Incorporating

RGT Solver Package with TAC Solver Agent

Alireza Moradkhani
Institute for Informatics and Automation Problems
Yerevan, Armenia
Alireza.moradkhani@gmail.com

ABSTRACT

TAC SCM (Trading Agent Competition for Supply Chain Management) is an abstraction of the supply chain management problem in dynamic markets. The RGT Solver package is a RGT (framework of games where search space is specified by Reproducible Game Tree)based cognitive architecture that uses elementary structures to model the RGT domain problems and produce behaviour from the given expert knowledge. It is capable of modelling all types of RGT domains into a single platform. The goal is to employ the RGT Solver Package as a cognitive architecture and decision maker for the TAC Solver agent. To achieve this goal we have done adaptations in both sides in RGT Solver Package and in TAC SCM agent. The agent sends situations as messages into the RGT Solver Package through its percept manager. These situations are analysed and the best corresponding action is suggested and sent back to the TAC Solver agent. To accomplish this goal the RGT Solver Package models the given situation by building the corresponding game tree. A profit maximizing evaluation functions is used for selecting optimal moves path in the game tree. In this paper we describe the middleware module which integrates the TAC Solver Agent with the RGT Solver package. We summarize our work with conclusions.

Keywords

Cognitive systems, Trading agent Competition, Multi agent systems (MAS), Supply chain management.

1. Introduction

The Trading Agent Competition for Supply Chain Management (TAC SCM) [1] is designed to create a platform for teams to compete in a supply chain scenario. It allows the different teams to create their agent and compete with other teams. The agents concurrently compete in multiple markets with interdependencies and incomplete information.

The RGT Solver package is a RGT (framework of games where search space is specified by Reproducible Game Tree) based cognitive architecture which models the RGT domain problems and produces behavior from the given expert knowledge.

1.1. The Solver Package

The Solver package is a cognitive system which is designed to acquire strategic expert knowledge to become comparable with a human in solving hard combinatorial competing and combating problems [11-14]. Expert knowledge acquisition tasks are:

1. construction of the package of programs sufficient to acquire the meanings of the units of vocabulary (UV) of problems,
2. construction of procedures for regular acquisition of the meanings of UV by the package,
3. provision of means measuring the effectiveness of solutions of RGT problems.

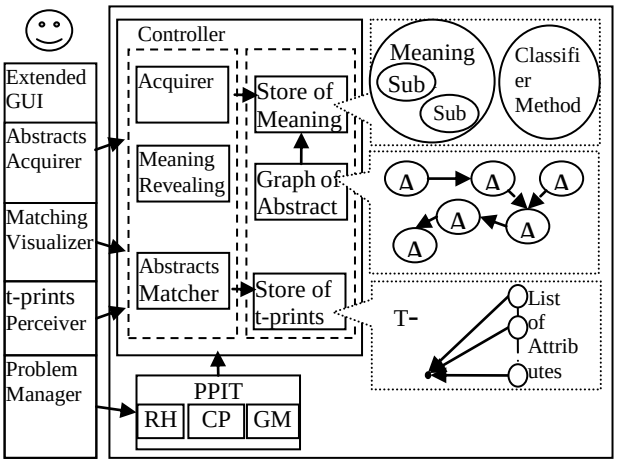

Fig 1; The solver-package

1.2. TAC SCM Game

A TAC SCM game [1,3] consists of a number of days where six personal computer assembly agents compete for customer orders and for procurement of a variety of components. Each day, customers send requests for quotes and select from quotes submitted by the agents, based on delivery dates and prices. The agents are limited by the capacity of their assembly lines and have to procure components from a set of eight suppliers. Four types of components are required to build a PC: CPUs, Motherboards, Memory, and Disk drives. Each component type is available in multiple versions. Customer demand comes in the form of requests for quotes for different types of PCs.

A game begins when an agent connects to a game server [3]. The server simulates the suppliers and customers, and provides banking, production and warehousing services to the participant agents. The game lasts for 220 TAC days. The agent with the highest amount of money is the winner of the game.

Four basic decisions that an agent must make during each simulated day [1]:

1. Procurement: To decide what parts to purchase, from whom and when to have them delivered.
2. Production: To schedule its manufacturing facility.
3. Sales: To decide which customer RFQs to bid to what prices.
4. Fulfillment: To ship completed orders to customers.

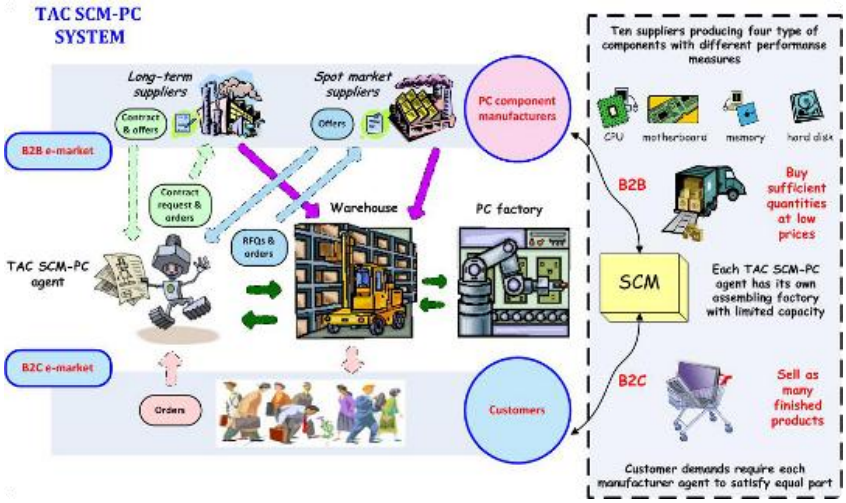

Figure 2; The TAC SCM game overview

To succeed, agents will have to demonstrate their ability to react to variations in customer demand and availability of supplies, as well as adapt to the strategies adopted by other competing agents. Competing agents must be able to sense and model the environment and predict their own impact on the environment.

1.3 The RGT Methodology

An arbitrary problem for the RGT class can be represented as compositions of the following components [6]:

- List of situations
- List of participants
- List of valid actions that can be executed
- List of triggering rules
- List of the goals of the participants

We model the markets by applying game tree based approach [4]. We apply common and expert knowledge to narrow our search space [4]. We also suggest a regular transformation of cases into the rules [6].

Our objectives are using RGT game tree based methodology to model the markets and increase the effectiveness of decision making processes in TAC Solver agent. We study these through framework of games where search space is specified by reproducible game trees (RGT) and target solutions have to be discovered by methods able to systematic acquired knowledge about them [4]. Solutions are known as strategies that are a sequence of actions performed by competing parts. Management Optimal Strategy Provision (MOSP) problem includes in particular, sub models of the market, alternative strategies and competitions, evaluation and selection of strategies [6].

In the variety of problems we identify the class where space of possible solutions can be specified by Reproducible combinatorial Game Trees (RGT) and develop unified software, RGT Solver, for elaborating optimal strategies for any input specified problem of the class [4]. RGT is a class of problems with the following requirements as demonstrated in [11]:

- There are (a) interacting actors (players, competitors, etc. performing (b) identified types of actions at the (c) specified moments of time and (d) specified types of situations
- There are identified benefits for each of the actors
- The situations the actors act in and transformed after the actions can be specified by certain rules, regularities.

We solve games of RGT class with meanings we do specify by states, situations, actors, actions of players, evaluators of situations and regularities of transformation of situations [6]. Many security and competition problems belong to RGT class since those problems are always interactions and RGT requirements include the most common of them. Specifically, these are network Intrusion Protection (IP), Management in oligopoly competitions and Chess-like combinatorial problems, various security problems. Unified RGT specification of problems makes possible to design a unified Solver for the problems of the class [4, 5]. RGT Solver package aimed to acquire strategic expert knowledge to become comparable with a human in solving hard combinatorial competing and combating problems.

2. The Adaptation

To integrate the TAC Solver agent with the RGT Solver package we do adaptation in different levels. In the graphical user interface we have defined a separate tab called TAC SCM which includes all TAC SCM specific domain functionalities.

2.1 Adaptation of Concepts

In order to define the TAC SCM game into the RGT Solver package we have extracted meanings from the TAC SCM game and categorized under different RGT solver package terminology. These terminologies are nucleus, primitive and composite abstracts, R1s, actions, and t-prints which are represented in tables 1-5. They are defined and inserted into the RGT Solver Package using the meanings definition screen in GUI in the RGT Solver Package [13].

Nucleus Concepts			
RFQID	ProductID	Penalty	ReservePrice
Factory.address	BankIntrestRate	Customer.daysBeforeVoid	Revenue
OfferID	SimulationID	DueDate	UnitPrice
Factory.storageCost	BankDepositInterestRate	NumberOfSecondsPerDay	Profit
OrderID	FactoryID	Quantity	Factory.capacity
Customer.daysBeforeVoid	Manufacturer.<count>	NumberOfDays	PotentialProfit

Table 1; TAC SCM related nucleus concepts

Primitive Concepts		
Low-end-product Parent ProductID	Mid-end-product Parent ProductID	High-end-product Parent ProductID
Low-Penalty Parent Penalty	Mid-Penalty Parent Penalty	High-Penalty Parent Penalty
Low-ReservePrice Parent ReservePrice	Mid-ReservePrice Parent ReservePrice	High-ReservePrice Parent ReservePrice
Low-UnitPrice Parent UnitPrice	Mid- UnitPrice Parent UnitPrice	High- UnitPrice Parent UnitPrice
Low-FCapacity Parent Factory.capacity	Mid-FCapacity Parent Factory.capacity	High-FCapacity Parent Factory.capacity
Game-Start Parent NumberOfDays	Mid-Game Parent NumberOfDays	Game-End Parent NumberOfDays

Table 2; TAC SCM related primitive concepts

Composite Concepts				
GameInfo	CustomerOffer	SupplierRFQ	SupplierOrder	DeliveryReport
CustomerRFQ	CustomerOrder	SupplierOffer	SupplierDeliveryNotice	FactoryReport

Table 3; TAC SCM related composite concepts

Set Concepts		
CustomerRFQBundle	CustomerOfferBundle	CanceledCustomerOrders
SupplierRFQBundle	SupplierOrderBundle	ComponentsSetforToday
ProfitableCustomerRFQs	CustomerOrderBundle	SentCustomerOffers
SupplierOfferBundle	SupplierDNBundle	DeliveriesToday

Table 4; TAC SCM related set concepts

T-print Concepts			
Game-start	Customer-offer-generated-sent	Supplier-RFQ-generated-sent	Supplier-order-generated-sent
Customer-RFQ-received	Customer-order-received	Supplier-offer-received	Delivery-notice-received

Table 5; TAC SCM related T-print concepts

2.2 Adaptation of the GUI

In order to enter expert knowledge into RGT Solver package we need to extend the RGT Solver package's GUI with the additional screens. These new screens are organized under a tab called "TAC SCM game" and described in the following subsections.

2.2.1 Procedural Expert Knowledge Entry Screen

We have designed an entry screen where the expert user inserts the expert knowledge pieces in form of rules into the RGT Solver package. This expert knowledge is stored in the database.

2.2.2 Game Progression Screen

This screen shows updated information about each competitor agent in an ongoing TAC SCM game. This information are sent from the TAC SCM game server via perception manager into the RGT Solver.

3. RGT Perception Manager

We extend the RGT Solver package with an additional module called perception manager to integrate the TAC Solver agent with the RGT Solver package. The perception between the RGT Solver Package and the TAC SCM game is performed by the RGT Perception manager module. This module is responsible for translating the perceptions using specific predefined maps.

The management of perception is performed by the perception manager module. It has the responsibility of transforming We define a temporary class (TAC-SOLVER class) to function as an API for interface between TAC SCM and the solver-package. It has the following capabilities:

1. Translate TAC SCM messages into the solver language.
2. Describe actions in the solver language. From the core language make all specific API language (java).
3. Allow function calls outside the solver through a messaging protocol that solver post conditions are able to make external calls.
4. Generate messages to be sent to TAC SCM.

There are two event handlers in the percept manager each handling the events of their own side.

- **RGT TAC Interface**: It receives messages from the game program and maps them into the percept manager's internal representation. It functions as a bridge between the percept manager and the SSRGT game agent. It updates the output whenever the internal representation is changed. It contains a set of predefined maps for each RGT game.
- **TAC Agent Event Handler:** The event handler is triggered whenever perception manager receives a new message from the TAC SCM solver. It contains functions that can call other functions. List of the functions in TAC Solver Agent:
  AssignSolverActions()
  GenerateCustomerOffer()
  GenerateSupplierRFQ()
  GenerateSupplierOrder()
  SendActions()

- **RGT Solver Event Handler**: The event handler is triggered whenever an action is received from the RGT Solver at the end of the Solver's decision cycle. It gets the actions and buffers them in the action queue. List of the functions in RGT-Solver: ProcessNewSolverOutput()
  SelectCustomerOffer()
  SelectSupplierRFQ()
  SelectSupplierOrder()
  GetNewActions()
- **RGT Solver Interface:** It functions as a bridge between the RGT Solver and the percept manager. It receives actions from the RGT Solver and interprets them into the percepts manager's internal representation using the RGT Solver map which includes the translation of RGT Solver Language into actions and vice versa.

We define a temporary class (TAC-SOLVER class) to function as a API for interface between TAC Solver agent and the solver-package. It has the following capabilities:

1. Translate TAC SCM messages into the solver language
2. Describe actions in the solver language. From the core language make all specific API language (java).
3. Allow function calls outside the solver through a messaging protocol that solver post conditions are able to make external calls.
4. Generate messages to be sent to TAC SCM.

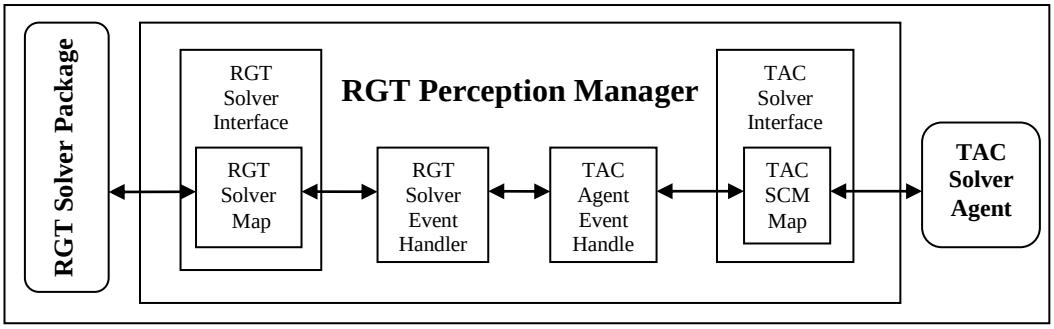

Figure 3; Perception management architecture in RGT Solver Package

4. Conclusions

We have designed and implemented the means to integrate the TAC Solver agent with the RGT Solver package. The suggested TAC specific concepts are inserted into the RGT Solver package through its graphical user interface (GUI) implemented before. The middleware required is a module called a perception manager containing respective maps and event handlers. This adaption makes TAC agent to send the situations into RGT Solver and receive the corresponding action to be performed. This allows the TAC Solver agent to utilize the ability of the RGT Solver Package fully to solve the daily decision questions.

5. ACKNOWLEDGEMENT

I would like to thank Karen Khachatryan and Sedrak Grigorian for their constructive comments.

REFERENCES

[1] A. Moradkhani, "On Trends in Supply Chain Management Trading Agent Competition", Proceedings of the CSIT2011, 8th. International Conference in Computer Science and Information Technologies, 190p, Yerevan (2011).

[2] A. Moradkhani, T. Baghdasaryan, "Towards a Knowledge-based Approach for Autonomous Trading Agent", Developments in Business Simulation and Experiential Learning, Vol. 39, 295-302., 39th. International Conference in Computer Science and Information Technologies, 190p, San Diego, California, USA (2012).

[3] Collins, J.; Arunachalam, R.; Sadeh, N.; Ericsson, J.; Finne, N.; and Janson, S., "The supply chain management game for the 2007 trading agent competition". Technical Report CMU-ISRI-07-100, Carnegie Mellon University, Pittsburgh, PA 15213, 2007.

[4] E. Pogossian, "Focusing Management Strategy Provision Simulation", CSIT2001 3d International Conference in Computer Science and Information Technologies, 5p., 2001.

[5] E. Pogossian: Specifying Personalized Expertise. International Association for Development of the Information Society (IADIS): International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006), 8-10 Dec., Barcelona, Spain (2006) 151-159.

[6] E. Pogossian, "Effectiveness Enhancing Knowledge Based Strategies for SSRGT Class of Defense Problems", Proceedings of NATO Advanced Study Institute (ASI), 2011.

[7] K. Khachatryan, V. Vagradyan, "Graphical Language Interpreter Unified for SSRGT Problems and Relevant Complex Knowledge", CSIT2011 8th International Conference in Computer Science and Information Technologies, pp. 178-182, 2011.

[8] K. Khachatyan, S. Grigoryan "Java Programs For Presentation And Acquisition Of Meanings In SSRGT Games", Proceedings of SEUA Annual conference, 7p., 2012.

[9] K. Khachatryan, S. Grigoryan, "Java Programs for Matching Situations to the Meanings of SSRGT Games", Proceedings of SEUA Annual conference, 5p., 2012.

[10] TAC SCM web page: http://tac.sics.se/page.php?id=1.

[11] E. Pogossian, V. Vahradyan, A. Grigoryan "On Competing Agents Consistent with Expert Knowledge". Workshop on Autonomous Intelligent Systems - Agents and Data Mining.. Petersburg, 2007.

[12] Z. Naghashyan, E. Pogossian, "Developing Java Software for Representation, Acquisition and Management of Strategy Knowledge" Mathematical Problems of Computer Sciences, Proceedings of IPIA, p. 10pp. Yerevan, 2010.

[13] K. Khachatryan, S. Grigoryan, "Java Programs for Matching Situations to the Meanings of SSRGT Games" . Proceedings of SEUA Annual conference. pp. 135-141, Yerevan, 2013.

[14] E. Pogossian, "Modeling of Meaning Processing and its Applications in Competition and Combating Games". Proceedings of SEUA Annual conference. pp. 141-152. Yerevan, Armenia, 2013.