Architecture Design

VH-Danshal

1 Introduction

This document provides an overview of the architecture of the system we are going to build and the goals our code should strife for in order to produce a good product. The architecture will be given in a high level description of systems which will be separated in subsystems.

Design goals

In order to deliver a good working and maintainable system we have set goals for our code

1.1 Design goal: Performance

Our code should be able to run on most modern consumer pc's. This is possible because of the architecture of our product. The connectors should handle all low level functions and the GOAL agent should provide high level artificial intelligence, simulating the decision making of a real human player

1.2 Design goal: Availability

Our team uses Travis for continuous integration in order to make sure our main branch is always a working product. This way we can show a prototype to our client each week, the client can then review the changes and help us improve on our product and implement the features our client needs

1.3 design goal: Code quality

Our code should be of high quality so it is easy to modify and maintain by both us and future users. To reach this goal we make use of checkstyle, junit and we will use java-doc to generate documentation for our code

1.4 design goal: Functionality

We implement an agent that can fulfill the following given the TU Delft environment in the Tygron engine and given that the role assigned is that of the municipality. Make decisions through knowledge of the environment and communication with other players and change the environment to improve it in favor of the indicators of the municipality.

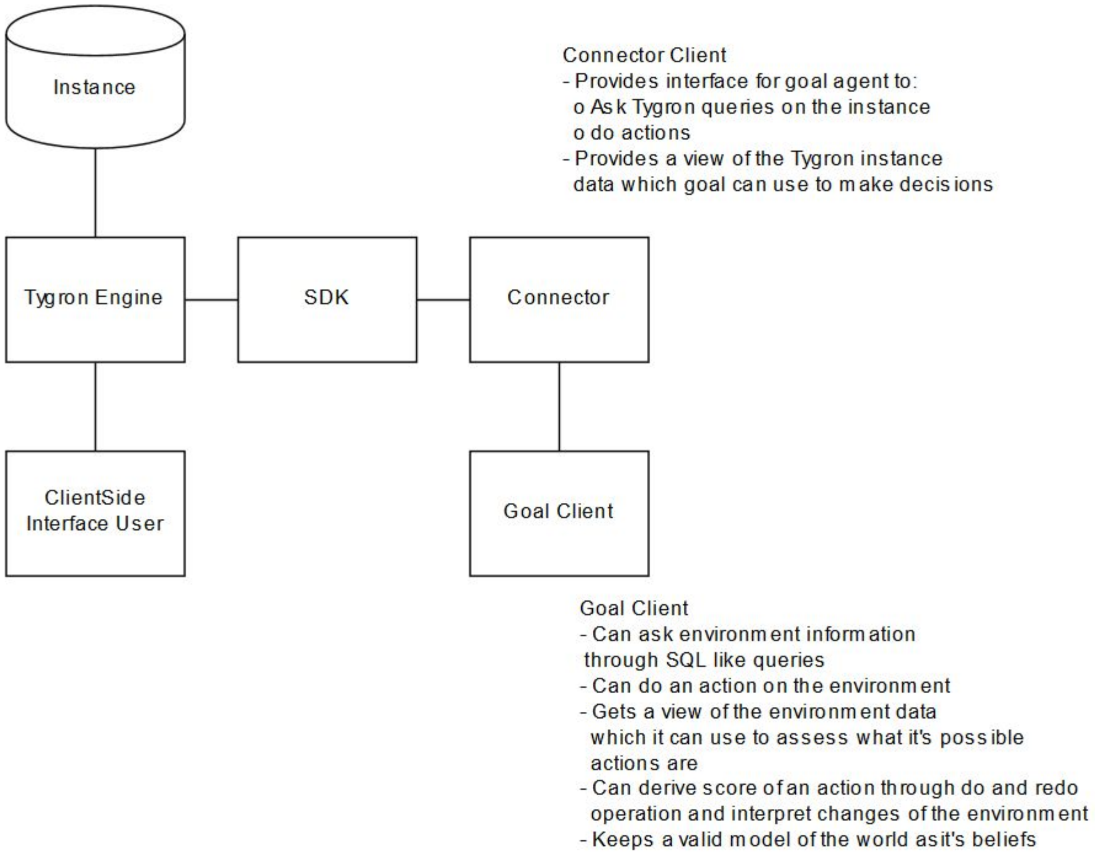
2 Software architecture views

2.1 Programing languages

For this project we'll be using two programming languages. The Goal Programming language that supports a declarative style of programming and java, an strict OOP language. Java is the language in which the SDK and the connector is written, it is used to connect the Goal agent with the Tygron server.

2.2 Subsystem decomposition

The following diagram reflects our current understanding of the architecture and what behaviours should be handled by what systems.



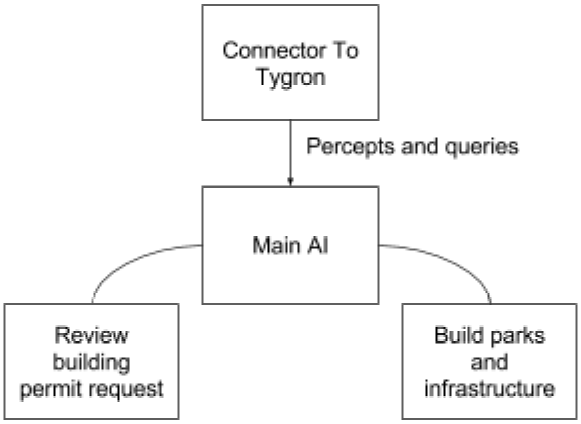
Connector Client
- Provides interface for goal agent to:
  o Ask Tygron queries on the instance
  o do actions
- Provides a view of the Tygron instance data which goal can use to make decisions

Goal Client
- Can ask environment information through SQL like queries
- Can do an action on the environment
- Gets a view of the environment data which it can use to assess what it's possible actions are
- Can derive score of an action through do and redo operation and interpret changes of the environment
- Keeps a valid model of the world as it's beliefs

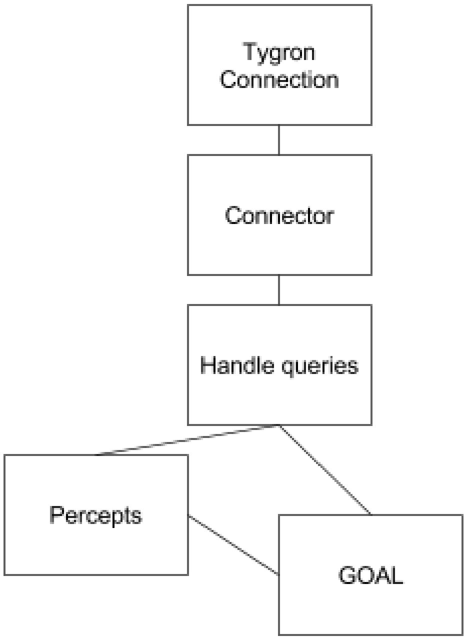
We will primarily use the Goal client to achieve the functionality as specified as a design goal. But because the connector is unfinished and has few features, we can add features to it to support the actions our Goal agent needs to improve the situation in it's favor and the information our Goal agent needs to make informed decisions.

2.3 The AI

Our AI will have 2 main roles. As municipality we will need to review building permit requests. And we also have to reach our own goals by building parks and infrastructure. To perform these tasks we will use GOAL to perform high level decision making.


Connector To Tygron
Percepts and queries
Main AI
Review building permit request
Build parks and infrastructure

2.4 The environment


Tygron Connection
Connector
Handle queries
Percepts
GOAL