

컨테이너 터미널 시뮬레이터의 객체지향 설계

윤원영* · 류숙재** · 김귀래*** · 김도형**** · 최용석*****

*부산대학교 산업공학과 교수, **,***,****부산대학교 대학원,*****순천대학교 물류학전공 교수

The Development of the Object Oriented Simulator of the Container Terminal

Won-Young Yun* · Sook-Jea Ryu** · Gui-Rae Kim*** · Do-Hyung Kim**** · Yung-Suk Choi*****

*,**,***,****Department of industrial Engineering , Busan National University

*****Major of Logistics , SunChon National University

요 약: 컨테이너 터미널은 육상수송과 해상 수송의 연결기능을 수행하는 복합 운송시스템이다. 이 시스템은 내부적으로 본선 작업 시스템, 구내 이적 시스템, 장치/보관 시스템, 게이트 작업 시스템, 정보 관리 시스템등의 운영시스템을 가진다. 본 논문에서는 터미널의 배치 평가와 컨테이너 장비의 활용도 평가를 주요 목적으로 하여 장치장과 선석의 배치 대안에 대한 평가, 취급 장비의 개별적인 운영 효율성을 평가는 시뮬레이터의 개발이 목적이다. 시뮬레이션 모형의 개발은 설계 및 모델링 단계에서 재사용성이 높고 모듈화하여 이식이 용이한 객체 지향 기법을 이용하였다.

핵심 용어 : 객체지향, 장치장, 선석, 시뮬레이션, 설계

ABSTRACT: The container terminal is the unified transportation system which connects between a land transportation and a sea transportation. This system has many subsystems such as ship operation, yard transfer operation, yard storage system, gate operation, and information manage system. This paper presents a method of modeling a simulator with which user can evaluate the efficiency of the equipment and allocation policies in the container terminal. The final purpose is to estimate the efficiency of each equipments and the distribution policies. In a design step in the simulator development. We suggest the Object Oriented method with which the developer can easily design, because the object oriented method has the advantages of reusability and modularity.

KEY WORDS : Object Oriented, Class, Yard, Berth, Object-Oriented, Yard, Berth, Simulation, Design

1. 서 론

현재 컨테이너 수송의 가장 큰 변화는 선박의 대형화로 인해 컨테이너 터미널에서 발생하는 비용이 해마다 증가하고 있다. 그러므로 내부 장비(Query Crane, Transfer Crane, Yard Truck)수 증가 와 장비의 성능 향상 및 시설 확충(Yard 공간 확보, Gate 수 증가)이 필요하게 되었다. 시설 및 장비의 중요성과 더불어 장비와 시설의 거대화로 인한 효율적인 터미널 운영이 더욱더 중요한 비중을 차지하게 되었다.

컨테이너 터미널의 운영 관리는 서로 다른 위치에서 전체적인 전략부터 단순한 컨테이너 움직임을 제어하는데 까지 아

주 다른 수준에서 수많은 결정 변수를 가지는 매우 복잡한 업무이다. 터미널 사용자가 가장 빠르고 편리하게 서비스를 받도록 하기 위한 장치장 할당 정책, 장비 할당 정책, 양/적하 정책에서 결정변수를 찾는 방법은 수학적 모델링으로 해결하기는 쉽지 않다. 그러므로 이러한 운영 정책들을 평가하기 위해 가장 좋은 방법은 시뮬레이션을 이용하는 것이다.

컨테이너 터미널 설계와 운영 그리고 그 결과에 의한 운영 평가에 관한 많은 연구가 이루어지고 있으며 시뮬레이션 모델에서 객체지향 기법을 이용한 모델링도 많은 연구가 이루어지고 있다.

윤원영 & 최용석(1999)은 컨테이너 터미널의 운영 방식,

*대표저자: 윤원영(중신회원), wonyun@pusan.ac.kr 051)510-2421

**류숙재(일반회원), ryu21sky@pusan.ac.kr 051)510-1472

***김귀래(일반회원), grrk@pusan.ac.kr 051)510-1483

****김도형(비회원), rta6151@naver.com 051)510-1472

*****최용석(중신회원), drasto@sunchon.ac.kr 061)750-5115

즉 선석계획, 양/적하계획, 장치장 계획 등을 평가하기 위한 시뮬레이션 모형을 제시하여 실제 시뮬레이션을 수행하여 결과를 제시하였다. 장성용 & 이원용(2002)은 가상의 특정 터미널을 시뮬레이션 기법을 이용한 컨테이너 터미널 운영 규칙의 평가에 관한 연구가 이루어졌다. 최용석 & 하태영(2004)은 객체지향 기법을 이용하여 컨테이너 터미널 시뮬레이터의 설계 및 모델링 방법과 개발된 시뮬레이터를 이용한 터미널 평가에 관한 연구가 이루어졌다. Maurizio Vielli & Azedine Boulmakoul(2005)은 객체 지향 기법을 이용한 시뮬레이터 모델링 방법과 시뮬레이터를 이용한 할당 규칙의 평가에 관한 연구가 이루어졌다. 그러나 일부 연구들은 특정 터미널에만 한정적이다. 시뮬레이션의 주목적인 컨테이너의 평가를 위한 구체적인 값을 구하기 위한 세분화된 설계가 미흡했다. 본 논문의 시뮬레이션 설계 방식을 제시하며 보다 정확하고 빠른 시뮬레이터를 만드는 것이 목적이다.

2. 컨테이너 터미널 시스템 개요

컨테이너 터미널에 사용되는 주요 장비는 안벽 크레인(Container Crane), 내부 운송장비(Yard Truck), 장치장 크레인(Transfer Crane), 외부 트럭(Trailer)으로 구성되어 있다. CC(Container Crane)은 배에서 컨테이너를 내리거나 컨테이너를 배의 특정 위치에 싣는 작업을 한다. TC(Transfer Crane)은 YT(Yard Truck)나 Trailer가 운반해온 컨테이너를 장치장의 적절한 위치에 놓거나 장치되어 있는 컨테이너를 가져와서 YT나 Trailer에 싣는 역할을 한다. YT은 CC와 TC 사이의 컨테이너 이동을 담당한다. Trailer는 터미널 외부에 컨테이너를 가져오거나 내부에서 가지고 나가는 역할을 담당한다.

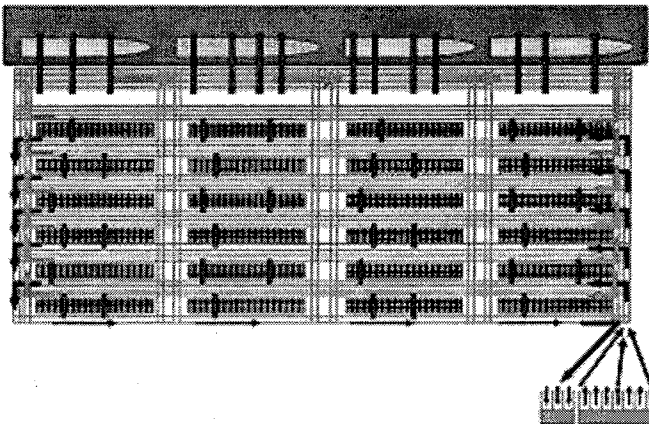


Fig. 1 Horizontal Layout of a Container Terminal

컨테이너 터미널에서 주요 작업은 선석(Berth) 부분에서 일어나는 양하, 적하와 Gate에서 발생하는 반입, 반출로 크게 나눌 수 있다. 양하(Unloading)는 배에 실린 컨테이너를 내려서 장치장에 장치하는 작업이며, 반출은 배

(Ship)에서 내려 장치장에 장치 후 컨테이너를 Trailer가 터미널 밖으로 가지고 나가는 작업이다. 반입은 외부에서 Trailer가 컨테이너를 운송하여 장치장에 장치하는 작업이며 적하>Loading)의 경우에는 반입되어 장치장에 장치되어 있는 컨테이너를 배에 싣기까지의 과정이다. 양/적하의 경우 하역 장비 중 CC, YT, TC이 작업을 수행하며, 반입, 반출의 경우 TC, Trailer가 작업에 연관되어 있다.

터미널의 생산성은 CC의 작업이 얼마나 원활하게 이루어지느냐에 달려 있으며 이는 단순히 CC의 작업을 최적화하는 것만으로는 달성되지 않는다. CC가 쉬지 않고 지속적으로 작업하기 위해서는 터미널 내 다른 장비의 생산성을 최대화하고, 동시에 각 장비간의 효율적인 운영 동기화를 통해 물류의 흐름을 최적화해야만 한다. 반, 출입 작업 운영은 Gate에서 Trailer가 수행해야 할 업무에 따라 이동해야 할 위치(Transfer Point)를 결정하고 TC에게 작업을 할당하는 것이다. 본선 작업 운영은 양/적하 계획에 의해 생성된 작업에 따라 CC의 작업을 할당하고 작업 순서에 따라 YT을 할당하고 할당된 작업 순서에 따라 YT에 작업을 지시한다. 장치장 운영은 장치장 Block 내에서 TC의 작업할당과 Trailer와 YT에 의해 도착된 컨테이너 처리 순서를 결정하는 운영 시스템이다.

3. 컨테이너 터미널 시뮬레이터 설계

3.1 Class Diagram

가장 먼저 해야 할 일은 Class를 개념적으로 정의하는 일이다.

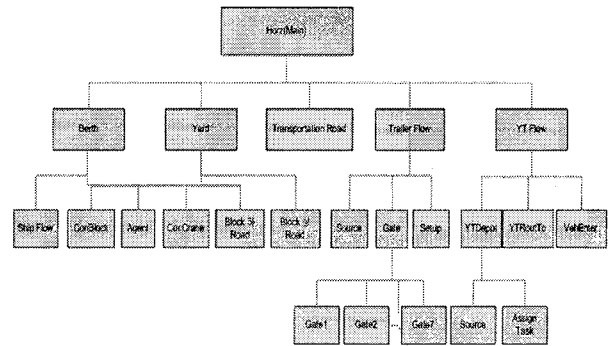


Fig. 2 The object infrastructure

객체 설계에서 클래스와 객체의 관계뿐만 아니라 객체의 속성까지도 포함하며 객체의 속성은 객체의 상태를 파악하거나 의사 결정에 의한 갱신 값이라고 할 수 있다.

Table 1 과 Table 2 는 컨테이너 터미널에서 사용되는 기본 Class의 속성과 메소드를 나타내었다. 속성 값은 다른 Class에서 메소드로 해당 Class의 속성 값을 갱신한다. 즉, 메소드를 통해 Class 간 통신이 가능하다.

- Block

YardBlock 혹은 Berth Block으로 상속되어 Yard 내의 Bay, Row, Tier을 가지고 각각의 Container의 위치 정보를 바탕으로 점유, 비점유 등 블록내의 물리적인 공간 할당을 나타낸다.

• Crane

Yard 내에서 작업하는 있는 TC와 Berth 내에 작업하는 CC로 상속되며 각각의 작업할당에 따라 작업을 수행하며 loading, unloading작업을 담당한다.

Table.1 The attribute of an object

객체	속성
Container	size20 : Integer : 20*TEU 크기 size40 : Integer : 40*TEU 크기 TYPE_IMPORT : Integer : Container Type TYPE_EXPORT : Integer : Container Type TYPE_TRANS : Integer : Container Type WIDTH : Double : Container 폭 HEIGHT : Double : Container 높이 LENGTH20 : Double : 20*TEU Container 길이 LENGTH40 : Double : 40*TEU Container 길이 id : Integer : Container id location : Integer : Container 위치정보전체 bay : Integer : Container 위치정보(bay) row : Integer : Container 위치정보(row0) tier : Integer : Container 위치정보(tier0)
ContainerVehicle	No : Integer : Vehicle 번호 width : Integer : Vehicle 폭 length : Integer : Vehicle 길이 speed : Integer : Vehicle 속도 tasks : containerTaskList : Vehicle 의 Container 의 작업 VehicleTP : Integer : 목적 TP(Transfer Point)
Ship	id : Integer : 선박 ID departureTime : Double : 출항 시간 arrivalTime : Double : 입항시간 delayTime : Double : 접안 지연시간 waitingTime : Double : 대기시간 berth_id : Integer : 선석 번호 tasks : containerTaskList : 선박작업 List
Block	BayNumbers : Integer : Bay 수 RowNumbers : Integer : Row 수 TierNumbers : Integer : Tier 수 Type : Integer : Block Type ContainerLocation : Integer : 컨테이너 위치 정보 InitialContainers : Integer : 최초 컨테이너 수 container : Container : 컨테이너 정보
Crane	InitialBay : Integer : 최초 Bay 위치 StartBay : Integer : 출발 Bay 위치 EndBay : Integer : 도착 Bay 위치 StatsEnabled : Boolean : 작업가능상태 onVehicleMount : code : 이동장비에 실는 작업 중 onVehicleUnmount : code : 이동장비에서 내리는 작업 중 onBlockMount : code : Block에 컨테이너를 적하 중 onBlockUnmount : code : Block에서 컨테이너를 양하 중 onSelectTask : code : 작업 선택 craneShape : ShapeBase : 크레인 모형 next_ct : ContainerTask : 다음 작업 컨테이너 작업 정보 container : Container : 현재 작업 중인 컨테이너 정보 done : Boolean : 작업 마침 next : TrolleyTask : 다음 작업정보
Agent	BayNumbers : Integer : Bay 수 RowNumbers : Integer : Row 수 LoadingTime : real : 적재 시간 UnloadingTime : real : 양하 시간 MoveSpeed : real : 속도 toBlockBay : Integer : 이동 Block Bay toBlockRow : Integer : 이동 Block Row toCraneBay : Integer : Crane 의 Bay 위치 toCraneRow : Integer : Crane 의 Row 위치 selectTask : Integer : 작업 선택 crane : ConCrane : Crane 정보 task : containerTask : 작업 정보 container : Container : 컨테이너 정보 taskList : list : 전체 작업 정보
Gate	Gate_No : char : Gate 번호, Status : char : Gate 상태

Gate_InOut : char : Gate-In, O: Gate_Out 구분
Process_Time : real : 차량 한 대의 처리시간

Table 2 The operation of an object

객체	메소드
ContainerVehicle	getTask(i : Integer) : containerTask : 새로운 작업 추가 onCompleted(task : containerTask) : void : 작업 완료 getContainer() : Container : 새 컨테이너 정보 갱신 getBlock() : ConBlock : Block 위치 정보 갱신 getRow() : Integer : Row 정보 갱신 getBay() : Integer : Bay 정보 갱신 isLoading() : Boolean : 양하 작업 isUnloading() : Boolean : 적하 작업 numOfTasks() : Integer : 총 작업의 수 갱신
Ship	addTask(task : containerTask) : void : 작업 추가 numOfTasks() : Integer : 총 작업의 수 갱신 getTask(i : Integer) : containerTask : 새로운 작업 추가 onCompleted(task : containerTask) : void : 작업 완료 onMount(task : containerTask) : void : 적하 작업 onUnmount(task : containerTask) : void : 양하 작업 getContainer() : Container : 새 컨테이너 정보 갱신
Block	YTIIsStop(vehicle : RVehicle, index : Integer) : Boolean : YT 정지 YTJump(vehicle : RVehicle) : YT Lane 변경 departure(cv : myVehicle) : Vehicle 출발 onArrival(vehicle : RVehicle, index : Integer) : void : Vehicle 도착 onEnterLane(vehicle : GenericVehicle) : void : Block Lane선택 onExitLane(vehicle : GenericVehicle) : void : Block Exit Lane 선택 trailerIsStop(vehicle : RVehicle, index : Integer) : Boolean : Trailer 정지
Crane	isForward() : Boolean : 앞으로 이동 onArrival(t : ContainerTask) : void : Crane 도착 doStartup() : void : 이동 시작 isTrolleyLoading() : Boolean : Trolley 적하 작업 시작 isTrolleyUnloading() : Boolean : Trolley 양하 작업 시작 currentBay() : int : 현재 Bay 위치 확인 isTargetBay() : Boolean : 이동 Bay 확인 currentRow() : int : 현재 Row 확인 targetRow() : int : 이동 Row 확인 moveToBay() : void : Bay 로 이동 moveToRow() : void : Row 로 Trolley 이동 selectTask() : Boolean : 작업 선택 loading() : void : 적하작업 unloading() : void : 양하 작업 onCompleted() : void : 작업완료 add() : void : 작업 추가
Agent	addArrival(ct : containerTask) : void : Vehicle 도착 확인 distanceBay(from : int, to : int) : real : Bay까지의 거리 확인 distanceRow(from : Integer, to : Integer) : real : Row까지의 거리 확인 doneTask(ct : containerTask) : void : 작업 완료 getCrane(task : containerTask) : ConCrane : Crane 할당 getTask(task : containerTask) : ConCrane : 작업 할당 mount(cont : Container) : void : 적하 작업 지시 unmount(b : Integer, r : Integer) : Container : 양하작업 지시
Gate	hold(vehicle : Rvehicle) : void : Vehicle 대기 release(vehicle : Rvehicle) : void : Vehicle 통과

• Agent

Yard 내 혹은 Berth 내에 작업하는 Crane과 Block 사이의 작업을 배정 하는 Controller역할을 하며, YardBlock 혹은 BerthBlock에 양/적하 하려는 Vehicle에 의해 발생한 작업을 우선순위결정 및 해당 Block의 내의 컨테이너 정보들을 관리한다.

• ContainerVehicle

외부로부터 터미널로 들어오는 Trailer 와 내부 Yard 내에서 이동하는 있는 YT로 상속되며 Container의 정보를 가지고 이동하며 장치장내 효율성을 평가하는 객체가 된다..

3.2 Activity Diagram

Activity Diagram은 객체의 프로세스 흐름, 즉 물리적인 흐름을 나타내고 의사결정과정에 의해서 프로세스 흐름의

변화를 나타낸 것이다.

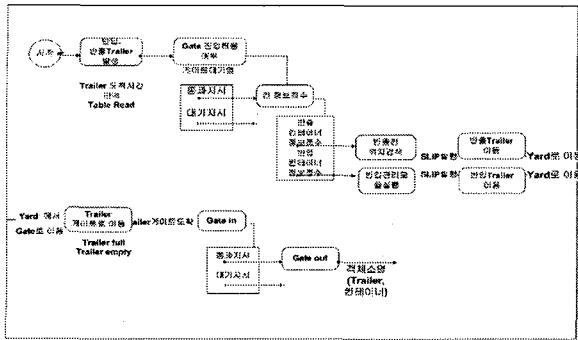


Fig. 3 The activity diagram in a Gate

1) Fig.3 는 Gate에서 반입/반출을 위해 Trailer가 Gate로 진입하여 블록에서 게이트로 퇴출하는 과정에서 발생하는 사건들의 순차적 다이어그램이다.

2) Fig.4 는 장치장에서 TC의 사건들을 중심으로 반입/반출 서비스를 받는 Trailer와 양/적하 서비스를 담당한 YT의 작업을 나타내는 활동 다이어그램이다.

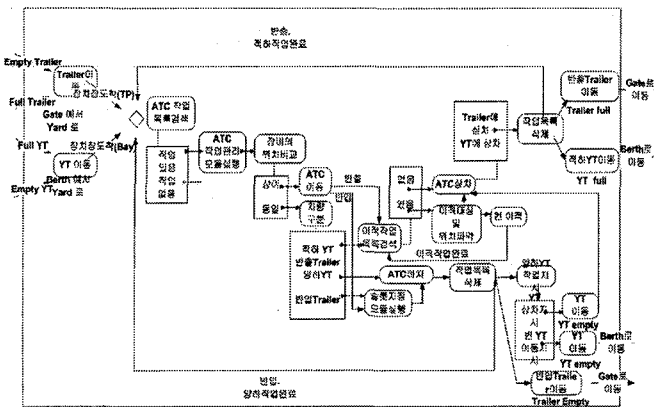


Fig. 4 The activity diagram in a Yard

3) Fig. 5는 선석에서 양/적하 작업을 수행 하는 CC와 YardBlock과 선석을 오가며 컨테이너를 운반하는 YT의 활동 다이어그램이다.

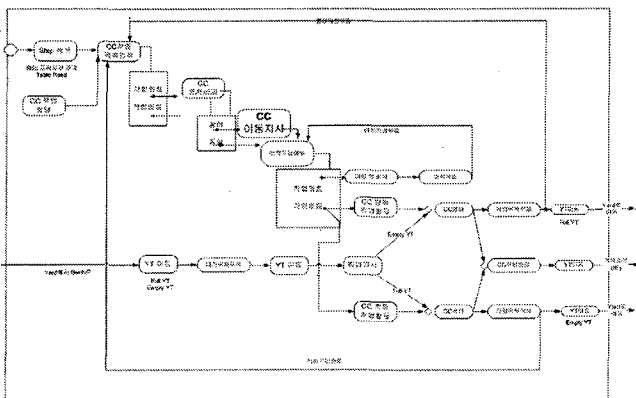
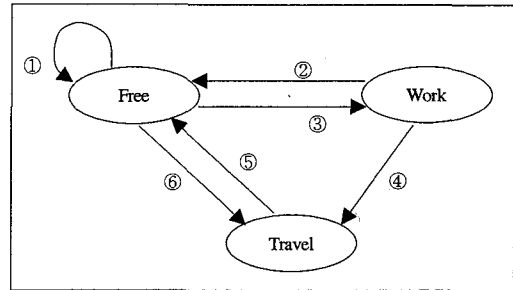


Fig. 5 The activity diagram in Berth

3.3 State Transition Diagram

상태 전이 다이어그램은 각 객체의 사건의 변화를 도식화하여 표현한 것이다. 이벤트로 인하여 객체의 상태의 변화를 나타내며 평가 지표로 결과값에 나타난다.

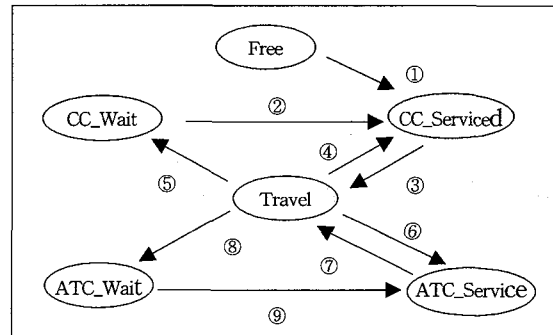
Fig 6은 TC가 작업지시를 받으면 목적지로 이동하여 작업을 하고 작업이 없으면 대기(Free) 상태로 변화는 과정이다.



- ①: TC_Work_End[반/출입 완료, 양/적하 완료, 동일 bay]
- ②: TC_Work_Starts[반/출입 Trailer도착, 양/적하 YT도착]
- ③: Work_List_Empty
- ④: TC_Travel_Starts[반/출입 Trailer 도착, 양/적하 YT 도착, 동일 bay아님]
- ⑤: TC_Travel_End
- ⑥: TC_Travel_Starts[반/출입 완료, 양/적하 완료, 동일 bay 아님]

Fig. 6 The state transition diagram of a YT

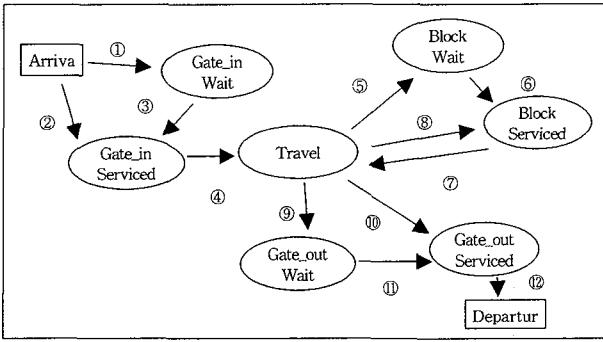
Fig. 7은 CC가 작업지시를 받으면 해당 현재의 크레인 위치를 확인하고 YT의 도착 유무에 따라 작업 상태가 바뀐다.



- ①: Ship_Arrival
- ②: CC_Read
- ③: CC_Service_End
- ④: Shipbay_Arrival[CC 도착]
- ⑤: Shipbay_Arrival[CC 미도착]
- ⑥: Yardbay_Arrival[TC 도착]
- ⑦: TC_Service_End
- ⑧: Yardbay_Arrival[TC 미도착]
- ⑨: [TC ready]

Fig. 7 The state transition diagram of a CC

Fig 8은 반출/입 발생정보에 의해 차량이 발생되고 Gate를 통과하여 Block으로 이동하고 TC의 서비스를 받고 다시 Gate로 이동한 후 Gate를 빠져 나가는 과정에서의 Trailer 상태변화를 나타낸다.



- ①: Gate_in_Arrival[빈 Gate 없음]
- ②: Gate_in_Arrival
- ③: Gate_in_Empty
- ④: Gate_in_Service_End
- ⑤: Block_TP_Arrival[TC 미도착]
- ⑥: Block_TP_Arrival[TC 도착]
- ⑦: TC_Service_End
- ⑧: Block_TP_Arrival[TC 도착]
- ⑨: Gate_out_Arrival[빈 Gate 없음]
- ⑩: Gate_out_Arrival
- ⑪: Gate_out_Empty
- ⑫: Gate_out_Service_End

Fig. 8 The State Transition Diagram of a Trailer

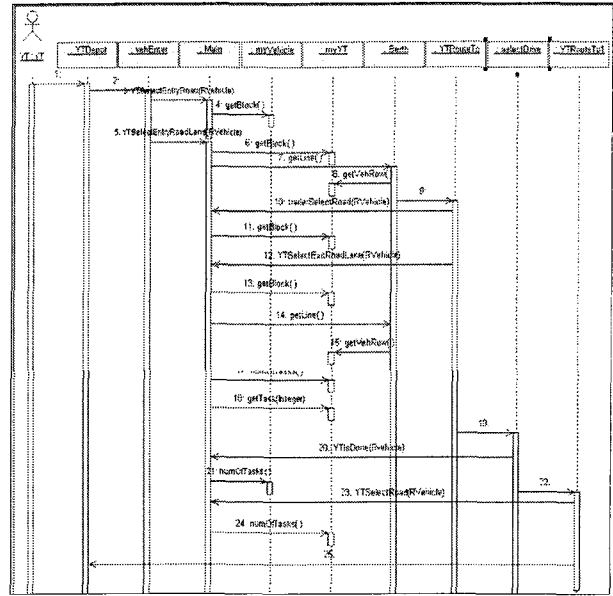


Fig. 10 The sequence diagram of a YT

3.4 Sequence Diagram

시퀀스 다이어그램은 시간의 흐름에 따라 객체의 상호 작용을 나타낸 다이어그램이다. 이는 시간의 흐름에 따라 진행되는 사항을 나타내므로 Trailer(Fig. 9), YT(Fig 10)의 시간에 따른 객체 상호간의 흐름을 나타내었다.

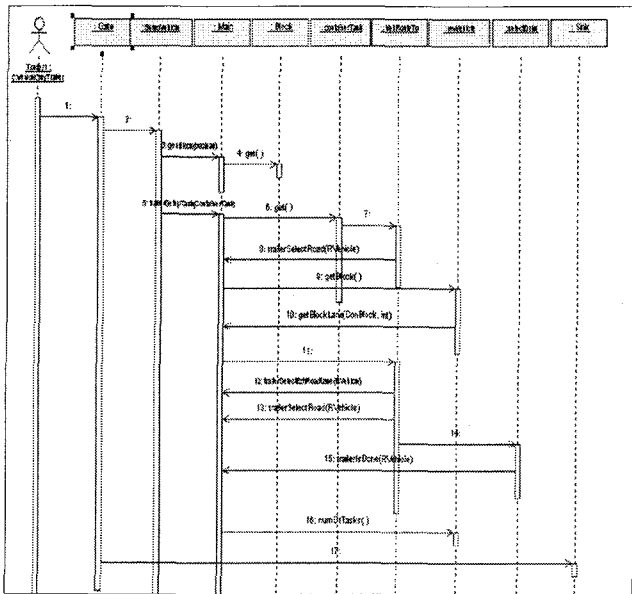


Fig. 9 The Sequence Diagram of a Trailer

4. 컨테이너 터미널의 입, 출력 설계

입력 값은 장비, 시설 그리고 운영에 필요한 값으로 개발된 시뮬레이터가 시뮬레이션 하고자하는 터미널의 입력 정보로써 사용자에게 따라 달라질 수 있다.

출력 값은 컨테이너 내부의 장비와 시설 등의 시간에 관한 데이터로 운영 시스템을 평가하기 위해 시뮬레이터로부터 얻고자하는 결과 값이다.

4.1 시뮬레이션 입력 설계

사용자 입력 요소(Table 3)는 컨테이너 부분, 운용, 장비 및 시설 부분으로 구분되고 장비부분은 CC, TC, YT로 세분화 되고 시설부분은 장치장, 선석, 그리고 Gate 부분으로 구분된다. 수출, 입 컨테이너 양, 컨테이너 반입 시작, 마감시간 할당 수 등 현재 터미널이 사용하고 있는 수치적 값의 입력뿐만 아니라 정책적인 변화에 의한 입력값을 말한다.

Table 3 Total input data list

구분	내역	구분	내역	
장비	CC	컨테이너 데이터	수입 컨테이너 양	
			장비 대수	TEU/Van 비
			작업 시간	반입 시작 시간
	TC		장비 대수	반입 마감 시간
			운행 속도	무료 장치 일수
			작업 시간	선석당 CC 할당 수
YT	장비 대수			
	운행 속도			
시설	장치장 Block	운용	CC당 YT 할당 수	
			열 수	배 도착 시간
	단 수		평균 집안 시간	
	선석개수		선석당 CC 할당 수	
Gate	선석	Gate	Block당 TC 할당 수	
			총 처리 선박대수	
			선석과 장치장 거리	
Gate	Gate	Gate	선석의 길이	
			입구 개수	
			출구 개수	
			Gate별 서비스 시간	

4.2 시뮬레이션 출력 설계

시뮬레이션 출력 결과(Table 4)는 장비, 고객 및 시설로 나뉘어 제공되었고 장비 생산성, 가동률, 이동시간, 대기 시간이 있으며 이동장비는 장치장 및 선석에서 대기시간과 서비스 시간, 그리고 선회 시간 등이 있다.

이러한 출력 결과는 시뮬레이터에 직접적으로 생산되는 결과 값과 시뮬레이터에 의해 구해진 1차적 데이터를 이용하여

산출적 계산으로 얻어지는 값이다.

Table 4 Total output data list

구분	구분	통계량	산출방법	
장비	CC	장비생산성	총 처리물동량/총 사용 시간	
		순 장비 생산성	총 처리 물동량/총 작업시간	
		실제 장비 능력	장비 당 처리 물동량/제원상의 처리능력	
		장비 가동률	순 작업시간/(연간작업일수*24h)	
		순 작업 시간 비율	순 작업 시간/전체 시뮬레이션 시간	
		이동시간 비율	이동시간/전체 시뮬레이션 시간	
		이동 지연 시간 비율	이동 중 지연시간/이동시간	
		서비스대기시간 비율	대기시간/전체 시뮬레이션 시간	
	TC	장비생산성	총 처리물동량/총 사용 시간	
		순 장비 생산성	총 처리 물동량/총 작업시간	
		실제 장비 능력	장비 당 처리 물동량/제원상의 처리능력	
		장비 가동률	순 작업시간/(연간작업일수*24h)	
		순 작업 시간 비율	순 작업 시간/전체 시뮬레이션 시간	
		이동시간 비율	이동시간/전체 시뮬레이션 시간	
		이동 지연 시간 비율	대기시간/전체 시뮬레이션 시간	
		서비스대기시간 비율	(이동+대기+작업)시간/전체 시뮬레이션 시간	
	YT	평균선회시간	CC to CC, TC to TC 사이의 걸린 시간	
		평균 선회 수	하루당 Turn-around 회수	
		평균 TC 서비스 대기시간	TC의 TP도착시간-TC작업시작시간	
		평균 CC 서비스 대기시간	CC의 TP도착시간-CC작업시작시간	
	고객	Trailer	평균 선회 시간	Gate-in time-Gate-out Time
			평균 CC 서비스 대기시간	TC의 TP도착시간-TC작업시작시간
			평균 GATE 서비스 대기시간	Gate에 진입시간-Gate-in Time
		선박	평균대기 시간	접안시간-선석이 할당되는 시간
			서비스 지연시간	첫 컨테이너 적하/양하 시작시간-끝나는 시간
			평균 선회시간	입항시간-출항시간
	시설	선석	선석점유율	선박접안시간/(연간작업일수*24h*선석수)
			총선석 생산성	총 처리 물동량/총 접안시간
순 선석 생산성			총 처리 물동량/ 총 작업시간	
평균 컨테이너처리시간			직하시작시간-지정 YT 작업완료시간	
장치장		장치장 평균 점유율	평균 장치량(TEU)/Yard 면적 *100	
		야드 번적당 처리실적	연간 처리 물동량(TEU)/Yard 면적 *100	
		평균 야드 장치일수	평균 장치 시간/평균 장치 물량	
Gate		Gate-in 서비스지연시간	도착시간 - Gate-in시간(TP시간 포함)	
		Gate-in 순서 대기시간	도착시간 - Gate-in 도착 알림 시간	
		Gate-out 서비스지연시간	도착시간 - Gate-out시간(TP시간 포함)	
		Gate-out 순서 대기시간	도착시간 - Gate-out 도착 알림 시간	

5. 결론 및 추후연구

본 논문에서 객체지향 기법을 이용하여 컨테이너 터미널 내부에 Crane의 활동과 Trailer, YT, Ship 과 같이 컨테이너를 운송하는 활동 등 수 많은 활동들이 복잡하게 행하여지는 시스템의 설계 방법을 나타낸다. 제안된 설계 방법을 이

용하여 객체지향기법을 가장 잘 반영하는 Java 기반의 시뮬레이션 툴 이용해서 구현할 수 있다. 구현된 시뮬레이터는 컨테이너 터미널의 운영 시스템을 평가하거나 최고의 운영 정책을 찾는 데 사용되고 시뮬레이터로부터 얻어지는 결과값으로 운영시스템을 평가 할 것이다.

시뮬레이션으로 얻은 결과값으로 전체의 운영 시스템을 평가가 가능지만 내부적으로 양/적하 계획, 선석계획, 장치장 계획, 그리고 실시간 운영등 세부적인 부분 모두를 평가할 수 없다. 또한 구체적인 문제점이나 개선책을 찾기 어렵다. 그러므로 개발된 시뮬레이터로 얻어진 1차 데이터를 이용해서 세부적인 정책이나 실시간 운영시스템을 구체적으로 평가할 수 있도록 결과값과 일력값과의 관계 혹은 개별 결과값이 세부적 운영시스템에 미치는 영향등의 연구가 필요하다.

후 기

본 연구는 해양수산부의 '지능형 항만물류 시스템 기술개발' 사업의 일환으로 수행되었습니다.

참 고 문 헌

- [1] 광용저, 오종범(2004), "초보자를 위한 UML 객체지향 설계", 정보 문화사
- [2] 서운준(2004), "UML, ROSE, RUP, 객체지향 분석설계 그리고 소설같은 실전 프로젝트 이야기", 가남사
- [3] 장성용, 박진우(1988), "시뮬레이션 기법을 이용한 컨테이너 터미널의 운영시스템 결정", 산업공학, 제1권, 제1호, pp.49-62
- [4] 장성용, 이원영(2002), "시뮬레이션 기법을 이용한 컨테이너 터미널의 운영계획 평가", 한국 항만 경제 학회, Vol 18,
- [5] A.E.Rozzoli, L.M. Gambardella, M. Zaffalon, M. Mastrolilli(2004), "Simulation for the Evaluation of Optimised Operations Policies in a Container Terminal", HMS99, Maritime & Industrial Logistics Modeling and Simulation, Genova, Italy, 16-18th
- [6] Maurizion Bizelli, Azedine Boulmakoul, Mohamed Rida(2005), "Object Oriented Model for Container Terminal Distributed Simulation", European Journal of Operational Research, Vol, 175, pp, 1731-1751
- [7] Won_Young Yun, Yong_Seok Choi(1999), "A Simulation Model for Container-Terminal Operation Analysis using an Object-Oriented Approach", International Journal of Production Economics, Vol. 59 Issue 1-3, p221-230
- [8] Yong_Seok Choi, Tae_Young Ha(2004), "Simulation System for Port Container Terminal Using An Objected Oriented Approach", International Journal of Navigation and Port Research, Vol.28, No.9 pp. 791-798