Analysis and Design of Integrated Vehicle and Crew Scheduling System Using GAIA Methodology

Abdul Samad Shibghatullah
School of Information System, Computing and Mathematics
Brunel University, Uxbridge, UB8 3PH, UK. cspgass@brunel.ac.uk

Nazri Abdullah
College of Information Technology, Universiti Tenaga Nasional,
43009 Kajang, Selangor, Malaysia. anazri@uniten.edu.my

Abstract

An integrated vehicle and crew scheduling is a complex problem to solve because of the large number of resources that need to be managed, and complexity of rules regarding allocating vehicle and crew shifts. However, there are potential benefits from integrating both scheduling including cost saving in regional area where routes have fewer relief points. The aim of this research is to design the integrated vehicle and crew scheduling system using Multi-Agents System (MAS). The paper presents agents analysis and design using Gaia methodology.

Keywords: *Vehicle Scheduling, Crew Scheduling, Integrated Scheduling, Multi-Agent System, Gaia Methodology*

1. Introduction

Buses are most used means of public transport especially in big metropolitan areas. For example in New York, there are 298 routes served by 4860 busses, Paris has 246 routes served by 3860 busses and London has 700 routes served by 6,500 scheduled busses [2]. There are several important steps in operational and scheduling planning of buses; timetabling, vehicle scheduling, crew scheduling and crew rostering. Most of these steps are treated separately due to their inherent complexity [4]. This article concentrates on integrating vehicle and crew scheduling problem.

An integrated vehicle and crew scheduling is a complex problem to solve because of the large number of resources that need to be managed, and complexity of rules regarding allocating vehicle and crew shifts. However, there are potential benefits from integrating both scheduling including cost saving in regional scenario where routes have fewer relief points. The problems in long vehicle rotations are either hard to cover with legal shifts or covered at very high cost. In this situation, it is very hard to find an optimal schedule. Thus, a way to deal with this is by combining the process of vehicle and crew scheduling to produce acceptable results.

There are few researches in this area [1][5][6][7]. Most of them follow the traditional way of scheduling which is sequential approach. Freling et al. [5][6] have proposed a dynamic integrated bus and crew scheduling system that will reschedule the crew and bus simultaneously, whenever unpredictable events or late occur. Several rescheduling could occur in a single day. The method used mathematical programming and stochastic programming. The method produced good results but there a few assumptions in the research that are not feasible in a real world. First, the passengers have a higher priority than crew. Thus, there is a possibility of violation of the crew rules whenever a bus late occur before the break time. The crew has to shorten the break or not take a break just to make sure the bus operate on-time. Although this is appropriate to guarantee the bus services run smoothly, the EU (European Union) driving rules should not be broken. Furthermore, this is not acceptable to the crew. Second, a trip can only start late due to a delay of the

vehicle and thus not due to the crew. This assumption is not real due to the fact that crew is one of the courses of unpredictable events. Third, the number of vehicles and crews is unlimited. This is not possible. Most of the time, bus company has a limited number of bus and crews.

The approach of MAS to scheduling is not necessarily the best approach when compared with existing mathematical approaches such as integer linear programming and column generation. Mathematical approaches have the ability to search optimal or near optimal schedule but they also have some limitations, for example, they are usually slow to produce results in real-time because they are computationally intensive when it comes to complex situation. Speed is an important issue when it comes to day-to-day operation management. Also mathematical methods do not necessarily capture all requirements or preferences of every crew. Whilst in MAS, the computational effort is dramatically reduced because each agent knows its attributes and tries to solve the problem through negotiation with relevant agents (not with every agent), and each agent can also capture requirements and preferences of its owner. MAS have many ways for resolving conflicts of resources. Probably the most widely used is Smith's Contract Net and its variants. Other types include voting mechanism, auction mechanisms, and market mechanisms [3][8].

The aim of this research is to design the integrated vehicle and crew scheduling system using MAS. This paper presents the vehicle and crew scheduling problem and demonstrates the use of Gaia as a tool for analyzing and designing an agent-based system. This paper has been organized as follows: Section 2 gives overview on Gaia methodology and explains why it is has been chosen for this research. Section 3 presents the propose analysis models and section 4 describes the design models of crew scheduling system. Section 5 concludes the discussion and suggests further research in the matter.

2. Gaia methodology

Gaia is a methodology for agent-oriented analysis and design. The methodology is introduced by Wooldridge at al. [9], and was extended by Zambonelli et al. [10]. The extended version introduced the concept of organizational abstraction, including organizational rules, structure and pattern; and add architectural design phase. Gaia is used because it is easy to learn and use, and flexible. It is intended to allow an analyst to systematically move from a statement of requirements to a design that is sufficiently detailed that it can be directly implemented. Analysis and design can be thought of as a process of increasingly developing detailed models of the system to be constructed. It provides an agent-specific set of concepts through which a software engineer can understand and model a complex system.

In Gaia, there are analysis and design phases. The objective of the analysis stage is to develop an understanding of the system and its structure (without reference to any implementation details). This understanding is captured in the system's organisation. An organization can be thought of as a collection of roles, that takes certain relationships to one another, and that take part in systematic, institutionalised patterns of interactions with other roles. A role is defined by four attributes: responsibilities, permissions, activities, and protocols. A role is also identified with a number of protocols, which define the way in which an agent can interact with other roles. The links between roles are represented in the interaction model. The outputs of analysis phase is roles model and interaction model.

The Gaia design process involves generating three models; agent model, services model, and acquaintance model. The agent model identifies the agent types that will make up the system, and the agent instances that will be instantiated from these types. The services model identifies the main services that are required to realize the agent's role. Finally, the acquaintance model documents the lines of communication between the different agents.

3. The proposed analysis model

This section will present the analysis model of the integrated vehicle and crew scheduling problem using Gaia methodology. The objective of the analysis stage is to identify roles and interaction between them. For each role, identify and document the associated protocols. Protocols are the patterns of interaction that occur in the system between the various roles. The outputs of analysis phase is roles model and interaction model.

3.1 Identify roles

In this system four roles have been identified. The roles are:

a) Crew (Figure 1) – A crew is a driver that works for the bus company. Their main responsibility is to drive bus according to prescribe schedule.
b) Bus (Figure 2) – A bus represents a vehicle that uses for everyday operation. It responsible for carrying passengers from one point to another.
c) Supervisor – A supervisor has various responsibilities. The main responsibility is to make sure all the buses run on time based on the predetermined schedule. Other than this responsibility, the supervisor has to manage the schedules (times, duty and ROTA), manage the crews, and the busses in everyday operation. Whenever there is any change or unavailable crew the supervisor must perform changes.
d) User Handler – Receive schedule request from user and oversee process to ensure appropriate schedule is returned. It also updates real time information for crews and duties.
e) Database Manager – It establish connection with the existing data store. It can query for pre-determine crew schedules and crew assignment.

Role: CREW (CR)
Description: This role is responsible to drive bus according to prescribed crew schedule
Protocols and Activities: SignIn, Drive, Break, SignOff, StandBy, RespondReAssignment
Permissions: reads Crew Schedules, Duty Assignment, Crew Details
Responsibilities:
Liveness:
CREW = (OnDuty | StandBy)$^{\omega}$
ONDUTY = (SignIn.(Drive||Break).SignOff)$^{\omega}$
Safety:
driving continuously <= 4.5 hours
break >= 45 minutes
total driving hours in a day <= 10 hours

Figure 1: Schema for role crew

Role: BUS (BS)
Description: This role is responsible to carry passenger according to prescribed schedule
Protocols and Activities: Moving, Stop, OnService
Permissions: reads Time Schedules
Responsibilities:
Liveness:
BUS = (Moving | Stop | OnService)$^{\omega}$
Safety:
running continuously <= 12 hours
break >= 45 minutes
total running hours in a day <= 10 hours

Figure 2: Schema for role bus

3.2 Interaction protocols

With the respective role definitions in place, the next stage is to define the associated interaction models for these roles. Here we focus with the two roles i.e. SUPERVISOR and USERHANDLER. This is because the initiator of the interaction is from either of these two. The SUPERVISOR role interacts with DATABASEMANAGER role to obtain schedules (RequestSchedules, Figure 3), and update schedules (UpdateSchedules). Then if SUPERVISOR role want to do reschedule, protocols ReSchedule is enacted.

The role USERHANDLER interacts with DATABASEMANAGER role to request for schedules (SchedulesRequest), and repair schedules (RepairSchedulesRequest). USERHANDLER role also asks DATABASEMANAGER role to update schedules (UpdateScheduleRequest).

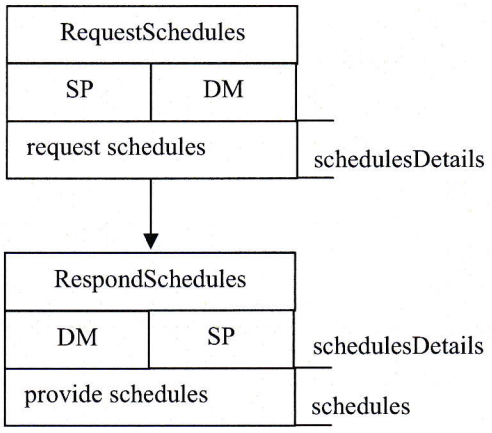

Figure 3: Definition of request schedules protocol

4. The proposed design model

Having completed our analysis of the proposed system, we now turn to the design phase. The aim of a design process is to transform the analysis models into a sufficiently low level of abstraction that any technique may be applied in order to implement agents. As described in Section 2, the Gaia design process involves generating three models i.e. agent model, services model and acquaintance model. The details are presented below.

4.1 Agent model

Agent model is the first model to be generated (Figure 4). In the figure it shows that a one-to-one correspondence between roles and agent types.

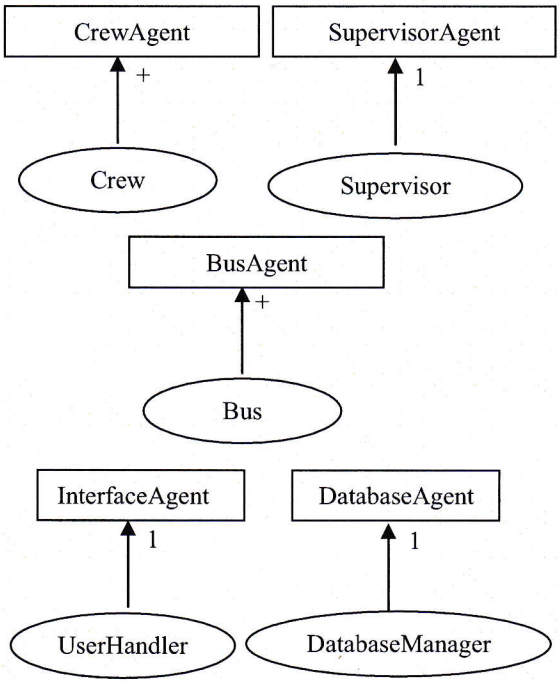

Figure 4: The agent model

4.2 Services model

The second model is the services model (Table 1). By a service, Gaia means a function of the agent. A service is a coherent block of activity in which an agent will engage. For each service that may be performed by an agent, it is necessary to document its properties. Table 1 shows all the services. It shows inputs, outputs, pre-conditions, and post-conditions of each service.

Table 1: An example of the services model

Service	Inputs	Outputs	Pre-condition	Post-condition
Sign-In	SignIn Time, DutyNo, CrewID	Status OnDuty	Status = off	Status ≠ off
On Driving	Start Time, DutyNo, CrewID	Status On Driving, Crew Driving Report	TotalDriving Hours <= 10, Continous Driving <=4.5	Total Driving Hours <= 10, Continous Driving <=4.5 hours
On Break	Start Time, CrewID	Status OnBreak	TotalHours Break >=45 minutes	Total Hours Break >=45 minutes
Sign-Off	End Time, CrewID	Status Off	Crew is available	Status = off

4.3 Acquaintance model

The final model is the acquaintance model, which shows the communication pathways that exist between agents (Figure 5).

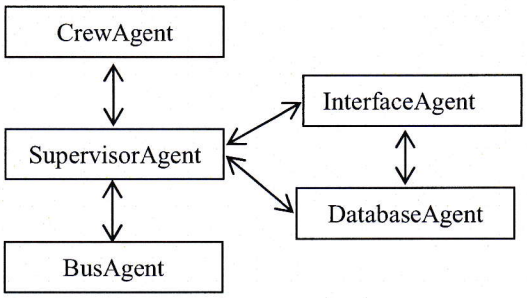

Figure 5: The acquaintance model

4.4 Discussion

This paper has presented agents analysis and design phase using Gaia methodology. Gaia is a Methodology which has been specifically tailored to the analysis and design of agent-based systems. Gaia provides an agent-specific set of concepts through which a software engineer can understand and model a complex system. In particular, Gaia encourages a developer to think of

building agent-based systems as a process of organizational design. In this research we use Gaia because of it is easy to learn and use, and flexible, so that it is iterative through analysis, design and implementation phases. It deals with internal aspects of agent and interaction between agents in a system. However, Gaia provides very basic notation in designing the interaction and communication between agents. Probably, other methodologies such as AUML, and TROPOS could be used as a useful companion to Gaia. The outputs of those phases are a set of roles, interaction protocols, agent model, service model, and acquaintance model. These outputs will be used for system implementation.

5. Conclusions

This paper has described the integrated vehicle and crew scheduling problem then presented agents analysis and design phase using Gaia methodology. In Gaia, there are analysis and design phases. The objective of the analysis stage is to develop an understanding of the system and its structure. This understanding is captured in the system's organization. The aim of design phase is to transform the analysis models into a sufficiently low level of abstraction that traditional design techniques may be applied in order to implement agents. Gaia is used because of it is easy to learn and use, and flexible, so that it is iterative through analysis, design and implementation phases. However, as discussed in above section one of the shortcoming of Gaia is the interaction and communication between agents are not properly captured. Communication is one of the important parts in agent-based system because agents are interacting through sending and receiving messages. This shortcoming can be covered by using other methodologies such as AUML, and TROPOS.

6. References

[1] K. Darby-Dowman et al., "Integrated Decision Support Systems for Urban Transport Scheduling: Discussion of Implementation and Experience", German, 1988, PP 226–239.

[2] G. Desaulniers, "Bus and driver scheduling in urban transit systems" [online]. Available via http://www.ima.umn.edu/talks/workshops/11-11-15.2002/desaulniers/desaulniers.pdf.[accessed: 2004].

[3] J. Ferber, Multi-agent systems: An introduction to distributed artificial intelligence, Addison-Wesley, 1999.

[4] S. Fores et al., "TRACS II: a hybrid IP/heuristic driver scheduling system for public transport", Operational Research Society, Palgrave Macmillan, 2002, pp. 1093-1100.

[5] R. Freling et al., "Models and algorithms for integration of vehicle and crew scheduling", Tech. Report EI2000-10/A, Econometric Institute, Erasmus University Rotterdam, 2000.

[6] R. Freling et al., "Applying an integrated approach to vehicle and crew scheduling in practice", Computer-Aided Scheduling of Public Transport: Lecture Notes in Economics and Mathematical Systems, Springer Verlag, 2001, pp. 73–90.

[7] K. Haase, and C. Friberg, "An exact branch and cut algorithm for the vehicle and crew scheduling". Computer-Aided Transit Scheduling: Lecture Notes in Economics and Mathematical Systems, Springer Verlag, 1999, pp. 63-80.

[8] M. Wooldridge, *An introduction to multiagent systems*, J. Wiley, 2002.

[9] M. Wooldridge et al., "The Gaia methodology for agent-oriented analysis and design". Autonomous Agents and Multi-Agent Systems, Springer Netherlands, 2000, pp. 285-312.

[10]F. Zambonelli et al., "Developing multiagent systems: the Gaia methodology". ACM Transactions on Software Engineering and Methodology, ACM Publications, 2003, pp.317-370. 2003.