An Agent-based Simulation System for Dynamic Project Scheduling and Online Disruption Resolving

(Extended Abstract)

Hui Xi
Chi Keong Goh
Partha Sarathi Dutta
Rolls-Royce Singapore Pte Ltd, Singapore
{Hui.Xi, ChiKeong.Goh, Partha.Dutta}@Rolls-Royce.com

Meng Sha
Jie Zhang
Nanyang Technological University, Singapore
{shameng, ZhangJ}@ntu.edu.sg

ABSTRACT

This work proposes an agent-based simulation system to address challenges in resource constrained multi project scheduling for a technology intensive industrial product development programme. The system aims at helping users develop robust programme schedules by taking care of intra- and inter- dependencies between projects, potential knock-on effect of unpredictable events, and resource constraints. With the interactive graphic user interface embedded in the system, users are enabled to investigate project process, evaluate modified programme schedule and conduct various what-if analysis.

Categories and Subject Descriptors

H.4 [**Types of Systems**]: Decision support

General Terms

Management

Keywords

Agent-based modeling, Interactive simulation platform, Dynamic project scheduling, Online decision support system

1. INTRODUCTION

Scheduling multiple projects that run simultaneously in a dynamic environment remains a major challenge in project management. On one hand, projects differ in size, required resources and task priorities. One the other hand, disruptions can occur during the project execution, which requires rescheduling that may impose impact on other projects due to interdependencies among projects. While classic mathematical approaches help deal with scheduling problems with low complexity or in a relatively static environment, it has limitation in describing interrelationships in multi-project environments and adapting the analysis to dynamic change [2].

Agent-based approach has been identified as an effective approach for scheduling problems with high complexity [1, 3]. However, many existing studies focus on the offline scheduling algorithms. Very few studies have been conducted on decision supporting for online rescheduling caused by on-the-fly changes from unpredictable disturbances or user intervention. Furthermore, it is usually inefficient for conventional scheduling algorithms to handle frequent online rescheduling because of the high complexity. One promising direction is to combine multi-agent approaches with optimization techniques [3] as incorporating optimization techniques with agents' behavior logic would increase their intelligence which leads to optimal decision making for both short-term and long-term objectives.

Thus, in this paper, we propose an agent-based simulation system to help users monitor project execution process and provide support for online decision making. An interactive graphic user interface (GUI) has also been integrated with the simulator to enable user-in-the-loop experimentation.

2. AGENT-BASED SIMULATION SYSTEM

A real-life industrial product development programme is adopted to illustrate our proposed system. This programme consists of multiple projects being executed concurrently, and its objective is to complete all the projects by the deadline. Each project is assigned to a dedicated developer who will respond to disruptions occurring in the execution. A project consists of several tasks each of which needs requires a particular facility. Each task contains dozens to hundreds of subtasks. During the task execution, all the subtasks are performed using certain instruments in every single day in parallel. The instruments may become faulty during the execution, and lead to subtask failure when the number of faulty instruments reaches the limit. Any subtask failure is considered as a disruption that will discontinue the whole programme, which requires a decision support to resolve.

Figure 1 depicts the structure of our proposed simulation system where input data, embedded agent behaviors and programme logic are accessible to users through an interactive GUI. Both objects and agents are employed in the modeling of the system. Tasks, subtasks and facilities are modeled as objects as they do not perform autonomous behaviors. On the other hand, developers and projects are modeled as developer and project agents. Each developer agent is assigned with one or more projects. Their responsibilities include monitoring project progress, adjusting task setting, and coordinating with other developers to resolve

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May, 4–8, 2015, Istanbul, Turkey.*

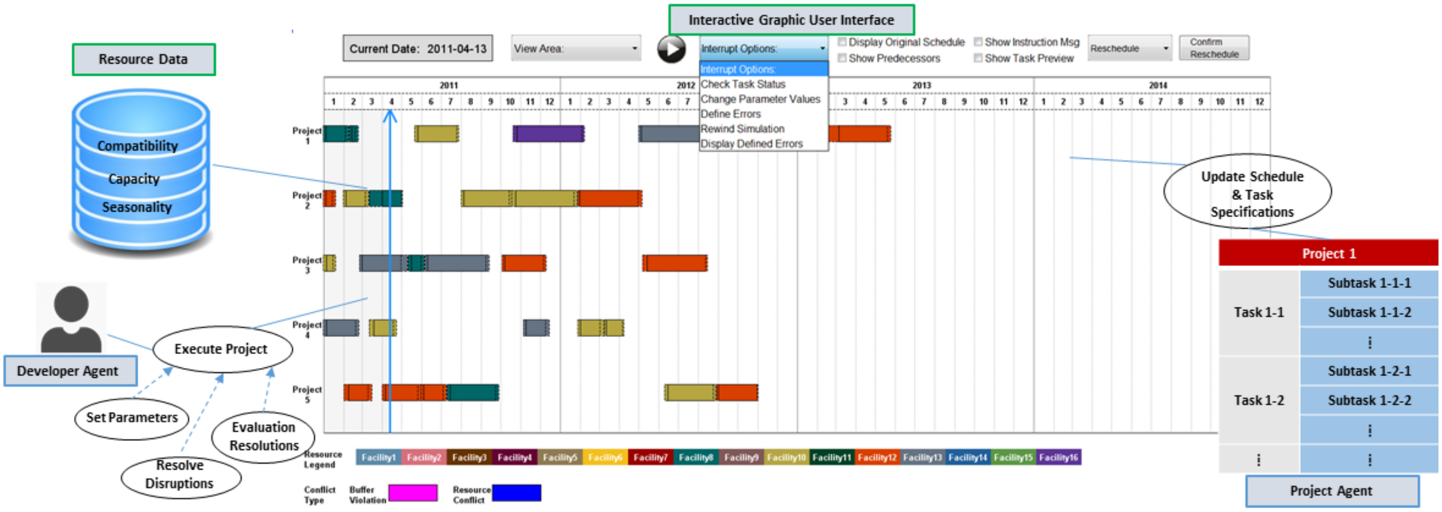

Figure 1: Proposed Simulation System Structure

disruptions and resource conflicts. Each project agent is responsible for bookkeeping its own specifications as well as updating its schedule visualization.

The proposed simulation has been implemented in AnyLogic® 7 Multi Simulation Software. The middle of Figure 1 is a snapshot of a running simulation as viewed in the interactive GUI. The interface design is based on the Gantt chart so as to minimize users' learning curve from conventional programme management tools such as Microsoft Project. In addition to making basic project information available in normal Gantt charts, our enhanced design also displays task-facility relations and schedule deviation to the current time. A vertical line in blue color representing "Today" moves with simulation progress to give an overview of the current programme status. Each facility is designated with a particular color which matches with that of tasks requesting it.

Several options are available for users to interrupt the running simulation. For instance, users can do on-the fly parameter change without restarting the simulation. Besides, the system enables user-in-the-loop experimentation in two ways: 1) defining subtask failures; and 2) manually adjusting schedules for future tasks. If users want to evaluate how the programme will be affected if certain tasks fail on specific days, they can schedule them either before or during the simulation run. By clicking on an unfinished task, a list of subtasks will pop up for users to choose any subtasks to be failed before the task completes. Manual scheduling is achieved by dragging and dropping future tasks. Once users are done with manual rescheduling, they will need to click on the "Confirm the Schedule" button and the system will check if there is any newly introduced resource conflicts before resuming simulation with the new schedule. The resource conflict can be resolved by manual rescheduling or the embedded optimization solver based on Choco 2.1.5 [4].

In the case of online disruption, developers have two alternatives to resolve it: 1) rerun the task with failed subtask; or 2) move the failed subtask to downstream tasks. If going with the first option, the failed task will be rerun one day after the failure date. Depending on the occurrence day of failure, rescheduling the failed task may affect the downstream tasks to a different extent. For the second resolution, the developer needs to select a target task from downstream to accommodate the failed subtask. A qualified target task should have a requested facility that has the instrument capacity to execute the failed subtask. Since the delay resulted from moving failed subtask is very short compared to that caused by rerunning the task, we assume it can be ignored in the simulated programme. However, this resolution will introduce additional cost due to reconfiguration of the target task. Therefore, users' decision between the two resolutions can vary according to their preference on shorter completion time or less cost.

3. CONCLUSIONS AND FUTURE WORK

In this paper, we have developed an interactive agent-based simulation system to provide decision support for a real-life industrial product development programme. The developed system offers an interactive GUI for user-in-the-loop experimentation. We are exploring several directions for future research. For example, a negotiation mechanism is under development to automate the procedure of finding the best available target task for moving failed subtask. In addition, more complexities can be incorporated by relaxing some constraints.

REFERENCES

[1] S. Adhau, M. Mittal, and A. Mittal. A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach. *Engineering Applications of Artificial Intelligence*, 25(8):1738–1751, 2012.

[2] J. A. Araúzo, J. Pajares, and A. Lopez-Paredes. Simulating the dynamic scheduling of project portfolios. *Simulation Modelling Practice and Theory*, 18(10):1428–1441, 2010.

[3] T. Chaari, S. Chaabane, N. Aissani, and D. Trentcsaux. Scheduling under uncertainty: Survey and research directions. In *Proceedings of Advanced Logistics and Transport (ICALT)*, pages 229–234. IEEE, 2014.

[4] F. Laburthe and N. Jussien. Choco solver documentation. *Ecole de Mines de Nantes*, 2011.