SORTS Tech Report

James Irizarry, Sam Wintermute, Joseph Xu

July 21, 2006

0.1 ORTS Overview

The Open Real Time Strategy software is a highly configurable game engine used to play real time strategy (RTS) games [**?**]. The main purpose for ORTS is to serve as an open source, open interface RTS game engine for RTS AI tournaments. ORTS is undergoing active development as of July 2006 at the University of Alberta under the direction of Michael Buro.

There are several reasons why ORTS is especially suitable for use in AI tournaments. It has a (relatively) straightforward C++ API, making interfacing with your favorite AI system easy. All the specific game mechanics, ranging from types of units, actions, and physics, are specified via C++ style scripts called blueprints. This means that ORTS can be easily configured to simulate a wide range of environments, from arbitrarily simple ones like Wumpus World to complex ones like Starcraft. Finally, ORTS has a client/server architecture in which the server maintains the state of the world and only report to the clients information they are supposed to have for a fair game. This is in contrast to most commercial RTS games, in which each client maintains the entire world state and prevents the player from accessing forbidden information such as other players' locations only by hiding them from the GUI. The result is that ORTS is impervious to "memory hack" cheats that are widespread in commercial RTS games. This feature is particularly important if tournaments are to be run across the Internet.

0.2 SORTS Overview

SORTS is a piece of software that allows the Soar cognitive architecture to act as a client to the ORTS game server, so that ORTS game playing agents can be written in and executed on Soar. SORTS is much more than an interface bridge in that it intentionally constrains the space of possible Soar agents in important ways and also handles aspects of low-level game control that are not suitable for Soar. In particular, SORTS heavily abstracts the world state information obtained from the ORTS API before feeding them into Soar as perceptions, and also interprets and executes high-level commands from Soar as low-level actions sent to the ORTS API.

0.3 Soar IO Description

0.3.1 The SORTS input-link

There are five top-level attributes on the SORTS input link, "groups", "game-info", "feature-maps", "vision-info", and "query-results". The groups, feature-maps, and vision-info structures are all part of the main visual system (see XXX), while game-info contains higher-level information about the game world, and query-results is used to communicate the results of specialized queries from Soar to the middleware.

The exact data structures are as follows:

Attributes of io.input-link	
attribute	description
vision-info	Contains information on the current state of the vision system.
vision-info.center-x vision-info.center-y	The coordinate of the center of the region in view.
vision-info.focus-x vision-info.focus-y	The coordinate of the center of focus (spotlight of attention).
vision-info.num-objects-visible	The maximum number of objects (groups) present on the input-link. All other objects within the view window are present in feature maps.
vision-info.grouping-radius	All objects of the same type (except as below) and owner within this distance of each other are in the same group (set to 0 for individuals).
vision-info.owner-grouping	Ignore type when grouping, only group by owner (1 if enabled, 0 if disabled).
groups	The set of groups being attended to.
groups.group	Multi-valued, one instance for each group. Detailed below.

Attributes of io.input-link.groups.group objects		
attribute	which groups	description
num-members	all groups	how many individuals comprise the group.
type	all groups	the type of the group (ex: worker, mineral).
x-pos y-pos	all groups	the x,y location of the center of gravity of the group.
x-min x-max y-min y-max	all groups	the bounding box of the group.
health	all groups	the sum of the health of all units in the group.
taking-damage	all groups	the number of members of the group currently taking damage (under attack).
shooting	all groups	the number of members of the group currently attacking an enemy.
speed	all groups	the average speed of the group.
heading	all groups	the average heading of the group.
dist-to-focus	all groups	the distance from the center of gravity of the group to the attentional focus point.
dist-to-query	all groups	the distance from the center of gravity of the group to the last query location.
owner	all groups	the player number of the group's owner.
enemy	all groups	1 if the group belongs to an enemy player, 0 otherwise.
sticky	friendly groups	1 if the group is sticky- sticky groups remain together even if they are no longer spatially close.
command	friendly groups	The last command issued to the group ("none" if no command has been issued).
command-running	friendly groups	The number of members of the group currently executing a command.
command-success	friendly groups	The number of members of the group that successfully completed the last command.
command-failure	friendly groups	The number of members of the group that unsuccessfully completed the last command.