

1F-4

日常的世界観にもとづく
実行可能な仕様記述言語 NAIVE
— その概要と世界観 —

日野克重
富士通㈱

1. はじめに

ソフトウェア開発の生産性および品質の向上にとって、ソフトウェア記述言語は最も根本的な役割を果たすものと考えられる。なぜなら、ソフトウェアの開発とは、つきつめるところ、「問題(要求)自体の表現とソフトウェア記述言語との間の写像操作」と言うことができ、ソフトウェア開発のあらゆる困難は、それら両者の間の階位的、構造的乖離からきていると考えられるからである。ここでもし、「問題自体の表現に近いレベルで仕様を書けば、それがそのまま実行もできる」ような言語が実現できれば、設計からプログラミング、ならびにテスト工程が大幅に省略され、ソフトウェア開発の困難のおおかたは解消する。

こうしたねらいをもつ言語は、実行可能な仕様記述言語とよばれ、すでに種々のものが提案されている。しかし、現時点では、記述・読解性の点で難解、動的システムや大規模システムの記述が困難、データベース操作などを含む実用システム全体を作る能力が十分でない、などの難点を抱えており、その本来のねらいを十分達成したものは出現していない。

ここで提案する言語NAIVEは、日常的世界観にもとづき、自然語に近い形で問題対象を記述する言語である。具体的には、実体、関係、行為、変化、行為主体(エージェント)などの基本概念からなる日常的世界観にもとづいており、データ構造、ポインタ、プロセス、データ入出力、同期通信などの従来の計算機プログラミング上の概念を隠蔽した言語である。さらに、静的事態の記述についてはすでに実績のある述語論理をカバーしている上、並行処理などの実世界の動的側面の記述までを一つの論理体系の中に収めた言語である。データベースプログラム言語としての機能も備えている。すでにインプリメントされており、実用規模のソフトウェアシステムの記述実験も進められている。

本稿では、言語NAIVEについて、その背後にある世界観および論理体系について述べる。

2. 言語NAIVEの世界観

仕様記述言語の設計において、その言語の背後に明確な世界観を設定しておくこと、そして、そのとき、いかに自然な世界観を設定するかが、決定的に重要である。言語NAIVEでは、それが依拠する世界観として、日常的世界観を採る。

日常的世界観を採ることの妥当性については、およそ次のように考えれば納得されるだろう。すなわち、

われわれが開発するソフトウェアシステムの多くは、われわれの日常生活の中に埋め込まれて使用される。したがって、このようなシステムに対する要求仕様も、その原初時点では、日常的概念や言葉で表現されているだろう。そもそも動的かつ複雑なシステムの開発が必要になるのも、われわれの日常生活の中ですでに高度な論理処理や並行処理が行われているからである。そうだとすれば、こうしたソフトウェアの仕様を自然に記述する言語を設計する上で、われわれが日常世界を眺めるときの世界観(日常的世界観)や自然語に規範を求めるのが妥当である。また、日常的世界観や自然語の中には、こうしたソフトウェアを記述するのに必要な要素はすべて備わっているはずである。

ところで、その自然語の論理的な側面を形式化した形式言語としては、すでに述語論理やその拡張としての内包論理などがある。それらは、個体、関係、および時間の概念を基本的枠組みとした言語であり、その点で自然語の最も基底にある世界観を反映した言語であるといえる。言語NAIVEもその枠組みを引き継ぐ。しかし、日常的自然語をもう一度観察してみると、これらの既成の論理系では、実世界の記述あるいは動的システムの記述にとって重要ないくつかの概念がとり残されていることがわかる。それは、以下に示すような概念である。

(1) 実体に関する概念 — 種、属性、実在性 —

種：自然語における普通名詞は、個体の属する種(クラス)を表すものである。この普通名詞は、自然語表現においては動詞や形容詞などとならんで中心的役割を果たしている。どの個体もなんらかの種に属しており、かつ、種の指定が人間が頭の中で行っているであろう情報検索においても効率上有利であることなどがその理由であろう。ソフトウェアの仕様記述においても、種概念は、プログラムの自動生成や仕様の検査において有効にはたらく。

属性：これは、自然語における「～の住所」や「～の年齢」のような属性名詞に相当する概念である。既成の論理系でも関数の概念がこれに相当するが、ここでは、関数は「一意的な関係」の別名として、いわば記述の便法として扱われている。言語NAIVEでは、実体概念に付帯する本質的な概念として属性概念を導入する。実際上も、関係概念とは別に属性概念を導入することは、アクセス効率のよいデータ構造を自動設計する上で不可欠である。

実在性：これは、自然語における「有る」や「無い」に相当する概念である。述語論理のように本来静的な世界を記述する言語においては、個体の生成や消滅は生じないので本概念は必要ないが、世界の動的側面を記述する言語では、この概念は必要となる。実際、オペレーティングシステムやシミュレーションシステムなどでは、個体の生成・消滅に相当する処理は頻繁に現れる。また、「物(たとえば動物)は生成された瞬間から自発的に行動をはじめるといふ日常的世界

The Specification Language NAIVE based on
Everyday World View - its Outline and World
View -

Katsushige HINO
FUJITSU, Ltd.

観に着目すれば、この実在性の概念は、後述の行為主体の概念とも有機的な関連をもつべきこともわかる。

(2) 行為に関する概念 - 行為, 主体, 脈絡 -

「世界は変化するものであり、世界の変化はかならず、なにがしかの主体による行為によってもたらされる」というのは、日常的世界観の中の基本的なものの一つであろう。この点から、実世界の動的側面を自然に記述するために行為の概念を導入することは自然なことであると考えられる。

さらに、行為の概念は、おのずと、行為主体の概念および行為脈絡の概念を導く。実世界の複雑な現象の系列も、いくつかの行為主体のおこなう脈絡のある行為の重ね合わせとして捉えるとき、はじめて整然と理解できる。同様に、動的並行システムの記述および検証に際しても、行為主体の概念と行為脈絡の概念の導入は、仕様全体を直和的に分割して把握する上で有効にはたらく。

この点、時間概念はあっても行為概念が導入されていない伝統的論理(内包論理や通常的时间論理)では、行為主体や行為脈絡の概念が現れる余地がないので、接続、反復、および入れ子構造などの世界変化の脈絡や、行為主体の概念に本質的に依拠する並行処理が自然には記述しにくい。行為概念をもたないこのような言語で動的システムを記述する場合、問題文には現れない多数の内部的な状態識別子を使って、状態遷移的プログラミングをせざるを得なくなる。

(3) 世界への外乱に関する概念 - 神, 天使 -

日常的世界観においても、世界は完全に閉じた自律系とは考えられていない。世界を創成するものは、世界の外にあり、かつ、世界ができた後も、なんらか目にみえないものからの外乱があるものと捉えられている。ソフトウェアシステムの場合も、システム操作者の操作コマンドの投入や、当該システムの下位システムや隣接システムからの種々の事象通知などが、この外乱に相当する。記述した仕様の正当性が外乱の生起系列に依存する場合もあるように(無飢餓性の議論はその典型例である)、動的システムの記述にとって、外乱が記述できることは本質的である。神と天使の概念は、それぞれ、システムへ外乱を発生させるものとその外乱自体とを日常的に言い表した概念である。

上に挙げた諸概念を従来の述語論理的世界観に付加することにより、新たに図-1および図-2のように要約できる世界観が得られる。言語NAIVEは、基本的にこの世界観にもとづく。

-) 実体と関係 : 世界の中にはいくつかの実体が存在し、それらの間には種々の関係が成り立っている。実体とは、種が付与された個体のことである。実体は、実在性をもっている。実体はまた、いくつかの属性をもちえる。なお、成立している関係のことを事態とよぶ。世界の状態とは、事態の集まりのことである。
-) 行為と変化 : 世界の状態は時間とともに変化する。そして、その変化は行為によってもたらされる。行為とは、世界の状態を参照しながら新たに事態を成立(非成立)させることである。行為には脈絡がある。
-) 行為主体 : 行為にはかならずその行為の主体が唯一存在する。行為主体になり得るのは、実体と天使だけである。行為主体たちは、それが実在しているかぎりその行動本性にしたがって、世界をなかめながら、おのおの独立に行動する。
-) 神と天使 : 世界を創成するのは神である。神の意思は天使を介して世界に伝えられる。神は、世界の不変的性質、すなわち、どんな実体および関係が世界に現れ得るか、ならびに、実体および天使がどう行動するものか(それらの行動本性)を知っている。天使は行動するが、実在しない。
-) 論理 : 世界はつねに論理にしたがっている。

図-1 言語NAIVEの世界観

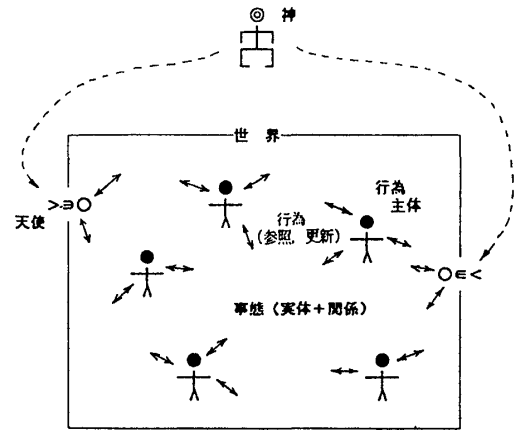


図-2 言語NAIVEの世界観(図解)

3. 論理体系

2章で示した当言語の世界観は、モンタギューの内包論理を基本的枠組みとした一つの論理体系として形式化できる。ただし、標準的な内包論理に対していくつかの拡張が必要である。主な拡張点を以下に挙げる。

- ① 基本タイプとして、e (個体), t (命題), s (時刻, 世界)に加えて、p (行為)を新たに導入する。また、種概念の導入に対応して、タイプeのもとにサブタイプを定義する。
- ② 実在概念に対応して、個体領域は、時間とともに変動するものとする必要がある。
- ③ タイプt式(通常の命題式)の解釈が時刻(1時点)を指定すれば定まるのに対し、タイプp式(行為式)の解釈は、行為主体と時区間(時点の対)とを指定しなければ定まらない。

内包論理に対してこれらの拡張を施すことにより、当論理体系は、「行為主体の概念をもつ時区間内包論理」と呼ぶべき性格をもつことになる。

4. おわりに

言語NAIVEの世界観を提示し、その論理体系についてもふれた。この世界観は、次のような特徴をもったソフトウェアモデルになっている: ①マルチエージェントによる並行協調動作やオブジェクトデータベースなどの重要なソフトウェア概念を統一的に含んでいる、②日常的世界観に近く計算機概念を隠蔽したものであるがゆえにモデルとしての了解性、連想性、および抽象性が高い。

謝辞

本研究について日頃御指導戴く東京大学 田中英彦教授ならびに当社技術担当部長 河田汎博士に深く感謝致します。

参考文献

- 1) Dowty, D.R., Peter, S. and Wall, R.E.: Introduction to Montague Semantics, D.Reidel, Dordrecht, Germany, 1981
- 2) 榎本肇編: ソフトウェア工学ハンドブック, オーム社, 1986
- 3) 日野克重: 日常的世界観にもとづく実行可能な仕様記述言語NAIVE - その言語仕様と記述例 -, 本予稿集, 1992