IE Interfaces
Vol. 20, No. 4, pp. 438-447, December 2007.

# Aspect-Oriented Approach를 이용한 통합 물류 시스템의 시뮬레이션 설계 및 분석 방법

김태호 · 엄인섭 · 이홍철†

고려대학교 정보경영공학과

# The Simulation Design and Analysis Method of Integrated Logistics System using an Aspect Oriented Approach

TaeHo Kim · InSup Um · HongChul Lee

Department of Industrial Systems and Information Engineering

This paper presents an aspect-oriented approach to simulation design and analysis in system design phase for integrated logistics system simulation. The integrated logistics system composed of AS/RS (Automated Storages and Retrieval System), AGVs (Automated Guided Vehicle System), STVs (Sorting Transfer Vehicle System) and Conveyor System is designed by using the aspect-oriented approach and UML (Unified Modeling Language). The multi-factorial design of experiments and regression analysis are used for design parameters of the system and Evolution Strategies is used to verify each parameter. Aspect-oriented approach for the integrated logistics system simulation shows the advantages of code reusability, extendible, modulation, easy improvement and a better design technique.

***Keyword:*** simulation, aspect-oriented approach, UML, evolution strategy

## 1. 서 론

최근에 물류시스템은 AS/RS(Automated Storages and Retrieval System), AGVs(Automated Guided Vehicle System, STVs(Sorting Transfer Vehicle System), Conveyor, Sorter등의 시스템과 같은 자동 물류 시스템들이 연계 설치되어 사용되고 있는데 이와 같은 복합 시스템은 시스템적 특성이 매우 복잡하여 수리적인 분석에 많은 어려움이 있다(Um InSup *et al.*, 2004). 그러므로 지금까지의 연구에서는 위의 시스템을 각각 분석하여 설계에 반영하거나, 프로토타입의 시스템을 간단한 시뮬레이션 기법으로 분석하는 방안이 많이 제시되어 왔다. 그러나 최근의 시뮬레이션 기법들은 전문화 된 시뮬레이션 툴을 3D 기반으로 하여 현재의 시스템과 동일한 시스템을 컴퓨터의 지원을 받아 설계 및 분석을 하는 방법이 우세한 경향이다. 이와 같은 방법은 기존 방법보다 시간을 줄이고, 복잡한 시뮬레이션 설계를 체계적이고 효율적으로 만들어 주는 장점도 있게 된다. 또한 최근에는 객체지향의 개념을 접목시켜서 시뮬레이션을 수직적인 구조로 표현하여 시뮬레이션의 모델링 및 설계를 효율적으로 하는 방안이 주로 사용되어왔다. 하지만 객체지향 시뮬레이션 방법은 시뮬레이션 설계에 있어서 수직(핵심)적인 요소만을 고려하고 세부적인 사항을 추가하는 방법의 시뮬레이션 설계를 하게 되어, 모델링의 효율성을 낮게 만드는 원인이 되고 있다. 따라서 본 논문에서는 객체지향의 개념과 관점지향(Aspect-Oriented) 개념을 접목 시킨 통합 물류시스템의 시뮬레이션 설계를 제시하고자 한다. 물류시스템의 설계를 Movement System과 Process System으로 구분하여 Movement System은 객체지향의

본 논문은 2007년도 두뇌한국 21사업에 의하여 지원되었음.
†연락저자 : 이홍철 교수, 136-713 서울시 성북구 안암동 5가 고려대학교 정보경영공학과, Fax : 02-3290-3767,
E-mail : hclee@korea.ac.kr
2006년 11월 접수, 1회 수정 후 2007년 03월 게재확정.

관점으로 설계하고, Process System은 관점 지향의 관점으로 설계를 함으로써 전체적인 시뮬레이션 설계의 소프트웨어적인 효율성과 체계적인 구조화를 도모하려고 하였다. 또한 통합물류시스템의 분석 과정은 AS / RS, AGVs, Conveyor, Sorter시스템의 중요 요소를 분석하여 AGV 이용률, ASRS 이용률, 총 처리량을 출력 변수로 선정하였다.

그리고 출력변수에 가장 큰 영향을 주는 변수를 입력변수로 선정하여 Multi-Factorial 분석을 실시하고 회귀분석을 통한 Multi-Objective Nonlinear Programming으로 구성함으로서 설계 최적화 값을 구하였다. 그리고 그 결과값을 진화 전략을 이용한 최적화 값과 비교함으로써 본 논문에서 제시한 객체지향과 관점지향 설계의 적합성 검증을 하였다. <Figure 1>은 이와 같은 통합 물류 시스템의 분석 절차를 나타내고 있다.

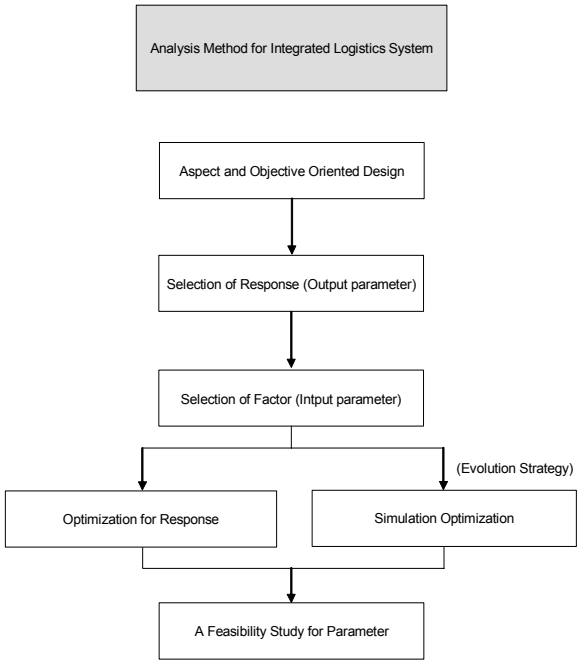



**Figure 1.** 통합 물류 시스템의 분석 절차

## 2. 기존 연구

관점지향 방법론을 설계에 효과적으로 적용하기 위해선 기본적으로 객체지향 방법론을 잘 이해하고 있어야 한다. 현재까지의 관점지향 방법론은 객체지향 방법론의 단점들을 보완해 주는 역할을 하고 있어 완전히 새로운 설계 과정이라고 하기에는 어려움이 있다.

제조 시스템 시뮬레이션 분야에 객체지향 방법론을 활용한 기존 연구를 살펴보면 John M. Usher가 기존의 제조 시스템 설계 방식과 객체지향 설계 방식을 비교하고 객체지향의 장점인 추상화, 캡슐화, 모듈화, 상속 등을 이용해 제조 시스템을 설계하는 방법을 제시 하였다(John M. Usher, 1996). Young *et al.*은 SDDS(Self Drill Drive Screw) 시스템을 모델로 잡아 객체 지향 툴을 이용해 제조 제어 시스템을 UML로 모델링 하는 방법을 제시하고 설계 하였으나 정형화된 UML 기법을 제공하지는 못했다(Young *et al.*, 2001). Manfredi Bruccoleri *et al.*는 UML을 이용하여 FMS의 분석과 설계를 하고 시뮬레이션 하였지만 Real-Time 제어 시스템은 고려하지 않았다(Manfredi Bruccoleri *et al.*, 2003).

관점지향 방법론에 대한 기존 연구를 살펴보면, Gregor Kiczales *et al.*가 관점지향 방법론과 객체지향 방법론을 비교하고, 관점지향 방법론의 필요성을 언급하면서 관점지향 방법론의 기본 개념과 횡단관심사(Cross-cutting Concern: 횡단관심사는 여러 개의 모듈에서 구현되는 시스템 포괄적인 요구사항이다. 기업 업무를 예를 들면 사용자 인증, 보안, 자원 공유 등이 횡단관심사의 예가 될 수 있다)의 의미를 설명하였다(Gregor Kiczales *et al.*, 1997). Gregor Kiczales *et al.*는 또한 관점 지향 방법론을 구현할 수 있는 언어인 AspectJ(AspectJ는 관점지향 방법론을 구현하기 위해 새로 제안된 Java기반 언어이다)에 대해서 설명 하였다(Gregor Kiczales *et al.*, 2001). Erick Putrycz *et al.* and Bart Dc Win *et al.*는 관점지향 방법론을 Load balancing 서비스와 실제 보안 어플리케이션 소프트웨어에 적용해 봄으로써, 관점 지향 방법론이 실제 어플리케이션에서 어떻게 적용되어 질 수 있는지 보여 주었다(Erick Putrycz *et al.*, 2002; Bart Dc Win *et al.*, 2005). Roger Alexander는 관점지향 방법론을 실제 프로젝트에 적절하게 도입하기 위해서는 어떤 준비가 필요하고, 어떤 위험요소가 있는지에 대해 설명하였다(Roger Alexander, 2003). 이렇듯 관점지향 방법론에 대한 다양한 연구가 있었지만 관점지향 방법론을 물류 시스템에 적용한 연구는 아직까지 미흡하다.

## 3. 관점지향 방법론을 이용한 통합 물류 시스템의 시뮬레이션 설계

### 3.1 관점지향 방법론

기계언어에서 절차 프로그래밍과 객체지향 프로그래밍에 이르기까지 소프트웨어 공학은 여러 과정을 거치면서 발전해 왔다. 객체지향 방법론은 현재 대부분의 소프트웨어 개발 프로젝트에서 사용하고 있는 방법이다. 객체지향 방법론의 강점은 공통된 동작을 모델링 하는 능력이다. 그러나 많은 소프트웨어 개발자들이 경험한 것처럼 객체지향 방법론은 많은 모듈들에 걸쳐있는 횡단관심사를 모듈화 하지 못하는 단점이 있다. 따라서 관점지향 프로그래밍은 횡단관심사를 포괄적이고 체계적으로 모듈화 함으로써 횡단 관심사의 구현 방법을 새롭게 변경한다(Ramnivas Laddad, 2003). 시뮬레이션 설계에 있어서도 본 논문에서 제시한 방법은 객체지향 설계와 관점지향 설계를 병행한 설계가 된다. <Figure 2>에 보이는 방법처럼 전체 Movement System은 객체지향 설계를 하고 Process System은 관점지향 설계를 하여 두 가지 방법을 통합한 설계 방법이 된다.

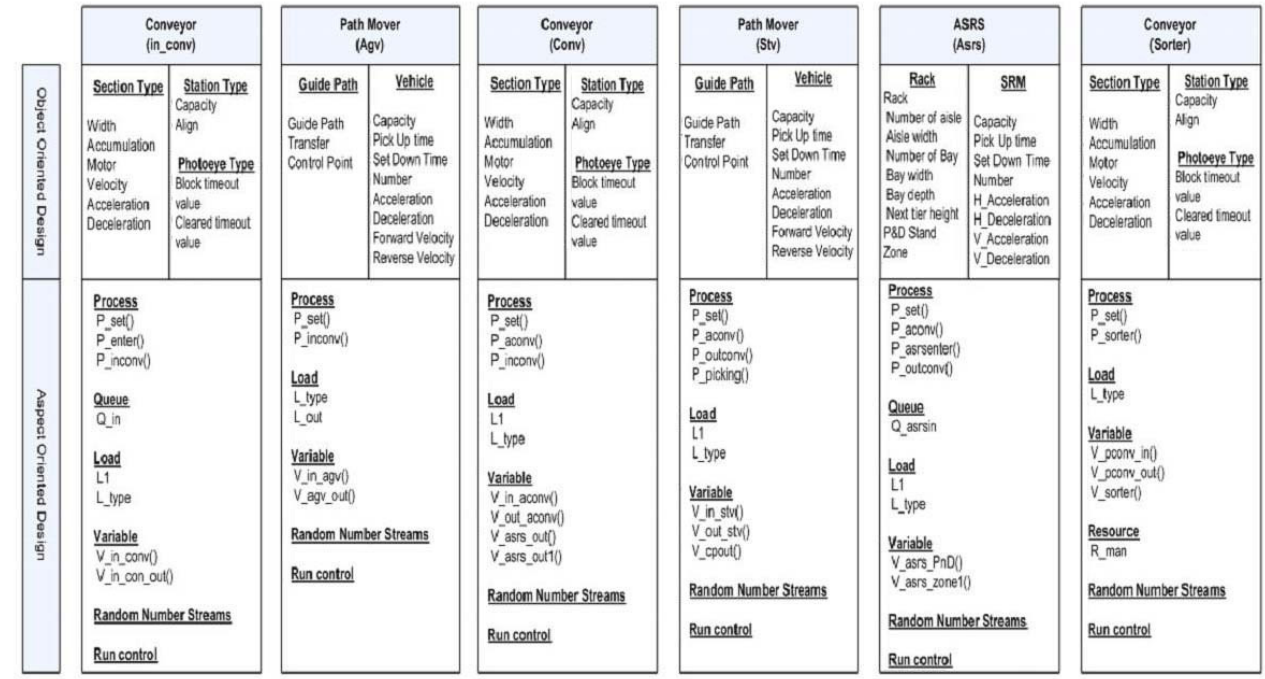



Figure 2. 객체지향과 관점지향 설계를 이용한 통합 설계 방법

Movement System은 컨베이어, AGVs, AS/RS, Sorter등으로 구분하여 설계를 하였고, Process System은 Process, Queue, Load, Variable, Random Number Stream, Run Control을 횡단 관심사의 관점에서 설계를 함으로써 전체적인 시스템 설계를 나타내고 있다.

### 3.2 통합 물류 시스템의 UML 설계

시뮬레이션 설계가 관점지향 방법론을 효과적으로 적용하기 위해서는 다음의 내용들을 고려해야 한다. 첫째 해당 시스템의 구조를 객체지향 관점에서 잘 정의할 필요가 있다. 앞서

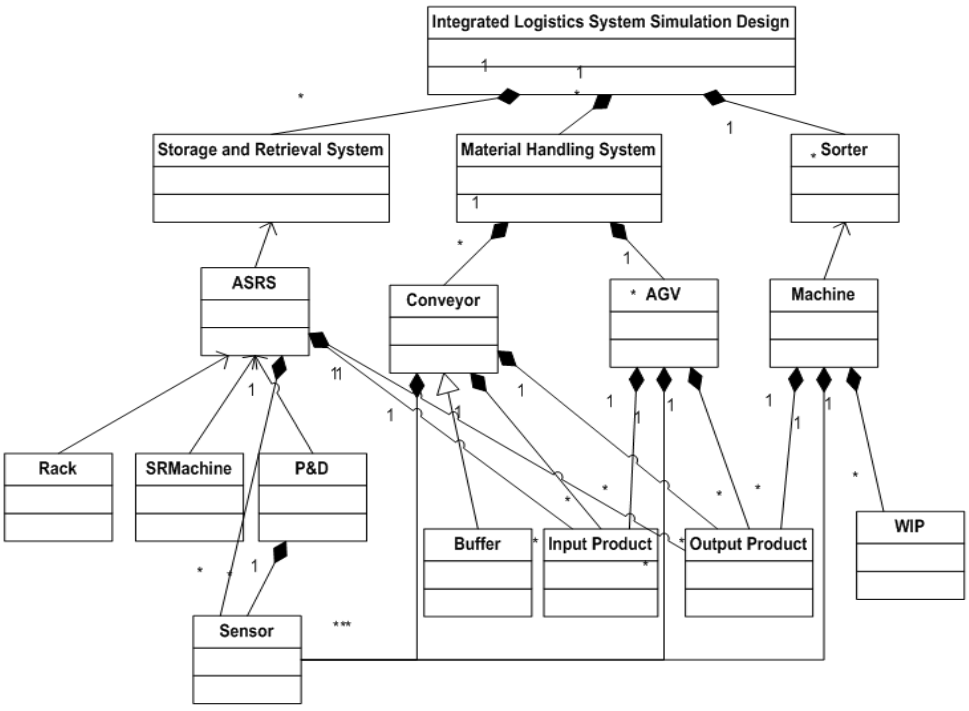



Figure 3. 전체 시스템 구조도(Main Class Diagram)

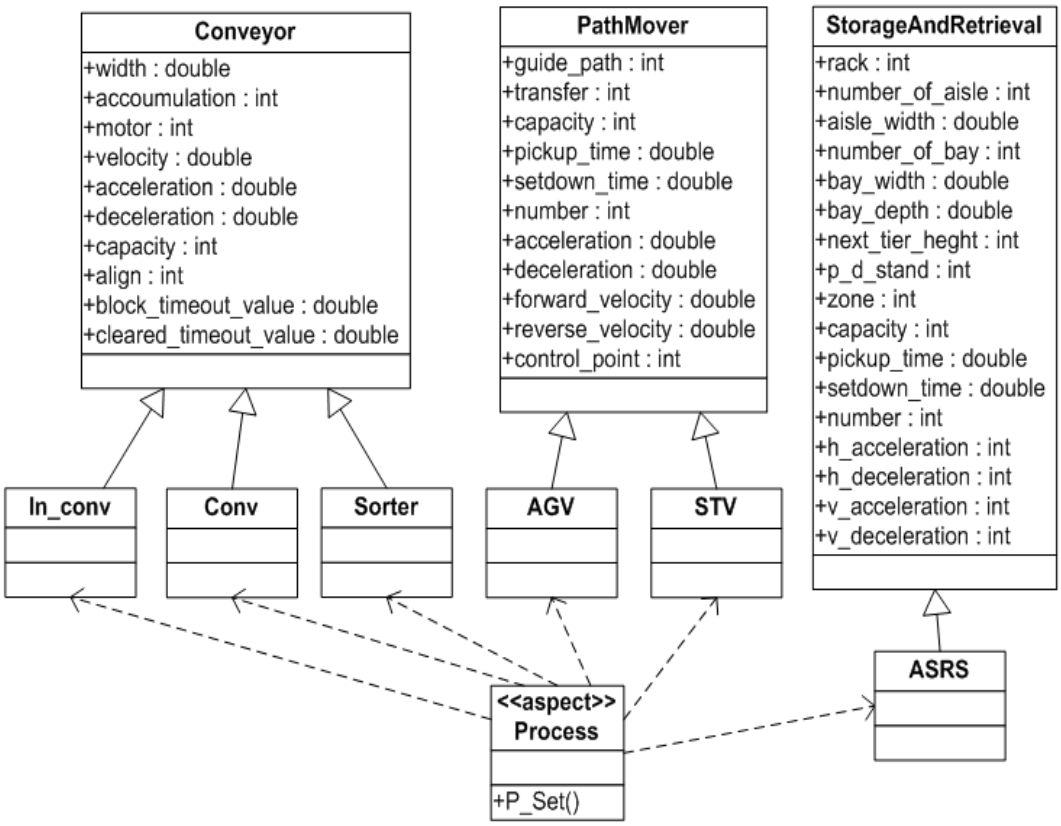
언급했듯이 객체지향 관점에서 잘 정의된 구조일수록 관점지향 방법론을 적용했을 때 그 효과가 커지기 때문이다. 객체지향 관점에서 분석한 대상 시스템의 전체 클래스 구조는 <Figuer 3>과 같다. 전체 통합 물류 시스템은 다수의 Storage and Retrieval System, Material Handling System, Sorter로 구성되어 있고, 각각의 시스템은 하나 또는 다수의 세부 개체들로 이루어져 있다. 둘째, 개발자는 설계에서뿐만 아니라 요구분석, 구현 등 시스템 개발의 모든 단계에서 관점지향 방법론의 개념을 숙지하면서 설계를 진행하여야 한다. 관점지향 방법론의 장점인 횡단관심사를 효과적으로 추출하기 위해서는 모든 개발 과정에서 관점지향 방법론으로 분석하고 생각하여야 한다. 셋째, 관점지향 프로그래밍이 어떻게 동작하고, 관점지향 프로그래밍을 도입하면 우리 설계에 어떤 변화가 생기는지 정확하게 이해하고 있어야 한다. 충분한 사전 테스트와 관점지향 프로그래밍에 대한 정확한 이해가 설계에 관점지향 방법론을 효과적으로 적용시킬 수 있는 발판이 된다(Ramnivas Laddad, 2003; Roger Alexander, 2003).

본 논문에서 제시한 관점지향 구조의 세부 시스템은 <Figure 4>에 제시되어 있다. <Figure 4>는 Aspect 모듈로 분리해낸 Process의 P_Set()이 세부 시스템에서 어떻게 연관이 되어 있는지 보여준다. 세부시스템은 객체지향 방법론에 의해 설계하여 In_conv, Conv, Sorter 클래스는 Conveyor를 상속하고 있고 AGV와 STV 클래스는 PathMover 클래스를 상속하고 있는 구조이다. <Figure 2>와 <Figure 4>에서 확인할 수 있듯이 관점지향 구조로 살펴보면 P_Set()은 거의 모든 클래스에 연관이 되어 있는 것을 확인할 수 있다. Aspect 모듈과 일반 클래스 사이에 연관관계가 있다는 것을 나타내기 위해서 점선 화살표를 사용하였으나 이는 정식 UML 표기법과 다를 수 있다. 이와 같이 모든 클래스에 연관이 있는 Process의 P_set()을 Aspect 모듈로 분리해서 설계하는 것이 관점지향 방법론의 횡단 관심사 추출의 핵심 이라고 할 수 있다. 이렇게 추출된 Aspect 모듈은 완전히 독립적이어서 기존 시스템의 구조를 변경하거나 수정하지 않고 적용이 가능하다. 또한 이런 Aspect 모듈은 재사용이 가능하기 때문에 추가의 노력 없이 다른 설계에서 사용이 가능하다.

이렇듯 관점지향 방법론을 이용하여 시뮬레이션을 설계·구현하면 고도의 모듈화, 시스템 개선의 용이, 설계 결정 시점의 연기, 코드 재사용성의 확대 등의 장점을 가지게 된다(Ramnivas Laddad, 2003).

### 3.3 통합 물류 시스템 모델

본 논문에서 사용된 시스템 모델은 위의 관점지향과 객체지향 설계를 기반으로 자동창고, AGVs, STVs, Conveyor, Sorter의 시스템이 통합된 물류 시스템 모델로서 <Figure 5>와 같이 설계되었다. 자동창고는 10개의 랙을 가지고 있고 5개의 통로(Aisle)와 P&D Station(Pickup and Dropdown Station)을 가지고 있




**Figuer 4.** 세부 시스템 클래스 구조도(Detailed System Class Diagram)

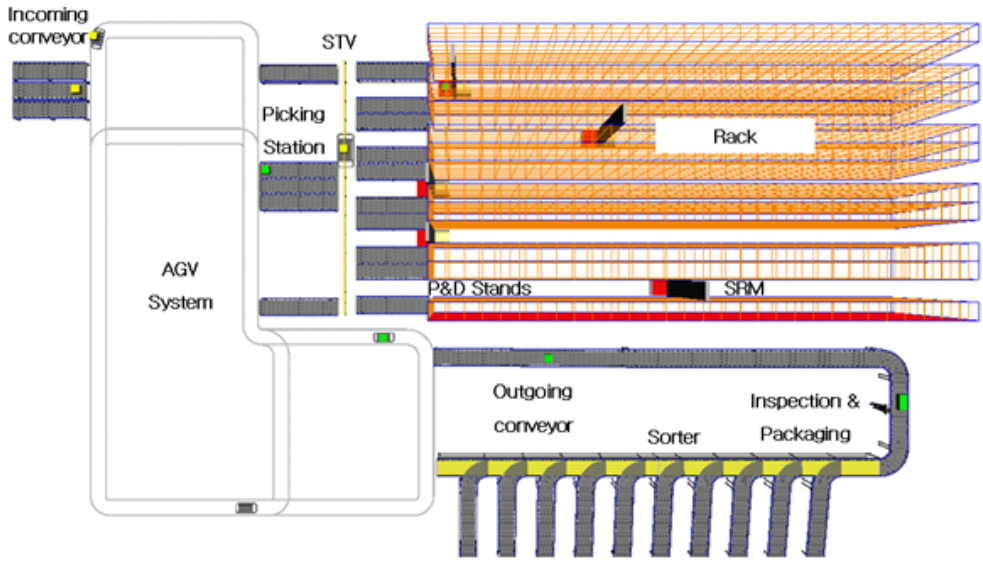



**Figure 5.** 통합 물류 시스템 시뮬레이션 모형

는 전형적인 형태이다. AGV 시스템은 Incoming Conveyor에서 자동창고 입고 Conveyor까지의 이동과 자동창고에서의 Picking Conveyor에서 Outgoing Conveyor까지의 이동 역할을 하게 된다. 컨베이어 시스템은 Incoming, Asrs Picking Conveyor, Outgoing 컨베이어로 구분하여 이동과 버퍼의 역할을 병행하게 된다. Sorter는 10개의 Station으로 구분을 해주는 전형적인 구분기의 역할을 하게 된다.

## 4. 실험 계획

### 4.1 입 · 출력 변수의 선택

설계에 있어서의 입력변수와 출력변수의 선정은 중요한 의미를 가지게 된다. 즉, 시스템의 목적을 반영하도록 출력 변수를 선정하여야 하고, 입력변수는 설계 변수 중 가장 출력변수에 영향을 주는 변수들을 선정하는 것이 중요하다. 본 논문에서 사용 된 출력 변수는 <Table 1>과 같다.

**Table 1.** 출력변수의 정의

| Response(출력변수) | | |
|---|---|---|
| Symbol | Contents | Unit |
| Y1 | AGV_Utilization(이용률) | % |
| Y2 | ASRS_Utilization(이용률) | % |
| Y3 | Throughput(총처리량) | EA |

AGV와 ASRS의 이용률은 기계의 효율성을 나타내는 지표로서 이용률을 잘 활용하면 적정 가동시간과 계획 정비 시스템에 적용하여 운영의 효율성을 높일 수 있게 된다. 이용률은 최대화 시키는 것이 가장 효율적인 방법은 아니고 일반적으로 80%가 적정 설계 수준으로 이용되고 있다. 또한 총 처리량은 시스템의 성과를 측정하는데 가장 중요하고 기본이 되는 출력변수로서 시스템 능력의 예측과 평가에 활용된다. 따라서 본

**Table 2.** 입력변수의 정의와 실험 수준

| Factors | | Level | | | |
|---|---|---|---|---|---|
| Symbol | Contents | Low | MID | HIGH | Unit |
| $x_1$ | AGV_Number(운행대수) | 3 | 5 | 7 | AGV |
| $x_2$ | AGV_Vel(주행속도) | 40 | 60 | 80 | m/min |
| $x_3$ | Asrs_H_Vel(주행속도) | 40 | 60 | 80 | m/min |
| $x_4$ | Asrs_V_Vel(승강속도) | 60 | 80 | 100 | m/min |
| $x_5$ | Sorter_Vel(구분기속도) | 80 | 100 | 120 | m/min |
| $x_6$ | AGV_Acc(가속도) | 0.5 | | | $m/sec^2$ |
| $x_7$ | Asrs_H_Acc(가속도) | 0.5 | | | $m/sec^2$ |
| $x_8$ | Asrs_V_Acc(가속도) | 0.5 | | | $m/sec^2$ |

문은 위의 3가지 출력변수를 기반으로 입력변수를 아래 <Table 2>와 같이 선정하였다. 각 기계의 대수와 속도, 가속도는 기본적인 설계 변수로써 위에서 언급한 출력변수에 가장 큰 영향을 주는 변수라고 할 수 있다. <Table 2>에 보이는 것과 같이 입력변수의 실험수준이 $x_1 \sim x_5$까지는 Low, Mid, High의 세 수준이고 $x_6 \sim x_8$은 동일한 값을 갖는 Constant Factor가 된다. 따라서 본 논문의 실험 회수는 243회 반복 회수는 3회가 된다. 출력변수 $y_i$에 대하여 회귀 식의 1차항(Linear: $x_i$), 교호작용(Interaction: $x_i x_j$ ), 2차항(Square: $x_i^2$)에 대한 일반적인 회귀식은 다음과 같다.

$$y_i = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{j=i+1}^{k} \sum_{i=1}^{k} \beta_{ij} x_i x_j + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \epsilon$$
$$\text{for } \epsilon \sim N(0, \sigma^2)$$

본 논문에서는 분산분석(ANOVA)을 이용하여 위 5개의 입력 변수($x_1 \sim x_5$)들의 1차 항 변수, 교호작용 변수, 2차 항 변수가 출력변수에 대한 중요도 분석을 실시하였다. 분산분석의 결과는 <Table 3>과 같다. <Table 3>을 보면 P_Value가 유의수준($\alpha = 0.5$)보다 작은 값을 유의한 값으로 인정하여 굵게 표시하였고 큰 값은 유의하지 않은 것으로 판단하였다. 분산분석의 결과 AGV 이용률과 총 처리량은 AGV 대수, 속도가 가장 큰 영향이 된다고 할 수 있고, ASRS 이용률은 SRM의 주행속도가 가장 큰 영향이 된다고 분석 할 수 있다. 또한 위의 변수 중 유의한 변수를 통하여 회귀모형을 선정하였다.

(1) AGV 이용률:

$$y_1 = 0.188 + 0.303x_1 - 0.004x_2 - 0.025x_1^2 - 0.001x_1x_2 - 1.02E-05x_1x_3 - 7.01E-06x_1x_4 + 2.013E-17x_1x_5 + 6.733E-05x_2^2 + 3.350E-07x_2x_3 + 4.057-07x_2x_4 - 2.23E-18x_2x_5$$

(2) ASRS 이용률:

$$y_2 = 0.970 - 0.011x_3 + 1.116E-05x_1x_3 + 8.745E-07x_2x_3 + 6.055E-05x_3^2 - 9.68E-07x_3x_4 - 1.33E-19x_3x_5$$

(3) 총 처리량:

$$y_3 = -883.539 + 310.426x_1 + 23.69x_2 - 19.641x_1^2 - 1.045x_1x_2 - 0.004x_1x_3 + 0.006x_1x_4 - 0.132x_2^2 - 9.86E-06x_2x_5$$

**Table 3.** 분산분석 결과

| Source | AGV 이용률 | | ASRS 이용률 | | 총 처리량 | |
|---|---|---|---|---|---|---|
| | *F* | *P* | *F* | *P* | *F* | *P* |
| $x_1$ | **62.783** | **0.000** | 0.046 | 0.831 | **347.734** | **0.000** |
| $x_2$ | **20.482** | **0.000** | 0.037 | 0.848 | **44.175** | **0.000** |
| $x_3$ | 0.011 | 0.971 | **4,991.713** | **0.000** | 0.004 | 0.949 |
| $x_4$ | 0.001 | 0.972 | 0.031 | 0.861 | 0.004 | 0.951 |
| $x_5$ | 0.000 | 1.000 | 0.000 | 1.000 | 0.000 | 0.996 |
| $x_1^2$ | **89.962** | **0.000** | 0.045 | 0.832 | **266.474** | **0.000** |
| $x_1x_2$ | **134.489** | **0.000** | 0.117 | 0.733 | **356.469** | **0.000** |
| $x_1x_3$ | **32.898** | **0.000** | 133.878 | 0.000 | **118.949** | **0.000** |
| $x_1x_4$ | **41.033** | **0.000** | 0.019 | 0.890 | **170.884** | **0.000** |
| $x_1x_5$ | **46.549** | **0.000** | 0.036 | 0.850 | **207.320** | **0.000** |
| $x_2^2$ | **17.925** | **0.000** | 0.032 | 0.857 | **39.227** | **0.000** |
| $x_2x_3$ | **9.893** | **0.002** | **193.921** | **0.000** | **18.963** | **0.000** |
| $x_2x_4$ | **12.502** | **0.000** | 0.004 | 0.948 | 26.107 | 0.000 |
| $x_2x_5$ | **14.429** | **0.000** | 0.026 | 0.871 | 30.272 | 0.000 |
| $x_3^2$ | 0.010 | 0.919 | **2,459.298** | **0.000** | 0.005 | 0.945 |
| $x_3x_4$ | 0.012 | 0.911 | **376.902** | **0.000** | 0.000 | 0.990 |
| $x_3x_5$ | 0.008 | 0.930 | **531.288** | **0.000** | 0.003 | 0.954 |
| $x_4^2$ | 0.001 | 0.971 | 0.030 | 0.863 | 0.004 | 0.950 |
| $x_4x_5$ | 0.001 | 0.978 | 0.018 | 0.892 | 0.002 | 0.965 |
| $x_5^2$ | 0.000 | 1.000 | 0.000 | 1.000 | 0.000 | 0.996 |

위의 회귀 식은 수정결정계수($adj\ R^2$)의 값이 각각 (0.831, 0.989, 0.976)으로 모형을 설명하기에 충분히 타당한 것을 알 수 있었다. 또한 본 논문에서는 AGV, ASRS, Conveyor, Sorter의 정량적 변수만을 기준으로 회귀 식을 산출하였기 때문에 수정결정계수가 높게 나온 것을 알 수 있었다. AGV의 운영방식, Guide Path, Dispatching Rule, ASRS의 존 할당 방식, 저장 및 출고 방식 등의 요소를 반영하게 되면 수정결정계수는 조금 낮아질 것을 예측 할 수 있다. 따라서 본 논문에서는 이 회귀 식을 기초로 아래와 같은 제약식 만을 추가하여 Multi-Objective Nonlinear Programming의 식을 도출 하였다.

<Constraints>

$$3 \le x_1 \le 7,\ 40 \le x_2 \le 80,\ 40 \le x_3 \le 80,$$
$$60 \le x_4 \le 100,\ 80 \le x_5 \le 120$$
$$where\ x_1\ is\ integer\ x_2, x_3, x_4, x_5\ are\ real\ value.$$

본 논문에서의 설계변수의 타당성 검증을 위하여 위의 Multi-Objective Nonlinear Programming의 최적 해를 구하는 것 보다 각각의 목적함수에 따른 최적 해를 구하여 시뮬레이션 최적화와 비교하는 방법을 사용하였다. 따라서 위의 목적함수를 만족시키는 각각의 입력변수와 출력변수는 <Table 4>와 같다.

### 4.2 설계 변수를 확인하기 위한 시뮬레이션 최적화

4.1절에서의 설계변수가 본 시스템이 얼마만큼의 영향이 있는지와 결과값의 유의성 검증을 위하여 본 논문에서는 최적화 실험을 통한 검증을 수행하였다. 즉, 객체지향과 관점지향의 설계의 타당성 검증 진화전략을 이용한 최적화 실험으로 수행하였다.

#### 4.2.1 진화 전략 프로세스

시뮬레이션 최적화를 위한 진화 전략 프로세스는 아래와 같은 절차로 수행된다.

***진화전략의 수행 절차***

Step1: Generate first run
Setp2: Randomly create the first generation of children
Step3: Make the runs for each child
Step4: Select the parents
Step5: Randomly pick two of the parents
Step6: Combine them
Step7: Mutate the factor value
Step8: Repeat step 3～7 until the termination criteria are met

첫 번째 런을 생성하여(초기화), 최초로 자식세대를 생성한 후 각 세대에 대한 런을 수행하고, 이 중에서 각 객체의 적합도(fitness)에 근거하여 다음 세대에 이용 될 부모를 선택하게 된다. 선택된 부모로부터 돌연변이를 생성하여 새로운 자식 개체들을 생성하면서 조건을 만족 시키거나 수행 회수를 만족시키게 되면 종료하고 그렇지 않으면 step3부터 step7까지의 과정을 반복을 하게 된다(Um InSup *et al.*, 2004).

#### 4.2.2 시뮬레이션 최적화 수행 및 결과

시뮬레이션 프로그램으로는 AutoMod V. 11을 사용하였고, Pentium IV 3.0GHz의 노트북을 이용하였다. 또한 적용된 실행조건은 다음 <Table 5>와 같다.

**Table 4.** 각 목적함수에 따른 최적화 결과

**AGV 이용률 최적화**

| AGV | | ASRS | | Sorter | AGV 이용률 | ASRS 이용률 | 총 처리량 |
|---|---|---|---|---|---|---|---|
| 운행 대수 | 주행 속도 | 주행 속도 | 승강 속도 | 구분기 속도 | | | |
| 5 | 60 | 40 | 100 | 80 | **0.7713** | 0.6323 | 811.67 |

**ASRS 이용률 최적화**

| AGV | | ASRS | | Sorter 구분기 속도 | AGV 이용률 | ASRS 이용률 | 총 처리량 |
|---|---|---|---|---|---|---|---|
| 운행 대수 | 주행 속도 | 주행 속도 | 승강 속도 | | | | |
| 5 | 40 | 40 | 80 | 80 | 0.7477 | **0.6427** | 704.33 |

**총 처리량 최적화**

| AGV | | ASRS | | Sorter 구분기 속도 | AGV 이용률 | ASRS 이용률 | 총 처리량 |
|---|---|---|---|---|---|---|---|
| 운행 대수 | 주행 속도 | 주행 속도 | 승강 속도 | | | | |
| 7 | 60 | 40 | 80 | 80 | 0.5517 | 0.624 | **812.33** |

Table 5. 시뮬레이션 최적화 실행 조건

| 시뮬레이션 최적화 실행 조건 | |
|---|---|
| 적합도 함수<br>(Fitness Function) | MAX AGV 이용률<br>MAX ASRS 이용률<br>MAX 총 처리량 |
| 각 세대당 부모의 개체수(Number of Parents for each generation) | 3 |
| 반복 실험 회수(Replication) | 5회 |
| 예비 시간(Warm-up Time) | 1시간 |
| 실험 시간(Snap Length) | 9시간 |
| 종료 조건<br>(Terminating Condition) | 1. 50세대 동안 5% 미만으로 향상 되었을 때<br>2. 각 세대의 최대 실험 회수가 100이 되었을 때 |

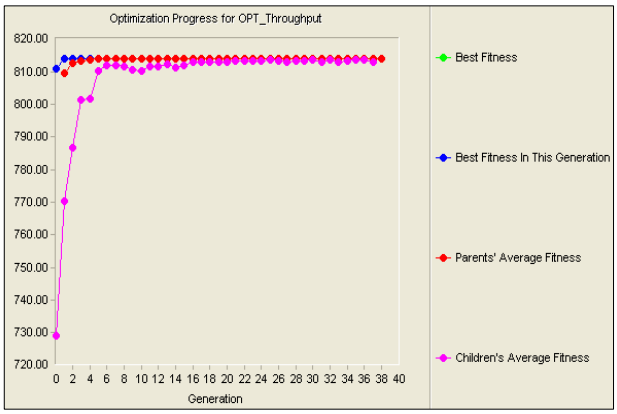



Figure 6. 시뮬레이션 최적화 모형

위의 실험 조건에 의해 수행된 시뮬레이션 수행 결과를 <Table 6>에 나타내었다. <Table 6>를 분석하여 보면 각각의 최적화 실험은 출력변수에서 그렇게 큰 차이를 보이지는 않았다. 하지만 입력 변수의 선정에 있어서 운행 대수와 속도는 통합 물류 시스템을 운영하는데 있어서 중요 변수로서 최적의 입력 변수의 결합이 요구된다. 또한 AGV와 ASRS의 이용률은 0.80이 최적의 값으로 선정하여 최적화를 수행하였다. <Figure 6>은 총 처리량에 따른 최적화 모형을 나타낸다.

Table 7. 관점지향 설계 최적 값과 시뮬레이션 최적 값의 비교

| | AGV 이용률 | ASRS 이용률 | 총 처리량 |
|---|---|---|---|
| 관점지향 설계 최적값 | 0.7713 | 0.6427 | 812.33 |
| 시뮬레이션 최적값 | 0.7732 | 0.6412 | 813.6 |

4.1절과 4.2절의 결과값을 비교하여보면 <Table 7>과 같이 나타낼 수 있다. 두 가지 최적화 방법에 의한 결과 값을 비교해 보면 종속변수(총처리량)의 경우 (812.33, 813.6)로 매우 작은 차이를 보이며 설계변수($x_1$, $x_2$, $x_3$, $x_4$, $x_5$)의 값들도 매우 유의한 값들을 나타냄을 알 수 있다. 두 가지 값들에 대한 통계적 분석(T_Test)결과는 <Table 8>과 같다.

Table 8. 총처리량 최적화 결과에 대한 통계분석 결과

| | 관점지향설계 | 시뮬레이션설계 |
|---|---|---|
| Mean | 179.9 | 183.4 |
| St.Dev | 311 | 310.4 |
| SE Mean | 126.9 | 126.7 |
| Estimated for Difference<br>Confidence Interval for Difference | | -3.545<br>-12.022<br>4.932 |
| T-Test of Difference<br>T_Value = -1.075, P_Value = 0.331, DF = 5 | | |

N = 6, Confidence Level = 95%

Table 6. 시뮬레이션 최적화 수행 결과

| AGV 이용률 최적화 | | | | | | | |
|---|---|---|---|---|---|---|---|
| AGV | | ASRS | | Sorter | AGV 이용률 | ASRS 이용률 | 총 처리량 |
| 운행 대수 | 주행 속도 | 주행 속도 | 승강 속도 | 구분기 속도 | | | |
| 5 | 60 | 40 | 100 | 120 | **0.7732** | 0.6332 | 813.67 |
| ASRS 이용률 최적화 | | | | | | | |
| AGV | | ASRS | | Sorter | AGV 이용률 | ASRS 이용률 | 총 처리량 |
| 운행 대수 | 주행 속도 | 주행 속도 | 승강 속도 | 구분기 속도 | | | |
| 5 | 40 | 40 | 80 | 100 | 0.9552 | **0.6412** | 710.2 |
| 총 처리량 최적화 | | | | | | | |
| AGV | | ASRS | | Sorter | AGV 이용률 | ASRS 이용률 | 총 처리량 |
| 운행 대수 | 주행 속도 | 주행 속도 | 승강 속도 | 구분기 속도 | | | |
| 7 | 60 | 40 | 80 | 100 | 0.5524 | 0.6248 | **813.6** |

<Table 8>의 결과에 의하면 두 가지 샘플의 추정된 오차(Difference) = -3.545로 신뢰구간인 -12.022과 4.932사이에 있으므로 두 샘플의 평균의 차이는 없다고 볼 수 있으며 또한 T_Value = -1.075, DF = 5 일때의 P_Value값도 0.331로서 유의수준($\alpha = 0.05$)보다 매우 큰 값을 가지므로 두 가지 방법에 의한 값의 차이는 유의하지 않다($H_1 : \mu_1 - \mu_2 \neq \delta_0$)는 것을 보여준다. 이 결과로부터 관점지향 설계의 설계 변수 선정이 시뮬레이션 모델에 대한 근사함수로서의 타당성을 충분히 갖는다는 것을 확인 할 수 있었다.

## 5. 결 론

최근의 통합 물류 시스템 같이 복잡하고 그 규모가 큰 시스템의 경우 시뮬레이션 설계와 분석에 많은 어려움이 있다. 이에 본 논문에서는 객체지향 개념과 관점지향의 개념을 활용하여 시뮬레이션 설계의 최적화와 효율적인 분석 방법을 제시하였다.

본 논문은 객체지향 개념을 활용하여 개별 프로세스의 공통된 동작들을 모듈화하고, 관점지향 개념을 활용해서 전 프로세스에 걸쳐 있는 횡단 관심사를 추출하여 설계하였다. 이렇게 설계함으로써, 객체지향 개념의 장점인 추상화, 캡슐화, 상속 등을 이용하여 해당 시스템의 가시성과 모듈화를 높일 수 있었고, 또한 객체지향 설계의 단점인 횡단 관심사 문제는 최근 새롭게 연구되고 있는 관점지향 개념을 활용해 설계함으로써 관점지향의 개념의 장점인 각 모듈에 대한 명확한 책임 소재, 코드 재사용성의 확대, 효율적인 기능 구현 등 전체 통합 물류 시스템의 시뮬레이션 모델을 효율적으로 설계하는 방법을 제시하였다. 또한 객체지향과 관점지향 설계의 타당성 검증을 위하여 본 시스템의 Factor(입력변수)와 Response(출력변수)의 통합 분석을 통하여 최적 값을 도출하였고, 도출 된 값의 타당성 검증을 위하여 진화전략을 이용한 시뮬레이션 최적화 기법을 적용하였다. 그 결과 본 논문에서 제시한 시뮬레이션 설계가 객관적인 타당성을 가짐을 제시할 수 있었다. 연구가 많이 되어 있지 않은 물류 시스템의 관점지향 설계에 대해 본 논문에서 제시된 시뮬레이션 설계 방법이 하나의 대안으로 제시될 수 있다.

## 참고문헌


Bart De Win, Wouter Joosen and Frank Piessens (2005), Developing Secure Applications through Aspect-Oriented Programming, *Aspect-Oriented Software Development*, Addison-Wesley.

Erik Putrycz and Guy Bernard (2002), Using Aspect-Oriented Programming to build a portable load balancing service, *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, 473-478.

John M. Usher (1996), A tutorial and review of object-oriented design of manufacturing software systems, *Computers & Industrial Engineering*, **30**(4), 781-798.

Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier and John Irwin (1997), Aspect-Oriented Programming., *Lecture notes in computer science*, **1241**, 220-242.

Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm and William G. Griswold (2001), An Overview of AspectJ., *Lecture notes in computer science*, 2072, 327-353.

K. W. Young, R. Pigging and P. Rachitrangsan (2001), An Object-Oriented Approach to an Agile Manufacturing Control System Design, *International journal of advanced manufacturing technology*, **17**(11), 850-859.

Manfred Bruccoleri, Sergio Noto La Diega and Giovanni Perrone (2003), An Object-Oriented Approach for Flexible Manufacturing Control System Analysis and Design Using the Unified Modeling Language, *The International Journal of Flexible Manufacturing System*, **15**(3), 195-216.

Ramnivas Laddad (2003), *Aspectj in Action: Practical Aspect-Oriented Programming*, Manning Publications.

Ramnivas Laddad (2003), Aspect-Oriented Programming Will Improve Quality, *IEEE software*, **20**(6), 90-91.

Roger Alexander (2003), The Real Costs of Aspect-Oriented Programming, *IEEE software*, **20**(6), 92-93.

Um InSup, Lee HongChul, Kang Jungyun (2004), The Analysis Method of Integrated Logistic System using Evolution Strategies and Data Envelopment Analysis, *Journal of the Korea Society for Simulation*, **13**(4), 17-29.

**김태호**
고려대학교 산업시스템정보공학 학사
현재: 고려대학교 산업시스템정보공학과 석사 과정
관심분야: 시스템통합, SCM, SOA

**엄인섭**
고려대학교 수학, 산업시스템정보공학 학사
고려대학교 산업시스템정보공학 석사
현재: 고려대학교 산업시스템정보공학과 박사과정
관심분야: 시뮬레이션 모델링 및 분석, SCM, 물류시스템

**이홍철**
고려대학교 산업공학 학사
Univ. of Texas 산업공학 석사
Texas A&M Univ. 산업공학 박사
현재: 고려대학교 정보경영공학과 교수
관심분야: 생산 및 물류 정보시스템, SCM