BW4T3 Instructions

W. Pasman, M. B. van Riemsdijk,

Contents

1 Introduction

Blocks World for Teams (BW4T) is a testbed for team coordination. BW4T allows for games with human-human, agent-agent and human-agent teams of variable sizes. The goal is to jointly deliver a sequence of colored blocks in a particular order as fast as possible. A complicating factor is that the players cannot see each other.

BW4T is a client-server system. The server is responsible for the administration, simulation and visualization of the virtual world: it keeps track of robots, rooms, blocks, connected GOAL agents, etc. The server uses Repast, software to simulate virtual environments, to do part of this administration. The client is GOAL, which runs a multi-agent system (MAS) and connects to the server. The agents in the GOAL client get percepts from the server, and send actions to the server. Client and server can run on a different computer. This document describes how to install the BW4T server, configure it, place the BW4T client in GOAL, configure the MAS file, and run the system.

This chapter descibes how users can install and use the Blocks World for Teams environment. For developer information please go to the project page on github https://github.com/eishub/BW4T and click on the Link to Developer Details.

We use the following names to refer to directories of BW4T:

- <GOAL> refers to the directory where you installed GOAL.
- <SERVER> refers to the directory where you have put the server.jar, documentation and utilities.
- <CLIENT> refers to the directory where you have put the client.jar.

1.1 System requirements

To use BW4T you need Java JDK 7 or higher. The BW4T3 environment has been tested on Windows 7, Windows 8 and OSX. Note that OSX may come with java 6 pre-installed. You need to install java 7. You can then keep java 6 installed as well, but ensure that java 7 is used as the default.

1.2 Installation

This document describes how to install Blocks World For Teams (BW4T) for use with GOAL. We briefly comment where necessary how it would work with a different agent system. You can also use BT4T stand-alone but this is not described in this document.

You need to install a number of items to run BW4T from GOAL.

1. (if not yet done): Install Java 7 (download from http://www.java.com) and made it the default Java (such that double clicking on a jar makes it open with Java 7).
2. You can download GOAL from $http://ii.tudelft.nl/trac/goal$. You can choose for the plugin for Eclipse or the stand-alone SimpleIDE.
3. GOAL provides a sample Multi-agent-system 'BW4T3' that is demonstrating the Blocks World for Teams system. [1]

[1] If you run in a different agent system, you can download the client-side environment from *https* :

4. You can download the BW4T server from *https* : *//github.com/eishub/BW4T/releases* and save it in <SERVER>. Select the latest *bw4t* − *server* − *X.Y.Z.jar*.

5. You can optionally download the BW4T map editor from *https* : *//github.com/eishub/BW4T/releases* and save it in <SERVER>. Select the latest *bw4t* − *environment* − *store* − *X.Y.Z.jar*.

6. You can optionally download the BW4T scenario editor from *https* : *//github.com/eishub/BW4T/relea* and save it in <SERVER>. Select the latest *bw4t* − *scenario* − *editor* − *X.Y.Z.jar*.

2 Running the server

Run the server before running the client, as otherwise the client cannot connect to the server. Start the server by opening (typically: double clicking) the server.jar in <SERVER>. This should open the server window (Figure 1). Note that during this opening, two other maps are created:

- Maps: The folder in which all possible maps are put.
- Log: The folder in which all log files will be placed.

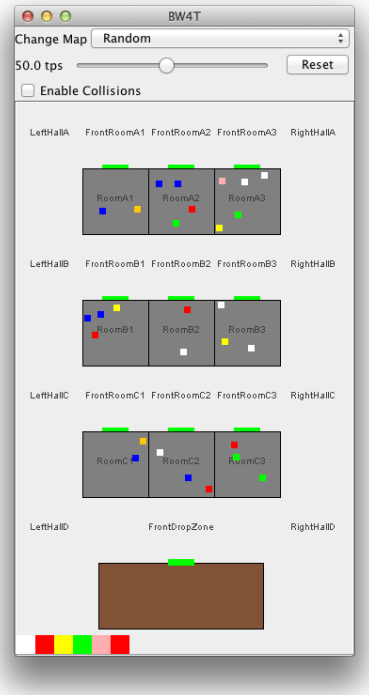


Figure 1: Server window

//github.com/eishub/BW4T/releases. Select *bw4t* − *client* − *X.Y.Z.jar*. This is the environment that should be attached to your agent system. You then will have to write your own agents that can run in the BW4T world.

2.1 Advanced server settings

The default settings for the server can be changed in the command line of your OS.
For example, to change the server ip and/or server port 999 execute the following:
java -jar bw4t-server.jar -servip <server ip here> (default: localhost) -serverport 999
The server will now start using the new values.
These are all the options for the server:

- -scenario The path to the scenario folder required by repast. Default is BW4T.rs. We recommend to not change this.
- -map the name of the map file in the /maps/ folder (which is created at startup if it does not yet exist). Default is 'Random'. These are really xml files that you can create and edit with the map editor.
- -serverip The ip to bind this server to. Default is 'localhost'. If you change the serverip and/or serverport, change the client settings correspondingly (see below).
- -serverport The port to bind this server to. Default is 8000. You can change this if there already is another service on your machine using the default port. Make sure that you also set up the client accordingly if you change this.
- -msg the message to be made available to the clients.
- -gui true if GUI should be enabled, false if server should run without GUI. Defaults to true. Setting this to false may be useful particularly for batch runs.
- -key The key necessary to remotely kill the server
- -collision whether the environment should check the collisions. Defaults to false. Note that collision checking still has some issues.
- -paths True if draw paths is enabled. This allows you to see planned paths for entities.

3 Running the client (Human Controlled Bot)

Before running the client, make sure you have already started the server. Start (e.g., double click) the client.jar in <CLIENT>. Figure 2 will show up and the bot is automatically added to the server. You can now control the bot by clicking (left or right) at different spots in the Client. Please refer to Section 6 for details on using the this client.

Notice: if you run both agents and human controlled bots, you should use the HUMANGUI option and launch the human GUIs in coordination with the agent platform. Please refer to the Advanced run settings below.

4 Running the client (Agent Controlled Bot)

Before running the client, make sure you have already started the server. Start your favourite Agent platform and load run the MAS. If you use GOAL, this means open the .mas2g file and press the run button. This time, no new windows will appear. The agent will control your bot, and agent behaviours can be seen only by introspecting the agents with your agent

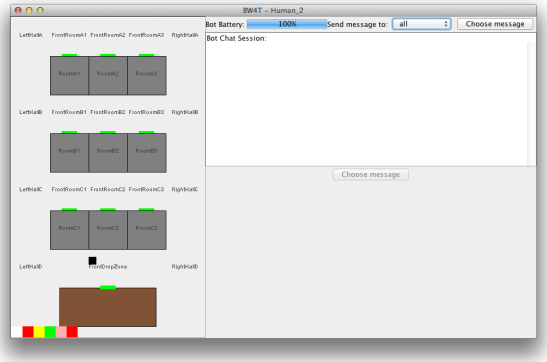


Figure 2: Client window

platform tools. In the server window you will see your bot(s) move according to the rules applied in the .goal file(s).

4.1 Advanced run settings

The following MAS init parameters are available (the default values are shown inside the brackets). These init parameters should be placed inside the config.xml file.

The HUMANGUI option is needed if you have a human GUI connected with an agent (and the agent should receive percepts as well).

In GOAL these settings can be changed also from the mas2g file. The CONFIGFILE option is the only needed option in the MAS file, if you use a config file.

- CLIENTIP("localhost"): client's own IP. Passed to server so server is able to find us with this IP.
- CLIENTPORT("2000"): The port that the client listens to.
- SERVERIP("localhost"): The IP address of the server.
- SERVERPORT("8000"): The Port that the server listens on.
- AGENTCOUNT("0"): the amount of agents (also specified in the launchpolicy section, see below), that the client should load. If the agentcount is higher than the amount of entities in the map then they won't be loaded. (default: 0). You can use the map editor (Chapter 9) to change the amount of entities on the map.
- LAUNCHGUI("true"): whether to launch a separate GUI for each bot (controlled by an agent or human) can be set to true or false. This GUI shows the environment from the perspective of the bot. (default:false).When this option is used, the human GUI is connected with the agent platform. A special agent in the agent platform can then communicate with the HumanGUI using entity actions that are available specifically for this purpose. This way, a human player can be a fully qualified player in the agent platform (e.g., use the native communication mechanisms in the agent platform).

- HUMANCOUNT("1"): the amount of human players that should be loaded. If the humancount is higher than the amount of entities in the map then they won't be loaded. You can use the map editor (Chapter 9) to change the amount of entities on the map.
- AGENTCLASS("nl.tudelft.bw4t.client.agent.BW4TAgent"): The java agent class to load when new entities appear.
- GOAL("true"): are we connected with GOAL? This param should be auto detected, it will be set to false if the program is started from commandline.
- KILL(""): The key we should try to use to kill the remote server.
- CONFIGFILE(""): The file from which the client reads the configuration. This is an xml file that can be generated with the scenario editor. The file is read relative to the directory containing the client environment jar file, unless you use an absolute path. It is recommended to place the xml relative to the client environment location, and not use an absolute path. WARNING: currently the scenario editor exports MAS files with absolute paths. You need to fix this manually.
- GOALHUMAN("false"): Forces the use of the human GUI with an GOAL agent to translate the commands. You should turn on this option if you have a human GUI but still want the agent to receive the percepts.
- MAP(""): The map name to be loaded. If you specify a map, the server will reset to load the new map, which disconnects all entities.
- SPEED(""): The speed (fps) at which the simulation runs on the server. Must be between 5 and 100. If left empty, the server will keep its current setting.
- LOG("ERROR"): The log4j log level to be used. Available values: OFF, FATAL, ERROR, WARN, INFODEBUG and ALL.

5 Using the Server Interface

The server interface (Figure 1) offers very limited controls, as the main control is running through software (the agents, management system, buttons and settings in the agent platform). There are only 3 controls available:

- Reset: By pressing this button, the environment is completely killed and restarted, all entities are killed (which kicks out all agents), the map is reloaded from scratch. We recommend killing from your agent platform as that offers the agent platform the opportunity to handle this take-down in a nice way.
- Enable Collisions: When enabled, bots can collide with each other. When disabled, they can run over each other without any interference. We recommend to keep this *disabled* as this option is not yet fully supported (you may have problems with path planning).
- Change Map: By selecting a map, you effectively *Reset* the system, to restart it with a different map.

6 Using the Human Player GUI

When the GUI is open (Figure 3), two main parts can be seen:

1. The map panel, where the map and the block sequence are being displayed.
2. The message panel, where the remaining bot battery and messages between bots and e-partner are being displayed. Messages can be be sent to other bots and your e-partner.

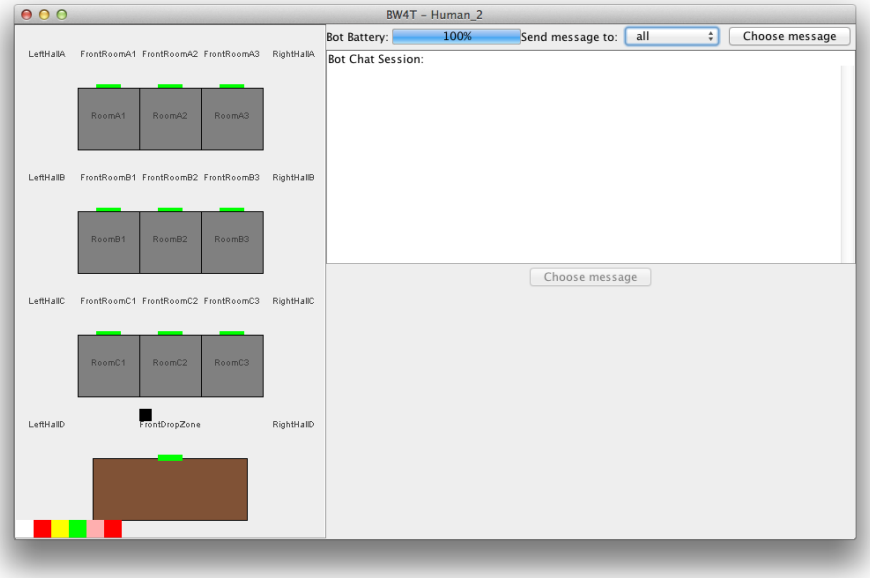


Figure 3: Human Player GUI with map panel (left) and message panel (right).

6.1 General use

Figure 3 shows the human player GUI, with the map panel in the left half, and the message panel in the right half. To use the human player interface, the user clicks with the left (occasionally the right) mouse button in the GUI. Depending on where the user clicks, different menus appear. The user then picks the appropriate action from the menu to execute that action. In the following sections the possibilities are explained.

With the map panel, the bot can be directed by clicking on the map. Different pop-up menus will appear upon clicking on different entities on the map.

With the message panel, the battery capacity of the bot that can be controlled can be found. Bots can be selected to which messages need to be sent and messages can be sent here. The bot chat session and the e-partner chat session are also displayed.

The map panel shares the message target with the message panel. So all messages that are sent from the map panel are sent using the actual message target setting in the message panel. A message will be send to all bots by default. Please refer to the Message Panel section below to adjust the receiver(s) of messages.

6.2 Map Panel

A picture of the Map Panel is shown in 4.

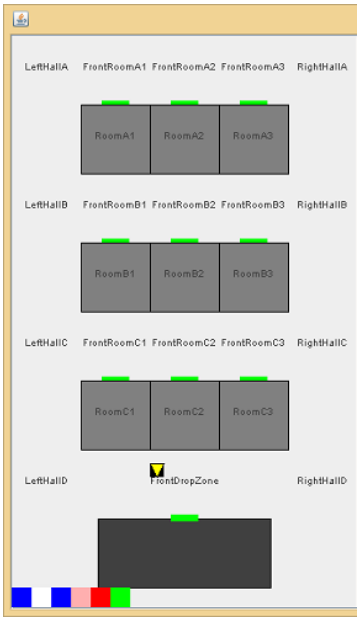


Figure 4: Map Panel

Here the following actions can be done:

- **Go to:** "*place*": To command the controlled bot to go to a room or a certain place in the corridors, click on the room or place where the bot needs to go to. The corridor menu (figure 5) will appear next to your mouse pointer with the option *go to here* or *Go to* "*room*".

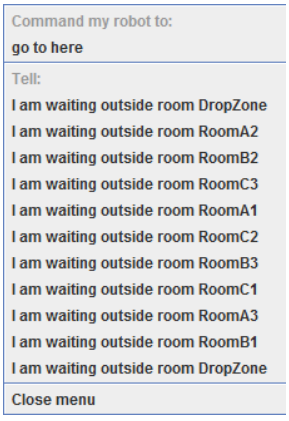


Figure 5: Corridor menu

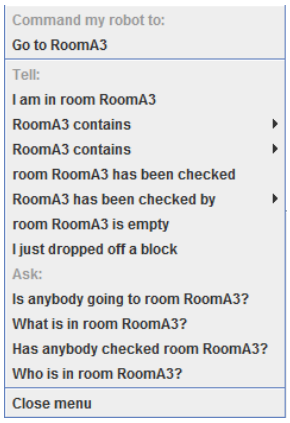


Figure 6: Room menu

- **Send message: I am waiting outside** "*room*": To send a message to a bot or to

all bots, click in the corridor. The corridor menu (figure 5) will appear next to your mouse pointer with the options *I am waiting outside "room"*.

- **Send message: I am in** "*room*": To send a message to a bot or to all bots, click in the room. The room menu (figure 6) will appear next to your mouse pointer with the option: I am in "*room*".
- **Send message: tell other bot(s)** "*information*": To send a message to a bot or to all bots with certain information about a certain room, click on the room. The room menu (Figure 6) will appear next to your mouse pointer with the messages you can send.
- **Send message: tell other bot(s) about blocks**: By clicking on a block (particularly, those below the drop zone), the user can tell a bot or all bots something about that block, or ask others for information about that block.
- **Send message: ask other bot(s)** "*question*" **about certain room**: To ask a bot or all bots a question about a certain room, click on the room. The room menu (Figure 6) will appear next to your mouse pointer with the possible questions that can we asked.
- **Pick up block**: To pick up a "*block*", click on the block that needs to be picked up. The block menu (Figure 7) will appear next to your mouse pointer. Click on *Go to* "*color*" *block*. When the controlled bot is standing on the block, click on the bot. The block menu when standing on block (Figure 8) will appear. Click on *Pick up* "*color*" *block*.

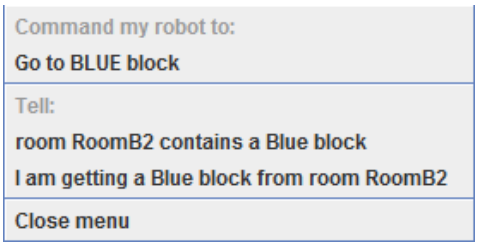


Figure 7: Block menu

Figure 8: Block menu when standing on block

- **Drop block**: To drop the block that is currently being held, click on the room in which it needs to be dropped. The room menu when holding block will appear next to your mouse pointer. This menu is almost identical to (Figure 6 but it has an extra item 'Put down block'). Click on *Put down block*. Do note that the controlled bot should be in the same room as where you want to drop the block.
- **Pick up e-partner**: To pick up an e-partner, click on the e-partner that needs to be picked up. The e-partner menu (Figure 9) will appear next to your mouse pointer. Click on $Go\ to\ e-partner$. When the controlled bot is standing on the e-partner, click on the e-partner again. The e-partner menu when standing on e-partner (Figure 10) will appear. Click on $Pick\ up\ e-partner$.
- **Send message to e-partner**: To send a message to the e-partner that is currently being held, click on the e-partner. The e-partner menu when holding e-partner (Figure

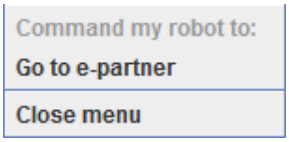


Figure 9: The e-partner menu

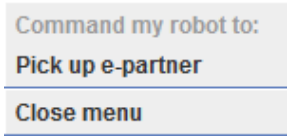


Figure 10: E-partner menu when standing on e-partner

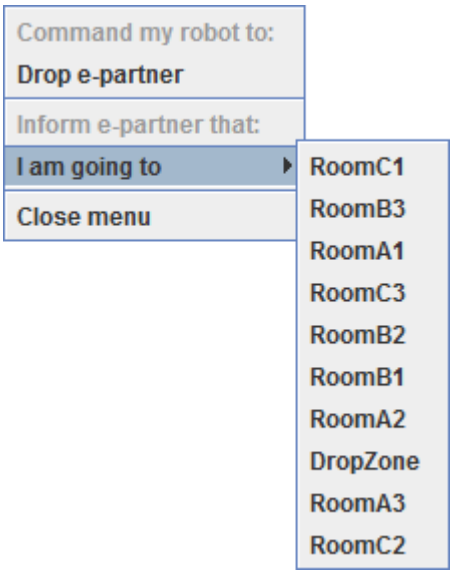


Figure 11: E-partner menu when holding e-partner

11) will appear next to your mouse pointer. Click on *I am going to* "*room*". This option will only appear if the e-partner has GPS enabled.

- **Drop e-partner**: To drop the e-partner which is currently being held, click on the e-partner. The e-partner menu when holding e-partner (figure 11) will appear next to your mouse pointer. Click on $Put\ down\ e-partner$.

6.3 Message panel

Figure 3 (right half) and 12 show the Message Panel. The second choose message button will only be enabled when the bot is holding an e-partner. Below this button, the e-partner chat session will appear when the bot holds an e-partner for the first time. Figure 12 shows the Message Panel when the e-partner has been held and dropped.

- **Select message receiver** To select who a message needs to be sent to, click on the dropdown box (Figure 13) at *Send message to* :. The receiver is by default *all*. When set to *all*, all other bots receive the message. When set to a specific bot, only that bot receives the message.

- **Send message to bot(s)** To send a message, click on the *Choose message* button. A menu (Figure 14) will appear next to your mouse pointer with the possible messages to

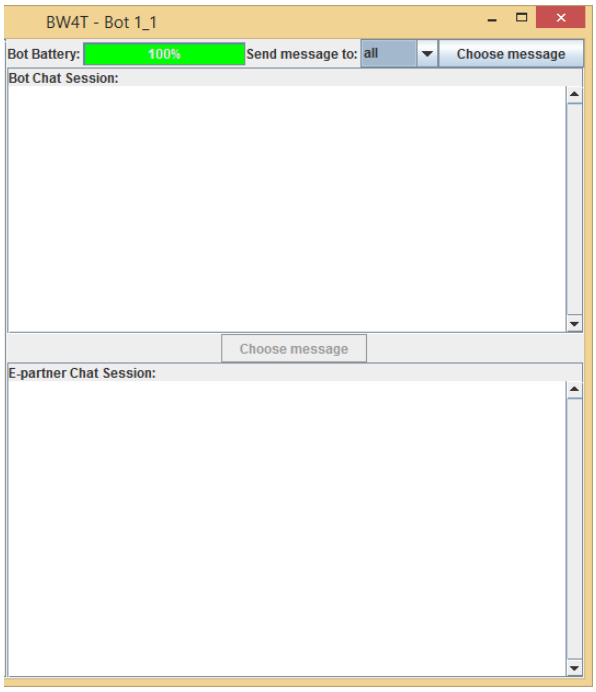


Figure 12: Message Panel with e-partner section. Refer to Figure 3 for the panel without e-partner section.

be sent.

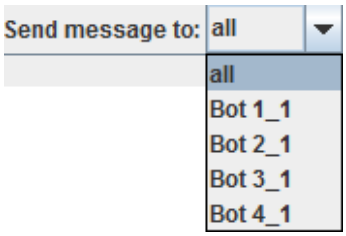


Figure 13: Select receiver dropdown

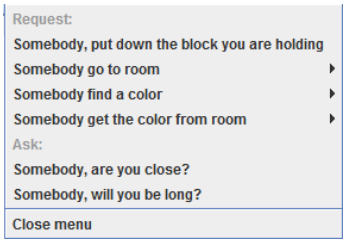


Figure 14: Choose message button menu

- **Answer a question** To answer questions asked by other bots, click in the bot chat session box. A menu (figure 15) will appear next to your mouse pointer with the possible answers to be sent.
- **Send message to e-partner** To send a message to the e-partner that is currently being held, click on the choose message button. A menu (Figure 11) will appear next to your mouse pointer. Click on *I am going to "room"*. This option will only appear if the e-partner has GPS enabled.
- **Drop e-partner** To drop the e-partner that is currently being held, click on the choose message button. A menu (figure 11) will appear next to your mouse pointer. Click on *Drop e − partner*. The choose message button will be disabled after dropping the e-partner.

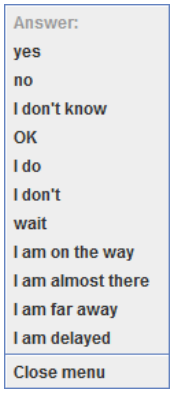


Figure 15: Bot answer menu

7 Hints for GOAL users

In this section we give some hints that are specific for using BW4T from the GOAL agent system.

7.1 mas2g file

The number of agents is specified at two places in the mas2g file. First, the agentcount and humancount specify the number of entities of the corresponding type that should be created in the environment. Second, the launchpolicy specifies which and how many agents should be connected to these entities. Make sure that the agentcount and humancount in the initialization parameters are in line with the launch policy section in the mas2g file. If you use BW4T from a batch runner, you may want to reset the server after each run. This is done by specifying a map in the mas file init parameters. See also Section 4.1.

7.2 Restarting, pausing and resuming the system

If you are running from SimpleIDE, you can pause and resume the system at any time by clicking the pause or step buttons.

When running from Eclipse, to be able to pause the system, don't run the mas2g file as described above. This time, open the mas2g file and click on the debug button (the small bug icon).

Eclipse will ask you whether you would like to go to the debug screen (see Figure 16), click yes and watch your screen adjust. Now you can see the running bots listed at the left side, and the current goals, beliefs and knowledge of the selected bot on the right side. Beneath the list the goal file belonging to the bot is shown and at the bottom of the screen, all debug actions are being run (Figure 17). To pause the system, select the bot to be paused and click on the pause button. To resume, select the bot and click resume. To restart the system, do the following

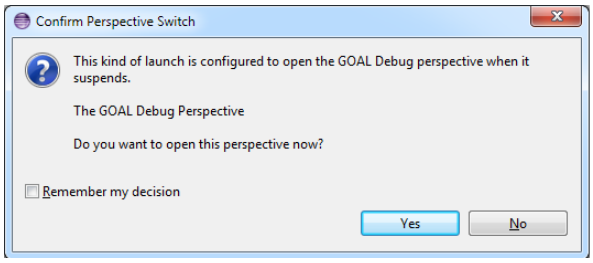


Figure 16: Choose yes

1. In eclipse, kill the MAS by clicking the red box (stop button) at the right of the console window.
2. In the server window, press Reset, or choose a new map from the Change Map menu.
3. Run the MAS in Eclipse as described above.

7.3 Running on multiple computers

BW4T server can handle multiple clients out of the box. However, if you run multiple agent systems separately, agents in these separate platforms will not see each others' agents.

To make a real distributed agent system where all agents can see each other, the agent system itself has to be run distributed.

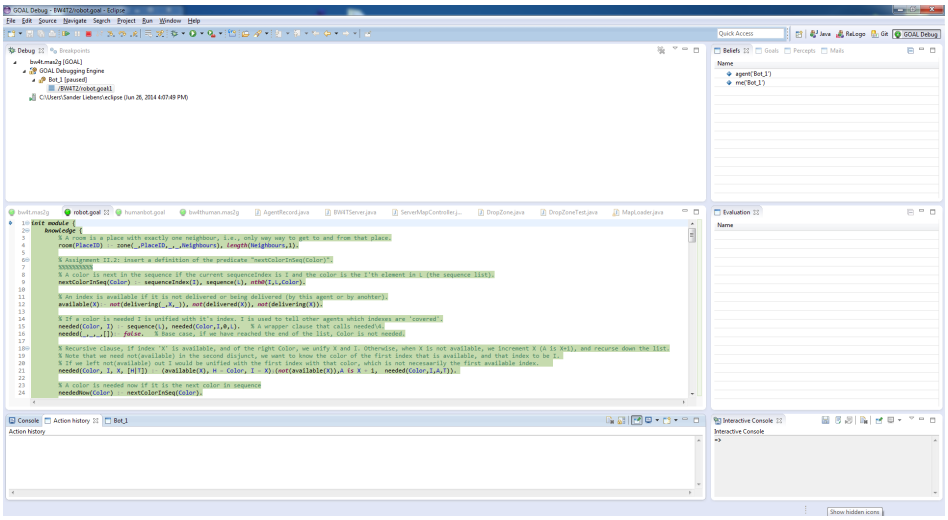


Figure 17: Debug mode

For GOAL, this is done as follows. You should designate one of these computer as the server. On this computer you can start the server as described in the first section of this guide. You must use RMI messaging (check the GOAL Run menu) to allow other GOAL runtimes to connect with the main GOAL instance that runs on the server. You need to check a few things in the MAS that you use here:

- specify a map such that the environment resets when you start up the MAS
- make sure that the map that is used has enough entities to accommodate all agents in all computers that want to connect
- make sure that the entities get the proper type, by specifying the proper agentcount and humancount.

The other computers will then function as client. Create a MAS file for each of these, and configure this MAS as follows (see also bw4thuman.mas2g in the GOALagents directory of GOAL):

- set the serverip and serverport initialization parameters to the ones that the server is listening on (default for the server is localhost and port 8000).
- Set the humancount and agentcount parameter on each client to reflect how many human or agent players that client should load.
- Use humanbot.goal for human agents
- use env = <CLIENT> /client.jar". in the environment section. Do not connect to an already running environment in another MAS. This is because BW4TClient creates GUIs for humanbots, on the machine where it is running.
- Check that the launchpolicy picks up the proper entity type, so use 'human' if you want to attach to human entities etc.

7.4 Distributed Human GUIs

This section describes how to run a set of distributed human GUIs such that they all communicate through GOAL. Before running a set of distributed Human GUIs with Eclipse, make sure that the server is installed on one machine. Furthermore, make sure that the server map can contain a sufficient number of entities. To run a set of distributed Human GUIs with GOAL, do the following:

- Start the server
- For each machine where you want to have a human GUI:
    1. Start your agent platform (SimpleIDE or Eclipse IDE)
    2. Open the MAS file of the bw4thuman.mas2g
    3. adjust the serverip to correctly point to the machine ip number where the server runs
    4. Start the MAS

7.5 Programming a BW4T Agent

To program your own BW4T agent, use the same actions as specified in the demorobot. You can choose to change the pre- and post conditions of each action.

Percepts are retrieved automatically by GOAL, see the percept specifications for what percepts you can expect.

In order for messages received by other GOAL agents to appear on the GUI of your agent, you should add the following lines to your GOAL agent's code:

1. Add the following line to your knowledge base:
   #import "messageTranslation".
2. Add the following line at the end of your goal file.
   #import "message.mod2g".
3. Add the following line to the start of your event module:
   if bel(true) then message.
   This will make sure that the message module that was just imported will be run first when entering the event module.
4. Your own message handling code should be performed after this line and should delete any messages after handling them. Otherwise they will be continuously posted to the GUI as they are not deleted by the message module. If you don't do any message handling yet you can add the following line below the one provided in step 3 to delete all received messages:
   forall bel(received(Agt,Msg)) do delete(received(Agt,Msg)).

7.6 Testing your agent

In order to test your agent you should edit the bw4t.mas2g file as follows.

- Add your agent to the agentfiles list.

- Add a line to the launchpolicy (copy the line of the demorobot and replace 'demorobot' with the name of your agent).

Make sure that you set GOAL to launch the desired number of your agent. Also make sure that the initialization parameter of agentcount is not set to 0 as then only humanbots will be loaded.

Note that besides in the bw4t.mas2g file, the number of agents is also specified in the map that you use. This number determines the maximum amount of bots that can appear on the map. If the maximum number of agents to be launched as specified in the bw4t.mas2g file is bigger than the number of agents specified in the map, some of the agents will not appear on the map and not be part of the team.

8 Log file

Repast logs the following for each run into a file. The filename is "BW4TXXXX.log" where XXX is the date and time to make the filename unique. The file is saved in the <SERVER> /log directory. It contains:

1. sequence: goal sequence (which block colors are to be dropped)
2. room: initial blocks per room
3. action: log of each action of a bot, with timestamp
4. total time: total time to complete task. Begin time is determined by first incoming action. End time is determined by the last block of the sequence dropped.
5. agentsummary: for each agent:
    - the bot type containing its handicaps
    - # correct drops in dropzone
    - # incorrect drops in dropzone
    - total time of standing still
    - # messages to other agents
    - # rooms entered

Logs are written to the file as soon as possible, so that information won't get lost when the system is being killed before the end of the sequence is reached.

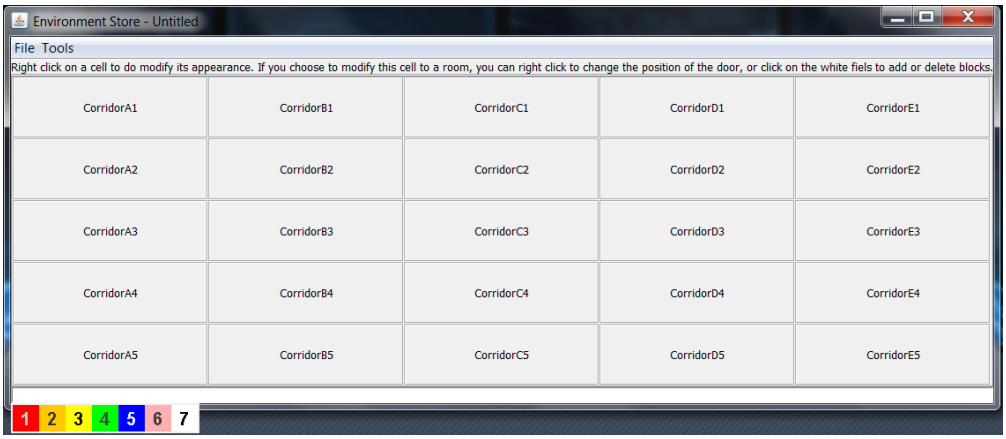


Figure 19: the map editor with a 5x5 freshly created map

9 Map Editor

This section describes how you can edit existing maps or create new maps from scratch. This can be done with a editing tool, or by directly editing the map files manually.

9.1 Using the Environment Store

The Map Editor is a tool for editing maps for the BW4T server. Double click the jar file environment-store.jar in the <SERVER> directory to start the editor.

After starting up, the map size dialog (Figure 18) appears, which can be used as follows. Here you can choose the amount of rows and columns you want the environment to consist of. Each cell in the "table" can be used to fulfill one piece of the map. This could either be a room, a corridor, a dropzone, a charging zone, start zone or a blockade. By pressing the colours at the bottom you can generate the sequence to be completed. This can also be done by pressing the according numbers on the keyboard.

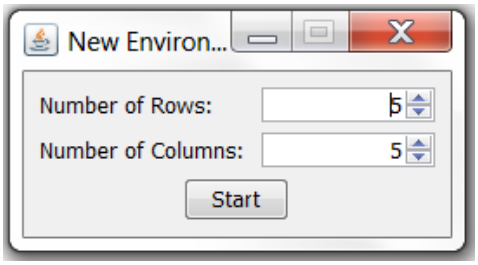


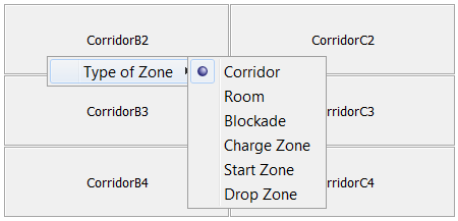
Figure 18: Choose the amount of columns and rows

At the moment, for an existing map to be loaded in the map editor, the correct sizes of the map have to be entered before the map can be loaded.

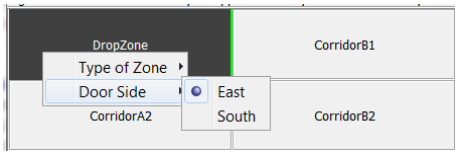
In the Map editor GUI above you can add blocks to rooms. To add blocks to rooms, right click in the cell, adjust the type of the zone to Room. Now you can add blocks the way you did with the sequence. Right clicking the cell again lets you choose at which side of the room you want the door. When you're done, press save. By pressing Tools in the menu bar, you are able to randomize zones, blocks and sequences.

WARNING: Randomizing these will not always guarantee that the sequence can be completed. By pressing File in the menu bar, you are able to watch a preview of your map, and you are able to save the map. Save the map in the <SERVER> /maps folder [1].

[1]This map directory appears after the first run of the server.



(a) the drop down menu generated when a zone is right clicked



(b) the drop down menu generated when a room is right clicked

Figure 21: Drop Down menus

Figure 22: The rectangle at the bottom of both the map editor and the rooms.

9.2 Map editor

After specifying the amount of rows and columns the map should have, and clicking on the Start button, the map editor is launched. Figure 19 shows the environment store after launch, with a 5x5 map.

The map editor has many functionalities. The functionalities are detailed bit by bit in this documentation.

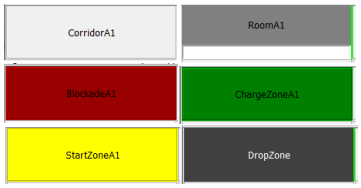


Figure 20: the color palette at the bottom of the map editor

9.3 Editing a room

Right click on any corridor (or any zone of the matrix for that matter), and the drop down menu is shown (Figure 21a).

This allows a zone to be converted to any one of a corridor, a room, a blockade, a charge zone, a start zone, and a drop zone. Figure 20 shows the different appearances of the zones in the map editor.

As can be seen, the room and drop zone contain a green border. This is to indicate which way the door to enter the room is faced. This is also editable. When right clicking on a certain room, a drop down menu pops up (Figure 21b). This drop down menu lists the directions the door can face while still being accessible. In this case, as it is the top left zone, only doors facing east or south are possible.

9.4 Editing a sequence

Under the zones of the map editor is a white rectangle that contains the sequence for that map. This same rectangle is also present in the rooms that aren't drop zones (Figure 22).

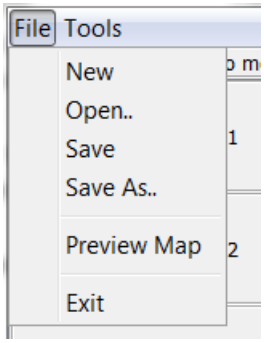
The rectangle in the rooms can be used to enter colors, which are then used in the blocks that are placed in that room. The rectangle under the zones can be used to enter the colors that make up the sequence that has to be completed for that map. To add colors, the color palette is useful.

9.5 Color palette

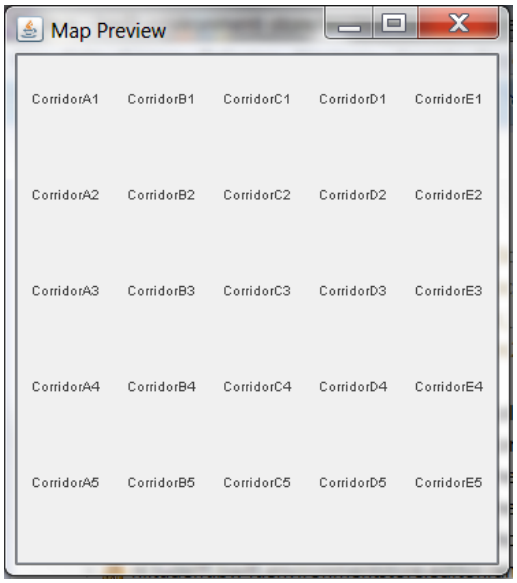
At the bottom of the map editor is the color palette. A larger picture is given in Figure 23.

Figure 23: the color palette at the bottom of the map editor.

The color palette is used for adding blocks to rooms and the sequence to be completed in the map. When clicking on the box containing either the sequence of the map or the blocks in the room, the color palette is shown. It allows for the adding of colors by clicking on the respective colors on the palette, but colors can also be added by entering the respective numbers in the colors (pressing 1 adds a red block to the room or sequence, for example) and also by entering the first letter of the color (so pressing R adds a red block).



(a) the drop down menu created when the 'File' menu item is clicked.



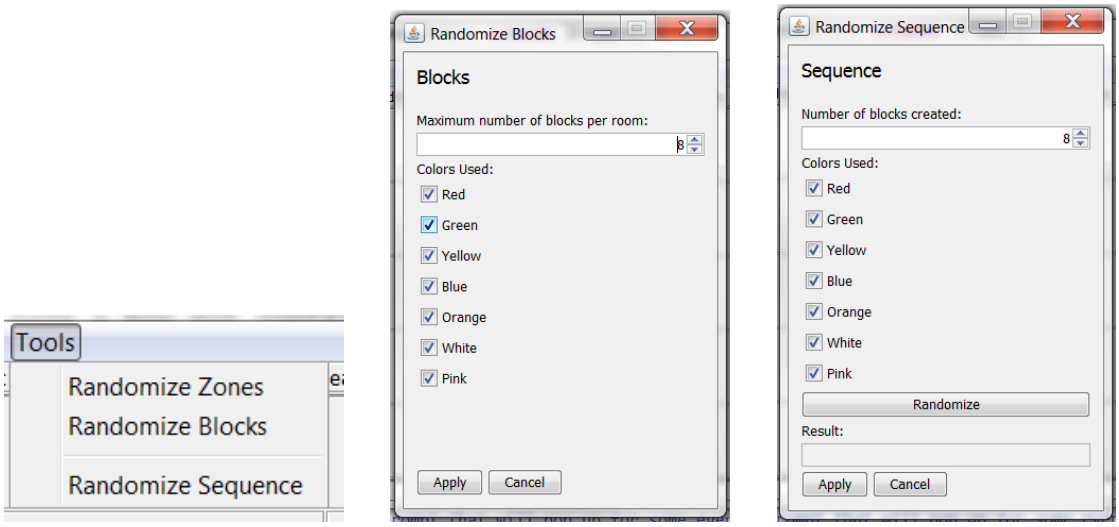
(b) The preview of a 5x5 non-edited map, with no blocks in the sequence.

Figure 24: Drop Down menus

9.6 File menu

The file menu (Figure 24a) contains not only the standard operations for exporting a created map to a file, so that it can be used later, but also a few other options.

The New-, Open...-, Save- and Save as...-options are the standard file operations and need no further explanation. However, as added functionality to the options, it is impossible to save a map not containing either a start- or drop zone, as well as a map not containing a block in its sequence, and saving an unsolvable map will give a warning describing what's wrong. The Exit-option also needs no further explanation. The Preview-option gives a preview of the created map, as how it would be rendered when used as an actual map for bots. For a non-edited 5x5 map, the preview looks as shown in Figure 24b.



(a) The drop down menu created when the 'Tools' menu item is clicked.

(b) The menu appearing when the randomize blocks option is clicked in the menu.

(c) The menu appearing when the randomize blocks option is clicked in the menu.

Figure 25: menus

9.7 Tools menu

The other drop down menu at the top of the map editor is the tools menu (Figure 25a). This menu contains options to use several tools.

Clicking on the 'Randomize Zones' option creates a random map, based off a vanilla map with reclassification of rooms to blockades, charge zones or corridors, and with reclassifications of existing corridors to charge zones. The map is always solvable, given that all the blocks in the sequence are present in the map and no further changes are made to the map. The 'Randomize Blocks' option randomizes the blocks contained in every room according to parameters that the user has to specify. Figure 25b shows the menu.

This interface allows the user to specify which block colors they want to be spawned, and also the max. amount of blocks in a room. The interface for random sequence generation is slightly different (Figure 25c).

The differences between this interface and the previous interface are the Randomize button and Result label, that create and display resp. the random sequence, and the apply and cancel button which enter the sequence in the map or cancel the creation of the sequence, respectively.

9.8 Manual editing of Maps

It is also possible to manually edit a map as it is a plain text XML file.

1. Copy an existing map file in the <SERVER> /maps folder to a new file [1].

2. Open the copied map with a text editor. You can then edit the colors in the rooms. Each <blocks>COL< /blocks> line inside a <zones> of type ROOM adds another block to the room. The currently available colors are BLUE, ORANGE, RED, WHITE, GREEN, YELLOW AND PINK. A room has place for at most 10 blocks.

3. You can change the number of rooms by adding or removing <zones> of type ROOM and position the rooms correctly on the map by editing the <x> <y> <width> and <height> inside the room's <boundingbox>. Also you need to add a <door> properly positioned on the border of the room.

4. You can edit the goal sequence by adding <sequence> items to the map, with colors as mentioned above.

5. You can prevent multiple entities to enter corridor zones by putting true in the item <oneBotPerCorridorZone> in the map.

6. You can add entities to the environment by creating more <entities> items. Make sure they have a unique <name> and that their start position is in a different <zone> if you have <oneBotPerCoridorZoneidorZone> set to true.

7. You can let the server pick a random sequence of a given length by setting the <randomSequence> to a positive value. These random blocks are added to the existing sequence items and the random blocks are placed randomly on the map.

8. You can let the server pick random extra blocks to be placed in the rooms on the map by putting a positive number in the <randomBlocks> in the map. Note that this addition is on top of all blocks that are already placed in the map; so normally you leave the room zones empty when using this option.

9. You can set per-zone visibility of the zone namelabel in the server map renderer, using a <renderOptions> <labelVisible> false < /labelVisible> < /renderOptions> block to disable visibility.

10. Save the map in the <SERVER> /maps directory.

11. Edit the map initialization parameter for the server to your new map file. See the section on customizing the server settings. If you now start the server and your new map should be loaded.

10 Scenario Editor

10.1 Introduction

This chapter describes how to make use of the Scenario Editor (Figure 26). When you open the editor, you see two main parts:

1. The Configuration panel, which is used to create, open and save configuration files.
2. The Entity panel, where a list of bots and a list of e-partners are being displayed. Here you can create, modify, rename and delete bots and e-partners.

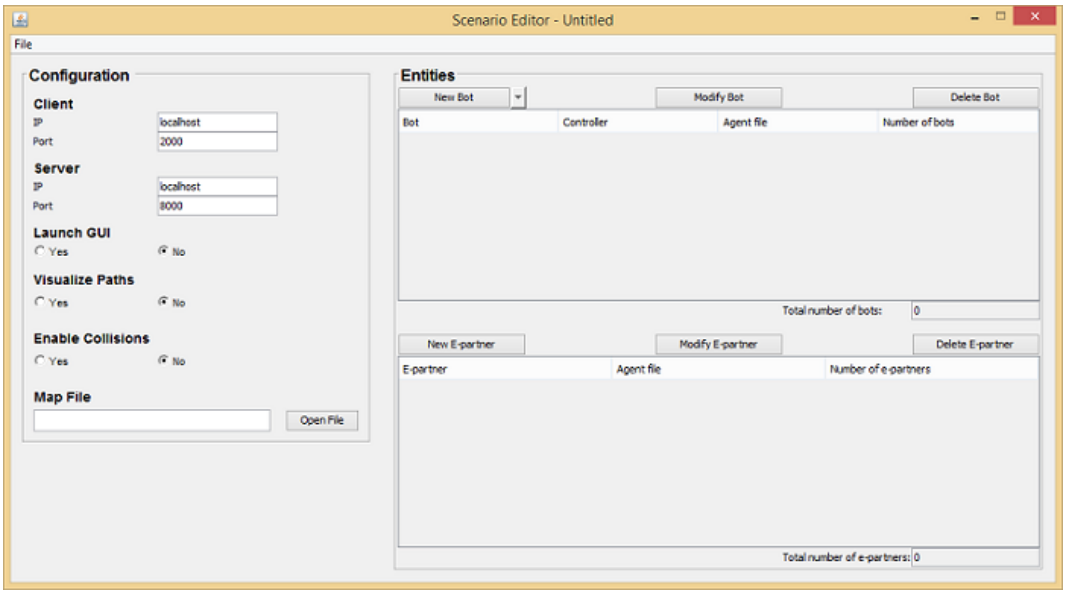


Figure 26: Scenario Editor

10.2 General use

At the top of the editor you can see a *File* menu. If you click on that, you get the options to create a new configuration, open an existing configuration, save your configuration, export your configuration and to exit the editor.

On the left side of the editor you can configure a scenario as you like. The Client IP, the Client Port, the Server IP and the Server Port should already contain the default values. You can change them if you need to. You can also indicate whether you want to open a GUI, visualise the paths the bots take or enable collisions. If you have a map file you want to use, you can import it by selecting the *Open file* button at the right section.

On the right side of the editor you can create, modify, rename and delete bots and e-partners. There also are a list of bots and a list of e-partners that are created, and below each list there is an indication of how many bots or e-partners are created in total.

Linked to the Scenario Editor are the Bot Store and the E-partner Store, which will also be discussed in this document.

10.3 Configuration Panel

A picture of the configuration panel can be found in Figure 27. You have the following options:

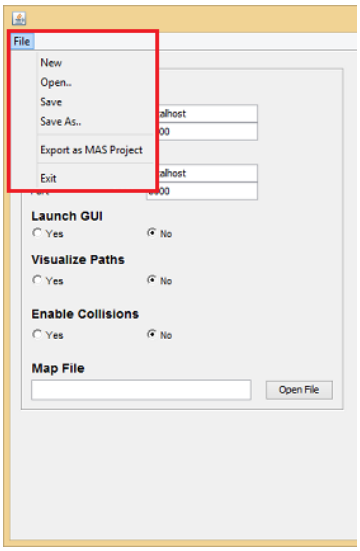


Figure 27: Configuration Panel

- New configuration file: To create a new configuration file, select *File* → *New* in the menu bar. You will now get a new configuration with the default values. Creating a new configuration will reset all previous changes you have made, so make sure you save your current configuration first if you want to keep it.

- Open configuration file: To open an existing configuration file, select *File* → *Open* in the menu bar. A window will pop up where you can select the folder where the configuration file is saved to. Once you have selected the right folder, select the file and click the *Open* button.

- Save configuration file: To save your configuration file, select *File* → *Save* in the menu bar. If your file hasn't been saved before, a window will pop up where you can select the folder you want to save your configuration file to and enter the file name you want to use. Once you have selected the right folder and entered the file name, click the *Save* button.

- Save configuration file as: To save your configuration in a new file, select *File* → *Save As* in the menu bar. A window will pop up where you can select the folder you want to save your configuration file to and enter the file name you want to use. Once you have selected the right folder and entered the file name, click the *Save* button.

- Export configuration to mas2g: To export your configuration to mas2g, you have to have saved your configuration first. Once you have done that, select *File* → *Export as MAS project* in the menu bar. A window will pop up where you can select the folder you want to export the configuration to and the file name you want to use. Once you have selected the right folder and entered the file name, click the *Export MAS project* button.

  WARNING: we recommend to use a relative file path to the configuration file. But currently, the generated mas2g file will contain an absolute path reference to the config file. We strongly recommend to fix the file reference in the mas2g file manually for cross-platform compatibility (so that others can also run your *MAS*).

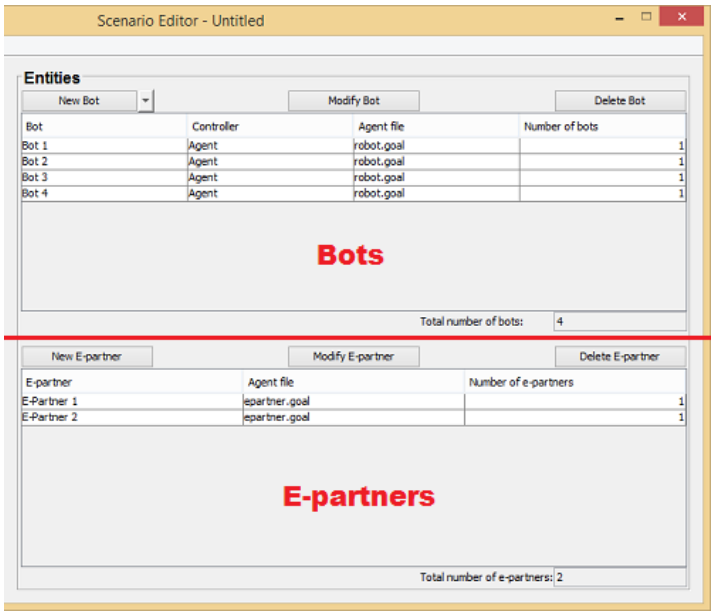


Figure 28: Entity Panel

- Exit the Scenario Editor: If you want to close the Scenario Editor, select *File* → *Exit* in the menu bar. Closing the Scenario Editor will not save any changes you have made, so make sure you save your configuration first if you want to keep it.

10.4 Entity Panel

The entity panel exists of 2 parts; the top part shows the bot options and the bottom part shows the e-partner options. A picture of the entity panel can be found in Figure 28. You have the following options:

- Create new bot: To create a new bot, click the *New Bot* button. A new bot will appear in the bot list.
  If you want to create a standard bot (which already has default values), click on the arrow next to the *New Bot* button. A menu will drop down where you can select one of the standard bots. If you click on one of the standard bots, the bot will appear in the bot list.

- Modify bot: To modify a bot, select the bot you want to modify in the bot list and click the *Modify Bot* button. The Bot Store will open in a new window where you can modify the properties of the bot.

- Rename bot: To rename a bot, you can either use the *Modify Bot* button to rename in the Bot Store, or you can rename in the Scenario Editor. To rename in the Scenario Editor, select the bot you want to rename in the list with bots. Double click the its name and enter a new name.

- Change controller type: To change how a bot is controlled, you can either use the *Modify Bot* button to change it in the Bot Store, or you can change it in the Scenario Editor. To change the controller type in the Scenario Editor, select the bot you want to change and click on its controller type. You should now be able to choose the controller type.
- Change the amount of entities of a bot: To change how many entities of a type of bot you want, you can either use the *Modify Bot* button to change it in the Bot Store, or you can change it in the Scenario Editor. To change the entity amount in the Scenario Editor, select the bot that you want to change the amount of. Now double click the amount given, and enter a new number.
- Delete bot: To delete a bot, select the bot you want to delete in the list with bots and click the *Delete Bot* button.
- Create new e-partner: To create a new e-partner, click the *New E-partner* button. A new e-partner will appear in the e-partner list.
- Modify e-partner: To modify an e-partner, select the e-partner you want to modify in the e-partner list and click the *Modify E-partner* button. The E-partner Store will open in a new window where you can modify the properties of the e-partner.
- Rename e-partner: To rename an e-partner, you can either use the $Modify\ E-partner$ button to rename in the E-partner Store, or you can rename in the Scenario Editor. To rename in the Scenario Editor, select the e-partner you want to rename in the list with e-partners. Double click its name and enter a new name.
- Change the amount of entities of an e-partner: To change how many entities of a type of e-partner you want, you can either use the $Modify\ E-partner$ button to change it in the E-partner Store, or you can change it in the Scenario Editor. To change the amount in the Scenario Editor, select the e-partner that you want to change the amount of. Now double click the amount given, and enter a new number.
- Delete e-partner: To delete an e-partner, select the e-partner you want to delete in the e-partner list and click the *Delete E-partner* button.

10.5 Bot Store

Figure 29 shows a picture of the Bot Store. You have the following options:

- Rename bot: To rename a bot, click on the field that contains its current name. Now you are able to edit the current name or you can enter a new name.
- Change controller type: To change how a bot is controller, you can click on its current controller type. Now you are able to choose the controller type.
- Change the amount of entities: To change how many entities of the bot you want, you can click on the field that contains the current number of entities. Now you are able to edit the amount.

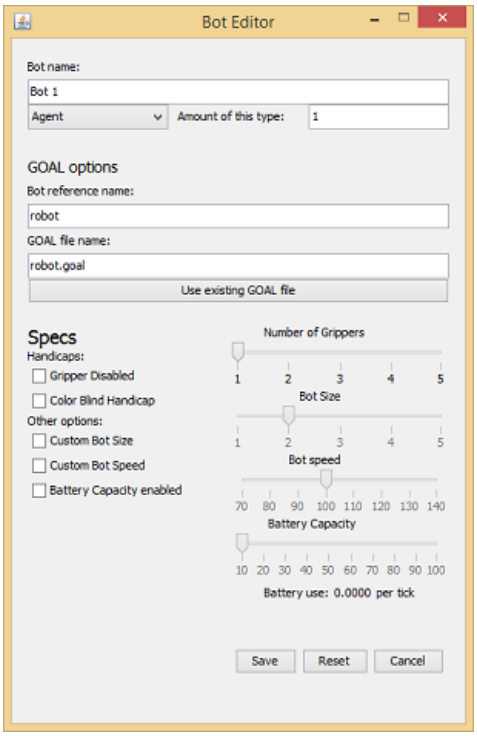


Figure 29: Bot Store

- GOAL options: You can enter the GOAL options for your bot under the *GOAL options* section. Here you can indicate what its reference name in GOAL should be, and you can select a GOAL agent file which will control the bot. To select the GOAL agent file, you can click on the *Use existing GOAL file* button. A window will pop up where you can select the folder where the GOAL file is saved to. Once you have selected the right folder, select the file and click the *Open* button.
- Bot properties: You can find the available bot properties under the *Properties* section. To change what properties you want the e-partner to have, you can select or deselect the checkbox next to the property description. Once you have enabled a property, you can use the sliders to change the value of that property.
- Save modifications: If you want to save the changes you have made to the bot, you can click on the *Save* button. The Bot Store window will close, and your changes will have been saved.
- Reset modifications: If you want to reset the changes you have made to the bot, you can click on the *Reset* button. Your changes will have been reverted to the last saved configuration.
- Cancel modifications: If you want to cancel modifying the bot without saving any changes, you can click on the *Cancel* button. The Bot Store window will close.

10.6 E-partner Store

WARNING: The e-partner component of BW4T is not yet completely ready for use.

A picture of the E-partner Store is shown in Figure 30. You have the following options:

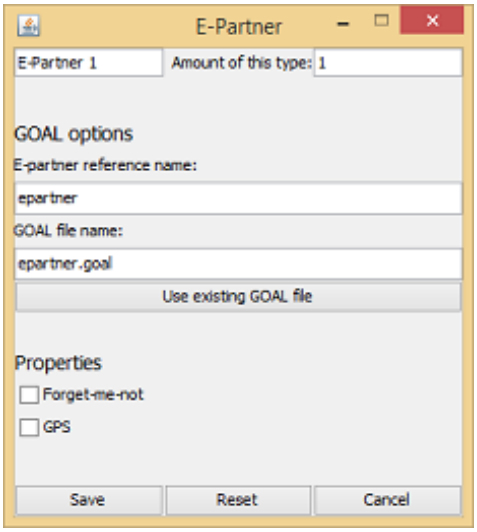


Figure 30: E-partner Store

- Rename e-partner: To rename an e-partner, click on the field that contains its current name. Now you are able to edit the current name or you can enter a new name.
- Change the amount of entities: To change how many entities of the e-partner you want, you can click on the field that contains the current number of entities. Now you are able to edit the amount.
- GOAL options: You can enter the GOAL options for your e-partner under the *GOAL options* section. Here you can indicate what its reference name in GOAL should be, and you can select a GOAL agent file which will control the e-partner. To select the GOAL agent file, you can click on the *Use existing GOAL file* button. A window will pop up where you can select the folder where the GOAL file is saved to. Once you have selected the right folder, select the file and click the *Open* button.
- E-partner properties: You can find the available e-partner properties under the *Properties* section. To change what properties you want the e-partner to have, you can select or deselect the checkbox next to the property description.
- Save modifications: If you want to save the changes you have made to the e-partner, you can click on the *Save* button. The E-partner Store window will close, and your changes will have been saved.
- Reset modifications: If you want to reset the changes you have made to the e-partner, you can click on the *Reset* button. Your changes will have been reverted to the last saved configuration.

- Cancel modifications: If you want to cancel modifying the e-partner without saving any changes, you can click on the *Cancel* button. The E-partner Store window will close.