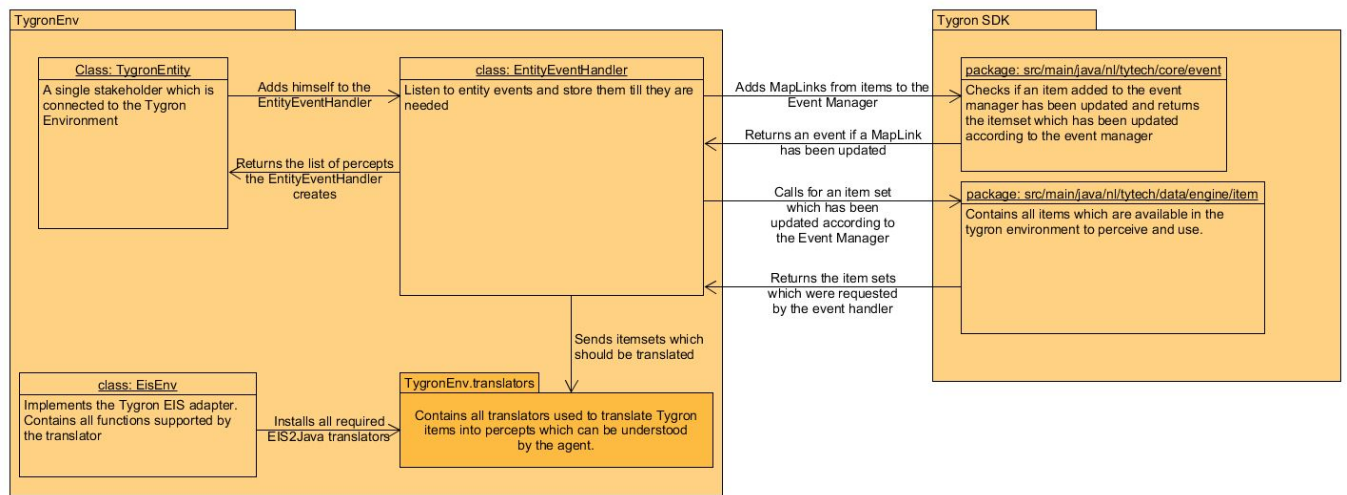
Our product will consist of the implementation of a single virtual human. For our product a basic connector has been provided but can be extended when desired. Most implementational logic will be required for the agent's decision making in GOAL.

Serious Game (Tygron Engine)



Koen Hindriks, "Kickoff Virtual Humans 2016" (Powerpoint Presentatie, Den Haag, April 21, 2016).

A full overview of the percepts in the system is shown in the following UML we made:

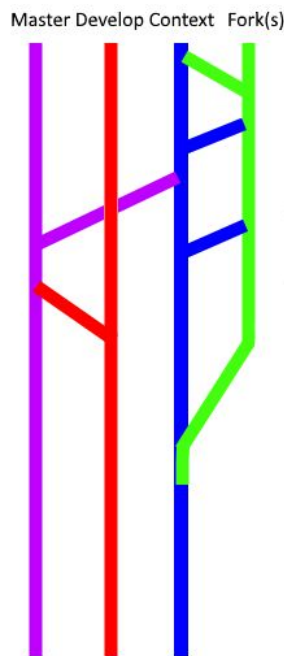


2.2. Web2EIS-Connector

This chapter will focus on the connector used to communicate between the Tygron Engine and the GOAL Agent. First a total overview of our workings with the connector is displayed in the Connector Overview, next is an explanation on how to work with the connector in Working with the Connector.

2.2.1. Connector Overview

The Web2EIS-connector is our base connector currently being expanded by several people involved with the (EIS)hub and by the groups of the Virtual Humans for Serious Gaming project. The groups all have a fork of the Tygron-EIS and are expected to develop their own Percepts and Actions for the virtual human they are developing.



The dev branch is being developed by the (EIS)hub programmers and when they are ready to release their branch is merged into master. After release and a reviewed pull request master will be merged into the context branch. The groups will fork the context branch so they can work on their percepts and actions. When one of the context groups has a working feature for the context branch it will make a pull request and when the request is positively reviewed will merge their changes into the context branch so that it can be used by the other project groups.

Github's issue system is being used by the context groups to keep a clear overview of who is working on which part of the connector. An issue can be created by any team member and can be assigned to team members of all context groups.

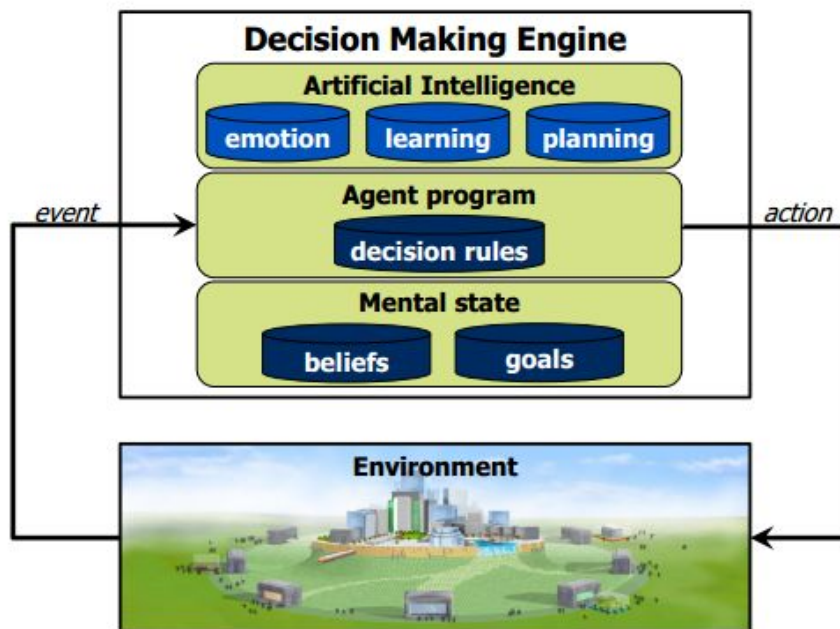
2.2.2. Working with the Connector

There are two ways in which an agent can interact with the environment, these are the percepts, which is an observation of something happening in your environment, and Actions, which are operations which an agent can do to influence the environment. For both

2.3. GOAL agent

Our virtual human will perceive events from the Tygron environment. The useful events will be processed in the agent as beliefs or goals for the mental state. Based on the mental state and the implemented decision rules, the agent will make smart decisions. Then the agent will observe what happens after its decision and will learn from this so that next time when there is a similar situation the agent can handle it better.

The connection from the agent with the environment is also shown in the overview below.



Koen Hindriks, "Kickoff Virtual Humans 2016" (Powerpoint Presentatie, Den Haag, April 21, 2016).

Our GOAL agent (which has the Private Housing Corporation role) will exist out of four modules.

In the main module the agent will use his percepts to decide what he wants to do in this cycle. First he processes the new changes in the environment. Then, based on his knowledge and percepts the agent will choose a location of a piece of ground that is in the environment. It then depends on what he wants to do with this ground which module(s) the agent will enter.

The first module will be the Buying Module. If the ground has a nice appeal for building houses the agent will enter this module. He will then come up with a fair bidding price for the ground and if he has the money will make a bid on the ground.

The second module will be the Selling Module. If the agent came across a piece of ground that he owns that does not live up to his expectations, the agent wants to sell it so that he can do something better with the money.

The third module will be the Demolishing Module. If the agent finds a piece of ground (with a building on it) that he owns but that does not live up to his demands (anymore) the agent wants to demolish the building so that he can do something better with the ground.

Finally the fourth module will be the Building Module. If our agent owns ground that lives up to his standards he wants to build a building there.

We are not yet sure about upgrading buildings but there might be a fifth Upgrade Module, where the agent will upgrade buildings if that is needed. However we have not yet decided if this is possible with the percepts, knowledge and actions that we have at the moment.

2.4. Hardware/software mapping

The virtual human will run a session on a computer. It is able to connect to the session via the Tygron-EIS connector and the SDK. The virtual human will then be connected to the server.

2.5. Persistent data management

All actions that are being executed that have an effect on the Tygron environment will be stored on the Tygron server. Our agent will collect this data (or at least the useful parts of the data) from the Tygron environment when it is needed for the strategy of the agent. The agent itself will have no knowledge of previous game sessions.