Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

A PROPOSED MULTIAGENT MODEL FOR BUS CREW SCHEDULING

Abdul S. Shibghatullah
Tillal Eldabi
Jasna Kuljis

School of Information System, Computing and Mathematics
Brunel University, Uxbridge,
UNITED KINGDOM, UB8 3PH

ABSTRACT

Bus crew scheduling is a complex problem to solve because of the large number of resources that need to be managed, complexity of allocating crew shifts, rising cost of crew and unpredictability of traffic and crew availability. This causes a difficulty to maintain an optimal schedule. Existing systems are excellent in producing optimal or near optimal schedules. However, to maintain such optimality for day-to-day operations, crew scheduling systems need to extend their capabilities by enabling crew reassignment, a feature that is not currently available in currently existing automated scheduling systems. The aim of this research is to model the crew reassignment process using agents and simulate agents' behavior in order to establish ways of automating management of unpredictable events. The model should assist supervisor in managing everyday bus operations. The paper presents agents analysis and design using 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 (Desaulniers 2002). 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 (Fores et al. 2002). This article concentrates on crew scheduling problem. There are three major issues associated with crew scheduling problem: *complexity*, *cost*, and *unpredictability*. The complexity issue can be attributed to the fact that crew scheduling involves many variables, such as the driving hour rules, labor agreement rules, and crew availability. The cost aspect is attributed to the fact that crew expenses involve a large proportion of a bus' operational costs (Yunes et al. 2000).

The aim of this research is to model the bus crew scheduling problem with agents, and study the feasibility of using crew reassignment process as a way to cope with unpredictable events. This paper presents the bus crew scheduling problem and demonstrates the use of Gaia as a tool for analyzing and designing an agent-based models. This paper has been organized as follows: Section 2 presents the bus crew scheduling problem which focuses on the unpredictable events problem and the role of supervisor in managing it. Then proposes a framework of crew scheduling management system. Section 3 gives overview on Gaia methodology and explains why it is has been chosen for this research. Section 4 presents the propose analysis models and section 5 describes the design models of crew scheduling management system. Section 6 concludes the discussion and suggests further research in the matter.

2 CREW SCHEDULING PROBLEM

Generally, the main aim for developing any bus crew schedule is to achieve optimum and dynamic schedules. Optimum schedules mean that resultant schedules should minimize the operation cost, whilst dynamic means it is able to maintain such optimality throughout the scheduled duration. However, the main obstacle of keeping such optimality throughout day-to-day operation is unpredictable events such as, late crew, sick while on duty, or absenteeism without prior notice (Cheng and Chang 1999). Bus service usually operates in an unpredictable environment, especially in high-frequency routes in a busy city. Whenever an unpredictable event occurs, it affects bus operations even for other routes as well. In the UK, causes of unpredictable events are categorized into four categories, which are traffic, staff, mechanical and others (Copley et al. 2003).

A supervisor usually is the person who is responsible to make sure that the bus service operates smoothly. The supervisor has various responsibilities; such as making sure all the buses run on time based on the predetermined schedule. The bus company has to comply with the schedule that has been agreed with the local authority. If the

1-4244-0501-7/06/$20.00 ©2006 IEEE

company does not perform well, the contract will be suspended. Other responsibilities include managing schedules (times, crew and ROTA), managing crews, and the busses in everyday operations. These responsibilities are immensely hard especially when dealing with unpredictable events.

The supervisor has times schedules, crew schedules, and ROTA schedules. The times schedules show movement of each bus, times against specified locations, whilst crew schedules show the activity for every duty from sign-in to sign-off. ROTA schedules show which person is assigned to what duty in a certain week. Then, the supervisor has to manage resources (such as crews and busses). When a certain unpredictable event occurs the supervisor must perform appropriate adjustments to the schedule or change resource allocation. The times and crew schedule will remain the same and only reallocation or reassignment is done to cover the schedule (Shibghatullah et al. 2006). For instance, if a bus is broken down then a spare bus will be allocated to cover the remaining schedule. Similarly, if a crew is absent, then their duty will be assigned to a spare crew. Usually supervisors tend to manually reallocate or reassign. This is the general practice in almost bus companies.

Existing crew scheduling systems such as, TRACS (Fores et al. 2002; Kwan et al. 2004), HASTUS (Rousseau, and Blais 1985), and IMPACS (Wren et al. 1985) are good in producing optimal or near optimal original schedules. However, to maintain such optimality throughout day-to-day operations (dynamic reassignment), crew scheduling systems should extend their capabilities to enabling crew reassignment as well. Crew reassignment means that if any of the crew becomes unavailable, the crew schedule will remain the same, only their duty will be assigned to other available crew without rules violation. Why there is a need to have such a system? There is nothing wrong with manual reassignment as such, however, it is hard to manage when it involves large crews and duties and when quick decisions are required. The supervisor should always make sure that the driving hour rules are followed when reassigning drivers. This makes the process more prone to errors.

2.1 The Proposed Framework for Crew Scheduling Management System

The proposed system is aimed at assisting supervisors in reassigning crews if needed (in case of unpredictable events). Figure 1 demonstrates the positioning of the new system. The box denoted as “Crew scheduling management system” represents the proposed system. The proposed system should be linked to any existing systems which are crew scheduling system and ROTA system. The proposed system needs to obtain information on crew schedules that were generated by the existing crew scheduling system. User can view predetermined crew schedules, which includes; duty number, route number, garage name, number of spell, sign-in times, start driving times, finish driving time, start break time, finish break time, sign-off time, total driving time, and total duty time.

The proposed system should also be able to store real-time information concerning crew and crew schedules. This is to make sure that the system is capturing the status of crews and crew schedules, because when the system needs to re-assign crew, then the current status of crews and crew schedules need to be known. Real time information includes; sign-in times, start driving times, finish driving time, start break time, finish break time, current status of crew, sign-off time, total driving time, and driving time left. The crew schedules real time information includes; total covering time, covering time left, total covering point, and covering point left.

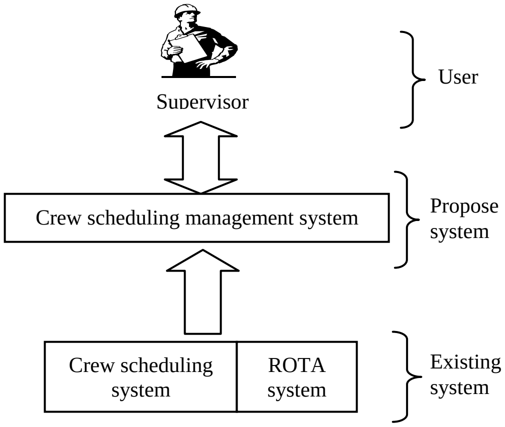


Figure 1: The Proposed Framework

The proposed system should also be able to reassign crews without violating EC Driving Hour Rules. Most importantly is that rules concerning break and daily driving hours (continuous driving hours should not exceed four and half, and total daily driving time should not exceed ten hours). The break should be at least forty-five minutes. Currently, reassignment should cover five events: (a) Crew comes late for duty; (b) Absent-without-prior notice; (c) Sick-on-duty; and (d) Unavailable part of the duty.

3 GAIA METHODOLOGY

Gaia is a methodology for agent-oriented analysis and design. The methodology is introduced by Wooldridge at al. (2000), and was extended by Zambonelli el al. (2003). 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.

4 THE PROPOSED ANALYSIS MODEL

This section will present the analysis model of the bus crew scheduling problem (as describe in Section 2) 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.

4.1 Identify Roles

In the bus crew scheduling system four roles have been identified. The roles are:

a) Crew (Figure 2) – A crew is a driver that works for the bus company. Their main responsibility is to drive bus according to prescribe schedule.
b) Supervisor (Figure 3) – 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.
c) User Handler (Figure 4) – Receive schedule request from user and oversee process to ensure appropriate schedule is returned. It also update real time information for crews and duties.
d) Database Manager (Figure 5) – It establish connection with the existing data store. It can query for pre-determine crew schedules and crew assignment.

4.2 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 crew schedules (RequestCrewSchedules, Figure 6a), duty assignment (RequestDutyAssignment, Figure 6b), real time crew information (RequestRealTimeCrew, Figure 6c), and real time duty information (RequestRealTimeDuty, Figure 6d). Then if SUPERVISOR role want to do reassignment, protocols ReAssignCrew (Figure 6e) is enacted.

The role USERHANDLER interacts with DATABASEMANAGER role to request for crew schedules (CrewSchedulesRequest, Figure 7a), and duty assignment (DutyAssignmentRequest, Figure 7b). USERHANDLER role also asks DATABASEMANAGER role to update crew (UpdateCrewRequest, Figure 7c) and to update duty information (UpdateDutyRequest, Figure 7d).

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) ω
    ONDUTY = (SignIn.(Drive||Break).SignOff)ω
```

Safety:

```
    driving continuously <= 4.5 hours
    break >= 45 minutes
    total driving hours in a day <= 10 hours
```

Figure 2: Schema for Role Crew

Role: **SUPERVISOR (SP)**

Description: This role is responsible to manage crew and make sure buses run on time.

Protocols and Activities: RequestCrewSchedules, RequestDutyAssignment, RequestRealTimeCrew, RequestRealTimeDuty, ReAssignCrew, ReAssignDecision

Permissions: reads Crew Schedules, Duty Assignment, Real-Time Crew Information, Real-Time Duty Information, change Duty Assignment

Responsibilities:

Liveness:

SUPERVISOR = (ReAssignment)*
REASSIGNMENT = (RequestCrewSchedules‖
RequestDutyAssignment).(RequestRealTimeCrew‖
RequestRealTimeDuty).ReAssignCrew

Safety: All duties covered

Figure 3: Schema for Role Supervisor

Role: **USERHANDLER (UH)**

Description: Receive schedule request from user and oversee process to ensure appropriate schedule is returned. Record the real time information for crews and duties.

Protocols and Activities: CrewSchedulesRequest, PresentCrewSchedules, DutyAssignmentRequest, PresentDutyAssignment, UpdateCrewRequest, UpdateDutyRequest, InformUpdateStatus

Permissions: reads User Request, Update Request

Responsibilities:

Liveness:

USERHANDLER = (ScheduleRequest ‖
UpdateRequest)$^\omega$
SCHEDULEREQUEST = (CrewScheduleRequest.
PresentCrewSchedules)|(DutyAssignmentRequest
.PresentDutyAssignment)
UPDATEREQUEST = (UpdateCrewRequest |
UpdateDutyRequest).InformUpdateStatus

Safety: true

Figure 4: Schema for Role User Handler

5 THE PROPOSED DESIGN MODELS

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 3, the Gaia design process involves generating three models i.e. agent model, services model and acquaintance model. The details is presented below.

Role: **DATABASEMANAGER (DM)**

Description: It connects with the existing data store. It can query for pre-determine crew schedules and crew assignment.

Protocols and Activities: RequestCrewSchedules, RespondCrewSchedules, RequestDutyAssignment, RespondDutyAssignment, RequestRealTimeCrew, RespondRealTimeCrew, RequestRealTimeDuty, RespondRealTimeDuty, UpdateCrewRequest, RespondCrewUpdate, UpdateDutyRequest, RespondDutyUpdate

Permissions: reads query detail, real time information

Responsibilities:

Liveness:

DATABASEMANAGER = (FindSchedules ‖
UpdateRealInfo)ω
FINDSCHEDULES = (RequestCrewSchedules.
RespondCrewSchedules)‖ (RequestDutyAssign
ment.RespondDutyAssignment)‖(RequestReal
TimeCrew.RespondRealTimeCrew)‖
(RequestRealTimeDuty. RespondRealTimeDuty)
UPDATEREALINFO = (UpdateCrewRequest.
RespondCrewUpdate)‖(UpdateDutyRequest.
RespondDutyUpdate)

Safety: true

Figure 5: Schema for Role Database Manager

5.1 Agent Model

Agent model is the first model to be generated (Figure 8). In the figure it shows that a one-to-one correspondence between roles and agent types.

5.2 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 show inputs, outputs, pre-conditions, and post-conditions of each service.

5.3 Acquaintance Model

The final model is the acquaintance model, which shows the communication pathways that exist between agents (Figure 9).

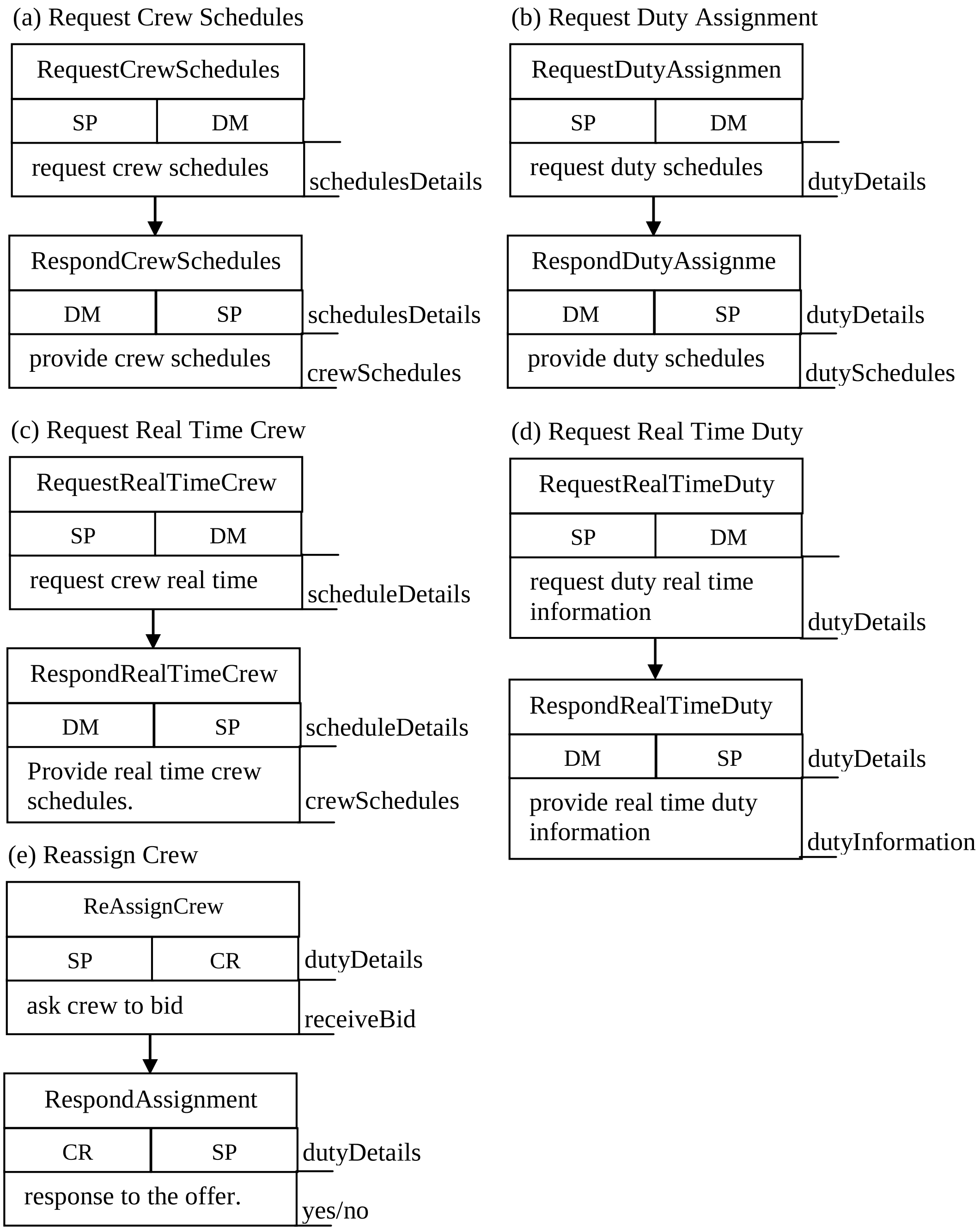
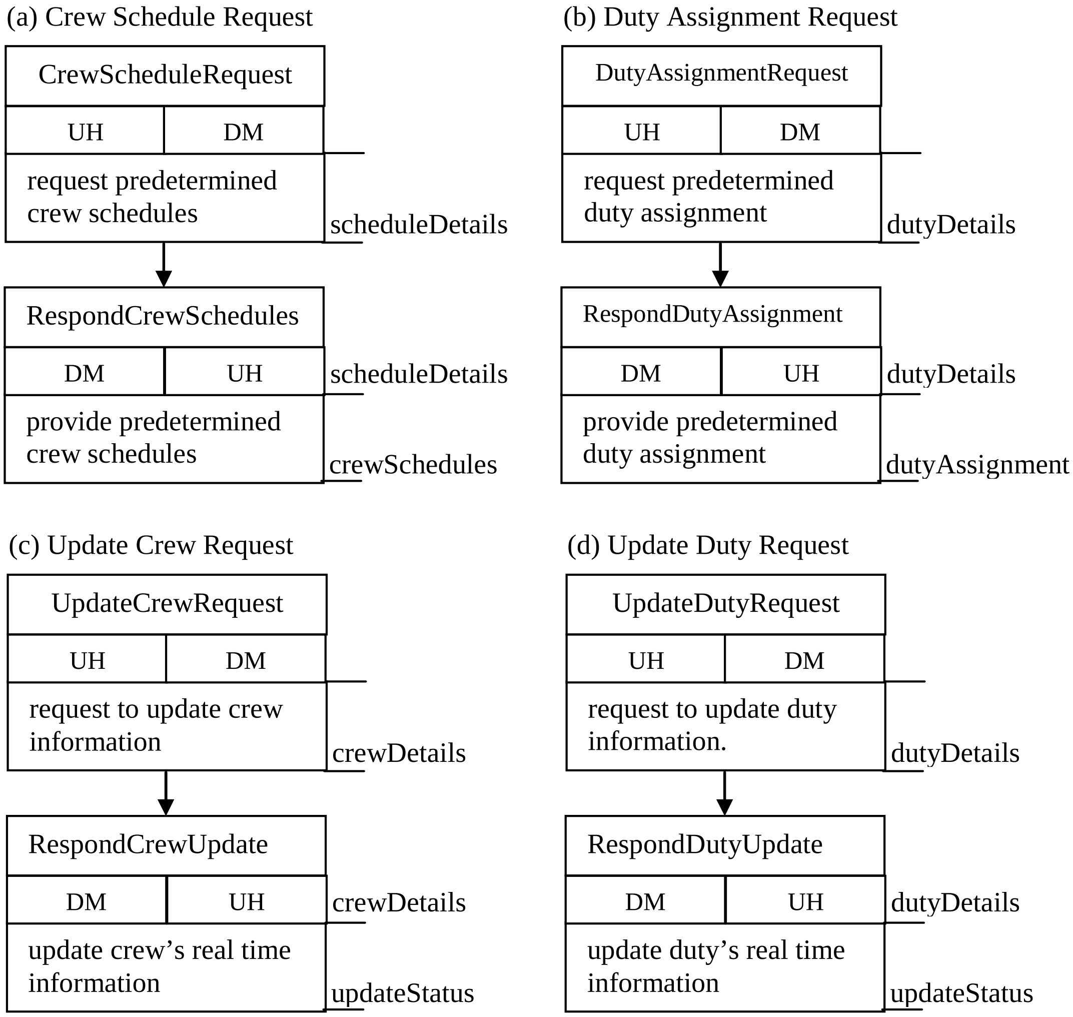


Figure 6 : Definition of Protocols Associated with the SUPERVISOR Role


(a) Crew Schedule Request

(b) Duty Assignment Request

(c) Update Crew Request

(d) Update Duty Request

Figure 7: Definition of Protocols Associated with the USERHANDLER Role

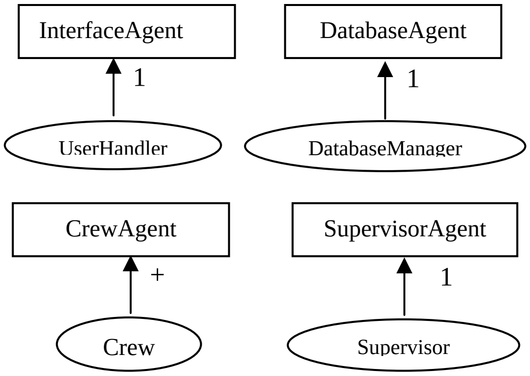


Figure 8. The Agent Model

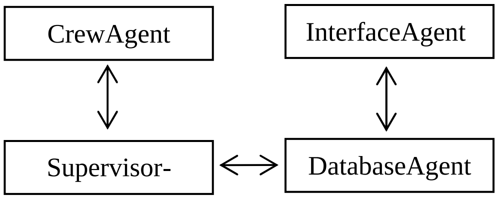


Figure 9. The Acquaintance Model

Table 1: The Services Model

Service	Inputs	Outputs	Pre-condition	Post-condition
Sign-In	SignInTime, dutyNo, crewID	statusOnDuty	status = off	status ≠ off
on driving	StartTime, dutyNo, crewID	StatusOnDriving, crewDrivingReport	totalDrivingHours <= 10, continousDriving <=4.5 hours	totalDrivingHours <= 10, continousDriving <=4.5 hours
On break	StartTime, crewID	statusOnBreak	totalHoursBreak >=45 minutes	totalHoursBreak >=45 minutes
Sign-Off	EndTime,crewID	statusOff	Crew is available	status = off
on Standby	crewDetails	statusStandBy	status = off	status = standBy
Obtain crew schedules	scheduleDetails	CrewSchedules	true	true
Obtain duty assignment	dutyDetails	Duty Assignment	true	true
Obtain real time crew information	crewDetails	crewInformation	true	true
Reassign Process	dutyDetails	receiveBidding	Crew unavailable	Crew status = on
Reassign Decision	crewDetails	newAssignment	totalDrivingHours <= 10, continousDriving <=4.5 hours	totalDrivingHours <= 10, continousDriving <=4.5 hours
Update real time crew information	crewDetails	Update is success	true	duty information is updated
Update real time duty information	dutyDetails	Update is success	true	Duty information is updated

6 CONCLUSIONS AND FUTURE WORK

This paper has described the bus crew scheduling problem, focusing on the unpredictable events problem. This problem causes a difficulty to maintain optimal schedules. Existing crew schedule systems are not capable of handling such problem. Existing systems aimed at solely developing optimal schedules. Once a crew schedule is produced then the resultant schedules are manually managed by supervisors at garage.

This paper has presented agents analysis and design phase using Gaia methodology. 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. 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 output 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. There are many difficult issues that need to be addressed. These include how to control the negotiation/ communication process between agents in case of a large number of simultaneous unpredictable events. Deadlock might occur when hundreds of agents sending and receiving messages while negotiating to reassign crews. A control mechanism is needed to prevent this deadlock.

REFERENCES

Cheng, J.H., and Y.H. Chang. 1999. Application of a fuzzy knowledge base on bus operations under uncertainty. In *Proceeding of the Fuzzy Systems Conference*. 1355-1360. IEEE International.

Copley, G., J. Dodgson, M. Bright, D. Coombe, B. Davidson, and G. Barrett. 2003. Second assessment report: 10 year transport plan monitoring strategy. Hertfordshire, UK: Commission for Integrated Transport.

Desaulniers, G. 2002. 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 June 13, 2004].

Fores, S., L. Proll, and A. Wren. 2002. TRACS II: a hybrid IP/heuristic driver scheduling system for public transport. *Operational Research Society*. 53: 1093-1100.

Kwan, A.S.K., M.E. Parker, R.S.K Kwan, S. Fores, L. Proll, and A. Wren. 2004. Recent advances in TRACS. In *Preprints of the 9th International Conference on*

Computer-Aided Scheduling of Public Transport; San Diego, California, USA.

Rousseau J.M., and J.Y. Blais. 1985. HASTUS: an interactive system for buses and crew scheduling. In *Proceedings on Computer-Aided Scheduling of Public Transport*, ed. J.M. Rousseau, 45-60. North-Holland.

Shibghatullah, A.S., T. Eldabi, and G. Rzevski. 2006. The requirements for a dynamic bus crew scheduling system. In *Proceedings of the 10th International Conference on Computer-Aided Scheduling of Public Transport*, Leeds, UK.

Wooldridge, M., N. R. Jennings, and D. Kinny. 2000. The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*. 3: 285-312.

Wren A, B.M. Smith, and A.J. Miller. 1985. Complementary approaches to crew scheduling. In *Proceedings on Computer-Aided Scheduling of Public Transport*, ed. J.M. Rousseau, 263-278. North-Holland.

Yunes, T.H., A.V. MOURA, and C.C.D. SOUZA. 2000. Solving very large crew scheduling problems to optimality. In *Proceedings of the 2000 ACM symposium on Applied computing*, 446-451. ACM Press.

Zambonelli, F., N. R. Jennings, M. J. Wooldridge. 2003. Developing multiagent systems: the Gaia methodology. *ACM Transactions on Software Engineering and Methodology*. 12: 317-370.

AUTHOR BIOGRAPHIES

ABDUL SAMAD SHIBGHATULLAH is a researcher at School of Information System, Computing and Mathematics at Brunel University, United Kingdom. He received a B. Accounting from the Universiti Kebangsaan Malaysia of Bangi. He received his M.Sc. in Computer Sciences from Universiti Teknologi Malaysia, Skudai. Now, he is pursuing his Ph.D. at Brunel University. His research interests includes scheduling, multi-agent system and simulation. His email address is <cspgass@brunel.ac.uk>.

TILLAL ELDABI is a lecturer at the School of Information Systems, Computing and Mathematics at Brunel University, UK. He received a B.Sc. in Econometrics and Social Statistics from the University of Khartoum. He received his M.Sc. in Simulation Modelling and his Ph.D. from Brunel University. His research is in aspects of healthcare management and the intervention of simulation and his main research also concentrates on the economy of healthcare delivery. He is looking to exploit the means of simulation on the wider information systems management area to assist in problem understanding. Dr. Eldabi's email address is <tillal.eldabi@brunel.ac.uk>.

JASNA KULJIS is a Reader in the School of Information Systems, Computing and Mathematics at Brunel University. She gained her Dipl. Ing. degree in Theoretical Mathematics from the University of Zagreb, Croatia. She gained her M.S. in Information Science from the University of Pittsburgh, USA, and a Ph.D. in Information Systems from the London School of Economics, University of London. Her current research is in Human Computer Interfaces. She is mostly interested in the design of graphical user interfaces and in the development of new paradigms that would further enhance the usability of interactive computer systems. She is Director of the VIVID Research Centre at Brunel University. Dr. Kuljis' email address is <jasna.kuljis@brunel.ac.uk>.