

Agent Based Modeling 기법을 활용한 시스템다이내믹스 모델링

전 소 연((주)시스템믹스)
이 혜 준(가톨릭대학교 경영학과)
곽 상 만((주)시스템믹스)

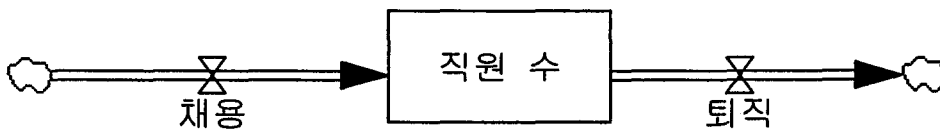
초 록

A system dynamics model is developed to investigate the applicability of the agent based modeling concept in the system dynamics model. The assumed problem is to forecast the size and structure of the organization with the developing market environment. The agent based modeling concept is applied to the organization part, and the other parts of the model such as market, facilities, etc. are developed with the traditional system dynamics technique.

The simulation results show the agent based modeling part can be combined with the traditional system dynamics modeling with more precisions. However, the complexity increases and the simulation times are longer than those of the traditional method.

I. 서론

시스템다이내믹스 기법의 특성 중 하나는 변수의 연속성이다. 예를 들어, 조직에 대한 시스템다이내믹스 모델의 경우 아래 그림과 같이 직원 수 혹은 이에 해당하는 변수를 레벨 변수로 취급하게 되는데, 일단 레벨로 유입된 변수(직원)는 그 History에 관계없이 모두 동일하게 취급하게 된다. 이 과정에서 직원 수는 인원입에도 불구하고 반드시 정수가 되지는 않는다.



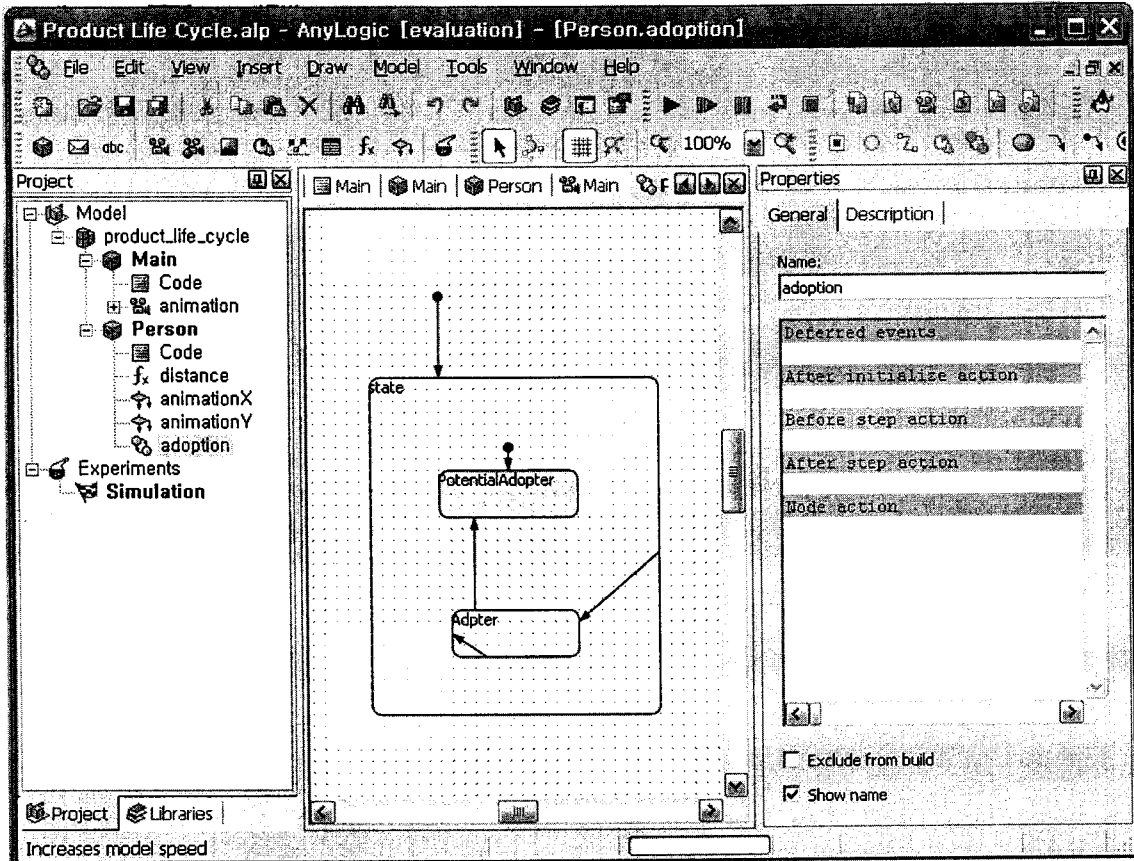
[그림 1] 조직에 대한 시스템다이내믹스 모델의 일부(예)

이러한 시스템다이내믹스의 특성 때문에 조직원의 특성에 따라 그룹화 하는 과정 (Aggregation Process)이 반드시 필요한데, 이와 같은 그룹화 과정에는 경험이 필요하며 이는 중요한 가정사항의 하나가 된다.

그룹화 과정은 규모가 작은 조직일 경우 다음과 같은 문제점을 갖는다.

- 소수점으로 표현되고 1 이하의 값이 중요한 의미를 갖게 된다.
- 특히 Rate 변수에서 1 보다 작은 수의 직원을 계속 채용하거나 퇴직시켜야 하는 상황이 발생한다.

위와 같은 특징 때문에 직원 수와 같은 변수를 모델화할 경우 구성원 개개인에 대한 묘사를 통해서 전체의 특성을 파악하는 방법인 Agent Based Modeling 기법이 많이 사용되고 있다([그림 2] 참고). 기존의 Agent Based Modeling 기법은 모델화 과정이 시스템다이내믹스 기법에 비하여 복잡하여 미리 만들어진 다양한 Tool이 존재하지 않는 한 모델화에 투여 되는 시간이 많은 단점을 가지고 있다.



[그림 2] Agent Based Modeling 환경 (Any Logic)

Vensim DSS는 Subscript 기능이 강한 시스템다이내믹스 소프트웨어이므로 모델화 과정이 복잡한 Agent Based Modeling을 대체하여 사용이 가능하나, 모델화 방식 자체가 다른 시스템다이내믹스 문제와 상이하기 때문에 정확한 개념을 갖지 않으면 혼동을 초래하기 쉽다.

그러므로 이 논문에서는 Vensim DSS 소프트웨어를 활용하여 Agent Based Modeling에 필요한 기본 모듈들에 대해 정의하고 이를 이용한 Agent Based Modeling을 직접 수행해 보아 그 장단점을 분석해 보았다.

II. 방법론

1. Agent Based Modeling 적용 여부 결정

Vensim DSS를 이용한 Agent Based Modeling은 Agent Based Modeling 기법과 시스템다이내믹스 기법의 결합으로 볼 수 있다. 즉, 객체와 관련된 부분은 Agent Based Modeling 기법을 활용하고 나머지 부분은 전통적인 시스템다이내믹스 기법을 활용하는 것이다. 그러나 비록 객체로 이루어진 것이라 하더라도 고객 수와 같이 객체 수가 많아서 각 객체의 개별적 특성보다는 그 평균값으로 시스템을 충분히 묘사할 경우, Agent로 묘사하기 보다는 Aggregated 된 변수로서 (예; 잠재 고객 수, 고객 수 등) 묘사하는 것이 좋다.

우선, 어떤 부분에 Agent Based Modeling 기법을 적용해야 하는가에 대해서 간단히 정리할 필요가 있다. Agent Based Modeling 기법은 1) 정확성이 필요한 경우, 2) 그룹화 (Aggregated) 된 그룹의 숫자가 적은 것이 존재할 때, 3) 존재하는 자료가 객체특성으로 되어 있을 때의 세 가지 경우 적용하는 것이 바람직하다.

1) 정확성이 필요한 경우

조직에 대해 모델화 할 경우 조직원의 수가 4자리 이하(10000 이하)가 된다면 일단 Agent Based Modeling 기법을 고려해 볼 필요가 있다. 나아가 조직원의 그룹화가 요구되고 최소 단위의 그룹이 두 자리까지 된다면 Agent Based Modeling 기법이 효과적인 방법일 수 있다.

2) 그룹화 (Aggregated) 된 그룹의 숫자가 적은 것이 존재할 때

Agent Based Modeling 기법을 시스템다이내믹스 기법으로 구성할 경우 메모리를 많이 차지하게 된다. 특히, 객체의 수가 변하는 경우 시스템다이내믹스 소프트웨어의 특성상 충분한 객체 수를 결정하여 미리 메모리를 잡아 두어야 한다. 한 객체의 특성을 Level 변수로 묘사하려면 초기값과 Rate 변수까지 최소한 4개의 변수(초기값, Level 변수 및 두 개의 Rate 변수)가 필요하므로 객체 특성 상 '4x최대 객체 수'만큼의 변수 수가 필요하다. 변수의 수가 많으면 그에 비례해서 시뮬레이션 기간도 길어진다. 따라서 Agent Modeling 기법을 시스템다이내믹스 소프트웨어로 묘사할 경우 메모리 문제를 반드시 고려하여야 하며, 너무 많은 객체 수가 필요할 경우는 Agent Based Modeling 기법이 불편하게 된다.

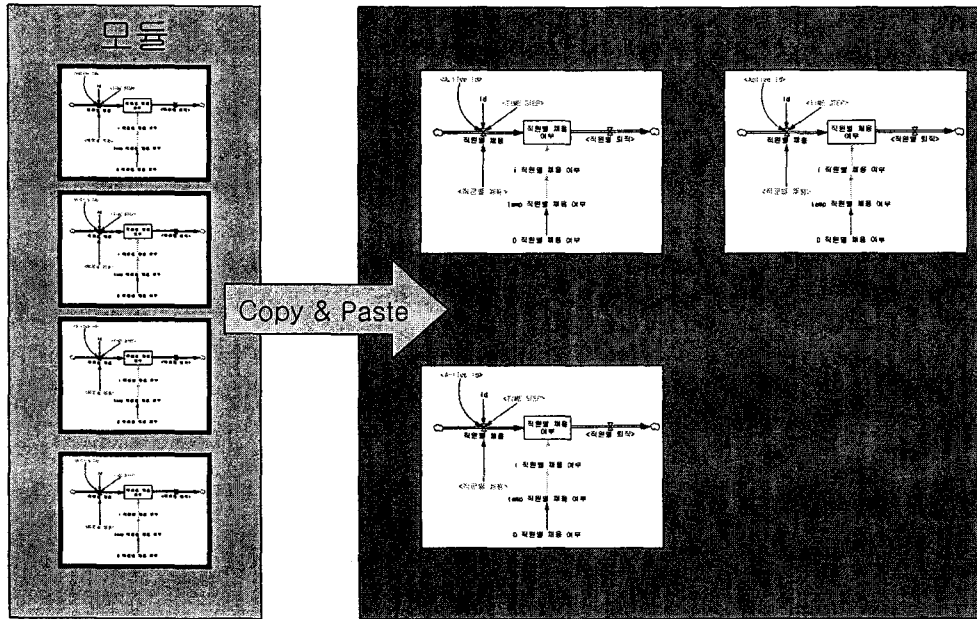
3) 존재하는 자료가 객체특성으로 되어 있을 때

기존에 존재하는 자료가 평균값 등 모집단의 Parameters로 되어 있지 않고 객체 특성으로 되어 있다면 Agent Based Modeling 기법을 적극적으로 검토해 볼 필요가 있다. 이렇게 하는 데에는 두 가지 이유가 있는데, 하나는 객체 특성을, 특히 시간의 함수로 되어 있을 경우, 모집단의 Parameters로 바꾸는 과정이 더 필요하고, 그 작업이 단순할지는 몰라도 모델 개발자가 아닌 사용자에게는 꽤 많은 노력이 필요하기 때문이다. 두 번째 이유는 객체의 특성으로 묘사되어 있다는 사실이 객체의 수가 Agent Based Modeling 기법에 적절한 크기일 가능성이 많기 때문이다.

2. 모듈을 이용한 모델링

Agent Based Modeling 기법과 시스템다이내믹스 기법의 결합은 모델화 과정에서의 사고의 혼선을 가져오기 쉽다. 두 가지 방법이 아주 상이한 사고를 필요로 하기 때문이다. 가장 간단한 예가 Level 변수의 사용인데, 시스템다이내믹스 기법에서는 객체 수 자체가 하나의 Level 변수인데 반하여, Agent Modeling 기법에서는 객체의 상태(예; 객체의 고용 상태, 객체의 직급 등)를 Level 변수로 취급해야 할 경우가 많다.

이러한 혼선을 줄이기 위해서는 [그림 3]과 같이 필요한 모듈을 미리 만들어 두고 이를 복사하여 활용하는 개념을 도입하는 것이 편리하다. 필요한 모듈은 문제의 특성 및 객체의 특성에 따라 달라질 것이다. 여기에서는 조직 문제에 대해 국한하여 필요한 모듈을 정의하기로 한다.



[그림 3] 모듈을 활용한 모델화 방법

3. 어휘 정의

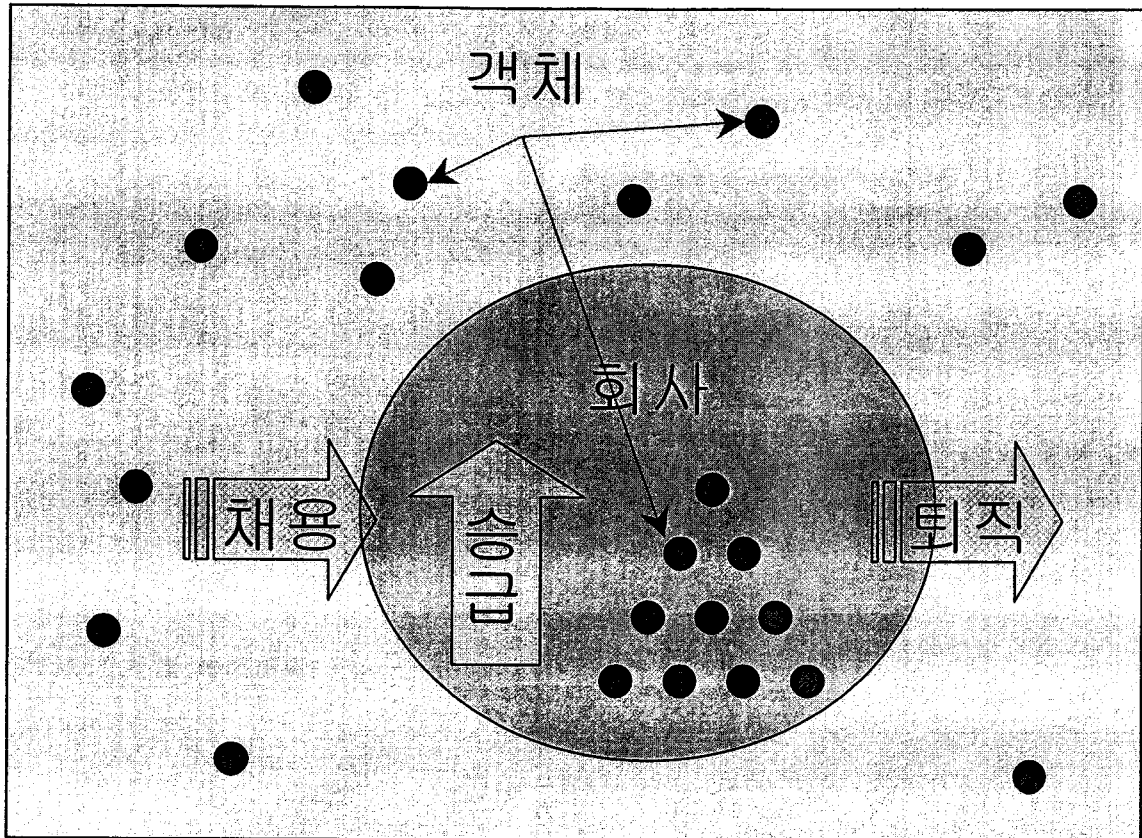
논리를 더 전개하기 전에 편의를 위해 다음과 같은 용어를 정의해 둔다.

- 객체(Objects 혹은 Agents): 시스템다이나믹스 모델 중 Agent Based Modeling 개념을 도입하는 대상
- 객체의 속성 혹은 속성(Attributes): 모듈(Modules): 한 개의 객체 속성을 묘사하기 위한 Agent Based Modeling 기법을 활용 모델 부분
- 객체 그룹: 직원, 비 직원 등 객체의 하부 그룹
- 객체 아이디: 객체를 구분하는 일렬 번호

4. 가정된 문제

여기에서는 조직의 문제로서 새로 형성되기 시작하는 시장에 초기 진출하는 회사로서 향후 조직의 규모, 조직의 구조 등에 관심을 갖고 있다고 가정한다.

이러한 문제에서 조직에 대해서는 Agent Based Modeling을 적용하고, 시장의 크기나 생산 규모 등은 전통적인 시스템다이나믹스 기법을 적용할 수 있다. 조직에서 객체의 역할을 하는 것은 조직의 구성원으로 직원(채용되지 않은 사람도 있으므로 엄밀한 의미에서는 사람)이 된다([그림 4] 참고).



[그림 4] 가정된 문제의 객체 및 조직

시장이 변하면서 생산 규모를 변화시켜야 하고 따라서 직원의 수도 변화한다. 직원이 입사를 한 후 일정 시간이 지나면 승급을 시켜주어야 하며 이 과정에서 승급을 판단할 수 있는 변수, 승급 심사 기준이 필요하다. 여기에서는 문제를 간단히 하기 위해 승급 심사의 기준을 입사 후 경력 혹은 해당 직급에서의 근무기간만 고려하는 것으로 가정한다. 퇴직에는 정년퇴직이 있으므로 객체의 연령도 고려하여야 한다. 이러한 가정 하에서 필요한 객체의 속성은 다음과 같다.

- 직급 (직원별 직급)
- 현 직급에서의 근무 기간 (직원별 현 직급 근무 기간)
- 직군 (직원별 직군)

이러한 객체의 속성들에 대한 시스템다이나믹스 모델 부분을 모듈이라고 정의하고 각각 어떻게 묘사하는 지 설명하고자 한다.

5. 객체 속성 모듈에 대한 묘사(조직 모델의 경우)

1) Subscript 변수 및 객체 아이디

Vensim DSS는 Subscript 변수를 활용하여 객체를 표현하여야 한다.

$$Object := (O_1 - O_{5000})$$

이 식에 따르면 5000 개까지의 객체, 즉 5000 명의 직원 수까지 다룰 수 있다.

규모가 어느 정도 이상의 회사는 사원을 관리하기 위해 사변을 도입한다. Agent Based Modeling에서 객체의 그룹이 변하는 경우 사변과 같은 개념의 도입이 반드시 필요하다. 우리가 다루는 문제의 경우 최소한 두 개의 객체 그룹이 필요하고 채용 및 퇴직 등으로 인하여 객체의 수가 변한다. 이 변수를 객체 아이디라고 정의하기로 한다.

$$Id[Object] = 1, 2, 3, \dots, 5000$$

이 식은 약간은 둔하게 보이나 Vensim DSS가 Agent Based Modeling을 위한 프로그램이 아니기 때문에 어쩔 수 없이 직접 만들어야 하며 엑셀의 복사 기능을 활용하면 손쉽게 만들 수 있다.

2) 입력 자료 파일

이 회사의 인력 관리 파일은 다음 [표 1]과 같은 형태로 되어 있을 것이다. 이 정보를 [표 2]와 같은 Vensim DSS 입력 자료 파일로 바꾸면 되는데, 이 작업은 Excel에서 쉽게 할 수 있다.

[표 1] 입력 자료 파일

사번	직군	연령	직급	채용여부	현직급 근무기간(년)
1	1	20	1	1	1
2	1	21	1	1	1
3	1	22	1	1	1
4	1	23	1	1	1
5	1	24	1	1	1
6	1	25	2	1	1
7	1	26	2	1	1
8	1	27	2	1	1

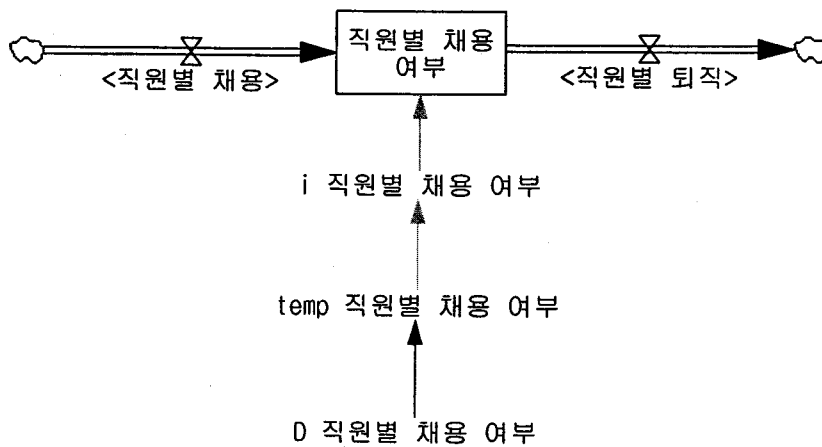
[표 2] Vensim DSS 입력 자료 파일

Time	0
D 직원별 직군[OJ1]	1
D 직원별 직군[OJ2]	1
D 직원별 직군[OJ3]	1
D 직원별 직군[OJ4]	1
D 직원별 직군[OJ5]	1
D 직원별 직군[OJ6]	1
D 직원별 직군[OJ7]	1
D 직원별 직군[OJ8]	1
D 직원별 직군[OJ9]	1
D 직원별 직군[OJ10]	1
D 직원별 직군[OJ11]	1
D 직원별 직군[OJ12]	1
D 직원별 직군[OJ13]	1
D 직원별 직군[OJ14]	1
D 직원별 직군[OJ15]	1
D 직원별 직군[OJ16]	1

3) 채용 여부

채용 여부는 각 객체가 회사에 채용되어 있는가를 나타내는 변수이다. 이처럼 상태를 나타내는 변수는 [그림 5]와 같이 레벨 변수로 모델화하는 것이 편리하다. 채용 여부 변수의 값은 0 혹은 1의 값을 갖게 되며 다음과 같이 정의 된다.

채용 여부= 1, 채용되었을 경우
0, 채용되지 않았을 경우



[그림 5] 채용 여부

D 직원별 채용 여부는 DATA 변수(외부 자료로부터 직접 읽어 오는 변수)이고, temp 직원별 채용 여부는 자료가 존재할 경우는 1을 존재하지 않을 경우는 0의 값을 갖는 변수로서 외부 자료에서 자료가 없을 경우 처리하는 방법을 결정해 준다. 이 채용 여부라는 변수는 Agent Based Modeling 부분과 전통적인 시스템다이내믹

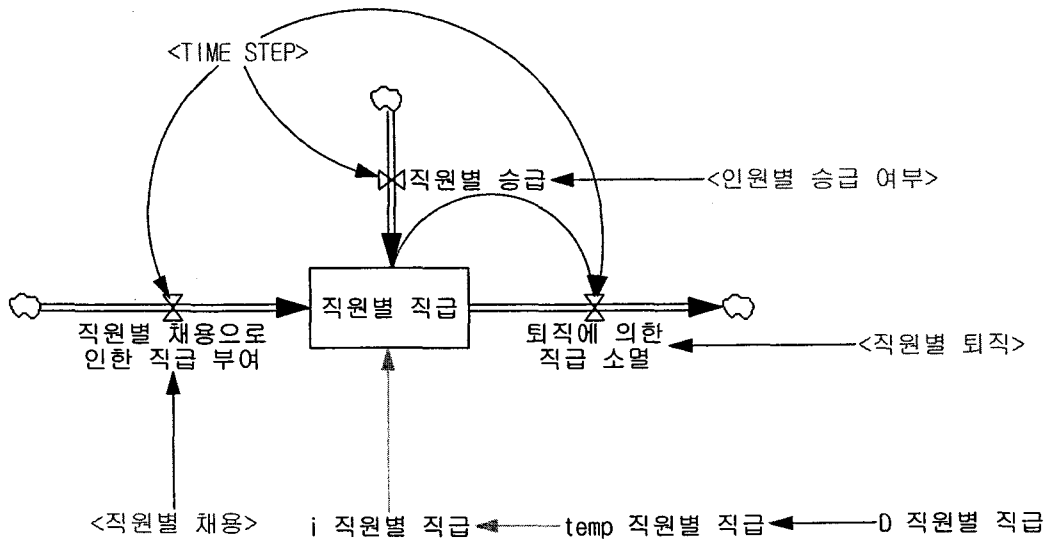
스 모델의 연결에 가장 많이 사용되는 변수이다.

직원별 채용과 직원별 퇴직에 대한 설명은 뒤로 미루겠다.

4) 직급

직급도 상태를 나타내는 변수로서 [그림 6]과 같이 레벨 변수로 취급하는 것이 편리하다. 직급은 1직급부터 7 직급까지 있는 것으로 가정하면 직급을 나타내는 Subscript 변수는 다음과 같다.

$$x \text{ 직급} := (\text{직급1-직급7})$$



[그림 6] 직급

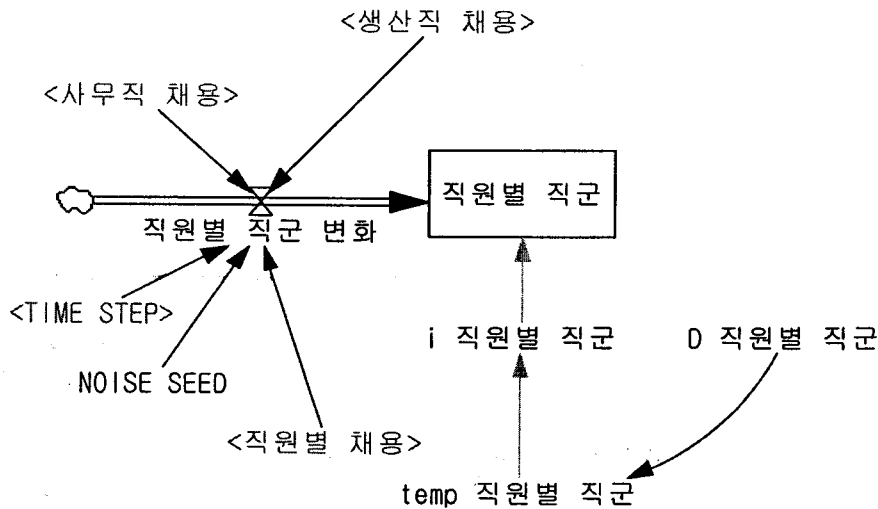
문제를 간단히 하기 위해 경력 사원의 채용은 없는 것으로, 승급은 1 직급만 승진하고, 역 승급(직급이 떨어지는 것)은 없는 것으로 가정한다. 위의 Stock Flow Diagram은 이상의 가정에 따른 것으로 변수 직원별 직급은 다음 식으로 정의 된다.

- 직원별 직급=0, 미채용 0,
- 1, 직급 1
- 2, 직급 2
- ...
- 7, 직급 7

5) 직군

직원의 직군도 상태를 나타내므로 레벨 변수로 모델화 하는 것이 편리하다. 직군은 채용과 함께 결정되기도 하지만 근무 중에 변경도 가능할 것이다. 그러나 여기에서는 채용 시 결정된 직군이 그대로 유지된다고 가정하였다([그림7] 참조). 직군은 Subscript 변수로 나타내어 추후에 조정이 가능하도록 하였으며 여기에서는 단순히 다음과 같이 두 개의 직군만 있다고 가정하였다.

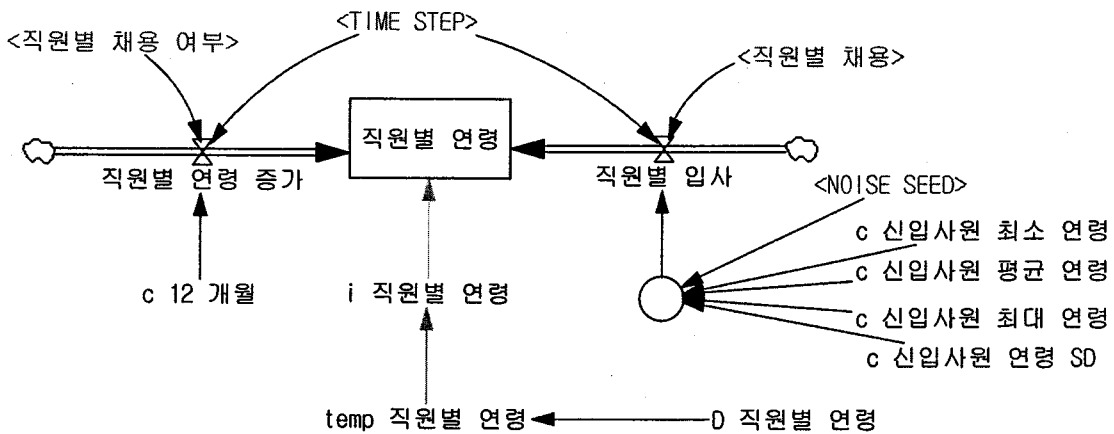
x 직군 := 생산직, 관리직



[그림 7] 직군

6) 연령

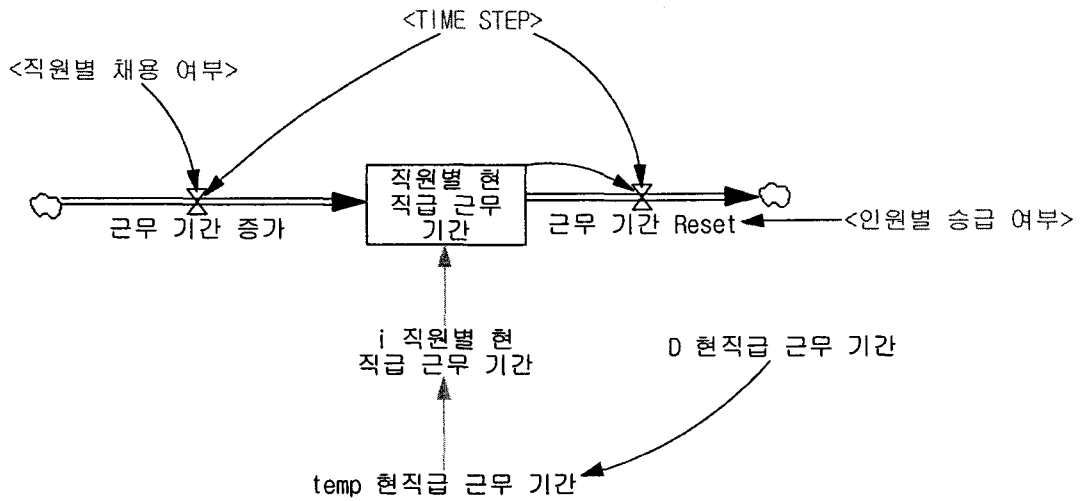
시간이 지남에 따라 변하는 직원의 연령은 [그림 8]과 같이 레벨 변수로 간단히 묘사할 수 있다. 직원이 되지 않은 객체에 대한 속성은 묘사하지 않기 때문에 직원의 입사와 함께 연령도 같이 정의해 주어야 하는데, 이에는 연령분포 함수(확률밀도함수)를 이용하여 난수를 발생하여 결정하도록 하고 있다.



[그림 8] 연령

7) 현 직급 경험

인력 구조를 결정하는 주요 변수 중의 하나는 승격이고, 이를 결정하는 요인으로 현 직급에서의 근무기간을 고려하여야 한다. 이를 위해 [그림 9]와 같이 각 직원의 현 직급 근무 기간을 묘사하는 "현 직급 경험" 모듈을 만들었다.



[그림 9] 현 직급 근무 기간

6. 필요한 모듈 요약

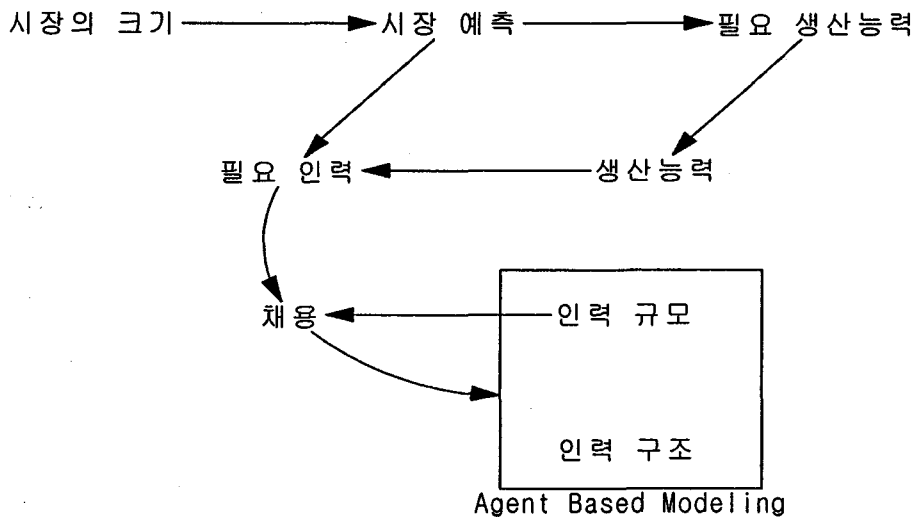
지금까지 묘사한 모듈에 대해 요약 하면 [표 3]과 같다.

[표 3] Agent Based Modeling에 필요한 모듈 (조직의 경우)

모듈 이름	내용	종류	입력 변수
아이디	구성원의 아이디	상수	없음
채용 여부		Level	채용, 퇴직
직급		Level	채용, 퇴직, 승급
직군		Level	채용
연령		Level	채용
현직급 경험	현직급의 근무 기간	Level	채용, 승급

III. 예제

앞서 언급한 Agent Based Modeling 개념은 조직에만 적용하고 나머지 부분은 모두 전통적인 시스템다이내믹스 기법을 적용하기로 한다. 다음의 Causal Loop Diagram ([그림 10])과 같이 두 부분의 만남은 채용에서 이루어진다.



[그림 10] 시장 예측 및 채용

Agent Based Modeling 부분과 다른 부분과의 연결은 다음 식과 같이 주로 직원별 채용 여부[x Object!]의 변수를 활용하게 된다.

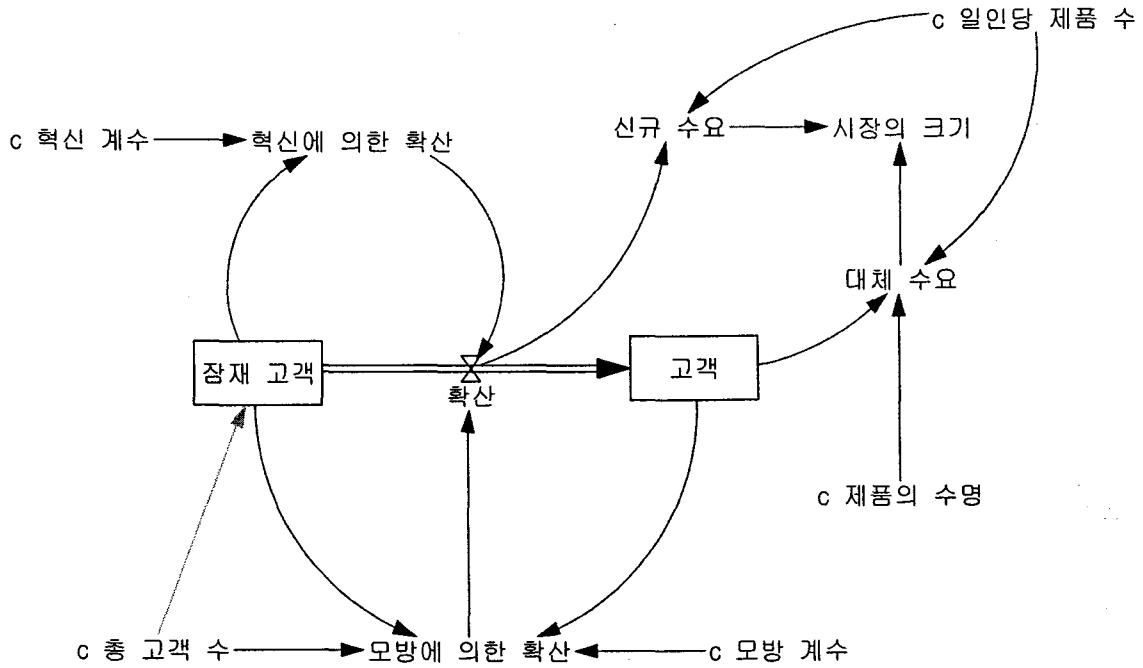
$$\text{생산직 인력} = \text{SUM}(\text{IF THEN ELSE}(\text{직원별 직군}[x \text{ Object!}] = 1, 1, 0) \cdot \text{직원별 채용 여부}[x \text{ Object!}]) \cdot c \text{ 명}$$

위 식에서 변수 c 명은 단위 변환을 위한 것이고, 직원별 채용 여부의 값은 직원일 경우는 1, 그렇지 않을 경우는 0의 값을 갖는다.

1. Stock Flow Diagram

1) 시장 및 시설

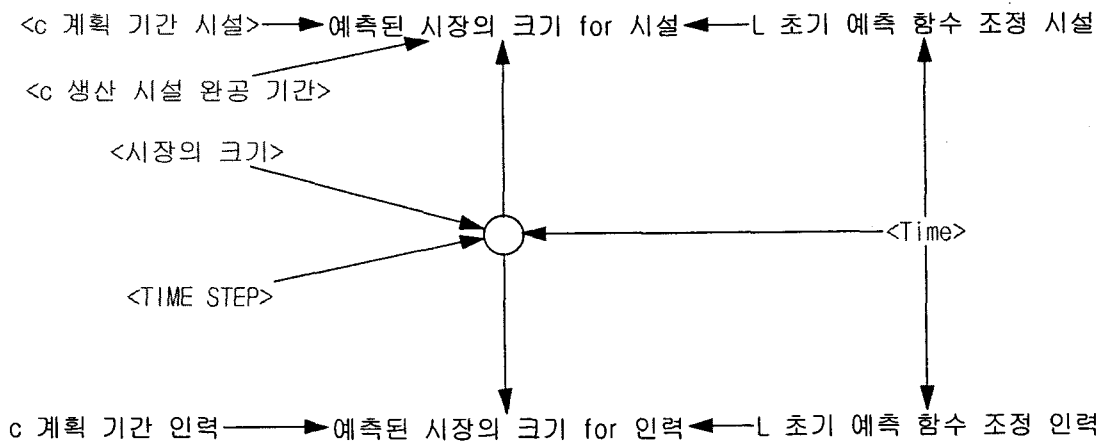
본 연구에서는 다음 그림과 대표적인 시장의 확산 모형으로 시장을 모델화 하였다([그림 11]).



[그림 11] 시장 예측 및 채용

2) 시장의 예측

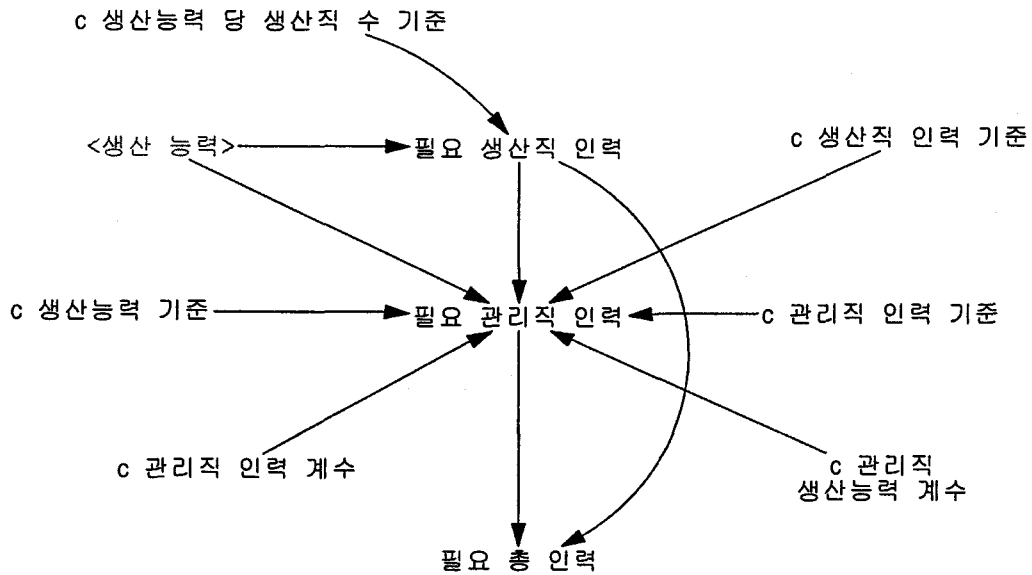
시설의 확장에 걸리는 자연시간과 인력의 확장에 필요한 자연시간이 서로 다르므로 시장의 예측도 [그림 12]와 같이 두 가지로 예측해야 한다.



[그림 12] 시장 예측 및 채용

3) 필요 인력 평가

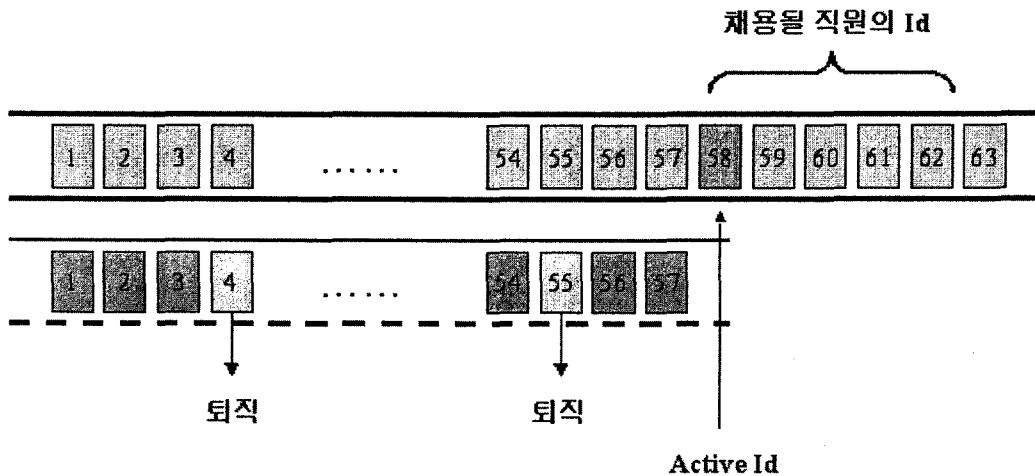
생산적 인력은 생산 시설의 함수가 될 것이며, 관리적 인력은 생산시설 및 생산적 인력의 함수가 될 것이다. ([그림 13] 참고)



[그림 13] 필요 인력 산정

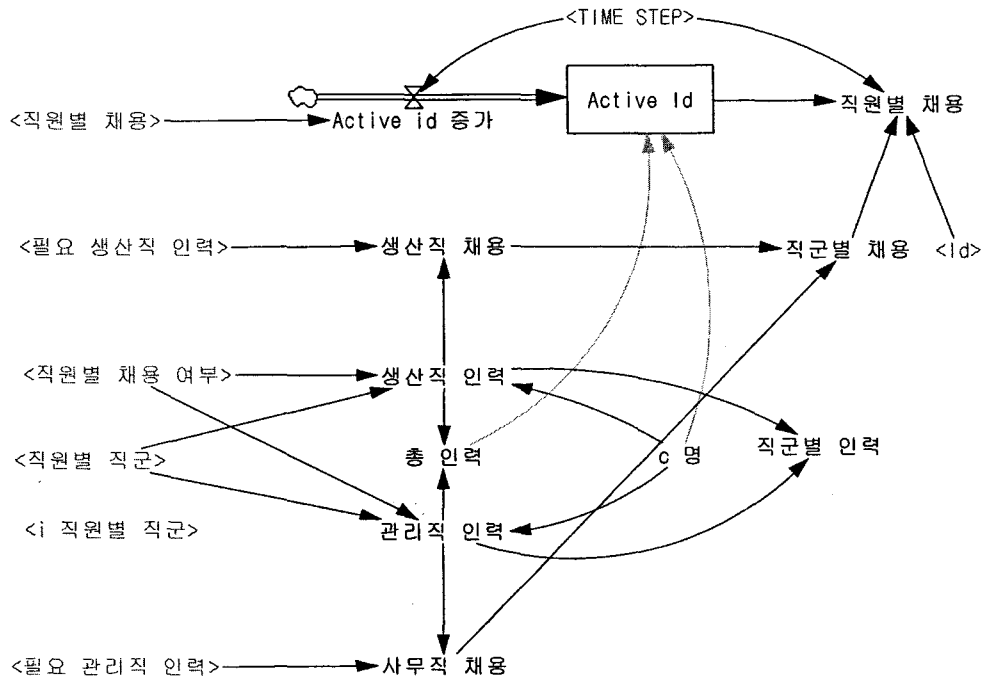
4) 채용

채용을 위해서는 Active Id라는 레벨 변수가 추가로 필요하다. 레벨 변수 Active Id는 새로 채용해야 할 직원의 사번 중 가장 작은 값이다. 다음 [그림 14]에서처럼 현재 57번까지의 사번을 부여하고 있고 앞으로 5명을 채용해야 한다면, Active Id = 58가 된다.



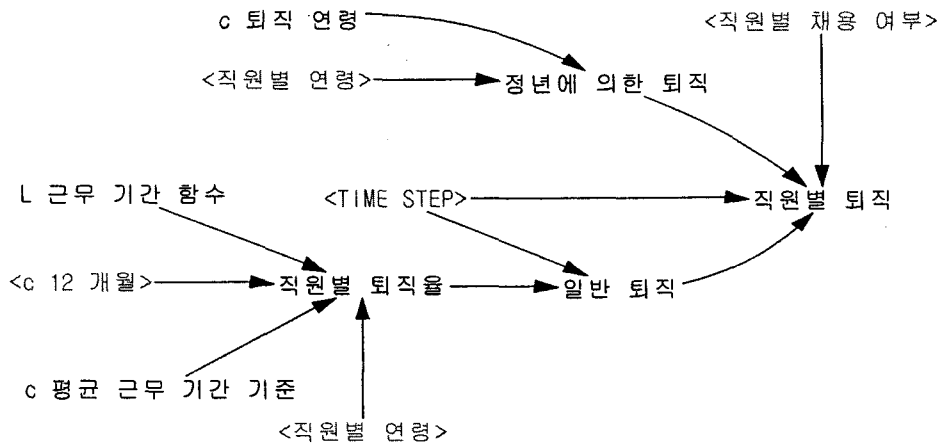
[그림 14] Active Id

다음의 Stock Flow Diagram은 이러한 개념을 활용하여 채용을 하고 새로 채용된 직원에 Id를 부여하는 과정을 묘사한 것이다.([그림 15])



[그림 15] 채용 및 Id 부여

5) 직원 수 및 퇴직



[그림 16] 퇴직

직원의 퇴직은 다음 그림과 같이 정년에 의한 퇴직과 자발적인 퇴직(일반 퇴직)이 있다. 정년에 의한 퇴직은 직원의 연령이 퇴직 연령에 도달하면 자동으로 결정되고 일반 퇴직은 퇴직 율에 따라 난수를 만들어 결정하도록 하였다.([그림 16])

2. 시물레이션 결과

모델의 작동이 제대로 되는가를 확인하기 위해 시물레이션을 수행하고 다음과 같

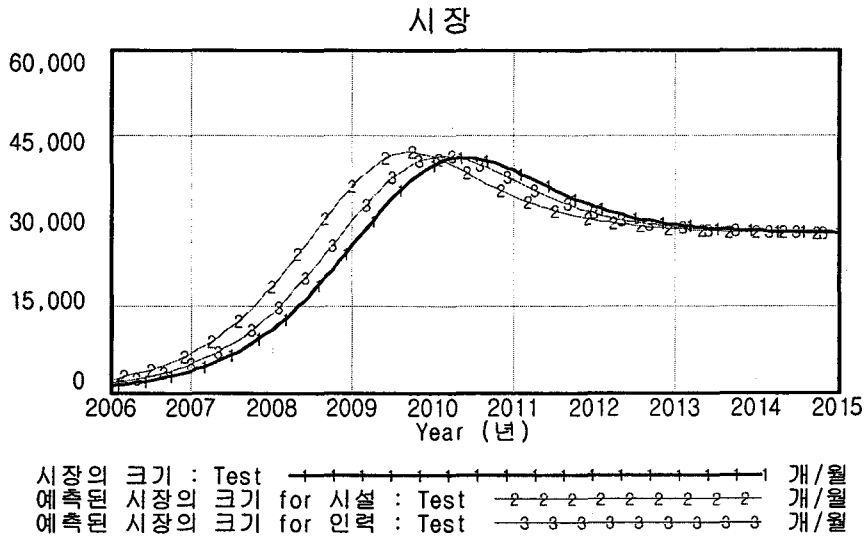
[2006년 춘계 학술대회]

은 변수들에 대해 검토해 보았다.

- 시장: 시장의 크기, 예측된 시장의 크기 for 시설, 예측된 시장의 크기 for 인력
- 시설: 생산 능력, 필요 생산 능력
- 인력 규모: 생산직 인력, 필요 생산직 인력, 관리직 인력, 필요 관리직 인력
- 인력 구조: 직군별 직급별 인원, 지군별 직급별 적정 인원

1) 시장

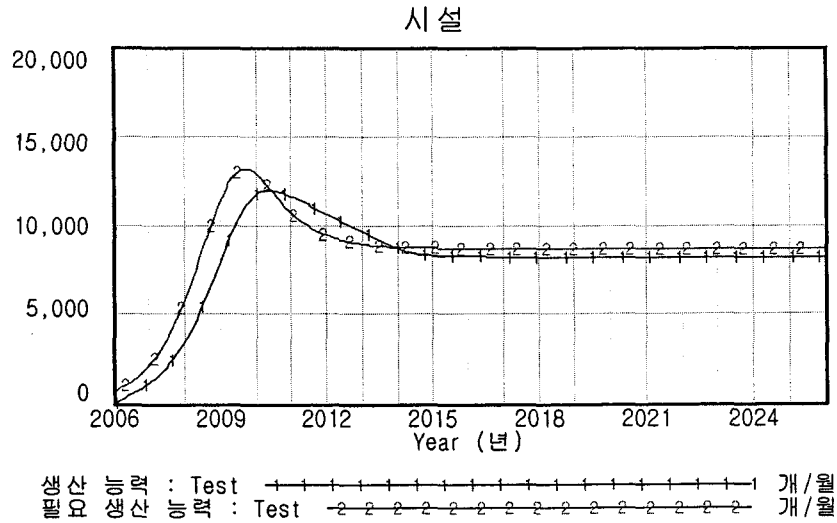
[그림 17]은 시장에 대한 시뮬레이션 결과이다. 굵은 선(1)은 실제 시장의 규모를 의미하고, 앞의 두 선은 그것을 미리 예측한 선이다. 그림에서 보듯이 두 가지 예측 변수 모두 적절한 수준 내에서 예측하고 있다고 볼 수 있다.



[그림 17] 시뮬레이션 결과 (시장)

2) 시설

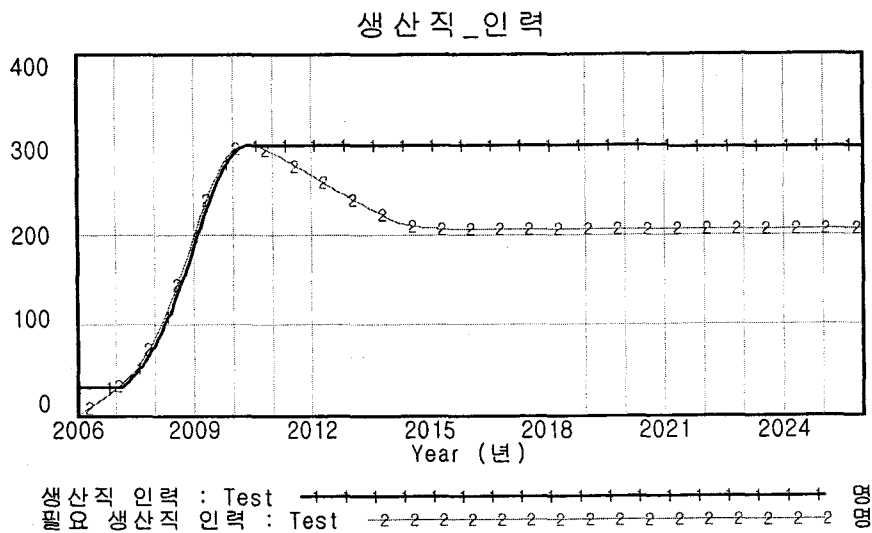
[그림 18]은 시설에 대한 시뮬레이션 결과이다. 2010년 ~ 2015년 사이에는 생산 시설의 증설은 없고 자연 감소분에 의해 생산시설이 조절되며, 그 이후는 노후화 되는 양 만큼 생산시설을 추가하게 된다.



[그림 18] 시뮬레이션 결과 (시설)

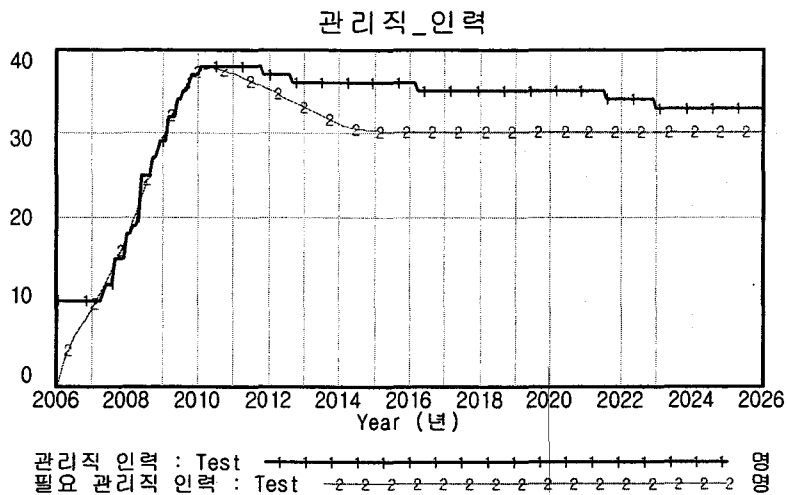
3) 인력 규모

[그림 19]는 생산직 인력에 대한 시뮬레이션 결과이다. 그림에서 보는 바와 같이 성장기인 2010년까지는 필요 인력과 실제 인력 간의 차이가 없으나 그 이후 감소기 및 안정기에는 실제 인력규모가 필요 인력 규모보다 많게 되는 부작용을 일으키게 된다. 이는 인력규모 감소가 자연 감소(정년퇴직과 자발적 퇴직)만을 고려하였기 때문인데, 이 문제를 어떻게 해결할 것인가는 이 논문의 영역이 아니므로 생략하기로 한다.



[그림 19] 시뮬레이션 결과 (생산직 인력)

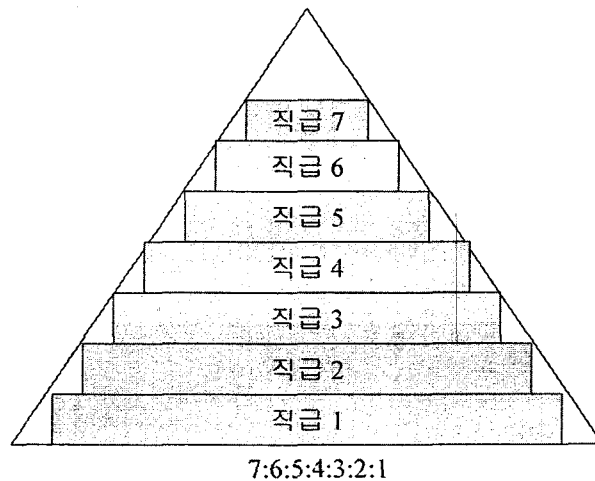
[그림 20]은 관리직 인력에 대한 시뮬레이션 결과로 생산직 인력과 마찬가지로 시장 감소기 및 안정기 초기에 문제가 발생하는 것을 알 수 있다.



[그림 20] 시뮬레이션 결과 (관리직 인력)

4) 인력 구조

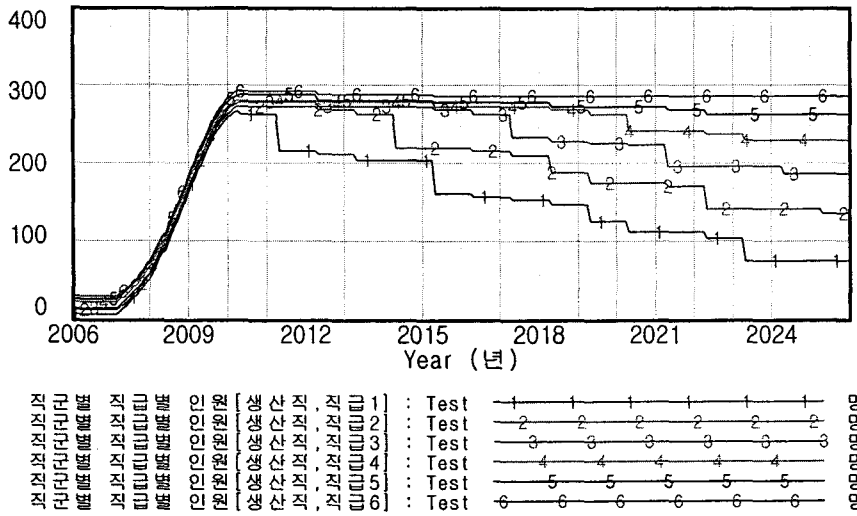
적정 인력 구조로 [그림 21] 같은 피라미트 구조를 가정하였다(직급1 : 직급2 : 직급3 : 직급4 : 직급5 : 직급6 : 직급7 = 7 : 6 : 5 : 4 : 3 : 2 : 1).



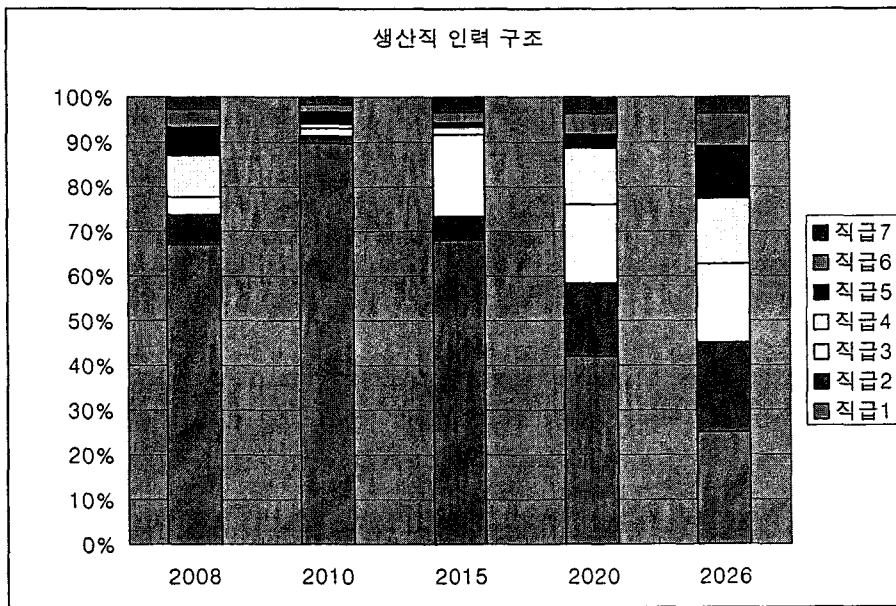
[그림 21] 가정된 적정 인력 구조

다음 그림들은 생산직 인력 구조([그림 22] 및 [그림 23]) 및 관리직 인력 구조([그림 24] 및 [그림 25])에 대한 시뮬레이션 결과로서 인력 규모와는 달리 성장기에 인력 구조상 문제점이 더 많이 발생하는 것을 알 수 있다. 이는 모든 채용이 1 직급에서만 이루어지고 승급을 일정 비율로 하도록 되어 있기 때문인데, 이 문제를 어떻게 해결할 것인가는 이 논문의 영역이 아니므로 생략하기로 한다.

생산직 직급별_인원

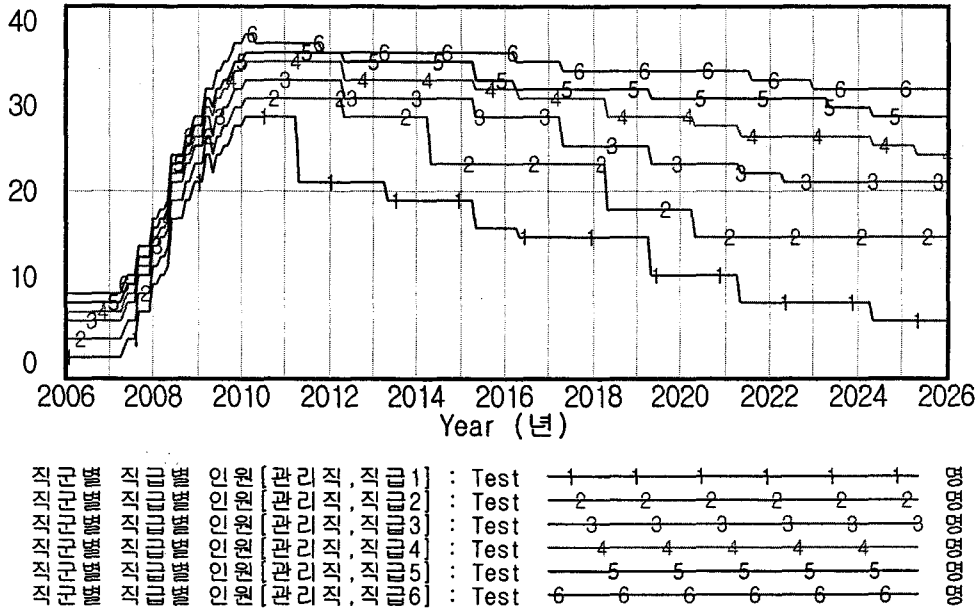


[그림 22] 시뮬레이션 결과 (생산직 인력 구조 1)

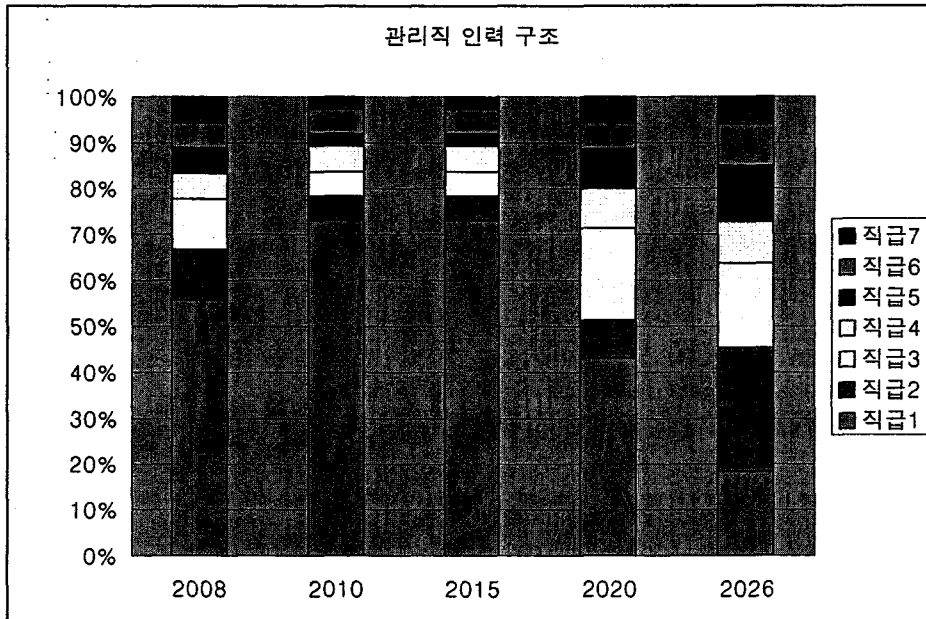


[그림 23] 시뮬레이션 결과 (생산직 인력 구조 2)

관리직 직급별_인원



[그림 24] 시뮬레이션 결과 (관리직 인력 구조 1)



[그림 25] 시뮬레이션 결과 (관리직 인력 구조 2)

IV. 결론

지금까지 Agent based Modeling 기법을 시스템다이내믹스 모델에 어떻게 활용하는가를 알아보기 위해, 상품을 만들어 파는 회사의 인력 규모 및 인력 구조를 예측의 목적으로 하는 시스템다이내믹스 모델을 개발하고 시뮬레이션을 수행하여 보았다.

조직 관련 부분은 Agent Based Modeling 기법을 시장, 시설을 비롯한 다른 부분은 전통적인 시스템다이내믹스 기법을 적용하였는데, 다음과 같은 결론을 얻을 수 있었다.

- 모델이 복잡해지고 시뮬레이션 기간이 길다.
- Agent Based Modeling 기법은 전통적인 시스템다이내믹스 기법의 “연속성”에 대한 가정 사항을 풀었기 때문에 상당히 정확한 예측을 할 수 있다.
- Agent Based Modeling 기법에서는 레벨 변수의 선택 등에서 전통적인 모델링 개념과 차이가 많아 혼선을 일으키기 쉬우므로 전체 모델에 대한 개념설계 시 Agent Based Modeling 기법의 활용 부분을 정확히 정의하고 그 연결 부분에 대한 세밀한 설계가 필요하다.

Agent Based Modeling 기법이 전통적인 기법과 개념상 차이가 많기 때문에 다음과 같은 경우에만 한정하여 활용하는 것이 바람직하다.

- 정확성이 필한 경우
- 전통적 방법상에서 소수점 이하의 객체 수 때문에 발생하는 문제점을 피해야 하는 경우
- 입력자료의 형태가 객체 지향적으로 묘사되어 있고 전통적인 방법에 따르면 그 처리가 복잡해 질 경우

-MEMO-