객체지향 시뮬레이터, AgvTalk의 가능성

Potential of an Object-Oriented Simulator, AgvTalk

김 경 섭*
Kyung Sup Kim

Abstract

In this paper, the modeling capabilities of an object-oriented simulator, AgvTalk is demonstrated by designing and simulating a conceptually different configuration of AGV systems, known as the tandem configuration. The characteristics and design methodology in AgvTalk are described between the tandem configuration for AGV systems and conventional AGV systems. Also, simulations between the conventional AGV system and its corresponding tandem AGV system are compared with AgvTalk in the job shop environment. From the simulation results, the characteristics of each system observed are discussed.

1. INTRODUCTION

Recently, Kim [6] developed the object-oriented simulation modeling environment for AGV systems, called AgvTalk, where Smalltalk-80 [5] had been used as an implementation language. In the process of developing AgvTalk, two major tasks had been performed. First, the existence of objects and their relationships in AGV systems was conceptualized. This conceptual organization was represented in the logical design of AGV systems by the object and class diagrams [2]. This design includes the object-oriented design of a model, an experiment, and a simulation framework for AGV systems. Second, this logical design was applied to the implementation of object-oriented classes providing the ability to create a model for AGV systems. The resulting simulation modeling environment, AgvTalk, includes 25 object classes and more than 300 object methods in its library for every detailed feature of AGV systems.

Also, AgvTalk provides a user-friendly simulation modeling environment. Since the complexities and specific natures of AGV systems do not allow easy construction of a simulation model with only stand-alone natures of objects, AgvTalk proposed the hybrid approach. In the hybrid approach, the possible life cycles of active objects were modeled in methods, and the methods were stored in the AgvTalk library. This hybrid approach eliminates the modeling process of behaviors of vehicles, work stations, parts, and repair stations in AGV systems; instead, it provides the selection process among behaviors already built-in in

* 삼성데이타 시스템 CIM개발실

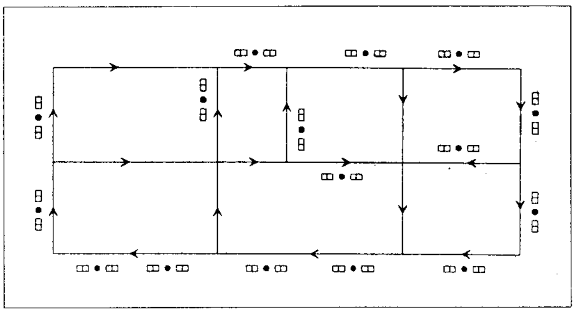
the library. This selection process and data input process for model construction can be performed through the window- or menu-based user interface.

In this paper, the potential of AgvTalk, that is, the modeling capabilities by extensibility and reusability, is demonstrated. To accomplish this objective, the tandem AGV system is designed and extended in the AgvTalk environment. The tandem AGV system has a radically different configuration from the conventional AGV systems in the system network layout and travel behaviors of vehicles. By redefining the system network layout and travel behaviors of vehicles, and reusing the AgvTalk library already developed for the conventional AGV systems, the tandem AGV system can be easily extended. Also, the simulation experiments are performed, and the results are analyzed and tested among the conventional, tandem, and semi-tandem AGV systems.

2. Tandem AGV System vs Conventional AGV System

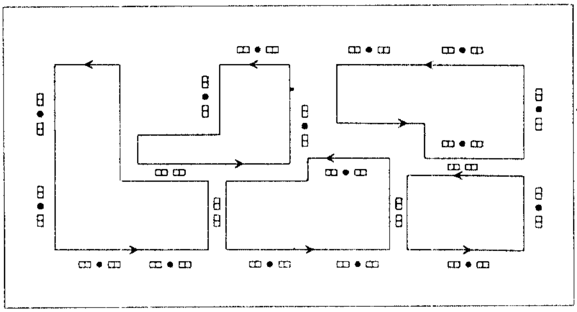
In conventional AGV systems in Figure 1, each AGV can move around the system along the system guidepath. If a station is reachable through the guidepath, the station is accessible to any AGV in the system. Since all stations are accessible to each AGV in conventional AGV systems, sophisticated control systems are required to manage vehicle dispatching, vehicle routing, and flow control. In AgvTalk, these control systems were fully implemented in the classes, Dispatching, Routing, Zone, and AGVSController.

In the tandem configuration for AGV systems as shown in Figure 2, which was suggested as an alternative to the conventional AGV system by Bozer and Srinivasan [3], the movement of AGVs is defined differently in the system guidepath which is different from conventional AGV systems. The system paths are divided into non-overlapping, single vehicle closed loops. The locations of the stations remain the same as those of the corresponding conventional AGV system; however, each station belongs to only one loop. Only one AGV is assigned to each loop, completely eliminating traffic problems such as vehicle collision and vehicle congestion due to zone blocking. The movement of the AGV in the loop is determined by the First-Encountered-First-Served (FEFS) dispatching policy presented by Bartholdi and Platzman [1]. Under this policy, an unloaded AGV keeps moving to the next (adjacent) station in its loop until it finds a part waiting in the output buffer. The first encountered part is loaded to the AGV and delivered to the next station in its routing sequence. The communication between adjacent loops is performed by the interface. The main reasoning behind the tandem configuration is to reduce control problems inherent to conventional AGV systems. Bozer and Srinivasan also pointed out that the efficient operation of tandem AGV systems would require balanced workloads among the loops to avoid creating bottleneck loops.



⟨Figure 1⟩ Conventional AGV System

For conventional AGV systems, many different configurations can be modeled very easily in AgvTalk because all elements in AGV systems are modularly designed as objects. Different configurations can be immediately implemented by assigning new or changed characteristics to the related objects. These processes may involve creating subclasses, modifying tasks, and



〈Figure 2〉 Tandem AGV System

assigning new instance variables.

However, from the design and operational points of view, the tandem AGV system is radically different from the conventional AGV system. The major differences fall into two categories, 1) system network definition and 2) travel of AGVs. The design of these differences in AgvTalk is discussed in the following sections.

2.1 System Network Definition

In AgvTalk, the system network is defined by the following three processes:

1. Definition of the control points (stations, intersections, etc) as instances of the class Node,
2. Connection of all possible combinations of two adjacent control points, and
3. Generation of the shortest routes between any combination of two control points.

From the operational point of view, the system network of the tandem AGV system is radically different from that of the conventional AGV system. However, the differences between two systems in defining the system network are minimal. The additional process required for the tandem AGV system is the assignment of an instance variable *loop* to the instances of the Node, which are defined as control points in process 1. For example, if station 1 belongs to loop 1, the process in AgvTalk is defined as follows;

Node new name: 'station1' inLoop: 1

In this process, the number of the loop to which the control point belongs is assigned to the *loop*. The interface among loops is defined just like another control point; however, the string of 'interface' is assigned to the variable, *loop*. The instance variable *loop* is used to prevent each AGV from moving into other loops.

In process 2, for each system path segment between two adjacent control points, the number of zones is defined for zone blocking. In the tandem system, only one AGV is allowed in each loop and each system path segment must belong to one loop. Thus, each system path segment is accessible to only one AGV, which completely eliminates traffic problems. Therefore, the number of zones between control points does not affect system performance, and each path segment is defined to have only one zone for simplicity purposes.

There is no difference in defining process 3 between the conventional and the tandem AGV systems.

2.2 Travel of AGVs

From the material handling point of view, the AGV system can be summarized as a group of repeating processes of AGVs which travel from one station to the other station in the given system network. The travel between two stations can be characterized in many different ways. It may be travel for pick up or deposit tasks. It may be travel to the next station without a prior task assignment. Also, it may be travel to the staging area, or the recharging station.

In AgvTalk, the repeating process of each AGV, that is, the behavior from the arrival at a station to the arrival at the next station for a given AGV is defined in the method tasks of the class AGV. That is, when an AGV arrives at the station, the appropriate interactions with the buffer of the station are performed according to the AGV's travel purpose to this station. For conventional AGV systems, the travel purposes are to pick up or deposit a part, to wait at the staging area, to be recharged at the recharging station, etc. Even though there are some changes in the system configuration, the processes defined in the method tasks remains the same as long as the system logic for the AGV movement behavior remains unchanged. However, different logic-based AGV systems such as a tandem AGV system should have different behavior in the method tasks, since the travel purposes of AGVs are different. In the tandem AGV system, the travel purpose of AGVs should be one of the following two: to pick up a part or just to travel to the next adjacent station in its loop until it encounters the pick up task.

Unlike conventional AGV systems, the tandem configuration includes interfaces. When an AGV arrives at the interface and there are parts waiting for an AGV to be delivered, the first part in the interface is examined if it is an incoming part from another loop. If it is an incoming part, it is picked up by the AGV and delivered to the next station in its routing sequence. For this process in AgvTalk, the instance of the Part has an instance variable *loop*, and the loop number of the instance AGV is assigned to the variable *loop* whenever a part is loaded onto the AGV. As long as the part instance remains in the loop, the value of *loop* is unchanged. When the AGV instance arrives at the interface, and if there are parts waiting and the loop number of the first part is the same as the loop number of the AGV, the part must be an outgoing part to another loop, and must not be picked up by the AGV. If the loop numbers of the part and the AGV are different, the part must be an incoming part from an adjacent loop. Thus, the part instance is picked up by the AGV instance, and the loop number of the part is updated to that of the AGV. The method tasks for the conventional and tandem AGV systems are presented in Figures 3 and 4.

When an AGV finishes its task upon arrival at a station, the detailed information of the next travel is provided by its controller, which is an instance of the class AGVSController. The process of generating information in the controller is different for the different logic-based AGV systems. The information is generated in the method updateInformationFor: anAGV. Thus, the repeating process for each AGV travel is completed by the combination of the method tasks in the class AGV and the method updateInformationFor: in the class AGVSController.

In the tandem configuration, if there are no parts waiting in the output buffer of the station, the next station for an AGV is determined by the FEFS dispatching rule. That is, the AGV travels to the next adjacent station along the path of the loop. This empty vehicle movement continues until the AGV encounters the station which has parts waiting to be delivered. Unlike other dispatching rules, the FEFS dispatching rule is defined in the instance variable of the AGV. Each AGV instance which belongs to the loop stores the sequence of stations along the path of the closed loop in the instance variable, *loop*, which is the instance of the class LinkedList. In the variable *loop*, the adjacent stations in the closed loop are linked to each other. If the name of one station is given, the name of the linked station is returned from the instance of the LinkedList. Therefore, vehicle initiated and workcenter initiated dispatching rules [4] are never invoked in tandem AGV systems.

If there are parts waiting, the next station is determined by the part routing. However, because the AGV cannot travel outside the loop, the location of the next station in the part routing should be examined first. If the location is in another loop, the first interface in the routing sequence between the current station and the next station in the part routing is determined as the next destination. The updateInformationFor: methods for the conventional and tandem AGV systems are presented in Figures 5 and 6.

```
tasks
        "Conventional system : This shows how a vehicle behaves from
        the arrival at the station to the arrival at the next station.  The AGV
        system includes a staging area."

        status = 'arrival'
        ifTrue:
                [currentStation := 'staging']
        ifFalse:
                [currentStation := self controller nextStation].

        [true]
        whileTrue:
        [
        (currentStation = 'recharging')
        ifTrue:
                [self beingRecharged
                ].

        (currentStation = 'staging')
        ifTrue:
                [self waitForAWhileAt: 'staging'
                ].

        (((currentStation = 'staging') not) and: [(currentStation = 'recharging') not])
        ifTrue:
                [task = 'pickUp'
                ifTrue:
                        [
                        Buffer releaseLoadForDeliveryAt: currentStation to: self
                        ]
                ifFalse:
                        [
                        Buffer acceptLoadForProcessingAt: currentStation from: self
                        ]
                ].

        self controller updateInformationFor2: self.
        self controller transport: self.

        currentStation := self controller nextStation
        ]!
```

〈Figure 3〉 The Method tasks for Conventional Configuration

```
tasks
        "Tandem configuration : This shows how a vehicle behaves from
        the arrival at the station to the arrival at the next station when an
        AGV system is a tandom configuration of loops "
        | outputBuffer |

        [true]
        whileTrue:
        [
        task = 'deposit'
        ifTrue:
                [
                Buffer acceptLoadForProcessingAt: currentStation from: self
                ].

        outputBuffer := (OutputBuffer at: currentStation).
        (outputBuffer isEmpty) not
        ifTrue:
                [
                (((Node interface includes: currentStation) not)
                or:
                [(Node interface includes: currentStation) and:
                        [(outputBuffer first loop = self loopNo) not]])
                ifTrue:
                        [
                        task := 'pickUp'.
                        Buffer releaseLoadForDeliveryAt: currentStation to: self
                        ]
                ifFalse:
                        [task := 'moveInLoop']
                ]
        ifFalse:
                [task := 'moveInLoop'].

        self controller updateInformationInLoopFor: self.
        self controller transport: self.

        currentStation := self controller nextStation
        ]!
```

〈Figure 4〉 The Method tasks for Tandem Configuration

3. Simulation Experiment

In this section, AgvTalk is used to experiment with three different AGV systems. The three systems are a conventional system, a semi-tandem system, and a tandem system.

Like a tandem system, the system paths of a semi-

```
updateInformationFor: anAGV
        | aJob |

        currentStation := anAGV station.
        start := self determineSpecificLocation: currentStation.
        task := anAGV task.

        (((anAGV recharge = 'on') and: [(currentStation = 'staging') not])
                and: [(currentStation = 'recharging') not])
        ifTrue:
                [
                nextStation := 'recharging'.
                task = 'pickUp'
                ifTrue:
                        [
                        aJob := anAGV showJobCarrying.
                        nextStationAfterRecharging := aJob route after: currentStation
                        ].
                anAGV turnOffLight
                ]
        ifFalse:
                [
                currentStation = 'staging'
                ifTrue:
                        [
                        nextStation := NextStation.
                        RequestQ removeOne: nextStation.
                        anAGV assignedTask: 'pickUp'
                        ]
                ifFalse:
                        [
                        task = 'deposit'
                        ifTrue:
                                [
                                RequestQ notEmpty
                                ifTrue:
                                        [nextStation := Dispatching new
                                        determineNextStationByRuleAt: currentStation.
                                        RequestQ removeOne: nextStation
                                        ]
                                ifFalse:
                                        [nextStation := 'staging'].
                                anAGV assignedTask: 'pickUp'
                                ].

                        task = 'pickUp'
                        ifTrue:
                                [
                                currentStation = 'recharging'
                                ifTrue:
                                        [nextStation := nextStationAfterRecharging
                                        ]
                                ifFalse:
                                        [
                                        aJob := anAGV showJobCarrying.
                                        nextStation := aJob route after: currentStation
                                        ].
                                anAGV assignedTask: 'deposit'
                                ].
                        ].
                ].

        task := anAGV task.
        end := self determineSpecificLocation: nextStation!
```

〈Figure 5〉 The Method updateInformationFor: for Conventional Configuration

```
updateInformationInLoopFor: anAGV
        | aJob  aStation |

        currentStation := anAGV station.
        start := self determineSpecificLocation: currentStation.
        task := anAGV task.


        task = 'moveInLoop'
        ifTrue:
                [
                nextStation := anAGV nextStationInLoop: currentStation
                ].

        task = 'pickUp'
        ifTrue:
                [
                aJob := anAGV showJobCarrying.
                aJob isAssignedTo: anAGV loopNo.
                aStation := self findNextStationInJobRoute: aJob.
                nextStation := self findNextStationOf: aJob
                                betweenLoopsFrom: currentStation to: aStation.
                anAGV assignedTask: 'deposit'
                ].

        task := anAGV task.
        end := self determineSpecificLocation: nextStation!
```

〈Figure 6〉 The Method updateInFormationFor: in Tandem Configuration

tandem system consist of a number of nonoverlapping, closed loops; however, more than one AGV can be assigned to each loop, and each AGV belongs to only one loop. Since the movement of AGVs is confined in each loop, extensive control systems are not required. However, because there can be more than one AGV in the loop, zone blocking is used to control traffic problems such as vehicle collision and congestion. The number of the loops in the semi-tandem system is expected to be smaller than that in the corresponding tandem system, while for the size of the loops, the opposite is true.

As pointed out by Bozer and Srinivasan [3], the development of an efficient tandem configuration from a given set of stations and part flow data is not a simple task and is another research area. Thus, tandem configurations are developed only with a given set of stations. Then, different sets of part flow data are provided for each configuration.

The job shop has been chosen for the experiment. The layout configuration of the job shop in the conventional AGV system is presented in Figure 7. The corresponding job shop configurations in the semi-tandem and tandem AGV systems are shown in Figures 8 and 9 respectively. The semi-tandem configuration has 3 loops, and the tandem configuration has 9 loops. There are 24 work stations in the job shop, and the locations of work stations are the same in three configurations. The conventional configuration has an additional staging area for idle AGVs, and the semi-tandem and tandem configurations have 2 and 12 interfaces, respectively, to allow part routing among adjacent loops.

Three different part types are produced in the job shop, and the arrival rates of each part type are as follows;

Part type 1 : Exponential (12)

Part type 2 : Normal (12, 2)

Part type 3 : Triangular (10, 13, 15)

There are three different sets of part routings. In the first set, the routing sequence of each part type is randomly selected. In the second set, the routing sequence of each part favorably matches the station sequences of loops in the semi-tandem configuration. In the third set, the routing sequence of each part favorably matches the station sequences of loops in the tandem configuration. The routing sequences of three part types in each set are as follows:

⟨Table 1⟩ Part Routing Sequence in Part Flow Set 1

	Part routing sequence (Station number)									
Part type 1	IN1	2	15	23	11	14	6	9	13	OUT1
Part type 2	IN1	1	22	19	10	3	8	7	24	OUT1
Part type 3	IN1	4	16	5	17	18	21	20	12	OUT1

⟨Table 2⟩ Part Routing Sequence in Part Flow Set 2

	Part routing sequence (Station number)									
Part type 1	IN1	1	14	16	18	9	10	11	12	OUT1
Part type 2	IN2	2	20	22	24	8	17	15	13	OUT2
Part type 3	IN3	3	4	5	6	7	23	21	19	OUT3

⟨Table 3⟩ Part Routing Sequence in Part Flow Set 3

	Part routing sequence (Station number)									
Part type 1	IN1	1	14	12	16	11	18	9	10	OUT1
Part type 2	IN2	2	20	13	22	15	24	8	17	OUT2
Part type 3	IN3	3	4	19	5	21	6	7	23	OUT3

Therefore, the conventional system with its part flow data has its tandem and semi-tandem counterparts. Likewise, the tandem system in which the part routing sequence matches the sequence of stations in its loops has its conventional and semi-tandem counterparts. The semi-tandem system also has its conventional and tandem counterparts. Since one of three different configurations can be combined with one of three part flow sets to complete one AGV system, there can be nine different AGV systems.

Each system was simulated for 480 minutes with a warm up time of 50 minutes, and this 480-minute simulation is replicated 12 times. Thus, the actual simulation time for each system is 5160 (430*12) minutes. There are nine AGVs in each AGV system. For the tandem configuration, one AGV is assigned to each loop. For the semi-tandem configuration, three AGVs are assigned to each loop. The processing time at each work station is given as 5 minutes; the loading and unloading times of each AGV are given as 0.75 minutes. The velocity of each AGV is 160 ft/min. Simulation results are in Tables 4, 5, and 6.

4. Observations

4.1 Conventional AGV System

As shown in Tables 4, 5, and 6, and as expected, the system performance of the conventional system is not as sensitive as that of the semi-tandem and tandem configurations to the part routing. In the conventional AGV system, the travel of AGVs to complete part routing is more flexible since each station is accessible to all AGVs, and there are many possible different ways to reach the destination station for the given locations of the AGVs. Thus, different part routing sequences do not affect the system performances very critically.

〈Table 4〉 Summary of Simulation Outputs for Conventional AGV System

	No. of parts arrived	No. of Parts inducted	Throughput	Average flow time (min)	% of loaded vehicle travel
Part flow set 1	1217	1184	987	126.49	57%
Part flow set 2	1252	1248	1102	100.89	42%
Part flow set 3	1195	1188	1064	104.15	43%

〈Table 5〉 Summary of Simulation Outputs for Semi-Tandem (3-Loop) AGV System

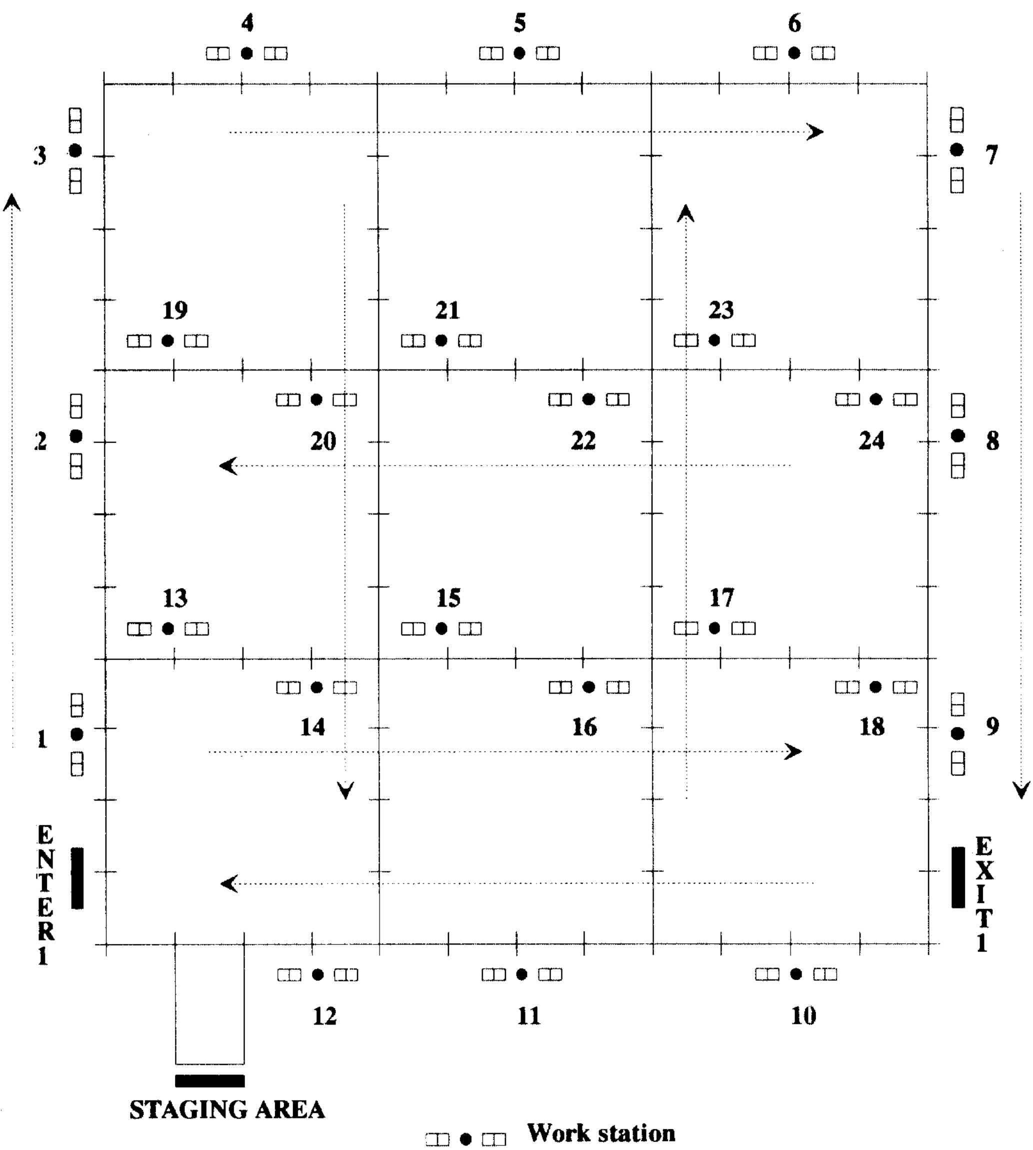
	No. of parts arrived	No. of Parts inducted	Throughput	Average flow time (min)	% of loaded vehicle travel
Part flow set 1	1239	1174	53	345.75	48%
Part flow set 2	1186	1184	1147	71.8	21%
Part flow set 3	1181	1170	1080	91.67	46%

〈Table 6〉 Summary of Simulation Outputs for Tandem (9-Loop) AGV System

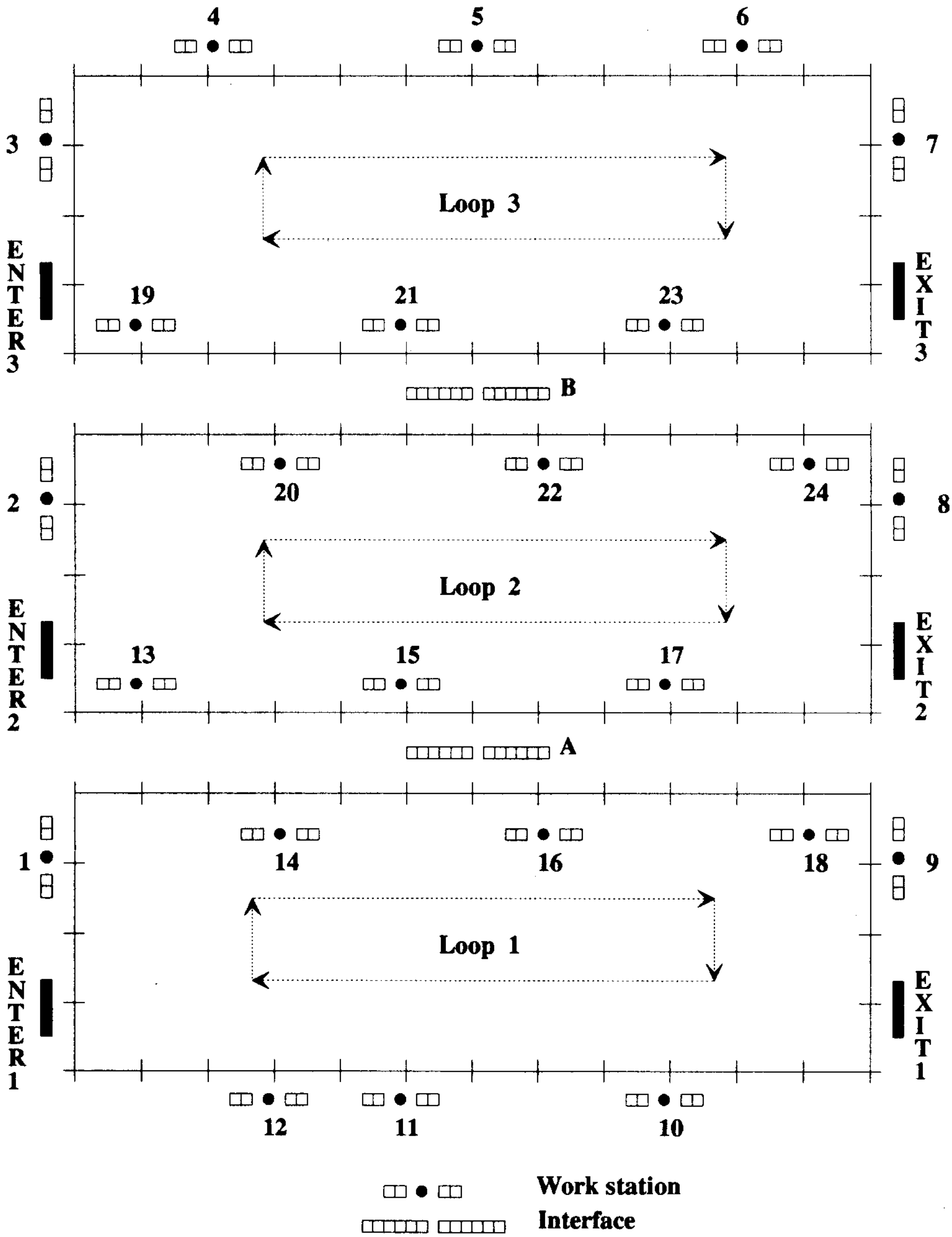
	No. of parts arrived	No. of Parts inducted	Throughput	Average flow time (min)	% of loaded vehicle travel
Part flow set 1	1212	1002	171	296.95	38%
Part flow set 2	1231	1212	1036	113.62	30%
Part flow set 3	1199	1196	1105	93.45	31%

4.2 Semi-Tandem (3-Loop) AGV System

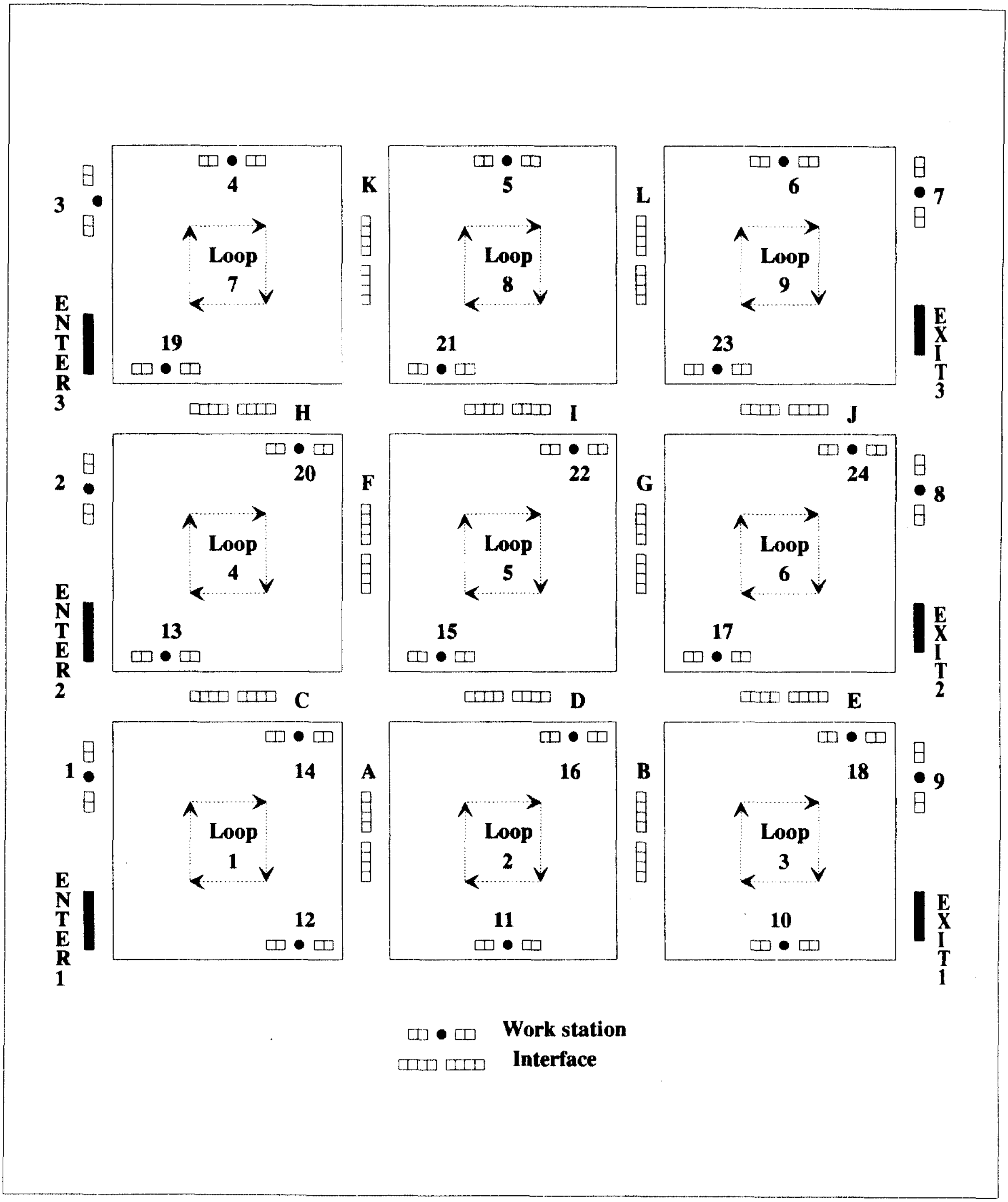
The system performance of this configuration is very sensitive to the part routing sequence. Since there are only 2 interfaces and the loops are large, the AGV travel time to complete one part routing sequence might be very large. If the stations in the part routing sequence are selected randomly among loops, the travel time between the loops will significantly delay the vehicles. As shown in Table 5, when parts are routed according to the sequences of part flow set 1, only 53 units out of 1174 units were produced and the average flow time of each unit is 345.75 minutes. This is very poor system performance compared with part flow sets 2 or 3. In set 1, the routing sequence of each part type requires several trips between loops, whereas, the routing sequence of each part type in set 2 and set 3 does not require trips between loops. Moreover, the routing sequence of each part type in set 2 perfectly matches the station sequence of each loop respectively. Thus, each part travels a minimum distance from the entering location to its destination in the system. This results in the best performance in all combinations of system configuration and part



〈Figure 7〉 Conventional Configuration for AGV System



〈Figure 8〉 Semi-Tandem (3-Loop) Configuration for AGV System



〈Figure 9〉 Tandem (9-Loop) Configuration for AGV System routing sequence.

In part flow set 3, stations in the routing sequence of each part type belong to one loop; however, the routing sequences do not match the station sequences of loops. This causes each part to travel several extra loops. However, system performance of this configuration is still very good because travel between loops is totally eliminated.

4.3 Tandem (9-Loop) AGV System

In a tandem AGV system, the loops which include the part-entering station have a high probability of being a bottleneck loop. Since only one AGV serves each part-entering loop, there is a limit to the rate of induction of parts to the system. In each part-entering loop, if all stations except the part-entering station do not require any delivery tasks and there always exist parts waiting to be delivered in the part-entering station, each time the AGV passes through the part-entering station, the part will be picked up and delivered to the interface of the entering loop. This maximizes the number of parts inducted into the system. Throughput will never exceed this limit. Even though the part arrival rate increases, the number of parts inducted will not increase.

Also, as shown in Table 6, the number of parts inducted is sensitive to the part routing sequence in the tandem system. With part flow set 1, 1002 units were inducted into the system. This is a low number of units compared to 1212 and 1196 units for part flow sets 2 and 3, respectively. Comparing the number of parts inducted with that of the other two configurations (Tables 4 and 5), the tandem configuration with part flow set 1 seems to create a bottleneck loop. In the tandem system, in order to induct as many parts as possible, the AGV which belongs to the entering loop should not spend much time performing delivery tasks. The more time the AGV spends in delivery tasks, the more frequently the AGV passes through the part-entering station without picking up a part. In part flow set 1 (Table 1), the stations of the part-entering loop 1, that is, stations 1, 14, and 12, are in the routing sequences of different part types. Similarly, the stations of the part-entering loops 4 and 7 are in the routing sequence of different part types or are not sequentially located. Thus, a part arriving at one of these stations will require a delivery task to a station in another loop. This results in additional delivery tasks at the interfaces of the entering loop. In part flow set 2, stations of the part-entering loops 1, 4, and 7 are in the routing sequence of one part type, however, they are not sequentially located. In part flow set 3, stations of the part-entering loops 1, 4, and 7 are in the routing sequence of one part type and are sequentially located. This minimizes the number of delivery tasks at the interfaces. Therefore, in the tandem system, the part routing sequence is very critical to system performance. It is suggested for the entering loop to have only a part-entering station and interfaces.

As shown in Table 6, the system performance with part flow set 2 is nearly the same as with part flow set 3. The part routing sequences in part flow set 3 perfectly matches the station sequences in loops. However, since AGVs only move in one direction in each loop, the travel time from one loop to another loop may require a longer trip than the actual distance. For example, when a part (type 1) of part flow set 3 needs to be moved from station 12 (loop 1) to station 16 (loop 2) in the routing sequence, the routing path in Figure 9 is

12 - ENTER - 1 - 14 - A - D - 16.

If all paths are bi-directional, the path will be

12 - A - D - 16.

In part flow set 2, the flow sequence of each part type does not match the station sequences in the loops. However, the flow sequence of each part type matches very well with the station sequences in the combination of three loops with minimum travel between loops in the part routing. The travel between loops in the part routing sequence is another important factor to be considered in the tandem configuration.

In part flow set 1, the part routing sequence was determined without considering the station sequences of loops in the semi-tandem and the tandem configurations. As shown in Table 5, with part flow set 1, only 53 units are produced in the semi-tandem configuration, whereas, 179 units are produced in the tandem configuration. The tandem configuration is not as sensitive as the semi-tandem configuration since there are more loops and more interfaces.

4.4 General Observations

In this experiment, the semi-tandem and tandem AGV systems with their matching part types in part routing sequences produce better system performances than the conventional AGV system. This is because the semi-tandem and tandem AGV systems have almost ideal configurations such as perfectly balanced load among loops, perfectly matching part types in their routing sequences, minimum number of parts' travel among loops, etc. In the real world, a conveyor system looks more suitable for these ideal environments than the tandem AGV system due to cost considerations. Also, these ideal semi-tandem and tandem configurations are not the corresponding counterparts for a conventional AGV system. The corresponding counterparts should be produced from the given conventional AGV system which has already determined each part routing sequence. In this experiment, except the locations of workstations, the semi-tandem and tandem configurations were constructed with their matching part routing sequences regardless of the configuration of the conventional AGV system. The methodology to produce the tandem counterpart has not been reported in the literature.

5. Summary

For the experiment of AgvTalk, a radically different configuration for AGV systems, that is, the tandem AGV system, was simulated and compared with the conventional AGV system. The concept and characteristics of the tandem configuration for AGV systems were discussed in the context of AgvTalk.

The tandem AGV system has been extended in AgvTalk by redefining the two different processes, the system network definition and the travel behaviors of AGVs. For the other processes necessary for the tandem AGV system, the AgvTalk library already developed for the conventional AGV systems is reused.

From the simulation results, it is observed that the performance of the conventional system is not very sensitive to the part routing sequences. Also, it is observed that there are some major factors that should be considered in developing the tandem configuration, and they are as follows:

- part routing sequences
- travel path between loops in part routing sequence
- design of the entering loop according to the part arrival processes
- vehicle speed

REFERENCES

[1] Bartholdi, J.J. and Platzman, L.K., "Decentralized Control of Automated Guided Vehicles on a Simple Loop", *IIE Transactions*, pp76-81, March 1989.

[2] Booch, G., *Object-Oriented design with applications*, The Benjamin/Cummings Publishing Company, Inc., 1991.

[3] Bozer, Y.A. and Srinivasan, M.M., "Tandem Configurations for AGV Systems Offer Simplicity and Flexibility", *Industrial Engineering*, pp23-27, Feb. 1989.

[4] Egbelu, P.J. and Tanchoco, J.A., "Characterization of automatic guided vehicle dispatching rules", *International Journal of Production Research*, Vol.22, No.3, pp359-374, 1984.

[5] Goldberg, A. and Robson, D., *Smalltalk-80: The language*, Addison-Wesley, 1989

[6] Kim, K.S., "Object-Oriented Simulation Approach for AGV Systems", *Journal of The Korea Society for Simulation*, Vol.2, No.1, pp107-124, December 1993.

● 저자소개 ●

김경섭

1982 연세대학교 기계공학 학사

1986 미국 University of Nebraska-Lincoln 산업공학 석사

1993 미국 North Carolina State University 산업공학 박사

현재 삼성데이타시스템 CIM개발실/생산물류정보팀

관심분야: 시뮬레이션, 물류시스템, 제조시스템