



Title	説明機能と知識獲得支援機能(1)
Author(s)	山口, 高平
Citation	大阪大学大型計算機センターニュース. 1985, 59, p. 27-37
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/65668">https://hdl.handle.net/11094/65668</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

# 説明機能と知識獲得支援機能（1）

大阪大学産業科学研究所 山口 高平

## 1. はじめに

今年の8月に米UCLAで、第9回人工知能国際会議（IJCAI85）が開催されたが、企業からの参加者が多く、AIワークステーションとエキスパートシステム構築支援ツール（通称シェル）が展示された会場が賑わった。日本からも計算機・ソフトメーカに限らず、機械メーカや金融関連企業等から多くの人々が参加し、人工知能、特にエキスパートシステムへの関心の高さは相当なものであった。また、論文発表会場でも、エキスパートシステムに関連したセッションが盛況であり、マン・マシン・インタフェースの向上及び知識獲得のテーマに大きな関心が寄せられていた。

本稿では、これらのテーマに関連して、まず説明機能を取り上げ、現在のエキスパートシステムが備える説明機能の問題点を明らかにし、新しいアプローチを紹介する。

## 2. 浅い説明と浅い知識

システムの挙動（質問及び結論）を明瞭にする説明機能は、ユーザに対しては理解度を向上させるために、システム設計者に対しては知識ベース中の誤りを発見させるために有効である。しかしながら従来の説明機能は、以下の前提を暗黙の内に持って非常に制限されたものになっている。すなわち、従来の説明機能は、基本的にはシステムの挙動を忠実に再現するものであり、ルールベースシステムにおいてはルールの連鎖を示すものとなっており、以下に示すコマンドが備わっているのみである。

### 従来の説明機能に関する2つの前提

1. プログラム挙動がユーザに理解可能
2. プログラム挙動の再現がユーザにとって適切な詳細さのレベル

### 代表的なコマンド

- |         |   |  |
|---------|---|--|
| WHY     | → | なぜそのような質問をするのか？<br>ゴール木上で現在の処理ノードから上位のノードを分析                           |
| HOW (×) | → | どのようにして×を決定した（する）のか？<br>×：ルールの細目内容を識別する番号<br>ゴール木上で現在の処理ノードから下位のノードを分析 |

これらのコマンドは、ゴール木を昇降することにより、ルール連鎖を抽出して説明を行う。車の故障診断システムにおいて使用したWHYコマンドの実行例を図1および計算木の処理過程を図2に示す。図1から分かる様に、WHYコマンドを用いてもその処理ノードで用いられたルールしか明示されず、仮にラジエータの機能等を知らないユーザを考えれば、それは妥当な説明文になっていない。このことは、ルールが表層的で具体的な因果関係を表現した知識（浅い知識：Surface Knowledge）であり、診断には使えても、より基本的な事項を説明させる事には使えないことを示唆している。すなわち、従来の説明は、浅い知識を用いた「浅い説明」になっていると言える。

```

! ?- start.
車の状態を入力して下さい。
>> オーバヒート。

私の質問に yes か no で答えなさい。

ラジエータ内の冷却水の水位がLOWレベルより下位にあるか？
>> why.

もし、
=>(2,1) ラジエータ内の冷却水の水位がLOWレベルより下位にある。
(2,2) エンジンの下周りから水が漏れていることが否定される。
ならば、
オーバーヒートは、冷却水不足によって引き起こされる。

```

図1 WHYコマンドの実行例

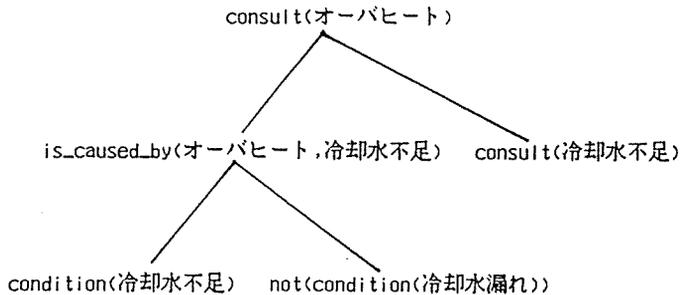


図2 WHYコマンドに対する計算木の処理

### 3. 深い説明と深い知識

従来の説明機能は、2.で述べたことから、システムの挙動の再現がユーザにとって理解しにくい状況や、説明の詳細さのレベルの変更を必要とする状況では、役立つものではない。このような状況においても妥当な説明を実現するには、診断用ルールベースの他に、ドメインを理解する根本的知識（深い知識：Deep Knowledge）を用意する必要がある。この深い知識とは、診断システムにおいて、ドメインの構造・機能・振る舞いに関する知識を指すが、これを利用すれば、なぜそのルールがその状況で使われるか（ルールの正当性）を説明することが可能になる。本節では、このような深い説明を提示するアプローチを2つ紹介する。

#### 3.1 XPLAINにおけるアプローチ〔1〕

XPLAINは、ジギタリス療法のためのエキスパートシステムに深い説明機能を付加するために検討されたシステムである。深い知識としては、Domain Model と Domain Principle が用意され、前者はジギタリス療法に関する様々な因果関係を表わし、後者は、ジギタリス療法に関する手続き的概念を提供する。XPLAINは、自動プログラミングの技術（ルールへのコンパイル概念）を利用して、これらの深い知識からジギタリス療法を行うプログラムコード（診断用知識）を生成する。また、ユーザから説明を求められた時は、これらの深い知識を活用してルールの正当性を提示する。

#### XPLAINの構成要素（図3参照）

Writer	→自動プログラマ
Domain Model	→ジギタリス療法に関する因果関係
Domain Principle	→ジギタリス療法の手続き的概念
Refinement Structure	→Writer が作成した機能プログラムコード
English Generator	→Refinement Structure の実行しているステップを調べることにより、説明文を提示する。

図4はDomain Model の一例であり、箱は状態を矢印は因果関係を示している。また、図5はDomain Principle の一例であり、Goal と Domain Rationale および Prototype Method から構成される。下線を引いたものはパターン変数であり、プログラムコードを生成する過程で具象化される。GoalはWriterが洗練化しようとしているサブゴールとマッチングをとるために利用され、Domain RationaleはDomain Model とマッチングをとるための枠組であり、Prototype Method は Domain Rationale の結果を受けてRefinement Structureを洗練化するた

めに直接使用されるものである。

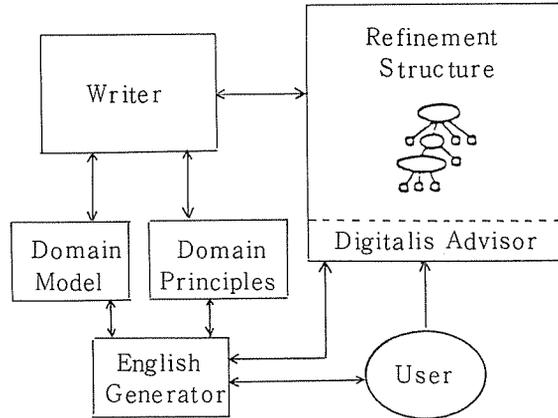


図3 XPLAINシステムの概観

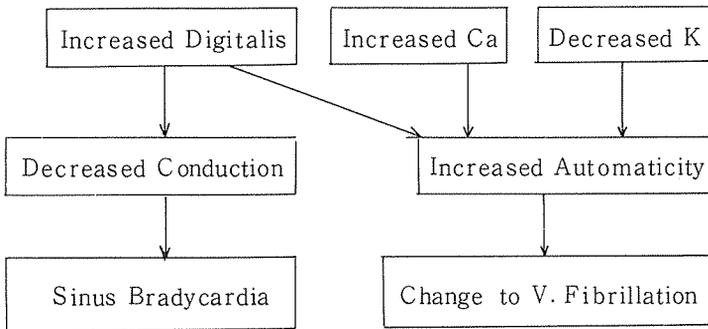


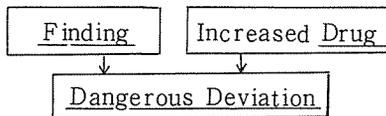
図4 Domain Modelの一例

XPLAINでは、まずRefinement Structure を生成することによって、ジギタリス療法の機能を実現する。例えば、図6において、Writerが「中毒を予期する」というサブゴールを洗練化する場面を考える。この時、図5のGoalとマッチングがとられ、Domain Rationalと図4のDomain Modelのマッチングをとる操作が行われる。この結果、以下の2つの具象化が行われる。

- (1) Finding ← Decreased K , Drug ← Digitalis ,  
Dangerous Deviation ← Increased Automaticity
- (2) Finding ← Increased Ca , Drug ← Digitalis  
Dangerous Deviation ← Increased Automaticity

Goal : Anticipate Drug Toxiciry

Domain Rationale :



Prototype Method :

If the Finding exists  
then : reduce the drug dose  
else : maintain the drug dose

図 5 Domain Principle の一例

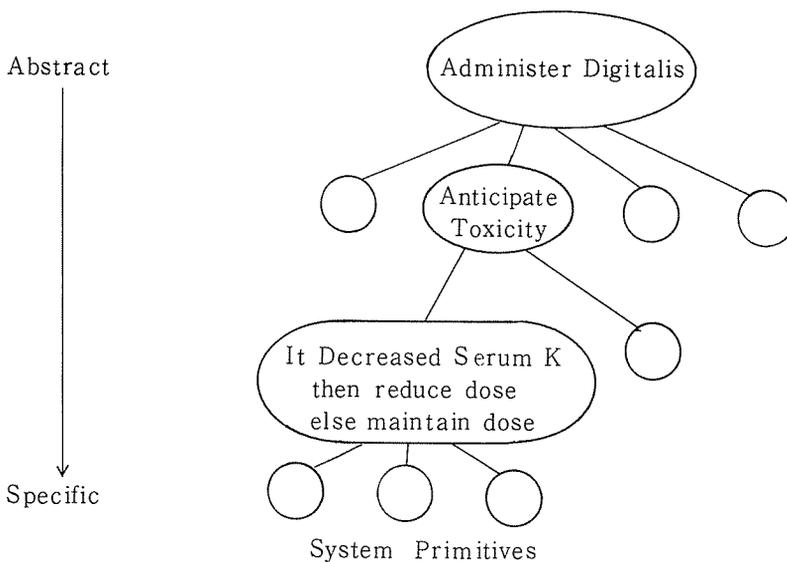


図 6 Refinement Structure の一例

これらのマッチング情報を Prototype Method に適用したものを現在処理されているサブゴールの下に加えることにより、洗練化を一步進ますことができる。この様にして、診断用知識を深い知識に関連させることにより、一般ユーザから説明を求められた時、深い説明文を提示することが可能になる。すなわち、診断用知識の正当性を示すには、それと関連する Domain Model と Domain Principle の 3 つの構成要素を提示すればよい。図 7 と図 8 に、XPLAIN の浅い説明例と深い説明例を示すので、比較してもらいたい。Swartout は、このアプローチを更に進めて、EES (Explainable Expert System) の構成法について検討している [2]。

WHAT IS THE LEVEL OF SERUM CALCIUM?

ENTER NUMERICAL VALUE = = = > why? (The user types "why?" indicating that he questions why he is being asked to supply the value of serum potassium.)

MY TOP GOAL IS TO BEGIN THERAPY. ONE STEP IN DOING THAT IS TO CHECK SENSITIVITIES. I AM NOW TRYING TO CHECK SENSITIVITY DUE TO CALCIUM.

図7 Xplainの浅い説明例

Please enter the value of serum potassium: why?

The system is anticipating digitalis toxicity. Decreased serum potassium causes increased automaticity, which may cause a change to ventricular fibrillation. Increased digitalis also causes increased automaticity. Thus, if the system observes decreased serum potassium, it reduces the dose of digitalis due to decreased serum potassium.

Please enter the value of serum potassium: 3.7

Please enter the value of serum calcium: why?

(The system produces a shortened explanation, reflecting the fact that it has already explained several of the causal relationships in the previous explanation. Also, since the system remembers that it has already told the user about serum potassium, it suggests the analogy between the two here.)

The system is anticipating digitalis toxicity. Increased serum calcium also causes increased automaticity. Thus, (as with decreased serum potassium) if the system observes increased serum calcium, it reduces the dose of digitalis due to increased serum calcium.

Please enter the value of serum calcium: 9

図8 Xplainの深い説明例

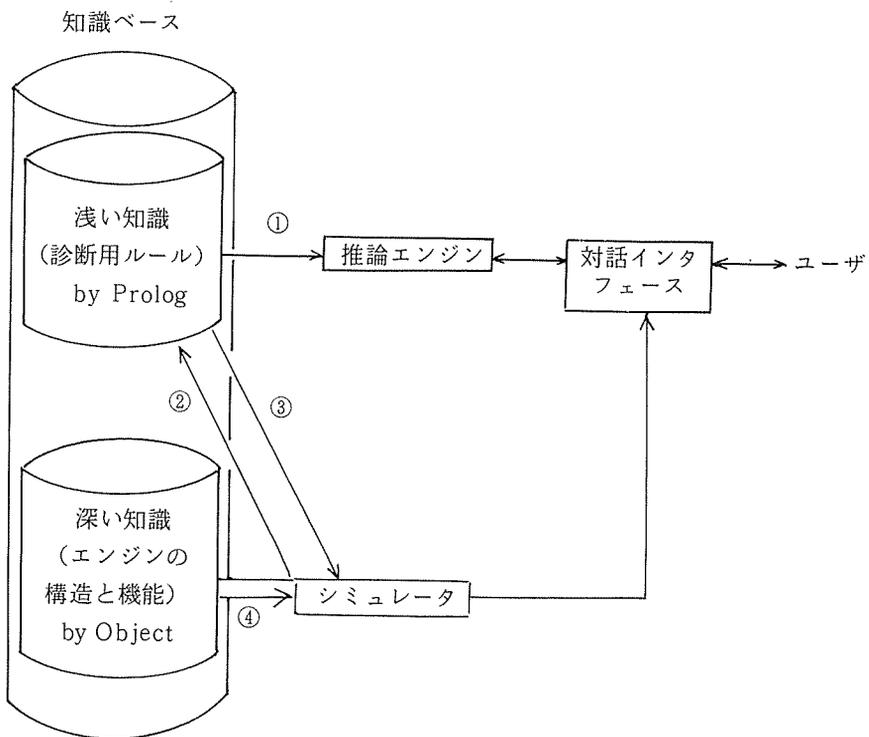
### 3.2 オブジェクト指向の概念を利用したアプローチ

XPLAINのDomain Modelは、概念間の因果関係を平坦に表現したもので、それ自身は静的なものであり、Domain Principleと組み合わせて始めてルールの正当性が処理される。しかしながら、ドメインの構造・機能等は組織的に表現可能であるため、深い知識はオブジェクト指向の概念によりクラス階層として記述できる。オブジェクト指向の概念を利用すれば、深い知識はそれだけで実行できる動的なものになり、構造化されているため拡張も容易となる。以下、文献[3]に基づいて、車の故障診断を例にとり、本アプローチを紹介する。

#### 3.2.1 システムの概観

図9にシステムの概観を示す。本システムは、知識ベースと推論エンジンとシミュレータおよ

び対話インタフェースから構成される。知識ベースは、診断のための浅い知識とエンジンの構造と機能を記述した深い知識から構成される。浅い知識はPrologで直接記述し、深い知識はProlog上に実現したオブジェクト指向の概念で記述している〔4〕。深い知識を表わすクラス階層を図10に示す。クラス components のインスタンス群は、図11の様に、種々の関係を表わすデータをインスタンス変数に蓄えているため、部品の間を関係を表わすネットワークが形成されていると見なすことができる。



- 診断機能 ①：ルールの適用モード  
 ②：ルールへのコンパイルモード  
 説明機能 ③：ルールの連鎖提示モード  
 ④：ルールの正当性提示モード

図9 システムの概観

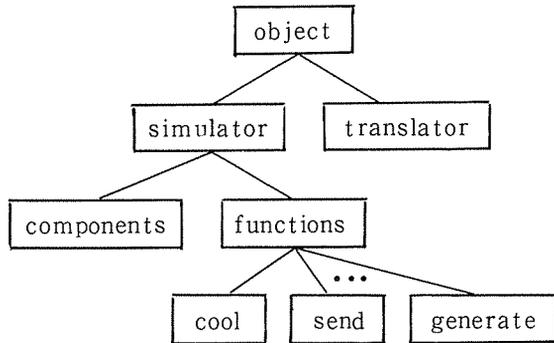


図 10 エンジンの構造と機能を表現するクラス階層

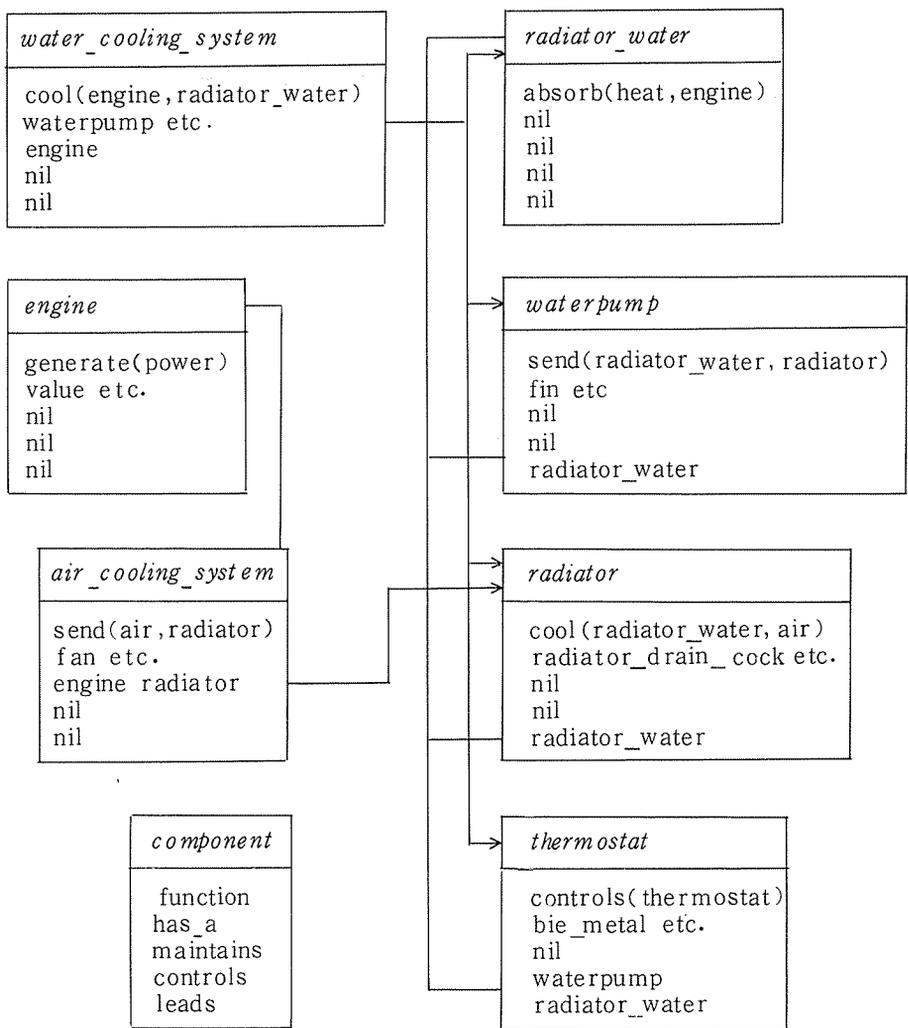


図 11 部品間の関係を表わすネットワーク

### 3.2.2 メッセージパッシングを利用した説明機能

説明機能には、専門家に対してルールの連鎖提示モード、および、一般ユーザに対してルールの正当性提示モードを設けている。ルールの正当性を提示するために、エンジンの構造と機能を表わしたクラス階層が用いられる。例えば「ウォーターポンプが作動していないか？」というシステムの質問に対して、ユーザが“WHY(×). ↵”と入力した場合、この質問を引き出したルールの構成要素 (waterpump) にメッセージ (作動していない = stop) がまず最初に送られ、waterpump の function である send\_radiator\_water\_to\_radiator が働いていないとする。そこで、今度は to 以下の radiator に stop を送る。次に、radiator の function である cool\_radiator\_water\_by\_air が働いていないとし、radiator\_water に stop を送り、radiator\_water の function (absorb\_heat\_of\_engine) が働いていないとする。このようなメッセージパッシングは、locality (部品の動作は接続関係に従って局所的に伝搬する性質) を調べていることに相当する。さらに、absorb\_heat\_of\_engine が働いていないとして、その locality を調べようとするが失敗する。そこで、その function の停止が、診断対象としている故障状況と一致しているかどうかを調べ、一致しているのでシミュレーションを終え (図 12 参照)、そのシミュレーション情報をルールの正当性としてユーザに提示する。ルールの正当性提示モードの実行例を図 13 に示す。

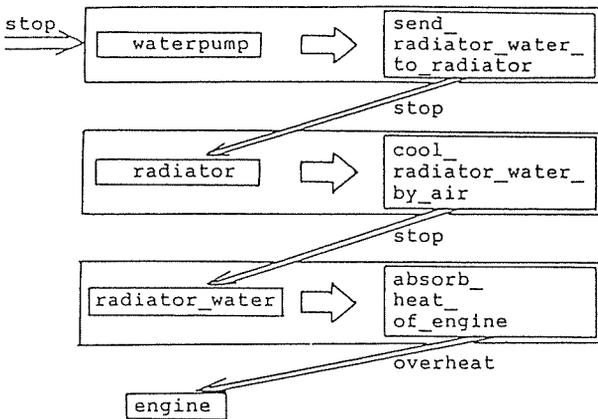


図 12 ルールの正当性を示すシミュレーション

ウォータポンプの機能が停止しているか？  
>> why.

もし、  
⇒ (5,1) ウォータポンプの機能が停止している。  
ならば、  
オーバーヒートはウォータポンプの停止によって生じた。 } ルールの連鎖

ウォータポンプの機能が停止しているか？  
>> why (5)

もしウォータポンプの機能が停止しているならば、ウォータポンプが冷却水をラジエータに送らなくなり、ラジエータが冷却水を冷却しなくなり、冷却水がエンジンの熱を吸収しなくなり、オーバーヒートを起こす。 } ルールの正当性

図 13 ルールの正当性を表示する説明文

#### 4. おわりに

一般ユーザに対して理解し易い説明となっている「深い説明」をドメインの構造・機能・振る舞いに関する知識である「深い知識」と関連させて述べた。この深い知識は、説明機能の向上だけでなく、診断機能の向上にも大きな期待が寄せられている。すなわち、従来のシステムにおいては、適用できるルールがなくなれば、それ以上診断を進めることができなかったが、深い知識を持たせれば、基本原理に立ち戻って考えることが可能になるため、深い知識から新しい診断ルールを自動的に生成して、さらに診断を進めることができるのではないかという期待である。この深い知識を利用したエキスパートシステムの研究は、まだ初期の段階であるが、次世代のエキスパートシステムの構築への足掛かりになるのは間違いないであろう。次回では、診断ルールの自動生成よりは現実的な話である「知識獲得支援機能」について、代表的な支援システムを紹介しながら述べる予定である。

#### 参考文献

- [1] W.R.Swartout : "XPLAIN : a System for Creating and Explaining Expert Consulting Programs", Artificial Intelligence Vol. 21, No. 3, pp. 285 - 325 (1983)
- [2] R.Neches, W.R.Swartout, J.Moore : "Explainable (and Maintainable) Expert System", 9 - th IJCAI, pp. 382 - 389 (1985)
- [3] 山口、溝口、小高、川口、野村、角所 : "エキスパートシステムにおける定性的推論と深い説明機能", 情報処理学会第31回全国大会, pp.953 - 954 (1985)

- [4] 小高、溝口、山口、川口、野村、角所：“オブジェクト指向の概念を導入したPrologとその支援環境”，情報処理学会第31回全国大会， pp. 955 - 956 (1985)
- [5] B. Chandrasekaran, S. Mittal：“Deep versus compiled knowledge approaches to diagnostic problem - solving”， Developments in expert systems, edited by M. J. COOMBS, pp.23 - 34, Academic Press (1984)
- [6] Davis, R. and Lenat, D.B.：“Knowledge - Based Systems in Artificial Intelligence” MacGrow - Hill (1982)
- [7] A. Barr/E. A. Feigenbaum 編、田中 / 淵 監訳 “人工知能ハンドブック第Ⅱ巻” 共立出版 (1983)
- [8] “Buiding Expert Systems” edited by F. Hayes - Roth, D. A. Warterman, D. B. Lenat, Addison - Wesley (1983)
- [9] K. L. Clark and F. G. McCabe：“PROLOG： the language for implementing expert system” Machine Intelligence 10, pp. 455 - 470 (1982)
- [10] Goldberg, A. and Robson, D.：“Smalltalk - 80, The Language and Its Implementation” Addison Wesley (1983)