

로봇 응용 생산시스템의 시뮬레이션 프로그램 개발을 위한 중간모델

국금환*

An Intermediate Model for Development of a Simulation Program of a Production System with Robots

Kum-Hoan Kuk*

ABSTRACT

In this study, an intermediate model is presented as a new method for development of a parametric simulation program. This model enables us to analyze effectively the static and dynamic structure of a real production system. The static structure of the real system can be modelled in an entity-relationship diagram and the dynamic structure of the real system in a Petri net. The intermediate model consists of an entity-relationship diagram and a Petri net. Using this intermediate model man can not only reduce the time and cost for simulation program development, but also increase the modelling reliability of the developed simulation program. To show the usefulness of this intermediate model, the intermediate models for two production subsystems, manufacturing sub-system and transport subsystem, are set up.

Key Words : Parametric Simulation Program(파라메트릭 시뮬레이션 프로그램), Entity-Relationship Diagram(실체-관계 다이어그램), Petri Net(페트리 넷), Robotic Cell(로봇 셀), Production System(생산 시스템)

1. 서론

생산시스템의 자동화율이 높아지고 생산시스템의 복잡도와 규모가 커짐에 따라 실제 생산시스템의 시뮬레이션 프로그램에 대한 요구도 커지게 되었다. 즉 종래의 설계와 분석용 도구 기능뿐만 아니라 최근 생산시스템의 실시간 제어기능과 가상

공장 구현을 위한 새로운 요구 기능들도 추가되었다.⁽¹⁻³⁾ 최근 생산시스템 자동화 추세 중 하나는 로봇 셀(robotic cell)의 형성이며, 이러한 셀 내에서 산업용 로봇은 이송과 반송 작업을 위한 중요한 결합요소로서 점차 응용 빈도가 커지고 있다.⁽⁴⁻⁷⁾ 또한 상품 시장의 요구인 다품종 소량 생산에 의하여 로봇 셀의 유연성이 중요해 졌다. 이러한 셀의 특징은 로봇 작업순서의 가변성이다.

* 경상대학교 제어계측공학과, 자동화 및 컴퓨터 응용기술 연구소(정회원)

로봇 셀의 점진적 도입에 의하여 전 생산시스템을 구성하는 경우 이러한 고가의 로봇 셀 도입에 따른 투자 위험도를 줄이기 위하여 단위 로봇 셀 내부의 복잡한 관계뿐만 아니라 로봇 셀들 사이의 상호 관계도 정확히 시뮬레이션 할 수 있는 도구가 요구된다. 통상 생산작업지원용 소프트웨어는 개발비용이 크고 유지와 확장이 어렵다. 시뮬레이션 프로그램에 대한 이러한 새로운 요구들은 보다 정도 높은 모델링 기능과 보다 복잡한 여러 생산 부시스템들 사이의 상호작용 모델링을 요구하기에 소프트웨어 개발과 유지 측면에서 더욱 어려움이 커진다. 따라서 타 소프트웨어 제품 개발과 마찬가지로 시뮬레이션 프로그램의 효율적 개발을 위한 새로운 방법이 요구되며 지금도 계속 연구되고 있는 실정이다.⁽⁸⁻¹²⁾

이러한 최근의 시뮬레이션 프로그램에 대한 여러 요구에 따라 파라메트릭 시뮬레이션 프로그램 개발의 경우도 보다 체계적이고 효율적인 방법론의 연구가 요구된다. 물론 이러한 연구는 일반 소프트웨어 제품의 생산자동화를 위한 여러 연구와 동일한 배경을 갖는다. 지금까지 개발된 시뮬레이션 프로그램의 동작 신뢰도는 시뮬레이션 프로그램을 설계하고 개발한 사람들의 작업 능력인 시뮬레이션 대상 시스템의 동작에 대한 이해도와 이해한 내용을 시뮬레이션 프로그램으로 구현하는 능력에 전적으로 의존 할 수밖에 없었다. 즉 실제 시뮬레이션 대상 시스템과 시뮬레이션 프로그램 모델 사이의 간격을 외적으로 표현하는 과정이 없이 전적으로 시뮬레이션 프로그램 개발자의 판단으로 메꾸어 왔다. 또한 개발된 시뮬레이션 프로그램의 모델링 신뢰도 검증도 개발한 시뮬레이션 프로그램을 비교적 단순하고 다양한 여러 실제 생산시스템들에 직접 적용한 후 시뮬레이션 전과정에 대한 사상들을 추적하고 분석함으로써 평가할 수밖에 없었다. 즉, 시뮬레이션 프로그램 자체의 동작 신뢰도는 개발 프로그램의 다양한 실제 적용을 통해서만 검증해 왔고 시뮬레이션 프로그램 개발 작업 자체의 생산성도 전적으로 시뮬레이션 프로그램 개발자들의 경험과 판단에 의존할 수밖에 없었다.

이러한 배경에서 본 논문의 목적은 파라메트릭 시뮬레이션 프로그램의 개발 작업 생산성을 향상하고 개발작업과정 자체를 통해 개발 프로그램의 모델링 신뢰성을 높이기 위한 “하나의 시뮬레이션 프로그램 개발방법”으로서 “실제 생산 시스템과

시뮬레이션 프로그램 모델 사이의 논리적 중간모델을 제안”하는 것이다. 또한 제안한 중간모델을 부품가공시스템과 운반시스템에 구체적으로 적용하고 이것을 토대로 개발한 로봇응용 생산시스템에 대한 시뮬레이션 프로그램의 구조를 보이고자 한다.

2. 중간모델

2.1 정의

본 논문에 사용된 주요 개념을 다음과 같이 정의한다.

- 시뮬레이션 프로그램(simulation program) : 시뮬레이션 프로그램은 명령어들로 만들어진 컴퓨터 프로그램으로서 하나의 동적 현상의 시뮬레이션을 위한 독립된 모델이고, 파라메트릭 시뮬레이션 프로그램은 특정 대상 시스템의 모델을 모델 파라미터에 의하여 정의해 주는 시뮬레이션 프로그램을 말하며 본 논문내의 시뮬레이션 프로그램은 파라메트릭 시뮬레이션 프로그램을 의미함.
- 로봇 셀(robotic cell) : 로봇 셀은 로봇을 가진 자동화된 가공 셀로서 몇 개의 가공기계와 버퍼 그리고 이것들을 상호 연결해주는 로봇으로 구성되며 부품가공 셀 내의 물류흐름 제어는 셀 제어가 담당함.

2.2 시뮬레이션 프로그램 모델의 개발 추이와 중간모델

생산시스템의 시뮬레이션 프로그램 모델 개발 관련 연구동향으로부터 추출한 그 동안의 주 연구내용은 다음과 같다.

- 실제 생산 시스템의 모델링 작업을 입력 데이터로 단순화하고 범용 프로그래밍 언어를 사용하여 시뮬레이션 모델의 이식 용이성을 개선함.⁽¹³⁻¹⁶⁾
- 그래픽 기능에 의하여 데이터 입·출력과 프로그램 흐름추적 기능을 향상시키고 가상현실 구현을 추구함.⁽⁸⁾⁽¹⁷⁻¹⁹⁾
- 인공지능(Artificial Intelligence) 기능에 의하여 시뮬레이션 모델이 지원하는 의사결정의 질을 높이고 페트리 네트(Petri net)를 최종 모델링

도구로 활용함.⁽²⁰⁻²³⁾

그러나 시뮬레이션 프로그램 개발작업 자체의 복잡도를 낮추기 위한 연구와 시뮬레이션 프로그램 모델의 개발과정 체계화를 통한 시뮬레이션 프로그램 자체의 모델링 신뢰성을 높이기 위한 연구는 상대적으로 미흡한 실정이다.

시뮬레이션 프로그램이 실제 생산시스템에 대한 하나의 모델로서 기능을 갖기 위해서는 모델 파라미터 결정을 통해 실제 시스템과 동정되며 이를 통해 실 시스템의 거동에 대한 충분한 예측능력을 가져야 한다. 이를 위해서 실제 생산 시스템 내부의 여러 중요한 정적·동적 구조와 생산시스템과 외계의 상호관계가 충실하게 모델에 구현되어야 한다. 이를 위하여 개발 첫 단계에 실제 시스템의 내부구조 및 외부와 관계가 정확히 분석되어야 하며, 둘째 단계에 이러한 분석을 통해 파악된 내용이 프로그램 형태로 구현되어야 한다. 개발 최종 단계에서 개발된 시뮬레이션 프로그램 모델을 활용하여 주어진 실제 생산시스템의 활동을 실험해 가며 개발한 시뮬레이션 모델과 실 시스템의 동정정도를 검증하는 작업이 수행된다. 시뮬레이션 모델 개발작업시 최초 작업단계에서 파악되고 이해된 실 시스템내의 이러한 구조와 동작은 통상 프로그램 개발자의 이해로만 존재하며, 파악된 내용의 타당성은 개발 최종 단계에서만 확인될 수 있다. 이것은 제품 품질검사에서 중간검사 없이 완제품 검사만 함으로서 제품 생산비용이 과대하게 커지는 것과 동일한 문제를 초래한다. 또한 이러한 개발방법은 프로그램 개발의 생산성 향상을 위한 표준화된 도구나 방법론 없이 개발자 개인의 경험과 판단에만 전적으로 의존하기에 대규모의 복잡한 시뮬레이션 프로그램을 여러 사람이 함께 개발할 때 개발비용이 과대해 진다.

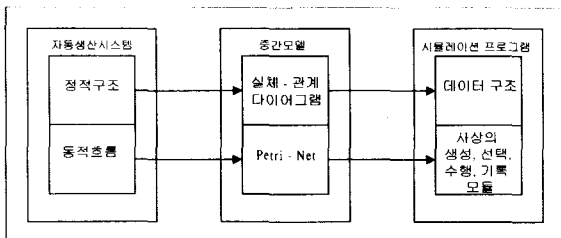


Fig. 1 Intermediate model between real production system and simulation program

본 논문에서 이러한 문제점을 개선하기 위하여 상기 개발 첫 단계의 작업결과를 명시적으로 표현할 수 있는 하나의 방법으로서 Fig. 1 과 같은 하나의 중간모델을 제안한다. 즉 전체 생산시스템의 정적구조를 나타내기 위하여 시스템을 구성하는 요소들간의 관련성을 실체-관계 다이어그램(Entity-Relationship Diagram)으로 표현하고, 전체 생산시스템의 동적 흐름인 요소들의 상태와 관련성이 사상(event) 혹은 사건을 통해 변해 가는 과정을 페트리 넷(Petri Net)로 표현한다.⁽²⁴⁻²⁵⁾

제안한 중간모델은 동적 시스템에 대한 상태방정식을 유도하기 위하여 본드 그래프를 하나의 중간모델로 활용하는 경우와 동일한 효과를 얻을 수 있다. 즉 중간모델을 통해 실제 생산시스템을 보다 체계적으로 분석할 수 있고, 분석 결과의 객관적 타당성을 중간모델을 통해 확인할 수 있으며, 얻은 중간모델을 보다 조직적으로 신속하게 최종 프로그램에 구현할 수 있다. 제안한 중간모델을 이용하여 실제 생산시스템에 대한 하나의 시뮬레이션 프로그램을 개발할 때 요구되는 두 가지 핵심 작업인 프로그램의 데이터 구조 설계작업과 프로그램의 모듈 설계작업을 다음절의 내용과 같이 보다 체계적으로 추진할 수 있다.

2.3 실체-관계 다이어그램과 데이터 구조

자동 생산시스템을 구성하는 기본요소들(영구 시스템요소, 일시 시스템요소)을 정의한 후에 이러한 기본요소인 객체(object)들을 시뮬레이션 프로그램 내에서 실체(entity)로 대응시켜 나타낸다. 이때 영구 시스템요소는 시뮬레이션 전과정을 통하여 시스템 고유 요소로 계속 잔존하는 로봇, 가공기계, 버퍼등을 의미하며, 일시 시스템요소는 시뮬레이션 과정중에 계속 만들어져서, 일정기간 동안 시스템속에 잔존하다가 생산 시스템을 떠나는 요소인, 가공부품, 가공공구, 측정공구등의 시스템입·출력 요소를 의미한다. 각 객체의 속성은 정적·동적·통계적 속성을 나타내는 데이터 값을 갖게 된다. 그리고 생산 시스템내 두 개 이상의 실체들간의 관계는 실체-관계 다이어그램(Entity-Relationship Diagram)으로 표현된다. 실체-관계 다이어그램의 관계는 프로그램 개발자가 정의한 관계이다. 구체적으로 생산시스템을 구성하는 부시스템들과 부시스템 요소들 사이의 실체-관계 다이어그램에 의하여 부시스템 요소들의 특정 부시스

템에의 소속성을 파악하고 나타낼 수 있으며, 한 부시스템내 영구요소와 영구요소, 영구요소와 일시요소 상호간의 소속성 또는 관계를 실제-관계 다이어그램에 의하여 쉽게 파악하고 나타낼 수 있다. 이러한 시스템 요소들 사이의 관계는 시스템 요소의 종류와 기술적 특성에 의하여 특징 지을 수 있으며 생산시스템의 복잡도에 비례하여 이러한 관계들의 개수는 증가한다.

1) 생산시스템의 실제-관계 다이어그램

생산시스템을 실제-관계 다이어그램으로 나타내기 전에 다음 두 가지 사항을 전제로 하였다. 첫째, 생산시스템이 네 개의 기본 부시스템인 부품가공·조립·운반·저장 부시스템들로 이루어진다. 둘째, 각 부시스템은 Fig. 2 와 같이 모두 동일적으로 셀 레벨로 취급된다. 셀 요소* 집합의 마디와 가지는 로봇 작업점과 이동 경로를 의미한다.

시스템 요소들 사이의 관계는 수정된 실제-관계 다이어그램에서 기능(1:1, 1:n, m:n)과 의존성(all:all, all:any, any:all, any:any)에 의하여 구분되어 Fig. 3 과 같이 표현된다. 의존성(all:all, all:any, any:all, any:any)은 한 집합의 요소들이 타 집합의 요소들

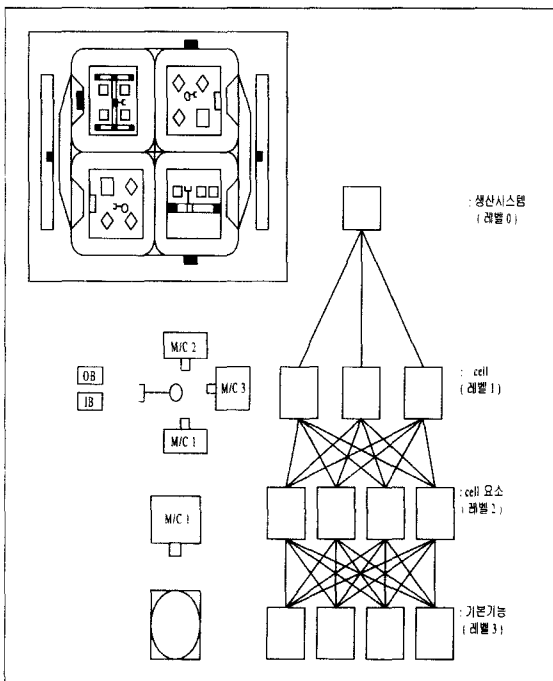
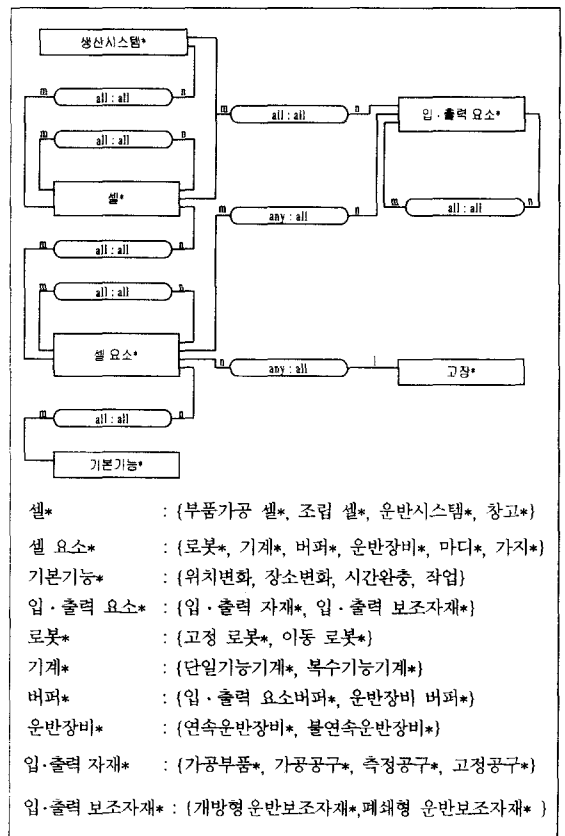


Fig. 2 Division of a production system into system elements

과 독립적으로 존재할 수 있는지를 기술해 준다. 즉, 한 집합의 모든 요소들이 타 집합의 일부 요소들에만 의존하여 존재할 수 있으면 "all:any"의 의존성을 가진다. 구체적인 예로서 Fig. 3 에서 실제 "셀 요소*" 집합 중 버퍼와 마디·가지는 고장을 고려하지 않았기에 "any:all"의 의존성을 가진다. 또한 Fig. 4 에서 로봇 집합과 기계 집합 사이의 관계가 "all:all"의 의존성을 가진 것은 셀내 모든 작업기계의 가공부품 장탈착이 로봇셀내 모든 로봇만에 의하여 자동으로 수행됨을 의미한다. 또한 기능(1:1, 1:n, m:n)은 한 집합의 요소 몇 개가 타 집합의 몇 개 요소와 관계를 맺고 있는지를 기술한다. 즉 한 실제 집합 A의 한 요소가 타 실



- 셀* : (부품가공 셀*, 조립 셀*, 운반시스템*, 창고*)
- 셀 요소* : (로봇*, 기계*, 버퍼*, 운반장비*, 마디*, 가지*)
- 기본기능* : (위치변화, 장소변화, 시간완충, 작업)
- 입·출력 요소* : (입·출력 자재*, 입·출력 보조자재*)
- 로봇* : (고정 로봇*, 이동 로봇*)
- 기계* : (단일기능기계*, 복수기능기계*)
- 버퍼* : (입·출력 요소버퍼*, 운반장비 버퍼*)
- 운반장비* : (연속운반장비*, 불연속운반장비*)
- 입·출력 자재* : (가공부품*, 가공공구*, 측정공구*, 고정공구*)
- 입·출력 보조자재* : (개방형 운반보조자재*, 폐쇄형 운반보조자재*)

*을 가진 단어 : 집합
 *이 없는 단어 : 집합요소
 고장* 은 한 원소를 갖는 집합

Fig. 3 Entity-relationship model of a production system

체 집합 B의 복수 개 요소와 관계를 가지나, 집합 B의 한 요소는 기껏해야 집합 A의 한 요소와 관계를 가질 때 "1:n"으로 나타낸다.

하나의 구체적 예로서 Fig. 3에서 실제 "고장*"을 요소 하나를 가진 집합으로 간주했기에 "셀 요소*" 집합과 "고장*" 집합 사이의 기능이 "n:1"이 된다.

2) 데이터 저장구조

실체-관계 다이어그램 내용을 나타내기 위한 데이터 구조는 시뮬레이션 프로그램내 데이터 영역에 구체적으로 구현되어야 한다. 이러한 구현은 데이터의 저장구조와 적용하는 알고리즘을 통해 이루어진다. 최적의 저장구조를 결정하기 위하여

세 가지 대표적 저장구조인 선형 벡터, 연결된 리스트, 수지형 구조의 장단점을 상호 비교하여 장점을 결합하고 단점을 최대한 제거해야 한다.⁽²⁶⁾

시뮬레이션 프로그램내 데이터의 저장구조 결정을 위하여 아래와 같은 평가항목을 도출하였다.

- 시뮬레이션 과정중의 데이터 량의 변화(동적 혹은 정적)
- 시뮬레이션 과정중의 데이터 참조빈도
- 미래의 데이터 수정빈도

Table 1과 같이 시뮬레이션 프로그램의 모든 데이터들은 상기 평가항목에 따라 평가되어 각 데이터에 대한 주속성과 부속성을 결정한 후에, 각 저장구조들에 대한 하중계수를 이용하여 최종평가되었다. 이때 세 가지 저장구조는 각각 동적 저장능력, 데이터 참조와 평가의 소요시간, 저장구조의 수정과 프로그래밍의 용이성에 의하여 상대평가된 후 이러한 평가결과가 하중계수로 활용되었다. 구체적인 하나의 예를 들면, Table 1의 9종 데이터들 중 첫 째 데이터인 "생산시스템 요소의 식별" 데이터에 대한 주속성은 상기 평가항목중

- 시뮬레이션 과정중의 데이터 참조 빈도"이고, 부속성은 "- 미래의 데이터 수정빈도"가 된다. 이 "생산시스템 요소의 식별" 데이터에 대한 Table 1 저장구조 값은, 주속성인 "- 시뮬레이션 과정중의 데이터 참조 빈도"에 해당되는 "데이터 참조와 평가 소요시간"의 하중계수(Table 1의 하단부)에 주속성의 하중계수 1을 곱한 값에 부속성 "- 미래의 데이터 수정빈도"에 해당되는 "데이터수정과 개발작업 용이성"의 하중계수에 부속성의 하중계수 0.5를 곱한 값을 더하여 얻게 된다 (선형 벡터: $10*1 + 5*0.5 = 12.5$).

2.4 페트리 넷트와 모듈설계

불연속 시스템의 동작은 임의 시각에 발생한 사상들로 명확히 모델링 될 수 있고, 시스템의 동작 제어는 사상들의 스케줄링을 통하여 행해진다. 사상은 한 시점에서의 상태변화를 의미하며 사상 자체는 시간경과를 조금도 동반하지 않는다. 불연속 시스템의 시간흐름은 이러한 사상들의 진행순서에 따라 정해진다. 이때 이러한 사상들은 시뮬레이션 중에 시뮬레이션 프로그램의 진행에 따라 계속 생성되고 프로그램 진행 중에 제어되어야 한다. 페트리 넷트를 이용하여 시뮬레이션 대상내의 동적

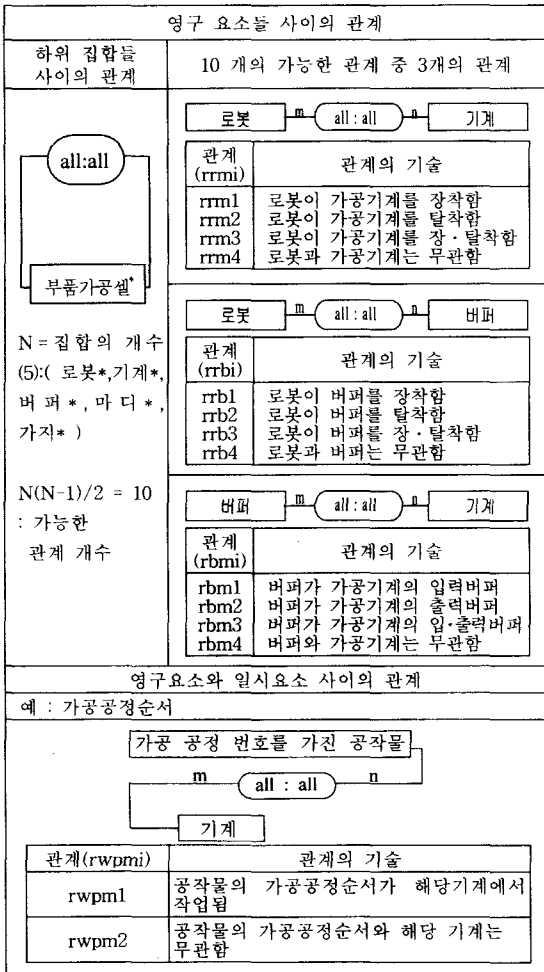


Fig. 4 Relationship between elements of a robotic cell

흐름을 검토할 수 있고, 동시에 시물레이션 대상이 갖는 기본상태와 기본사상 및 시스템요소들의 공통천이에 의한 상호접속 등을 파악할 수 있다.

Table 1 Data structures of a simulation program

데이터	저장 구조		
	선형 벡터	연결된 리스트	수지형 구조
생산시스템 요소의 식별	12.5	0	10.0
생산시스템 요소의 부가 속성	10	0	12.5
영구 시스템요소들 사이의 관계	12.5	0	10.0
영구요소와 일시요소 사이의 관계	12.5	0	10.0
일시요소들의 리스트	5	10	2.5
운반경로내 단위구간	5	10	2.5
이·반송 작업요구들의 리스트	5	10	2.5
사상들의 리스트	5	10	2.5
이송작업 리스트내 각 기계의 포함번호	5	10	2.5

□ : 결정한 저장구조

세 가지 저장구조의 하중계수:

(동적 저장능력: 선형벡터:0, 연결된 리스트:10, 수지형 구조:0;

데이터참조와 평가 소요시간 : 선형벡터:10, 연결된 리스트:0, 수지형 구조:5;

데이터수정과 개발작업 용이성: 선형벡터:5, 연결된 리스트:0, 수지형 구조:10)

데이터속성의 하중계수:(주 속성:1; 부 속성:0.5)

Table 2 Classifying characteristics of control rules

분류특성	예		
	EX1	EX2	
관련 시스템 요소의 종류	- 영구요소에 관련된 제어규칙	○	
	- 일시요소에 관련된 제어규칙		○
	- 영구요소와 일시요소 모두에 관련된 제어규칙		
시스템상태의 반영	- 제어규칙 적용시 시스템 상태를 고려하는 제어규칙		
	- 제어규칙 적용시 시스템 상태와 무관한 제어규칙	○	○
관련된 시스템 활동의 확정성	- 확정적 프로세스에 관련된 제어규칙		○
	- 텐덤 프로세스에 관련된 제어규칙	○	

○ : 만족함

EX1 : 로봇 셀 요소의 고장상태를 기술하는 제어규칙

EX2 : 수행할 반송작업을 선정하는 제어규칙

페트리 넷에서 실현된 상태는 하나의 마크로 채워진다. 하나의 사상이 발생하면 이 마크는 그 사상의 조건상태로부터 결과상태로 옮겨진다. 이때 실제 시스템내의 상태변화는 대응하는 페트리 넷의 사상발생으로 표현된다.

1) 프로그램의 모듈설계

본 논문에서는 기존 연구들과 같이 페트리 넷 자체를 실제 생산시스템에 대한 하나의 최종 모델로 직접 활용하지 않고 파라메트릭 시물레이션 프로그램 개발을 위한 중간모델로 활용하였다. 페트리 넷을 중간모델로 활용하는 경우 페트리 넷로부터 얻어진 정보들은 시물레이션 프로그램 개발의 주요 작업들을 위한 핵심정보가 된다. 구체적으로 다음의 “- 기본상태 ->”에서 시작되는 세 줄의 개조식 요약 내용과 같이 시물레이션 프로그램을 구성하는 기본모듈 네 가지 중 사상생성·사상선택·사상수행 모듈을 구성하는 프로시저들은 페트리 넷을 이용하여 결정할 수 있으며, 나머지 기본모듈인 사상순서 제어모듈은 생성된 사상들 중에서 가장 빠른 발생시각을 갖는 사상을 선택하여 시물레이션 시계를 전진시키고 다른 세 개의 기본모듈을 제어하는 역할을 한다.

- 기본상태 → 사상생성과 사상수행의 모델링
- 기본사상 → 사상수행과 사상선택의 모델링
- 시스템 요소들의 공통천이에 의한 상호접속 → 사상선택의 모델링

이때 실제 시스템의 여러 다양한 동작들이 이러한 기본상태와 기본사상의 조합에 의하여 모델링될 수 있어야 한다. 이를 위해 특정 시물레이션 대상에 대한 기본상태와 기본사상 및 시스템 요소들의 공통천이에 의한 상호접속의 정보를 얻기 위한 페트리 넷 모델링 대상인 실제 시스템 동작은 아래 조건을 반드시 만족시키는 가능한 최소의 시스템이어야 한다.

- 시스템 동작이 시스템 내부요소들 사이의 실제적 의미를 갖는 가능한 모든 관계를 포함할 것.
- 시스템 동작이 시스템 경계에서의 가능한 모든 사상을 포함할 것.

2) 프로그램의 모듈분할

모듈화는 문제의 복잡도를 줄이기 위한 하나의 프로그래밍 기법이다. 각 생산 부시스템에 대한 독립적인 시물레이션 모델을 작성하기 위하여 각 부시스템의 사상순서 제어모듈은 상호 독립적인

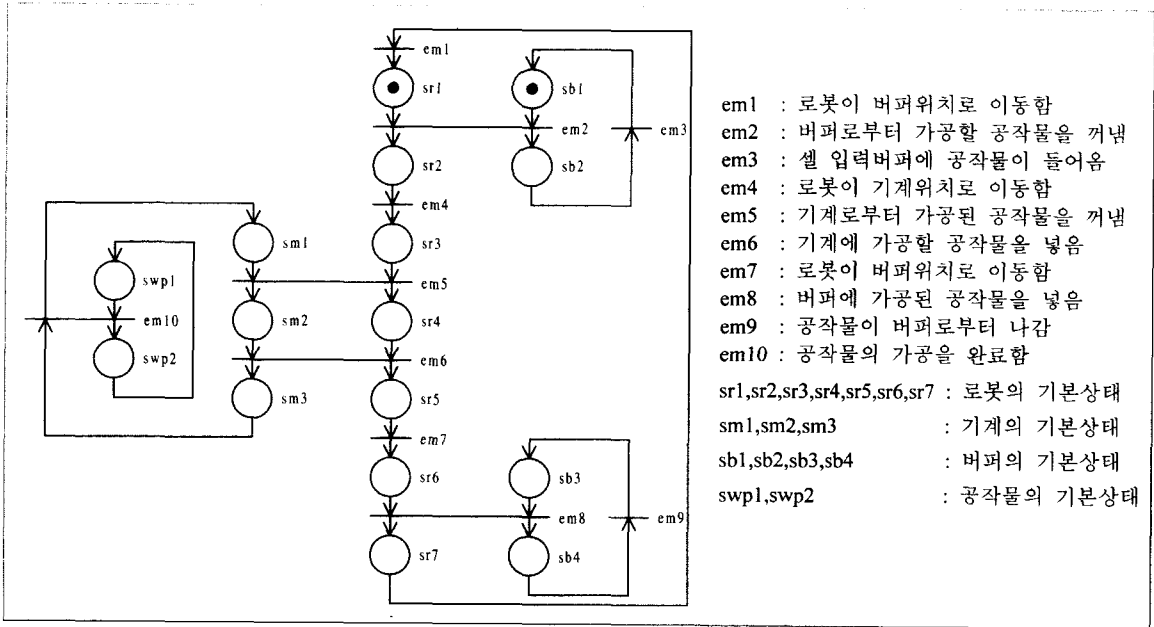


Fig. 5 Petri Net of a robotic cell

고유모듈로 구현하였다. 두 개 이상의 생산 부시스템이 상호 결합되어 하나의 생산시스템을 구성할 때는 각 부시스템에 대한 전용 사상순서 제어 모듈을 호출할 수 있는 상위의 사상순서 제어모듈이 요구된다. 또한 새로운 제어규칙의 추가나 모델 정밀도 수정작업이 용이하도록 입·출력요소 관련 사상생성, 시스템 요소의 고장 관련 사상생성, 사상선택, 사상수행, 사상기록, 데이터 입력, 데이터 출력등의 작업은 모두 독립된 모듈로 구현하고, 상위의 생산 부시스템 전용 사상순서 제어모듈로부터 호출된다. 또한 시뮬레이션 데이터 입력 모듈과 시뮬레이션 결과 출력 모듈을 독립적인 모듈로 분리하여 구현함으로써 필요시 프로그램 사용자의 데이터 입출력 방식 변경 요구를 신속히 만족시킬 수 있다.

2.5 제어규칙의 결정

시뮬레이션 프로그램에 구현된 제어규칙은 시스템요소들의 상태에 의존한 실제 시스템의 동적흐름에 대한 결정들을 모델링한다. 각 제어규칙들의 기능은 “페트리 넷상 시스템요소들의 공통천이에 의한 상호접속들과 시스템 경계에서의 사상들”로부터 결정된다. 하나의 제어규칙은 다음 두 가

지 내용을 통하여 정의 될 수 있다.

- 기능.
- 제어규칙의 적용 조건으로서의 시스템 상태.

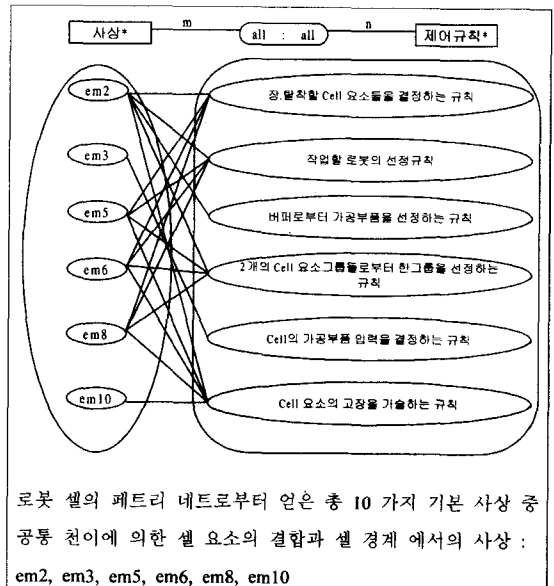


Fig. 6 Relationship between control rules and events

시뮬레이션 프로그램에 구현되는 모든 제어규칙들은 Table 2의 속성에 의하여 통일적으로 파악되고 보다 체계적으로 도출·분류 될 수 있다. 시뮬레이션 프로그램에 구현되는 모든 제어규칙들의 적용에 대한 전제조건은 생산시스템내의 물류흐름과 정보흐름의 시간적 지연없는 상호 동기성이다. 즉 실제 생산 시스템내 의사 결정 작업 순간에 시스템 내부의 물류흐름에 대한 정보가 시간적 지연 없이 얻어 진다는 조건을 가정한 것이다.

3. 중간모델의 적용

3.1 부품가공 로봇 셀의 실체-관계 다이어그램

부품가공 셀내의 이동로봇을 모델링하기 위하여 이동루트의 요소인 마디와 가지를 셀의 영구요소로 간주한다. 즉 로봇 이동루트는 가지들로 나뉘어 지고, 로봇이 가공기계와 버퍼를 장·탈착 할 위치는 마디가 된다. 그러나 운반시스템을 부품가공 셀 위에 맵핑하여 만들어진 셀 내의 마디와

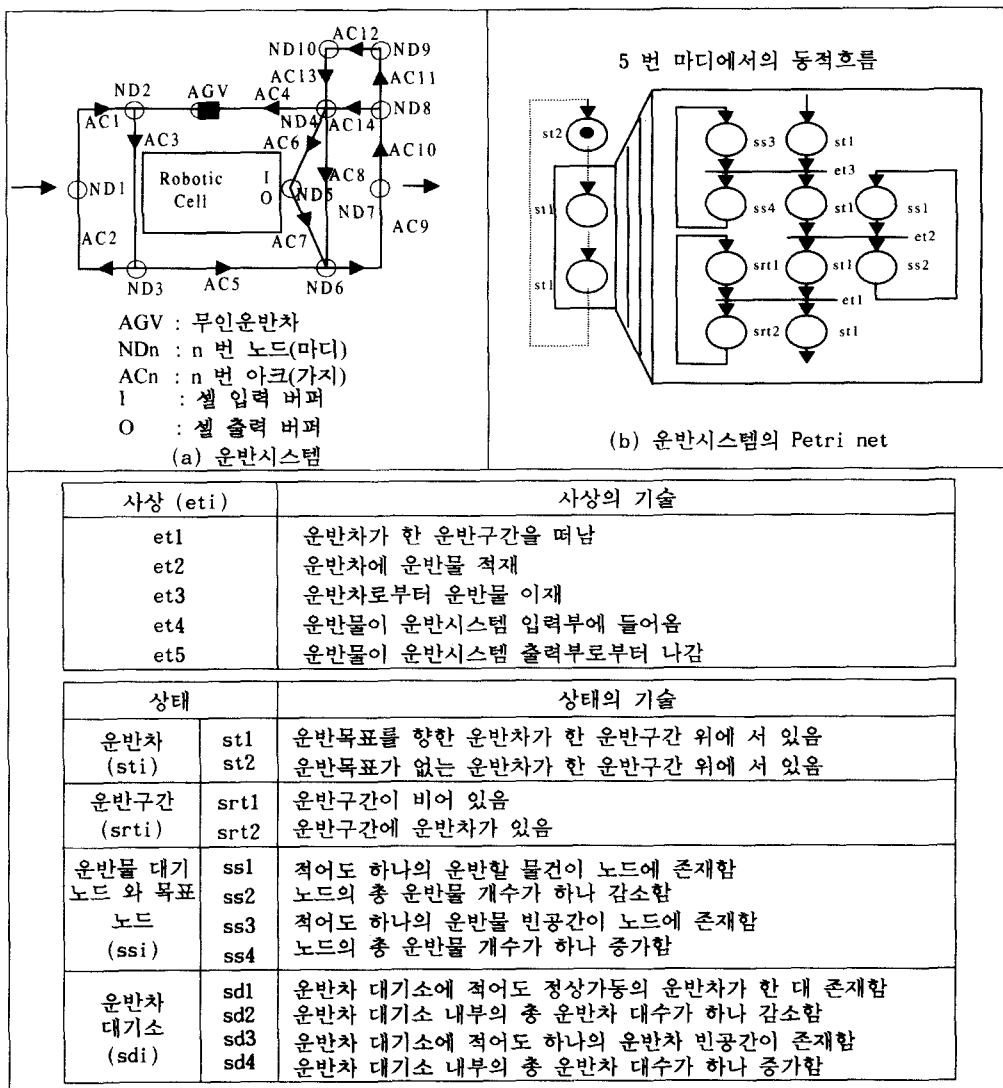


Fig. 7 Petri net of a transport system

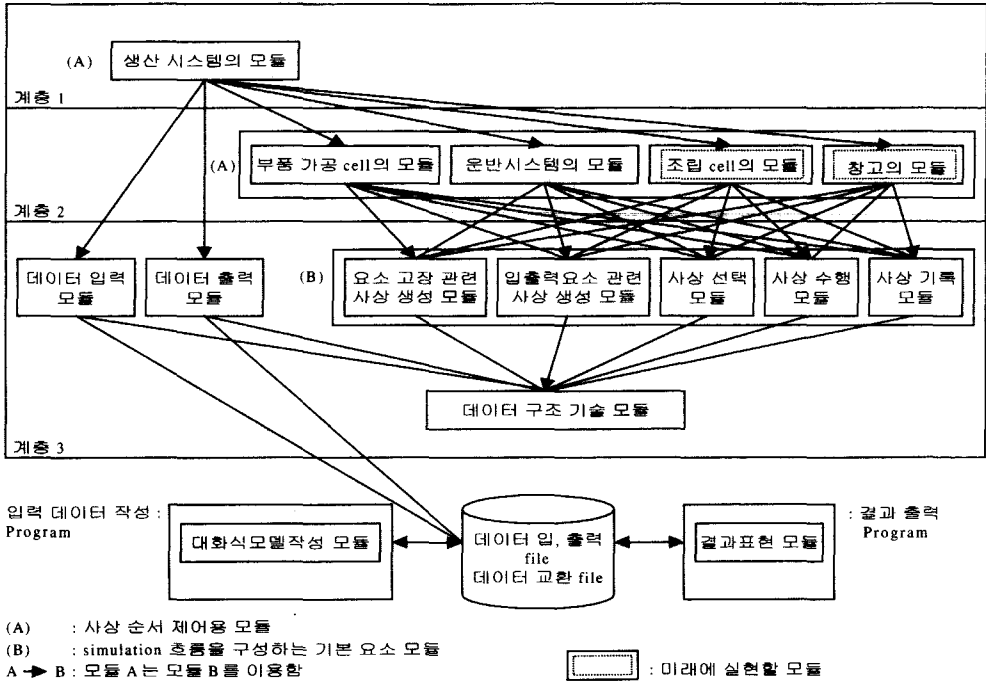


Fig. 8 Module structure of simulation program

가지는 운반 부시스템의 고유요소로 간주한다. 부품가공 셀을 구성하는 영구요소들 사이의 관계와 영구요소와 일시요소의 관계에 대한 실체-관계 다이어그램 Fig. 4 와 같다. 구체적으로 Fig. 4 는 하나의 자동 생산시스템에 대한 E-R 다이어그램 (Fig. 3)의 일부분 “셀 요소” 사이의 관계로서 부분 E-R 다이어그램을 부품가공 셀(부품 가공 로봇 셀)에 적용하여 상세히 작성한 E-R 다이어그램이다. 이 다이어그램에 포함된 관계 정보들(rrmi, rrb, rbmi, rwpmi)은 Table 1 의 영구 시스템요소들의 관계(rrmi, rrb, rbmi)와 영구요소와 일시요소의 관계(rwpmi)에 해당되어 시뮬레이션 프로그램의 데이터구조에 선형 벡터 구조로 구현된다. 특히 특정 공작물에 요구되는 가공작업공정들 역시 가공부품과 가공기계 사이의 관계로 표현되며, 이를 위해 가공작업 순서 번호와 부품 종류번호를 상호 결합하였다.

3.2 부품가공 로봇 셀의 페트리 넷트와 제어규칙

부품가공 로봇 셀에 대한 페트리 넷트는 전술한

모델링 대상 시스템에 대한 두가지 요구조건, 즉 시스템 동작이 시스템 내부요소들 사이의 실제적 의미를 갖는 가능한 모든 관계를 포함할 것과 시스템 동작이 시스템 경계에서의 가능한 모든 사상을 포함할 것을 만족하도록 Fig. 5 와 같이 작성되었다. Fig. 5 로부터 부품가공 시스템에 대한 10 개의 기본 사상과 시스템 요소들 사이의 공통 천이에 의한 상호접속(em2, em5, em6, em8, em10)과 셀 경계에서의 사상(em3,em9)을 얻었다. Fig. 6 은 이러한 사상들과 직접 관련되는 로봇 셀의 모든 제어규칙을 도출하기 위하여 작성한 실체-관계 다이어그램이다. 이렇게 도출된 모든 제어규칙들이 시뮬레이션 프로그램의 사상선택모듈 내부에 일단 모델링되면, 시뮬레이션 수행중에 이러한 제어규칙을 부를 때마다 관련 사상에 참여할 셀 요소들이 결정된다. 한편 로봇 셀 출력버퍼로부터의 공작물 출력방법은 하나의 독립된 제어규칙으로 모델링하지 않고 단순히 하나의 운반작업 요구로 간주하였기에 사상 em9 은 Fig. 6 에 포함되지 않았다. Fig. 6 에서 사상집합과 제어규칙 집합의 관계인 직선이 갖는 의미는 셀 요소의 결정을 나타낸다.

3.3 운반시스템의 페트리 넷트

운반시스템에 대한 기본상태, 기본사상과 시스템 요소들의 공통천이에 의한 상호접속을 결정하기 위하여 Fig. 7 과 같은 페트리 넷트를 작성하였다. 그림상의 노드 1 은 시스템 입력부 이고 노드 7 은 시스템 출력부이다. 노드 5에서 운반차와 셀 입출력 버퍼 사이의 운반물 전달이 이루어진다. 가지 12 는 운반차 버퍼로서 운반차 대기소를 나타낸다. 다음과 같은 노드 순으로 운반차가 이동하는 경우, 대응하는 페트리 넷트로부터 얻은 기본상태와 기본사상은 Fig. 7 하단부와 같다. 이때 노드 10 과 노드 1 사이 그리고 노드 7 과 노드 10 사이는 빈 운반차의 주행구간이고 나머지 구간

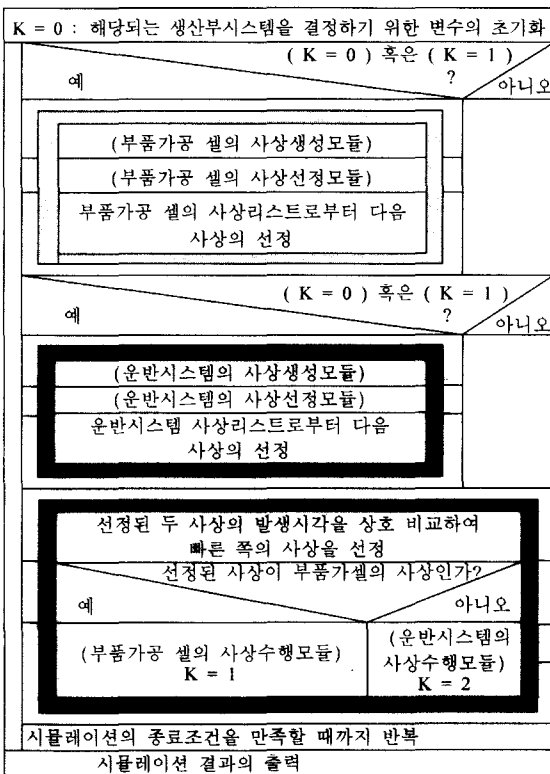
은 운반물을 실은 운반차의 주행구간이다.
 ND10 → ND1 → ND5 → ND7 → ND10

3.4 시물레이션 프로그램의 흐름제어와 적용

본 논문에서 하나의 생산시스템을 구성하는 기본 부 시스템을 네 가지 부시스템(부품가공, 조립, 운반, 저장(창고))으로 가정(Fig. 3) 하였다. 이러한 부시스템 하나만으로 이루어진 생산시스템을 독립적으로 모델링하고 동시에 필요에 따라 네 가지 부 시스템이 다 포함된 생산 시스템을 모델링할 수 있도록 시물레이션 프로그램의 사상순서 제어 모듈을 Fig. 8 과 같이 계층 1, 2 로 나누어 실현하였다. 만약 하나의 생산시스템이 부품가공 부시스템과 운반 부시스템(부품가공 부시스템을 구성하는 로봇 셀들을 상호 연결)만으로 이루어진 경우 이러한 생산시스템의 시물레이션을 위한 하나의 사상 순서 제어 방식을 Fig. 9 의 블록 다이어그램에 나타냈다. 즉, 두 생산 부시스템에 대하여 독립적인 전용 사상순서 제어모듈을 작성한 후 Fig. 8 과 같이 두 개 이상의 생산 부시스템이 상호 결합되어 하나의 생산시스템을 구성할 때 각 부시스템에 대한 전용 사상순서 제어모듈을 호출할 수 있는 상위 사상순서 제어모듈이 요구된다. 이 때 시물레이션 시계의 시간 진행 방법으로 사상(event) 중심 시간흐름 제어방법을 채택하였다. 그 이유는 원하는 정도에 따라 사상을 보다 정밀하게 모델링할 수 있고 이러한 사상들의 시각에 따라 시물레이션 시간 흐름을 정도 높게 추종할 수 있기 때문이다. 이러한 모델링 결과를 일반 프로그램 언어로 나타내기 위하여 본 논문에서는 PASCAL 언어를 사용하였다. PASCAL 과 C 는 모두 동적 메모리 관리 능력을 가지기에 소모 메모리 측면에서는 둘다 우수하지만 PASCAL은 프로그램 해독성 측면에서 보다 우수하다.

4. 결론

본 논문에서 파라메트릭 시물레이션 프로그램의 하나의 새로운 개발방법으로 중간모델을 제안하였다. 이러한 중간모델을 이용하여 실제 생산시스템을 컴퓨터 프로그램 모델로 전환하기 전에 생산시



- : 부품가공셀의 사상순서제어 모듈
- : 운반시스템의 사상순서제어 모듈
- : 생산시스템의 사상순서제어 모듈

Fig. 9 Event sequence control of a simulation program for production system with a machining subsystem and a transport subsystem

시스템의 복잡한 동작을 체계적으로 분석하고 표현할 수 있다. 이를 통해 시뮬레이션 프로그램 개발자가 이해한 내용이 하나의 중간결과로서 외적 표현이나 검증 없이 직접 시뮬레이션 프로그램 모델로 전환될 때 개입될 수 있는 가능한 오류들을 최소화할 수 있다. 특히 GASP·SIMSCRIPT 와 같은 시뮬레이션 전용언어가 아닌 일반 프로그래밍 언어인 FORTRAN·PASCAL·C 등을 이용하여 파라메트릭 시뮬레이션 프로그램을 직접 개발하는 것은 프로그램 개발비용이 상대적으로 크다. 또한 공장 자동화가 진전됨에 따라 복잡한 대규모 자동화 시스템의 제어 도구로 시뮬레이션 프로그램을 직접 활용하는 경우와 가상공장과 같은 사이버 공장 활용수요도 커질 것이 예상되기에 앞으로 파라메트릭 시뮬레이션 프로그램의 개발요구가 증대될 것이다. 왜냐하면 파라메트릭 시뮬레이션 프로그램은 시뮬레이션 정밀도를 보다 용이하게 요구되는 정밀도에 맞출 수 있기 때문이다. 또한 복잡도와 규모가 큰 프로그램의 개발비용을 줄이고 개발작업 자체의 체계화와 단순화를 위해 실제-관계 다이어그램과 페트리 네트를 본 논문에서 제안한 방법대로 활용할 수 있다. 구체적으로 제안한 중간모델을 부품가공 부시스템과 운반 부시스템에 적용하여 실제-관계 다이어그램과 페트리 네트를 작성한 후, 이들을 활용하여 개발한 시뮬레이션 프로그램의 구조를 보였다.

제안한 방법은 그 동안 주력해온 시뮬레이션 프로그램의 사용자 인터페이스 기능향상을 위한 연구에 비하여 상대적으로 등한시되었던 시뮬레이션 프로그램 개발 작업의 생산성과 시뮬레이션 프로그램 자체의 모델링 신뢰성을 향상시키기 위한 하나의 방법이 될 수 있다. 중간모델을 보다 적극 활용하여 시뮬레이션 프로그램 자체의 모델링 신뢰도를 기계적으로 검증하고 시뮬레이션 프로그램 개발작업의 부분 자동화를 위한 연구는 미래의 과제이다.

참고 문헌

1. Y.-L. Sun, Y. Yih, "An intelligent controller for manufacturing cells," *Int. J. Prod. Res.*, Vol. 34, No. 8, pp. 2353-2373, 1996.
2. S.G. Chen, M.Z. Wu, R.K. Li, "Design of a rule-based flexible manufacturing system controller using modified IDEF0 methodology," *Int. J. Prod. Res.*, Vol. 35, No. 10, pp. 2793-2820, 1997.
3. J.S. Smith, B.A. Peters, "Simulation as a Decision-Making Tool for Realtime Control of Flexible Manufacturing Systems," *IEEE Int. Conf. on Robotics and Automation*, pp. 586-590, 1998.
4. G. Handke, "Der Roboter als CIM-Komponente," *Praktiker-Tagung : Prozessintegrier Robotereinsatz*, Muenchen, 26-27 Maerz, 1987.
5. A.G. Burgess, I. Morgan, T.E. Vollmann, "Cellular manufacturing: its impact on the total factory," *Int. J. Prod. Res.*, Vol. 31, No. 9, pp. 2059-2077, 1993.
6. R. Logendran, C. Sriskandarajah, "Sequencing of robot activities and parts in two-machine robotic cells," *Int. J. Prod. Res.*, Vol. 34, No. 12, pp. 3447-3463, 1996.
7. H. Seifoddini, M. Djassemi, "Determination of a flexibility range for cellular manufacturing systems under product mix variations," *Int. J. Prod. Res.*, Vol. 35, No. 12, pp. 3349-3366, 1997.
8. O. M. Uelgen, T. Thomasa, "SmartSim: An object oriented simulation program generator for manufacturing systems," *Int. J. Prod. Res.*, Vol. 28, No. 9, pp. 1713-1730, 1990.
9. J. Haddock, N. Seshadri, V. R. Srivatsan, "A decision support system for simulation modeling," *Journal of Manufacturing Systems*, Vol. 10, No. 6, pp.484-491, 1991.
10. K.R. Christenson, C.A. Dogan, "A simulation generator for dual-card kanban-controlled flow shops," *Int. J. Prod. Res.*, Vol. 33, No. 9, pp. 2615-2631, 1995.
11. Y. Dessouky, C. Roberts, T. Beaumariage, "An objectoriented simulation architecture with real-time capabilities," *Int. J. Prod. Res.*, Vol. 33, No. 9, pp. 2471-2492, 1995.
12. 국금환, "모델 접속 기법에 의한 로봇 응용 생산시스템의 파라메트릭 시뮬레이션모델 개발," *한국정밀공학회지*, 제12권, 제5호, pp. 136-148, 1995.
13. G. Stemmer, "MFSP-Ein Verfahren zur Simulation Komplexer Materialfluss-Systeme," *Diss. Uni. Stuttgart*, 1976.

14. R.J. Mayer, J.J. Talavage, "Simulation of a Computerrized Manufacturing System," NSF Grant Nr. APR 7415256, Report Nr. 4, 1976.
15. W. Grosseschallau, "Simulation von Materialflusssystemen," Teil IV. Foerdern und Heben 30, Nr. 6, pp. 497-503, 1980.
16. G. Vettin, "Verfahren zur technischen Investition-splanung automatisierter flexibler Fertigungsanlagen," Diss. Uni. Stuttgart, 1982.
17. H. Steffens, "Ein Beitrag zur Optimierung der Prozessfuehrungsstrategien automatisierter Foerder- und Materialflusssystem," Diss. Uni. Stuttgart, 1983.
18. S.R. Hill, M.A.M. Rogers, "Practical experience contrasting conventional modelling and data-driven visual interactive simulation techniques," SIMULATION, Springer-Verlag, pp. 271-283, 1986.
19. W.S. Lee, J.H. Kim, J.H. Cho, "A Driving Simulator as a Virtual Reality Tool," IEEE Int. Conf. on Robotics and Automation, pp. 71-76, 1998.
20. I.B. Abdallah, H. EIMaraghy, T. EIMekawy, "An Efficient Search Algorithm for Deadlock-free Scheduling in FMS Using Petri Nets," IEEE Int. Conf. on Robotics and Automation, pp. 1179-1184, 1998.
21. S.I. Caramihai, H. Alla, "On the Synthesis of a Controllable Supervisor for Discrete Processes modeled by Temporal Petri Nets," IEEE Int. Conf. on Robotics and Automation, pp. 586-590, 1998.
22. W.R. Goodall, R. Roy, "Short term scheduling and control in the batch process industry using hybrid knowledge based simulation," Int. J. Prod. Res., Vol. 34, No. 1, pp. 33-50, 1996.
23. R.D. Souza, Z.Y. Zhao, "A case for intelligent representation of dynamic resources in simulation," Int. J. Prod. Res., Vol. 35, No. 12, pp. 3289-3302, 1997.
24. P.P.S. Chen, "The entity-relationship model-toward a unified view of data," ACM Trans. on Database Systems, Vol. 1, No. 1, pp. 9-36, 1976.
25. W. Eberlein, "CAD-Datenbanksysteme," Springer-Verlag, 1984.
26. M.E.S. Loomis, "Data Management and File Structures," Prentice-Hall, 1989.