

생명체 패러다임

고성범, 원일용

천안공업대학 컴퓨터공학과, 건국대학교 컴퓨터공학과,

Life Paradigm

Sung-Bum Ko, Il-Young Won,

Department of Computer Science, Chonan National Technical College

Department of Computer Science, Konkuk University

요약

미래의 시스템은 보다 동적이고 복잡한 환경에서 작동될 것으로 예측된다. 이러한 환경에서는 학습, 적응, 진화, 퍼지, 추론, 계획, 보안, 자기 조직화, 감성 등 소위 지능적 능력들이 필수적으로 요청된다. 본 논문에서는 생명체 패러다임 SAL(System As a Life) 을 제안한다. SAL 은 생명체 고유의 창발적 속성에 기반을 둔 시스템 설계 방법론으로 객체 패러다임을 확장한 구조를 갖는다. SAL 기반으로 시스템을 설계할 경우 상기의 지능적 능력들이 자연스럽게 구현될 수 있다.

1 장. 서론

최근 정보화 시대로의 본격적인 진입에 따라 주변 환경은 점점 더 동적이고 복잡한 속성을 띄게 되었다. 이에 따라 좀더 지능적인 능력을 갖춘 시스템에 대한 요구가 증대하고 있다. 이러한 능력 중에는 학습, 적응, 진화, 퍼지, 추론, 계획, 보안, 자기 조직화, 감성 등이 포함된다. 이들에 대한 기존의 접근 방법은 각각을 독립적으로 구현한 다음 결합하는 것이다[2]. 이런 환원론적 패러다임은 유연성과 효율성의 측면에서 많은 문제를 야기할 수 있으며 특히 시스템의 복잡성이 일정 한계를 넘을 경우 구현상의 어려움이 기하급수적으로 증가하게 된다. 그 이유는 누적된 예러가 또 다른 예러를 부르는 악순환을 야기하기 때문이다. 이것을 우리는 '환원론적 패러다임에서의 복잡성 문제'라고 부른다[3][4]. 생명체 패러다임은 이런 복잡성 문제를 해결하는 한 가지 대안이 될 수 있다. 생명체 고유의 창발적 속성에 기반한 생명체 패러다임에서는 시스템의 복잡성이 증가할수록 오히려 예러가 줄어드는 경향을 갖기 때문이다. 우리는 본 논문에서 기존의 객체 모델을 확장한 SAL(System As a Life) 모델을 제안한다. SAL 은 창발 이론에 기초하고 있으며 많은 생명적 속

성들을 창발적으로 발현시킨다. 이런 창발적 속성 때문에 SAL 을 하나의 설계 패러다임으로 이용할 경우 상기의 지능적 능력을 하나의 모델 안에서 쉽게 그리고 통합적으로 구현할 수 있다. 우리는 본 논문에서 이점을 증명해 보이려고 한다.

2 장. SAL(System As a Life) 모델

SAL 모델은 크게 보아서 TM 모델, PM 모델, Agent 모델 등 세 가지 부 모델로 구성된다. 여기서 TM 은 Cell 위에 구축되는 데, Cell 은 일반적인 의미의 테스크 처리기(Task Processor)로서 기존의 객체 모델로 구현된다. 즉, SAL 은 기존의 객체 모델 위에 세 단계의 상위 계층이 추가된 구조를 하고 있다.

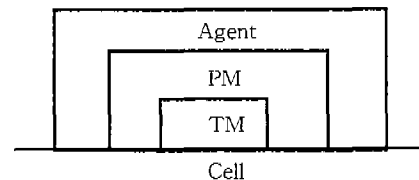


그림 2.1 SAL 모델의 구조

2.1 PM (Potential Module) 모델

SAL 모델에서 생명체의 세포에 해당하는 기본 구조가 PM 이다. 즉, SAL 기반 시스템은 PM 이라는 세포로 구성된 가상 생명체로 정의될 수 있다.

1. PM 의 구조[2]

PM 은 테스크 처리를 담당하는 TM 과 PM 간의 포텐셜 교환을 담당하는 PI(Potential Interface) 로 구성되는 데, 이들은 각각 생명체의 세포핵과 세포막에 대응한다. PM 은 정보 처리의 관점에서 아날로그적 정보 처리를 담당하는 PI 계층과 디지털적 정보 처리를 담당하는 TM 계층으로 나누어져 있다. 여기서 포

텐셜은 극도로 추상화 된 에너지 개념이며 수학적으로는 아날로그적인 양이다. PM 은 PF, LF, NF, AF 등 네 개의 입력 포트와 CF, OF 등 두 개의 출력 포트 등 함께 여섯 개의 Port를 갖는다. 이들 Port 각각에는 입력이나 출력을 아날로그적으로 제약할 수 있는 Filter 가 있고 Filter 는 한 개 이상의 Gate 를 포함할 수 있다. PM 들은 포텐셜을 주고받는 방식으로 추상적 상호 관계를 구현한다. 포텐셜은 동적 포텐셜과 정적 포텐셜로 나눌 수 있으며 또 내부 포텐셜과 외부 포텐셜로 구분할 수도 있다. 여기서 내부 포텐셜은 정적 포텐셜에 해당하며 외부 포텐셜은 동적 포텐셜에 해당한다. 한편 PI 계층의 DNA 는 감성에 관한 선험적 정보가 저장된다.

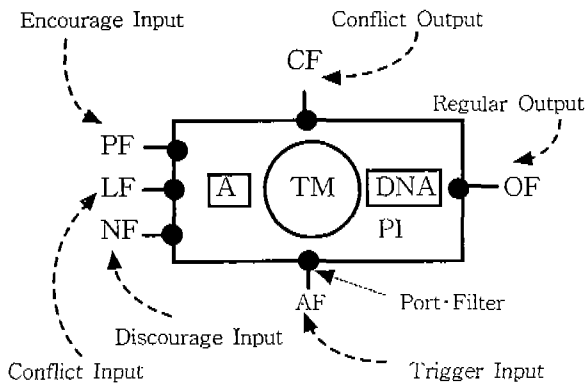


그림 2.2 PM 의 구조

2. PM 포트의 기능

PF 입력은 내부 포텐셜에 대하여 정적(+) 성격을 갖는 포텐셜이며 PIP(Positive Input Potential) 라고 부른다. NF 입력은 내부 포텐셜에 대하여 마이너스적 성격을 갖는 포텐셜이며 NIP(Negative Input Potential) 라고 부른다. PIP 와 NIP 가 결합하여 입력 포텐셜 (Input Potential, IP)을 생성한다. LF 입력은 외부로부터 입력되는 갈등 포텐셜(외부적 갈등 포텐셜)을 의미한다. 원래 갈등 포텐셜은 PM 내부에서 PF 입력과 NF 입력간의 충돌이 일정 한계를 넘게 될 때 생성되며 이를 내부적 갈등 포텐셜이라고 부른다. CF 단자로는 갈등 포텐셜이 출력된다. 갈등 포텐셜은 내부적 갈등 포텐셜과 외부적 갈등 포텐셜 간의 함수이다. 출력 포텐셜은 입력 포텐셜과 내부 포텐셜의 함수로 결정되며 OF 단자로 출력된다. 일반적으로 포텐셜 값은 정규화 된 값(0.0 에서 1.0 사이)을 갖는다. 포텐셜 간의 상호 관계에 있어서 가장 간단한 형태는 다음과 같다.

$$OF = Inner_P + Outer_P$$

$$Inner_P = A$$

$$Outer_P = PIP - NIP$$

$$CF = LF + \text{Min}(PIP, NIP)$$

3. PM 의 Trigger

PM 은 주로 AF 단자에 의해 트리거 된다. 트리거 방식에는 크게 보아 1 단계 트리거 방식과 2 단계 트리거 방식 등 두 가지 유형이 있다. 1 단계 방식에서는 AF 입력 포텐셜이 일정 값(보통 1.0)을 넘게 되면 PI 계층과 TM 계층이 동시에 살아난다(Active 상태). 2 단계 방식에서는 우선 PI 계층이 살아나고 그 다음 단계로 TM 계층이 살아나게 된다. 즉, 2 단계 방식에서는 AF 입력 포텐셜에 관하여 두 개의 임계값이 존재하는 것이다. PI 계층이 살아난다는 것은 포텐셜 교환(아날로그 정보 처리)이 시작된다는 의미이며, TM 계층이 살아난다는 것은 디지털 정보 처리가 시작된다는 의미이다. 입력 포텐셜이 일정 조건을 만족할 때만 PI 계층이 살아나도록 부가적 조건을 첨가할 수 있는데, 이를 조건부 1 단계 트리거 방식이라고 부른다. 이 방식은 특히 하이브리드 추론 방식에서 유용하게 사용된다. 한편 AF 포트만의 트리거 유형을 A 형 트리거 방식이라고 부르고, PF, NF 등이 참여하는 트리거 방식을 B 형 트리거 방식이라고 정의한다.

2.2 TM(Task Module) 모델

TM 은 직접 태스크 처리를 담당하는 모듈로 PM 의 코어 부분을 형성한다. TM 은 PM 의 한 부분으로서 PI 와의 인터페이스를 통해서 수직적인 상호 작용을 수행한다.

1. TM 의 구조

TM 은 제어 계층과 도메인 계층으로 구성된다. 제어 계층은 PM 간의 포텐셜 운행과 관련된 제어 기능(포텐셜의 디지털적 제어 기능)을 수행하며 이를 위해 별도의 지식 시스템을 운영한다. 포텐셜 운행으로 구현되는 PM 간의 상호 관계는 시스템 상태를 결정하는 주요 인자가 된다. 한편 도메인 계층에서는 주어진 태스크를 실제로 처리하는 기능을 담당하며 이를 위한 지식 시스템과 정보 시스템을 운영한다. TM 의 제어 계층에서는 포텐셜이 PI 의 어떤 Gate 를 통과했는지를 인식할 수 있으며, 이 점을 이용하여 Gate 각각에 대하여 특별한 의미를 부여할 수도 있다. TM 의 일 처리 결과는 주기적으로 평가되고 그에 근거하여 보상이 주어진다. 보상의 단위는 포텐셜이며 이것은 내부 포텐셜 형태로 저장된다. PI 계층의 선험적 감성 정보는 PM 의 경험에 의해 수정되는데, 이 정보는 TM 계

층의 KB(Soft-DNA 라고 부른다) 에 저장된다.

2. Cell 모델[1]

실제로 Task를 처리하는 주체는 Cell 이다. 다만 Cell 의 기능은 기본적으로 정적인 특징을 갖는다. 즉, 테스크 처리와 관련된 테스크 도메인, 처리 방법, 처리 성능 등이 특별한 범주 내로 고정되어 있다. TM 은 Cell 들에 대한 포인터를 동적으로 운영하며 좀 더 복잡한 테스크를 처리한다. 다양한 Cell 들이 모여있는 가상적인 영역을 Cell Forest 라고 부른다. 정의에 의해서 Cell Forest 는 성능 및 도메인의 측면에서 모든 가능성을 포함한다. 이를 Cell Forest 의 “Task 처리에 관한 완전성” 이라고 부른다. 우리는 인터넷이 Cell Forest 의 완전성을 지원할 수 있는 모든 조건들을 갖추고 있다고 생각한다.

3. 특성 함수 모델[5]

창발 현상은 상호 관계의 복잡성이 충분히 증가할 때 발생한다. 이를 위해 요소 자체가 충분한 유연성을 갖출 필요가 있다. 이러한 이유로 우리는 특성 함수 개념을 도입하게 되었다. 여기서 특성 함수라 함은 이벤트와 관련된 객체 고유의 속성으로 성능 함수와 관심함수로 나뉜다.

(1)성능 함수(Performance Function)

성능 함수란 테스크에 대한 객체의 성능을 나타내는 함수이다. 테스크 도메인은 이벤트 집합으로 정의되는데, TM 의 이벤트 처리 성능은 0.0 에서 1.0 사이에서 변화한다. 여기서 성능 값 1.0 은 이벤트에 대한 처리를 완벽하게 수행할 수 있는 능력을 의미하고 0.0 은 처리 능력이 전혀 없는 경우를 의미한다.

(2)관심 함수(Interest Function)

TM 은 이벤트에 대해 일정한 관심(혹은 흥미)을 보이게 되는데, 관심의 정도를 나타내는 함수를 관심함수라고 정의한다. 관심 값은 0.0 에서 1.0 사이의 값을 갖는다. 여기서 관심 값 1.0 은 이벤트에 대한 완벽한 관심을 의미하고 0.0 은 문자 그대로 무관심을 의미한다. 관심값은 TM 이 요구하는 자원의 양을 결정하는 기준이 된다.

2.3 Agent 모델

우리는 PM 연결망으로 구성된 시스템을 PM 기반 Agent(PM-Based Agent, PBA)라고 부른다. PBA 는 일반적으로 학계에서 사용되는 용어인 Agent 개념과 기능적 측면에서는 동일한 속성을 공유한다. 즉, PBA

는 PM 개념을 이용하여 특별한 방법으로 구현된 Agent 로 볼 수 있다.

1. Agent 의 구조

Agent 는 크게 보아서 제어 계층과 도메인 계층으로 구성된다. 제어 계층은 시스템 전체를 제어하거나 조정하는 정보 처리 시스템을 포함한다. 한편 도메인 계층은 다수 PM 들간의 연결망으로 정의된다. Agent 의 테스크 도메인은 PM 들의 성능 곡선(정규분포)의 벡터 합으로 정의된다.

2. 상호 관계 가정

창발 현상은 단순한 요소들의 상호 작용을 통해서 일어나는 것이며 이러한 관점에서 “상호 관계” 개념은 대단히 중요한 의미를 갖는다. 우리는 본 논문에서 상호 관계 원형 집합(Correlation Primitive Set, CPS) 이라는 개념을 도입하였다. CPS 는 7 가지의 기본적인 표준적인 상호 관계를 상징한다. 우리는 어떤 유형의 상호 관계도 CPS 위의 함수 관계로 정의될 수 있다고 주장하며 이를 SAL 모델에서의 “상호 관계 가정” 이라고 부른다. 다음은 PM 구조를 이용하여 CPS 를 구현하는 방법을 보여준다.

[1]적대 관계

$$PF \longrightarrow NF \text{ (Val(NF) > 0)}$$

[2]지원 관계

$$PF \longrightarrow PF \text{ (Val(PF) > 0)}$$

[3]협조 관계

$$PF \longleftrightarrow PF \text{ (Val(PF) > 0)}$$

[4]갈등 관계

$$PF \wedge NF \longrightarrow CF \text{ (Val(PF) > 0, Val(NF) > 0)}$$

[5]경쟁 관계

$$NF \longleftrightarrow NF \text{ (Val(NF) > 0)}$$

[6]생성시키거나 소멸시키는 관계

$$PF \longrightarrow AF \text{ (Val(AF) > 0 : 생성 관계, Val(NF) < 0 : 소멸 관계)}$$

[7]갈등을 증가시키거나 감소시키는 관계

$$PF \longrightarrow SF \text{ (Val(SF) > 0 : 증가 관계, Val(NF) < 0 : 감소 관계)}$$

3. 포텐셜 주입(Potential Injection)

Agent 를 구성하는 PM 들의 전체 집합을 S 라 할 때 S 의 부분 집합을 대상으로 특정한 단자 n ($n \in \{LF, PF, NF\}$) 에 대하여 일정한 양의 포텐셜을 주입하는 것을 “포텐셜 주입”이라고 정의한다. 포텐셜 주입의 유형에는 P 형 포텐셜 주입, N 형 포텐셜

주입, L 형 포텐셜 주입등 세 가지가 있다.

4. 엔트로피

우리는 임의의 Agent A 에 대하여 시스템 상태를 나타내는 속성의 하나로 엔트로피 개념을 다음과 같이 정의한다.

- ① 특정한 에이전트 A 에 대하여
- ② 특정한 시점 t에서
- ③ $P_s^*(t) > P_w(t)$ 이고 $P_s^*(t) = P_s(x, t)$ 일 때
- ④ 에이전트 A 는 시점 t 에서 x 로 표현되는 엔트로피 상태를 갖는다고 정의한다.
- ⑤ 이때 x 가 + 값이면 “엔트로피 부족 상태” 이고, - 값이면 “엔트로피 과잉 상태” 이다.

5. BB(Bulletin Board) 공간

Agent 는 일반적으로 솔루션 출력을 위한 BB 공간을 갖고 있다. Agent 는 PM 들로 구성되기 때문에 솔루션 출력의 주체들은 PM 들이다. PM 들은 자신이 원하는 이벤트(특정한 환경 혹은 프로젝트) 에 대하여 선택적으로 반응하게 되며 그 결과로 얻어진 솔루션을 BB 공간에 게시한다. 이때 PM 들은 자신의 솔루션을 자신이 원하는 크기로 게시하게 된다.

7. 충돌 원리

Agent 는 출력을 게시하기 위하여 BB 공간의 자원을 필요로 한다. 이 자원은 상위 계층이나 외부의 소비자로부터 공급된다. BB 공간이 충분치 못할 경우 자원에 대한 요구간에 충돌이 발생한다. 충돌 결과에 대한 가장 자연스러운 해석은 충돌 부분을 아무도 차지할 수 없다는 것이다. 우리는 이것을 BB 공간에서의 “충돌 원리 가설” 이라고 부른다.

제 3 장 지능적 속성의 구현

본 장에서는 2 장에서 정의한 SAL 모델을 이용하여 감성, 학습, 진화, 추론, 퍼지 개념, 자기 조직화, 보안 등의 지능적 개념들이 어떻게 구현될 수 있는지를 보인다.

3.1 감성

SAL 모델에서 PM 과 Agent 는 각기 고유한 감성 체계를 갖추고 있다. PM 의 감성은 좋고 싫음만을 구별하는 1 차원적 감성 체계인 데 비해, Agent 의 감성은 엔트로피 기반의 다차원적 감성 체계이다.

1. PM 의 감성

PM 은 두 가지 종류의 DNA 시스템을 갖고 있다. 하나는 Hard-DNA 라고 부르며 PI 계층에 존재한다. 다른 하나는 Soft-DNA 라고 부르며 TM 의 제어 계층에 존재한다. 이 중 Hard-DNA 에는 감성(다른 PM 에 대한 좋고, 싫음)에 대한 선형적인 정보가 각인 되어 있으며 Soft-DNA 에는 나중에 경험을 통해 새로 얻어진 감성 정보가 기록된다. PM 상호간의 포텐셜 교환은 DNA 정보에 기반한다. 즉, PM 은 좋아하는 상대를 부추키거나 지원한다. 반대로 싫어하는 상대를 공격하거나 위축시킨다. 누가 누구를 좋아하는가는 PM 이라는 생명체의 창조주(예를 들어 프로그래머)가 새겨 넣는 것이며, 그 이후로는 유전자 형태로 진화의 길을 밝게 된다.

2. Agent 의 감성[2]

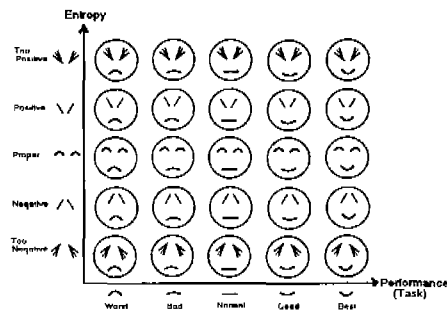


그림 3.1 감성 공간의 예

감성 개념의 근거는 자율 개념이다. 자율이 실수를 낳고, 실수가 후회를 낳고, 후회가 불쾌감을 생성한다. 그리고 불쾌감의 외연적 표현이 감성인 것이다. PM 은 자율성을 갖기 때문에 PM 연결망인 Agent 는 감성을 가질 수 있게 된다. 본 논문에서 감성은 불만(Complain)에 대한 외연적 표현으로 정의된다. 여기서 불만은 외부적 요인에 의한 불만과 내부적 요인에 의한 불만으로 나눌 수 있다. 내부적 불만과 외부적 불만의 구체적인 정의는 Task 처리와 관련해서 결정되어야 한다. 우리가 제안하는 한 가지 방법은 다음과 같다. 첫째로 내부적 불만은 주어진 조건에서 최선을 다했는지 여부로 정의된다. 둘째로 외부적 불만은 실제 Task 처리 결과가 예측치에 어느 정도 부합하는가로 정의된다. SAL 모델에서 내부적 불만을 재는 한 가지 방법은 엔트로피 개념을 이용하는 것이다. 즉, PM 들이 지나치게 Encouraged 되면 엔트로피 과잉으로 성능이 저하된다. 반대로 PM 들이 지나치게 Discouraged 되면 엔트로피 부족으로 역시 성능이 저

하된다. SAL 모델에서 외부적 불만이 정의되기 위해서는 계획(Planning) 능력이 필수적이다. 왜냐하면 계획을 통해서 예측이 가능해지기 때문이다. 감성적 관점에서 볼 때 모든 시스템은 동일한 목표 즉, 행복을 추구한다. 이것은 감성 개념을 사용할 때 시스템 제어를 보다 간결하게 다룰 수 있다는 뜻이 된다.

3.2 추론

SAL 모델에서 추론 문제를 다루기 위해서는 SAL 기반 정보 시스템이 먼저 정의되어야 한다. SAL 기반 정보 시스템은 TM 에 Fact 나 Rule 을 대입하는 방법으로 간단하게 구현 가능하다. PM 의 추론 메카니즘은 아주 간단한 원리에 바탕을 두고 있다. 즉, PM 은 감성 기반으로 작동하며 투자(입력) 받은 포텐셜을 자신이 좋아하는 상대에게 재투자하는 방식으로 추론 메카니즘을 구현한다. 즉, 전형적인 논리 규칙 “모더스 포넨스”가 감성 기반으로 구현되는 것이다.

1. PM에서의 트리거 메카니즘

PM 의 각 포트에는 Filter 가 있어서 외부 입력 포텐셜을 통제 및 제어하게 된다. 특히 AF-Filter 는 추론의 속성을 결정하는 중요한 역할을 한다. AF-Filter 는 Left-Part(LP), Center-Part(CP), Right-Part(RP) 등 세 부분으로 나뉘어져 있는데, LP 와 RP 로 입력되는 신호는 논리적 트리거 신호라고 부르고 CP 로 입력되는 신호는 감성적 트리거 신호라고 부른다. 논리적 트리거 신호는 다시 전진 추론 신호와 후진 추론 신호를 나뉘는 데, 전자는 LP 로 입력되고 후자는 RP 로 입력된다. 모든 Filter 는 0 과 1 사이의 실수 값으로 입력 포텐셜을 정규화 한다. 논리적 트리거는 PF 포트와 AF 포트 양쪽에 대한 포텐셜 입력으로 정의된다. 이에 비해 감성적 트리거는 AF 포트에 대한 포텐셜 입력으로 정의된다. 논리적으로 트리거 된 PM 은 외부로부터 투자 포텐셜을 받으므로 자신의 포텐셜을 별도로 투자할 필요가 없지만 감성적으로 트리거 된 PM 은 자신의 포텐셜을 투자해야만 추론을 시작할 수 있다. 즉, 자신이 최초의 투자자가 되는 셈이다. PM 은 AF 단자에 의해서 트리거 된다. 트리거(Active 상태) 된 PM 은 출력 포트(CF 포트와 OF 포트)를 통해서 포텐셜을 출력하게 된다. PM 이 트리거 되기 위해서는 두 가지 조건이 충족되어야 한다. 첫째로 AF 포트의 입력이 1.0 이상이어야 한다. 이러한 조건이 만족된 상태를 활성화(Focused) 되었다고 표현한다. 둘째로 3 개의 입력포트(PF, LF, NF) 중 최소한 하나에서 1.0 이상의 포텐셜이 입력되거나 TM 계층에서 트리거 신호를 발생 시켜야 한다. 전자를 외부 트리거라 하고

후자를 내부 트리거라고 부른다. 외부 트리거에서는 입력 포텐셜을 출력 포텐셜로 활용할 수 있다. 내부 트리거에서는 자신의 내부 포텐셜을 출력 포텐셜로 전용해서 사용하게 된다.

2. PM 에서의 논리 연산

전술한 트리거 모델을 이용해서 기본적인 논리 연산을 수행할 수 있다.

(1)NOT 연산

NF 포트를 이용한다.

(2)AND 연산

AND 연산은 두 단계로 이루어진다. 첫째로 AF-Filter 는 외부 입력을 각 PM 에 대해서 $1/n$ 로 제한한다. 둘째로 입력의 합이 1.0 이상일 때만 트리거 될 수 있게 제약한다. 예를 들어 규칙 $A \wedge B \rightarrow X$ 가 트리거 되기 위해서는 자료 A 와 B 가 각각 0.5 이상의 포텐셜을 AF 포트에 입력시켜주어야 한다. 셋째로 PF-Filter 는 n 개의 입력 신호 중에서 가장 작은 값을 취하게 된다.

(3)OR 연산

역시 두 단계로 이루어진다. 첫째로 AF-Filter 와 PF-Filter 는 n 개의 외부 입력 중 가장 큰 값을 선택하며 이 값이 1.0 이상일 때 트리거 된다. 둘째로 PF-Filter 는 n 개의 입력 신호 중에서 가장 큰 값을 취하게 된다.

3. 전진 추론

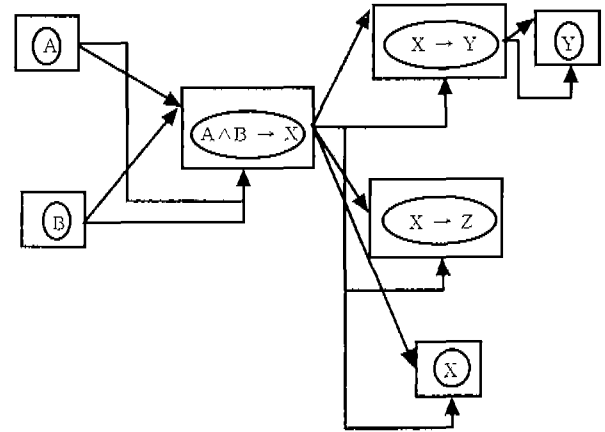


그림 3.2 전진 추론의 예

전진 추론의 원리를 그림 1.1을 이용해서 설명하기로 한다. 우선 자료 A 와 B 가 활성화된다. 그러면 규칙 $A \wedge B \rightarrow X$ 가 PF-Filter를 통해서 입력되는 A 값과 B 값 중에서 작은 값을 취해서 OF 포트를 통해

서 출력한다. 다음으로 출력 $A \wedge B \rightarrow X$ 는 출력 포텐셜을 이용해서 규칙 $X \rightarrow Y$ 와 규칙 $X \rightarrow Z$ 그리고 자료 X 를 트리거 한다. 이 경우 출력 포텐셜이 반드시 입력 포텐셜로 제한되는 것은 아니며 또 반드시 $1/n$ 씩 분배되는 것도 아니다. 출력 포텐셜의 양과 분배 방법을 전적으로 PM 각자의 자율에 맡겨진다. 분배 방법을 결정하는 데는 이성과 감성이 함께 작용한다. 이성적으로는 $A \wedge B \rightarrow X$ 는 규칙 $X \rightarrow Y$ 와 $X \rightarrow Z$ 그리고 자료 X 를 모두 트리거 해주어야 한다. 그러나 감성적인 이유로 이 중 일부의 트리거를 거부하거나 그 양을 조정할 수 있다.

4. 후진 추론

후진 추론에서는 규칙의 전진부에 해당하는 PM 들을 트리거 한다. 예를 들어 $A \wedge B \rightarrow X$ 의 경우 A 와 B 등 두 개의 PM 이 트리거 된다. 만일 A 나 B 에 해당하는 PM 이 없다면 직접 환경을 조사하게 된다. 후진 추론이 성공하게되면 거기서부터 다시 전진 추론이 시작된다. 후진 추론 과정에 포함되는 전진 추론에서는 지나온 History 가 함께 전달된다는 점이 순수한 전진 추론과 다른 점이다. 여기서 History 전달은 디지털적 정보 처리로 TM 계층에서 일어난다.

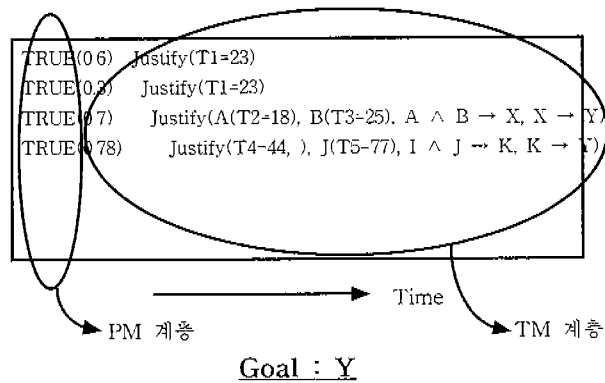


그림 3.3 후진 추론시 BB 공간의 출력

6. 연상 추론

인간의 감성은 사고 과정에서 수많은 개념들로 뒤범벅인 카오스 세계를 창조해 내며 이성은 그렇게 생성된 카오스의 세계에서 유용한 질서들을 추출해내는 것이다. 연상은 바로 카오스 생성 과정에 대응된다고 할 수 있다. 연상은 인간의 상상력을 가능하게 하고 상상력은 바로 창조력의 원천이 된다. 이런 식의 연상은 전통적인 의미 연결망을 통해서도 구현 가능하지만 SAL 에서는 좀더 동적인 속성을 갖는다. 특히 포텐셜 값이 임계치 부근에 있는 PM 들에게는 외부의 사소한

자극도 중요한 의미를 가질 수 있다.

7. 동적 기억 및 집중

SAL 기반 정보 시스템에서는 인간의 사고 과정에서 볼 수 있는 동적 기억 과정(순간적인 망각 혹은 갑자기 생각나는 일 등) 과 비슷한 과정이 일어날 수 있다. SAL 기반 정보 시스템에서 하나의 정보(혹은 지식)는 PM 구조의 TM 내부에 저장된다. 그림 1.1 은 임의의 정보 X 가 저장된 모습을 보여준다.

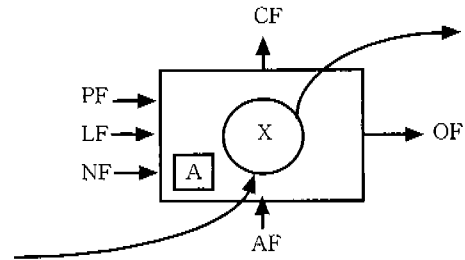


그림 3.4 SAL 기반 메모리 구조

[1]내부 포텐셜이 매우 클 경우

이 경우는 명료하게 기억하는 상태를 의미한다.

[2]내부 포텐셜이 정상이고 NF 가 상대적으로 클 경우

이 경우는 일시적 망각 증상을 의미한다.

[3]내부 포텐셜이 작고 PF 가 상대적으로 클 경우

이 경우는 일시적 기억 상태를 의미한다..

[4]내부 포텐셜에 비해 PF, NF 가 모두 클 경우

이 경우는 생각이 날 듯 말 듯한 상태를 의미한다.

X 에 대한 기억이 미미할 때라도, 일단 X 값이 생각이 난 다음에는 X 와 관련된 여러 PM 들이 연쇄적으로 강화될 것이기 때문에 최소한 당분간은 X 는 망각되지 않을 것이다. 흥분된 PM 들의 확산 과정에는 논리(TM 계층)뿐만이 아니라 감성(PI 계층)도 작용하기 때문에 추론의 흐름은 엉뚱한 방향으로 흘러갈 수도 있다. 이를 막기 위해 Agent 는 특정한 패턴의 포텐셜 주입을 반복하게 되는 데, 이것은 인간의 집중 과정에 대응한다고 볼 수 있다.

3.3 진화(성장과 증식)

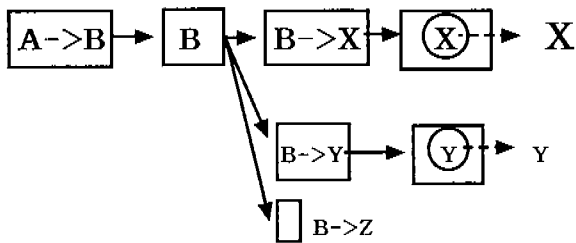
PM 은 보통의 생명체와 마찬가지로 몸과 두뇌로 구성되며 각각 별개의 성장 과정을 거치게 된다. PM 의 몸은 PI 계층에 해당되며 내부 포텐셜의 양적 크기를 외연(PM 의 폭)적으로 표현한다. 한편 PM 의 두뇌는 TM 계층에 해당하며 특성 함수의 질적 성능을 의미한다. 몸과 두뇌는 유기적으로 연결되어 있으며 사람과 마찬가지로 PM 역시 몸이 죽으면 두뇌도 죽고 만

다. 진화는 증식을 통해서 이루어지며 증식은 성장을 전제로 한다.

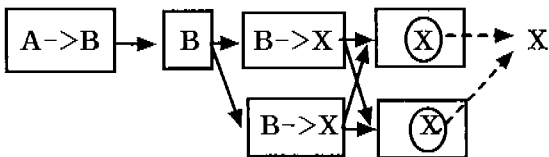
1. (PM 레벨의) 몸체의 성장과 증식

PM 에 있어서 몸의 크기는 내부 포텐셜의 크기로 표현되며 우리는 편의상 내부 포텐셜의 크기에 비례해서 PM 블록의 크기를 결정한다. PM 블록의 크기가 일정 값 이하가 되면 PM 은 소멸되게 된다. 충분한 내부 포텐셜을 확보한 PM 은 증식한다. 증식하기 위해서는 내부 포텐셜이 소비되므로 증식 속도는 내부 포텐셜에 의해 제한된다.

(i) X 의 소비 증가, Y 의 소비 감소, Z 의 소비 소멸



(ii) X 와 규칙 B->X 가 증식된 상태



2. (PM 레벨의) 두뇌의 성장과 증식[2][3]

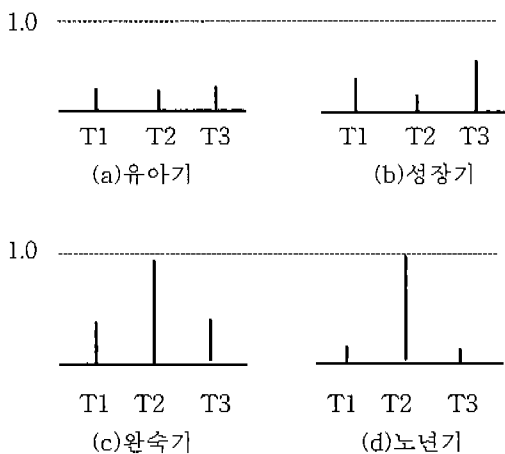
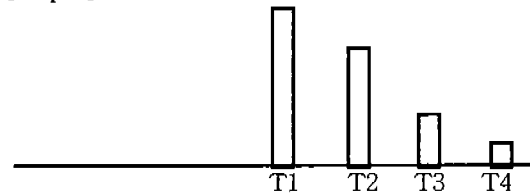


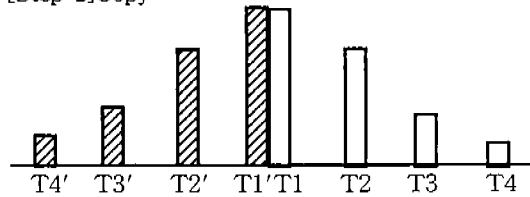
그림 3.5. 성장 과정

정의되며 이때 특성 함수는 정상적인 경우 정규 분포를 한다. 실제로 TM 의 태스크 처리 능력은 특성 함수 패턴의 질에 의해 결정된다. 일반적인 경우 PM 의 두뇌는 유아로부터 전문가로 성장해 간다(그림 3.5).

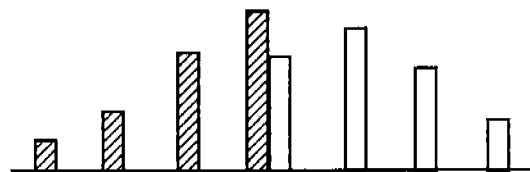
[Step 1]Birth



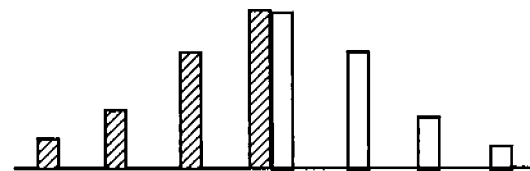
[Step 2]Copy



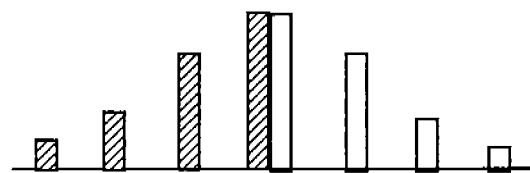
[Step 3]Learn



[Step 4]Sort & Modify



[Step 5]Update OLD-DNA



[Step 6]Reproduction

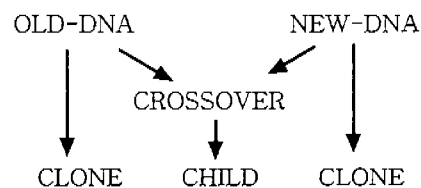


그림 3.6 PM 의 Breeding Cycle

TM 의 KB 는 Cell-Sequence 에 대한 특성 함수로

PM 모델에 있어서의 증식 과정은 여섯 단계를 거치면서 주기적으로 일어나는 데, 이 과정에는 학습 과정이 포함된다. 지면 관계로 자세한 설명은 생략하기로 한다(그림. 3.6)

3.4 학습

SAL 기반 생명체의 성능이 개선되려면 몸과 두뇌 모두가 개선되어야 하고 거기에 부분적인 개선들이 전체적인 관점에서 조화를 이룰 수 있어야 한다. 이처럼 SAL 모델 자체가 복잡한 상호 관계를 내포하므로 학습의 정의 자체부터가 애매할 수밖에 없다. 사실 이런 복잡성은 SAL 이 창발적 모델이라는 점에서 충분히 수용이 가는 일이다. 창발적 모델에서는 진화, 학습, 적용 심지어는 계획과 추론 과정까지도 Holistic 하게 상호 연결되어 있다.

1. PM 레벨의 몸체 학습

우선 PM 레벨에서의 몸체 학습은 몸체의 성장 과정에 해당하는 데, 한 마디로 내부 포텐셜을 잃거나 얻는 과정으로 설명할 수 있다. PM 들은 입력 포텐셜이나 내부 포텐셜을 이용해서 다른 PM 들에게 투자하거나 직접 BB 공간에 솔루션을 게시한다. 포텐셜 소비의 양은 출력 포텐셜의 크기와 출력시간의 함수로 정의된다. BB 공간에 게시한 PM 들은 BB 공간에서의 경쟁 상황을 주시하면서 결과를 예측하게 된다. 만일 예상했던 보상을 얻을 수 없다는 판단이 내려지게 되면 PM 은 즉각 솔루션의 크기를 줄이거나 아예 게시 자체를 철회하기도 한다. 그러면 이 솔루션에 관련된 모든 PM 들이 순환적으로 해당 포텐셜을 반납하게 되므로 포텐셜 투자자들의 투자액이 그만큼 줄어들게 된다. BB 공간에 게시된 솔루션은 최종 사용자(최종 소비자)에 의해서 평가되고 상응하는 보상이 지급된다. BB 공간에서 보상을 받은 PM 들은 솔루션 생성에 참여한 PM 들에게 순환적으로 보상을 분배하게 된다.

2. Rote Learning

동일한 Instance 가 자주 만들어지는 경우 그 Instance 에 해당하는 PM을 만들어서 독립시키게 되는데, 이 과정을 Rote Learning 이라고 부른다.

MAN(Lee).
 MAN(X)->OLD(X).
 OLD(X)->SICK(X).
 SICK(X)->DIE(X).

예를 들어 상기의 KB에서 “Goal : DIE(Lee)” 문제가

자주 등장한다면 OLD(Lee), SICK(Lee) 등의 새로운 PM 들이 생성될 것이고, 이 들에 의해 좀 더 빠른 추론을 기대할 수 있게 된다.

3. PM 레벨의 두뇌 학습

PM 레벨의 두뇌 학습이란 TM 의 KB 학습을 의미한다. TM 의 KB 는 Cell-Sequence 위의 특성 함수(성능 함수 와 흥미 함수)로 표현된다. 이 함수는 정상적인 경우 정규 분포 형태를 하게 된다. 따라서 TM 의 KB 를 정의하는 인자는 Cell-Sequence, 함수의 표준 편차, 함수의 최대 값 크기 등 세 가지이다. KB 의 성능은 바로 패턴의 질에 달려 있으며 따라서 이 경우의 학습이란 이상적인 패턴(환경에 잘 들어맞는 정규 분포 패턴)을 찾아내는 과정으로 정의된다. 이것은 기본적으로 새로운 지식을 습득하는 과정을 수반하지 않으며, 심지어는 학습의 결과로 얻어지는 성능 합(선형적인 성능 합)이 오히려 이전보다 감소될 수도 있게 된다. 정상적인 경우 성능 함수와 흥미함수는 동일하며 따라서 투자 액수는 성능 함수 값과 같다. 성능 함수 값 속에는 솔루션을 얻는 데 소요된 원가 개념이 포함된다. 포텐셜 투자 후 PM 은 자신이 투자한 만큼의 성과를 기대하게 된다. 여기서 성과란 투자액에 일정한 마진이 더해진 값을 말한다. 만일 기대에 대한 실망이 반복되게 되면 PM 은 결국 그 일에 대한 흥미를 잃게 되고 그에 대응해서 투자를 줄이게 된다. 이러한 변화가 안정된 경향을 보이는 경우 PM 은 성능 곡선을 바꾸게 된다. 성능 곡선을 바꾼다는 뜻은 대응하는 Cell 을 바꾼다는 의미이다. 기대치보다 나은 결과를 얻는 경우는 상기 과정이 반대로 일어나고 역시 Cell을 바꾸는 일이 일어날 것이다. PM 은 동적 상태가 안정기로 접어들게 되면 성능 함수 값들을 정렬시키고 크기를 조정하여 정규 분포로 만들게 된다. 이러한 과정에서 Cell-Sequence, 최대값 크기, 표준 편차 등 세 가지 파라미터가 변하게 되는 데, 이것이 PM 레벨에서의 두뇌 학습 사이클이다. 이러한 학습은 주로 동적 환경에서 일어난다. 정적 환경에서는, 일단 형성된 전공이, 점차 강화되는 경향이 있다.

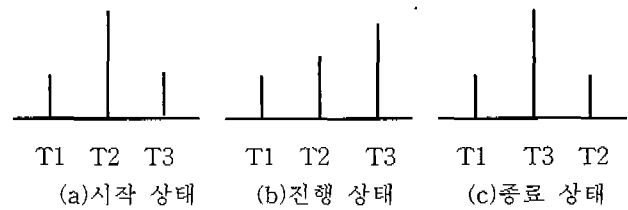


그림 3.7 PM 레벨의 두뇌 학습 과정

3.5 자기 조직화

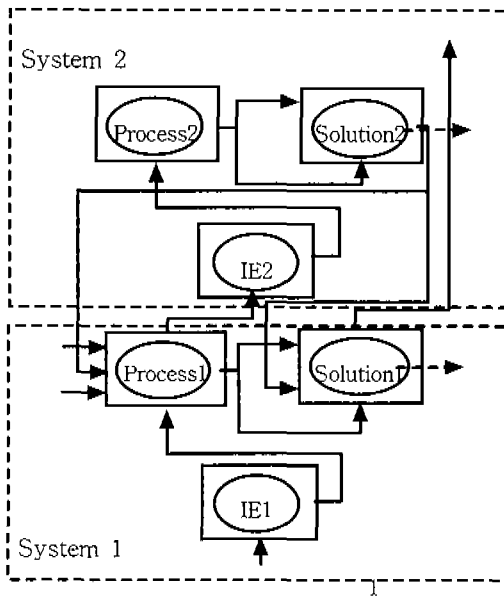


그림 3.8 자기 조직화의 예

그림 1.1에서 IE1 은 Process1 을 트리거하고 트리거된 Process1 은 Process2 를 트리거 한다. 이어서 Process2 는 Solution1 을 BB 공간에 게시한다. 이 시점에서 시스템의 주도권은 System1 에 있다. 한편 Process1 의 NF 포트와 PF 포트간의 충돌로 갈등 포텐셜이 발생되어 CF 포트출력된다. 그런데 CF 포트는 IE2 의 AF 포트에 연결되어 있기 때문에 이 값이 임계값을 넘게되면 IE2 가 트리거 된다. 트리거된 IE2 는 상기와 비슷한 과정을 거쳐서 Solution2 를 생성하게 된다. 생성된 Solution2 는 Solution1 을 억제한다. 한편, Process1 의 LF 포트에 포텐셜(외부 갈등 포텐셜)을 입력시켜 모순 포텐셜의 출력을 강화하게 된다. 즉, IE2 는 자신의 안정성을 높이기 위해서 맴돌이 현상을 이용하는 것이다. 이상 기술한 자기 조직화 과정은 얼마든지 계속될 수 있으므로 적어도 이론적으로는 시스템의 복잡성이 끝없이 증가할 수 있다.

3.6 보안[3]

일반적으로 정보 시스템에 대한 전형적인 공격 유형은 찾기, 훔치기, 파괴하기, 왜곡하기, 트로이 목마 숨기기 등 대략 다섯 가지로 분류할 수 있을 것이다. 그런데 전술한 바와 같이 SAL 기반 정보 시스템이 갖고 있는 창발적 특징들은 시스템에 대한 공격자 즉 해커의 입장을 곤란하게 만드는 측면이 있다.

1) 찾기(Find)

Agent 레벨의 정보들은 하나의 신념체제로 존재하며

n 개의 PM 들이 갖는 정보의 상호 관계에 의해서 창발되는 성격을 갖는다. 이처럼 뇌 세포와 비슷한 PM 들의 창발적 구조는 해커의 '공격 대상 찾기' 를 어렵게 할 것이다.

(2) 훔치기(Steal)

SAL 기반 정보 시스템은 대단히 동적인 특징을 갖기 때문에 비록 창발적 정보 자체의 일관성은 유지된다 할 지라도 그것을 창발시키는 n 개의 정보들은 수시로 변화하게 마련이다. 해커들은 정면 공격 보다는 이런 공격을 시도하게 마련인데, 그렇게 얻은 정보들은 무의미한 정보가 되기 쉽상이다. 왜냐하면 하나의 패턴을 구성하는 데 사용되어진 정보는 다른 패턴 속에서는 별로 의미 있는 역할을 하지 못하기 때문이다.

(3) 파괴하기(Destroy)

SAL 기반 정보 시스템에서의 정보는 하나의 창발적 신념 체계로 존재한다. 따라서 특정 노드를 파괴한다고 해도 시스템 전체로서는 거의 지장을 받지 않는다. 날이 파괴하자면 Family 전체를 공격해야 하는 데 이것은 일반적으로는 해커의 능력 범위를 넘는다. 설사 이것이 가능하다 해도 Family 의 경계가 분명치 않고 중첩되어 있기 때문에 이것 자체도 용이한 일이 아니다. 해커에 의해 Family 의 일부가 손상된다고 해도 일반적으로는 남은 부분에 의해서 즉각 복원된다.

(4) 왜곡하기(Distort)

SAL 기반 정보 시스템에서 PM 들은 포텐셜을 교환하는 방법으로 감성적 태도를 표현한다. 해커가 정보를 왜곡시키는 데 성공하였다 해도 그 왜곡된 정보가 시스템에 유의미한 피해를 주는 것이라면 그 효과는 지속되지 못할 것이다. 왜냐하면 왜곡된 정보에 의해 창발된 정보의 평가 값이 떨어지게 되는 경우 그런 원인을 제공한 PM 은 주위 PM 들에 의해 집중적인 견제를 당할 것이기 때문이다.

(5) 트로이 목마(Trojan Horses)

SAL 기반 정보 시스템에서 특히 동적인 환경에 놓여 있는 PM 은 구조적으로 유한한 수명을 갖는다. 즉, PM 은 태어나서 성장하고 늙어간다. 그리고 수명이 다하면 죽게 된다. 해커나 PM 속에 트로이 목마를 숨겨 놓는 데 성공한다 해도 결국 숙주 역할을 하는 PM 은 늙어 죽을 것이며, 따라서 그 안에 있던 트로이 목마도 함께 사라질 것이다.

3.7 퍼지 개념[4]

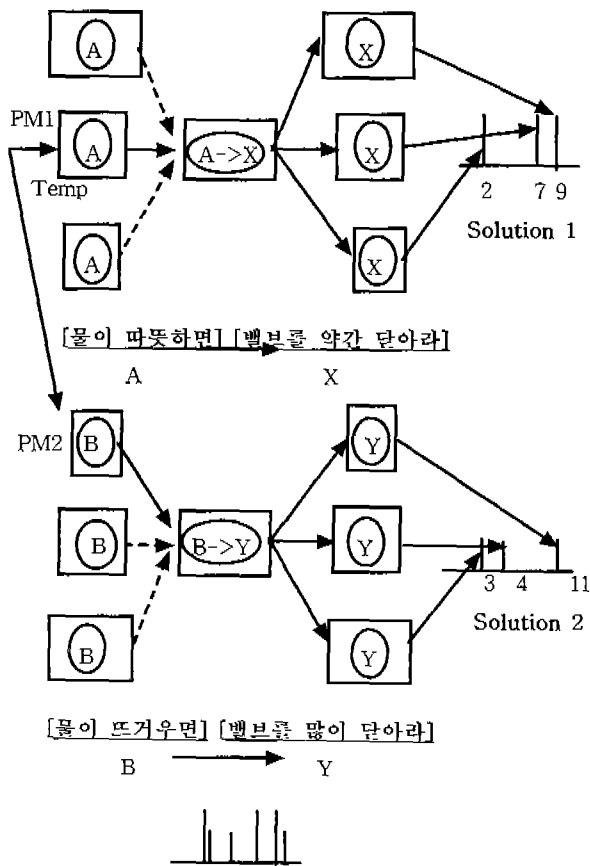


그림 38 퍼지 개념 처리의 예

PM은 다른 PM들과의 연합 행동을 위해서 자신의 전공(Major)을 광고하는 타이틀을 내걸게 되는 데, 이 타이틀이 갖는 추상적 라벨(Label)은 PM 스스로 결정하는 것이다. 이 말은 특히 추상적 라벨에 대한 해석이, 구체적 관점에서는, 제 각각일 수 있다는 의미이다. 본래 PM은 자율적 생명체이고 자신의 해석에 대한 책임 역시 본인 스스로 지는 것이므로 이것 자체가 문제가 되는 것은 아니다. PM이 갖는 이러한 자율성은 우리가 특별히 고려하지 않았음에도 불구하고 퍼지 개념의 처리를 일정 부분 가능케 한다.

3.8 SAL 기반 시스템의 인터페이스

SAL 기반 시스템은 모두 동일한 인터페이스 구조를 갖는다. 즉, 인터페이스는 표정계, 상대계, Agent, 계시판, 포텐셜 주입 장치, Task-Pool 등 5개 부분으로 구성된다. 여기서 표정계는 시스템의 명시적 감정 상태를 나타내며, 상대계는 시스템 내부의 감성적 엔트로피 상태를 나타낸다. Agent 내부는 PM 간의 결합 관계를 보여 주는 데, 여기서 PM의 크기는 해당 모

들이 감성적으로 부추겨진 정도를 의미한다.

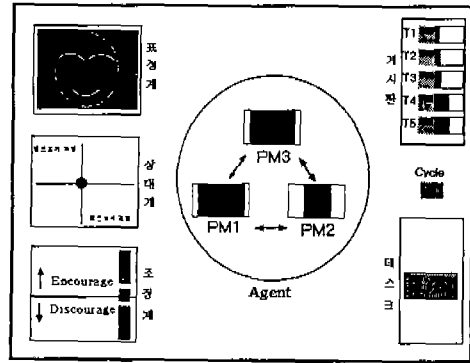


그림 3.9 SAL 기반 시스템의 인터페이스

4 장. 결론

생명체의 속성을 세 가지로 요약한다면, 첫째는 아날로그 기능과 디지털 기능의 결합이고, 둘째는 고차원적 개념(감성)과 저차원적 개념(이성)의 결합이고, 셋째는 단순한 요소(세포)들의 복잡한 상호 관계에 의한 창발 현상이다. 우리는 본 논문에서 이런 생명체 속성들을 기반으로 한 소위 생명체 패러다임 SAL을 제안하였다. 시스템 설계에 SAL 패러다임을 이용할 경우 감성, 적응, 학습, 진화, 퍼지, 자기 조직화, 추론 등의 지능적 기능을 자연스럽게 구현할 수 있다. 본 논문에서는 이점을 연역적으로 증명해 보였다. SAL 기반 시스템은 생명체 고유의 창발적 속성으로 인해 시스템의 복잡성이 증가할수록 누적 예러가 줄어드는 특징을 갖는 데, 이 점은 특히 복잡하고 동적인 환경에서 작동하는 시스템의 설계에 유용할 것으로 보인다. 본 논문에서 제안된 설계 패러다임을 실제 도메인에 적용해 보는 것이 다음 번 연구 과제이다.

참고문헌

- [1]고성범, "생태계로서의 인터넷", 한국생물공학회 춘계 학술 발표, S306, 2000.
- [2]Sung-Bum Ko, Gi-Young Lim "Life-Based System Design", 2000 International Workshop on Advanced Intelligent Systems", P.9, 2000.
- [3]고성범, 임기영, "창발 기반 보안 모델", 한국 퍼지 및 지능 시스템 학회 춘계학술 대회, P.158, 2000
- [4]임기영, 고성범, "창발 기반 정보 시스템", 한국 정보 처리학회 추계 학술 발표 대회, P.449, 2000
- [5]Sung-Bum, Ko, Gi-Young Lim "The development of EBO System as a Life", Proceedings of KFIS Fall Conference, 295-306, 1999.