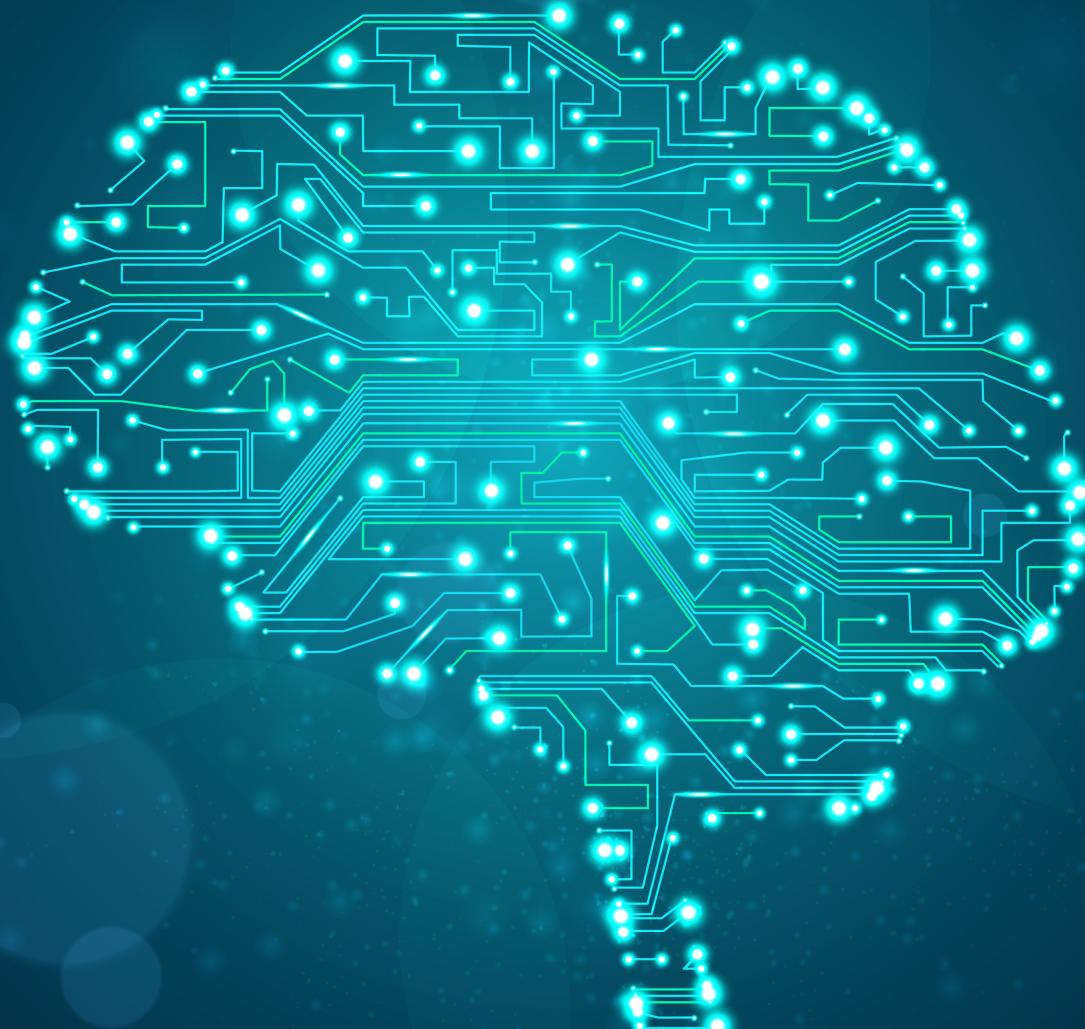


Virtual Human

Context Project
TU Delft



APRIL 20, 2015 – JUNE 19, 2015

T
Y
G
R
O
N

Contents

1	Virtual Humans	3
2	Design Goals	3
3	Software Architecture	3
3.1	Test Game	3
3.1.1	Stakeholders and Indicators	4
3.1.2	Map	4
3.1.3	Targets	5
3.2	Virtual Human	5
3.2.1	Goal File	5
3.2.2	The Municipality	6
3.2.3	Stakeholder Modules	7
3.2.4	Event Handling	7
3.2.5	Emotions	7
3.3	Tests	7
4	Glossary	8

1 Virtual Humans

The design of a virtual human is our main goal. This virtual human needs to connect to the Tygron engine to work in its environment. To create this we use the programming language 'GOAL' in which we use 'Gamygdala' for integrating emotions in our virtual human. Furthermore we need to create a test map/game to test our virtual human in.

2 Design Goals

We design our product according to the following goals:

- **Expendable**

The program will be expendable. Tygron is a growing company and the software they are developing is improving as well. Our product is an expansion on that software. So it needs to be adjusted when the Tygron changes its software. For this we focussed mainly on the stakeholders and strategy of these stakeholders. The strategy can be easily changed and new stakeholders can be added.

- **Simple**

The code will be simple. Not only for Tygron to understand but also to be well organized. We did so by handling all events in one place and defining the pre- and postconditions of actions when adopting the goal for performing that action (see 2.2.1).

- **Dynamic**

The program will be dynamic. From one file it can act as every stakeholder. Furthermore if targets are reached the program does not stop running, but it waits until a goal does not have a 100% progress any more. Then it just returns to running the normal program, performing actions to get back to that 100%.

3 Software Architecture

In this section we will explain the choices we made for the software architecture and how we will implement these. Since we are actually going to create two main products, the virtual human and its test game, we have two software architectures based on their product.

3.1 Test Game

To test our virtual human we will create a test game in the Tygron engine. For creating this game properly we decided upfront what we wanted to put in the game. Therefore we are taking different factors into account like stakeholders and indicators.

3.1.1 Stakeholders and Indicators

In the game there are multiple roles (stakeholders) that need to be played. In our case there are three: the municipality, the company and the housing corporation. Each with their own or shared indicators. Though every stakeholder has a different interest in each indicator.

The municipality is the deviating one since this stakeholders most important job is to decide if land may be bought and permits for building are granted. So the municipality has a lot of influence on the other stakeholders development. This makes it complicated to implement this stakeholder. In section 2.2 we will explain more about what we are going to do with the municipality.

For designing the stakeholders we wanted to know what the indicators could be that the stakeholders find important. For example the housing corporation will probably have *housing* as their most important indicator.

We decided to use the following indicators:

1. Green
2. Parking
3. Budget
4. Building

The different stakeholders have these interest in the indicators:

	Green	Parking	Budget	Building
Municipality	55%	30%	15%	
Company		30%	50%	20%
Housing			40%	60%

The percentages of importance per stakeholder are made up in such a way that it creates the most ideal test environment.

3.1.2 Map

For the game to work we need a balanced map. By balanced we mean that the distribution of buildings, roads, parks (etc.) and free land will be fifty-fifty. Because of this distribution the virtual human has the space to build but it still needs to keep in mind that there is already some civilisation built. For example it should not build a lot of houses if half of the map already consists of houses. So the virtual human should initialize its goals right in a way that it will plan to fill the other half of the map in a way that satisfies its goals at best.

The map we have chosen contains waterways. Water in a map adds an extra dimension in building because it opens the possibility to build structures that can only be built on or above water, like bridges and harbours. We removed the water from our map to lessen the complexity since none of our stakeholders has any interest in water.

3.1.3 Targets

Every stakeholder has the goal to complete their targets. These targets are based on indicators. Just as the indicators we made up the targets as well. A target consists for example of a type of building and the amount that needs to be made of that building. All targets combined would take more space than there is free space for buildings. This is chosen on purpose to provoke conflicts. For example if there is just enough land for the housing corporation to complete their building indicator by placing the type of building that is needed for the target of this indicator and company has an equal goal but with offices, then there is a conflict between the two stakeholders.

3.2 Virtual Human

The test game didn't require programming. To determine the initial settings and building the map we did not use any code. Creating the virtual human is a lot different, because here it is all about programming.

Since we made the test game before we began making the virtual human, we determined where conflicts would occur. This is beneficial for our software architecture.

3.2.1 Goal File

The goal file is the most important part of the virtual human. This is basically the main file (see figure 1). This file consist of multiple modules which we will discuss one by one.

- **Init module**

The first module is the init module. In this module the knowledge and actions are defined and the beliefs are initialized. The knowledge consists of Prolog rules where we do our calculations, like the amount of land we need or the amount of rejected permits we received. The actions have a pre- and postcondition. In our case these are all set on true, because we process these condition in the engine. The beliefs are initialized in two different ways. Some are defined in the code while others are initialized if a percept is received from the environment. The ones in that are defined in the code are needed for our strategy, for example the price that is offered for a m² land. The others are beliefs that we get from the environment, like stakeholders and targets.

- **Main module**

The second module is the main module. This module consists of only the program. In here the standard steps to be taken are programmed, so only the steps that are the same for each stakeholder. Furthermore we direct the stakeholder to its own main module in here (see 2.2.3) where actions are performed based on the goals that are set.

- **Event module** The third and last module is the event module. Here the on-change percepts are processed, these percepts update the corresponding beliefs. It also checks if all targets are reached, because when this is the case, the agent has to stop performing actions. It should not stop

completely, but keep an eye on the targets since some of them are dynamic and could be changed when another stakeholder performs an action. So to prevent such a premature termination, we added a believe that is set on true when all targets are reached and that switches to false if the progress of a target is not 100 % any more. When it is true the bot waits and when it is false the bot runs its normal program.

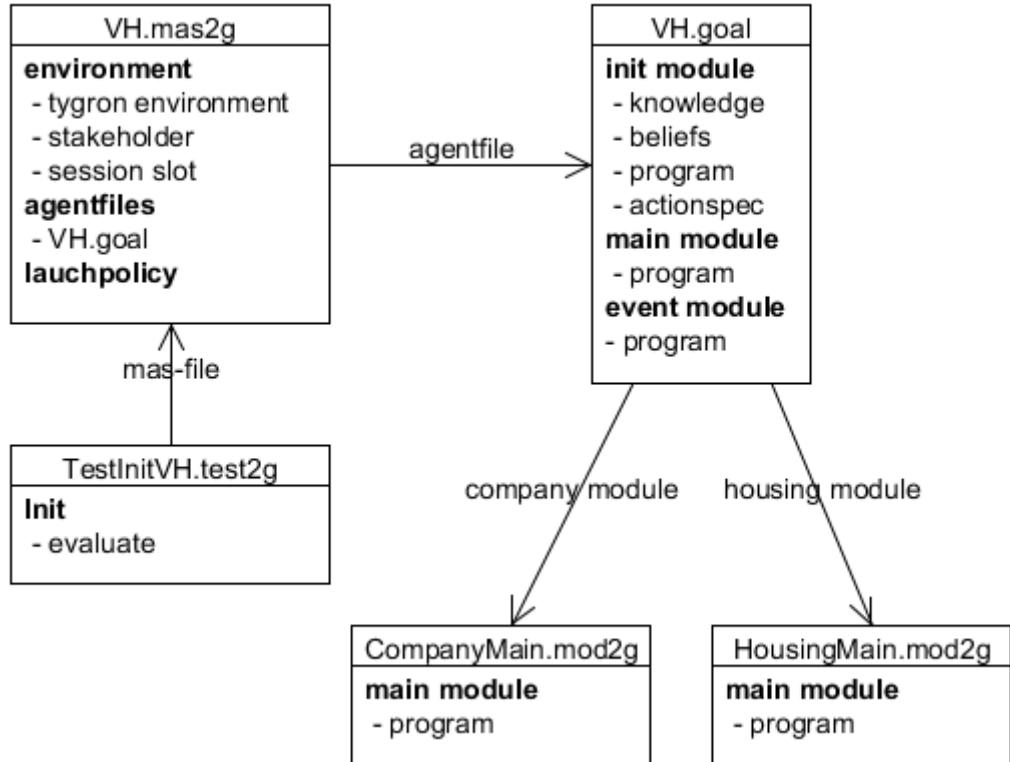


fig. 1 architecture of the virtual human

3.2.2 The Municipality

Like we explained shortly before, the municipality is a unique stakeholder which plays a big role in the game. The municipality has not only the task to complete its own target but it also has the task to accept or refuse permits that other stakeholders ask for.

There are two types of permits that stakeholders can ask for. They can ask for permission to buy land or build structures. The first kind of permit has influence on the distribution of free space among the stakeholders. The second kind of permit has influence on the development of the indicators.

With this in mind we made the decision to divide the process of the implementation in three steps. The first step is no implementation. In this step the role of municipality is played by a real person. The second step is a very

simple implementation which accepts all permits. In this case we can at least let the municipality be played by a virtual human, but might be a minor step back since accepting all permits would make the other stakeholders strategies useless since these are based on rejection. The third step is a full implementation where the municipality can decide whether or not a permit is granted. Not only taking the different indicators into account but also thinking about emotions of itself and other stakeholders.

The step we eventually can complete is dependant on the time we have.

3.2.3 Stakeholder Modules

For every stakeholders, except the municipality, we made a different module (see figure 1). These are company and housing corporation for now. The virtual human basically has only two actions, the build action and the buy land action. In these modules whether these actions are taken are based on their beliefs and defined in the program module. These beliefs are mostly about permits, the amount of available land and its budget.

3.2.4 Event Handling

There are not many events to handle in this virtual human, because there are not many beliefs to update. The events that do occur are handled in the event module of the main file. For both stakeholders the events are the same. Target update, available land update, budget update and amount of rejected permits are needed for every stakeholder.

3.2.5 Emotions

Another part of the implementation is the integration of emotions. We will use a port that connects GOAL and Gamygdala. The combination of beliefs and goals will provoke certain emotions via this port. The emotions that are returned will have influence in the decision making. If a stakeholder has positive emotions, it will easier cooperate with the other stakeholders, while negative emotions would do the opposite.

In our case, emotions will be used to determine the cost to buy land. The happier the stakeholder is, the more he will offer for a part of land since it has confidence from its previous success. When a stakeholder is happy, he will easily take more risk. If the stakeholder is really unhappy, which means that almost no building of his was accepted, it will pay less for land because he has little hope for success.

Emotions are a complicated part and for this reason only the price of land will be used to show the possibilities of emotion.

3.3 Tests

The code is tested in two ways. One is with test2g-files, where actions and events are tested via unit testing. Mainly the initialized events are covered.

Manual testing is the second way of testing. In the beginning the test were used to check whether the server would execute the implemented actions. So we simple started a server and ran our code on the same server slot. While the code runs the engine should visualize if the actions are working. For example if we started the server, joined it as the municipality and let our code run for the company stakeholder, then we should see a permit from company that he wants to buy land.

After the actions appeared to work, we focussed on testing the strategies of the virtual human. The order of actions, the way of achieving targets and the completion of progress are all tested that way.

4 Glossary

GOAL

A programming language for programming virtual humans.

Gamygdala

Emotion simulation program which can be integrated in GOAL.

Tygron

The company where the product is built for. They are specialized in serious games for city planning.

Prolog

A programming language that uses logic based rules to perform certain actions.