

생산제어시스템의 시뮬레이션모델 자동생성

이 상훈, 조 현보, 정 무영

포항공과대학교

산업공학과/제품생산기술연구소

Abstract

This paper describes an intelligent user interface to define simulation models from the process and resource models. It also explains an automatic program generator of discrete event simulation model for shop floor control in a flexible manufacturing system. Especially, the paper is focused on the design and development of methodology to automate simulation modeling from the system description. Describing a shop floor control system in simulation is not an easy task since it must resolve various decision problems such as deadlock resolution, part dispatching, resource conflict resolution, etc. The program generator should be capable of constructing a complete discrete simulation models for a multi-product and multi-stage flow shop containing the above mentioned problems.

keyword : production control system, simulation model, automatic program generator, process and resource model

1. 서론

최근에 산업, 정부 및 사회에서 발생하는 문제는 방대하고 복잡해지는 경향이 있다. 이들 분야의 특성과 구성요소는 지속적으로 변화하기 때문에 이들을 제어하고 통제할 안정된 방법이 필요하다. 또한 고객의 수요 다양화와 제품수명 단축화로 시스템의 구성품 추가와 삭제가 빈번히 일어나고 있다. 이와 같은 복잡하고 불안정한 시스템의 문제를 모델링하고 해결하기 위한 방법론으로 시뮬레이션이 가장 많이 활용되고 있다. 실제로 시뮬레이션의 결과를 통하여 얻을 수 있는 효과는 매우 다양하다. 첫째, 새로운 프로세스의 추가나 삭

제가 미치는 영향을 예상하고 분석하여 이의 경제성을 평가할 수 있다. 둘째, 기존의 프로세스의 특성을 파악하여 갑작스러운 수요 증가나 원료 감소에 따르는 대체 방안을 도출할 수 있다. 셋째, 복잡한 공정의 전략 및 기술적인 제어 변수를 발견하여, 이를 실시간 운용에 활용할 수 있다. 뿐만 아니라, 일정계획 (Scheduling) 문제를 해결하여 시스템의 효율을 증대시킬 수도 있다. 이와 같이 비록 시뮬레이션이 생산 시스템이나 서비스 분야에 미치는 영향이 지대하나 이를 구현하고 운용하는데 있어서는 그 만큼 고통이 따르게 된다.

시뮬레이션 모델을 개발하는 입장이나 운용하는 입장에서는 다음과 같은 필수 불가결한 규칙사항 (Requirements) 을 고려해야 한다. 첫째, 시뮬레이션 모델을 개발하기 위해서는 관련된 시스템의 기능과 문제점을 정확하게 확인하고 이해해야 한다. 둘째, 시뮬레이션을 위한 언어를 이해해야 한다. 셋째, 특정한 시스템의 구성요소가 변화되었을 때, 시뮬레이션 모델 또한 그 변화에 대응하여 수정 또는 보완되어야 한다. 특히 이 문제는 엄청난 시간과 노력을 요구하는 유지, 보수 및 관리 문제를 야기하게 된다. 즉, 시뮬레이션의 장점에 비례하여 이의 개발, 운용 및 관리에는 많은 비용이 따른다. 따라서 이의 대안으로 시뮬레이션 프로그램을 쉽게 작성할 수 있는 그래픽 지원 방법이 제공되고 있다. 그러나 이는 단지 시뮬레이션 프로그램의 상위에 존재하는 하나의 주변환경이기 때문에 근본적인 프로세스, 객체의 존재확인 및 획득에는 적합지 못하다. 또한 이 그래픽 지원 방법은 프로세스 모델을 추상화시키지 못하므로 모델 그 자체가 매우

복잡하다. 따라서, 이의 해결책으로 프로세스 모델에 근거한 분석 및 자동화된 시뮬레이션 모델 개발 기법이 필요하게 되고 본 논문에서는 이를 제시하고자 한다.

2. 기존 연구 고찰

프로세스 모델을 정의한 대표적인 방법은 IDEF3(Integration DEFinition 3)이다. IDEF3는 시스템내의 사건(Events, Activities) 들에 시간 개념을 도입하여 순서를 정하는 방법론으로 IDEF3는 제조산업뿐만 아니라 비즈니스 분야에서도 적용시킬 수 있는 모델링 기법이다.

프로그램 생성기에 관해서는 많은 연구가 되어왔다. Mathewson[1984]은 시뮬레이션 모델 생성기를 다음과 같이 정의하고 있다.

' It is an interactive software tool that translates the logic of a model described in a relatively general symbolism into the code of a simulation language and so enables a computer to mimic model behavior.

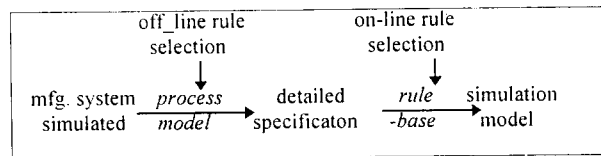
Mathewson[1984]은 Entity cycle diagram을 이용한 프로그램 생성 및 이산 사건 시뮬레이션 모델링에로의 응용에 관한 연구를 하였으며, Ely 와 Kimber[1985]은 하나의 로봇, 최대 다섯대의 머신 그리고 하나의 검사 스테이션으로 구성된 로봇틱 워크 셀(Robotic work cell)에 대한 시뮬레이션 생성기를 디자인하였다. Haddock[1987]은 전문가 시스템을 지닌 시뮬레이션 생성기를 개발하였으며, 이는 사용자에게 코드를 생성시켜줌과 동시에 시뮬레이션 결과까지도 해석, 설명해주도록 디자인 되었다. 또한 최근에는 Christenson 과 Dogan[1995] 이 dual-card kanban에 의해 제어되는 flow shop에 대한 시뮬레이션 모델 생성기를 개발하였다.

전통적인 시뮬레이션 모델링 접근법은 시스템 분석가가 주어진 시스템을 분석하여 필요한 정보를 습득한 후 일반적인 프로그래밍 언어 또는 시뮬레이션 언어를 이용하여 시뮬레이션 모델을 생성하였다. 물론 분석가는 사용하는 시뮬레이션 언어를 잘 알고 있어야 한다. 그러나, 여기서 시뮬레이션 정보 습득 및 프로그램 생성은 자동화 되어질 수 있다. 시뮬레이션 모델 생성을 위한 IDMS (Internally Detailed Model Specification)을 정의 하기 위한 자동 정보 습득 및 평가에 대한 방법론으로는 여러 대안이 제시되었다. Mayer 와

Young[1984]은 자연언어처리를 이용하여 정보를 습득하였으며, graphics interface 또는 interactive dialog monitor 를 통해 시뮬레이션 정보를 습득하는 방법도 제시되었다(Sinclair, et al.[1986]). 그러나 이런 방법들은 SFCS(Shop Floor Control System)을 표현하는 데에는 복잡하고 많은 노력과 시간이 걸릴 뿐만 아니라 수정이 어렵다는 단점을 안고 있다.

3. 접근 방법

본 연구에서는 시뮬레이션 모델의 자동 생성을 위해서 [그림 1]과 같은 방법론을 따른다. 시스템 분석가는 생산 시스템에 관한 정보를 프로세스 모델을 이용하여 모델링하고 프로세스 모델로부터 시뮬레이션 정보가 추출되어지며 이 정보는 IDMS 에 저장되어진다. 시뮬레이션 모델 생성기는 IDMS 에 근거하여 시뮬레이션 코드를 생성한다.



[그림 1] 시뮬레이션 모델 생성 방법

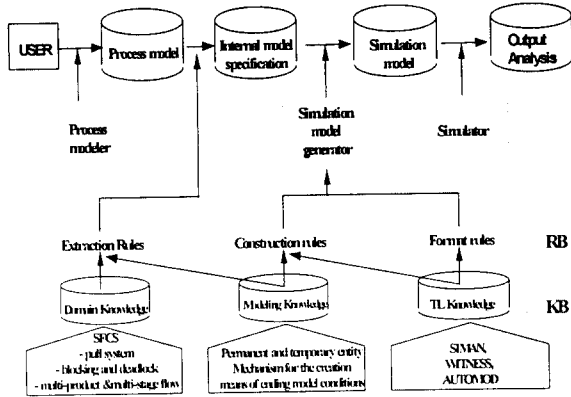
프로세스 모델은 공정 흐름 정보와 함께 랜덤 분포, 버퍼링 규칙등의 정보를 나타낸다. 대부분의 생산 시스템은 프로세스 모델에 의해 가공공정이 표현되어진다. 또한 프로세스 모델의 사용은 시스템을 모델링하는 데 많은 유연성을 제공할 뿐만 아니라 시스템을 표현하는데 길고 복잡한 질문들을 제거할 수 있게 한다. [그림 2]는 이러한 접근 방법에 대한 전체적인 시스템 구성도를 나타낸다.

4. 프로세스 모델에서 시뮬레이션 정보 추출

ASQ(Automated specification acquirer)는 사용자 입력에 대한 질문을 하거나 사용자 선택을 위한 메뉴를 제시함으로써 프로세스 모델에 시뮬레이션 정보를 추가, 그리고 시뮬레이션을 위한 정보를 추출 IDMS에 저장한다. ASQ의 개발은 다음과 같은 두 단계로 이루어진다.

- ① 1단계 : Shop Floor Control에 필요한 정보를 정의하고, 사용자로부터 적당한 정보 추출 방법을 결정한다.

② 2 단계 : 시뮬레이션 모델링 지식에 근거한 질의와 메뉴를 구성한다.



[그림 2] 시스템 구성도

4.1 Conceptualization

시뮬레이션 모델을 생성하기 위해서 세 종류의 지식이 요구되어진다. 첫째, 도메인 지식은 생산 제어 시스템에 관한 지식을 나타내며, 둘째, 시뮬레이션 모델링 지식은 일반적인 모델링 기법에 관한 지식을 나타낸다. 끝으로, 타겟 시뮬레이션 언어에 대한 지식을 필요로 한다. 정보 추출 룰의 예제는 다음과 같이 정의 되어진다.

if the mobile components of the system are parts, the temporary entities are called parts so create a temp_entity structure for parts.

4.2 시뮬레이션 정보 추출 (Extraction process)

t_e_info 는 프로세스 모델로부터 temporary entities 의 정의 및 도착 공정에 대한 정보를, p_e_info 는 프로세스 모델 풀(pool)로부터 permanent entities 의 정의에 대한 정보를 저장하고 있다. t_e_type 에 대해 공정 흐름 정보는 생산 제어 시스템의 동적인 면을 정의하는 것으로, 각각의 temporary entity 에 대한 공정 흐름 정보는 프로세스 모델로부터 추출되어진다. 또한 각 공정, 또는 분기점에서의 결정 문제(Decision problems)에 대한 룰도 함께 정의되어진다. 마지막으로, 필요한 통계치와 시뮬레이션 종료에 관한 조건이 프로세스 모델로부터 정의 되어진다. 이와 같은 모델 정의 시나리오는 IDMS(Internally Detailed Model Specification)에 저장되어져 시뮬레이션 모델 생성을 위한 기반 정보로 사용되어진다.

4.3 Internally detailed model representation

IDMS는 시뮬레이션 정보 추출뿐만 아니라 시뮬레이션 모델 생성에도 이용되어지므로 IDMS의 디자인은 이 두 기능을 동시에 만족시킬 수 있도록 정의 되어져야 한다. IDMS 디자인의 목적은 특정의 시뮬레이션 언어에 의존하지 않는 generic data structure를 정의하기 위한 것이며, 이는 생산 제어 시스템에서 필요로 하는 정보를 충분히 표현할 수 있어야 하며, 주어진 시스템을 모델링하기 위해 필요한 정보를 유지하고 있어야 한다.

5. 시뮬레이션 모델 생성기 개발

5.1 Formalization

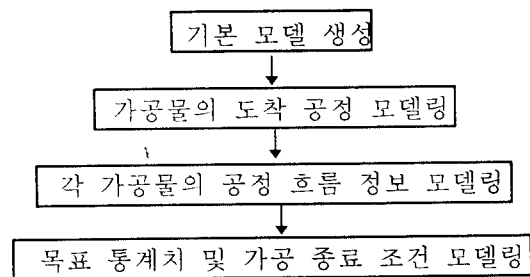
여기서, Formalization이란 위의 세 가지 종류의 지식을 룰로 재구성하는 것을 말한다. 예를 들면, 도메인 지식과 모델링 지식을 이용하여 정보 추출 룰을 구성하고 모델링 지식과 타겟 시뮬레이션 언어에 대한 지식을 이용하여 모델 생성 룰베이스를 형성한다. 룰의 형태는 아래와 같다.

condition(s) → action(s)

정의된 조건들이 만족되면 그에 상응하는 액션을 취함으로써 룰을 실행하게 된다.

5.2 시뮬레이션 모델 생성 (Construction process)

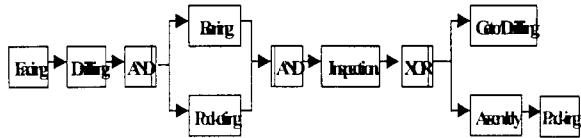
모델 생성물을 이용하여 아래 [그림 3]에서와 같이 시뮬레이션 모델을 생성하고, 포맷 룰을 이용하여 타겟 언어에 대한 에러를 검사, 수정함으로써 완전한 시뮬레이션 프로그램이 생성되어진다.



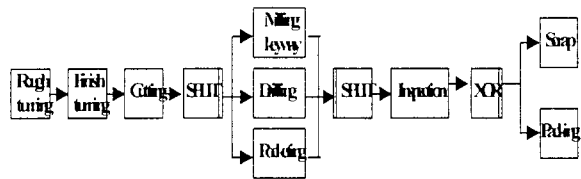
[그림 3] 시뮬레이션 모델 프레임 생성

프로세스 모델에서의 각 공정은 가상의 워 스테이션으로 간주하여 시뮬레이션 모델에서는 모두 스테이션으로 변환되어진다. SFCS은 풀(pull) 시스템의 성격도 함께 지니고 있으며

로 가공물이 한 스테이션에서 다른 스테이션으로 옮겨가기 위해서는 다음 스테이션에서의 리소스가 프리(free)해야한다. 그렇지 않으면, 가공물은 AS/RS 또는 다른 버퍼로 옮겨 져야한다. 여기서는 각 공정을 가상의 워 스테이션으로 간주하였기 때문에 하나의 리소스는 여러곳의 워 스테이션에서 사용가능하게 된다.



(a) PartType A 의 프로세스 모델



(b) PartType B 의 프로세스 모델

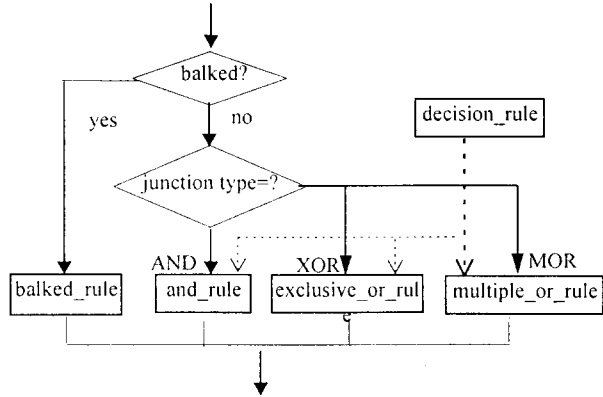
[그림 4] PartType A, B에 대한 시나리오

예를 들면, [그림 4]의 AND junction의 경우, boring과 pocketing은 둘 다 가공을 해야 하지만 둘 간의 순서는 상관없다. 이런 경우 프로세스 순서는 미리 정의되어진 룰에 의해 오프라인 상에서 미리 결정되어질 수도 있고, 시뮬레이션 도중에 어떤 룰에 의해 프로세스 순서가 결정되어질 수도 있다. 예를 들면, 오프라인 프로세스 모델에서 shortest processing time first 룰에 의해 공정 순서가 미리 결정되어질 수도 있고, 시뮬레이션 도중에 리소스의 상황에 따라 순서가 결정되어질 수도 있다. [그림 5]는 라우팅 룰에 관한 분기 로직의 예를 나타낸다.

6. 결론

생산량의 변화를 예측하고 수요의 변화에 따르는 생산량 결정 등 다방면에서 시뮬레이션의 역할은 지대하다. 그러나 지금까지는 각 공정마다 시뮬레이션 모델을 독립적으로 구축했기 때문에 전체적인 통합화가 어려웠으며 시간과 노력의 소모가 많았다. 즉, 어떤 공정에 적당하게 개발된 시뮬레이션 모델은 다른 공정에 응용할 수 없을 뿐만 아니라 공정 변화에 따르는 시뮬레이션 모델의 변화를 융통성

있게 조절하지 못하였다. 본 연구에서는 어떠한 공정이 어떻게 변화되던지 프로세스 모델링 단계에서 모두 흡수되므로 시뮬레이션 모델과 무관해진다. 프로세스 모델링은 추상적이며 자연적이기 때문에 한 공정의 수정, 보완 또는 다른 공정에의 응용이 용이하게 된다. 결과적으로 주어진 생산환경의 공정의 변경, 삭제 및 추가에 따르는 효과를 신속하게 분석할 수 있게 한다.



[그림 5] 분기 로직

참고 문헌

- Christenson, K.R., Dogan, C.A., A simulation generator for dual-card kanban-controlled flow shops, *International journal of production resreach*, 33,9,1995.
- Haddock, J., An expert system framework based on a simulation generator, *Simulation*, 2, 1987.
- Mayer, R.J., Young, R.E., Automation of simulation model generation from system specification, *Proceedings of 1984 Winter Simulation Conference*, Dallas, TX, 1984.
- Mathewson, S.C., Simulation program generators, *Simulation*, 1974.
- Shannon, R.E., Introduction to simulation using SIMAN, McGRAW-Hill, 1991.
- Sinclair, J.B., Madala, S., A Graphical interface for specification of extended queueing network models, *Proceedings of the 1986 Fall Joint Computer Conference*, Dallas, TX, 1986.