World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 30-35, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.csea.2212

A Frame Work for Modeling Task Coordination in Multi-Agent Systems

R. Udayakumar, K.P. Thooyamani Khanaa and A.V. Allin Geo

School of Computing Science, Bharath University, Chennai - 73, India

Abstract: An agent-based system of activities in a product lifecycle has given a way to the utilization of concurrent engineering principles. Mainly agent works coordinately with the other agents in heterogeneous communication environments for solve intricate problems. In this paper we consider the resource constrained of the local behavior of individual agents to provide an appropriate system-level behavior. The objective is to minimize duration when constrained by precedence relationships among activities control task model processes effectively. Improved communication and flow of task information between activities are necessary for process management. The model determines the shortest duration by allocating available resources to a set of activities simultaneously. This paper introduces a methodology for designing the necessity of Task Control Agent along with the issues in Multi Agent Systems to establish a set of feasible start times for all activities such that the entire project is completed in a minimum makespan.

Key words: Concurrent Engineering · Multi Agent · Task controlling · Agent-based Systems · Controlling and Communication

INTRODUCTION

Concurrent Engineering (CE) has been defined as a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. [Concurrent engineering] is intended to cause the developers, from the outset, to consider all elements of the product life cycle from conception through disposal, including quality, cost, schedule and user requirements [1]. In the Agent-based information systems have gained attentions due to the proliferation and readily available information. Since the information required for solving a problem is continuously generated, processed and maintained within the distributed nodes, it should be recognized and integrated by a coordinator. In order for the coordination to be proper, the coordinator should be able to recognize the exceptions that are raised among different tasks and resolve them properly.

The techniques to share data and process information among teams are the area of active research. Developing techniques for effective collaboration is the key to cross-functional teams. Sharing information about local constraints, priorities and preferences early in the design process is a powerful conflict-management technique. Conflict avoidance is one of the founding principles of concurrent engineering. Since many processes are performed in parallel, incomplete or uncertain information can be used at an early stage. Sharing information or imperfect information may result in increasing the number of life-cycle areas and is described as concurrent chaos.

The coordination framework should be responsible for communicating the information and for ensuring progress towards the goal. The ability to capture and manage exceptions is essential for enhancing the effectiveness and efficiency of the coordination mechanism [2]. We developed a tool that aims to construct concurrent engineering facilities to aid the cooperation/collaboration of people designing and producing products and services of any kind of industry.

Multi-Agent Systems [MAS]: Agent technology is becoming increasingly important with every passing day since it promises to change software construction, as stated in [3]. Multi Agent system (MAS) is concerned with understanding a broad range of issues related to the task control and actions in environments involving

Corresponding Author: R. Udayakumar, School of Computing Science, Bharath University, Chennai – 73, India.

multiple tasks. MAS deal with the interactions between agents and working together to complete a given work [4]. As agent systems grow in complexity the density and diversity of the interconnections between agents increase rapidly [5].

A process is carried out to solve a problem in other words to attain a specified goal. The problem is usually decomposed into smaller sub-problems [6]. Once the individual solutions of the lower most sub-problems are obtained, the sub-solutions at the lower levels can be aggregated to reach the solution of their parent problem. This process involves cooperative sharing of information among a group of agents. The global behavior of these complex systems can be considered to be an emergent property of the interactions between the different agents. An alternative way to manage this form of agent-based system is to utilize emergent properties to make self-organizing and self-regulating Multi-Agent systems [7]. Agents have to decide not only which local action to perform, but also whether it is worthwhile to perform a communication action before deciding the local action [8]. It is suggested that this would provide a foundation for formal study of cooperative activities and may lead to some insights to the design of cooperative agent policies.

Multi-Agent Coordination: The primary phase of multi-agent coordination involves gathering the agents in to appropriate groups to carry out the process [9]. Each higher-level task (process) in the system is divided into partial sub-tasks and distributed to the relevant agents. The group generates in turn a set of collectively agreed objectives through an agent interaction model to attain a common goal. The processes are carried out by a group of agents. Each output of an agent is valuable in its own right and also helps to generate the next process. Cooperative enquiry is a general method of organizing a group of agents to work out a collectively agreed understanding of a problem. The approach calls for a group of agents to afford equal results using the following phases [10].

Proposition: The agent proposes and aggress the next action it will take.

Active: The agent takes the agreed action

Reaction: The agent looks at the action and gives its immediate reaction to it.

Reflection: The agent quietly reflects on what has been achieved by the action and considers what else if everything needs to be done to complete the action, whether the result is correct and what action to take next.

Communication Through Cooperative Agents: Coordination is the orderly arrangement of group efforts to provide units of action in pursuit of a common purpose. For example, agents must coordinate their activities during the design process to produce quality designs and to use resources effectively. Each task performed by a task agent (henceforth called coordinator) should accomplish a link with the agents that perform a subtask (henceforth called cooperative agent or cooperator). The primary purpose of cooperative agent is to retrieve relevant information by hiding the geographical distribution of data from the view of the coordinator.

Cooperation plays a vital role in the decision making process. A cooperating agent in multi-agent coordination can be developed through the following steps:

(1) Identify the agents (2)Fix the scope (3)Assess feasibility and (4)Evaluate the risk

The implemented agents are able to retrieve relevant updated information from any remote node connected through the network using these agents. The developed system concentrates only on simple (Data Manipulation) queries and it is cooperative up to some extent [9]. The limitation of this system is the agents reside in the server only for a specific period of time. In case of any errors in cooperative agents, it was not properly intimated to the requesting coordinator. The coordinator unknowingly waits for the cooperative replies until the expiry time of the query, not being aware of the exceptions occurred in the cooperating agent. In order to rectify this problem the authors proposes a model for exception handling services to guide the coordinator.

Team Working: This paragraph covers what teams are, what makes teams work and how to resolve teamwork problems for Task Control Agent with their agent members(Agent A, B,C).

Team Goals: A team [11] consists of at least two people, who b) are working toward a common goal/objective, where, c) each person has been assigned specific roles or functions to perform and where, d) Completion of the mission requires some form of dependency among the group members.

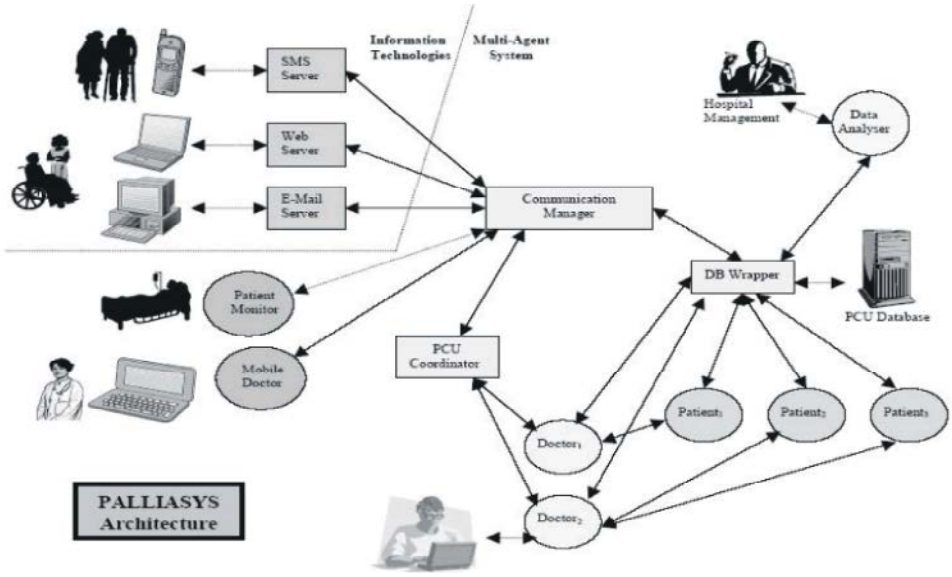

Fig. 3.1: Architecture of the PalliSys system

The criteria for successful teams are that the team is clearly identified, its tasks are clear and distinct, the team members have control on their tasks and there is a real need for the team. The team's success depends on all team members contributing there personal best efforts, supporting other team members and working cooperatively to resolve issues and disagreements.

For example we take *PalliaSys* project, in which both information technology and multi-agent systems are used to improve the care given to palliative patients [12].

One of the basic aims of the *PalliaSys* system is to improve the process of gathering and collecting the information of the palliative patients. This information is stored in a central data base located at the PCU of the medical centre. The first goal of all team members (Agents) is to work cooperatively with the entire team to produce a high-quality product on time.

Effective Teamwork
Effective Teamwork Requires:

- Agreed team goals, Established team-member roles, A supportive environment in which to work, A common team work process, A plan for the work, A mutual team commitment to the goals, roles and plan, Open and free communication among all the team members, The mutual respect and support of all the team members

Task Coordination in Multi-Agent Systems: A *multi-agent system* [13] may be defined as a collection of autonomous agents that communicate between themselves to coordinate their activities in order to be able to solve collectively a problem that could not be tackled by any agent individually. Specifically, the fact that agents are implemented as parallel threads of execution leads to issues related to the synchronization and the ordering of their actions in the environment.

Algorithm: 4.1

/ Initialization of the TCA */*
//Task regions <- recursive-split (task) Create the TCA agent
For all task region: task regions do
Create new agent [level 0; Inherit behavior from default pattern; Initialize with task region)
Initialize the acquaintances of the agent End for
//Send start task agent to TCA

Then the product development has increasingly become a cooperative endeavor that requires effective coordination in the face of complex dependencies over agents, time and functional perspectives. Distinct coordination support communication technologies have emerged for each of those kinds of distribution, but all the face important limitations. They developed a unified model [14] of CE coordination that synergistically combines existing approaches in a way that avoids many of their individual limitations and combines their communication strengths.

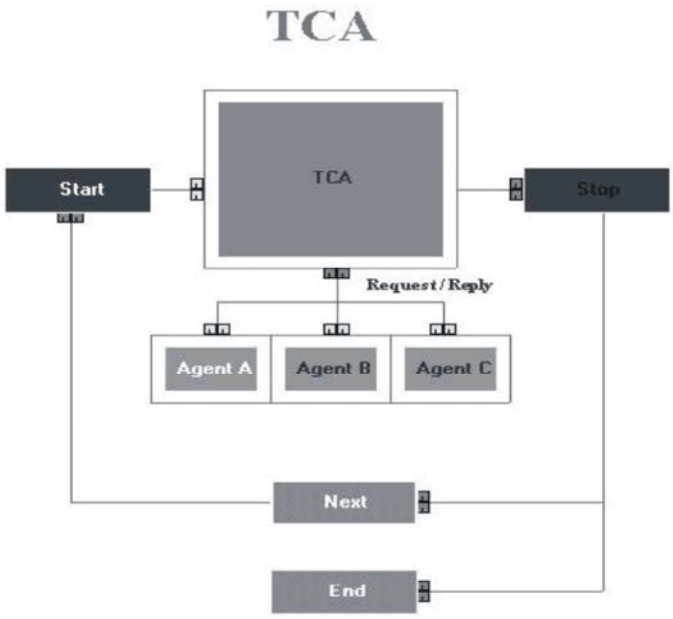

Fig. 4.1: Sample Proposed System for Task Scheduling Activities

RESULTS

MAS are concerned more with the design of the individual agent and how autonomous, self-motivated and even heterogeneous agents can function together. The MAS is accessible live in the *AgentCities* network of platforms (see access instructions at http://grusma.etse.urv.es/~agentcities/). The agent invokes taking a large job that is divided into dependent tasks and finding a way to schedule the tasks among a group of networked agents with the goal of completing all of the tasks in the least amount of time.

TCA (or) CMA is one of the deemed processes Management whose taking into one cycle will be completed then thereafter the process will properly Completed to announces by the informal action via Agent enables the "Process Completed" event. Otherwise the Information Details hiding in the command box by the process oriented stages. TCA will post any kind of information to the agents and wise versa. To model the Task Control Agent, the following figures are distinguished:

In CE, TCA interfaces will be provided base Agent models. Generally Agents can be categorized into three types. Agent-A, Agent-B and Agent-C are working with multitasking environment (TCA). Each of the above three agents are having 5 tasks. Totally three agents establish 15 tasks should be performed, each task separately gets into individual properties. Agent A, Agent B and agent C are working autonomously. Each tasks enables for every 5 seconds at the moment it should completed its work. Agents are created the task base criteria's as per Request/Response for scheduling the each task services. Agent A supported for limitation of access for 5 tasks such maintains constraints control into TCA. Each Task is maintaining 3 kinds of information viz., wait (), Process () and completed ().

Agent A, Agent B and agent C were working under priority basis with controlling of TCA System. As per Agent A, work starting first with first task by the condition approval of TCA system Basis. To start the next task, the First task (Task 1) must be completed. Hereby Each task can be performed for every 5 seconds it's watched by the Agent A, Agent B and Agent C through TCA. If any kind of interruption occurs then the agent will immediately pass the Information to the TCA System. TCA alerts the message into the Particular Agent (Agent A, Agent B and agent C) and stop the particular Work immediately. The TCA information (Data's) can be stored into the Database permanently via Agent A, Agent B and agent C. The Agent A, Agent B and agent C are Cooperate the TCA System according to the situation of the states coherent. TCA is one of the deemed processes Management whose taking into one cycle will be completed then thereafter the processes will properly Completed to announce by the informal action via Agent enables the "Process Completed" event. The result is a hierarchy of processes which are carried by agents. It is easier to establish roles and interactions between roles than processes and interactions between processes [15].

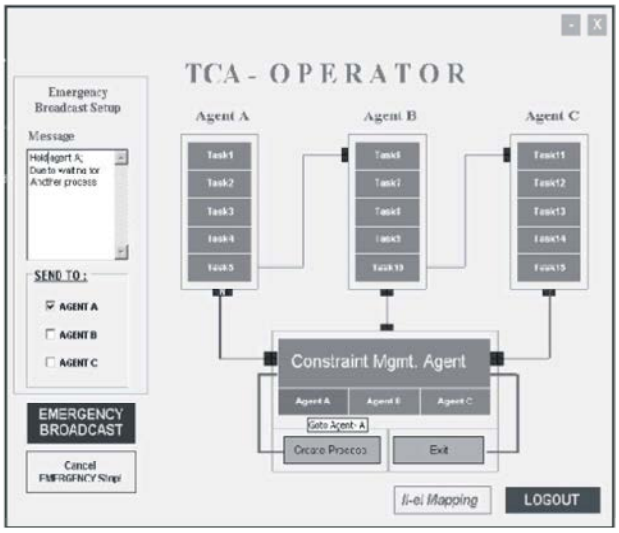

Fig. 4.2: Multi-Agent System Concepts Using Constraint Management Agent (CMA)

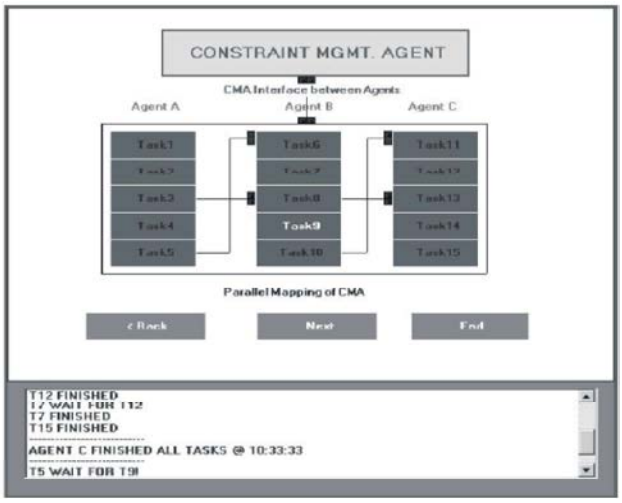

Fig. 4.3: MA Interface Between Agent Task Coordination

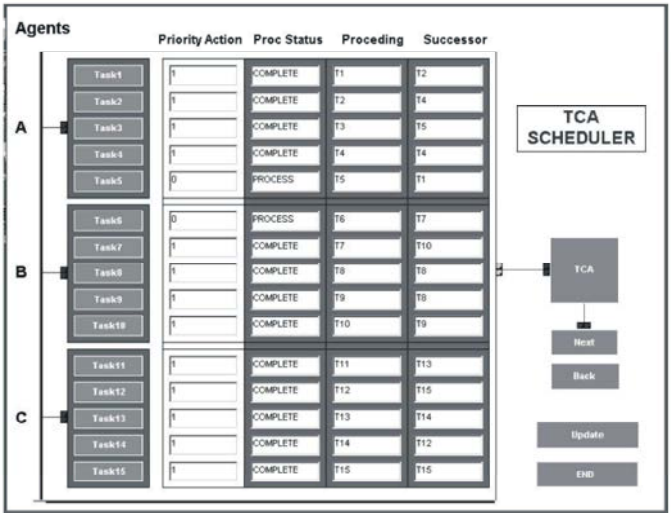

Fig. 4.4: Sample task table for interact to achieve the result of table

CONCLUSION

Agent-based working plays a key role in the design and implementation of distributed systems for cooperation purposes. Since the multifunctional teams have the same objectives, communication and information sharing among multiple teams tend to be more frequent. Hence, open and free communication and information sharing play a vital role in team working for Concurrent Engineering. This approach produces substantially superior performance without complicating agent development, since important type of failures is detected by handlers and not by coordinators. This paper focuses exclusively on some exceptions occurring in the sub-task level at runtime and not on exceptions concerned with handling agents or information of alternative problem solving method.

Future Work: The future development will continue with a main focus to implement task process enhancement in Data Mining(*Data clustering* is the task of partitioning a multivariate data set into groups maximizing intra-group similarity and inter-group dissimilarity). Efforts to improve efficiency by reducing the effects of interference by the sub-agents will also be aimed at execution time estimation model for Data Mining jobs.

REFERENCES

1. Prasad, B., 1997. Concurrent Engineering Fundamentals, Volume II: Integrated Product Development, USA: Prentice Hall.
2. Kleinmann, K., R. Lazarus and R. Tomlinson, 2003. "An Infrastructure for Adaptive Control in Multi-Agent Systems", IEEE Conference on Knowledge-Intensive Multi-Agent Systems (KIMAS), Cambridge.
3. Griss, M.L., 2000. My agent will call your agent … But will it respond. Software Development Magazine, Feb 2000.
4. Wooldridge, M., 2002. An introduction to MultiAgent Systems. Wiley Ed. (2002).
5. Weyns, D. and T. Holvoet: 2004, 'A Formal Model for Situated Multi-Agent Systems.'. Fundam. Inform., 63(2-3): 125-158.
6. Odrey, N.G. and G. Mejía, 2003. "A re-configurable Multiagent system architecture for error recovery in production systems." Robotics and Computer Integrated Manufacturing, 19: 35-43.
7. Alexander, I.F., 1998. Supporting a co-operative requirements engineering process, Conference on European industrial requirements engineering, Londen, 19-20 Oct. 1998.
8. Weyns, D., H.V.D. Parunak, F. Michel, T. Holvoet and J. Ferber, 2004. 'Environments for Multiagent Systems, State-of-the-Art and Research Challenges'. In (Weyns *et al*., 2004).
9. Brinn, M. and M. Greaves, 2003. "Leveraging Agent Properties to Assure Survivability of Distributed Multi-Agent Systems," 2nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Melbourne.
10. Parunak, H.V.D., S. Brueckner, M. Fleischer and J. Odell, 2003. 'A Design Taxonomy of Multi-agent Interactions.'. In: P. Giorgini, J. P. Muller and J. Odell (eds.): Workshop on Agent-Oriented Software Engineering, Vol. 2935 of Lecture Notes in Computer Science, pp: 123-137, Springer.
11. Davidsson, P. and F. Wernstedt, 2002. "A multi-agent system architecture for coordination of just-in-time production and distribution." Proceedings of SAC, Spain, pp: 294-299.
12. Antonio Moreno, 2004. Medical Applications of Multi-Agent Systems Proceedings of the 5th International Conference on Enterprise Information Systems, ICEIS-02. Ciudad Real, Spain, pp: 431-438.
13. MABS: 1998, 2000, 2002, 2003, 2004, 'Multi-Agent Systems and Agent-Based Simulation, International Workshop Series', Vol. 1534, 1979, 2581, 2927, 3415 of LNCS. Springer.
14. Sycara, K. and M. Wooldridge, (eds.): 2006, Autonomous Agents and Multi-Agent Systems: Special Issue on Environment for Multi-Agent Systems. Springer.
15. Saravanan, T. and R. Udayakumar, 2013. Comparision of Different Digital Image watemarking techniques, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1684-1690.