

컨테이너 터미널 운영 분석을 위한 형식 프레임워크

A Formal Framework for Analyzing Performance of Container Terminal Operations

박은정(Eun-Jung Park)*, 하병현(Byung-Hyun Ha)**

초 록

정보 기술이 적용된 장비 및 운영 시스템의 발달로 실시간 운영 데이터를 수집할 수 있게 되었으며, 현재 많은 터미널이 생산성 분석을 위해 상세한 작업이력을 추적하고 있다. 터미널 운영자는 수집된 작업이력 데이터를 통해 운영 중 발생하는 문제점들을 해결할 수 있는 방안을 도출하기 원하지만, 방대한 양의 자료를 수작업으로 분석하기는 쉽지 않다. 본 연구에서는 터미널 운영 중 획득된 작업이력을 바탕으로 터미널 운영자가 원하는 컨테이너 터미널의 성능지표를 손쉽게 도출하기 위한 방안을 제시한다. 이를 위해 다양한 컨테이너 터미널의 환경과 운영 방식, 운영 장비 등을 지원하기 위해 범용적으로 사용될 수 있는 형식 프레임워크(formal framework)를 고안하였다. 본 연구에서 제안하는 프레임워크는 컨테이너 처리 객체라는 개념을 기반으로 터미널 형태, 로그 데이터, 장비의 워크플로우, 통계 처리 및 보고서를 각각 정의하는 사용자 명세 부분으로 구성된다. 그리고 제안된 프레임워크를 기반으로 터미널 성능 분석 시스템을 개발하여 그 유효성을 검증하였다.

ABSTRACT

Today, information technologies have been applied to operations in container terminals, and it is possible to collect operational log data due to development of equipment and operations technology. Terminal operators are collecting event log data and try to figure out the way of resolving operations problems. Operators want to analyze event logs to determine the causes of the operation problems, but it can hardly be done manually. In this paper, we suggest a formal framework to evaluate performance measures using the collected log data of operations in container terminals. The proposed formal framework supports different container terminal layout, operational processes, and equipment. Our formal framework is composed of specification of terminal layout, log data, workflow, statistics, and report, based on the concept of container handling objects. For validation of our framework, we have implemented a terminal performance analysis system based on the proposed framework.

키워드 : 컨테이너 터미널, 운영 이력, 성능 지표, 형식 프레임워크

Container Terminal, Operations History, Performance Measure, Formal Framework

본 연구는 2013년 정부(교육부)의 지원을 받아 수행하였음(산학협력 선도대학(PNU-LINC) 육성사업).

* Ph.D. Candidate, Dept. of Logistics and Information Technology, Pusan National University

** Corresponding Author, Assistant Professor, Dept. of Industrial Engineering, Pusan National University
(E-mail : bhha@pusan.ac.kr)

2013년 05월 06일 접수, 2013년 05월 14일 심사완료 후 2013년 05월 15일 게재확정.

1. 서 론

컨테이너 터미널은 수출입 화물을 컨테이너 선박에 싣고 내리는 양하와 적하 작업을 수행하는 부두를 말한다. 많은 컨테이너들을 빠른 시간에 양적하 하기 위해서 다양한 장비들이 사용된다. 투입된 장비들의 대수와 그들의 상호 작용, 터미널 운영방법 등의 영향으로 장비 간의 간섭 및 고장, 특정 지점에서의 병목 등 다양한 문제점들이 발생하게 된다. 터미널 운영 중 발생하는 이와 같은 문제점들은 터미널의 생산성을 떨어뜨리는 원인으로 작용한다. 특히 터미널 생산성의 저하는 여러 문제가 복합적으로 작용하여 일어나는 경우가 많으며, 터미널 운영자가 터미널 생산성의 감소 원인을 파악하는 것이 쉽지 않다.

컨테이너 터미널 운영자는 문제 발생 시 원인 파악을 위해 작업이력 데이터를 획득하고 터미널 운영 성능지표를 분석하기를 원한다. 하지만 일반적으로 막대한 양의 작업이력이 수집되며, 이러한 데이터를 수작업으로 분석하기란 쉽지 않다. 이와 같은 어려움을 극복하기 위하여, 본 연구에서는 터미널의 작업이력을 바탕으로 각 장비의 작업을 분석하고, 그를 바탕으로 터미널의 운영 성능지표를 효과적으로 도출할 수 있는 형식 프레임워크(formal framework)를 제안한다.

현재 컨테이너 터미널의 운영과 관련된 연구는 대부분 터미널의 운영능력을 산정하는 방법이나 터미널의 효율성 평가방법에 초점이 맞춰져 있다. 김선구와 최용석은 DEA 및 AHP 통합모형을 적용하여 하역장비에 따른 터미널의 효율성을 비교 평가하였으며[6], 이훈은 컨테이너 터미널 운영 중 발생된 문제

의 인과관계를 신속하게 확인할 수 있도록 사례 기반의 사용자 의사결정지원 시스템을 설계하는 방안에 대해 연구하였다[7]. 하지만 특수한 목적을 위한 방법론으로 운영 성능을 평가하기 위해 일반화시키기는 쉽지 않다.

운영 성능을 평가하기 위한 연구는 생산 제조 시스템 분야에서 활발히 이루어지고 있다. Muthiah et al.은 자동화 공장의 생산 효율성을 평가하기 위하여 OTE(overall throughput effectiveness) 지표를 제안하고, 자동화 공장의 하부조직을 직렬, 병렬, 조립, 확장으로 구분하고, 공장 레벨 성능(factory-level performance) 개념 및 OTE를 적용하여 공장의 생산능력을 평가했다[8]. Beamon은 제조 환경에서의 물류 처리 시스템을 분석하기 위해 사용되는 다양한 성능지표를 구하는 방법에 대해서 연구하고, 제조 시스템 전반에 걸친 작업 생산성을 비롯하여 이송장비의 이동거리 등과 같은 요소를 통해 시스템의 성능을 평가하였다[2].

Comuzzi et al.은 변화하는 비즈니스 네트워크 내에서 프로세스를 모니터링 할 수 있는 능력을 유지하는 문제점을 해결하기 위한 형식 프레임워크를 제안하였다[3]. 이들은 우선 비즈니스 네트워크, 계약, 모니터링 요구사항을 형식 모델을 통해 제안한 후, 다른 형태의 네트워크 변화를 모니터링 할 수 있도록 네트워크 진화에 대한 방법을 모델링하는 프레임워크를 제안하였다. 안경림 등은 화물 추적 업무에 있어서 단일화되고 통합된 비즈니스 협업을 위한 비즈니스 프로세스를 정의하고, 그 결과를 이용해 표준 데이터를 정의하였다. 또한 개발된 프로세스 모델과 표준 데이터가 상호운용성을 지원할 수 있도록 하여 수출입 화물에 대한 흐름파악이 용이하도록

록 하였다[1]. 김훈태와 이용한은 RFID 태그를 부착한 부품의 워크플로우 모델링 방법을 연구하고, RFID 시스템의 모델링 결과를 이용하여 정의된 워크플로우를 따라 이동하는 태그의 정확한 흐름 현황을 시스템에서 모니터링하고 판정하기 위한 알고리즘을 제안하였다[4].

본 연구에서 제안하는 형식 프레임워크는 다양한 형태의 컨테이너 터미널 레이아웃 및 운영방법, 운영장비 등을 지원할 수 있도록 개발하였다. 형식 프레임워크는 사용자의 직접 입력을 통해 작업이력 데이터를 쉽게 처리할 수 있도록 지원함으로써 범용성과 활용도를 높였다. 성능지표의 도출 결과는 컨테이너 터미널 운영 중 발생하는 문제의 원인을 규명하는 정보로 활용할 수 있으며, 터미널의 운영 문제를 개선하는데 도움을 줄 수 있을 것으로 기대된다.

본 논문의 제 2장에서는 컨테이너 터미널 운영의 다양성을 소개하였으며, 제 3장에서는 본 연구에서 제안하는 형식 프레임워크에 대해 기술하였다. 제 4장에서는 사용자들 위해 제공되는 사용자 정의 언어에 대해 설명하였으며, 제 5장에서는 본 연구에서 제안한 형식 프레임워크의 검증을 위한 프로토타입 개발 및 구현 결과에 대해 설명하였다. 본 연구에 대한 결론은 마지막 제 6장에 기술하였다.

2. 연구 배경

컨테이너 터미널은 컨테이너 선박이 접안하여 작업을 하는 선석(berth)과 컨테이너를

보관하는 장소인 장치장(yard), 컨테이너 터미널과 내륙을 연결하는 지점인 게이트(gate)로 구성되어 있다. 터미널의 작업 영역에 따라 사용하는 장비의 종류가 다양하다. 선석에서 컨테이너 선박에 컨테이너를 양적하 하는 안벽 크레인(quay crane)이 있으며, 야드 크레인(yard crane)은 장치장에서 컨테이너를 쌓거나 꺼내는 작업을 수행한다. 야드 트랙터(yard tractor)와 셔틀 캐리어(shuttle carrier)는 선석과 장치장을 오가며 컨테이너를 이송하는 대표적인 장비이다.

이와 같은 장비의 종류 및 특성에 따라 터미널을 운영하는 방법이 달라진다. 예를 들어, 선석에서 작업하는 안벽 크레인의 경우, 20피트 컨테이너를 동시에 두 개 들 수 있는 트윈-리프트(twin-lift) 크레인인 경우, 40피트 컨테이너를 동시에 두 개까지 들 수 있는 탠덤-리프트(tandem lift) 크레인이 있다. 두 크레인의 경우, 처리할 수 있는 컨테이너의 규격 및 수량이 다르기 때문에 크레인을 운영하는 방안이 달라지며, 이에 따라 분석도 달라져야 한다. 그리고, 장치장에서 작업하는 RTGC (rubber tired gantry crane)의 경우 블록간 이동이 가능하지만, ASC(automated stacking crane)의 경우, 레일 위에서 움직이므로 블록간 이동이 불가능하며, 전원 공급 케이블의 길이 제한으로 인해 블록 내에서도 이동할 수 있는 거리가 제한적이다. 또한 선석과 장치장 사이를 오가며 컨테이너를 이송하는 AGV(automated guided vehicle)의 경우 자동으로 컨테이너를 이송하지만, 컨테이너를 직접 들고 내릴 수 없으며, 셔틀 캐리어는 컨테이너를 직접 들고 내릴 수 있다.

컨테이너 터미널은 다양한 배치 형태가 존재

하며, 이 또한 터미널을 운영하는 방법에 영향을 미치는 요소로 작용한다. 컨테이너 터미널의 레이아웃은 크게 수직과 수평으로 구분된다 [5]. 장치장의 레이아웃에 따라 각 블록에서 컨테이너를 교환하는 TP(transfer point) 위치 및 개수가 달라지며, 이는 터미널 전체의 운영방법에 영향을 미치게 된다. 또한 안벽 크레인의 트윈/더블 작업 방식, 이송장비의 싱글/듀얼 사이클 작업 방식 등 터미널마다 운영방법이 다르다[9].

위와 같이 터미널 형태 및 운영방법, 운영장비가 상이한 다른 터미널의 분석에는 다른 방법이 요구된다. 따라서 다양한 컨테이너 터미널의 분석을 일관적으로 지원하기 위해서는 범용적인 형태의 프레임워크가 요구된다.

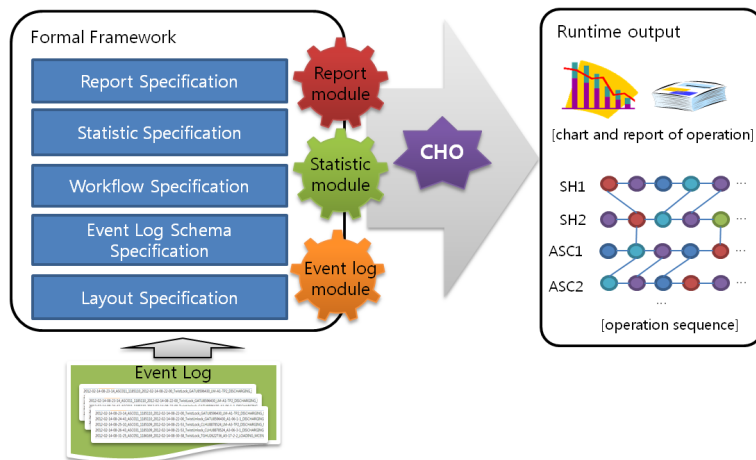
3. 형식 프레임워크

<Figure 1>은 본 연구에서 제안하는 터미널 운영 성능지표를 도출하기 위한 형식 프

레이워크의 구조를 보여준다. 다양한 컨테이너 터미널의 배치 형태 및 운영방안, 사용하는 장비 등을 지원하기 위하여, 본 프레임워크에서는 사용자가 직접 터미널의 배치형태 및 사용 장비의 워크플로우, 통계 수식 등을 직접 입력할 수 있다. 이를 통해 시스템의 유연성 및 확장성을 지원한다.

사용자는 본 연구에서 개발한 사용자 정의 언어를 사용하여 터미널의 배치(layout specification) 및 이벤트 로그 스키마(event log schema specification), 장비 워크플로우(workflow specification), 통계 항목(statistics specification), 보고서 형식(report specification)을 명세한다. 사용자가 정의한 명세 파일은 전용 컴파일러를 통해 실행 파일로 변환되며, 변환된 실행 파일은 엔진을 통해 실행되어 사용자가 정의한 보고서 및 각종 통계 정보를 생성한다.

터미널 레이아웃 명세를 통해 터미널의 배치형태 및 각 지점 사이의 거리 및 이동시간을 정의한다. 사용자가 입력한 거리 및 이동시간은 장비의 성능지표를 도출할 때 사용된다.



<Figure 1> Building Blocks of the Formal Framework

로그 이벤트 스키마 명세는 터미널 운영방법이나 사용하는 TOS(Terminal Operating System)의 종류에 따라 수집되는 이력 데이터 형태의 다양성을 지원한다. 사용자는 로그 파일 형태, 데이터의 필드 명, 데이터 유형 등을 지정할 수 있다. 워크플로우 명세는 각 장비의 특성을 명세 하는 부분과 장비의 운영 흐름을 명세 하는 부분으로 나누어져 있다. 통계 명세는 사용자가 원하는 통계 항목 및 수식을 기술하며, 목적에 따른 다양한 통계 정보를 획득할 수 있도록 지원한다. 분석 목적에 따라 다양한 보고서의 형식 및 차트 형태를 지원할 수 있어야 한다. 보고서 명세는 이와 같은 다양한 목적의 보고서를 지원하기 위해 보고서 형태 및 차트 형태를 정의할 수 있도록 지원한다.

또한 형식 프레임워크를 통해 터미널 운영장비의 작업 시퀀스를 도출하기 위해서 CHO(Container Handling Object)를 사용한다. CHO는 터미널 내에서 컨테이너를 취급하는 장비의 일반적인 모형으로, 각 장비의 작업능력, 워크플로우, 작업 시퀀스 정보를 표현한다. 본 연구에서는 터미널에서 컨테이너를 취급하는 장비의 동작을 크게 세 가지로 구분한다. 즉, 장비는 컨테이너를 집거나 싣는 동작, 컨테이너를 내리는 동작, 목표지점으로 이동하는 동작을 수행한다. 이를 위해 CHO는 in, out, move의 세 가지 메소드(method)를 지원한다.

4. 사용자 정의 언어

본 장에서는 사용자 명세 파일을 지원하는 사용자 정의 언어에 대해 설명한다.

4.1 터미널 레이아웃 명세

터미널 레이아웃 명세 언어는 이송지점 정의, 이동거리 정의, 이동시간 정의로 구성되어 있으며, 이들은 장비의 성능지표 도출을 위해 사용된다. <Figure 2>는 터미널 레이아웃 명세 언어의 문법을 BNF(Backus-Naur Form) 형식으로 보여준다. 사용자는 로그 데이터로 기록되는 이벤트 발생 지점명과 터미널 레이아웃 명세에서 사용할 이송지점 명을 <IDENTIFIER>로 정의한 뒤에, 각 이송지점 간 이동거리 및 이동시간을 <NUMBER> 요소, 즉, 숫자로 정의한다.

```

1. <specification> ::= 'Layout' '{' <description> '}'
2. <description> ::= <location-def> <distance-def> <traveling-time>
3. <location-def> ::= <location-expr> | <location-def> <location-expr>
4. <location-expr> ::= <IDENTIFIER> ':' <LocationID>
   | <IDENTIFIER> ':' '@realPoint' '('
   <LocationID> ')'
5. <distance-def> ::= 'distance' '(' <IDENTIFIER> ',' <IDENTIFIER>
   ')' '=' <NUMBER>
   | <distance-def> 'distance' '(' <IDENTIFIER>
   ',' <IDENTIFIER> ')' '=' <NUMBER>
6. <traveling-time> ::= 'trvTime' '(' <IDENTIFIER> ','
   <IDENTIFIER> ')' '=' <NUMBER>
   | <traveling-time> 'trvTime' '(' <IDENTIFIER>
   ',' <IDENTIFIER> ')' '=' <NUMBER>
    
```

<Figure 2> Syntax for Terminal Layout Specification Language

터미널 레이아웃 명세 언어는 각 지점간 이동거리를 구해주는 함수인 distance(from, to)와 이동시간을 구해주는 함수 trvTime(dep, arr)을 지원한다. 또한 상대적 위치정보를 절대적 위치정보로 변환해주는 realPoint(relativePoint) 함수를 제공한다. 이는 안벽 크레인과 같이 이동 가능한 장비를 기준으로 이송지점을 정의한 경우, 장비의 위치 변경으로 인한 실제 이동거리 및 이동시간이 변경되는 것을

지원하기 위한 것이다.

```

Layout {
  A1-Seaside-TP : LM-A1-TP1, LM-A1-TP2, LM-A1-TP3,
                 LM-A1-TP4
  A1-Landside-TP : RM-A1-TP1, RM-A1-TP2, RM-A1-TP3,
                  RM-A1-TP4
  Bitt[1-20] : @realPoint(STS102-Lane[1-4]),
               @realPoint(STS101-Lane[1-4])

  distance(A1-Seaside-TP, Bitt1) = 50
  trvTime(A1-Seaside-TP, Bitt1) = 3
}
    
```

<Figure 3> An Example of Terminal Layout Specification

<Figure 3>는 터미널 레이아웃 명세 언어로 기술한 예이며, 이송지점 명을 정의한 내용을 확인할 수 있다. 로그 데이터에 남겨진 'LM-A1-TP1' 등의 이벤트 발생지점은 모두 'A1-Seaside-TP'로 인식되어 이동거리 및 시간을 구하게 된다. 그 다음 'A1-Seaside-TP'와 Bitt1 사이의 거리 및 이동시간이 정의되어 있다.

4.2 이벤트 로그 스키마 명세

이벤트 로그 스키마 명세는 이벤트 로그에 남겨진 데이터 형태의 다양성을 지원한다. <Figure 4>는 그 문법을 보여준다.

```

1. <specification> ::= 'LogSchema'{'<description> '}'
2. <description> ::= <File-def> <Field-def>
3. <File-def> ::= <IDENTIFIER> '!' <FILENAME>
4. <Field-def> ::= <expr> | <Field-def> <expr>
5. <expr> ::= <FILENAME> '.split' '(' <IDENTIFIER> ')'
           | 'FieldList' ':' '[' <LIST> ']'
    
```

<Figure 4> Syntax for Event Log Specification Language

<Figure 5>는 이벤트 로그 스키마 명세의 예시이다. 로그 파일 'test_log.txt'을 사용할 것을 지정하였으며, 이벤트 로그의 데이터를 '_' 기호를 기준으로 분리하여, 각 데이터를 그 다음에 정의된 필드의 값으로 하나씩 인식한다. 예를 들어 '2012-02-14-10-56-39 _ STS101 _ TwistLock _ CRXU1745038 _ STS101-Lane3'과 같은 형태의 이벤트 로그를 '_'를 기준으로 나누어 'evt_Time', 'evt_Equip', 'evt_Name', 'evt_Container', 'evt_Location' 필드의 값으로 지정한다.

```

LogSchema {
  evtLogFile : test_log.txt
  evtLogFile.split('_')
  FieldList : ['evt_Time', 'evt_Equip', 'evt_Name',
               'evt_Container', 'evt_Location']
}
    
```

<Figure 5> An Example of Event Log Specification

4.3 워크플로우 명세

워크플로우 명세에서 사용되는 키워드는 <Table 1>과 같다. 키워드 capacity는 장비의 컨테이너 최대 처리 개수를 정의할 때 사용하는 것으로, 이벤트 로그의 오류를 발견하기 위해서 사용된다. 예를 들어 안벽 크레인이 최대 처리할 수 있는 컨테이너 수가 2개인데, 이벤트 로그의 오류로 인해 컨테이너의 처리 수가 2개 이상이 될 경우가 있다. 이와 같은 경우 안벽 크레인 이벤트 로그의 오류를 발견하게 된다. 키워드 workflow는 장비마다 상이한 워크플로우를 정의할 때 사용한다. 또한 Object는 터미널 운영 시 사용된 장비 종류별 장비 ID를 명세한다.

<Table 1> Keywords for Workflow Specification

Keyword	Remark
capacity	Maximum number of containers can be handled
workflow	Workflow ID of CHO
Object	Equipment ID of a specific type
Start	Start of workflow description
self	A CHO that handles the relevant container
in	Designation of container pick-up operation
out	Designation of container release operation
cont	The container owned by the relevant CHO
owner	The CHO that handled the relevant container previously

<Figure 6>는 장비 특성을 명세 하는 언어의 문법을 보여준다. 여기서 1~3은 장비 타입 별 특성을 지정하기 위해 사용되는 문법으로, CHO 키워드를 통해 타입 별 특성 명세임을 구분할 수 있다. 2에서 CHO 다음의 <CHO-ID>는 장비 타입 종류 이름을 의미한다. 사용자는 장비 타입 별 특성에서 각 장비의 컨테이너 최대 수용 개수 및 워크플로우를 지정한다. 장비 타입 별 특성을 명세한 뒤 사용자는 장비 타입 별 소속된 장비의 ID를 지정한다. 이를 위해 사

1. <specification > ::= <cho_description>
| <specification> <cho_description>
| <obj-specification>
2. <cho- specification > ::= 'CHO' <CHO-ID> '{' <spec_desc> '}'
3. < spec-desc > ::= 'capacity' '=' <num_expression>
| 'workflow' '=' <identifier>
4. <obj- specification > ::= 'Object' '{' <obj-assign> '}'
5. <obj-assign> ::= <cho-type> ':' <obj-id>
6. <obj-id> ::= <identifier> | <obj-id> ',' <identifier>

<Figure 6> Syntax for Equipment Specification Language

용되는 문법은 4~6이다. 사용자는 Object 키워드를 사용하여 장비 타입 별 장비 지정 명세임을 구분할 수 있다.

<Figure 7>은 장비 특성을 명세한 예시를 보여준다. STS 장비 타입의 경우 최대 4개의 컨테이너를 처리할 수 있으며, 'wf_STIS'라는 워크플로우 명세를 따른다고 지정하였다 동일한 방법으로 SH 장비 타입을 지정하였으며, Object 명세에서 'STS'에 속한 장비는 'STS_101', 'STS_102', 'STS_103'임을 명시하였다. 'SH' 역시 동일한 방법으로 'SH' 타입에 속한 장비를 지정하였다.

```

CHO STS {
  capacity = 2 * 2
  workflow = wf_STIS
}
CHO SH {
  capacity = 2
  workflow = wf_SH
}
Object {
  STS : STS_101, STS_102, STS_103
  SH : SH_101, SH_102, SH_103
}
    
```

<Figure 7> An Example of Equipment Specification

장비 타입 별 워크플로우는 <Figure 8>의 문법을 따라 명세할 수 있다. 사용자는 Workflow 키워드 뒤의 <WF-ID>에 사용자가 원하는 워크플로우의 이름을 지정한다. 워크플로우 명세는 크게 세 부분으로 이루어져있다. 즉, 워크플로우 이벤트 명 지정, 워크플로우 전후 관계 지정, 워크플로우 이벤트 발생 시 수행할 기능 기술로 구분된다. 사용자는 이벤트 로그에 남겨진 이벤트 명과 워크플로우에 표현하고자 하는 이벤트 명이 다를 경우, <evt-assign>에서 워크플로우에 사용할 이벤트 명을 지정할 수 있다. 예를

들어 이벤트 로그상에는 ‘TwistLock’이라고 기록되어 있으며, ‘TwistLock’이라는 이벤트를 ‘TL:1’과 ‘TL:2’로 구분하여 사용하고 싶다면, 사용자는 <evt-assign>에서 이에 해당하는 명세를 기술할 수 있다. 예를 들어, <Figure 9>에 그 내용이 명시되어 있다.

```

1. <specification > ::= <wf_description> | <specification>
   <wf_description>
2. <wf_description> ::= 'Workflow' <WF_ID> '{' <evt_assign>
   < evt_flow> <evt_func> '}'
3. <evt_assign> ::= <assign> | <evt_assign> <assign>
4. <assign> ::= <EVT-TYPE> '=' <evt_id_list>
5. <evt_id_list> ::= <EVT-ID> | <evt_id_list> ',' <EVT-ID>
6. <evt_flow > ::= <flow> | <evt_flow> <flow>
7. <flow> ::= <EVT-ID> '>>' <EVT-ID>
8. <evt_func> ::= <func_desc> | <evt_func> <func_desc>
9. <func_desc> ::= <evt_id> '{' <func_list> '}' | <evt_type> '{'
   <func_list> '}'
10.<func_list> ::= <func> | <func_list> ',' <func>
11.<func> ::= <OBJ> '.in' '(' <PARAMETERS> ')' | <OBJ> '.out'
   '(' <PARAMETERS> ')'
    
```

<Figure 8> Syntax for Workflow Specification Language

```

Workflow wf_SH {
TwistLock = TL:1, TL:2
TwistUnlock = TU:1, TU:2

Start >> TL:1 TL:1 >> TL:2 TL:1 >> TU:2 TL:2 >> TU:1
TU:1 >> TL:2 TU:1 >> TU:2 TU:2 >> TL:1

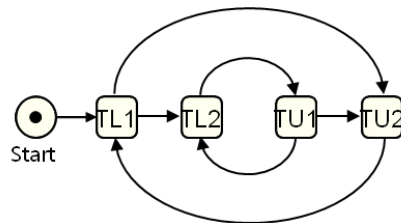
TwistLock { self.in(cont, event, time, loc), cont.owner.out(cont, event,
time, loc) }
TwistUnlock { self.out(cont, event, time, loc), cont.owner.in(cont,
event, time, loc) }
}
    
```

<Figure 9> An example of Workflow Specification

워크플로우 전후 관계는 <evt_flow>에 기술된다. 이벤트의 전후 관계는 '>>'를 사용하여 정의한다. 예를 들어, <Figure 9>의 ‘TL:1 >> TL:2’는 ‘TwistLock’이 발생한 뒤 또 다른 ‘TwistLock’이 발생할 수 있음을 나타낸다. 이는 동일한 장비가 컨테이너를 하나 집은 후 다른 지점에서 다른 컨테이너를 하

나 더 집었음을 의미한다. 이러한 표현방법은 이송장비의 싱글/듀얼 운영방법 또는 안벽 크레인의 트윈/탠덤 리프트를 표현할 수 있도록 해준다. <Figure 9>에 해당하는 워크플로우를 도식화하면 <Figure 10>과 같다.

사용자는 이벤트 발생시 수행할 기능을 <Figure 9>의 마지막 두 줄에서와 같이 명시할 수 있다. 이벤트 로그에서 ‘TwistLock’ 또는 ‘TwistUnlock’이 발생하면 ‘{’ 안에 기술된 CHO의 기능을 엔진이 수행하게 된다. 이는 각 이벤트가 발생할 때마다 다른 함수들을 호출하게 함으로써 다양한 작업처리를 기술할 수 있도록 지원한다.



<Figure 10> An Example of Workflow Specification

4.4 통계 명세

컨테이너 터미널의 레이아웃, 사용 장비, 운영방안 등이 다양하기 때문에 터미널 운영자마다 원하는 운영 성능지표 항목이나 통계 항목이 상이하다. 이러한 다양성을 지원하기 위해 명세 언어를 통해 사용자가 직접 통계 항목 및 수식을 지정할 수 있게 한다.

<Figure 11>은 본 연구에서 개발한 통계 명세 언어 문법의 일부를 나타낸다. 통계 명세 언어는 집합 이론에 기반하며, 크게 집합과 리스트(list)를 지원한다. 집합과 리스트는 각

각 '{}'와 '[]'로 표현되며, 각각을 원소 나열법(extensional definition)과 조건 제시법(intentional definition)을 사용하여 지정할 수 있다. 원소 나열법을 사용하는 경우 <Figure 12>의 첫째 줄과 같이 표현한다. 즉, 'Equip'은 'STS101', 'STS102', 'SH01', 'SH03'을 원소를 가지는 집합이다. 조건 제시법으로 기술하는 경우, {object ; domain : condition}의 형태를 사용한다. 사용자가 domain이 가지는 원소를 고려하여 condition을 만족하는 object들을 집합 또는 리스트로 만든다. 이때 ','와 '|'는 각각 'and'와 'or'을 의미한다.

```

1. <statement> ::= <extensional-def> | <intentional-def> | <assign>
2. <assign> ::= <IDENTIFIER> '=' <expr>
3. <extensional-def> ::= <identifier> '=' '{' <elements> '}' |
   <identifier> '=' '[' <elements> ']'
4. <elements> ::= <IDENTIFIER> <elements> ',' <IDENTIFIER>
5. <intentional-def> ::= < IDENTIFIER > '=' <intentional-stmt>
   | < IDENTIFIER > '=' '[' < intentional-stmt
   > ']'
   | < IDENTIFIER > '=' Sum '(' <
   intentional-stmt > ')'
6. <intentional-stmt> ::= '{' <object> ';' <domain> ':' <condition>
   '}'
   | '[' < object > ';' <domain> ':' <condition>
   ']'
7. <object> ::= < IDENTIFIER > | <expr> | '[' <object> ']'
   | < IDENTIFIER > ';' <object>
8. <domain> ::= <IDENTIFIER> 'in' <for-list> | <domain> ','
   <IDENTIFIER> 'in' <for-list>
9. <for-list> ::= '[' <for-list> '[' <for-list> '.' <for-list> | <functions>
   | <expr> | <for-list> <for-list>
    
```

<Figure 11> Syntax for Statistics Specification Language

집합 또는 리스트의 길이를 구할 때는 '|'| 기호를 사용한다. 예를 들어 '[SH01.event]'라 기술하면, 이는 'SH01'이 수행한 모든 이벤트의 개수를 구한다. <Figure 12>의 2번째 줄의 경우, 'SH01'이 수행한 모든 이벤트들 중에서 이벤트 타입이 'TwistUnlock'인 경우, 해당 이벤트가 처리한 컨테이너 이름을 리스트로 구한

뒤, 해당 리스트의 길이 값을 'SH01_th'에 저장한다. 이는 'SH01'이 처리한 컨테이너의 수를 의미한다. 추가적으로 sum(), dist(), time() 등의 함수를 제공한다. sum은 각 원소의 합을 구하며, dist와 time은 각각 두 지점 사이의 거리와 이동시간을 구한다.

```

Equip = {STS101, STS102, SH01, SH03}
SH01_th = |[c; e in SH01.event : c = e.cont, e.type = 'TwistUnlock' ]|
SH03_th = |[c; e in SH03.event : c = e.cont, e.type = 'TwistUnlock' ]|
SH_total = SH01_th + SH03_th
SH01_workTime = sum([t; i in 1..|SH01.event| : t = SH01.event[i].time
- SH01.event[i-1].time ])
SH03_workTime = sum([t; i in 1..|SH03.event| : t = SH03.event[i].time
- SH03.event[i-1].time ])
    
```

<Figure 12> An Example of Statistics Specification

5. 형식 프레임워크 검증

본 연구에서 제안한 형식 프레임워크의 검증 위해, 사용자 명세 파일을 해석하는 컴파일러 및 엔진을 구현하였다. 컴파일러는 명세 파일을 해석한 뒤 실행 파일을 생성하고, 생성된 실행 파일을 엔진이 구동하여 사용자가 입력한 통계 정보 및 보고서 파일을 생성한다. 본 연구에서는 Python 2.7을 사용하여 처리기를 개발하였으며, 각 명세 파일의 실행 파일을 Python 소스 코드로 변환하여 사용한다.

엔진은 이벤트 로그 모듈, 통계 모듈, 보고서 모듈로 구성되어 있다. 이벤트 로그 모듈은 작업이력 데이터와 관련된 실행 파일을 처리하기 위한 모듈로 레이아웃 실행 파일, 이벤트 로그 스키마 실행 파일, 워크플로우 실행 파일을 입력으로 받아 운영에 사용된 각 장비의 작업 시퀀스 및 터미널 전반에 걸

친 작업 흐름을 생성한다. 통계 모듈은 이벤트 로그 모듈로부터 생성된 작업 시퀀스 및 작업 흐름을 기반으로 사용자가 명세한 통계 항목에 따른 수식을 처리하고 사용자가 원하는 성능지표 통계 정보를 생성한다. 보고서 모듈은 보고서 실행 파일 및 통계 모듈에서 생성된 통계 정보를 바탕으로, 보고서 형식을

사용자가 원하는 형태로 생성한다. <Figure 13>은 본 연구에서 개발한 워크플로우 명세 언어를 사용하여 실행 파일을 생성한 결과를 보여준다. <Figure 14>은 운영 성능지표를 도출한 화면이다. 보고서는 상용 스프레드 시트 소프트웨어를 기반으로 개발되었다.

실제 터미널 운영 이력을 바탕으로 수행한

```

CHO STS{
  capacity = 2
  workflow = wf_STS
}
CHO ASC{
  capacity = 2
  workflow = wf_ASC
}
CHO SH{
  capacity = 2
  workflow = wf_SH
}

from CHO import CHO
import alge_wf as wf

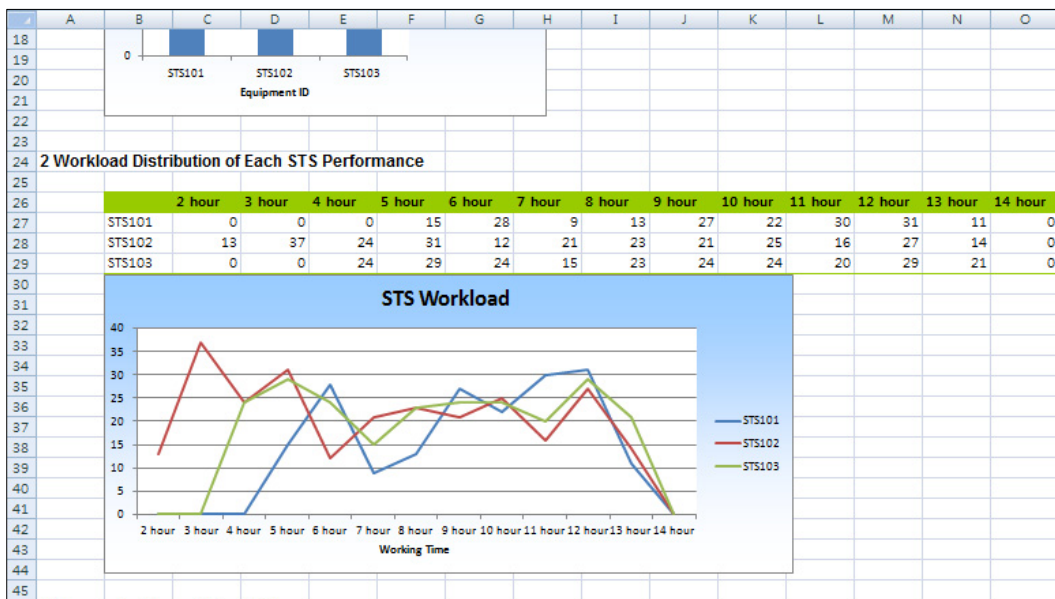
ASC011 = CHO('ASC011',2,wf.wf_ASC)
ASC031 = CHO('ASC031',2,wf.wf_ASC)
ASC061 = CHO('ASC061',2,wf.wf_ASC)
STS101 = CHO('STS101',2,wf.wf_STS)
SH19 = CHO('SH19',2,wf.wf_SH)
SH01 = CHO('SH01',2,wf.wf_SH)
ASC051 = CHO('ASC051',2,wf.wf_ASC)
SH03 = CHO('SH03',2,wf.wf_SH)
STS = set([STS101])

SH = set([SH01,SH03,SH19])

ASC = set([ASC011,ASC031,ASC051,ASC061])

OBJ = ['ASC011', 'ASC031', 'ASC061', 'STS101','SH19', 'SH01', 'ASC051', 'SH03']
    
```

<Figure 13> Translation Results of Specifications into Executable Programs



<Figure 14> Results of Terminal Performance Measures

구체적인 분석 사례는 다음과 같다. 특정 작업 시간대의 안벽 크레인 생산성이 하락하였으며, 도출된 터미널의 운영 성능지표를 통해 다음과 같은 사실을 파악할 수 있었다. 안벽 크레인의 생산성이 하락된 시간대의 야드 트랙터 작업을 분석한 결과, 평균 이동시간은 타 시간대와 비슷하였지만, 야드에서의 대기 시간이 타 시간대에 비해 높았으며, 외부트럭을 위한 컨테이너 처리량이 타 시간대에 비해 높다는 것을 알 수 있었다. 결과적으로 야드 크레인이 외부트럭을 서비스하기 위해 많은 시간을 할애하여 야드 트랙터를 서비스하는 시간이 부족하게 되었다. 이는 안벽 크레인을 서비스하는 야드 트랙터의 회전율을 낮추게 되며, 따라서 안벽 크레인의 생산성이 하락하였음을 알 수 있었다.

6. 결 론

본 연구에서는 다양한 터미널의 배치 및 운영방안, 운영장비를 고려한 범용 형식 프레임워크를 제안하였다. 사용자가 직접 터미널의 환경 및 운영방법 등을 기술할 수 있게 지원함으로써 형식 프레임워크의 용이성 및 확장성을 제고하였다. 본 연구에서 제안하는 형식 프레임워크는 터미널 레이아웃 명세, 이벤트 로그 스키마 명세, 워크플로우 명세, 통계 명세, 보고서 명세로 구성되어 있다. 각 명세를 기술하기 위하여 본 연구에서는 특화된 언어를 개발하였다. 제안된 프레임워크의 검증에 위해 실제 터미널 이력을 바탕으로 각 명세 파일을 정의하였고, 이를 실행한 결과 컨테이너 터미널의 운영 성능지표를 도출할

수 있었다.

컨테이너 터미널 운영자는 제안된 형식 프레임워크를 활용하여 다양한 환경의 터미널 운영에 대한 성능지표를 도출 할 수 있을 것으로 기대된다. 또한 도출된 운영 성능지표는 터미널 운영 시 발생하는 문제점을 분석하고 터미널 운영 개선 방안을 모색할 때, 유용한 정보를 제공할 수 있음을 확인하였다.

References

- [1] Ahn, K. R., Youn, K. Y., and Park, C., "BP modeling and data standardization for logistics cargo tracking process based on UN/CEFACT," *The Journal of Society for e-Business Studies*, Vol. 14, No. 4, pp. 299-313, 2009.
- [2] Beamon, B. M., "Performance, reliability, and performability of material handling systems," *International Journal of Production Research*, Vol. 36, No. 2, pp. 377-393, 1998.
- [3] Comuzzi, M., Vonk, J., and Grefen, P., "Measures and mechanisms for process monitoring in evolving business networks," *Data and Knowledge Engineering*, Vol. 71, pp. 1-28, 2012.
- [4] Kim, H. and Lee, Y.-H., "Workflow analysis for the process modeling of RFID systems," *The Journal of Society for e-Business Studies*, Vol. 15, No. 2, pp. 191-203, 2010.

- [5] Kim, K. H., "Decision-making problems for the operation of container terminals," *Journal of the Korean Institute of Industrial Engineers*, Vol. 33, No. 3, pp. 290-302, 2007.
- [6] Kim, S. and Choi, Y., "An AHP/DEA hybrid model for efficiency evaluation of container terminal," *The Journal of The Korea Port Economic Association*, Vol. 28, No. 2, pp. 179-194, 2012.
- [7] Lee, H., *Designing of a Decision Support System for Solving Real-time Operation Problems in Container Terminals*, Master's Thesis, Pusan National University, 2009.
- [8] Muthiah, K. M. N., Huang, S. H., and Mahadevan, S., "Automating factory performance diagnostics using overall throughput effectiveness (OTE) metric," *International Journal of Advanced Manufacturing Technology*, Vol. 36, pp. 811-824, 2008.
- [9] Zhang, H., and Kim, K. H., "Maximizing the number of dual-cycle operations of quay cranes in container terminals," *Computers and Industrial Engineering*, Vol. 56, pp. 979-992, 2009.

저 자 소 개



박은정
2005년
2008년
2010년~현재
관심분야

(E-mail : ejpark14@pusan.ac.kr)
동명정보대학교 컴퓨터공학과 (공학학사)
부산대학교 물류IT협동과정 (공학석사)
부산대학교 물류IT협동과정 박사과정 재학 중
항만물류, 시뮬레이션, 운영 분석 시스템



하병현
1998년
2005년
2006년~현재
관심분야

(E-mail : bhha@pusan.ac.kr)
서울대학교 산업공학과 (공학학사)
서울대학교 산업공학과 (공학박사)
부산대학교 산업공학과 조교수
비즈니스 프로세스 최적화, 물류 시스템 관리, 컨테이너
터미널 운영, 시뮬레이션