

生物発生メカニズムを取り入れた対象指向言語

2AG-4

今村 聡

機械技術研究所

1. はじめに

工業技術院・産業科学技術研究開発制度「新ソフトウェア構造化モデル」の1テーマとして、制約対象指向言語 FDL-II を開発している。特徴としては、(1)等式、不等式、述語を扱える多属性制約の伝播による、モデル間の情報通信、(2)サブモデル及びパラメータの共有によるアセンブリモデルの構造表現、(3)関係データ検索機能と制約との結合、(4)会話型設計支援環境を提供する構造化スプレッドシート・インタフェース、などがある¹⁾。さらにこの言語は、動的構造変更能力を持つことが重要な特徴であるが、「運賃計算の連続的改訂問題²⁾」など、仕様がしばしば変更されるソフトウェア開発問題一般に適用するには、十分な柔軟性を持つとはいえないことが判明した。

筆者は従来より、およそ 40 ～ 10 億年前にかけて起こった単細胞生物の進化過程³⁾、多細胞生物の発生過程⁴⁾に興味を持っていたが、そこから発想を得た、生物発生型アルゴリズム (biogenetic algorithm) 導入することによりソフトウェアの動的構造変更能力を高度化させることを考えた。以下、生物発生型アルゴリズムを紹介する。

2. 生物創世発生型アルゴリズム

2.1 単細胞生物の進化における共生と遺伝子収集

太古の原核細胞生物は他の原核細胞生物を細胞内に取り込み、共生することにより生存能力を高めた。ミトコンドリアや葉緑体は共生の名残であると考えられている。また、他の原核細胞生物の DNA を自分の DNA に取り込んで利用することもあったと考えられている。

細胞をオブジェクト、DNA をオブジェクト内

の記述すべてになぞらえると、共生や遺伝子収集の模擬アルゴリズムとして次のようなものが考えられる。

(1) 部品の取り込み

:add_part (Object,Slot,Class,Location) 操作により、Class のインスタンスとそれを格納する Slot を Object 内に動的に作成する。

(2) 構造融合

:merge (Instance1,Instance2) 操作により、インスタンス同士の構成要素 (パラメータ、部品、制約、メソッド) すべての和集合をつくる。同名の構成要素があった場合それらも再帰的に融合させる。パラメータ同士の融合の際、その値に違いがあった場合は前項を優先し、この操作をきっかけに制約伝播が起こることもある。

(3) 動的継承機能

インスタンスが任意の数のクラスを動的に継承できる機能である。継承は制約とメソッドについてのみ行われ、継承されるクラスのパラメータ集合は継承を行うインスタンスのパラメータ集合に包含されることが動的継承を許す条件である。動的継承はスイッチを onoff するように、任意タイミングで任意の数行うことができる。スイッチの on に相当する操作として :add_inherit (Instance,Class)、off に相当する操作として :delete_inherit (Instance,Class) が用意される。

2.2 発生における細胞の分裂と分化

多細胞生物の発生は一つの受精卵の分裂から始まり、分裂を繰り返して多くの細胞からなる生物体を構成していく。この時、発生の段階が早いほど細胞は多くの種類の細胞に分化できる可能性を持つが、発生が進むにつれてこの可能性は小さくなっていく。別の言い方をすると、細胞は発生の段階が早いほど多くの遺伝子を利用でき、発生が進むと少なくなる。

オブジェクトにも分裂操作を導入する。分化を伴わない分裂とはオブジェクトのコピー操作と同

等であるが、分化を伴う分裂では記述内容を二つのオブジェクトに分ける操作になる。この際、分裂によりできた二つのオブジェクトがそれぞれ整合性をもった記述を持つ必要がある。

分化を伴ったオブジェクトの分裂は次のように定義できる。オブジェクト $O = \{Sp, Sa, Sc\}$ で表現するここで、 Sp は部品集合、 Sa は属性集合、 Sc は制約（またはメソッド）集合である。 O をそれぞれ $\{Sp1, Sa1, Sc1\}$ 、 $\{Sp2, Sa2, Sc2\}$ を包含するようオブジェクト $O1$ と $O2$ に分裂させる場合、 $O1$ 、 $O2$ は次のような内容にならなければならない。

$$\begin{aligned} O1 &= \{Sp1 \cup \text{part}(Sc1), Sa1 \cup \text{atr}(Sc1), \\ &\quad Sc1 \cup \text{cons}(\text{atr}(Sp1) \cup Sa1)\} \\ O2 &= \{Sp2 \cup \text{part}(Sc2), Sa1 \cup \text{atr}(Sc2), \\ &\quad Sc2 \cup \text{cons}(\text{atr}(Sp2) \cup Sa2)\} \end{aligned}$$

ここで $\text{part}(Sci)$ は制約 $C \in Sci$ いずれかにより参照される部品集合、 $\text{atr}(Sci)$ は制約 $C \in Sci$ いずれかにより参照される属性、 $\text{atr}(Spi)$ は部品集合 Spi に含まれる属性の集合、 $\text{cons}(Sai)$ は参照する属性がすべて Sai に含まれる制約の集合を表す。

2.3 アポトーシス

動物の発生では、遺伝子によりプログラムされた細胞の自殺、アポトーシスと呼ぶ現象が起こる。熱、毒などにより引き起こされる細胞の壊死、ネクローシスでは、炎症など副作用を伴うが、アポトーシスを起こした細胞は周辺の細胞に吸収され、副作用を伴わない。

アポトーシスのように副作用を伴わずに、オブジェクトの構成要素を消去するための操作として、次のようなものを用意する。

- (1) `:delete_object(Instance)`
- (2) `:delete_slot(Instance)`
- (3) `:delete_constraint(Instance)`

(1)は、どこからも参照されていないオブジェクトを消去する。但しその構成要素のうち、他から参照されているものは消去しない。(2)はオブジェクトの格納場所であるスロットを消去し、格納されていたオブジェクトも他から参照されていない場合に限り同時に消去する。(3)はクラス継承にマスクをかけることにより制約記述を無効にする。

2.4 ホメオ遺伝子

ホメオ遺伝子は動物の器官を作るために必要となる何百もの遺伝子が連続的に発現される際のトリガーの役目を果たすもので、それぞれの器官に対応するホメオ遺伝子が一つずつ存在する。

ホメオ遺伝子に相当する操作として、配置演算子というものを用意する。これはオブジェクトの構造変更を行う操作列をひとまとめにし、その起動条件を付加したもので次の書式で記述する。

<起動条件> ==> <構造変更操作列>;

起動条件はオブジェクトのパラメータが満たさなければならない制約条件を記述し、構造変更操作列は部品の追加、アポトーシス、構造融合、分裂分化などの操作で記述する。記述例を一つ示す。

@req_num_of_pairs >= 2, @req_num_of_pairs <= 3 ==>

```
:add_part(self, counter_shaft, #counter_shaft),
:add_part(self, parallel_gears, #parallel_gears),
:merge(belt_drive!flw_shaft, counter_shaft),
:merge(counter_shaft, parallel_gears!drv_shaft),
:merge(main_shaft, parallel_gears!flw_shaft);
```

これは、歯車動力機構のオブジェクトモデルの構成を自動的に変更する配置演算子の例である。配置演算子は、全域的にも局所的にも適用でき、状況に応じてオブジェクトシステムを自動的に変更、構築していくことができる。

3. おわりに

旋盤設計¹⁾、運賃計算表改訂問題⁴⁾などの例題にとりくみ、生物発生型アルゴリズムの検証を行っている。操作のビジュアル化、undo 機能、デバッグ機能の強化などの課題がある。

参考文献

- 1) 今村聡、工業設計支援用制約対象指向言語の開発(第1報)、精密工学会誌, 60.9. (1994) 1242.
- 2) NHK 取材班、生命 40 億年はるかな旅、NHK 出版 (1994)
- 3) 岡田節人、からだの設計図、岩波新書 358 (1994)
- 4) 中島秀之、松原仁、本位田真一、協調プログラミング例題集、b i t 別冊、共立出版 (1996)