

디코히어런스 모델

고성범, 임기영

천안공대 컴퓨터공학과, 한밭대학 제어계측과

e-mail:sbko@dragon.cntc.ac.kr

Decoherence Model

Sung-Bum Ko, Gi-Young Lim

Dept of Computer Science, Chonan National Technical College

Dept of Control & Instrumentation, Taejon National University of Technology

요 약

지능은 창발적 현상이며, 창발적 현상은 환원론적 방법론으로 다루기에 적합한 대상이 아니다. 즉, 지능에 속하는 주제들을 하나의 전체론적 틀 안에서 다룰 수 있을 때, 지능의 본질에 보다 효율적으로 접근할 수 있다는 것이다. 본 논문에서는 이런 점에 착안하여 지능적 주제들을 보다 포괄적으로 다룰 수 있는 일반화된 디코히어런스 모델을 제안하였다.

1. 서 론

다른 분야와 마찬가지로 인공 지능 역시 환원론적 패러다임으로 연구되어 왔다고 할 수 있다. 환원론적 패러다임은 부분을 조립해서 전체를 얻을 수 있다는 가설에 기반을 두고 있다. 이에 따라 AI 의 제 분야가 지나치게 배타적으로 연구되어 온 감이 있다. 이런 방법론은, 최소한, 지능이 갖는 창발적 속성에 접근하는 데는 걸림돌이 될 수 있다고 본다. 이와 관련하여 Genetic, 신경망, 퍼지 모델 등을 대상으로 한 Hybrid 모델이 시도되고 있는 데[1], 이것도 기본적으로는 분업 모델이며 결국 기능상의 조립 수준을 넘지 않는 것이다. 최근 산타페 연구소를 중심으로 의식과 생명은 창발적 현상에 속한다는 주장이 제기되고 있다[2]. 그런데 창발적 현상은 하위 레벨로의 환원이 불가능한 현상으로 환원론적 패러다임으로 다루기에 적합한 대상이 아니다. 즉, 창발적 현상은 전체론적 패러다임으로 다루어야 한다는 것이다. 이 문제에 대한 산타페의 제안은 복잡 적응계 모델인데, 홀랜드의 Echo 모델이 대표적인 응용 사례일 것이다[3]. 동일한 관점에서 우리는 일반화된 디코히어런스 모델 (Generalized Decoherence System, GDS)을 제안한다. 원래 디코히어런스는 양자 컴퓨터에서 나온 개념으로 양자 컴퓨터에서는 정보 저장의 단위로 비트 대신 큐비트를 사용한다 [4]. 그런데 큐비트는 0 과 1 을 동시에 포함하는 모호한 상태를 유지한다. 큐비트의 정보가 명료해지는 것은 디코히어런스를 통해서이며, 결국 디코히어런스가 양자 컴퓨터의 프로그래밍에 해당되는 셈이다. 본 논문에서 제안된 GDS 모델은 양자 컴퓨터의 디코히어런스 개념을 일반화한 것이다. 우리는 GDS 모델을 이용하여 지능 문제와 관련된 다양한 주제들을 하나의 틀 안에서 다룰 수 있음을 보이고자 한다.

2. GDS 모델

2.1 GDS 의 속성

여기서는 GDS 모델을 기술하는 데 필요로 하는 속성들을 정의한다.

1)GDS, 타이틀(Title, T), 솔루션(Solution, X)

GDS 는 디코히어런스 기반 시스템으로 정의되며, 타이틀

T 에 대한 솔루션 X 를 출력한다.

2)도메인(Domain, D), 도메인 차원(Domain Dimension, DOD)

대상으로 하는 문제 영역을 하나의 집합이라고 가정하며 이 집합을 도메인 D 라고 한다. 여기서 도메인 D 가 갖고 있는 요소의 개수를 도메인 차원 DOD 라고 한다($X \in D$).

3)코히어런스(Coherence, CH), 히어런스(Herence, H)

타이틀 T가 복수개의 솔루션들을 공유하고 있을 때 GDS 는 코히어런스 CH 상태에 있다고 하며, 하나의 솔루션만을 갖고 있을 때는 히어런스 H 상태에 있다고 한다.

4)컨텍스트(Context, C), 전략(Strategy, S)

GDS 가 코히어런스 상태에서 벗어나기 위한 내부 인자로 전략 S 가 필요하고 외부 인자로 컨텍스트 C 가 필요하다. 여기서 C 는 시스템의 반응이 필요한 입력을 의미한다(C 에는 외부 솔루션 EX 가 포함된다).

5)디코히어런스 메카니즘(Decoherence Mechanism, DM)

CH 상태에서 H 상태로의 전이를 디코히어런스라고 하며 그 메카니즘을 디코히어런스 메카니즘 DM 이라고 한다.

6)코히어런스 타입(Coherence Type, CT)

CT 는 R(표현형) 타입과 D(유전자형) 타입으로 나뉘는데, R 타입은 도메인 요소의 타입을 말하고, D 타입은 특정한 조작을 통하여 R 타입으로 변환 가능한 타입을 말한다.

7)코히어런스 차원(Coherence Dimesion, CD)

GDS 가 T 에 관하여 갖는 표현형의 개수를 T 의 코히어런스 차원 CD 라고 한다.

8)히어런스 차원(Herence Dimension, HD)

인위의 타이틀 T 와 임의의 컨텍스트 C 에 대하여 디코히어런스 메카니즘 DM 이 끌어낼 수 있는 히어런스의 개수를 히어런스 차원 HD 라고 한다.

9)디코히어런스 차원(Decoherence Dimension, DD)

디코히어런스 차원 DD 는 디코히어런스의 특성을 나타내는 파라미터로 DD1 과 DD2 로 구분하며

$$DD1 = (HD-DD)/HD, \quad DD2 = HD/DOD$$

로 정의된다. DD1 은 표현형과 유전자형을 구분하는 파라미터로 이 값이 0.0 이라는 것은 코히어런스가 표현형이라는 의미이고 1.0 이라는 것은 유전자형이라는 의미이다. DD2 는 변신 능력을 나타내는 파라미터로 이 값이 1.0 인 경우는 완벽한 변신 능력이 있다는 의미가 된다.

10)시정수(Time Constant, TC)

디코히어런스에 걸리는 시간을 시정수 TC 라고 한다.

2.2 GDS 의 구조

GDS 는 그림 2.1 과 같이 7 개의 모듈로 구성되는 데, 여기서 H는 히어런스 상태를, CH 는 코히어런스 상태를 의미한다.

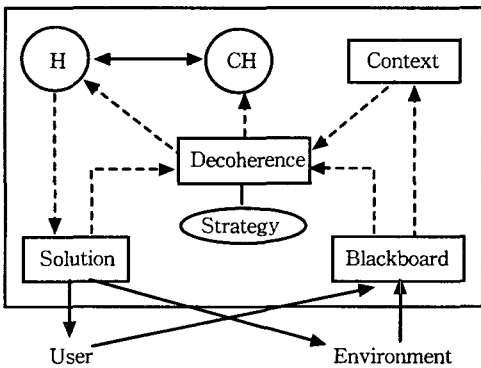


그림 2.1 GDS 의 구조

2.3 GDS 의 동작 원리

GDS 는 내부적 요인인 전략 S 와 외부적 요인인 컨텍스트 C 에 대한 반응으로 H 상태와 CH 상태 사이를 오간다.

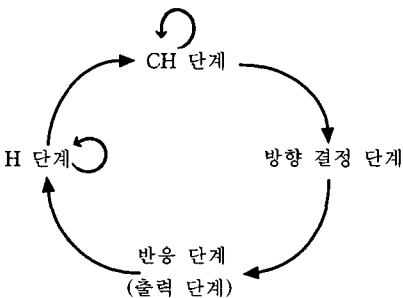


그림 2.2 GDS 의 상태 전이

①CH 단계

타이틀 T 에 관해 모호한 상태(항상성)를 유지한다. 컨텍스트 C 가 발생할 경우 시스템은 방향 결정 단계로 이동한다.

②방향 결정 단계

컨텍스트 C 와 전략 S 의 함수로 결정되는 지향점 D 를 계산(혹은 추론)해 낸다.

③반응 단계

지향점 D 에 부응하는 솔루션 X 를 출력한다.

④H 단계(혹은 Justify 단계)

솔루션 X 에 맞는 내부 상태(히어런스 H 상태)로의 변신을 시작한다(히어런스 H 를 솔루션 X 의 Justify 라고 한다).

⑤컨텍스트 C 가 사라지면 다시 ① 상태로 돌아간다.

2.4 디코히어런스의 유형(Decoherence Type, DT)

디코히어런스 메카니즘은 코히어런스가 유전자형이나 표현형이냐에 따라 다르다. 유전자형(D-Type)은 새로운 노드를 붙여 가는 방법으로 내부 패턴을 만들고, 표현형(R-Type)은 이미 있는 노드들을 잘라내는 방법으로 내부 패턴을 만들어 낸다.

①Instance(ins) : R-Type

변수 형태로 존재하는 CH 에 제약 조건을 만족시키는 값을 Instance 하여 원하는 패턴 H 를 생성해 낸다.

②Reduce(red) : R-Type

CH 값을 Concept-Tree 의 하위 레벨의 값으로 환원시켜서 원하는 패턴 H 가 드러나게 한다.

③Evolve(evo) : D-Type

유전자 형태로 존재하는 CH를 증식과 경쟁(평가 함수 사용)을 통해서 원하는 패턴 H 로 수렴시킨다.

④Correlation(cor) : R-Type

다수의 H 가 공존하는 CH 상태에서, 상호 작용(억제와 흥분)을 통해 원하는 패턴 H 가 부각되게 한다.

⑤Expand(exp) : D-Type

(S, O, G) 로 주어진 초기상태를 전개하여 표현형으로 만들면서 원하는 패턴 H 를 탐색해 나간다.

⑥Learning(lea) : D-Type

지식 획득 도구를 갖춘 초기 상태(CH 상태)에서 특정한 지식을 획득해나감으로서 원하는 패턴 H 로 나아간다.

⑦Computation(com) : D-Type

대수적 연산을 통하여 공존하는 H 중에서 제약조건에 맞는 H를 찾아낸다. 추론, 퍼지 연산 등이 여기에 포함된다.

⑧Self-Structuring(ses) : R-Type + D-Type

계층적 구조를 이용한 자기 조직화 방법(보통은 프랙탈 구조)으로 원하는 패턴 H(일반적으로 복잡)를 구성한다.

⑨Match(mat) : R-Type

CH 상태(여러 개의 열쇠) 중에서 특정한 컨텍스트(특정한 자물쇠)에 맞는 H 가 반응한다.

⑩기타

디코히어런스가 존재하는 것이 확실해도 명시적으로 그 메카니즘이 알려지지 않은 경우도 많다. 예를 들어 양자역학적 디코히어런스도 그러하고 줄기 세포의 분화 과정도 그러하다

2.5 만능 기계(Universal Machine)

만능기계는 기본 요소만으로 구성되며 자율적인 조작(자기 조직화, 진화, 전개, 추론)을 통하여 도메인 내의 어떤 표현형으로도 변신 가능한 만능 변신 시스템을 말한다. 만능기계는 이상적인 GDS 모델로서 세 가지 특징을 갖는다. 첫째로 만능기계는 자율적 시스템이다. 둘째로 만능기계는 코히어런스가 유전자 형태로 존재한다. 셋째로 만능기계의 디코히어런스 차원은 1.0 이다. 첫 번째 특징은 외부로부터의 도움을 필요로 하지 않는다는 의미이며, 두 번째 특징은

저장공간을 최소화할 수 있다는 의미이고, 세 번째 특징은 변신 능력이 완벽하다는 의미가 된다. 이로부터 다음과 같은 명제들을 끌어낼 수 있다.

[명제 1]만능기계는 충분히 작기 때문에 어떤 부분에도 저장될 수 있다.

[명제 2]만능기계는 정의에 의해서 어떤 도메인(만능기계에 의해서 생성 가능한 표현형의 전체 집합)도 만들어낼 수 있다.

[명제 3]도메인을 우주로 잡는다면, 우주의 모든 부분에 만능기계는 존재하고, 각각의 만능기계는 우주 전체를 포괄한다.

명제 3의 만능기계를 태극으로 바꾸면 주역의 태극 이론 체계가 만들어진다. 비슷한 예로는 생명체의 유전자와 홀로그램 필름을 들 수 있다. 유전자는 생명체를 구성하는 모든 세포에 들어 있고 세포는 생명체의 모든 기관을 포괄한다. 또 홀로그램 필름의 아주 작은 부분도 전체 영상 이미지를 포함한다(단, 해상도는 필름의 크기에 비례한다). 만능 기계를 지향하는 보다 실용적인 예로는 Utility Fog[5] 와 PolyBot[6] 이 있다.

2.6 GDS의 특징

GDS는 타이틀 T에 대해 복수개의 솔루션, 암묵적이든 명시적이든, 을 동시에 갖고 있다(즉, 모호한 상태를 유지한다). 이것은 GDS가 Multi-Solution을 기본 전제로 하고 있음을 의미한다(Multi-Solution은 상호 모순적일 수 있다). 여기서 솔루션 집합의 크기를 1로 하면 Single-Solution 체계가 되는 것을 알 수 있다. GDS가 모호한 상태에서 벗어나기 위해서는 전략 S와 컨텍스트 C가 주어져야 한다. GDS는 C와 S의 함수로 솔루션 X($X \in \text{Solution 집합}$)를 결정한 다음, 추후에 X를 정당화하는 조치(자신의 정보 체계를 X에 관하여 의도적으로 합리화하는 작업)를 취하게 된다. 기존의 정보 시스템과 GDS가 가장 다른 점은 전자가 솔루션을 내부의 정보 체계로부터 연역해내는 데 비해 GDS는 솔루션의 핵심 단서를 외부에서 찾는다는 점이다. 여기서 전략 S의 디폴트값은 적용이다(그 반대를 택할 수도 있다). 예를 들어 사용자의 구미에 맞는 솔루션을 내준다는 개념이다. GDS는, 말하자면, 정체성과 일관성을 버리는 대신 변신 능력을 극대화한 시스템인 것이다.

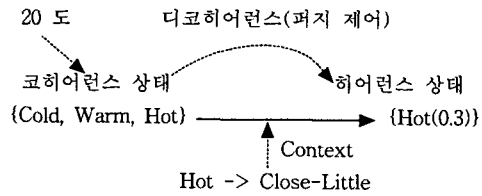
3. GDS의 적용

3.1 사례별 적용

GDS는 외부에서 찾은 솔루션에 맞추어 내부 상태를 바꾸는 변신 시스템인 데, 바로 그 때문에 강력한 추상성을 갖게 된다. 여기서는 구체적 사례를 통해 이점을 확인해보인다.

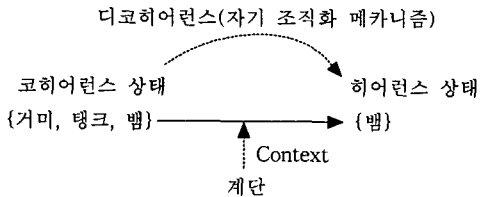
1)퍼지 모델(Fuzzy Model, FUM)

예를 들어 세 개의 소속함수가 온도 20도를 Cold(0.1), Warm(0.7), Hot(0.3)로 정의했다면 시스템은 도메인 $D = \{\text{Cold, Warm, Hot}\}$ 에 대한 모든 값을 동시에 갖고 있는 셈이다. 소속함수에 대응하는 규칙이 ①Hot \rightarrow Close-Little ②Cold \rightarrow Open-Wide 등 두 개라고 할 경우 Context가 ①이면 히어런스 H를 Cold로 만들 것이고 ②이면 Hot로 만들 것이다.



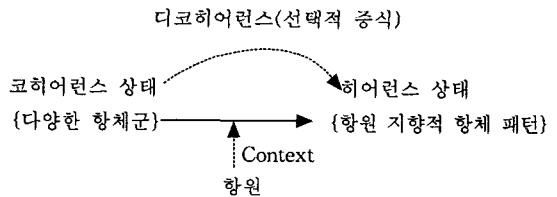
2)폴리 봇 모델(PolyBot Model, PBM)

PolyBot 로봇은 25개의 모듈 집합으로 구성되는 데, 각 모듈은 동일한 구조를 갖는다. PolyBot은 환경에 따라 거미, 뱀, 탱크 등으로 변신하므로, 비록 특정한 모습을 하고 있을 때라도, 기본적으로는 모호한 상태에 있는 셈이다(거미 모습을 하고 있다고 해서 PolyBot의 정체가 거미인 것은 아니다).



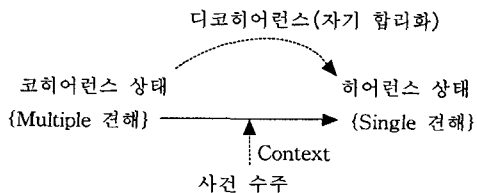
3)면역계 모델(Immune System Model, ISM)

면역계에서 항체는 다양한 공격 유형에 대항할 수 있도록 코히어런스 상태를 유지한다. T세포와 B세포로 이루어진 면역 메카니즘에 의해 침입한 항체에 잘 들어맞는 항원들이 만들어지고 또 증식하는 방법으로 효율적인 면역 패턴이 만들어진다.



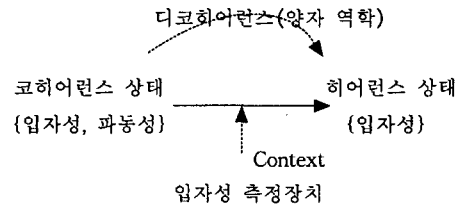
4)변호사 모델(Lawyer Model, LAM)

변호사는 모든 사람들이 갖고 있는 견해들을 한꺼번에 갖고 있다. 즉, 변호사는 견해라는 타이틀에 관하여 코히어런스(CH) 상태에 있는 것이다. 변호사의 견해(H)는 사건을 수주한 시점에서 비로소 명료해진다(이 경우 전략 S는 승소하는 것이고 솔루션 X는 의뢰인의 주장이다).



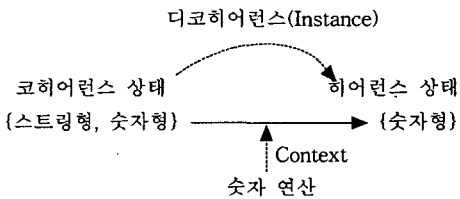
5) 입자성과 파동성(Particle Wave Model, PWM)

빛 속에는 입자성과 파동성이 공존하고 있다. 두 성질이 배타적이라는 점에서 빛은 모호한 상태를 유지하고 있는 셈이다. 이러한 빛의 모호성이 사라지는 것은 특정한 측정장치와 만난 시점에서이다.



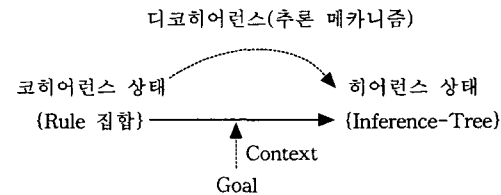
6) 변수 모델(Variable Model, VAM)

Basic 등의 고급 언어에서 명시적 변수 선언을 하지 않을 경우 변수의 형은 String 타입과 수치 타입 등 양쪽 모두가 가능한 모호한 상태를 유지한다(단 디폴트는 스트링이다). 이 변수의 Type 은 구체적인 사용 시점에서 비로소 명료해진다.



7) 추론 모델(Reasoning Model, REM)

KB 는 선언형 언어인 Rule 집합과 추론 도구로 구성된다. 선언형 언어는 멀티 솔루션에 대응한다는 점에서 코히어런스 상태로 볼 수 있으며, 절차형 언어는 단일 솔루션에 대응한다는 점에서 히어런스 상태로 볼 수 있다. Goal 에 의해 추론이 시작되는 데, 그 과정에서 Inference Tree(IT)가 만들어지게 된다. IT 는 절차형 언어에 대응하는 것이며 솔루션 지향적인 내부 상태라는 점에서 히어런스에 해당하는 것이다.



3.2 특성 비교

여기서는 3.1 에서 기술한 7 가지의 사례에 대하여 GDS 의 속성 값을 비교한다(단, CH, H, C, DM 은 3.1 참조)

모 델	S	X	CT	DD1	DD2	TC
FUM	적용	Hot(0.3)	R	-1.5	1.0	L
PBM	적용	이동	D	0.7	<1.0	M
ISM	적용	방어	D,R	<1.0	<1.0	H
LAM	적용(승소)	변론	R	<<1.0	<1.0	H
PWM	적용	입자성	R	0.0	1.0	L
VAM	적용	숫자	R	0.0	1.0	L
REM	적용	{Yes, No}	D	1.0	1.0	M

4. 결론

우리는 양자 역학적 현상, 피지 모델, 추론 모델, 변호사의 행동 패턴 등 다양한 주제(특히 지능과 관련된 주제)들이 GDS 모델에 의해 포괄적으로 다루어질 수 있음을 보였다. 우리는 이것이 GDS 모델이 갖는 광범위한 실용성을 암시하는 것이라고 생각한다. 오늘날의 세상은 멀티 솔루션 환경으로 특정 지워진다. 이런 환경에서는 절대적 의미의 최적해란 존재하지 않는다. 즉, 솔루션은 신성 불가침의 어떤 전제(혹은 공리계)로부터 연역되어지는 것이 아니라 그때 그 때 동적으로 만들어지는 것이다. 이런 환경에 효율적으로 대응하기 위해서 에이전트는, 목시적이든 명시적이든, 모든 해를 동시에 갖추고 대비할 필요가 있다. 즉, 에이전트는 마치 아메바처럼 모호한 상태를 유지하고 있다가 반응이 필요한 환경과 만나는 즉시 그에 맞게 변신을 시도할 수 있어야 한다(즉, 상황에 맞는 솔루션을 내출 수 있어야 한다). 이런 목적에 GDS 모델이 잘 들어맞는다고 생각되며 이것으로 한 단계 진보한 지능처리가 가능해질 것으로 본다. 이 점을 입증해 보이는 것이 다음 번 연구 과제이다.

참고 문헌

[1]LC Jain and RK Jain, "Hybrid Intelligent Engineering Systems", World Scientific Publishing Co. Ltd, 1997
 [2]Roger Lewin, "Complexity", Macmillan Publishing Co, A Division of Macmillan, Inc, New York, 1995
 [3]John H. Holland, "Hidden Order", Welsey Longman, Inc. Reading, 1995.
 [4]www.theory.caltech.edu/~preskill/ph229
 [5]Storrs Hall, "Utility Fog Part I", Extropy 1994.
 [6]http://www.parc.xerox.com/spl/projects/modrobot