

컨테이너 터미널 설계를 위한 의사결정 지원시스템*

원승환
부산대학교 산업공학과,
(shwon@pusan.ac.kr)

천봉경
부산대학교 컴퓨터공학과
(bkchun@pearl.cs.pusan.ac.kr)

전수민
부산대학교 산업공학과,
(1006sumin@pusan.ac.kr)

이병권
부산대학교 산업공학과,
(errorplus@pusan.ac.kr)

장동원
부산대학교 산업공학과
(dwjang@pusan.ac.kr)

조환규
부산대학교 컴퓨터공학과
(hgcho@pusan.ac.kr)

김갑환
부산대학교 산업공학과,
(kapkim@pusan.ac.kr)

컨테이너 터미널의 건설에는 아주 많은 시간과 비용이 소요되기 때문에 컨테이너 터미널을 효율적으로 설계하는 것이 필수적이다. 본 연구는 컨테이너 터미널의 엔지니어링 프로세스를 지원하기 위해서 개발된 소프트웨어 시스템을 소개하였다. 설계 프로세스는 데이터 입력, 능력 소요 계산, 대안 설계, 대안 평가, 결과 출력으로 구성된다. 대안을 신속히 설계하고 평가하기 위해서 컨테이너 터미널 전용 라이브러리가 개발되었다. 라이브러리는 Wharf, Yard, TravellingArea, Gate로 구성된다. 라이브러리의 구조는 객체 지향적으로 설계하여 확장성과 재사용성을 가지게 하였다. 시스템은 사용자에게 친숙한 인터페이스와 CAD 환경을 제공하며, 다양한 유형의 컨테이너 터미널 모형들이 데이터 입력과 라이브러리 구성요소의 조합에 의해 구성될 수 있도록 지원한다. 생성된 대안 모형들은 시뮬레이션 모형에 의해 평가될 수 있게 개발되었다.

논문접수일 : 2006년 07월 게재확정일 : 2006년 12월 교신저자 : 원승환

1. 서론

최근 컨테이너 터미널 간의 치열한 경쟁, 대형 선박의 출현, 컨테이너 터미널 운영의 자동화 등으로 인해서 세계 각국에서는 많은 컨테이너 터미널들을 건설 중이다. <표 1>은 아시아 주요항만들의 시설현황과 개발계획을 보여준다(한국컨테이너부두공단, 2005).

컨테이너 터미널을 건설하기 위해서는 장기간의 계획과 많은 투자가 필요하므로, 궁극적으로

요구사항을 충족하면서 비용 면에서도 효율적인 건설을 위한 설계절차가 필요하다. 그러나 현재 컨테이너 터미널의 설계 업무는 대부분 주먹구구식 성격을 가지고 있다. 컨테이너 터미널의 설계 업무를 이론적 기반을 갖춘 지식기반 업무로 변환시킨다면, 항만 물류시스템 설계기술의 향상을 통한 경쟁력 향상에 도움을 줄 것이다.

컨테이너 터미널의 설계와 관련된 기존 연구들은 대부분 선정한 대상 터미널에 대한 시뮬레이션 모형의 개발과 평가에 한정되어 있었다(조덕운,

* 이 논문은 교육인적자원부 지방연구중심대학 육성사업(차세대물류IT기술연구사업단)의 지원에 의하여 연구되었음.

<표 1> 아시아 주요항만의 시설현황 및 개발계획
(2003년 말 기준)

항 만	현재 선석수	계획 선석수
홍콩	23	18
싱가포르	42	49
선전	15	15
가오슝	19	23
상하이	23	65
칭다오	8	14
탄중펠레파스	6	6
닝보	4	18
톈진	8	10
부산	18	30
광양	4	33
계	170	281

1985; 장성용, 박진우, 1988; 김현, 1988; Ramani, 1996; Yun and Choi, 1999; Tahar and Hussain, 2000). 독일의 ISL에서 개발된 시뮬레이션 시스템 SCUSY는 유일하게 컨테이너 터미널의 설계를 지원하는 시스템으로 알려져 있다. 주요 기능으로는 터미널의 배치계획의 수립, 터미널의 장비계획 수립, 여러 가지 운영시나리오에 대한 성능 및 경제성 평가, 새로운 컨테이너 취급 시스템에 대한 성능 검토 등을 포함한다. 출력 정보로는 장비의 활용도, 각 장비의 대기시간 분포, 외부트럭에 대한 서비스 시간, 철도수송시설에 대한 서비스 수준, 상세한 비용 평가치 등이 있다. 현재까지 브레멘 하프의 BLG/Eurogate 컨테이너 터미널을 비롯하여 27개 정도의 시설들에 대한 적용 실적을 가진 검증된 시스템이다. Saanen(2004)은 자동화 컨테이너 터미널을 설계하기 위한 절차와 사례를 제시하였다. 그가 제안한 설계 절차는 4단계로 구

성되고, 모든 단계에서 시뮬레이션 모형을 활용한다는 특징을 가진다. 결국 현재까지의 연구들을 종합해보면 컨테이너 터미널의 설계과정 전체를 지원하는 시스템의 개발은 거의 시도되고 있지 않았다.

본 연구에서는 사용자와의 편리한 인터페이스로 가상의 컨테이너 터미널을 설계할 수 있는 시스템을 구현해보고자 한다. 사용자가 단기간에 최소한의 입력으로 설계 업무를 완료하도록 지원하기 위해서, 시스템은 컨테이너 터미널의 구성 요소를 미리 템플릿으로 준비하는 것이 필요하다. 또한 다양한 운영 논리를 간편하게 입력할 수 있도록 지원하는 방안이 필요하다.

컨테이너 터미널의 설계에는 많은 최적화 문제들이 포함되어 있다. 설계 과정에서 다양한 최적화 모형을 활용함으로써 설계 결과물의 품질을 향상시킬 수 있다. 도출된 설계안들의 경제성과 성능을 평가하여, 선행 과정으로의 피드백 과정을 거치면서 점차 만족스런 결과에 가까워질 것이다. 컨테이너 터미널의 성능을 상세하게 분석하기 위한 도구로는 시뮬레이션이 적당하다. 컨테이너 터미널은 많은 독립적인 요소들이 유기적으로 구성된 대규모 물류시스템이므로 평가의 대상과 상세화 정도에 따라서 시뮬레이션의 추상화 수준을 선택할 수 있다면 다양한 평가에 많은 기여를 할 수 있을 것이다.

이상의 시스템 요구사항을 만족시키기 위해서 시스템의 설계와 구현에 객체지향적인 접근을 적용하는 것이 바람직하다. 특히, 컨테이너 터미널의 구성 요소를 템플릿으로 제작할 때 이러한 접근방법은 필수적이며, 선택된 템플릿 요소들을 결합하여 완전한 컨테이너 터미널을 구성 가능하여야 한다. 각 템플릿 요소는 터미널 구성 요소의 특징을 추상화하여 모형화 된다.

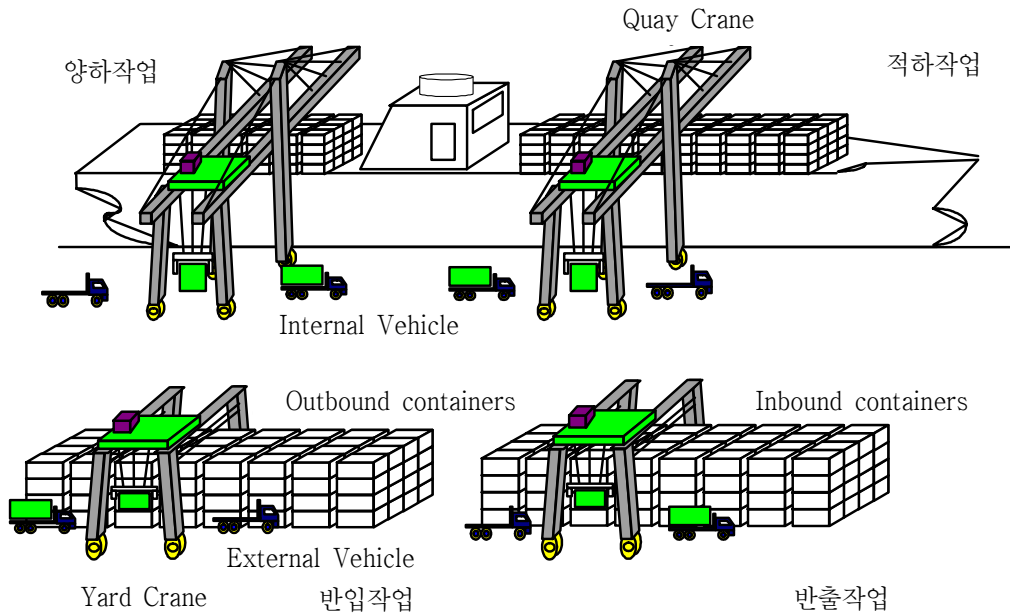
본 연구는 컨테이너 터미널의 설계 및 변경에 유연하게 대처할 수 있는 항만 물류시스템 설계용 통합 소프트웨어 개발을 목표로 한다. 기존 시스템과 (예를 들어 SCUSY 시스템)과의 차이로는 객체지향적인 접근방법을 사용하였으므로 다양한 터미널 형태에 대해서 단시간 내에 모델링을 통하여 평가를 해 볼 수 있고 새로운 형태의 장비나 터미널 형태를 추가하기 쉽다는 장점을 가지고 있다. 그리고 능력계획, 최적화 기능, CAD기능들을 하나의 시스템으로 통합하여 개발하였다는 장점을 들 수 있다.

2장에서는 컨테이너 터미널의 개요에 대한 설명을 하고, 3장에서는 제안하는 시스템의 분석 및 설계과정에 대한 설명을 한다. 4장에서는 제안하는 시스템의 구성과 구현된 설계 절차에 대해서 논하고, 5장에서는 결론을 짓는다.

2. 컨테이너 터미널의 개요

컨테이너 터미널에서의 작업은 본선 작업과 반출입 작업으로 구분된다. 본선 작업 동안, 컨테이너들은 선박으로부터 양하 되거나, 선박으로 적하된다. 반출입 작업 동안, 컨테이너들은 외부차량으로부터 반입 되거나, 외부차량으로 반출된다. [그림 1]은 컨테이너 터미널의 주요 작업을 설명한다. 이러한 작업들을 처리하기 위해서 안벽 크레인, 야드 크레인, 내부차량, 외부차량 등이 협력하여 작업하게 된다.

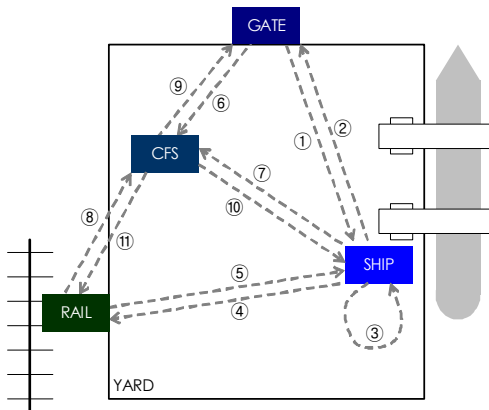
컨테이너 터미널 내에서 컨테이너 흐름의 유형은 수입 컨테이너, 수출 컨테이너, 환적 컨테이너로 구분된다. 수입 컨테이너는 양하 작업과 반출 작업으로 처리되고, 수출 컨테이너는 반입 작업과 적하 작업으로 처리되며, 환적 컨테이너는 양하



[그림 1] 컨테이너 터미널의 주요 작업

작업과 적하 작업으로 처리된다.

각 유형별 흐름량을 파악하기 위하여 [그림 2]와 같이 주요 기점간의 흐름을 정의한다. SHIP, GATE, RAIL, CFS를 주요 기점으로 정의한다. 각각 선박, 게이트, 철도수송 터미널, 컨테이너 작업장을 나타낸다. 컨테이너 작업장(Container Freight Station; CFS)은 하나의 컨테이너로 채울 수 없는 양의 화물(Less than Container Load; LCL)을 여러 하주로부터 인수하여 목적항별로 선별하여 컨테이너로 적재하거나, 한 컨테이너로부터 반출된 여러 하주의 화물을 각 하주에게 인도해주는 장소이다.



[그림 2] 주요 기점간의 컨테이너 흐름

주요 기점간의 컨테이너 흐름을 사용하면 수입 컨테이너는 SHIP⇒GATE(2), SHIP⇒RAIL(4), SHIP⇒CFS(7), CFS⇒GATE(9), CFS⇒RAIL(11)로, 수출 컨테이너는 GATE⇒SHIP(1), RAIL⇒SHIP(5), GATE⇒CFS(6), RAIL⇒CFS(8), CFS⇒SHIP(10)으로, 환적 컨테이너는 SHIP⇒SHIP(3)으로 표현된다.

[그림 2]와 같이 다양한 형태의 흐름이 공존하는 터미널을 대상으로 설계를 수행하여야 하므로

모델링이 복잡하고 고려하여야 할 사항이 많다고 할 수 있다. 이와 같은 흐름요구는 본 시스템의 입력 자료로서 입력을 위한 모듈의 개발이 이루어져야 할 것이다. 그리고 이들 흐름요구량은 시스템이 만족하여야 한 요구조건으로서 장비의 형태 선정, 장비나 시설의 규모결정 시 고려하여야 할 중요한 자료가 된다.

3. 시스템 분석 및 설계

3.1 시스템 요구사항

제안하는 시스템은 컨테이너 터미널의 증장기 계획, 운영 전략 및 알고리즘의 개발 등에 다목적 으로 활용할 수 있어야 한다. 그러기 위해서 시스템에 포함된 모형은 복잡한 컨테이너 터미널의 상황을 최대한 반영하여야 한다.

대안의 설계 및 평가 기능은 컨테이너 터미널을 구성하는 안벽, 게이트, 장치장 등의 단위 모듈들을 독립적으로도 사용 가능하고 각 모듈들을 결합한 통합 시스템으로도 사용 가능하여야 한다. 그리고 물류시스템의 설계 기술, 최적화 기술, 평가 기술 등이 융합된 통합시스템의 형태를 갖추어야 한다.

3.2 데이터 모형화

데이터 모형화는 시스템의 데이터를 체계화하고 문서화하는 기법이다(Whitten et al., 2000). 데이터 모형은 결국 데이터베이스로 구현되기 때문에 데이터베이스 모형화라고도 불린다. 혹은 정보 모형화라고도 불린다. <표 2>는 데이터베이스로 구현된 테이블의 리스트를 나타낸다.

<표 2> 데이터베이스 테이블 리스트

Name	Description
ALT_ID	name of alternatives
BERTH	information of Berth
BERTH_LIST	information of each Berth
BLOCK_LIST	information of each Block
BLOCK_LIST_SUB	information of tier's updown transfer time
CFS_TO_SHIP	from CFS to ship, each flow type container information
CONTAINER_FLOW	information of container flow
CUSTOMER_SERVICE_LEVEL	service type
DATA_SET_ID	name of project
DESIGN_ALTERNATIVE	information of time capa about alternative for a year
EXPORT_FLOW	each container flow type information in export flow
GATE	information of gate process
GATE_GENERAL	information of gatein and gateout
GATE_TO_SHIP	from gate to ship, each container flow type information
IMPORT_FLOW	each container flow type information in import
OD_FLOW	origin and destination flow in port
OPERATION_POLICY	How long to keep for free containers & operation time
OPERATION_POLICY_DWELL	dwel information about each container flow
OPERATION_POLICY_YARD_UTIL	information of each yard utilization
QUAY_CRANE	information of quay crane
QUAY_LIST	information of quay
QUAYCRANE_LIBRARY	quay crane's configuration information
QUAYCRANE_LIST	each quay crane's information
QUAYCRANE_LIST_SUB_LOAD	quay crane's sub information
QUAYCRANE_LIST_SUB_UNLOAD	quay crane's sub information
RAIL_TO_SHIP	container flow from rail to ship
SHIP_TO_CFS	container flow from ship to cfs
SHIP_TO_GATE	container flow from ship to gate
SHIP_TO_RAIL	container flow from ship to rail
SHIP_TO_SHIP	container flow from ship to ship
TERMINAL_PROCESS_RESULT	terminal process result
TRAVELLINGAREA_GENERAL	information of travelling area
TS_FLOW	transshipment flow
VEHICLE	information of vehicle
VEHICLE_LIBRARY	capacity of vehicle
VEHICLE_LIST	capacity of each vehicle
WORK_SHIFT	information of work shift
WORK_SHIFT_PAUSE	information of off duty work shift
YARD_CRANE	information of yardcrane
YARD_GENERAL	information of each yardcrane
YARDCRANE_LIBRARY	information of each yardcrane type capacity
YARDCRANE_LIST	information of each yardcrane

3.3 프로세스 모형화

프로세스 모형화는 시스템의 프로세스를 통한 데이터의 구조 및 흐름, 혹은 시스템의 프로세스에 의해 이행되는 논리, 방침, 절차를 체계화하고 문서화하는 기법이다(Whitten et al., 2000). DFD(Data Flow Diagram)는 시스템을 통한 데이터의 흐름과 시스템에 의해 수행되는 작업 혹은 처리를 표현하는 도구이다. [그림 3]은 context DFD를 나타낸다. context DFD는 시스템의 범위를 수립하는데 유용하고, 구체적인 프로세스로 분해된다.

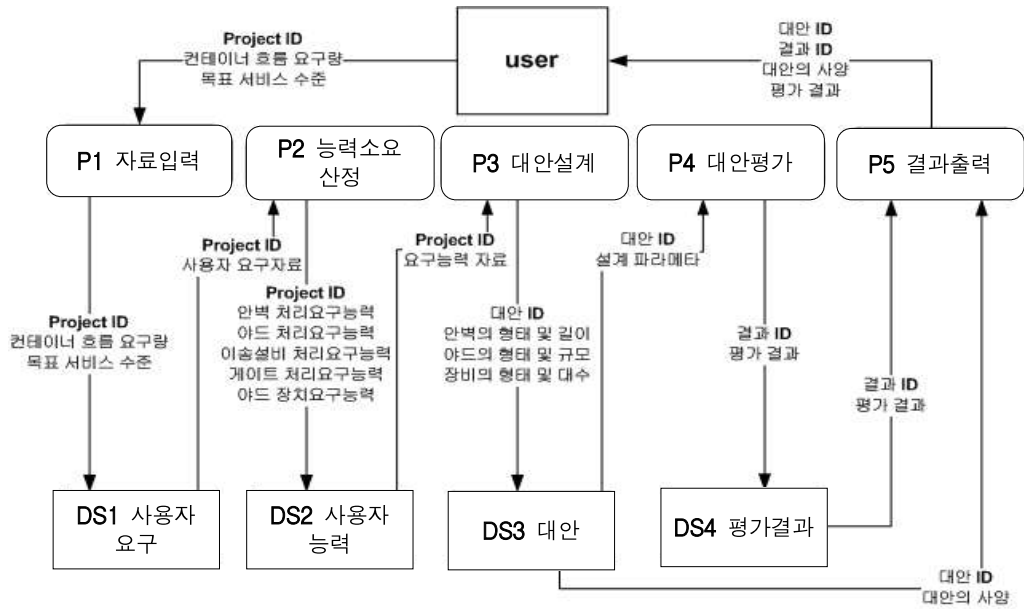
[그림 3]의 P2 능력소요 산정 프로세스가 5개

의 하위 프로세스(P2.1~P2.5)로 분해된 것이 [그림 4]에 보인다. 프로세스 뿐 아니라 데이터 흐름, 데이터 저장소(Data Store)도 보다 구체적으로 분해된 것을 볼 수 있다.

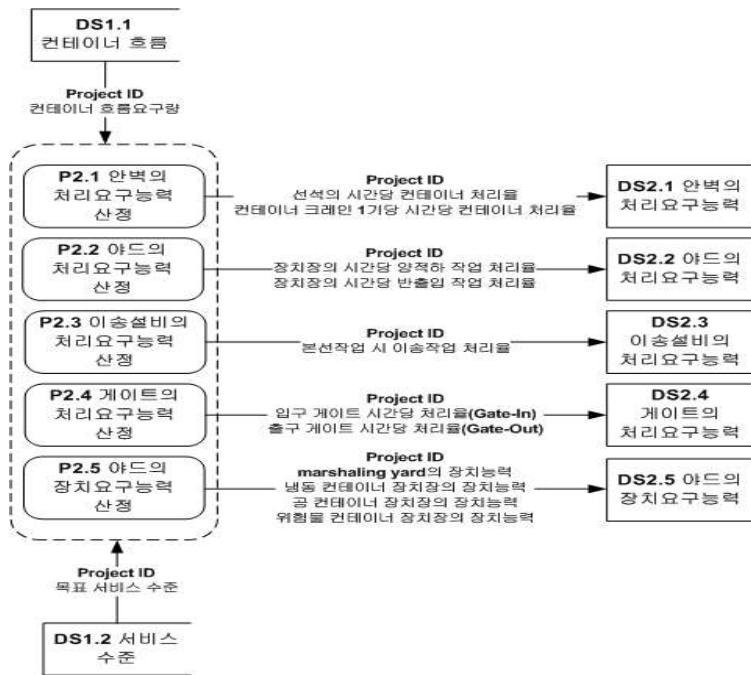
[그림 5]는 [그림 4]의 P2.1 안벽의 처리요구능력 산정 프로세스가 최종적으로 분해된 하나의 예를 보여준다. 이 예는 P2.1.1 선석의 시간당 컨테이너 처리율 산정을 표현한다. 최종적으로 분해된 다이어그램에는 단일한 사건에 대한 모든 기초 프로세스, 데이터 저장소, 데이터 흐름이 나타난다. 따라서 이러한 다이어그램을 primitive DFD라고 부른다. primitive DFD에 표현된 데이터 저장소는 데이터베이스의 테이블, 데이터 흐름은 데이터베이스 테이블의 필드에 해당된다.

3.4 객체 지향적 접근

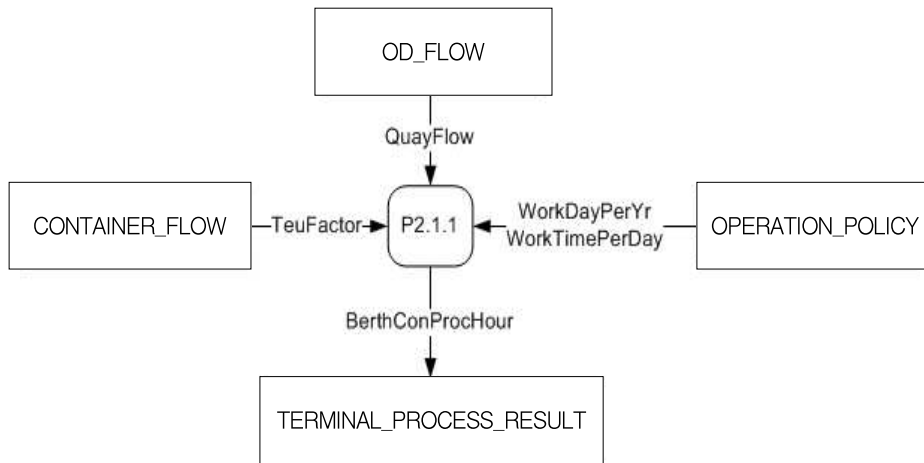
장비 라이브러리의 설계에 객체지향 접근법을 사용하였다. 컨테이너 터미널에 사용되는 각 장비들이 클래스로 정의된다. 컨테이너 터미널의 장비들은 지속적으로 진화하고 있고 새로운 장비의 출현이 잇따르고 있다. 이러한 추세를 반영하기 위하여 추상 클래스를 먼저 정의하고 상속을 통하여 구체적인 클래스를 정의해 나갔다. [그림 6]은 장비 라이브러리의 클래스 다이어그램을 나타낸다. 모든 장비의 공통적인 속성과 행위를 포함하도록 최상위에 *Transporter* 클래스를 정의하고 이로부터 *Crane* 및 *Vehicle* 클래스를 정의하였다. *Crane* 클래스는 제한적인 이동 경로를 가지는 장비를, *Vehicle* 클래스는 상대적으로 다양한 이동 경로를 가지는 장비를 묘사하는데 적합하다. *Crane* 클래스로부터 사용 용도에 따라서 *YardCrane*, *WharfCrane*, *RailCrane* 클래스를 각각 정의하였다. 이러한 클래스로부터 다양한 장비들을 정의할 수 있다.



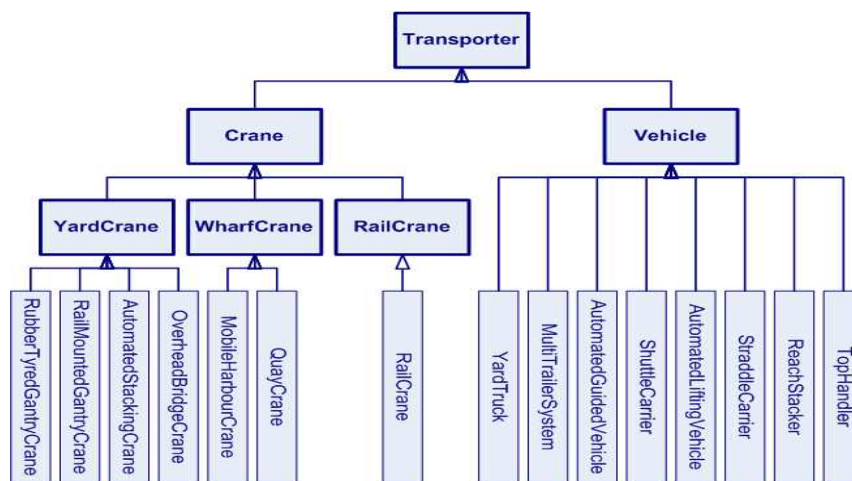
[그림 3] Context DFD



[그림 4] Decomposed DFD



[그림 5] Primitive DFD



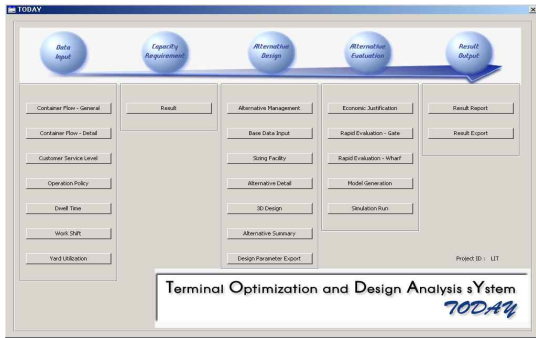
[그림 6] 장비 라이브러리의 클래스 다이어그램

4. 설계 지원시스템

본 연구에서는 컨테이너 터미널의 설계 프로세스를 지원하는 소프트웨어 시스템 *TODAY*(Terminal Optimization, Design, and Analysis system)를 개발하였다.

4.1 시스템 구성

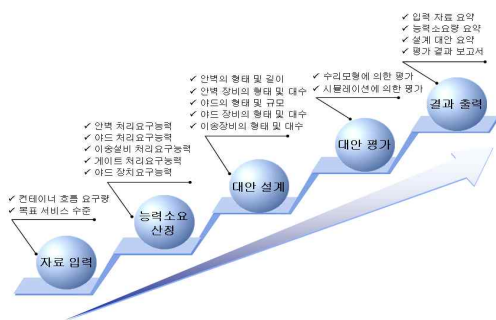
*TODAY*는 컨테이너 터미널의 설계 절차에 따라서 구성되며, 모든 절차가 통합된 시스템 안에서 이루어진다는 특징을 가진다. [그림 7]은 *TODAY*의 메인 인터페이스 화면이다. 메인 인터페이스의 버튼을 통해서 모든 기능이 사용 가능하다.



[그림 7] TODAY의 메인 인터페이스

4.2 설계 절차

컨테이너 터미널의 설계는 다음의 이유 때문에 귀납적인 접근이 바람직하다(Versteegt, 2004). 첫 번째, 컨테이너 터미널은 물류에 관련된 다양한 분야들로 구성되어 있다. 조직적인 면, 경제적인 면, 법적인 면, 기술적인 면 등이 모두 중요한 역할을 하기 때문에 여러 전문 분야에 걸친 접근이 필요하다. 두 번째, 컨테이너 터미널을 설계하는 것은 인정된 이론이 거의 없는 새로운 연구 분야이다. 따라서 연역적인 방식으로 컨테이너 터미널을 설계하는 것은 힘들거나 불가능할 수 있다. 마지막으로, 컨테이너 터미널에서 발생하는 다양한 문제점들을 모두 해결하는 단일한 해는 존재하지 않을 수 있다. 따라서 가능한 해들의 집합이 필요하다.



[그림 8] 컨테이너 터미널의 설계 절차

본 연구에서 제시하는 설계 절차는 [그림 8]과 같다. 이 절차는 귀납적인 접근을 기초로 만들어졌고, 출력된 결과는 피드백 되어 필요한 선행 단계가 반복되게 된다. 각 단계별 세부적인 설명은 다음 절부터 이어진다.

<표 3> 주요 입력 자료

범 주	항 목
컨테이너 흐름	주요 기점간 연간 물동량 (TEU) 컨테이너 종류의 분포 (%) TEU 환산계수 피크(peak) 계수 컨테이너 유형별 체류시간 (일)
고객 서비스 수준	외부차량의 왕복시간 (분) 외부차량의 대기시간 (분) 선박의 왕복시간 (시간)
운영정책	무료 장치 기간 (일) 교대별 작업시간 (시간) 연간 작업일수 (일) 야드 이용률 (%)

<표 4> 주요 능력소요 산정 항목

범 주	항 목
안벽 처리능력	시간당 처리율 (개/시간)
야드 처리능력	시간당 해측 작업 처리율 (개/시간) 시간당 육측 작업 처리율 (개/시간) 시간당 냉동 컨테이너 처리율 (개/시간) 시간당 공 컨테이너 처리율 (개/시간) 시간당 위험 컨테이너 처리율 (개/시간) 시간당 비규격 컨테이너 처리율 (개/시간)
게이트 처리능력	시간당 진입 외부차량의 수 (대/시간) 시간당 출입 외부차량의 수 (대/시간)
야드 장치능력	수입 컨테이너 야드 슬롯의 수 (TEU) 수출 컨테이너 야드 슬롯의 수 (TEU) 환적 컨테이너 야드 슬롯의 수 (TEU) 냉동 컨테이너 야드 슬롯의 수 (TEU) 공 컨테이너 야드 슬롯의 수 (TEU) 위험 컨테이너 야드 슬롯의 수 (TEU) 비규격 컨테이너 야드 슬롯의 수 (TEU)

4.2.1 자료 입력

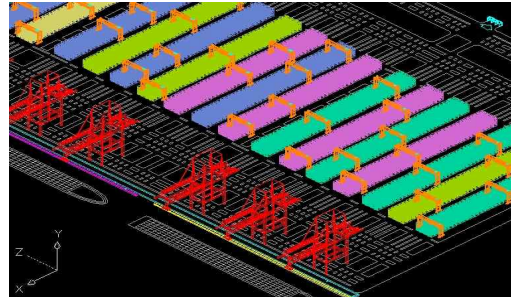
컨테이너 터미널의 설계 대안을 작성하기 위하여 시설 또는 장비에 대한 요구사항을 입력하는 단계이다. <표 3>은 주요 입력 자료를 나타낸다.

4.2.2 능력소요 산정

자료 입력 단계에서 입력된 자료를 사용하여 각 자원에 대한 능력을 계산하여 제시하는 단계이다. <표 4>는 주요 능력소요 산정 항목을 나타낸다.

4.2.3 대안 설계

사용자의 요구사항과 능력소요 산정 결과를 기초로 컨테이너 터미널에 대한 대안을 설계하는 단계이다. 사용자는 수많은 대안을 생성할 수 있지만, 능력소요 산정 결과는 요구사항을 만족하는 유효한 대안을 설계하도록 도와준다. <표 5>는 주요 설계 항목을 나타낸다. 설계 항목 가운데 일부는 데이터로 직접 입력되고, 일부는 CAD 환경에서 입력된 후 데이터 형태로 추출된다. [그림 9]는 3차원 CAD 환경에서 작성한 설계 대안의 예이다.

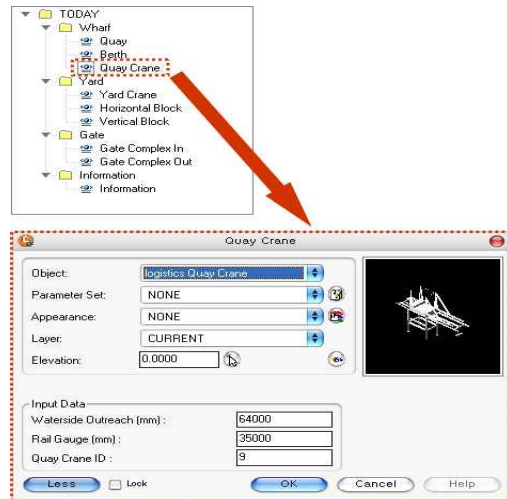


[그림 9] CAD 기반 설계대안 작성 예

CAD 환경에서 쉽고 빠르게 설계 대안을 작성하는 것을 지원하기 위하여 컨테이너 터미널 전용 라이브러리를 개발하였다. 컨테이너 터미널의 구성요소들을 *FactoryCAD 10.0*의 XML object를 사용하여 구축하였다. 모든 구성요소들은 3차원으로 구축되었으며 기반이 되는 환경은 *Windows*와 *AutoCAD 2005*이다. [그림 10]은 구축된 라이브러리의 구조와 하나의 대화창 예를 보여준다. 라이브러리에서 생성하고자 하는 요소를 선택하면 대화창이 열리고, 대화창에서 속성을 입력한 후 도면에 개체를 놓으면서 설계를 진행하게 된다.

<표 5> 주요 설계 항목

범 주	항 목
안 벽	선석의 유형, 개수, 길이 안벽 장비의 유형, 대수, 사양
야 드	야드의 유형, 면적 야드 블록의 개수 블록별 베이, 열, 단의 개수 야드 장비의 유형, 대수, 사양
이송영역	레인의 개수, 폭 이송장비의 유형, 대수, 사양
게이트	게이트의 유형 게이트 진입 레인의 개수 게이트 출입 레인의 개수



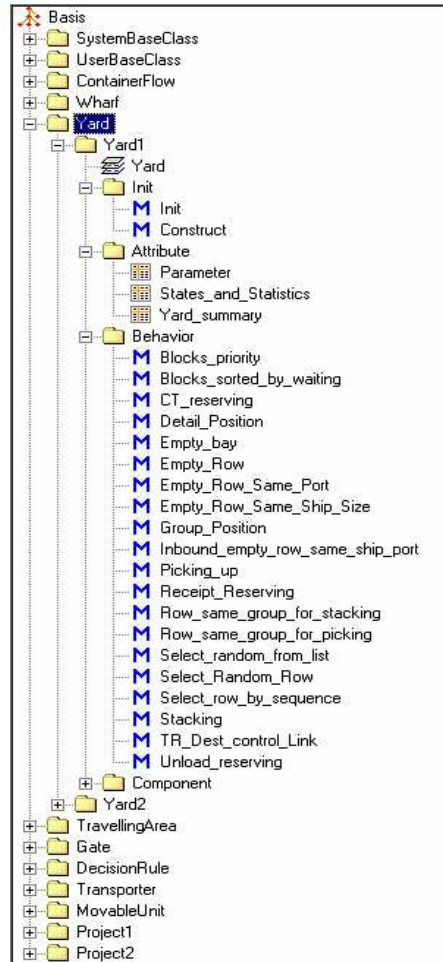
[그림 10] CAD 라이브러리

CAD 도면은 설계에 유용한 많은 자료를 포함하고 있다. 일부 자료는 쉽게 데이터화가 가능하지만, 일부 자료는 내부 데이터베이스에 접근한 후 일정한 처리를 거쳐야만 원하는 데이터로 얻어진다. 이러한 과정이 *AutoLISP*으로 프로그래밍되어 빠르고 정확한 데이터 추출을 가능하게 한다. 또한 추출된 데이터는 통합 데이터베이스에 설계 정보로 저장되어 후속 단계에서 활용할 수 있다.

4.2.4 대안 평가

설계된 대안의 경제성과 성능을 평가하는 단계이다. 성능의 평가는 수리모형과 시뮬레이션에 의하여 평가된다. 보다 상세한 평가를 위해서는 시뮬레이션을 사용한다.

시뮬레이션 모형은 저장된 데이터베이스의 설계 정보를 사용하여 자동으로 생성된다. 이를 위해서 모형 생성에 필요한 모든 파라메타가 데이터베이스에 저장되어 있어야 하고, 시뮬레이션 모형의 구성요소가 라이브러리로 구축되어 있어야 한다. [그림 11]은 시뮬레이션 라이브러리 가운데 야드 구성요소를 보여준다. 시뮬레이션 라이브러리는 *eM-Plant 7.5*를 사용하여 개발되었다.



[그림 11] 시뮬레이션 라이브러리

<표 6> 시뮬레이션 라이브러리의 구성요소

구성요소	설 명
<i>ContainerFlow</i>	컨테이너 흐름의 모형화
<i>Wharf</i>	안벽의 모형화
<i>Yard</i>	야드의 모형화
<i>TravellingArea</i>	이송영역의 모형화
<i>Gate</i>	게이트의 모형화
<i>DecisionRule</i>	의사결정 규칙의 모형화
<i>Transporter</i>	장비의 모형화
<i>MovableUnit</i>	이동하는 개체의 모형화

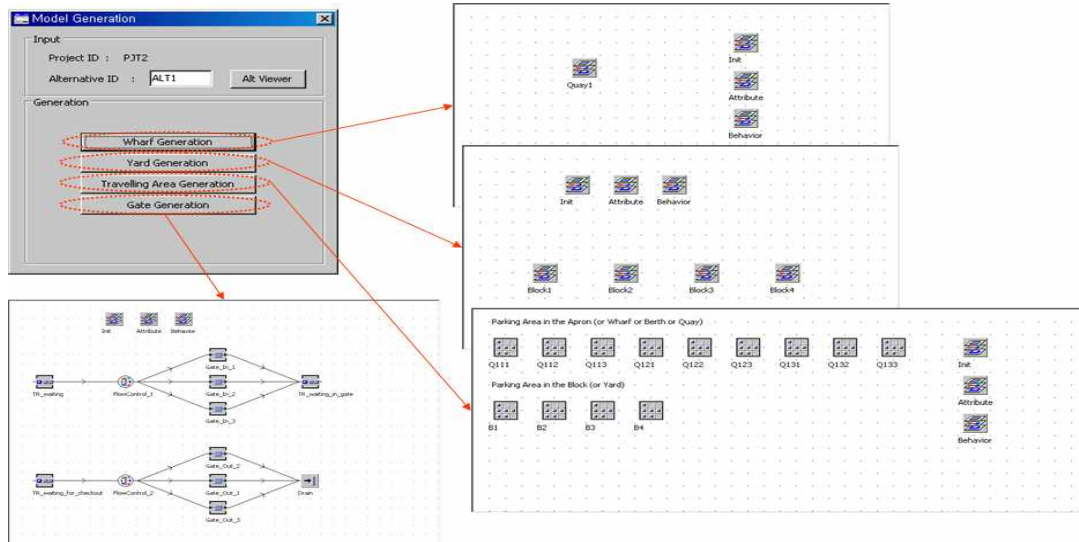
시뮬레이션 라이브러리의 구성요소에 대한 설명이 <표 6>에 제시된다. 각 구성요소 내에는 다양한 유형을 표현할 수 있도록 하위 구성요소가 존재할 수 있다. 예를 들어 [그림 11]의 *Yard1*은 수평형 야드를 모형화 하는데 사용되고, *Yard2*는 수직형 야드를 모형화 하는데 사용된다. 결국 각 구성요소의 조합으로 컨테이너 터미널 모형이 완성된다. 각 구성요소는 <표 7>과 같은 구조를 사용한다.

<표 7> 시뮬레이션 라이브러리 구성요소의 구조

항 목	설 명
Init	각 구성요소의 초기화와 관련된 메소드를 정의
Attribute	각 구성요소가 가지는 속성들을 정의
Behavior	각 구성요소가 가지는 메소드를 정의(초기화 관련 메소드 제외)
Component	설계 파라메타에 따라서 시뮬레이션 모형을 구성하기 위한 요소들을 정의

[그림 11]의 구성요소 가운데 <표 6>에 설명되지 않은 *SystemBaseClass*는 시뮬레이션 패키지에서 제공하는 기본 클래스를 담고 있으며, *UserBaseClass*는 *SystemBaseClass*를 사용하여 추가로 개발된 기본 클래스를 담고 있다. *Project1*, *Project2* 등의 폴더는 시뮬레이션 모형이 생성되는 작업 영역에 해당된다.

[그림 12]는 데이터베이스의 설계 정보와 시뮬레이션 라이브러리를 사용하여 자동으로 생성되는 시뮬레이션 모형의 예를 보여준다.



[그림 12] 시뮬레이션 모형 생성의 예

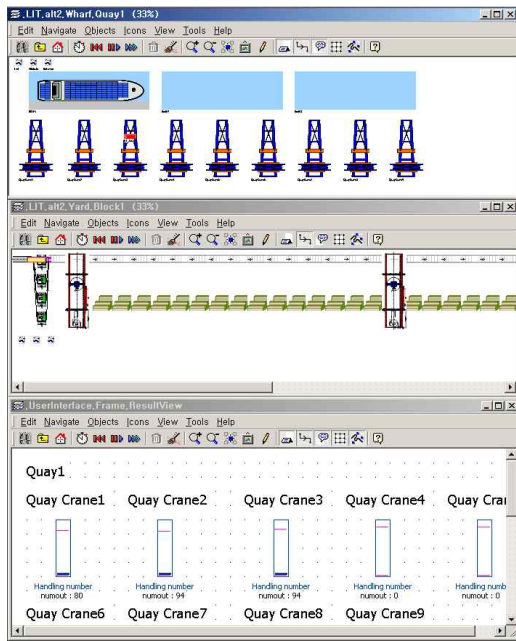
[그림 13]은 시뮬레이션 실행화면을 보여준다. 위에서부터 차례대로 안벽 모형, 야드 모형, 진행 상황 모니터링을 나타낸다.

4.2.5 결과 출력

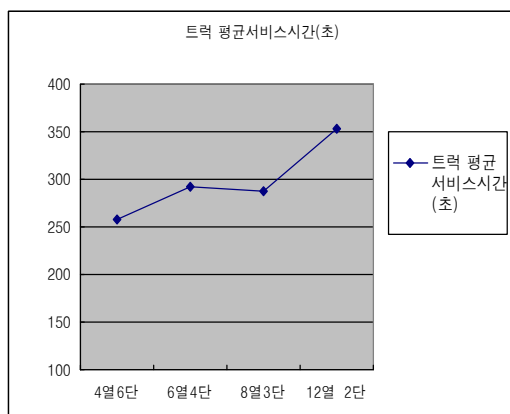
설계된 대안들의 능력 및 평가에 대한 최종 결과를 출력하는 단계이다. 만약 요구되는 능력과

차이가 있을 경우에는 다른 대안들을 모색해야 한다. 주요 출력 항목으로는 사용자 입력 자료, 능력 소요 산정 결과, 설계 대안의 요약, 설계요소별 성능평가 결과, 수리모형 성능 평가 결과, 시뮬레이션 성능 평가 결과 등이 있다. [그림 14]는 시뮬레이션 결과 얻을 수 있는 통계치의 하나를 예시하고 있는데, 서로 다른 블록의 사양을 사용한 터미

널의 장치장에 대해서 트럭에 대한 서비스시간을 보여 주고 있다. 그 결과 6단4열의 장치장이 가장 최단의 트럭 서비스시간을 제공함을 알 수 있다.



[그림 13] 시뮬레이션 실행 화면



[그림 14] 다양한 블록사양에 대한 트럭서비스시간의 예

4.3 가시화 시스템 구현

본 연구에서는 항만 터미널 시뮬레이션 결과를 검증하기 위하여 3차원 항만 물류 가시화 시스템인 ViCAL(Virtual Complex for Advanced Logistics)을 개발하였다. ViCAL은 시뮬레이션 시스템에 종속적인 기존의 가시화 시스템과 달리, 표준 스크립트(standard script) 파일을 통해 시뮬레이션 시스템과 독립성을 추구하여, 본 논문에서 소개하고 있는 TODAY 시스템뿐만 아니라 다른 시뮬레이션 시스템과 연동이 가능한 장점을 가진다.

4.3.1 세부 기능

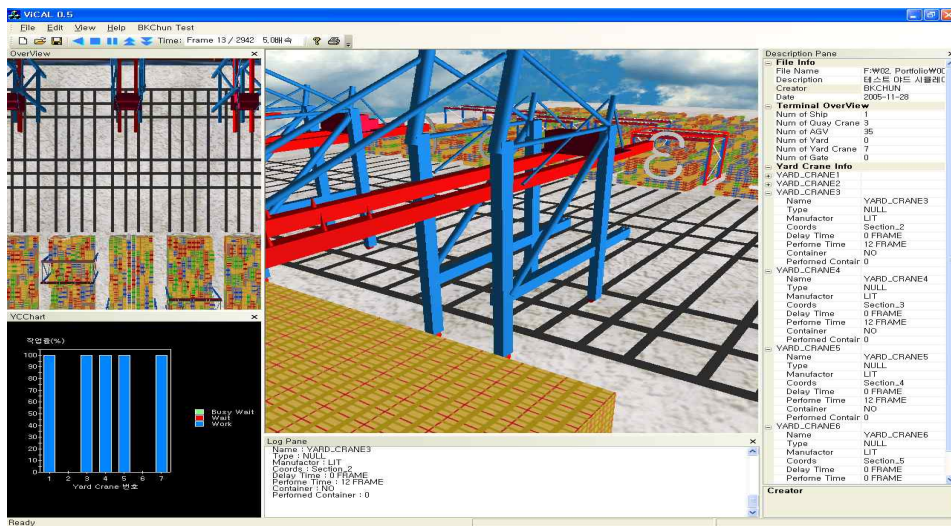
ViCAL 시스템은 표준 스크립트 파일을 통한 시뮬레이션 시스템과의 독립성 추구를 외에도 다양한 특징을 가진다. [그림 15]는 ViCAL 시스템의 실행 화면을 보여준다. 그림에서 보듯이 ViCAL 시스템은 3차원 가상 환경 뷰(view) 뿐만이 아니라 다양한 기능을 위한 뷰를 제공한다. ViCAL 시스템의 세부적인 기능들을 정리하면 다음과 같다.

- (a) ViCAL 시스템은 컨테이너 터미널 시스템 뿐만 아니라 다른 분야의 시뮬레이션 시스템의 결과를 표준 스크립트 파일을 통해 가상 환경을 구축하여 3차원으로 가시화 해준다.
- (b) ViCAL 시스템은 구축된 3차원 가상 환경을 사용자 상호작용을 통하여 사용자가 가상 환경을 향해할 수 있는 기능을 제공한다. 이 기능을 통해 사용자가 현재 보기를 원하는 객체, 예를 들어, 야드 크레인의 시뮬레이션 결과를 검증하고 싶다면 해당 지점으로 이동하여 이를 보고 검증할 수 있다.

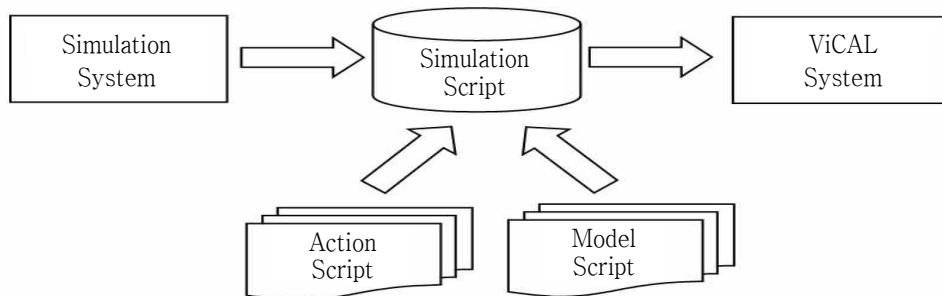
- (c) ViCAL 시스템은 시뮬레이션 진행 상황을 사용자 시점뿐만 아니라 다양한 객체들의 시점을 통해 시뮬레이션 진행 상황을 볼 수 있다. 예를 들어, 사용자가 AGV의 시점으로 시뮬레이션 진행 상황을 보기를 원한다면 AGV에 카메라를 설정함으로써 이를 수행할 수 있다.
- (d) ViCAL 시스템은 시뮬레이션 진행 상황을 애니메이션 형태로 보여주는데, 이를 위한 재생, 정지, 일시정지 등의 컨트롤 기능을

제공할 뿐만 아니라 다양한 속도의 재생 기능을 제공한다.

- (e) ViCAL 시스템은 가시화 기능뿐만 아니라 현재 로드된 시뮬레이션 모델의 전반적인 내용들을 사용자에게 보여주는 기능을 제공한다. 또한, 3차원 가상 환경에서 픽킹(picking) 기능을 통해 사용자가 원하는 객체(예를 들어, 야드 크레인)의 현재 상태를 보여주며, 객체의 작업율을 그래프를 이용하여 보여준다.



[그림 15] ViCAL 시스템의 실행 화면



[그림 16] ViCAL 시스템과 시뮬레이션 시스템과의 관계도 및 표준 스크립트 파일

4.3.2 시뮬레이션 시스템과의 인터페이스

ViCAL 시스템은 시뮬레이션 시스템에서 생성된 시뮬레이션 결과과의 연동을 위해 simulation script, action script, model script의 총 3개의 표준 스크립트 파일을 사용한다. [그림 16]은 ViCAL 시스템과 시뮬레이션 시스템간의 관계도를 보여준다. 그림에서 보듯이 각각의 표준 스크립트 파일의 목적은 다음과 같다.

- (a) Model script : Model script는 시뮬레이션 시스템에서 각 객체의 3차원 모델 데이터를 정의한 파일이다. 예를 들어, 항만 물류 시스템인 TODAY 시스템에서는 컨테이너, 안벽 크레인 AGV(Automatic Guided Vehicle) 과 같은 객체들의 3차원 모델 데이터라고 할 수 있다. 따라서, model script를 수정함으로써 원하는 모양의 객체를 보여줄 수 있다. 이 파일은 3차원 가상 환경 그래픽 표준 파일인 VRML(Virtual Reality Modeling Language)를 사용하여 정의한다.
- (b) Action script : Action script는 model script와 연동되는 파일로, 각 model script가 표현하는 객체의 움직임을 정의한 파일이다. 예를 들어, 안벽 크레인의 action script 파일은 안벽 크레인이 취할 수 있는 행위들을 기술한 파일이다. 새로운 타입의 장비가 개발되었을 때, 사용자가 보조 툴을 사용하여 장비의 행위를 기술함으로써 다양하게 확장이 가능하다.
- (c) Simulation script : Simulation script는 시뮬레이션 시스템에서 수행한 시뮬레이션 모델의 시간별 진행 상황을 기술한 파일이다. ViCAL 시스템은 최종적으로 이 파일을 로드하여 시간별 각 객체들의 시뮬레이션

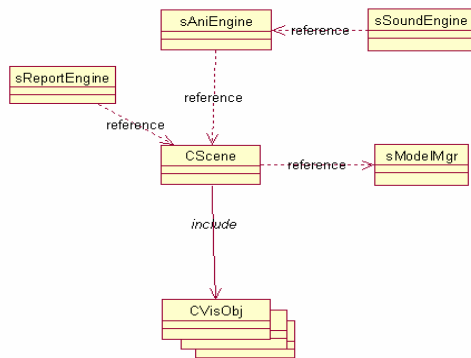
진행 상황 데이터를 분석하여 3차원 가상 환경에 렌더링 해준다.

4.3.3 클래스 관계도

[그림 17]은 ViCAL 시스템의 클래스 관계도를 보여준다. ViCAL 시스템 설계시 클래스는 크게 2가지 타입으로 나누어 설계하였다. 하나는 *sClass*로 이 클래스는 정적 클래스(static class)를 의미하고, 다른 하나는 *cClass*로 개체(concrete class)를 의미한다. *sClass* 타입의 클래스의 목적은 렌더링 엔진, 사운드 엔진과 같이 전반적인 시스템의 운영을 담당하는 클래스로, 하나의 인스턴스(instance) 만을 생성하여 가지고 있다. *sAniEngine*, *sSoundEngine*, *sModel Mgr*, *sReportEngine* 클래스가 이 타입에 속한다. *cClass* 타입의 클래스는 시뮬레이션 모델 내에서 실제적인 객체를 나타내는 클래스로, 다수의 인스턴스를 생성하여 가질 수 있다. 예를 들어, 안벽 크레인을 나타내기 위한 부모 클래스인 *CVisObj*가 여기에 속한다. 각 클래스의 목적은 다음과 같다.

- (a) Class : *sAniEngine*
sAniEngine 클래스는 시뮬레이션 시스템에서 생성한 simulation script를 로드하여 3차원으로 가시화 해주는 역할을 담당한다. 이 클래스는 *CScene*을 참조하여 해당 객체를 렌더링한다.
- (b) Class : *sSoundEngine*
sSoundEngine 클래스는 시뮬레이션 진행 시간에 따라 음향을 재생하는 기능을 한다. 따라서 이 클래스는 *sAniEngine*을 참조한다.
- (c) Class : *sModelMgr*
sModelMgr 클래스는 model script와 action

script를 CVisObj 클래스 형태로 로드하고 이를 관리하는 클래스이다. sModelMgr 클래스로 객체들을 관리함으로써 객체들이 중복으로 로드되는 것을 막는 장점을 가진다.



[그림 17] ViCAL 시스템의 클래스 관계도

(d) Class : sReportEngine

sReportEngine 클래스는 시뮬레이션 모델에서 특정 객체에 대한 정보를 사용자에게 보여주는 클래스이다. 이를 위해 전체 시뮬레이션 모델의 내용을 가지고 있는 CScene 클래스를 참조한다.

(e) Class : CScene

CScene 클래스는 로드된 시뮬레이션 모델이 저장되는 클래스로, sModelMgr를 참조하여 필요한 객체의 3차원 모델을 CVisObj로 저장한다.

(f) Class : CVisObj

CVisObj 클래스는 각 객체를 표현하는 추상 클래스(abstract class)로, 각 객체는 하나의 CVisObj로 표현된다. 예를 들어, 안벽 크레인의 model script와 action script는 결국 CVisObj로 표현되며, 이는 sModelMgr와 CScene 클래스에서 사용된다.

4.3.4 개발 환경

시스템 개발에 있어서 개발 언어와 라이브러리 선택은 상당히 중요하다. 잘못된 개발 언어와 라이브러리 선택은 시스템 성능의 저하 및 유지 보수 비용 증가로 이어질 수 있다. ViCAL 시스템은 이를 고려하여 개발 환경을 선택하였으며, 이는 <표 8>과 같다.

<표 9> ViCAL 시스템의 개발 환경

DEVELOPMENT LANGUAGE	C++
APPLICATION TYPE	MFC(Microsoft Foundation Classes) application
GRAPHICS LIBRARY	Coin3D
GUI LIBRARY	Xtreme Toolkit Pro
OPERATING SYSTEM	Microsoft Windows

<표 8>에서 보듯이, ViCAL 시스템은 Microsoft Windows 기반의 실행 환경에, 객체 지향적인 언어인 C++을 이용하여 개발하였다. 그리고 그래픽 라이브러리는 OpenGL 기반의 Coin3D를 이용하여 개발하였으며, 그래픽 유저 인터페이스(GUI: Graphics User Interface)로 Xtreme Toolkit을 사용하였다.

5. 결론

본 연구에서는 컨테이너 터미널의 설계 프로세스를 지원하는 TODAY를 개발하였다. 이를 위해 설계 절차를 제안하였고, 시스템 개발을 위해서

데이터 및 프로세스를 분석하고 설계하였다. 또한 사용자가 대안 설계와 평가를 신속하게 하도록 지원하는 CAD 및 시뮬레이션 라이브러리를 개발하였다.

시스템의 전체 구성이 보다 통합되고, 사용자에게 보다 편리한 인터페이스 개발이 필요할 것이다. 컨테이너 터미널 전용으로 개발된 CAD 및 시뮬레이션 라이브러리가 추후 새로운 유형의 시설, 레이아웃, 장비 및 운영논리의 개발에 맞춰 지속적으로 갱신된다면 컨테이너 터미널의 설계를 위한 주요 지원도구로 자리 잡을 것이다.

본 연구에서 개발된 시스템은 터미널의 설계를 위하여 개발되었다. 그러나 터미널의 시뮬레이션을 위하여 시뮬레이션 프로그램의 내부에 다양한 운영로직들이 구현되어 있다. 터미널의 성능은 이런 운영로직의 영향을 많이 받을 수 밖에 없다. TODAY의 성능 향상을 위해서는 이런 운영논리의 개선이 추후 계속되어야 할 것으로 보이며 그 과정에서 실제 운영시스템의 구축에 도움이 되는 운영논리의 개발도 부수적으로 이루어질 것으로 보인다. 본 연구에서 개발된 시뮬레이션 시스템은 현재까지 여러 현실적인 시스템의 성능추정에 활용되었으며 이를 통하여 기능의 보완이 이루어져 왔다. 향후 추가적으로 시스템이 활용되면서 프로그램의 타당성에 대한 점검과 수정이 추가적으로 뒤따라야 할 것으로 사료된다.

참고문헌

- [1] 김현, *시뮬레이션에 의한 부산 컨테이너 터미널 운영의 체계적인 분석*, 석사학위논문, 한국해양대학교, 1988.
- [2] 장성용, 박진우, “시뮬레이션 기법을 이용한 컨테이너 터미널의 운영시스템 결정”, *산업공학*, 1권 1호(1988), 49-62.
- [3] 조덕운, “컨테이너항 전산 모의실험 모형의 개발”, *대한산업공학회지*, 11권 2호(1985), 173-187.
- [4] 한국컨테이너부두공단, *세계 주요항만 2004년도 물동량, 시설, 개발계획 현황 및 분석*, 2005.
- [5] Ramani, K. V., “An Interactive Simulation Model for the Logistics Planning of Container Operations in Seaports”, *Simulation*, Vol.66 No.5(1996), 291-300.
- [6] Saanen, Y. A., *An Approach for Designing Robotized Marine Container Terminals*, Ph. D. Thesis, Delft University of Technology, 2004.
- [7] Tahar, R. M. and Hussain, K., “Simulation and Analysis for the Kelang Container Terminal Operations”, *Logistics Information Management*, Vol.13 No.1(2000), 14-20.
- [8] Versteegt, C., *Holonic Control for Large Scale Automated Logistics Systems*, TRAIL Thesis Series T2004/9, The Netherlands, 2004.
- [9] Whitten, J. L., Bentley, L. D., and Dittman, K. C., *System Analysis and Design Methods*, Irwin/McGraw-Hill, New York, 2000.
- [10] Yun, W. Y. and Choi, Y. S., “A Simulation Model for Container-Terminal Operation Analysis Using an Object-Oriented Approach”, *International Journal of Production Economics*, Vol.59 No.1-3(1999), 221-230.
- [1] 김현, *시뮬레이션에 의한 부산 컨테이너 터미*

Abstract

A Decision Support System for Designing Container Terminals

Seung Hwan Won* · Bong Kyung Chun** · Su Min Jeon* · Byung Kwon Lee* ·
Dong Won Jang* · Hwan Gue Cho** · Kap Hwan Kim*

This study introduces a software system for supporting the engineering process of container terminals. The design procedure consists of data input, capacity requirements, alternative design, alternative evaluation, and results output. In order to design and evaluate alternatives rapidly, a set of libraries for container terminals was developed. The libraries consist of Wharf, Yard, TravellingArea, and Gate. The architecture of the set of libraries was designed by an object-oriented approach. Various types of container terminal models can be constructed by using a user friendly interface, the CAD based environment, and the combinations of the elements of the libraries. Alternative models can be evaluated by a simulation.

Key Words : Container Terminal, Logistics Systems Design, Decision Support System

* Department of Industrial Engineering, Pusan National University

** Department of Computer Engineering, Pusan National University