

객체지향접근법을 사용한 컨테이너 터미널 시뮬레이터의 설계†

윤원영¹ · 최용석¹ · 이명길² · 송진영³

¹부산대학교 산업공학과 / ²(주)태평양화학 / ³(주)토탈소프트뱅크

Design for Container Terminal Simulator Using an Object-oriented Approach

·Won-Young Yun¹ · Yong-Seok Choi¹ · Myung-Gil Lee² · Jin-Young Song³

This paper proposes a design procedure to develop the object-oriented simulator of port container terminal. The design methodology uses an object-oriented approach to support an object-oriented simulation and the design procedure consists of object scheme and event scheme. The object-scheme is a procedure to determine the structure of material flow objects and information flow objects and a relation diagram between objects that have attributes and methods. The event scheme is a procedure to define methods and to connect messages of objects. We assume that the container terminal system consists of gate, container yard, and berth and the equipment used in the container terminal are container cranes, transfer cranes, yard tractors, and trailers.

1. 서론

우리 나라의 경제규모가 확대되고 무역량이 급격히 증가함으로써 해상물 통한 컨테이너 운송이 크게 증가하고 있다. 따라서 이를 지원하기 위한 기반시설인 항만의 시설확충 및 운영 효율화에 대한 필요성이 증대되고 있다. 국내의 주요 컨테이너항인 부산항의 컨테이너 취급물량은 1999년 631만 TEU (Twenty Equivalent Unit)의 컨테이너를 하역함으로써 컨테이너 처리실적면에서 세계 5위의 자리를 차지하고 있으며, 컨테이너 취급물량은 앞으로도 지속적으로 증가할 것으로 전망하고 있다. 이에 따라 컨테이너를 신속하고 원활하게 하역, 보관할 수 있는 컨테이너 터미널의 개발 및 확충이 진행중에 있다.

이러한 개발과 확충은 고도의 기술적 장비 및 인력을 확보함으로써 생산성의 향상과 함께 선사, 화주 등의 항만이용자에게 질 높은 서비스를 제공하는 것을 주목적으로 하고 있다. 또한 이러한 주목적을 달성하기 위해서 컨테이너 터미널의 구조, 장비, 정보시스템 등의 향상을 지속적으로 도모하며, 터미

널의 운영절차를 개선시키려고 노력한다. 그러나 컨테이너 터미널의 운영절차를 분석하거나 설계하는 문제는 현장에서 직접 시도하여 그 결과를 분석하기에는 시간과 비용이 많이 소요되므로 이를 효과적으로 행하기 위하여 시뮬레이션 방법을 이용한 연구가 경제적인 것으로 알려져 있다.

이 분야의 국내의 연구는 조덕운(1985)이 GASP-IV를 이용하여 컨테이너 터미널의 시뮬레이션 모델을 개발한 이후 지속적으로 연구되어 왔으며, 최근에는 윤원영, 최용석(1997)에 의해 객체지향접근법을 이용한 컨테이너 터미널 시뮬레이션 모델과 윤원영, 안창근, 최용석, 김갑환(1998)이 컨테이너 터미널의 운영계획 평가용 시뮬레이션 시스템을 개발하여 시뮬레이션 모델이 시스템화되어 가고 있으며, 그 개발 방법에서도 설계와 모델링 단계에서 객체지향접근법을 도입하고 있다.

컨테이너 터미널 시뮬레이터에 대한 국외의 연구를 살펴보면, 현대식 컨테이너 터미널을 포함한 현대식 항만을 SLAM으로 모델링한 시뮬레이터가 Park and Noh(1987)에 의해서 개발된 후 이스라엘의 컨테이너 터미널을 이산사건 시뮬레이션 방에 따라 C언어로 시뮬레이터를 개발한 Hayuth, et al.(1994)의 연

† 본 연구는 한국과학재단 지정 동아대학교 지능형통합항만관리 연구센터의 지원에 의한 것임.

구가 시뮬레이터 개발에 대한 방법론을 제시하고 있다. 국내의 컨테이너 터미널의 운영방식과 유사한 현대화된 컨테이너 터미널을 상세하게 모델링 한 Ramani(1996)의 연구가 있었다. Thiers and Janssens(1998)은 최근의 시뮬레이터 개발 형식론의 하나인 DEVS Scheme을 이용하여 벨기에의 Antwerp 항에 대한 시뮬레이터를 개발하였으며, 최근에는 객체지향 시뮬레이터를 개발하여 컨테이너 터미널의 분석에 적용하려는 연구가 이루어지고 있으나 개발 도구가 MODSIM, GPSS/H 등 시뮬레이션 소프트웨어 위주로 이루어지고 있으며(Gambardella, et al., 1998; Merkurjev, et al., 1998; Nevins, et al., 1998; Nevins, et al., 1998; Raczynski, 1988; Silberholz, et al., 1991), 객체지향 시뮬레이션 소프트웨어인 SIMPLE++를 이용한 객체지향 컨테이너 터미널 시뮬레이션 모델도 개발되었다(Yun and Choi, 1999).

지금까지의 시뮬레이터 개발 방향을 요약하자면, 터미널 내의 시설물의 성능분석과 터미널 계획정책을 평가하려는 거시적 운영수준을 다룬 연구와 개별장비의 할당과 세부적 운영정책을 다루는 미시적 운영수준을 다룬 연구로 대별될 수 있다. 또한 시뮬레이션 소프트웨어의 발전으로 인해 사용자 인터페이스와 상세한 출력 통계량을 제공하고 있으며, 시뮬레이션 모델의 신속한 유효성 확인(verification)과 검증(validation)을 위해 애니메이션이 강조되고 있다. 개발 방법론에서는 재사용성(reusability)과 확장성(extensibility)을 제공하기 위한 객체지향 시뮬레이션이 보편화되어 가고 있다.

따라서, 컨테이너 터미널을 시뮬레이션하기 위해서는 시스템 변화에 신속하고 쉽게 변경될 수 있는 시뮬레이션 모델을 구축하기 위한 강건한(robust) 시뮬레이션 도구가 필요하므로 본 연구에서는 시스템의 변경을 쉽고 신속하게 처리하기 위한 재사용성과 모듈가능성을 제공하는 객체지향접근법을 이용한 시뮬레이터 설계과정을 다루고자 한다. 그리고 시뮬레이션 시스템의 활용을 성공적이고 경제적으로 달성하기 위해서 실세계 시스템의 모델링에 더 유연한 시뮬레이션과 모델링 과정이 필요하므로 설계단계에서 시뮬레이터의 입력과 출력요소를 고려하여 재사용성과 모듈가능성을 제공하는 객체지향 모델링 과정을 설명하고자 한다. 컨테이너 터미널 시스템의 시뮬레이션 모델링 과정은 Martin and Odell(1992)의 객체지향 설계방법을 활용하여 설계하였다.

본 연구는 객체지향접근법을 사용하여 컨테이너 터미널 시뮬레이터를 개발하는 과정에서 선행되는 객체지향 시뮬레이션 모델의 설계를 목적으로 하고 있으며, 모델링 과정을 객체지향분석과 객체지향설계를 통합한 객체지향접근법을 적용하고자 한다. 이하 2절에서는 컨테이너 터미널 시스템의 운영과 계획을 분석한 후 시뮬레이터의 입출력 요구사항을 제시하며, 3절에서는 객체지향 시뮬레이션 방법을 소개할 것이다. 4절에서는 컨테이너 터미널 시스템의 객체지향 분석 및 설계를 다루며, 5절에서는 시스템의 구조와 로직을 예로 들며, 6절에서는 결론을 요약한다.

2. 컨테이너 터미널 시스템 분석 및 사양

2.1 대상 시스템 분석

컨테이너 터미널 시스템에 대한 시뮬레이션 모델을 설계하기 위하여 우선 시스템의 운영을 분석하는 것이 선행되어야 한다. 일반적으로 컨테이너 터미널이란 컨테이너의 수송에 활용되는 항만에 존재하며, 컨테이너의 운송과 저장을 위한 터미널이라고 인식되며, 이러한 컨테이너 터미널의 가장 일반화된 유형을 Ramani(1996)의 연구에서 잘 설명되고 있으며, 컨테이너 터미널의 기능을 컨테이너 수송목적지에 따라 수입기능과 수출기능으로 구분하며, 장비운영과 보관측면에서 수출은 반입작업과 적하작업으로 수입은 반출과 양하작업으로 분류한다.

컨테이너 터미널 시스템의 유형은 컨테이너 취급방식에 따라 온-차시 시스템(on-chassis system), 스트래들 캐리어 시스템(straddle carrier system), 트랜스퍼 크레인 시스템(transfer crane system)으로 구분하는 것이 가장 일반적이며, 이러한 구분기준은 주로 장치장의 사용장비에 따르며, 스트래들 캐리어와 트랜스퍼 크레인이 가장 일반적인 장비로 사용되고 있다. 현재 부산항의 경우 현대 부산 컨테이너 터미널(HBCT)에서 부분적으로 스트래들 캐리어 시스템을 사용하고, 신선대부두(PECT), 우암터미널(UTC), 감만부두(KTC) 등에서는 트랜스퍼 크레인 시스템을 사용하고 있다.

특히 최근의 컨테이너 터미널의 장비 운영방식이 Transfer Crane(TC)-Container Crane(CC)-Yard Truck(YT) 체제를 따르고 있으므로 전형적인 TC-CC-YT 체제로 운영되고 있는 컨테이너 터미널 시스템을 연구 대상으로 하여 트랜스퍼 크레인 시스템 체제의 컨테이너 터미널로 일반화하였다.

컨테이너 터미널 시스템의 주요 시설물은 크게 게이트(gate), 장치장(container yard), 선석(berth)의 세 부분으로 구성되어 있으며, 사용되는 장비로는 장치장에서 사용되는 트랜스퍼 크레인, 선석에서 사용되는 컨테이너 크레인, 장치장과 선석간을 이동하는 야드 트럭, 그리고 터미널 외부에서 게이트를 통과하여 장치장과 게이트간을 이동하는 외부트럭 등이 있으며, 이를 그림으로 표현하면 <그림 1>과 같다.

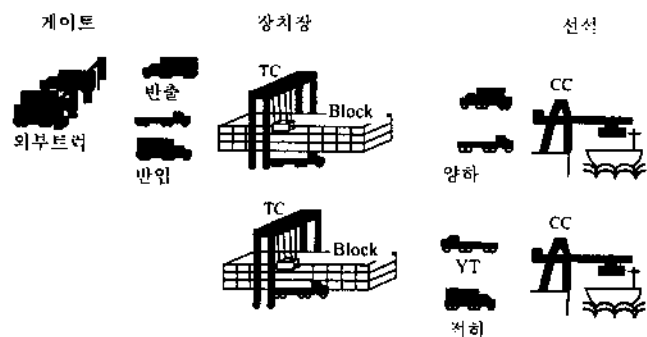


그림 1. TC체제의 컨테이너 터미널 시스템.

<그림 1>과 같이 컨테이너 터미널에서의 작업 유형은 크게 반입(receiving), 반출(delivery), 양하(unloading or discharging), 적하(loading or charging)의 네 가지 유형으로 나눌 수 있으며, 이러한 작업에 CC, TC와 같은 컨테이너 취급장비와 YT, 외부트럭 등의 컨테이너 운송장비가 사용되며, 이러한 장비들의 운영은 네 가지 작업 유형에 따라 다음과 같이 서로 연계되어 있다(윤원영, 안창근, 최용석, 김갑환, 1998).

(1) 반입작업은 수출용 컨테이너를 실은 외부트럭이 게이트에 도착하여 본선적하계획에서 결정된 컨테이너의 장치위치를 받아서 그 위치로 이동하여 TC가 외부트럭에 실린 컨테이너를 장치위치에 옮겨 놓으면, 빈 외부트럭은 게이트를 빠져나간다. 이때 사용장비는 TC와 외부트럭이다.

(2) 반출작업은 빈 외부트럭이 수입된 컨테이너를 실기 위해 게이트에 도착하여 수입장치장계획에서 반출위치를 받아서 작업위치로 이동한 후 TC가 컨테이너를 외부트럭에 옮겨 실으면 수입 컨테이너를 실은 외부트럭은 게이트를 빠져나가며, 이 경우에도 사용장비는 TC와 외부트럭이다.

(3) 양하작업은 양하계획에서 작성된 CC의 작업순서목록(sequence list)에 따라 선박에 실린 수입 컨테이너를 CC가 YT에 옮겨 실으면, YT는 수입 컨테이너의 장치위치로 이동하고 TC가 컨테이너를 장치위치에 옮겨 놓으면, YT는 다시 할당된 CC의 작업장소로 이동한다. 사용장비는 TC, CC, YT가 사용된다.

(4) 적하작업은 YT가 적하계획에서 작성된 수출 컨테이너의 장치위치로 이동하고 TC가 수출 컨테이너를 YT에 옮겨 실으면, YT가 할당된 CC의 작업장소로 이동하며, CC는 작업순서목록의 순서에 따라 컨테이너를 선박에 실게 된다. 이 경우의 사용장비는 양하작업과 같이 TC, CC, YT이다.

위에서와 같이 컨테이너 터미널의 운영절차는 복잡한 것이 일반적이며 이에 대한 표준적인 운영절차는 사실상 없다고 해도 과언이 아니다. 다만 컨테이너 터미널의 작업을 능률적이고 효율적으로 처리하기 위하여 사전에 치밀한 작업계획이 수립되어야 함은 물론 부문별로 유기적인 협조체계가 필요하기 때문에 이에 대한 표준적인 작업절차를 확립하는 것이 필수적이다. 따라서 본 연구에서는 컨테이너 터미널의 운영을 위해서 분류한 네 가지 작업유형으로 하고 운영에 필요한 계획으로는 본선작업계획과 장치장계획 등이 고려되어야 한다. 따라서 선석계획, 본선양하계획, 본선적하계획, CC 작업계획 등의 본선작업계획과 수출장치장계획과 수입장치장계획을 포함한 장치장계획이 운영을 위하여 필요하다. 이들의 역할을 정의하면 다음 <표 1>과 같다.

<표 1>의 세부계획들은 실제 컨테이너 터미널의 운영에 사용되는 정보들이며, 시뮬레이션에서도 운영을 묘사하기 위해서 계획들을 작성하고 갱신하는 정보의 처리가 시뮬레이션의 실행에 기초정보가 된다. 따라서 시뮬레이션에서도 운영에 필요한 계획을 작성하고 계획의 내용을 정보화하여 관리하는 부분을 데이터베이스로 구현하여야 운영을 더 상세히 묘사할 수 있다.

표 1. 컨테이너 터미널 운영에 관련된 작업계획

계획 구분	세부 계획	계획 내용
본선 작업계획	선석계획	입항선박의 선석할당, 본선하역 장비 할당 및 작업계획
	본선양하 계획	입항선박에 대한 CC의 컨테이너 양하작업순서 결정
	본선적하 계획	입항선박에 대해서 장치장의 컨테이너 정보를 참조하여 CC의 컨테이너 적하작업순서 결정
	CC작업계획	본선작업에 투입된 1~3대의 CC에 대한 작업영역 및 작업순서 결정
장치장 계획	수출장치장 계획	수출 컨테이너의 장치공간 할당
	수입장치장 계획	수입 컨테이너의 장치공간 할당

2.2 시뮬레이터 요구사항

컨테이너 터미널 시뮬레이터는 사용목적, 복잡성, 상세도, 고려되는 요소에 따라 매우 다양하게 나타난다. 즉 목적을 계획 관점인지, 운영 관점인지 혹은 실제 운영의 분석인지에 따라 시뮬레이터의 개발방향이 달라진다(Hayuth, et al., 1994). 컨테이너 터미널의 계획 관점에서는 두 가지 유형으로 성능평가가 이루어지며, 첫 번째가 터미널 후보지의 치형에 대해서 선석 길이와 장치장 면적이 주어질 경우 터미널에서 어느 정도의 연간 취급량을 기대할 수 있는가를 구하는 경우이고 두 번째는 항만건설이나 확충계획을 입안하여 대상 컨테이너 터미널에 어느 정도의 선석 길이, 장치장 면적을 부여할 것인가를 검토하는 경우이다. 이 두 가지 유형의 성능평가가 기본계획 단계이며, 이 단계가 이루어지면 이 기본계획을 바탕으로 상세한 계획 및 설계를 행하여 터미널 고유의 운영정책을 반영하는 운영관점까지 고려하게 된다. 따라서 기본계획단계가 주어진 경우 상세한 계획 및 설계를 지원할 수 있는 기능을 제공하는 것이 시뮬레이터의 요구사항이 된다.

그러나 계획단계에서부터 고려될 수 있는 가능한 요소는 물리적인 항만시설물과 화물, 선박, 작업자, 운영자, 내륙수송 등이 있다. 본 연구에서의 목적은 기존 컨테이너 터미널의 성능평가와 새로운 컨테이너 터미널 개발시에 설계단계에서 성능분석 및 대안평가에 주안점을 두었다. 따라서 성능평가를 위해 다양한 통계량을 제공하여야 하며, 대안평가를 위해 기본 시스템의 사양에 대한 입력 외에 사용자의 대안에 대한 입력을 최대한 허용하도록 하여야 한다.

컨테이너 터미널 시뮬레이터에서 입력으로 고려되는 항목은 크게 연간 컨테이너 물동량정보, 시설물정보, 사용장비정보, 운영정책정보 등으로 나눌 수 있다. 이를 항목별로 상세히 나누면 <표 2>의 입출력 항목표와 같이 세분된다. 이들 입력 정보들 중 운영정책정보를 제외한 정보들로 컨테이너 터미널의 운영 묘사가 가능하여야 하며, 추가로 사용자에 의해 운영

표 2. 입출력 항목과 내용

입력 항목		입력 내용	출력 항목	출력 내용	
컨테이너 물량 정보 (TEU/연간)		수출 컨테이너 물량	시설물 통계량	총처리 선박 대수	
		수입 컨테이너 물량		선박 평균 선석점유 시간	
		환적 컨테이너 물량		총처리 컨테이너 개수	
		20', 40' 물량비율		컨테이너 평균 처리시간	
시설물 정보	게이트	게이트 입/출구 수		평균 선석 점유율	
		게이트 입/출구 서비스시간		선석의 서비스 지연시간	
		게이트와 장치장간의 거리		장치장 점유율	
	장치장	장치장배치, 블록의 사양		장비 통계량	TC 평균 활용도
		선석별 수입/수출 장치블록 할당			TC의 작업, 이동, 대기, 유티 시간비율
	선석	선석수, 연간 입항선박수			CC 평균 활용도
선석과 장치장간 거리		CC의 작업, 이동, 대기, 유티 시간비율			
장비정보		TC대수, 이동속도, 작업시간	YT의 평균 활용도		
		CC대수, 이동속도, 작업시간	YT 선회(turnaround)시간		
		YT대수, 이동속도	외부트럭 서비스시간		
		외부트럭 이동속도			
운영정책		반입 시작시간			
		반입 마감시간			
		수입 컨테이너 무료장치기간			
		선석당 CC 할당 대수			
		CC당 YT 할당 대수			
	시뮬레이션 준비시간				

정책 정보를 가질 경우 특정 터미널의 상황을 반영한 운영 대안을 수립하여 평가할 수 있다. 즉, 컨테이너 터미널 시스템을 구성하기 위해 필요한 기본입력과 시뮬레이션을 위한 입력, 시스템 성능평가를 위한 입력-등이-사용자에 의해 주어질 수 있다.

출력의 경우 시스템의 시설물인 게이트, 장치장, 선석의 기본 통계량을 구하고, TC, CC, YT, 외부트럭 등의 장비에 대한 장비 활용도의 통계량이 구해져야 한다. 또한 시뮬레이션 실험을 위한 준비시간과 실행시간 길이 등도 사용자가 결정할 수 있도록 하여야 한다.

3. 객체지향 시뮬레이션 방법

객체지향 시뮬레이션(object-oriented simulation)은 객체지향접근법(object-oriented approach)에 기초한 도구(tool)이며, 본 연구에서 객체지향접근법은 객체지향분석(object-oriented analysis)과 객체지향설계(object-oriented design)를 통합한 방법으로 정의하였다. 객체지향분석은 실제 시스템을 분석하여 개념적인 모델을 구축하는 과정이며, 객체지향설계는 개념적인 모델을 시뮬레이션 모델로 변환하는 과정이다.

객체지향접근법은 프로그래머와 분석가가 초기에 생성된 모델을 재사용할 수 있게 하여 복잡한 모델을 만들어 낼 수 있

도록 데이터의 추상화(abstraction), 캡슐화(encapsulation), 정보은닉(information hiding), 상속성(inheritance), 동적결합(dynamic binding) 등을 제공하므로 프로그래밍과 모델링에 아주 유용한 것으로 알려지고 있다(Martin and Odell, 1992).

특히 데이터 추상화와 캡슐화는 모듈화를 달성하기 위해 객체지향접근법에서 사용되는 개념이며, 상속성과 동적결합은 재사용성을 확보하는 방법이다. 따라서 컨테이너 터미널과 같이 여러 종류의 장비가 상호작용하는 복잡한 시스템을 시뮬레이션 하기 위해서 객체지향접근법에서 제공되는 모듈화와 재사용성의 두 가지 특성을 이용하는 것이 효과적이다. 또한 객체지향 시뮬레이션은 언어적 시뮬레이션(linguistic simulation)과 모수적 시뮬레이션(parametric simulation)과 같은 알고리즘 시뮬레이션(algorithm simulation)보다 우월한 장점인 모듈적 시뮬레이션(modular simulation)과 일반적 시뮬레이션(generic simulation)이라는 점이다. 모듈적 시뮬레이션은 객체들의 독립이 모델을 재구성을 하지 않고도 객체의 추가와 삭제로 모델의 변환을 쉽게 할 수 있도록 하는 것이며, 일반적 시뮬레이션은 고수준의 추상화로 모델을 구축하는 능력이 모델의 개발자가 초기에 일반적이고 정교한 모델을 구축할 수 있게 해주어 특정의 사양을 부수적으로 사례화(instance)할 수 있게 해준다. 그리고 모델링의 일반적 단위인 클래스(class)와 슈퍼 클래스(super class)가 저장된 서브모델을 구축하기 위해 통합화될 수 있으며, 더 복잡한 모델로도 구축이 가능하다.

모델링 목적을 위해서 각 객체들은 실세계의 대응물(counterpart)과 유사한 속성을 가진 클래스로서 생성되며, 클래스들의 집합(class library)에 정의된 객체들을 사용함으로써 효과적인 통신(communication)을 위한 객체들간의 메시지(message)의 수신과 송신이 복잡성을 관리하기 위한 장점들을 제공하며, 메시지를 통하여 객체들간에 영향을 줄 수 있도록 하여 객체의 자율권을 부여한다.

특히, 객체지향접근법을 사용하여 컨테이너 터미널 시뮬레이션 설계를 하는 것이 타당한 이유는 다음과 같다.

- 1) 객체와 실물 개체의 대응 : 시뮬레이션 모델 내의 객체와 실제 시스템의 개체간에 일대일 대응관계를 확립할 수 있다. 이러한 대응관계를 통해서 실제 사용되는 장비인 TC, CC, YT, 외부트럭 등에 대응되는 운반장비에 대한 클래스를 설계할 수 있다. 또한 모델 내의 객체 단위와 실제 시스템의 요소들간의 대응관계도 분석과 설계단계에서 고려된다면 더욱 직관적인 모델링이 가능하며, 이러한 대응관계가 모델에 기초한 사고 과정을 더 신속하고 효과적으로 만든다.
- 2) 모델의 계층과 시스템 계층의 대응 : 시뮬레이션 모델과 실제 시스템간의 여러 수준의 사상(mapping)이 구조적인 수준을 달성하게 한다. 이러한 계층의 명확한 표현이 모델링단계에서 중요하며, 여러 수준의 계층을 구조화할 수 있다. 또한 계층적 모델링이 추상화되어 있는 다양한 단계의 시스템 모델을 관찰하고 처리하는 데 유리하다.

시뮬레이션의 설계단계에서 고려되는 모형은 Mesarovic and Takahara(1975)가 제안한 일반 동적 시스템 모형(General System Specification: GSS)을 사용하며, 이 모형에서는 연속상태변화형

시스템뿐만 아니라 이산사건형 시스템도 모델링할 수 있으며, 이를 통하여 일반적 시뮬레이션을 가능하게 한다.

$$S = \langle A, B, C, \varphi, \lambda \rangle$$

$$\varphi : C \times A \rightarrow C$$

$$\lambda : C \times A \rightarrow B$$

여기서 A는 입력, B는 출력, C는 시스템의 상태를 나타내는 집합이며, φ 는 상태전이함수, λ 는 출력함수를 표시한다. 상태전이함수 φ 는 시스템의 현재 상태에서 입력에 따라 다음 상태로의 시스템의 변화를 나타내기 위한 것이고, 출력함수 λ 는 시스템의 현재 상태에서 입력에 의한 외부 시스템으로의 출력을 표현한 것으로 이 두 가지 함수가 시스템의 동적 특성을 반영한다. 이러한 일반 동적 시스템 모형을 객체지향 시뮬레이션에 적용하기 위해 시뮬레이션 구조를 <그림 2>와 같이 설계하였다. <그림 2>에서는 상태전이함수인 φ 를 상태전이 네트워크(state transition network)로 출력함수인 λ 를 다음 사건(next event)으로 적용한 것을 볼 수 있으며, 상태전이네트워크에서는 이동 가능한 객체에 정의되는 네 가지 상태와 이동 불가능한 고정위치 객체에 정의되는 두 가지 상태를 정의하였다.

4. 객체지향 분석 및 설계

컨테이너 터미널 시스템에 대한 시뮬레이션 모델을 설계하기 위해서 객체지향모델의 설계를 두 가지 기준인 객체설계

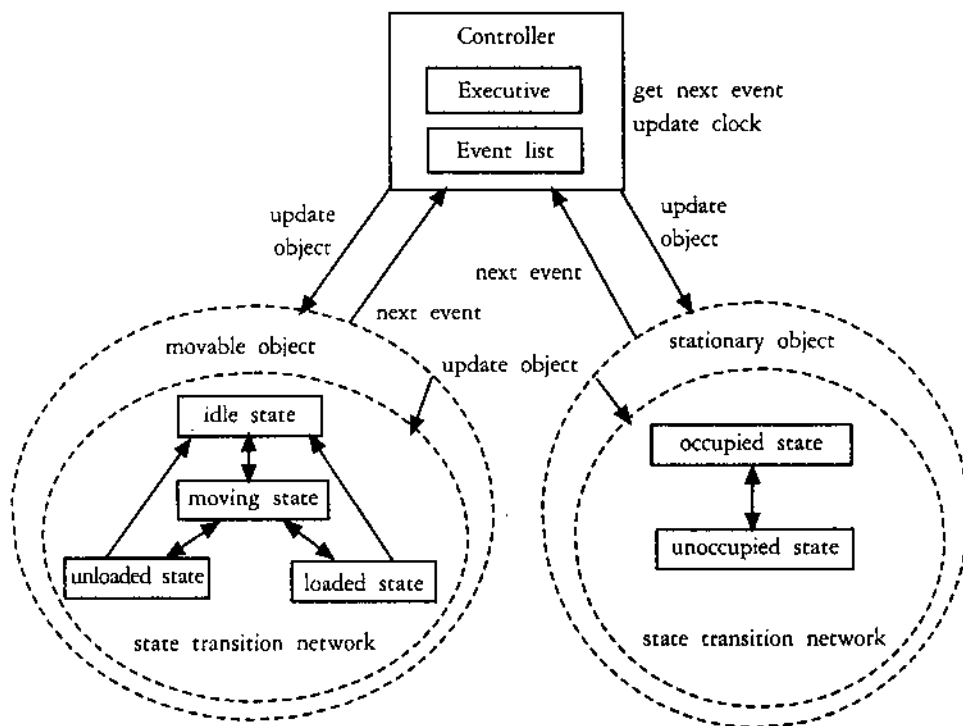


그림 2. 객체지향 시뮬레이션 구조.

(object scheme)와 사건설계(event scheme)로 구분하였으며, 이를 3단계로 나누어 설계하였다(David, 1996; Martin and Odell, 1992).

1, 2단계는 객체설계, 3단계는 사건설계이며 단계별 수행내용은 다음과 같다.

단계 1: 객체발견과 구조화

단계 2: 클래스-객체의 식별 및 속성 정의

단계 3: 메소드 정의 및 메시지 연결

정의된 3단계 과정을 사용된 용어의 정의와 함께 단계별로 설명한다.

4.1 객체발견과 구조화

객체의 발견은 객체지향 시뮬레이션을 위하여 컨테이너 터미널 시스템에 대해서 객체와 실물 개체의 대응, 모델의 계층과 시스템 계층의 대응을 고려하여 이루어진다. 이러한 대응 관계를 이용하여 발견된 객체들을 구조화하기 위해서는 사용되는 목적에 따라 분류가 가능하다. 크게는 물류객체(material flow object)와 정보흐름객체(information flow object)의 두 가지로 나누어진다. 물류객체는 기계장비, 운송장비, 시설물 등 물리적인 요소이며, 정보흐름객체는 자료, 정보, 객체속성과 같이 정확한 물류를 위해 사용되는 정보에 관한 객체이다. 이러한 발견된 객체를 구조화하기 위해 객체를 특성별로 분류하면 다음 <그림 3>과 같다.

물류객체는 객체의 구동능력 유무에 따라 이동 가능한(movable) 객체와 이동 불가능한(immovable) 객체로 나누어지는데, 이동 가능한 객체는 자신의 구동능력으로 위치를 변경시킬 수 있고 다른 물류객체에 귀속할 수도 있으며 생성과 소멸이 가능한 객체로 정의하며, 이동 불가능한 객체는 자신의 위치를 변경할 수 없으며 다른 물류객체에 귀속할 수 없는 객체로 정의하여야 한다. 이동 가능한 객체는 CC, TC, YT, Ship, Trailer 등과 같이 구동능력을 가지는 능동적(active) 객체와 컨

테이너와 같이 구동능력이 없는 수동적(passive) 객체로 나누어진다. 그리고 이동 불가능 객체는 능동적으로 작업수행에 관계하는 작업자(processor)와 대기장소(buffer), 수동적인 시설물의 역할을 하는 게이트, 장치장, 선석, 블록 등으로 분류된다.

정보흐름객체의 경우도 자신의 위치를 변경시킬 수 있고 정보흐름요소를 포함하며 생성, 삭제 및 복사될 수 있는 이동 가능한 객체와 자신의 위치를 변경할 수 없고 이동 가능한 정보흐름객체를 받아들이며 생성 및 삭제가 불가능한 이동 불가능한 객체로 나누어진다. 그리고 이동 가능한 객체와 이동 불가능한 객체는 다시 객체의 갱신의 주체에 따라 자신이 갱신할 수 있는 능동적 객체와 자신이 갱신할 수 없고 다른 객체에 의해 갱신되는 수동적 객체로 나누어진다. 예를 들면, 사용자가 정의한 모든 메소드는 이동 불가능한 객체이면서 능동적인 객체이며, CC, TC, YT 등의 장비의 작업일정정보는 이동 불가능한 객체이면서 수동적인 객체이다.

<그림 3>에서 물류객체로서 장비, 컨테이너, 시설물 등이 정의되었으며, 사용자 정의 속성, 메소드, 운영정보 등이 정보흐름객체로 정의되었고 또한 사용목적에 따라 특성별로 분류하여 객체의 구조화를 달성하였다.

4.2 클래스-객체의 식별 및 속성 정의

하나의 클래스와 그 클래스에 속한 객체들을 클래스-객체(class-object)로 정의하며, 한 클래스는 동질의 속성들을 가진 하나 이상의 객체로 정의하였다. 이러한 클래스-객체 관계를 묘사한 것이 객체설계이며, 이 객체설계로 컨테이너 터미널의 컨테이너 취급에 관련된 모든 객체들의 관계를 설명할 수 있으며, 객체설계에서 정의된 5개의 주요 클래스는 트랜스포터, 컨테이너, 게이트, 장치장, 선석 등이며, 이들 중 게이트, 장치장, 선석은 트랜스포터 클래스를 포함할 수 있는 것으로 정의한다.

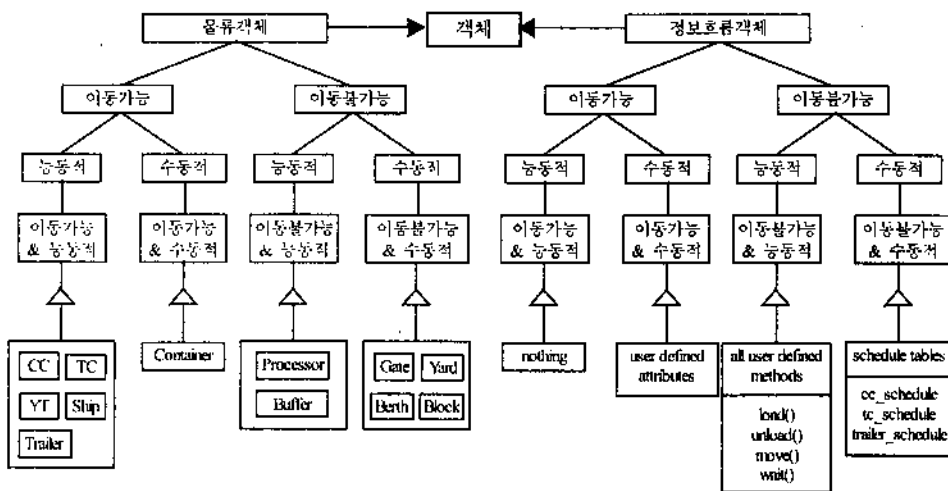


그림 3. 물류객체와 정보흐름객체의 구조.

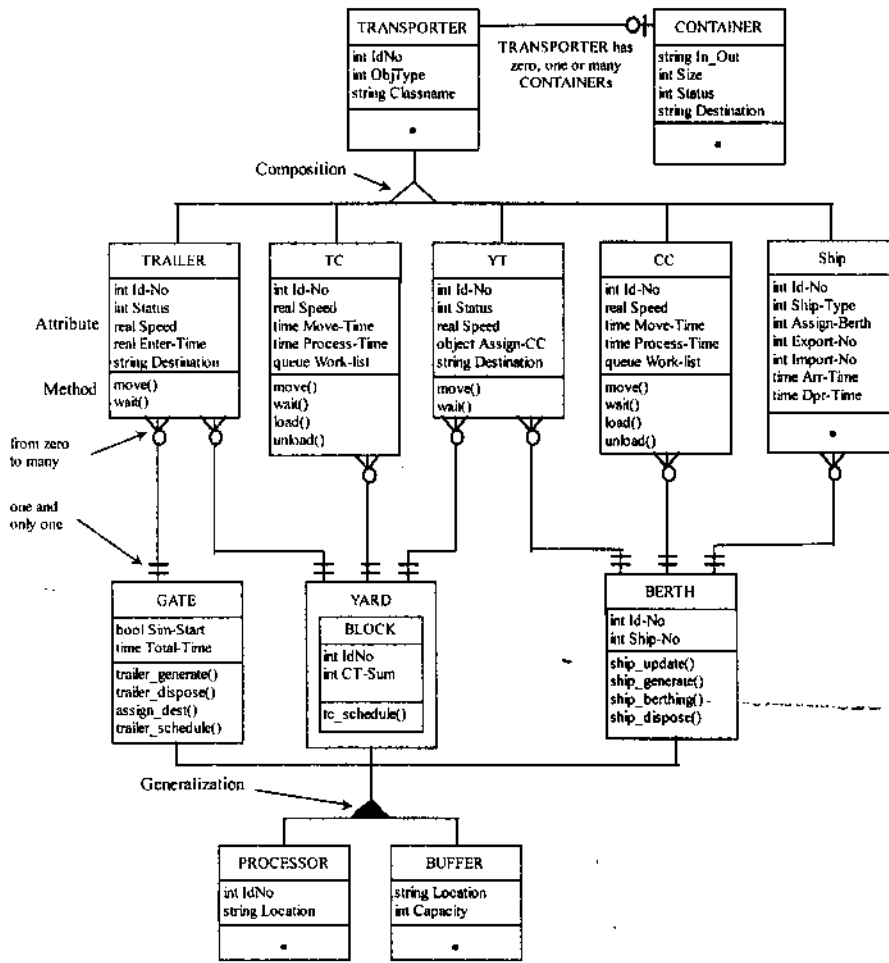


그림 4. 컨테이너 터미널에서의 객체설계.

<그림 4>의 객체설계와 같이 객체들간의 클래스 유사성을 계층적인 구조로 표현하는 것이 바람직하며, 이 구조는 최상위 수준에서 최하위 수준까지의 공통성(commonality)을 허용해야 한다. 공통성이 허용될 경우, 상위 수준의 클래스로부터 상속받은 모든 속성을 필요한 하위 수준의 클래스에게 일반화(generalization)시킴으로써 공통의 속성이 모듈방식과 확장 가능성을 제공한다. 따라서, 객체설계에서 다중상속(multiple inheritance)을 허용할 수 있다. 예를 들면, <그림 4>에서 서브 클래스인 작업자(processor)와 대기장소(buffer)는 게이트, 장치장, 선석 등 하나 이상의 클래스의 요소가 될 수 있다.

객체설계에서 정의된 객체들의 속성을 파악하는 데는 다음과 같은 컨테이너 터미널의 특성이 고려되어야 한다. 즉 컨테이너 터미널에는 컨테이너 흐름과 컨테이너 정보의 흐름의 두 가지 흐름이 있으며, 컨테이너의 흐름은 YT와 외부트럭 같은 운송차량에 의한 이동과 TC, CC 같은 컨테이너 취급 장비에 의한 이동이 있는데 이것은 컨테이너의 다음 목적장소를 찾아 전진적인(forward) 이동을 하며, 이에 반해 컨테이너 정보는 사용자 정의 속성과 메소드를 이용하여 후진적인(backward) 이동을 한다는 것이다. 따라서 컨테이너와 사용자 정의 정보의 행위가 객체설계에서 구별될 필요가 있다.

<그림 4>에서와 같이 객체설계에서는 클래스-객체의 관계 뿐만 아니라 객체의 속성까지도 포함시키는 것이 필요한데, 객체의 속성은 객체의 정적 상태와 동적 상태를 파악하거나 의사결정의 결과를 나타낼 수 있어야 한다. 따라서 클래스-객체에서 속성의 선택은 시스템의 상황과 시뮬레이션의 목적을 반영하는 것이라 할 수 있다. 예를 들면, 컨테이너에 정의된 'In_Our' 속성은 수출 컨테이너인지 수입 컨테이너인지를 판단하기 위한 의사결정용 속성이며, 'Destination'은 컨테이너가 놓여질 장치위치를 나타내는 속성으로 객체의 상태를 파악하기 위해 정의된다.

4.3 메소드 정의 및 메시지 연결

객체지향설계의 마지막 단계는 어느 메소드가 특정 객체를 처리하는가와 어떤 메시지가 객체들간에 통신되는가를 결정하는 것이다. 메소드는 운영 제어 의사결정을 하기 위해 필요한 절차(procedure)와 지식(knowledge)을 담고 있으며, 메시지는 한 객체를 나타내기 위한 특별한 메소드이다. 그러므로 메소드의 정의를 통해서 운영절차와 지식이 결정되면 이를 제어하기 위해 객체들간에 메시지를 통신시키는 것이 메시지 연결

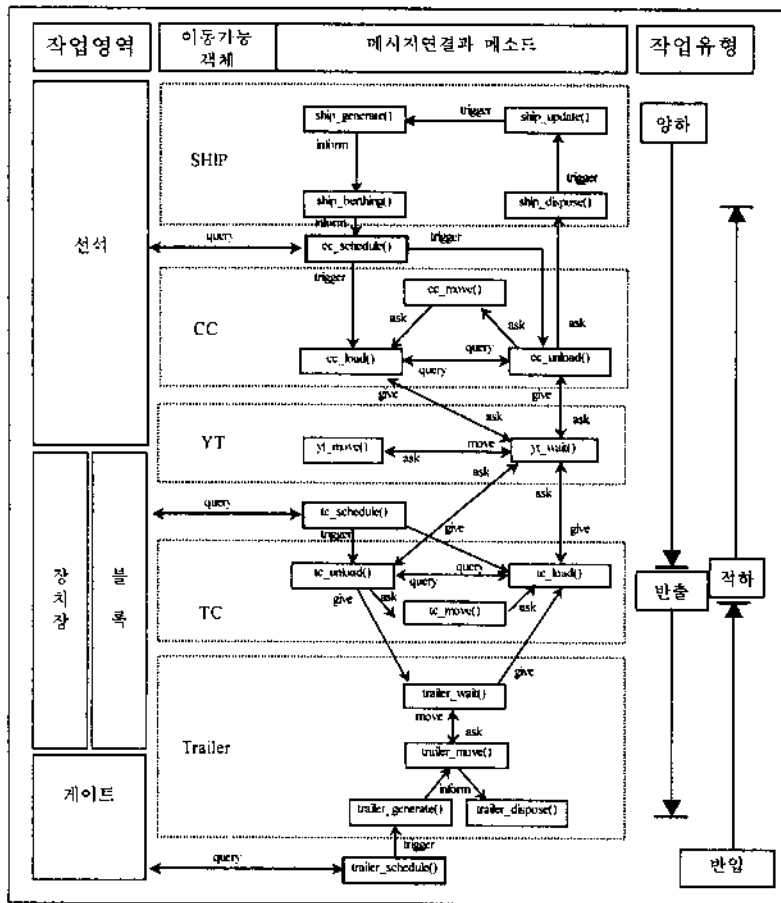


그림 5. 메시지 연결을 표현한 사건설계.

(message connection) 또는 메시지 통행(message passing)이라 하고, 메소드에 대한 메시지 연결을 정의한 것이 <그림 5>의 사건설계이며, 사건설계는 시뮬레이션 운영의 순서를 나타내는 역할을 한다.

메시지 연결은 한 객체가 자신의 메소드를 실행한 후 다른 객체에 메시지를 보내는 하나의 사상(mapping)을 나타낸다. 사건설계에서 컨테이너는 트랜스포터에 의해 메시지를 받는 과정이 메시지 연결로서 정의되어야 한다. 그러므로 TC와 CC처럼 이동 가능한 객체는 그 작업영역인 장치장과 선석 내에서 작업을 위해 메시지가 연결되며, YT는 장치장과 선석을 작업영역으로 하고 외부트럭은 게이트와 장치장을 작업영역으로 하여 메시지를 연결한다. 이러한 메시지 연결을 통해서 컨테이너의 장치위치, 컨테이너 작업목록 등의 컨테이너 운영정보를 작업영역과 이동 가능한 객체들간에 메시지를 전달할 수 있도록 한다.

<그림 5>의 사건설계는 메소드를 실행하여 그 결과값을 객체에 전달하거나 다음 사건을 위한 객체를 호출하기 위한 메시지를 정해진 메시지 통신규약(protocol)에 의해 메시지 연결을 한 것이며, 정의된 메시지 통신규약은 다음의 6가지를 사용하였다.

- (1) 정보전달(inform) : 한 객체가 다른 객체에게 정보를 전달한다.

- (2) 정보질의(query) : 한 객체가 다른 객체에게 정보를 질의한다.
- (3) 작업요구(ask) : 한 객체가 다른 객체에게 작업을 요구한다.
- (4) 대기(wait) : 객체의 상태가 특정 조건을 만족시킬 때까지 대기한다.
- (5) 사건유발(trigger) : 객체의 작업이 계속되도록 사건을 일으킨다.
- (6) 객체전달(give) : 한 객체에서 다른 객체로 컨테이너를 전달한다.

위에서 정의된 6가지 통신규약을 통하여 객체간에 메시지를 통신시키게 되는데, 예를 들어 정보질의 메시지는 객체의 속성값을 갱신하거나 메시지를 전달하기 위해 메시지를 사건의 과정으로 선언하는 것이다. 그리고 메시지간에도 일대일 대응관계가 있으므로 한 객체와 그 객체가 가진 메소드에 의해서 메시지를 주고받으므로 메소드가 서브루틴(subroutine)으로 처리되며, 메시지는 서브루틴을 호출(call)하는 역할을 한다. 따라서 캡슐화된 메시지만을 주고받도록 객체간의 통신을 제약함으로써 정보은닉을 달성할 수 있다. 예를 들면, 한 객체의 메소드가 다른 객체에서 접근하여 속성을 새로 쓰거나 읽을 수 없도록 하는 것이다. 이러한 메시지 통신규약에 의한 메시지 연결이 확정되면 객체들은 개별적으로 완전히 독립되므로 모듈화된 프로그램을 개발할 수 있게 된다.

5. 시뮬레이션 시스템 구조 설계

5.1 시스템 구조

시뮬레이션 시스템의 구조는 모델의 구성과 묘사를 위한 모델부(model subject)와 시뮬레이션의 제어와 애니메이션을 위한 시뮬레이션부(simulation subject)의 두 개의 모듈로 구성하였다. 시뮬레이션부는 시뮬레이션을 위한 초기화, 사건제어, 출력제어 등 시뮬레이션의 실행을 위한 입력자료의 처리에서 이벤트의 처리, 이벤트에 의한 메소드의 호출 및 변환된 상황의 애니메이션 적용, 출력통계량 산출까지의 전반적인 관리자의 역할을 한다. 그리고 모델부는 <그림 4>에서 정의된 객체설계를 구체화한 것으로 객체들간의 관계와 객체내의 메소드까지 실제 모델의 행동이나 로직을 표현하는 부분으로 애니메이션까지 고려된 계층구조이다. 이러한 모델부와 시뮬레이션부의 관계는 <그림 6>의 시뮬레이션 시스템의 구조와 같다.

<그림 6>에서 이 두 가지 모듈은 서로 메시지를 주고받으며, 시뮬레이션부에서 정의된 속성은 다음과 같다.

■ 시뮬레이션 관리(Simulation Management)

식별자, 시스템 관리자, 시스템 이름, 시스템 시간

① 초기화(Initialization)

식별자, 프로젝트 이름, 분석자 이름, 실행시간, 워밍업시간, 시스템 사양

역할: 입력에서 사용자 인터페이스 제공, 초기화 메소드 실행, 메소드 호출시간 결정

② 사건제어(Event Control)

최대반복수, 반복횟수, 다음 사건

역할: 사건목록 작성, 다음 사건 결정, 사건목록 처리

③ 실행기(Executive)

실행속도, 시뮬레이션 시간

역할: 예정된 메소드 호출, 메소드의 재귀호출, 객체의 상태

변화 표현

④ 출력 생성기(Output Generator)

제목, 부제목, 데이터파일, 출력파일

역할: 개별 통계량 출력, 요약보고서 출력, 출력 통계량 분석

위에서 정의된 시뮬레이션부와 모델부를 모듈화한 시뮬레이션 시스템을 <그림 2>의 시뮬레이션 구조에 적용하는 단계가 본 연구에서의 시뮬레이터의 설계단계이며, 이를 완성시키기 위해 설계를 바탕으로 한 주요 로직을 유사코드(pseudo code)로 묘사하였다.

5.2 주요 로직 설계

시뮬레이터에서 수행되는 전체 로직은 아래와 같이 크게 사용자 입력, 입력에 의한 초기화, 시뮬레이션 실행, 그리고 출력 산출기 실행 등의 순으로 구성된다.

```
//setup the system for simulation, defaults and input;
//display the system in default windows;
Begin{
  read user_input module; //사용자 입력정보를 읽는다.
  execute init_module to setup the system for simulation; //사용자
  입력을 초기화한다.
  execute simulation_run methods; //시뮬레이션 실행 로직 처리
  execute output generator; //출력 산출기 실행
}End;
```

위의 4가지 실행순서에서 특히 시뮬레이션 실행 부분의 로직 중에서 이동 가능 객체의 하나인 TC 장비에 대해서 주요 실행 로직을 설명하면, 아래와 같이 workcontrol(), load(), unload() 등의 메소드로 TC를 제어하는 과정을 묘사할 수 있다. TC와 CC가 유사하게 정의되었으므로 TC의 경우를 예로 주요 로직

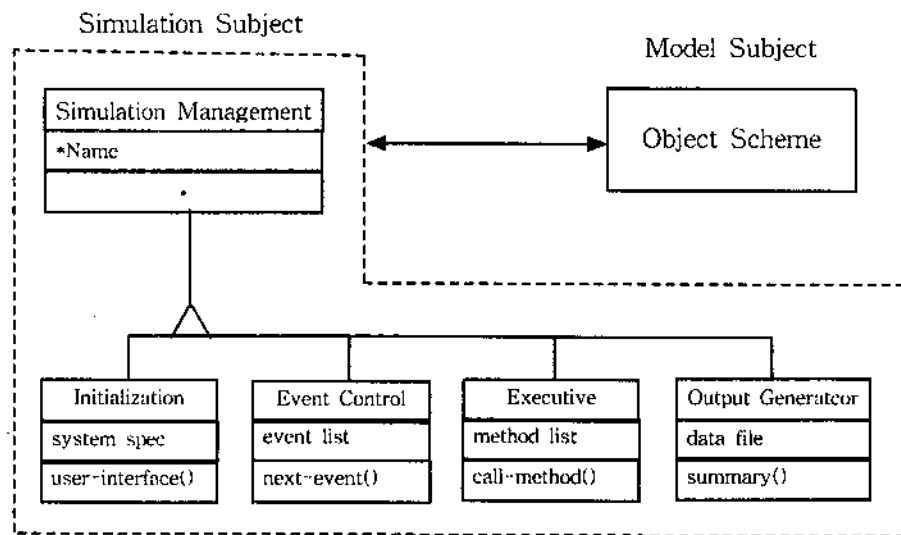


그림 6. 시뮬레이션 시스템의 모듈 구조.

을 설명하면 CC, YT, 외부트럭의 로직 또한 비슷하게 정의될 수 있다. 그래서 아래의 유사코드에서 YT로 설명된 부분에 외부트럭이 놓이더라도 같은 속성의 차량으로 간주하여 동일하게 로직을 수행하도록 하였다.

TC.workcontrol() //TC의 작업을 제어하기 위한 메소드

```
Begin{
if(TC.work_list(queue) == empty)
    TC.work_state="idle";
elseif(TC.work_list(queue)<>empty)
    while(TC.work_list(queue)<>empty)
    {
        TC.dest_bay=TC.work_list(first_row);
        while(TC.current_bay == TC.dest_bay)
            TC.move(dest_bay)
        if(Block.YT.arrived == true)
        {
            if YT.ct.in_out="In"
                TC.unload();
            elseif YT.ct.in_out="Out"
                TC.load();
        }
    }
elseif(Block.YT.arrived == false)
    TC.work_state="wait";
    Block.TC.arrived=true;
}
}End;
```

TC.unload() //TC의 unload작업

```
Begin{
    delay processing_time;
    update Tb_block_assign;
    YT.state=empty;
    delete TC.work_list(first_row);
}End;
```

TC.load() //TC의 load작업

```
Begin{
    delay processing_time;
    update Tb_block_assign;
    YT.state=full;
    delete TC.work_list(first_row);
}End;
```

위의 유사코드의 예는 일부의 로직을 설명하였지만 위의 예에서와 같이 로직을 유사코드로 설계하고 나면, 실제 구현할 경우에는 객체지향 프로그래밍이 가능한 C++의 형식으로 코

딩을 하여 시뮬레이터의 로직을 구현할 수 있다. 여기서 로직의 코딩은 정의된 클래스에 코딩하며, 이를 템플릿(template)으로 사용할 수 있다. 이러한 과정으로 클래스를 개발해 나간다면 시스템 구현시에 프로그래밍 단계에서 코딩에 소요되는 시간을 많이 줄일 수 있을 것이다.

6. 결론

본 연구에서는 컨테이너 터미널 시스템의 시뮬레이터를 개발하기 위한 객체지향형 시뮬레이션에 대한 설계과정을 제안하였으며, 이러한 목적에 부합되도록 객체지향분석과 객체지향설계를 통합화한 객체지향접근법을 사용하였다. 또한 시스템의 구조를 일반 동적 시스템 모형에 적용하여 설계하였다.

객체지향분석은 실제 시스템을 분석하여 개념모델을 구성하기 위해서 사용되었으며, 객체지향설계는 개념 모델을 시뮬레이션 모델로 변환하기 위해서 사용되었다. 그러나 객체지향분석과 객체지향설계가 순차적으로 이루어지는 것이 아니라 객체지향접근법이라는 하나의 테두리에서 동시적으로 이루어지는 것이 더 바람직하며, 시뮬레이션 시스템의 설계시간을 줄일 수 있을 것이다. 그리고 시스템 구조는 일반 동적 시스템 모형에서 정의된 객체의 상태전이 네트워크에 의해 구체화하여 표현할 수 있으며, 시뮬레이션부와 모델부의 두 모듈로 시스템의 구조를 설계하였으며, 주요 로직에 대한 유사코드를 예로 들어 구체적인 설계안을 기술하였다.

따라서 본 연구에서는 객체지향접근법을 사용한 설계과정을 객체발전과 구조화, 클래스-객체의 식별 및 속성 정의, 메소드 정의 및 메시지 연결의 3단계를 하나의 연속적인 과정으로 설명하였다. 이러한 제안된 설계과정을 이용한다면 트랜스퍼 크레인 시스템을 사용하는 컨테이너 터미널뿐만 아니라 스트래들 캐리어 시스템을 포함한 다양한 컨테이너 터미널의 시뮬레이션 시스템의 설계에도 활용될 수 있으며, 또한 시뮬레이터를 개발하기 위한 효과적인 도구의 역할도 할 수 있을 것이다.

참고문헌

- 윤원영, 안창근, 최용석, 김갑환 (1998), 시뮬레이션을 이용한 컨테이너 터미널의 운영계획 평가, *한국시뮬레이션학회 논문지*, 7(2), 91-104.
- 윤원영, 최용석 (1997), 자영 컨테이너 장치장의 평가를 위한 시뮬레이션 모델의 개발, *한국시뮬레이션학회 논문지*, 6(1), 109-122.
- 조덕운 (1985), 컨테이너항 전산 모의실험 모형의 개발, *대한산업공학회지*, 11(2), 173-187.
- David, R. C. H. (1996), Object-Oriented Analysis and Simulation, *Addison-Wesley Publishing Company*.
- Gambardella, L. M., Rizzoli, A. E., and Zaffalon, M. (1998), *Simulation and Planning of an Intermodal Container Terminal*, *Simulation*, 71(2), 107-116.
- Hayuth, Y., Pollatschek, M. A., and Roll, Y. (1994), *Building A Port Simulator*, *Simulation*, 63(3), 179-189.
- Martin, J. and Odell, J. J. (1992), *Object-Oriented Analysis and Design*,

- Prentice-Hall, New Jersey.
- Merkuryev, Y., Tolujew, J., Blumel, E., Novitsky, L., Ginters, E., Viktorova, E., Merkuryeva, G., and Pronins, J. (1998), *A Modeling and Simulation Methodology for Managing the Riga Harbour Container Terminal*, *Simulation*, 71(2), 84-95.
- Mesarovic, M. D., Takahara, Y. (1975), *General System Theory: Mathematical Foundations*, Academic Press.
- Nevins, M. R., Macal, C. M., and Joines, J. C. (1998), *A Discrete-Event Simulation Model for Seaport Operations*, *Simulation*, 70(4), 213-223.
- Nevins, M. R., Macal, C. M., Love, R. J., and Bragen, M. J. (1998), *Simulation, Animation and Visualization of Seaport Operations*, *Simulation*, 71(2), 96-106.
- Park, C. S., Noh, Y. D. (1987), *A Port Simulation Model for Bulk Cargo Operations*, *Simulation*, 48(6), 236-246.
- Raczynski, S. (1988), *Process Hierarchy and Inheritance in PASTOR*, *Simulation*, June, 249-251.
- Ramani, K. V. (1996), *An Interactive Simulation Model for the Logistics Planning of Container Operations in Seaports*, *Simulation*, 66(5), 292-300.
- Silberholz, M. B., Golden, B. L., and Baker, E. K. (1991), *Using Simulation to Study the Impact of Work Rules on Productivity at Marine Container Terminals*, *Computers & Operations Research*, 18(5), 433-452.
- Thiers, G. F., Janssens, G. K. (1998), *A Port Simulation Model as a Permanent Decision Instrument*, *Simulation*, 71(2), 117-125.
- Yun, W. Y., Choi, Y. S. (1999), *A Simulation Model for Container-Terminal Operation Analysis Using An Object-Oriented Approach*, *International Journal of Production Economics*, 59, 221-230.



윤원영

서울대학교 산업공학과 학사
한국과학기술원 산업공학과 석사
한국과학기술원 산업공학과 박사
현재: 부산대학교 산업공학과 교수
관심분야: 신뢰성 공학, 시뮬레이션



이명길

경성대학교 산업공학과 학사
부산대학교 산업공학과 석사
현재: (주)태평양화학
관심분야: 컴퓨터 시뮬레이션 응용



최용석

창원대학교 산업공학과 학사
부산대학교 산업공학과 석사
부산대학교 산업공학과 박사과정 수료
관심분야: 객체지향 시뮬레이션



송진영

동아대학교 산업공학과 학사
부산대학교 산업공학과 석사
현재: (주)토탈소프트뱅크
관심분야: 항만 운영시스템