

Multi-agent Architecture for RTS

Case Study: Glest Game

¹Esraa.A.Abdelhamid , ²Safaa. E. Amin , ³ Mostafa M. Aref

Computer Science Department

Faculty of Computer and Information Science Ain Shams University
Cairo, Egypt

Abstract

This study proposes a multi-agent architecture in Real Time Strategy (RTS) games. This architecture considers different artificial intelligence strategy levels in RTS game. Every strategy level has its' working agents. RTS games have complex and dynamic nature and real time environment. Artificial intelligence planning research in such domain is a rich environment. The need for more computer intelligence game playing provides more challenging for human players. Multi-agents architecture has been proposed in order to compete in game playing. This architecture provides a good background to researchers to understand hierarchy of different strategy levels.

1. Introduction

Research in artificial intelligence may take advantage of the wide variety of problems that videogames offer, such as adversarial planning, real-time reactive behaviors and planning, and decision making under uncertainty [1]. RTS games, which are a portion of the whole videogames, are being used as testbeds and frameworks for brand new artificial intelligence techniques [2].

RTS games are kind of strategy games that work in real time. Artificial intelligence (AI) is the science that makes intelligent computer programs that acts like human. Artificial intelligent researchers concern with RTS games as it provides a rich and challenge environment for research. RTS games are based on getting some kind of resources to build up an army and destroy your enemy. RTS games are simply “harvest, build and destroy” [3]. RTS games are a genre of computer war games [4]. RTS games are games in which several players choose races and struggle against enemy factions by harvesting resources, producing buildings and units, and fighting one another in order to set up economies, improve their technological skill and level, and win battles [5]. Environment of RTS games is dynamic, hostile, and smart [6]. Winning the game necessitates executing a strategy by placing game-playing units in a spatial environment and giving them tasks to do at the right time [7]. The best AI systems for RTS games still perform poorly against good human players. Hence, the research community is focusing on developing RTS agents to compete against other RTS agents to improve the state-of-the-art. Such competition has sparked increased interest in RTS game AI research [8].

There are three levels of strategies: high level strategy, medium level strategy and low level strategy. In high level Strategy, AI player decides whether to attack fast, build strong economy or go fast for research. Medium-level strategy is used to group units into fighting elements and control them in a larger war sense [9]. The primary role is to make sure that the attack takes all units strengths and weaknesses into consideration to make sure that the army is not wasted where the individual unit behavior is the same for both human players and computer players [10]. Low level strategy works as executer leader to medium level and takes unit intelligent behavior into consideration as shown in figure 1.



Figure 1: AI level Strategies Hierarchy

RTS games need better strategic team leadership than individual-unit intelligence that outdoes the human in speed. Most games rely on the computer's ability to quickly micromanage attacking units on an individual basis. Good strategic systems in RTS games will go a long way toward making the AI in these games more human and more fun to play against [9].

A multi-agent system is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems that are difficult or impossible for an individual agent [11]. They offer complex and partially unknown environments in which main focus is on high-level decision making. Strategy selection is a complex problem in which a human may be very good but a computer is not [12]. Current research in game playing agents for RTS games has focused on the details of unit micromanagement, or on high level strategy that leave tactics and micro-management to the built-in unit AI. The failure to build integrated agents has resulted in agents not be able to play the game at competitive levels [13].

The paper proposes an integrated multi-agent architecture in GLEST game. First, GLEST game will be explained. Second, the proposed multi-agent architecture will be discussed. This architecture will be applied in different AI Strategy levels in GLEST. Third, the conclusion of this work will be presented.

2. GLEST Game

GLEST is an open source RTS where your goal is to build up an army and defeat your foe [14]. GLEST is a free 3D RTS game, where you control the armies of two different factions: Tech, which is mainly composed of warriors and mechanical devices, and Magic, that prefers mages and summoned creatures in the battlefield [15]. **Tech Faction:** Uses human warriors and mechanical devices. **Magic faction:** Most of their units are morphed or summoned from others [5]. Glest game is shown in figure 2.



Figure 2: Glest game

In this work, Tech tree will be used in multi-agent architecture.

3. Proposed Multi-agent (MAS) Architecture

MAS is a group of agents working together towards a common goal. MAS is used in problems where single agent is difficult to solve. The effectiveness of AI in video games depends heavily on how well game characters are able to cooperate and react to the opponent player [16]. RTS games have complex and dynamic nature of different actions working together in Real Time, The need of MAS architecture has arised.

Proposed AI Strategies hierarchy are high level strategy, medium level strategy and low level strategy. Every level has different agents according to their functionality. At high level there is an agent who takes highest level decision whether to attack fast or wait to build strong defense ... etc. Medium level translates highest level decision into strategy that takes all units' strengths and weaknesses into consideration. Medium level has some agents. It has Resources Obtainer, Military Units

Organizer, City Builder and Research Upgrader as shown in figure 3 and figure 4.

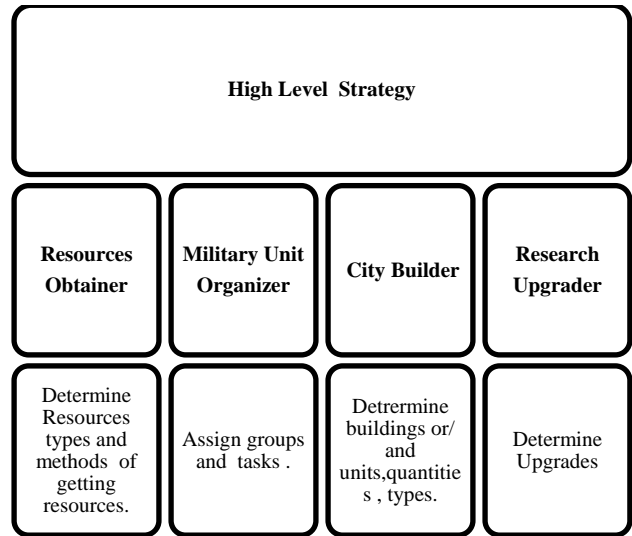


Figure 3: Proposed High and Medium levels MAS architecture

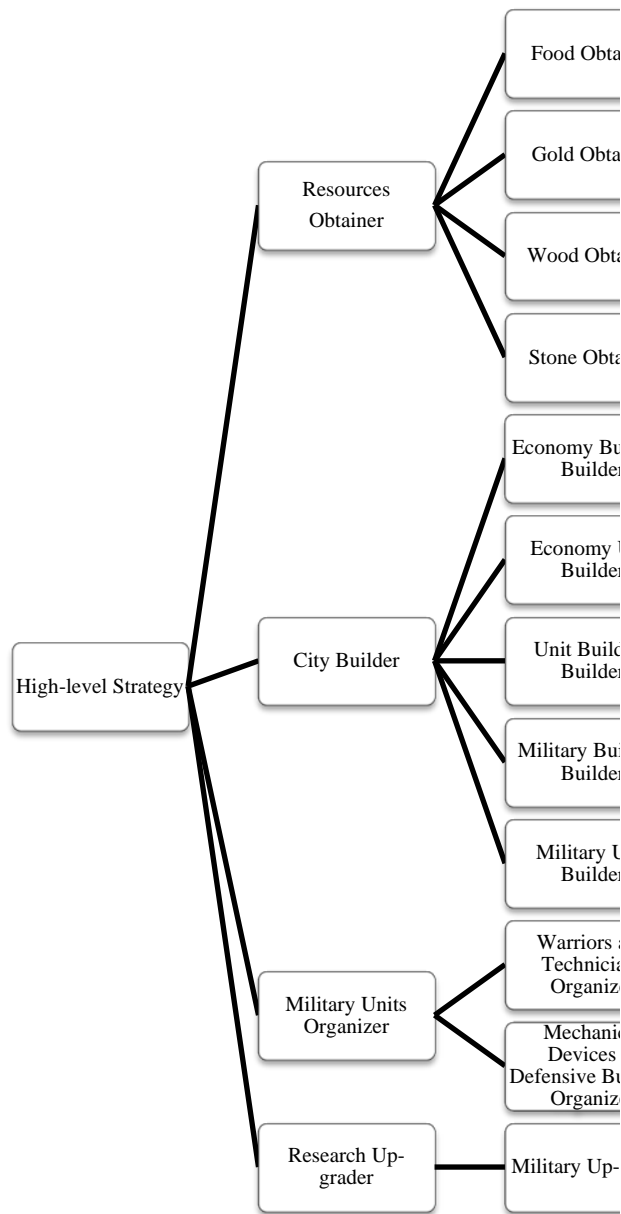


Figure 4: Proposed MAS architecture

3.1. Proposed High level strategy :-

In this level, the high level strategy is determined, by selecting suitable strategy. There are many types of high level strategies like: Rush strategy and Turtle Strategy. Rush strategy focuses on creating cheap,

fast units and attempt to attack the foe [17]. Turtle strategy focuses on a strong defense, keeping all your units in your base, surrounding yourself with defensive structures [17]....etc.

According to selected high level strategy, medium level agents will be categorized. This categorization will be according to their role in this strategy. Some will be main agents. Main agent's function is to generate self strategy and execute required tasks needed from other agents. In rush strategy, there will be three main actions get resources, build city and attack enemy. Main agents will be resource obtainer, city builder and military unit organizer. In turtle strategy, there will be four main actions get resources, build city, upgrade research and attack enemy. Main agents will be resource obtainer, city builder, military unit organizer and research up-grader.

3.2. Proposed Medium-level Strategy:-

Proposed Medium level group units and control them. In this level, strategy is determined according to high level. Each main agent will generate each own strategy and passes it to next level as shown in figure 5.

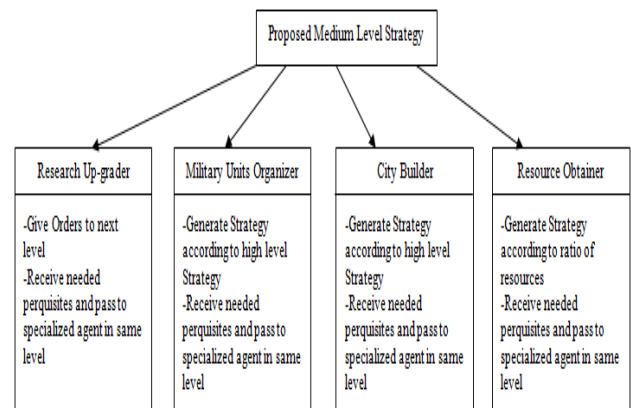


Figure 5: Proposed Multi-agent Function in medium level

Medium level consists of four agents: city builder, research up-grader, resources obtainer and military unit organizer. City builder generate strategy of different types (unit, building) and quantities to the next level according to strategy type. Research up-grader assigns needed upgrade to next level. Resources Obtainer generates strategy of different types and quantities to the next level according to strategy type. Military Unit organizer generates strategy of different types and quantities according to strategy type as shown in table 1.

Table 1 Agents descriptions in Medium Level

Agent	Attributes	Goal
City Builder	Unbounded buffer	Generate Strategy of different types (unit , building) and quantities to the next level
Research Up-grader	Unbounded buffer	Assign needed upgrade to next level
Resources Obtainer	Unbounded buffer	Generate Strategy of different types and quantities to the next

		level
Military Unit Organizer	Unbounded buffer	Generate Strategy of different types and quantities

In rush strategy, there will be three main agents: resource obtainer, city builder and military unit organizer. Resource obtainer: Group of orders is generated within all type of resources and production of them. City Builder: Group of orders is generated. Generate strategy from high level focusing on building cheap and short time units and buildings. Military Unit Organizer: Generate strategy from high level focusing on cheap and short time military units only. In turtle strategy, there will be four main agents: resource obtainer, city builder, military unit organizer and research up-grader. Resource Obtainer: Group of orders is generated within all type of resources and production of them. City Builder: Group of orders is generated. Generate strategy from high level focusing on building expensive and long time units and buildings. Military Unit Organizer: Generate strategy from high level focusing on expensive and long time military units. Research Up-grader focus in performing upgrades to enhance production and militaries capabilities.

I. Proposed Value based Ranking Planning in Medium level

Medium level planning selects according to high level decision. Resource, units and buildings are categorized. Some are main

types. Main types are required in every game. Other types are either cheap types or expensive ones. Medium level planning selects cheap units in rush strategy. Medium level planning selects expensive units in turtle strategy. This choice based on their ranking value. Example: barracks ranking is main type, swordman is cheap type and aerodrome is expensive type.

3.3. Proposed Low level Strategy:-

Low level acts as executer leader to medium level strategy. In this level, it acts as an executer leader of order passed from medium level. It manages flow of executing orders, waiting orders until prerequisites are done. Low level consists of groups of agents. Each agent has its own manager from medium level agents.

Low level has twelve agents. There are Economy Unit Builder, Military Unit Builder, Military Building Builder, Economy Building Builder, Unit Building Builder, Military Up-grader, Food Obtainer, Gold Obtainer, Wood Obtainer, Stone Obtainer, Warriors & Technicians Organizer and Mechanical Devices & Defensive Building Organizer. Economy Unit Builder builds Economy units according to type and quantity. Military Unit Builder builds military units according to type and quantity. Military Unit Builder builds military units according to type and quantity. Military Building Builder builds military buildings according to type and quantity. Economy Building Builder builds economy buildings according to type and quantity. Unit Building Builder builds unit

buildings according to type and quantity. Military Up-grader upgrades military according to its type. Food obtainer gets food according to its quantity. Gold Obtainer gets gold according to its quantity. Wood obtainer gets wood according to its quantity. Stone obtainer gets stone according to its quantity. Warriors and Technicians Organizer assigns warriors and technicians tasks. Mechanical Devices & Defensive Building Organizer assigns mechanical devices & defensive buildings tasks as shown in table 2.

Table 2 Proposed Agents descriptions in low level.

Agent	Attributes	Goal
Economy Unit Builder	Unbounded buffer	Build Economy units according to type and quantity
Military Unit Builder	Unbounded buffer	Build Military units according to type and quantity
Military Building Builder	Unbounded buffer	Build Military buildings according to type and quantity
Economy Building Builder	Unbounded buffer	Build Economy buildings according to type and quantity
Unit Building Builder	Unbounded buffer	Build Unit buildings according to type and quantity
Military Up-grader	Unbounded buffer	Upgrade Military according to its type
Food Obtainer	Unbounded buffer	Get Food according to its quantity
Gold Obtainer	Unbounded buffer	Get Gold according to its quantity
Wood Obtainer	Unbounded buffer	Get Wood according to its quantity
Stone Obtainer	Unbounded buffer	Get Stone according to its quantity
Warriors & Technicians Organizer	Unbounded buffer	Assign warriors and technicians tasks
Mechanical Devices & Defensive Building Organizer	Unbounded buffer	Assign Mechanical Devices & Defensive buildings tasks

4. A Case Study

4.1. Game scenario without MAS planning

We are going to demonstrate multi-agent architecture in built in scenario called “duel”. Duel scenario composes of two opponents: tech and magic.

- a) **Tech initial resources** are fifty food amounts, five hundred amounts for each gold, wood and stone. Tech initial units are three workers, two swordsmen, one archer and one cow. Tech initial buildings are castle and defense tower as shown in table 3.

Table 3 Initial Tech resources, units and buildings in duel

Resources	Units	Buildings
50 food unit	3 workers	1 defense tower
500 gold units	1 cow	
500 stone units	2 swordsmen	1 castle
500 wood units	1 archer	

- b) Magic performs an attack with group of daemons, initiate and battlemage.
- c) Swordmen and archer kill part of them.
- d) Swordmen and archer are died. Castle is damaged. Food, gold, wood and iron in stock are damaged.
- e) Defense tower kills the remaining.
- f) Magic performs another attack with group daemon and battlemage is done. This attack kills the workers and cow. The defense tower is damaged.
- g) The magic wins.

4.2. Game scenario with Applying the proposed MAS Architecture

- a) Tech initial resources are fifty food amounts, five hundred amounts for each gold, wood and iron. Tech initial units are three workers, two swordsmen,

one archer and one cow. Tech initial buildings are castle and defense tower.

We apply our architecture in tech team. High level strategy selects turtle strategy. In turtle strategy, every agent in medium level selects both main and expensive types. Types are units, buildings and resources. The following parameters are applied to our planning technique as shown in table 4.

Table 4 Example Planning Parameters

Military Amount	Building Amount	Resource Amount
6	7	1000

- b) Resource obtainer selects 1000 unit of each (gold-wood-food-stone) item as order to next level.
- c) Military organizer selects 6 of each (guard-horseman-technician-ornithopter-air ballista-airship-battle machine-catapult-defense tower) item that matches the criteria as order to next level.
- d) City builderselects 7 of each (worker-farm-barracks-castle-cow-blacksmith-technodrome-aerodrome) item that match the criteria as order to next level.
- e) City builder sends requests to research up-grader to upgrade (barracks-farms-technodrome-blacksmith).
- f) Research up-grader performs upgrades (training field-stables-advancedarchitecture-robotics-blade weapons-piercing weapons-shield level 1-shield level 2).
- g) In low level, the execution is done or asks for perquisites from their manager from medium level. Low level flow example: when military organizer sends to defense tower attack to mechanicaldevices &defensive building organizer.
- h) It replies to military organizer with to build a defense tower. Military organizer sends a request to city builder.
- i) City builder passes the request to military unit builder that executes defense tower building. Military unit builder send confirm to city builder.
- j) City builder passes confirmation to military organizer. Military organizer passes the confirmation to mechanical devices &defensive building organizer. Mechanical devices & defensive building organizer executes defense tower attack.
- k) After applying the planning architecture, the initial resources, buildings, units and upgrades became as shown in table 5. Minimization in some resources due to the production.

Table 5 Initial tech resources, city buildings and military in duel after planning

Resources	City buildings	Military	Upgrades
150 food unit 450 gold units 1500 stone units 350 wood units	10 workers 8 cow 7 farm 7 barracks 8 castle 7 blacksmith 7 technodrome 7 aerodrome	6 guard 6 horseman 6 technician 6 ornithopter 6 air ballista 6 airship 6 battle machine 6 catapult 7 defense tower 2 swordmen 1 archer	7 training field 7 stables 7 advanced architecture 7 robotics 7 blade weapons 7 piercing weapons 7 shield level 1 7 shield level 2

- l) Magic performs an attack with group of daemons, initiate and battlemage.
- m) Magic destroys some of barracks, blacksmith, technodrome, farms, aerodrome and castle.
- n) Defense tower and catapult destroys attacking group of magic.
- o) Magic attacks with group daemon and battlemage. This attack destroys some of castle, aerodrome and farm.
- p) Defense tower and catapult destroys the attacking group.
- q) Magic attacks with group of drake rider. This attack damages some defense tower.
- r) Defense tower and catapult destroys this group.
- s) Magic attacks with group daemon, battlemage and drake rider. This attack destroys some of defense tower and aerodrome.
- t) Defense tower and airship destroys attacking group.
- u) Magic attacks with daemon, battlemage and summoner.
- v) The airship destroys the attacking group.

- w) Magic attacks with group daemon, battlemage and drake rider.
- x) Airship and ornithopter destroys the attack.
- y) Airship performs an attack to magic energy sources, mage tower and summoner guild.
- z) The game announce tech as winner.

This architecture leads to tech winning which has a positive effect on a highly competitive game.

Given that playing an RTS game is a very challenging task, researchers tend to divide such task into smaller tasks, which can be individually addressed by AI techniques [18]. In the light of our work, this architecture provides a full game play which gives a great advantage. It provides a good management to RTS games planning. The concept of cooperation between agents is illustrated which provides parallelism to orders. Agent specialization to different functions provides a good distribution. This architecture is step towards more intelligence to full game play. It supports more competence to human players.

5. Conclusion

Multi-agent RTS game architecture is discussed and GLEST game example is used. AI strategies have been demonstrated. High level strategy takes the decisions of attacking enemy fast, building strong defense...etc. Medium level strategy generates strategy according to highest level decision. This level determines types, quantities in building city, attacking enemy and getting resources. Low level tasks have been illustrated. In this level, Agents

monitors and execute the strategy according to medium level decision.

This work demonstrates MAS architecture in RTS game architecture has been demonstrated. Different levels are discussed economy and military operations. Variations of action are considered according to capabilities of GLEST RTS games. Some of actions mentioned above are found in some RTS games. Other actions are common along RTS games. This architecture provides a good background to researchers to understand hierarchy of different Strategy levels in RTS games.

Reference

1. Raúl Lara-Cabrera, Mariela Nogueira-Collazo, Carlos Cotta and Antonio J.Fernández-Leiva, "Game Artificial Intelligence : Challenges for the Scientific Community", Proceedings 2st Congreso de la Sociedad Española para las Ciencias del Videojuego Barcelona, Spain, June 2015.
2. Lara-Cabrera, R., Cotta, C., Fernández-Leiva, A.J , "A Review of Computational Intelligence in RTS Games.", In: Ojeda, M., Cotta, C., Franco, L. (eds.) 2013 IEEE Symposium on Foundations of Computational Intelligence. pp. 114_121 (2013).
3. <http://gamejolt.com/profile/made-of-tree/blog/news/evolution-of-rtsgames/3113/> access on 20/03/2015.
4. http://en.wikipedia.org/wiki/Real-time_strategy access on 1/05/2015.
5. Vlasios K. Dimitriadis , "Reinforcement Learning in Real Time Strategy Games: Case Study on the Free Software Game Glest " Diploma Thesis, Department of Electronic and Computer Engineering, Technical University of Crete, December 2009.
6. Michael Buro, Timothy M. Furtak, "RTS Games and Real-Time AI Research", Proc. Behavior Representation in Modeling and Simulation, Citeseer , Virginia ,2004.
7. Ronald Metoyera, Simone Stumpfa, Christoph Neumannb, Jonathan Dodgea, Jill Caoa, Aaron Schnabelc, "Explaining How to Play Real-Time Strategy Games", 29th International Conference on Artificial Intelligence (SGAI), Cambridge, England, December 2009.
8. [M Stanescu](#), [NA Barriga](#), M Buro, "Introducing Hierarchical Adversarial Search, a Scalable Search Procedure for Real-Time Strategy Games", European Conference on Artificial intelligence, Prague, Czech Republic; August 2014.
9. Brian Schwab, "AI Game Engine Programming", Charles River Media; 2nd edition, September 2009.
10. Anders Walther, " AI for real-time strategy games", Master Thesis IT-University of Copenhagen, 2006.
11. http://en.wikipedia.org/wiki/Multi-agent_system access on 20/04/2015.
12. Eric Kok , "Adaptive reinforcement learning agents in RTS games", Master Thesis Intelligent Systems group University Utrecht, The Netherlands,2008 .

13. Josh MCCoy and Michael Mateas,”
An Integrated Agent for Playing
Real-Time Strategy Games”,
Proceedings of the Twenty-Third
Association for the Advancement of
Artificial Intelligence (AAAI)
Conference on Artificial Intelligence
,Chicago 2008.
14. <http://Glest.wikia.com/wiki/Glest>
[access on 15/02/2015.](#)
15. <http://Glest.org/en/index.php> [access](#)
[on 24/1/2015.](#)
16. Marcel van der Heijden, Sander
Bakkes, and Pieter Spronck
,”Dynamic Formations in Real-Time
Strategy Games”, P. Computational
Intelligence and Games (CIG),
December 2008.
17. Mathijs M. de Weerdt and Brad J.
Clement , “Introduction to Planning
in Multiagent Systems “, *Multiagent*
and Grid Systems An International
Journal 5(4):345-355. [BibTeX],
2009.
18. Santiago Ontañón, Gabriel
Synnaeve, Alberto Uriarte, Florian
Richoux, David Churchill, Mike
Preuss,”A Survey of Real-Time
Strategy Game AI Research and
Competition in StarCraft”, IEEE, P
293 – 311, October 2013.