TODO: A Survey of StarCraft AI Techniques

FirstName LastName, *Member, IEEE,* Jim Raynor, *Fellow, RR,* and Sarah Kerrigan, *Life Fellow, ZS*

***Abstract*—TODO**

***Index Terms*—review, RTS, StarCraft, machine learning, planning, TODO ...**

I. Introduction

STARCRAFT AI competitions have caused many AI techniques to be applied to RTS AI. We will list and classify these approaches, explain their power and their downsides and conclude on what is left to achieve human-level RTS AI. TODO (test [?])

II. Challenges, why is it hard to do a good RTS AI?

Here we can use Buro's 2003 paper as a starting point. Much has changed since, so we should update, and put his predictions in perspective

III. AI Techniques review

A. Overview

B. Case study 1: EISBot

C. Case study 2: NOVA

D. Case study 3: BroodwarBotQ

IV. AI Architectures for RTS AI

[SANTI: This section is WIP, so, don't read it yet :)]

Playing an RTS game involves dealing with all the problems described above. A few approaches, like CAT [?], Darmok [?] or XXX [?] (Which was that system by Stuart Russell that played Warcraft?) try to deal with the problem in a monolithic manner, by using a single AI technique. This resembles approaches to solve other games, such as Chess or Go, where a single game-tree search approach is enough to play the game at human level. However, none of those systems aims at achieving near human performance. In order to achieve human-level performance, RTS AI designers use a lot of domain knowledge in order to divide the task of playing the game into a collection of sub-problems, which can be dealt-with using individual AI techniques (as discussed in the previous section). Thus, an integration architecture is required to put together all of those techniques into a single coherent system, which is the focus of this section.

...

Present some representative examples (Figure 1), which have slightly different architectures. I could mention THANATOS as something different using learning, and Darmok as a monolithic architecture that aims at learning how to play the game, without hoping to play at human level.

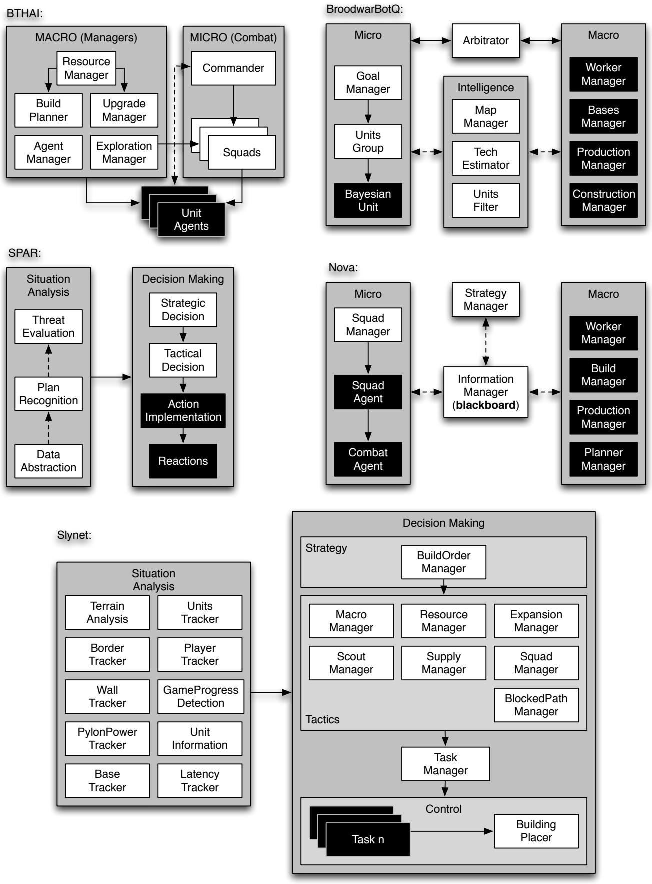

Fig. 1. General architecture of 5 Starcraft AI bots. (This figure is unreadable as of now, but we'll figure out a way to better present this info)

Then, draw some general conclusions:

- two types of aggregation, hierarchical, and decentralized (bidding?). Hierarchical is multiscale, with actions at different levels of abstraction.
- high level strategy is hard-coded (unlike in games like Chess or Go): different expressivity of the high-level strategy of the different systems, compare them. It's like a very high level programming language for RTS bots.

V. Discussion, what is left to do / open challenges

VI. Conclusion

Acknowledgment

FirstName LastName is with the Department of Names GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).

J. Raynor and S. Kerrigane are with the Romeo&Juliet Inc.

Manuscript received April 19, 2499; revised January 11, 2500.

Jim Raynor Jim Raynor was a Confederate marshal on Mar Sara at the time of the first zerg incursions on that world. He is now with Raynor's Raiders Inc.