PB2001-107263

Dynamic Coordination Framework for Resource Allocation in Trucking Operations

Satish Bukkapatnam

Greg Gao

Vimal Krishnan

University of Southern California

5/10/01

METRANS
Transportation Center
USC / CSULB

1. Report No. FHWA/CA/OR-2001/14	2. Government Accession No.	3. Recipients Catalog No.
4. Title and Subtitle Dynamic Coordination Framework for Resource Allocation in Trucking Operations		5. Report Date May 10, 2001 6. Performing Organization Code
7. Author/s Satish Bukkapatnam, Greg Gao, and Vimal Krishnan		8. Performing Organization Report No. 99-27-2000-7
9. Performing Organization Name and Address California Department of Transportation New Technology & Research, MS#83 P.O. Box 942873 Sacramento, CA 94273-0001		10. Work Unit No. (TRAIS) 11. Contract or Grant No. 65A0047
12. Sponsoring Organization Name and Address California Department of Transportation Washington, DC 20590-0001		13. Type of Report and Period Covered Research, July 1, 1999 - June 31, 2000 14. Sponsoring Agency Code

15. Supplementary Notes

This project was performed in cooperation with the U. S. Department of Transportation.

16. Abstract

The research conducted under this project has led to the architecting of a novel multi-agent framework to coordinate dynamic resource allocation. The research approach has been developed for a specific problem of allocating in real-time, trucks to "on-call" pickup requests, which is a common and important issue in several parcel and goods movement industries.

Here, each entity including different trucks and the dispatcher is represented by an autonomous and self-interested computational entity referred to as an *agent*. All on-call pickup jobs are announced to the trucks by the dispatching agent. The truck agents bid for the pickup order based on their knowledge about the current and anticipated loads as well as the location of the announced pick-up job, and they negotiate with the dispatching agent to win the pickup order. During negotiation, the bids of the truck agents as well as counter-bids from the dispatching agent are progressively revised until a bid value acceptable to both a truck agent as well as the dispatcher is reached. At this point, the pick-up job is awarded to the truck.

This research has also led to the emergence of understandings on the effects of some (heuristic) evaluation and pricing policies, as well as reinforcement learning strategies on the overall performance of a coordination system. The coordination system for resource allocation resulting from this research was applied to a specific resource allocation scenario resembling local goods pickup operations of a major company involved in commercial goods movement. Our simulation studies, built based on an actual goods pickup environment, reveal the potential for a multi-agent negotiation based method for real-time coordination of pickup and other resource allocation operations.

17. Key Words Truck routes, Resource allocation	18. Distribution Statement No Restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161		
19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. Of Pages 42	22. Price

Form DOT F 1700.7 (8-72) **Reproduction of form and completed page is authorized.**

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, and California Department of Transportation in the interest of information exchange. The U.S. Government and California Department of Transportation assume no liability for the contents or use thereof. The contents do not necessarily reflect the official views or policies of the State of California or the Department of Transportation. This report does not constitute a standard, specification, or regulation.

Disclosure

Project was funded in entirety under this contract with California Department of Transportation.

TABLE OF CONTENTS

Abstract

The research conducted under this project has led to the architecting of a novel multi-agent framework to coordinate dynamic resource allocation. The research approach has been developed for a specific problem of allocating in real-time, trucks to "on-call" pickup requests, which is a common and important issue in several parcel and goods movement industries.

Here, each entity including different trucks and the dispatcher is represented by an autonomous and self-interested computational entity referred to as an *agent*. All on-call pickup jobs are announced to the trucks by the dispatching agent. The truck agents bid for the pickup order based on their knowledge about the current and anticipated loads as well as the location of the announced pick-up job, and they negotiate with the dispatching agent to win the pickup order. During negotiation, the bids of the truck agents as well as counter-bids from the dispatching agent are progressively revised until a bid value acceptable to both a truck agent as well as the dispatcher is reached. At this point, the pick-up job is awarded to the truck.

This research has also led to the emergence of understandings on the effects of some (heuristic) evaluation and pricing policies, as well as reinforcement learning strategies on the overall performance of a coordination system. The coordination system for resource allocation resulting from this research was applied to a specific resource allocation scenario resembling local goods pickup operations of a major company involved in commercial goods movement. Our simulation studies, built based on an actual goods pickup environment, reveal the potential for a multi-agent negotiation based method for real-time coordination of pickup and other resource allocation operations.

1. INTRODUCTION

In crowded cities, well architected transportation and information infrastructure is necessary for efficient goods movement. A recent draft of California Transportation plan for goods movement, developed by CALTRANS, identifies capacity and congestion, safety, geometric and surface conditions, and intermodal connections as four major factors affecting the modern day trucking operations. The emerging Information Technologies including mobile communications and Geographical Information Systems, have not been effectively harnessed to deliver useful engineering solutions to address the aforementioned issues in trucking operations (Regan and Golob, 1999). Also, earlier research (Fischer et al., 1996) has determined that performance of trucking operations can be significantly improved through

- optimal and real-time route generation and route guidance for truck movement, and
- timely, dynamic and near-optimal allocation of trucks and other resources to transport goods

In transportation literature, vehicle routing, pickup and delivery, and dial-a-ride with various load and time constraints are typically modeled as deterministic or stochastic optimization problems. These models are solved using "traditional" mathematical programming methods as well as monolithic algorithms founded thereon. This solution methodology does not meet the requirements of modern day trucking operations where tasks are created dynamically, and the customers of goods movement, and hence the pickup and delivery location, are spatio-temporally distributed over a metropolitan landscape. The task allocation system must enable customers to place orders without any concern about how various tasks are allocated to different vehicles. Also, if a vehicle fails to execute a committed task, another vehicle must complete the failed task. Finally, the task allocation must be planned and executed in real-time, considering vehicle routing as well as various *possible* scenarios.

These requirements make multi-agent systems (MAS) the most appropriate means to coordinate task allocation among the distributed decision makers including customers, vehicles and the trucking company offices. Besides, solutions arrived through negotiations are viewed to be superior to those evolved from a centralized task allocator.

For the past five years, several researchers have applied MAS to solve a few common issues in the transportation domain. The work in MAS by Fischer (Fischer et al., 1996) is one of the best known research efforts in cooperative transportation scheduling, which is solved using Contact Net Protocol for task distribution among companies and task assignment among vehicles. The approach, however, assumes a deterministic scenario and ignores uncertainty. COSY/DASEDIS (Burmeister and Sundermeyer, 1992; Burmeister et al.,1997) is an agent based work in traffic simulation and guidance. Several agent applications are reported in air traffic control (Ljunberg and Lucas, 1992) and negotiation among airlines for landing in a airport (Sastry et al., 1995) using game theoretic approaches. Wellman presents a market-oriented approach (WALRAS algorithm) to solve an atemporal and steady-state multi-commodity problem, distributedly (Wellman, 1993).

The reported research is an attempt to employ the emerging technologies for providing effective and thoroughly validated methodologies for resource allocation in trucking operations. The research conducted under this project has led to the adaptation of a novel multi-agent framework to coordinate dynamic resource allocation. The framework was founded on dynamic game theory (Basar and Olsder, 1995) to evolve near-optimal, robust and tractable policies to allocate trucks and other resources to move goods from point to point. The resource allocation method resulting from this research was applied to a specific resource allocation scenario resembling operations of UPS - a major company involved in commercial goods movement. The remainder of this report is organized as follows: Section 2 presents an account of operations of UPS which shows that dynamic allocation of resources may be commercially viable; Section 3 describes our research methodology; Section 4 describes the essential architecture of the adapted MAS. Descriptions of the main elements of our methodology (namely, the strategies for coordination and learning) are presented in Sections 5 and 6, respectively. Details of our computer simulation implementations as well as results from our simulation experiments are presented in Sections 7 and 8.

2. MOTIVATION: SIMULATION STUDIES ON THE CURRENT PICK-UP PRACTICE

United Parcel Service is the nation's largest package transportation service. The company currently has a regular client base in addition to the individuals who also employ their services. Deliveries vary with individual customer needs and are predominantly seasonal. Peak season is experienced towards the end of the year, building up to the holiday season. Pickups however are fairly consistent, and the drivers (known as service providers) make these pickups as a regular part of their route. The route traced by a driver is determined by the type of area covered (commercial or residential, or both). Optimal routes are developed keeping in mind factors like one-ways streets and typical traffic conditions, and the like. Time standards are developed for every aspect of the driver route, by routine meticulous time studies and are specific to the coverage area.

For enhanced administrative ease, every city that UPS delivers to is fragmented into sections known as *loops*. These loops are further subdivided into areas known as *units*. It is typical to have as many as five or six drivers in any one loop. For the remainder of the report, these units will be referred to as coverage areas. Typically, there is only one driver within one coverage area. This assumption is maintained throughout the analysis.

Every "center" is assigned a section of a city that is it's designated coverage area. The center management team comprises a Center Manager and typically three On-Road Supervisors (as is the case with the Rodeo Center). The On Road Supervisors are responsible for determining everything from staffing levels to constantly maintaining contact (through the DIADs) with the drivers while they are on-road. It is also their responsibility to respond to requests for unscheduled pickups for certain air packages, that may arise while the drivers are already on their routes. These are referred to as on-call air pickups.

Upon receiving a request for service, the on-road supervisors relay the pertinent information to the drivers through appropriate wireless system integrated with the Digital Information Acquisition

Devices (DIADs). This information includes the address of the request, the package type, number of packages, the estimated time in which the package will be ready and also the closing time of the establishment. All this information is dispatched to the driver assigned to the unit in which the request arises. The driver then has a few minutes to respond to the dispatch. The two alternatives available to the driver are to either accept the dispatch, or the driver can also "kick-back" the request whereupon the management team will dispatch the pickup information to another driver. The reasons for a kick-back may include anything from insufficient capacity in his/her package car, to personal reasons like health or the need to finish the route quickly.

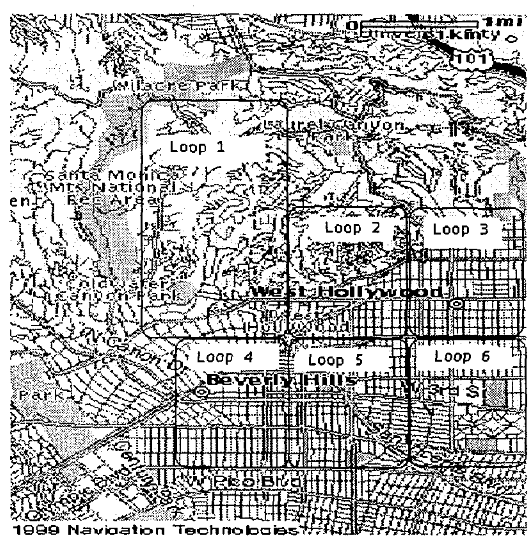


Figure 1. Illustration of loops in UPS pickup operations

? Figure 1

This method is not necessarily the best system. The assignments are made based solely on the driver's coverage area and not necessarily on the closest driver to the point of origin of the request for service. Additionally, the center management team needs to know the approximate whereabouts of the drivers at all times of the day. This is a lot of information for an individual to remember with a high level of fidelity on a regular basis. The containment of this amount of knowledge with one individual can hamper smooth functioning of the center, creating a knowledge gap too huge to surmount in some cases.

UPS, in an effort to streamline its operations, is always trying to reduce driver on-road miles and consequently, the amount of time the drivers spend on the road (known as windshield hours). There is a definite cost associated with every mile the drivers traverse. Inefficient dispatching could result in hundreds of additional miles a week. This figure extrapolated corporate-wide, amounts to millions of dollars.

To identify alternate methods that would make the assignment process more efficient, a simulation model of the driver route was developed. The model differed from the current practices in that the assignment of a pickup request was made on the basis of the driver closest to the source of the request for service. Initial models were limited to two adjacent loops. Priorities were assigned in a random manner to simulate kickbacks. The results from this simulation were compared to a simulation of the current practice where a single driver responds to any and all requests originating within the loop. The latter case resulted in 19 missed pickups out of a possible 100. The model involving the two drivers had a slighter lower number of missed pickups of 16 out of a possible 100.

This served as the premise to expand the model wherein loops all around the study loop were considered (i.e., 9 loops in all). The assignment was once again based on the closest driver to the point of origin of the request for service. This time the results were even more encouraging. The number of missed pickups was reduced to 14 out of a possible 100. These results clearly indicate that there is potential for improvement from the current method of package allocation. In addition to the reduced number of missed pickups is the reduction in driver miles, which again translates into cost savings. The study clearly establishes that the current system leaves a lot to be desired, and the system as a whole could use some corrective action.

3. RESEARCH METHODOLOGY

An existing MAS system, described in section 4, has been adapted for coordinating pickup operations similar to what one encounters in UPS. Resource allocation is achieved through negotiations among the various entities present in the scenario. The overall methodology involves modeling the pickup scenario as a MAS negotiation problem with learning. Here, each entity including different trucks and the dispatcher is represented by an autonomous and self-interested computational entity referred to as an *agent*. All on-call pickup jobs are announced to the trucks by the dispatching agent. The truck agents bid for the pickup order based on their knowledge about the current and anticipated loads as well as the location of the announced pick-up job, and negotiate with the dispatching agent to win the pickup order. During negotiation, the bids of the truck agents as well as counter-bids from the dispatching agent are progressively revised until a bid value acceptable to both a truck agent as well as the dispatcher is reached. At this point, the pick-up job is awarded to the truck.

The overall incentive for a truck driver, and overall profits for UPS, depends on the number of on-call pickups made. We use the protocol described in Section 5 to facilitate coordination among multiple agents in the scenario. The major stage in the coordination involves how one evaluates proposals from various parties. The evaluation method is presented in Section 6. We also developed a reinforcement learning method to gradually improve the evaluation of proposals and submission counterproposals as presented in Section 7.

4. MULTI-AGENT SYSTEMS ARCHITECTURE

We have adapted British Telecom's freeware multi-agent system called ZEUS. ZEUS's agent development tools provide basic components, communication language and visualized administration user interface for ease of implementation [5] [6]. Details of the architecture and the important modules therein are described in the following two subsections.

Agent society structure

The context diagram (Figure 2) illustrates some of the issues involved in knowledge level multi-agent collaboration. The central agent needs to perform a complex task that requires it to collaborate with other agents. To do so, it uses the Facilitator which identifies agents with the capabilities matching the requirements for performing a particular aspect of a task, and an Agent Name Server to determine the addresses of these agents. The inter-agent communication language called KQML is used to communicate with the Agent Name Server, Facilitator and other agents. The communication requires a shared representation and understanding of common domain concepts, i.e. a common ontology.

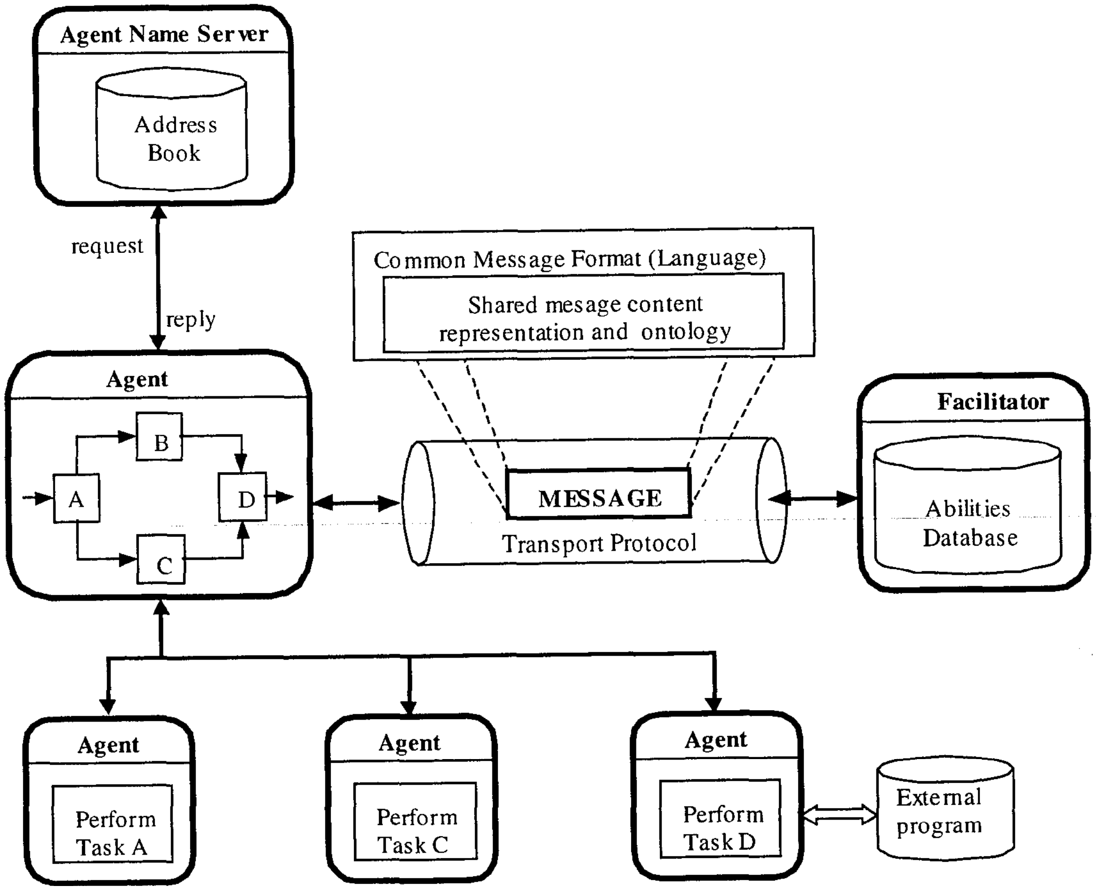


Figure 2. Society structure

Individual agent structure

As Figure 3 depicts, the generic ZEUS agent includes the following components:

- a Mailbox that handles communications between the agent and other agents.
- a Message Handler that processes incoming messages from the Mailbox, dispatching them to the relevant components of the agent.
- a Coordination Engine that makes decisions concerning the agent's goals, e.g. how they should be pursued, when to abandon them, etc. It is also responsible for coordinating the agent's interactions with other agents using its known coordination protocols and strategies, e.g. the various auction protocols or the contract net protocol.
- an Acquaintance Database that describes the agent's relationships with other agents in the society, and its beliefs about the capabilities of those agents. The Coordination Engine uses information contained in this database when making collaborative arrangements with other agents.
- a Planner and Scheduler that plans the agent's tasks based on decisions taken by the Coordination Engine and the resources and task specifications available to the agent.
- a Resource Database that maintains a list of resources (referred to in this paper as facts) that are owned by and available to the agent. The Resource Database also supports a

direct interface to external systems, which allows it to dynamically link to and utilize proprietary databases.

- an Ontology Database that stores the logical definition of each fact type — its legal attributes, the range of legal values for each attribute, any constraints between attribute values, and any relationships between the attributes of the fact and other facts.
- a Task/Plan Database that provides logical descriptions of planning operators (or tasks) known to the agent.
- an Execution Monitor that maintains the agent's internal clock, and starts, stops and monitors tasks that have been scheduled for execution or termination by the Planner/Scheduler. It also informs the Planner of successful and exceptional terminating conditions of the tasks it is monitoring. In order to manage tasks, the Execution Monitor also has a direct interface to external systems. It is assumed that the domain realizations of tasks are external programs.

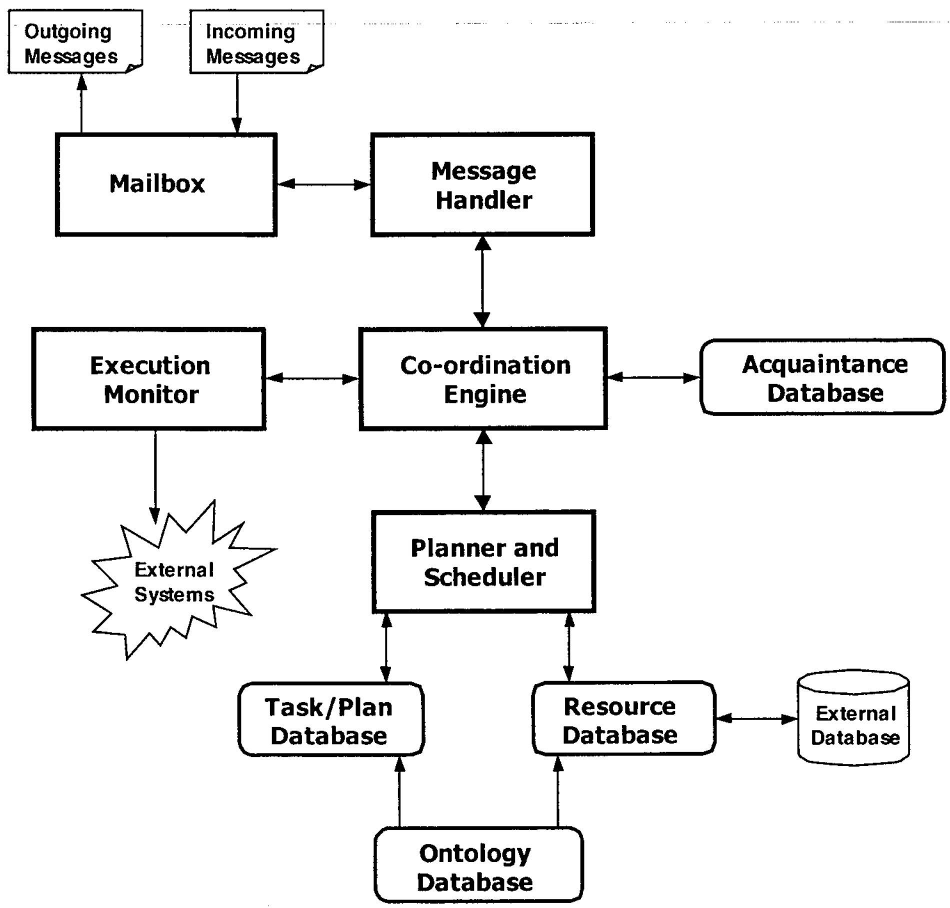


Figure 3. Individual agent structure

When a message from another agent is received, the agent's mailbox passes the message to the Message Handler for processing. On receipt of the message, the Message Handler interprets it as a request to achieve a goal. Hence, it forwards the message to the Coordination Engine to determine whether to achieve the goal and if so, to devise and coordinate an appropriate plan of action.

The Coordination Engine decides to attempt the goal, and invokes the Planner to construct a plan to achieve the goal. The Planner creates a plan for the goal, utilizing action descriptions from its Plan Database, and reserving the resources that are required by the plan and available in its Resource Database. However, the Planner finds that there are some other resources that are required by the plan, but are not available in its Resource Database, and which it cannot produce. Thus, it calls the Coordination Engine to seek external assistance in producing those resources.

The Coordination Engine then begins to attempt to contract out the task of providing the required resources at the required time. To do this, it checks its Acquaintance Database for the names of other agents that it believes can produce the required resources. Finding no acquaintance agents with the appropriate abilities, the Engine uses the Mailbox to send a message to a known facilitator, requesting a list of all "active" agents with the required abilities. On receipt of a reply from the facilitator, the Mailbox forwards the reply message to the Coordination Engine (through the Message Handler).

Now, given the list of agents with the needed abilities, the Coordination Engine first stores this information in its Acquaintance Database, and then proceeds to send messages to the agents, asking them to bid for a contract to produce the required resource. Again the outgoing messages are sent through the Mailbox and their replies returned to the Coordination Engine via the Mailbox and Message Handler.

Once all contractor agents have returned their bids for the tasks, or the reply deadline has expired, the Coordination Engine passes the returned bids to the Planner, which selects suitable contractors for providing the required resources. The suitability of each bid depends on factors such as its cost, and how well it fits in with the overall plan to achieve the original goal. With the bid selections made and the plan completed, the Planner returns to the Coordination Engine a list contractor agents to whom send contract award messages should be sent, and another list to whom the Engine should send bid rejection messages.

However, before sending out the contract award and bid rejection messages, the Coordination Engine first sends a message to the agent that originally asked it to achieve the goal, informing the agent that it can perform the goal and the cost of doing so. Next, the Engine waits for a response to its bid. If a favorable response is received, it then sends out the contract award and bid rejection messages to its own contractor agents and informs the Planner that the plan for the goal should be executed when appropriate. If, on the other hand, an unfavorable response was received, bid rejection messages are sent out to all contractor agents, and the Planner is told to cancel the plan.

Once a scheduled plan is ready for execution, the Execution Monitor executes the actions specified in the plan by invoking the external program declared in each action description. If the entire plan is successfully executed, the final results are sent through the Coordination Engine and Mailbox to the agent that requested the goal.

As can be seen from the use case scenario, the components of the Agent Component Library work together to provide the necessary agent-level functionality. For instance, the Mailbox and the Ontology Database facilitate communication. The former provides agents with the ability to send and receive messages in a 'standard' format, whilst the latter enables each agent to understand what other agents communicate to it. Once agents can communicate, we can raise the level of abstraction to the coordination level (or social interaction), wherein bargaining and negotiating is possible. This is realized via the Coordination Engine employing various defined coordination protocols. It is also clear from this example that co-operative problem solving between agents in task-oriented domains requires some planning and scheduling capabilities.

5. COORDINATION STRATEGY

We have implemented a version of Contract Net Protocol (CNP) as our coordination protocol. Figure 4 illustrates the relationship between protocols and interaction strategies. At each state the agent may need to make decisions about how to behave or respond to its current circumstances, these decisions are decided by strategies. This is best illustrated with an example.

Consider an agent that needs to obtain a resource because it can not produce it locally it must contact another agent to supply it. Hence the Initiator begins in the Initialization state by analyzing its pickup requirements and evaluating how much it is willing to pay for the resource and how quickly it needs it. Using the expertise encoded into its tendering strategy it formulates a CFP message containing its requirements, this is then broadcast to all potentially interested parties and the agent moves into the Negotiation state to await responses.

The arrival of a CFP message causes Respondents (i.e., truck agents) to move into its Initialization state. If a Respondent decides to reply (it is under no obligation to do so) it will move into its Negotiation state. The Respondent will now use its evaluation strategy to formulate a counter-proposal to the initial tender, which is then sent back to the Initiator in the form of a propose message. The Respondent then moves into a wait state for a finite period of time to await a response. If no response is received by the end of its time-out period the agent will terminate its part of the conversation.

When the Initiator receives a proposal it is analyzed using its own evaluation strategy. The evaluations can have one of the following three outcomes:

1. If the proposal is acceptable the conversation ends. The Initiator has not committed itself however, and will send a message at some point in the future either accepting or rejecting the offer.

2. If the proposal is not acceptable and the Initiator decides there is little point in negotiating further, it can end the conversation immediately.

3. If the proposal is not acceptable the Initiator can send a new, modified CFP message to the Respondent in question, whereupon it enters a wait state until a response arrives or its time-out period passes.

For Outcome 3, the Respondent is awoken by the arrival of a new CFP message, which is analyzed using its local evaluation strategy. This will cause the Respondent to do one of the following:

- It will decide not to bid again and end its side of the conversation.
- It will formulate a new proposal message, return it to the Initiator, and move into a wait state until a response arrives or it times out.

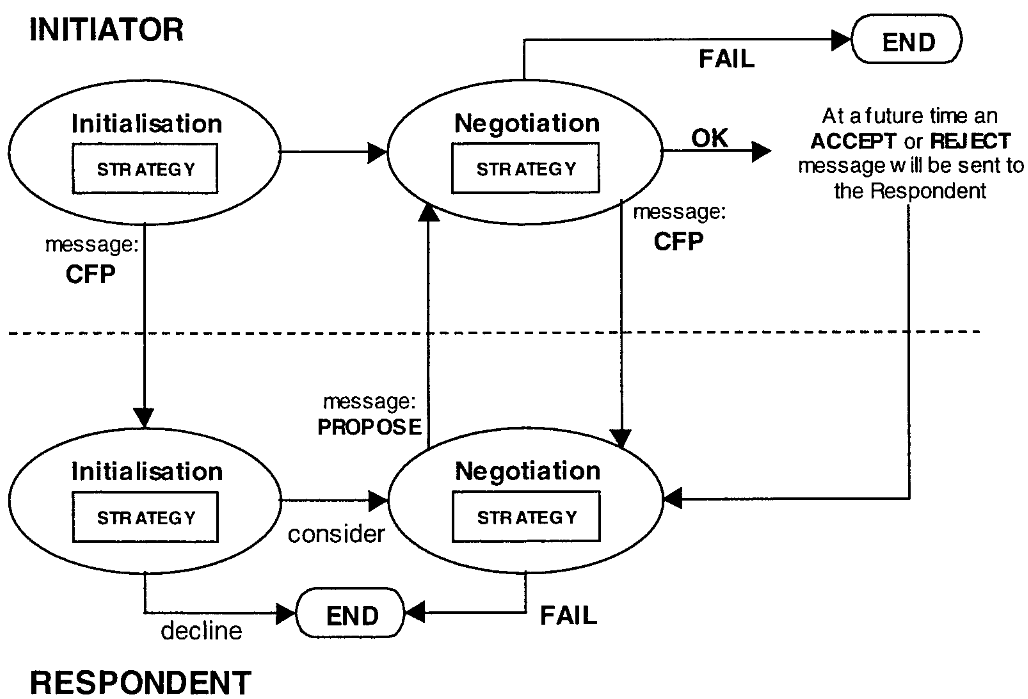


Figure 4. Illustration of the coordination protocol

In the specific context of package pickup operations, the overall strategy of a truck agent upon receiving a proposal from the dispatcher on a pickup request may be summarized as a flow chart shown in Figure 5. A pickup request is said to be in conflict if accepting this request will lead to the violations of pickup deadlines with other scheduled pickups. The algorithm to detect conflict is summarized in Figure 6.

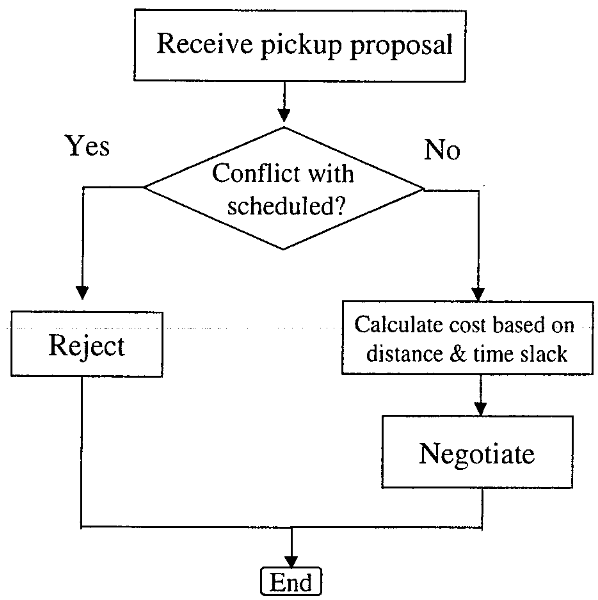


Figure 5. Flowchart summarizing the overall evaluation strategy of a truck agent

Here the top priority of each truck is to make all scheduled pickups in order to avoid severe penalties, or, alternatively maximize incentives, which in turn are based on the extent and quality of service. We did not explicitly consider penalties as part of the cost structure of agents in our scenarios. The truck will bid for any on-call job only after making sure that it does not violate any scheduled ones. Conflicts with scheduled jobs are avoided by making sure that the travel to the on-call destination and then to next scheduled destination will not violate the time deadline for the next scheduled job.

After ensuring against any conflict, the agent proceeds with cost calculations for tendering a counter-proposal. The cost is computed based on two major elements: (i) distance to the pickup destination and (ii) the available slack (for the day). When a truck receives a pickup job announcement, it first calculates time to reach the specified on-call pickup location from its current (about to perform) pickup location. The time will include both travel times as well as dwell times at intermediate stops between the current and the announced pickup location that have not yet been serviced by the truck. If the on-call pick location falls in the same loop, the job will be inserted in the job-queue for pickup based on its location in the loop and the truck's current location, which will determine whether the truck is better-off making this on-call pickup by persisting with the current counter-clockwise travel, or, if it should to make a special clockwise trip to make this pickup). Then the agent will calculate the expected end time which is an affine function of the available slack, i.e., larger the available slack, the smaller the value of expected end time. Then the distance and the expected end time are combined to yield the minimum cost (or the reserve price) for negotiation by assigning appropriate weights to each of the two elements. The weights are selected based on the agent's experience and requirement.

```
  {   start_time = Expected time at which the truck will
          complete the pickup of the current scheduled or
          committed job

      travel_time1 = Expected time taken to reach the on-call
            pickup location from the current (scheduled or
            committed) pickup location

          /*get to destination after making stops at all
            locations where pickups have not been made*/

      travel_time2 = Expected time taken to travel from the
            on-call pickup location to the next scheduled
            pickup location

     dwell_time = Expected time to travel off the loop to and
           from the current on-call pickup location, plus the
           loading time at this location

      total_time = travel_time1 + dwell_time + travel_time2 }

      Next_schedule_end_time  = Time before which the next
                    scheduled pickup needs to be made

     if  ( total_time > Next schedule_end_time - start_time)

           then, conflict and reject

     else  { calculate cost and negotiate}
  }
```

Figure 6. Conflict detection algorithm

Based on the counter-proposals, the dispatcher awards the job to the truck that makes the best offer, provided that the offer price is below the reserve price of the dispatcher. The price may be selected using appropriate dynamic pricing algorithms. However, we use a simple heuristic based on the overall slack available in the network to compute the reserve price of a pickup request. During negotiations, the truck agents and the dispatcher gradually will, respectively, decrement or increment their offers till a contract is successfully made, or the entire negotiation process terminates without a contract. We have developed a reinforcement learning algorithm that will help in gradually improving the computation of decrement or increment of offers. After, awarding the pickup, the individual truck is responsible for ensuring that the pickups are made within the specified deadline.

6. REINFORCEMENT LEARNING STRATEGY

Reinforcement Learning Modules

A reinforcement learning module underlies the overall coordination module of an agent as depicted in Figure 7. The following three key components comprise the reinforcement learning module:

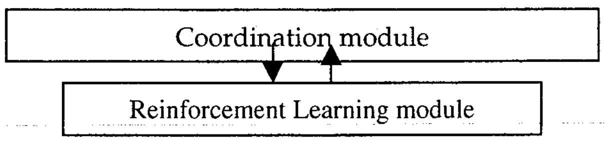


Figure 7. Relation between the coordination and the learning strategy modules

1. Reinforcement learner: Its responsibility includes conceiving the goal at the initialization state to start negotiations, receiving and extracting contract information from respondent agents for evaluation, formulating appropriate counterproposals using the experience contained in the Experience handler, and judging whether the goal is reached, based on which passing MESSAGE, OK or FAIL feedback to the coordination module.

2. Experience handler: Its responsibility includes maintaining designated history of recent deals, with attributes necessary for reinforcement learning, including cost rate, time rate, reinforcement, etc., which will be defined in the following two subsections.

3. Reinforcement handler: Its function is to evaluate how well the Reinforcement learner has performed in a deal in terms of its price and time considerations.

When an evaluation strategy is initiated, the Reinforcement learner component is invoked to determine the value of a counterproposal or decision to accept or reject the deal, invoking the information contained in the Experience handler. When a deal is closed, the Reinforcement handler is invoked to evaluate the effect of reinforcement, and the Experience handler is invoked again to store the deal records. In the following section, the details of our reinforcement methodology involving an interaction among these components are described. The negotiation initiator and the respondents may have parallel strategies, hence we focus only on the performance of the overall system relative to the initiator.

Reinforcement Learning Methodology

At the initialization state of each negotiation event, the agent invokes its Reinforcement learner. Next, a goal object containing the following parameters is passed to the Reinforcement learner:

- P_{min} : the minimum price that the agent is instructed to start with upon receiving respondent's counterproposal
- P_{max}: the maximum price that the agent can accept
- P_{NQ}: the maximum absolute price difference between the agent's and its respondent's prices, below which the agent will accept the offer
- t_{min}: the time at which the negotiation starts
- t_{max}: the final time before which the deal has to be closed

Then the agent will send out the first CFP as a MESSAGE. The negotiation between initiator and respondent proceeds along the lines of a non-zero sum game as described in Section 2 [9,10]. Upon receiving a counterproposal, the Reinforcement learner will start its analysis by initializing the following parameters:

- t: a system parameter identifying the current time
- P: the price that agent is currently willing to pay. Initially, $P = P_{min}$
- P_o: the requested price the respondent offered
- $r_p = (P - P_{min}) / (P_{max} - P_{min})$
  This parameter, called the cost rate, identifies how well the agent performed from a cost perspective on a particular deal
- $r_T = (t - t_{min}) / (t_{max} - t_{min})$
  This parameter, called the time rate, identifies how well the agent performed from a time perspective on a particular deal
- R: the reinforcement received for this deal. a detailed algorithm to compute R is presented in Section 3.3

Then, Reinforcement learner will move to the evaluation phase. The following are the different evaluation outcomes and decisions of an agent:

If $P_o < P + P_{NQ}$

=> the required price falls in the range of what the agent deems acceptable

- Reinforcement learner will firstly invoke the Reinforcement handler by passing r_P and r_T to evaluate its Reinforcement for the current deal
- Reinforcement learner will invoke the Experience handler to record the r_P, r_T and Reinforcement of this deal
- An "OK" information will be returned to the respondent

If $t \geq t_{max}$

=> a sudden death decision has to be made

If $P_o < P_{max}$

=> The price falls in the feasible range though not acceptable

- Reinforcement learner will firstly invoke the Reinforcement handler by passing r_P and r_T to determine its Reinforcement
- Reinforcement learner will invoke the Experience handler to record the r_P, r_T and R of this deal
- An "OK" information will be returned to the respondent

If Respondent has sent a counterproposal and $P_o > P + P_{NQ}$

=> proposal is not acceptable but a new request for proposal with updated price might facilitate an agreement

- Reinforcement learner will increase the current price with a calculated step for sending out the counterproposal

(Detailed step calculation algorithms are presented in the next subsection)

- A "MESSAGE" information will be returned to the respondent

Else

i.e., the agent never received, in the required time range, any proposal with an offer falling in the range of acceptable price range, hence the deal can not be made

➢ A "FAIL" information will be returned to the respondent

Computing reinforcement Increments

Price step Learning Algorithm

In this section the step learning algorithm is defined. Following parameters are extensively used in the content of the step learning algorithm:

- δt : Time step from last interaction: $\delta t = t^{(i)} - t^{(i-1)}$, initial value $t^{(0)} = t_{min}$
- δP : Price increment at the current time
- r_P^A : Average r_p of the previous rounds of transactions recorded by Experience handler
- r_T^A : Average r_T of the previous rounds of transactions recorded by Experience handler
- r_P^B : Value of r_p at the best Reinforcement level recorded by Experience handler
- r_T^B : Value of r_T at the best Reinforcement level recorded by Experience handler

When a counterproposal (offer price) from a respondent is not acceptable, the agent will compute the price increment □P for sending the next counterproposal $P^{(i)}$ as

$P^{(0)} = P_{min}$

$P^{(i)} = P^{(i-1)} + \delta P$

Two parameters must be determined before enforcing the learning increments:

(a) δP*_default*, the default value, is obtained from the goal object

(b) δP*_linear* is calculated to make sure that the agent is able to close the deal before t_{max} is reached:

$$\delta P\_linear = \frac{\delta t * (P_{\max} - P_{\min})}{(t_{\max} - t_{\min})}$$

1)

Next, the Reinforcement learner invokes the Experience handler to calculate r_P^A, r_T^A, r_P^B, r_T^B. In this case the reinforcement learning step calculation involves a semi-positive approach: i.e., the average value of the cost and time ratios under the best and the average reinforcement conditions are used as

$$\delta P\_learned = \frac{((r_P^B + r_P^A)/2) * (P_{\max} - P_{\min}) * \delta t}{((r_T^B + r_T^A)/2) * (t_{\max} - t_{\min})}$$

2)

Thus, an increment of *δP_learned* to the current *P* yields a value lying within the specified price range that should be accepted, considering the elapsed time.

Reinforcement Shaping Algorithm

The major objectives of this reinforcement learning function are to: 1) reach best achievable price 2) use shortest time, to reach an agreement. The task of an initiator agent is to use the least amount of time to achieve a lowest price for procurement. On the respondent side it will be expressed as use least time to achieve a highest price. In order to simultaneously address the influence of time and cost factors on the overall evaluation result, we specify R as follows:

$$R = C_w*(1-r_p)+T_w*(1-r_T)$$

where C_w and T_w are constant weights. Thus the initiator maximizes R by minimizing r_P and r_T.

7. IMPLEMENTATION DETAILS

Implementation of evaluation strategy

We implemented a hypothetical goods pickup scenario, similar to what one finds in a company like UPS. Our scenario consisted of two adjoining loops where trucks move as shown in Figure 8, and its visual representation as part of the implemented

Zeus environment is shown in Figure 9. There are eight possible pickup locations in each loop, namely locations 1-8 for Loop 1, and locations 11-18 for Loop 2. Travel times between any two adjacent locations were assumed to be uniformly distributed between 2.9-3.1 time units. Dwell time was assumed to be the same at all pick locations and equal to 1 time unit. We randomly generated six scenarios consisting of different on-call pickup requests occurring at random grid locations on the two the loops, under different levels of available slack.

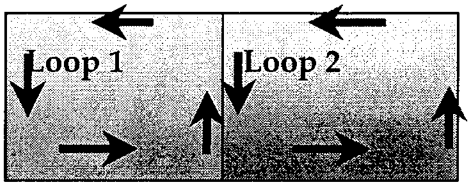


Figure 8. Schematic of the basic scenario

Table 1 summarizes the particulars of the generated six scenarios. Here, the scheduled pickup locations for a truck are denoted by a binary string of 1's and 0's, where the leftmost bit refers to location 1 for Loop 1 and location 11 for Loop 2, respectively. A zero at a particular position on the string denotes that the particular position is a scheduled pickup location. For example, for the string 1 0 1 1 0 ..., corresponding to a truck assigned to Loop 1, pickups are scheduled at locations 1, 3 & 4, and no pickups are scheduled at locations 2 & 5, and so on.

Table 1. Summary of simulation experiments to study evaluation strategy

Scenarios	Scheduled pickup locations		On-call pickup locations and times	
	Loop 1 (Locations 1-8)	**Loop2** (Locations 11-18)	**Loop 1**	**Loop2**
1 Light load Loop Slack = 5 time units	1 0 1 0 1 1 0 1 1	1 1 0 1 1 0 0 1	a. Location 7 @ time 12 b. Location 1 @ time 23	a. Location 17 @ time 12 b. Location 16 @ time 27
2 Heavy load Loop Slack = 5 time units	1 1 1 1 1 1 0 1	1 1 0 1 1 1 1 1	a. Location 7 @ time 12 b. Location 1 @ time 23	a. Location 17 @ time 12 b. Location 16 @ time 27
3 Light load Loop Slack = 10 time units	1 0 1 0 1 1 0 1 1	1 1 0 1 1 0 0 1	a. Location 7 @ time 12 b. Location 4 @ time 17 c.Location 1 @ time 23	a. Location 17 @ time 12 b. Location 16 @ time 27
4 Heavy load Loop Slack = 10 time units	1 1 1 1 1 1 0 1	1 1 0 1 1 1 1 1	a. Location 7 @ time 12 b. Location 4 @ time 17 c.Location 1 @ time 23	a. Location 17 @ time 12 b. Location 16 @ time 27
5 Light load Loop Slack = 10 time units	1 0 1 0 1 1 0 1 1	1 1 0 1 1 0 0 1	a. Location 7 @ time 12 b. Location 4 @ time 17 c.Location 1 @ time 23 d.Location 3 @ time 31	a. Location 15 @ time 12 b. Location 18 @ time 27 c. Location 12 @ time 33 d. Location 15 @ time 39
6 Heavy load Loop Slack = 10 time units	1 1 1 1 1 1 0 1	1 1 0 1 1 1 1 1	a. Location 7 @ time 12 b. Location 4 @ time 17 c.Location 1 @ time 23 d.Location 3 @ time 31	a. Location 15 @ time 12 b. Location 18 @ time 27 c. Location 12 @ time 33 d. Location 15 @ time 39

Scenarios 1, 3 & 5 have relatively few locations (about 62.5% of the possible pickup locations) scheduled for pickup, and scenarios 2,4 & 6 have relatively large fraction (~90%) of possible locations scheduled. The ratio of the number of on-call pickup requests to the number of scheduled pickup requests is low (~25%) for scenarios 1&2, and medium (~35%) for scenarios 3&4, and high (~50%) for scenarios 5&6. For each generated scenario, we first simulated the "as-is" case where no negotiation occurs, and each truck is allowed to make only the pickups occurring in its designated loop. Next we simulated the case where truck agents negotiate with the dispatching agent to allocate trucks for on-call pickups, and trucks are free to make pickups in any loop.

The ontology based on which the agents negotiate for allocating trucks for specific on-call pickup requests is divided into two categories: abstract and entity. The abstract ontology contains ontology that agents use during negotiation process, such as money. The entity ontology is the domain-specific ontology items that agents negotiate over. It defines the attributes of each pickup jobs. The attributes include the location of pickup work, and the time deadline before which the pickup must be made. A screen-shot showing the visualization of this ontology-base is shown in Figure 10. Visual screen captures of various stages of the negotiation strategy are shown in Figure 11.

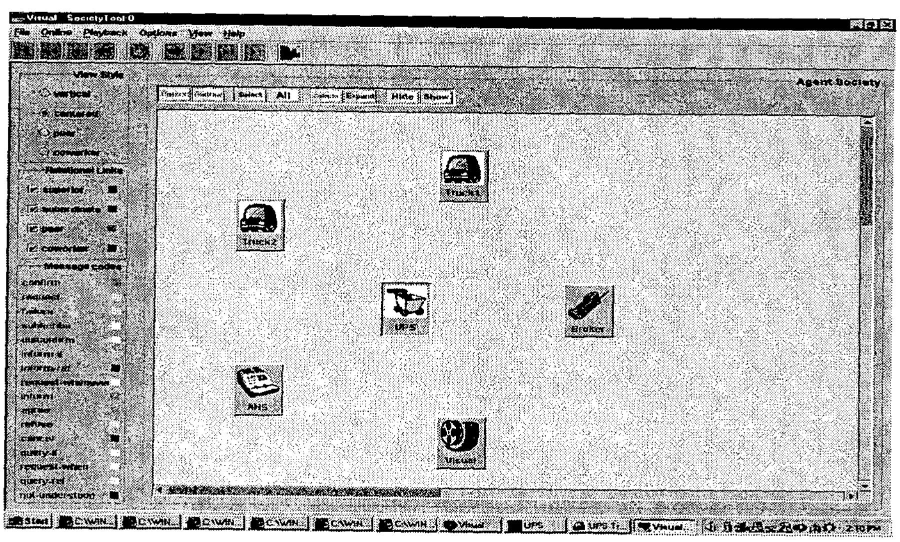


Figure 9. Visualization of the implemented scenario within Zeus environment

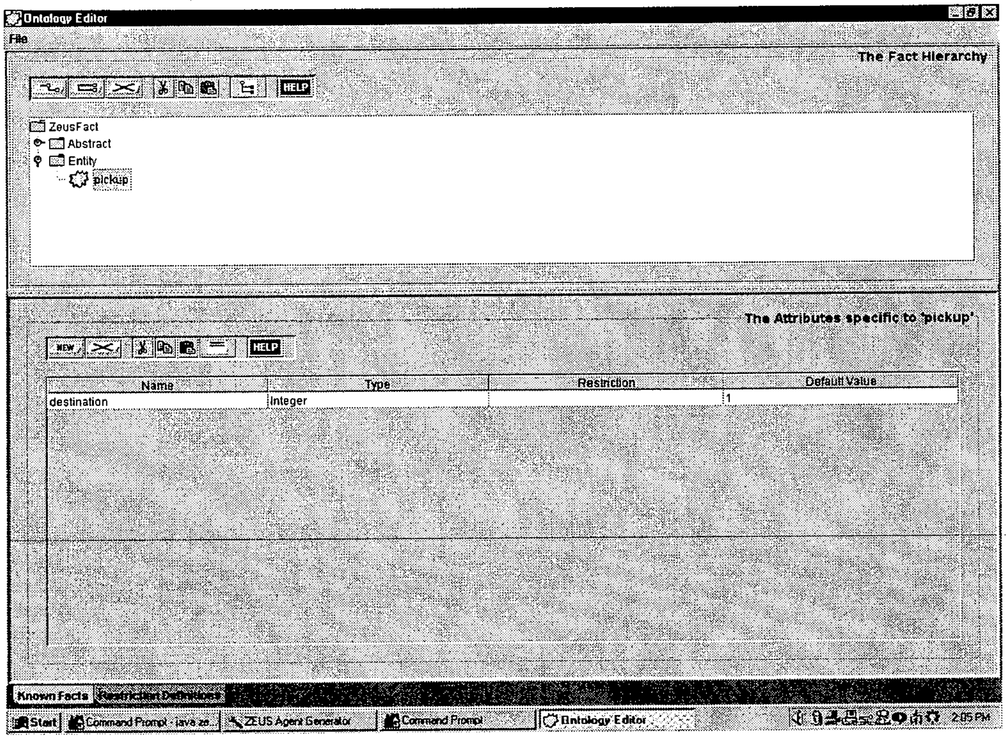


Figure 10. Visualization of the implemented ontology

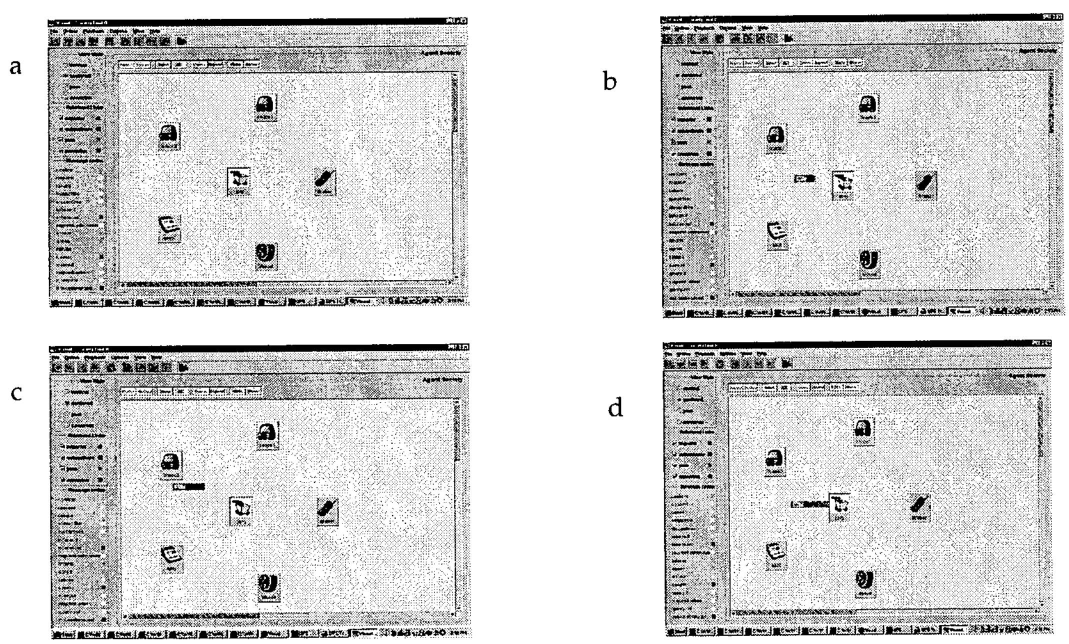


Figure 11. Visualization of the various stages of negotiation (a) initialization stage (b) proposal stage (c) counter-proposal stage (d) awarding stage

Implementation of learning strategy

We conducted extensive testing of the learning strategy in the foregoing simulation environment. Since the initiator and the respondent strategy were parallel, our analysis focused on the initiator. Through this, we measured the group behavior of the overall environment. Reinforcement learning required multiple rounds of simulations to arrive at a stable state. But due to some difficulties associated with executing the large-scale simulations, we conducted 100 simulation runs at each strategy. This means that the initiator has accumulated experience based on 100 deals it has made. We tracked the values of r_p, r_T and R for each deal. Following cases were selected to study the effects of reinforcement learning:

1. No reinforcement: The initiator did not use any reinforcement. Instead it only used linear evaluator strategy.
2. Reinforcement vs. Non-Reinforcement with cost weight (C_w) =2 and time weight (T_w) =1: Only the initiator was equipped with a reinforcement learning strategy. All other agents were equipped with linear evaluator strategy.
3. Reinforcement vs. Non-Reinforcement with cost weight (C_w) =2.5 and time weight (T_w) =0.5: Only the initiator was equipped with reinforcement learning strategy. All other agents were equipped with linear evaluator strategy.
4. Reinforcement vs. Reinforcement with cost weight (C_w) =2.5 and time weight (T_w) =0.5: All agents were equipped with reinforcement learning strategy. So the initiator and respondents would use their reinforcement strategies to compete with each other.

8. Results

Results of evaluation strategy

From the simulation results (see Table 1) we found that dynamic allocation of trucks, as opposed to a pre-assigned dispatching lead to a significant improvement of overall efficiency of pickup operations. This translates to more incentives for truck agents (perhaps truck drivers), and more income for the dispatcher from being able to commit to more on-call jobs. Evidently, as we started to increase the available slack, the trucks were able to make more on-call pickups irrespective of the distribution of slack on each loop.

Table 2. Summary of results of evaluation strategy simulations

Scenario	No Negotiation				With Negotiation			
	# of on call made		Total Slack Remaining		# of on call made		Total Slack Remaining	
	Loop 1	Loop 2	Loop 1	Loop 2	Loop 1	Loop 2	Loop 1	Loop 2
1	0/2	1/2	5/5	5/5	1/2	2/2	5/5	5/5
2	0/2	0/2	5/5	5/5	1/2	1/2	5/5	5/5
3	0/3	1/2	10/10	10/10	1/2	2/2	10/10	6/10
4	0/3	0/2	10/10	10/10	1/2	2/2	10/10	6/10
5	0/4	1/4	10/10	10/10	1/2	3/4	10/10	6/10
6	0/4	0/4	10/10	10/10	1/2	2/4	10/10	6/10

Results of learning strategy

The simulation results of our implmentation of our reinforcement strategy are presented in the Figures 12-15, and the results after 100 completed transactions are summarized in Table 2.

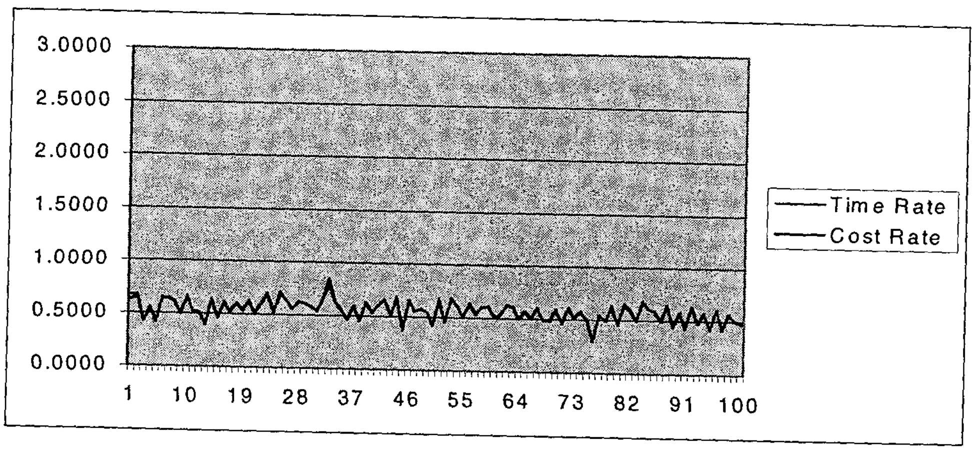


Figure 12. No reinforcement for all agents

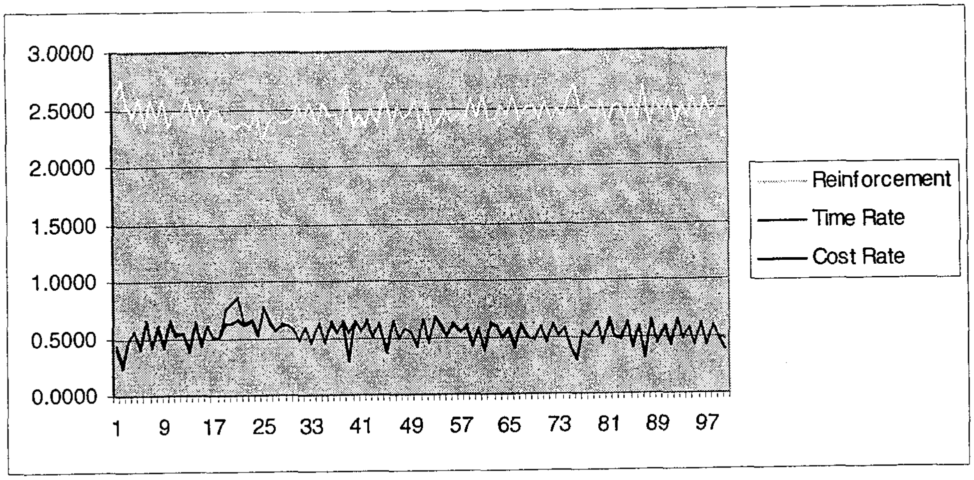
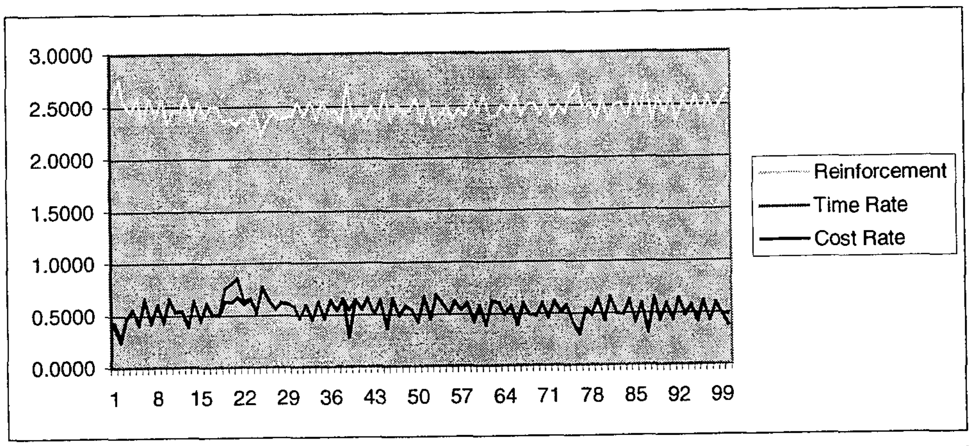


Figure 13. Reinforcement for the intiator with C_w =2 and T_w =1, and no reinforcement for all respondents



Fiture 14. Reinforcement for the intiator with C_w=2.5 and T_w =0.5, and no reinforcement for all respondents

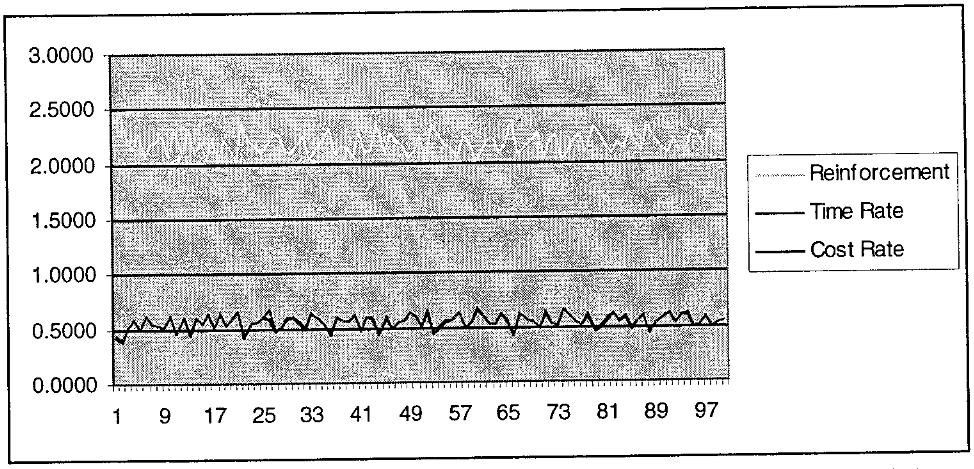


Figure 15. Reinforcement for all agents with C_w =2.5 and T_w =0.5

Table2. Summary of learning strategy implementation results with all agents equipped with learning, and C_w = 2.5, T_w, = 0.5

Cost Rate	Time Rate	Reinforcement
0.5584	0.0066	None
0.5339	0.0138	1.9190
0.5330	0.0069	1.6616
0.5500	0.0062	1.6219

In our simulations, t_{max} was set to an arbitrarily large value. Therefore r_T was very small and all deals were generally closed long before the deadline. As one may notice, r_C, r_T and R hover around 0.54, an insignificant value, and 1.7, respectively. Furthermore, the values of r_C, r_T and R are all positive, which implies a postive effect of reinforcement learning.

Judging from the simulation results (Figures 12-15), we can see that in linear strategy result, r_p was oscillating from the very beginning. There was no significant time period where the final results seemed to converge to a certain value. For cases with reinforcement learning, there was a oscillatory phase at the beginning and then R values were close to a static equilibrium. This equilibrium slightly shifts (increases) as the negotiation proceeds. But this trend, however, was not significant. More simulations are required to demonstrate a substantial improvement in the agent's performance through the use of our simple reinforcement learning strategy.

Also, we found that the R values exhibited a tendency to converge faster to a near-equilibrium as we increase C_w. When we increased C_w and decreased T_w, we found that r_T was anomalously high for some cases, although the average values remained the same. This could have resulted because of anomalously high completion times in these cases. Compared to results from cases with no reinforcement, those that used reinforcement learning consistently yielded lower r_T and higher R. When both the initiator and the respondents were equipped with reinforcement learning, however, R values for the intiator seemed to oscillate around a lower value compared to the value when only the initator was equipped with reinforcement learning. From the foregoing results we can see that, in a group environment, if an agent does not use a proactive learning strategy while some others do, the agent's performance tends to be adversely affected .

9. SUMMARY

In this report, we have investigated the effects of: (1) multi-agent negotiation, initiated based on real-time information, as well as (2) a simple reinforcement learning strategy in the context of improving coordination of agents in a goods pickup scenario. The agents used in our context are behavior based, and each behavior is triggered by a certain kind of conditions. Our future extensions will include clustering of jobs in agent coordination protocol. The result of binding several related jobs together as one cluster may be different from negotiating on each individual job separately and binding their results together. Further research in agent learning scenarios is currently being investigated to extend our understanding of coordination methodologies and multi-agent systems in a transit coordination environment.

10. IMPLEMENTATION

Our simulation studies indicate that real-time coordination, perhaps using a multi-agent systems approach, has the potential to improve the efficiency of goods pickup operations. In fact, currently only one out of every three on-call pickup requests is honored, although tremendous amount of slack is built into the schedules of truck operators. The coordination methodology developed in this research can improve the number of on-call pickups that can be made. However, further advances in research as well as business practices are necessary in order to render real-time distributed coordination commercially viable for trucking and goods movement industries. Specific future advancements should include:

(a) *Development of algorithms* that can improve upon our algorithms for bidding, cost computations, and bid evaluation, which underlie our negotiation approach. Also, algorithms for near-optimal insertion of on-call pickup requests can also be incorporated as part of the evaluation strategy. The negotiations can be significantly improved through job-clustering, where we can allocate multiple on-call pick requests whose locations are close to each other to a single truck in order to save time for negotiations. The simple algorithms used in this research for facilitating reinforcement learning can be replaced by neural-network-based learning methods, which can facilitate fast computations and real-time information retrieval.

(b) *More widespread use of two-way communication devices*, so that evaluations from individual truck agents can be quickly communicated among other agents and the dispatcher, thus leading to a dynamic negotiation process to coordinate task allocation.

(c) *Changes in business policies* that can take advantage of the dynamic nature of negotiations. For example, a company like UPS can allow trucks to make pickups over multiple regions, and also create an incentive-based system for truck drivers so as to maximize the service on on-call pickup requests.

Although most of the research results have been explain in reference to the operation of a goods delivery company like UPS, most of the results are applicable to a range of application domains. For example, a few trucking companies can take advantage of automated negotiations to contract out jobs to company's and/or independent truck drivers. With a few minor modifications, the computational framework including our evaluation strategy and reinforcement learning methods can be adapted to these scenarios. A similar solution framework can be designed for dispatching Caltrans trucks for road-side assistance.

BIBLIOGRAPHY

Basar, T., and G.J. Olsder, *Dynamic Noncooperative Game Theory*, Academic Press, 1995.

Benerecetti, M., Giunchiglia, F., and Serafini, L., "Model checking in multi-agent systems," *Technical Report IRST-9708-07*, Isituto per la Ricera Scientifica e Technologica, 1997.

Povo (Trento), Italy, http://mambo.cs.unitn.it/bene/pub.html

BT Labs-Intelligent Systems Research Group, *Zeus realization guide*, http://www.labs.bt.com/projects/agents.htm,1999

BT Labs-Intelligent Systems Research Group, *Zeus technical guide*, http://www.labs.bt.com/projects/agents.htm,1999

BT Labs-Intelligent Systems Research Group, *Zeus role model guide*, http://www.labs.bt.com/projects/agents.htm, 1999

Beck, J.C., "KQML users guide," 1993

Colman, A.M., *Game Theory & its Applications*, 1992

Burmeister, B., Haddadi, A., and G., M, "Application of multi-agent systems in traffic and transportation," *IEEE Transactions on software Engineering*, 144(1): 51-60, 1997.

Burmeister, B., and K. Sundermeyer, "Application of multi-agent systems in traffic and transportation," *IEE Transactions on Software Engineering*,144(1):51-60, 1992.

Fischer, K., Muller, J. P., and Pischel, M., "Cooperative transportation scheduling: An application domain for DAI," *Applied Artificial Intelligence*, 10(1):1-34 1996.

Hall, R.W., "Vehicle Scheduling at a Transportation Terminal with Random Delay en Route," *Transportation Science* 19, 308-320, 1985.

Jensen, F., *An Introduction to Belief Networks*, Springer Verlag (UCL Press), 1996.

Kanaris, A., P. Ioannou, and F.S. Ho, "Spacing and capacity evaluation for different AHS concepts," *In Proceedings of American Control Conference*, Albuquerque, NM, 1997.

Knoppers, P. and T. Muller, "Optimized Transfer Opportunities in Public Transport," *Transportation Science* 29, 101-105, 1995.

Ljunberg, M. and Lucas, A., "The OASIS tra_c management system," *In Proceedings of the Second Pacific Rim International Conference on AI* (PRICAI-92), Seoul, Korea, 1992.

Newell,G. F., "Control of Pairing Vehicles on a Public Transportation Route, Two Vehicles, One Control Point," *Transportation Science* 9, 248-264 (1974).

Newell, A.A. ,The Knowledge Level, *Artificial Intelligence* 18, 87-127, 1982

Regan, A.C., and T.F. Golob, "Freight operators' perceptions of congestion problems and the application of advanced technologies: Results from 1998 survey of 1200 companies operating in California," *Transportation Journal*, 57-67, Spring 1999.

Sandholm,T. W. and Lesser, V.R., "Advantages of a leveled commitment contracting Protocol," http://www.cs.wustl.edu/~mas/

Sastry, S., Meyer, G., Tomlin, C., Lygeros, J., Godbole, D., and Pappas, G., "Hybrid control in air traffic management systems. *In Proceedings of the 1995 IEEE Conference in Decision Theory and Control*, pp. 1478-1483, New Orleans, LA, 1995.

Satapathy, G., and Kumara,S.R.T., "Object oriented design based agent modeling," *Proceedings of PAAM99Conference*, 1999

Wellman, M.P., "Market-Oriented Programming Environment, and its Applications to Multicommodity Flow Problem," *Journal of AI Research*, 1:1-22, 1993.

SIMULATION OF ON-CALL-AIR PICKUP DISPATCH PROCESS AND FAILURE ANALYSIS AT UPS

ABSTRACT

UPS being the world's largest package distribution company, streamlining of all its processes and operations is of paramount importance. Keeping in mind that UPS is a worldwide corporation, any deviation from the optimal method results in copious amounts of financial losses for the company as a whole. Corporate clients have established accounts with UPS and these clients ship out packages on a regular basis. Picking up these packages constitutes a major portion of the latter half of the driver's workday. In addition to these scheduled pickups are the pickup requests that originate based on individual needs. The current method of assignment of these pickups to the drivers is in question, and a potentially improved system is suggested. The feasibility of this proposed system is studied through a simulation analysis and the findings of the study have been presented.

INTRODUCTION

United Parcel Service is the nations largest package transportation service. For the time being at least, direct shipment is still the highest source of revenue for UPS. The company currently has a regular client base in addition to the individuals who also employ their services. Deliveries vary with individual customer needs and are predominantly seasonal. Peak season is experienced towards the end of the year, building up to the holiday season. Pickups however are fairly consistent, and the drivers (known as service providers) make these pickups as a regular part of their route. The route traced by a driver is determined by the type of area covered (commercial or residential, or both). Optimal routes are developed keeping in mind factors like one ways and typical traffic conditions, and the like. Time standards are developed for every aspect of the driver route, by routine meticulous time studies and are specific to the coverage area.

For enhanced administrative ease, every city that UPS delivers to is fragmented into sections known as *loops*. These loops are further subdivided into areas known as *units*. It is typical to have as many as five or six drivers in any one loop. For the remainder of the report, these units will be referred to as coverage areas. Typically, there is only one driver within one coverage area. This assumption is maintained throughout the analysis.

Package types have different priorities assigned to them. The Next Day Air packages have the highest priority followed by the other package types. Ground deliveries in residential areas have the lowest priority of all. Next Day Air packages are guaranteed by 10:30 am the day after the package is received. Another package type is the Next Day Air Saver. These packages are guaranteed to be delivered by 12:00 pm the day after the package has been received by UPS. The priority on this package type is high as well. Consequently, the drivers spend the entire first half of every day making these commits and have little or no time to make deliveries of any other package type. All or most pickups are made in the afternoon, after all the high priority packages have been delivered. The afternoon route is pretty much similar to the morning route. The drivers again follow trace to their regular pickup stops, while at the same time make any deliveries (mostly low priority packages) that were not taken care of in the morning route.

The drivers are each given a *DIAD* (Delivery Information Acquisition Device) that maintains a log of the driver's delivery and pickup stop time and the locations of each of these stops. These DIADS can receive dispatch information through the wireless communication system currently existent, much like a paging service, with marginal time delays. This device however, is not a real time two-way communication device. Communication between the DIAD and the UPS computer system takes place when the driver replaces the DIAD in the DIAD consol holder located in the dashboard of the package car. Only critical information associated with the high priority packages is communicated back to the center, like made/failed delivery (of high priority packages) and the name of the package recipient. All other information is uploaded at the end of the day when the divers return to the center, where the DCS (DIAD Control System) aids in uploading all the delivery and pickup information.

CURRENT OPERATIONAL SYSTEM AND PROBLEM STATEMENT

Every "center" is assigned a section of a city that is it's designated coverage area. The center management team comprises a Center Manager and typically three On-Road Supervisors (as is the case with the Rodeo Center). The On Road Supervisors are responsible for determining everything from staffing levels to constantly maintaining contact (through the DIADs) with the drivers while they are on-road. It is also their responsibility to respond to requests for unscheduled pickups for certain air packages, that may arise while the drivers are already on their routes. These are referred to as on-call air pickups.

Upon receiving a request for service, the on-road supervisors relay the pertinent information to the drivers. This information includes the address of the request, the package type, number of packages, the estimated time in which the package will be ready and also the closing time of the establishment. All this information is dispatched to the driver assigned to the unit in which the request arises. The driver than has a few minutes to respond to the dispatch. The two alternatives available to the driver are to either accept the dispatch, or the driver can also "kick-back" the request whereupon the management team will dispatch the pickup information to another driver. The reasons for a kick-back may include anything from insufficient capacity in his/her package car, to personal reasons like health or the need to finish the route quickly.

This method is not necessarily the best system. The assignments are made based solely on the driver's coverage area and not necessarily on the closest driver to the point of origin of the request for service. Additionally, the center management team needs to know the approximate whereabouts of the drivers at all times of the day. This is a lot of information for an individual to have a high level of fidelity in reproducing on a regular basis. The containment of this amount of knowledge with one individual often hampers smooth functioning of the center, creating a knowledge gap too huge to surmount in some cases.

UPS, in an effort to streamline its operations is always trying to reduce driver on-road miles and consequently, the amount of time the drivers spend on the road (known as windshield hours). There is a definite cost associated with every mile the drivers traverse. Inefficient dispatching could result in hundreds of additional miles a week. This figure extrapolated corporate wide results, literally in millions of dollars.

It remains within the scope of this study to determine if the dispatching process currently being employed is indeed the most efficient, and if not, to lay the groundwork for the development of an improved dispatching system for the on-call air packages.

APPROACH TO THE ANALYSIS

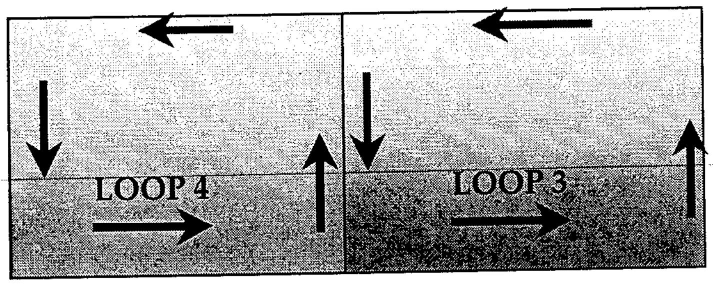
RELATIVE DISTANCES

The information that is uploaded into the UPS computer system every evening about the driver delivery and pickup stops is saved into a database. This information if called GD2 data. GD2 contains the details of every stop made by every driver. This information covers both delivery and pickup information and also contains the details for all package types. Probably the most important aspect of this information is the fact that these data can be retrieved by a program called Driver Mapping, which can actually plot the driver's route, in the sequence of his/her stops, providing a great level of detail at each stop. The driver trace is plotted over a map of the coverage area, and this proves a vital asset in understanding what the driver does on road without actually having to go along with the driver on his route. But, probably the greatest gains to be had from using the program comes from the fact that the program has a very useful tool called the ruler. This tool allows one to measure the distance (directly in miles) between any two points on the map.

DRIVER ROUTE SIMULATION AND PICKUP ASSIGNMENTS

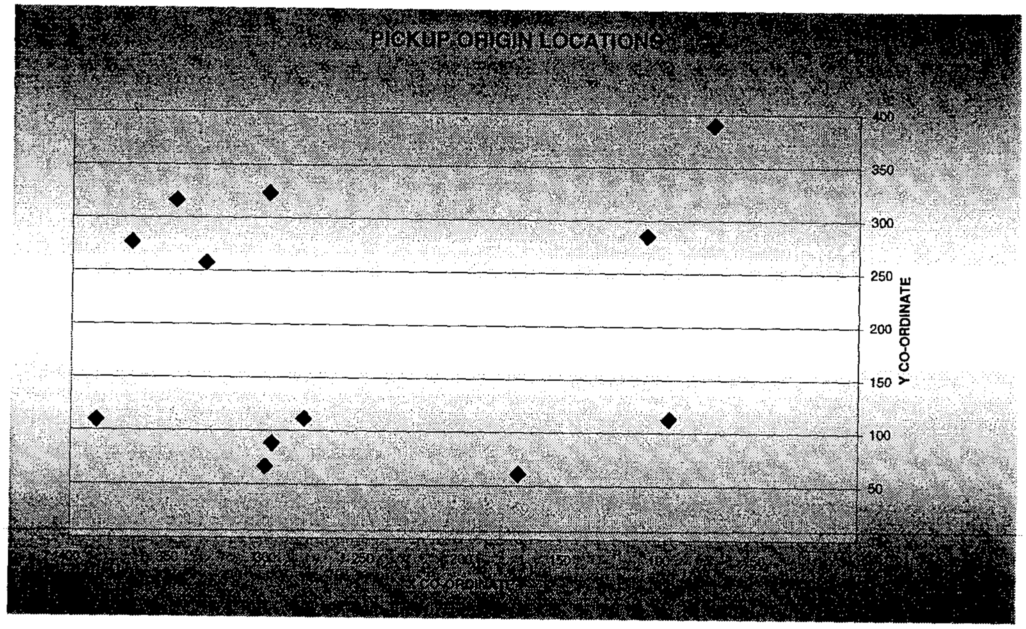
AWESIM®, developed by the Pritsker Corporation, is a very powerful simulation software. An attempt is made here to simulate the driver's daily trace and also to simulate the assignment of packages based on the existing system and also a proposed new system. The new system would not rely on any one person having to know the location of the driver within a unit. Additionally, the simulation incorporates the possibility of the driver kicking back the request for service, and the dispatch being reassigned to another driver. The redirected dispatch is based on the concept of closest driver to the origin point of the pickup request.

Before an in-depth description of the model is made, an understanding of the driver trace is required.



LOOP 4	DRIVER 2
LOOP 3	DRIVER 1

Typically, every driver starts his/her route at the base of the loop and works around the loop, eventually returning to the point of commencement. Pickups mainly being done in the afternoon, the drivers often retrace the original morning route with the appropriate changes being made to the route. In many cases, different drivers will actually take different sides of the same streets. This may involve traversing great distances just to make those pickups, just because it falls within the designated coverage area.



The chart shows the plot of the origin locations. By default, driver 2 would be assigned all stops that occur on the left half of the plot (loop 4) and the remaining would go to driver 1(loop 3). These figures were generated in a random fashion, mimicking the actual fashion of origin of pickup requests.

REALITY MODELED

For the purpose of the simulation, and to avoid undue initial complications, two adjacent units from units 3 and 4 were chosen. These units were of almost the same size and were modeled as equal in most aspects. However, the difference in the units lies in the time between stops and also the actual time for the stop itself.

UPS, through the years has developed allowances (time standards) for every aspect of a drivers work day, including the two aspects that most concern the simulation, namely the time allotted for a stop and the time it takes the driver to reach the coverage area itself. Another important aspect of the simulation times, namely the time between stops, was obtained by analyzing the GD2 data. The time between stops was laboriously tabulated and analyzed. The frequency and bin size were determined, and the time between stops was computed. The analysis showed this factor to vary as an exponential distribution. This result was consistent with the nature of the exponential distribution, which is typically used to model time between events. The analysis was done for both units and the times were inputted into the simulation.

MODEL DESCRIPTION

The model is built to represent two loops and the drivers making their routes within them. The time standards developed by UPS and the information gathered from GD2 data were used in the simulation. The simulation was run for a period mimicking the actual routes. The time at stops and the time between stops were also incorporated into the model.

The drivers, as they traced their routes stop to make their regular pickups. The on call air pickups are simulated by random occurrences of events that interrupt the model and divert the entities (representing the drivers), to the assignment portion of the model. The assignments are made, as mentioned earlier, on the basis of the closest driver to the point of origin of the request for service. The instantaneous distances are compared and the closer driver is assigned the pickup. Also, before the pickup is assigned, the capacity constraints are examined. These constraints are simulated by generating a random number that represents situations wherein the driver accepts the pickup or if there is a kick back situation. As long as the capacity of the driver who is closer to the point of origin of the call has not been exceeded, that driver is assigned the pickup. In the event that the first driver's capacity has been exceeded, the load of the adjacent driver is checked, and as long as this capacity is not exceeded as well, the pickup is "bumped" from the first driver to the second (or vice-versa). The other condition under which a pickup can be bumped from one driver to another is, as mentioned earlier, when a driver kickback occurs.

The third situation that could arise is that both drivers capacities have been met or exceeded, in which case neither driver would be assigned the pickup. The event would simply be marked as a missed pickup, and the entity is then rerouted to its point of branching.

With the existing system of assigning packages at UPS, there is no structured form of re-negotiating the assignments. Capacity constraints, though not always a big player in the equation, are really not considered in the assigning process. Assignments are made on the basis of calculated guesses, at best.

SIMULATION RESULTS AND INTERPRETATION

For the purpose of simulation and analysis, capacities for drivers 1 and 2 were arbitrarily assigned. Driver 1 was assigned a capacity of three additional stops (over and above the regular pickups he/she makes) and driver 2 was assigned a capacity of five. When an assignment was to be made to either of these drivers, these capacity figures were checked and barring excess, the assignments were made.

AWESIM generates a report of the activities of the simulation. The interpretation of the results is of paramount importance. The number of entities generated (pickup requests) varies depending on the position of the drivers within the loop. The travel times being exponentially distributed, the work-day (number of hours worked) will vary.

To maximize fidelity in the results, the simulation was run ten times. The results, in all cases were consistent with the constraints incorporated into the model.

LABEL	MEAN	STD	NUM. OF	MINIMUM	MAXIMUM
	VALUE	DEVIATION	OBSERVATONS	VALUE	VALUE
TIME	138.465	57.483	110	32.072	234.301
TIME2	119.290	53.949	53	27.418	225.125
STOPS_D1	3.000	0.000	1	3.000	3.000
STPS_BUMPED_2 D2	2.000	1.000	3	1.000	3.000
STOPS_D2	1.500	0.707	2	1.000	2.000
STPS_BUMPED_2 D1	1.500	0.707	2	1.000	2.000
FAILED PICKUPS	3.000	1.581	5	1.000	5.000

The final number of packages assigned to each driver, in all the cases does not exceed the load capacity of the drivers. The total number of packages finally assigned to each driver is the sum total of the packages directly assigned by virtue of driver proximity, and those bumped to the driver because of either capacity or personal priority constraint violation. The inference that can be drawn from the entire simulation is that the current process of assignment being followed at UPS leaves a lot to be desired, in terms of process optimization. Excessive driver miles are driving up operational costs, which in turn drive up the service costs incurred by the customer, and consequently provide openings in the package transportation market to competitors.

To further explicate the aforementioned statements, for instance, referring to the table or results below: (Results reproduced are from run 1 of 10)

The AWESIM summary of results shows the time of occurrences of events, the stops assigned to each driver and also the stops that have been 'bumped' from one driver to the other. The bumped stop count is represented by the column STPS_BUMPED_2D2 and STPS_BUMPED_2D1 which are to be read as 'stops bumped to driver 2' and 'stops bumped to driver 1' respectively. These figures are the crux of the entire simulation. What these numbers represent (STPS_BUMPED_2D1 FOR INSTANCE) are the stops that were originally to be assigned to driver 2, but because of capacity constraint violation, have been reassigned to driver 1. This number represents the number of stops that would have surely resulted in pickup service failures, had it not been for the re-allocation process.

Another interesting case recorded by the simulation is as follows: This particular example is from rum 8 of 10. The results are tabulated in the table below. The interesting part about the results in this case is the assignment of stops to driver 1. As the table shows, in none of the cases was driver 1 closer to the point of origin of the pickup location. In spite of this, there were 3 stops bumped to driver 1, after driver 2 filled his load to capacity.

LABEL	MEAN	STD	NUM. OF	MINIMUM	MAXIMUM
	VALUE	DEVIATION	OBSERVATONS	VALUE	VALUE
TIME	139.389	59.843	96	31.105	233.717
TIME2	139.594	56.686	89	36.245	234.401
STOPS_D1	NO VALUES RECORDED				
STPS_BUMPED_2D2	1.000	0.000	1	1.000	1.000
STOPS_D2	2.500	10291	4	1.000	4.000
STPS_BUMPED_2D1	2.000	1.000	3	1.000	3.000
FAILED PICKUPS	4.500	2.449	8	1.000	8.000

The above results just go to show that the assignment process, when not solely based on the loop and unit concept, can actually produce results far superior in nature. Of course, this simulation is not the absolute answer, by any means. There is a myriad of factors to be considered in the assignment process, ranging from technical factors to human factors, only a few of which have been incorporated in this simulation.

The results from this simulation were compared to a simulation involving a single driver responding to any and all requests originating within the loop (current method of assignment). The latter case resulted in 19 missed pickups out of a possible 100. The model involving the two drivers had a slighter lower number of missed pickups of 16 out of a possible 100.

This served as the premise to expand the model wherein loops all around the study loop were considered i.e. 9 loops in all. The assignment was once again based on the closest driver to the point of origin of the request for service. This time the results were even more encouraging. The number of missed pickups was reduced to 14 out of a possible 100. These results clearly indicate that there is potential for improvement from the current method of package allocation. In addition to the reduced number of missed pickups is the reduction in driver miles, which again translates into cost savings.

The study clearly establishes that the current system leaves a lot to be desired, and the system as a whole could use some corrective action. Maybe, by incorporating some kind of negotiation principle between the drivers and the center management team, a more effective and efficient method of package pickup dispatch could be implemented. Although the actual development of the negotiation process is out of the scope of this work, the groundwork proving the feasibility and need for such a system has been established. This improved method would be beneficial to the company as a whole, and also to the customer that employs the services of the company.