

6F-6

仕様記述の実行処理に必要なとする推論機能

橋本 雄治 鴨志田 稔 榎本 肇

芝浦工業大学

1. はじめに

プログラム中で定義と手続きの混在というある種の混乱を除くためには、仕様定義部と手続部分とを明確に分け、ソフトウェア記述プロセスの中に仕様記述から実際に実行される記述を得るプログラミングの段階を取り入れる必要がある。プログラムは仕様定義部と制御定義部とを結合したものと考え、仕様定義部にある種の制御情報を与えることによって並行手続が導出される。本論文ではTELLの研究成果に基づいた新しいソフトウェア環境-TELL-SCOOPS (Total Elaboration Language for Sentence Constraint and Object Oriented Programming Supports.<sup>[1]</sup>)、以下TELL-SCPと略)のもとでの推論機能を用いた並行手続の導出を行う。推論機能については、画像の描画を例にとってWELL-PPP<sup>[2]</sup>に組み込むことなどを述べる。これらは制御定義部の動的な記述において、そしてオブジェクト間の関係に基づいて働くものである。

2. TELL-SCPにおける仕様・手続きの表現

従来からソフトウェアの仕様は自然言語で書かれているが、自然言語は分かりやすい反面、記述の内容があいまいになるという欠点がある。さらに各種の文書合成、検証といった仕様の意味をコンピューターに実行させるのも不相当であると考えられてきた。また、手続きを記述する通常のプログラミング言語では、実行内容を直接手続の形で記述するために、問題の本質的な定義とその制御情報が混在した形となり混乱を招くこともある。

そこでTELL-SCPでは、あいまい性を捨象した擬似的自然言語を採用し、その一方で問題の本質的な定義とその制御を分離した記述方法を採用した。またソフトウェアの規模が大きくなった現在、階層的仕様から生成されたオブジェクトの集まりでプログラムを構築する、オブジェクト指向によるプログラミングを目指している(図1)。

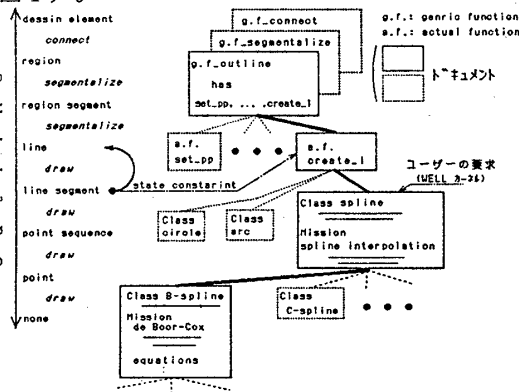


図1. TELL-SCPによるオブジェクトの階層的仕様 (WELL-PPPのフレーム構造を例にして)

TELL-SCPにおいて、オブジェクトを実現するための表現は、仕様(specification)とミッション(制御情報、mission)の二つのカテゴリーに分類されるが、この仕様とミッションを合わせて手続きを生成するドキュ

メントとなる。並行手続きを形成するために必要とするすべての記述単位は静的、あるいは動的な性格を持ち、さらに項目的、関数的性格を持ち、直交性を表現している。TELL-SCPでは静的項目をクラス、動的項目をダイナミッククラス、静的関数をfunction、動的関数をactionと呼ぶ。このようなオブジェクトの直交的分析は、ソフトウェア記述を効率よく進めることができる。ドキュメントは自然言語の箇条書き単文によって記述され、逐次翻訳されて実行に移されるが、表現に関してはandやorといったキーワードが設定されており、重文表現も可能となっている。

仕様部にはオブジェクトが満たすべき制約条件が記述され、ここで用いられるシンタクス、セマンティクスは内包的に定義され、階層構造をなしている。

ミッション部は効率よく並行手続きを導出するために、与えられたスペックを統合する役割を担っている。そのため、ミッション部には以下の三つのカテゴリーが考えられ、実行処理のための順序等に関する制約条件の集合が記述される。

- Configuration: 共有変数の記述、他のオブジェクトからは情報隠ぺいされる
- Order : スペック中のオブジェクトのもつ値の順序指定
- Timing : 仕様動作のための起動、実行中、終了条件の記述

ダイナミッククラスでのドキュメントはミッションを複数もち、ミッションがミッションと呼ぶサブミッションを持たせ、ミッションの階層構造を構築することができる。TELL-SCPにおける並行手続の導出とは、上述した動的なプロセスが記述されるダイナミッククラスのドキュメント中で、仕様動作がさらにある条件(この条件にもまた動的なプロセスが考えられるが)を満たしながら実行されるというような並行的な動作の記述から手続きを導出することである。本論文においては従来のTELL-SCPに新たに推論機能というものを導入して、手続きを導く方法について述べることにする。

3. 推論機能を用いた並行手続の導出

推論機能を用いた並行手続の導出とは、ある処理を記述した文から対象となるオブジェクトの制約条件に基づき、システムがどういった順序でその処理を行っていくかというその手続きを推論することである。この推論過程に関数的な考えが導入されるとき、この推論を行う関数を推論関数と呼ぶ。

順序という動的な要素が処理の実行プロセスにあらわれるとき、TELL-SCPのドキュメント中に実際に記述される文の中では、おもにTiming部に用いられる動詞の役割に注目する必要がある。動詞の扱いという点から考えると、ある動詞を用いてある処理を行う文が記述されたとき、推論するということは、この文中に現れる動詞を総称的(generic)な働きをするものとみなし、動詞の後に現れる表現(前置詞句など)をオブジェクト間の関係をあらわす制約条件として、それに従って一連の具体的な関数(actual function)を導出すること、すなわち複数の手続きに分解して適切な順序で処理を行うことである。実際に実行される具体的な関数はユーザーからまったく隠ぺいされ、システム内部で働くもので、ユーザーは内部のことを全く意識せずにプログラミング作業ができるようになる。このことはWELL-PPPで取り入れられている、generic, actual関数の考えを包含したものであるともいえる。

推論機能の具体的な機能は、妥当性の検証(validity check)である。前述のように実際に手続きが導出されるには、オブジェクト間の関係に基づく制約条件を満たさなくてはならない。このとき本当にそのオブジェクト間

の関係を表現した記述が妥当であるかを検証し、正しければそれらを一連の関数群にして処理、そうでなければユーザー、システム側にその旨を伝えるサービスマクロとしての機能を果たす、といったことが推論機能の具体的な働きである(図2)。

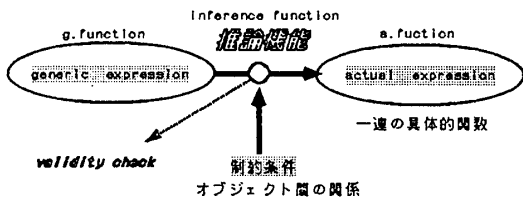


図2. 推論機能の概念

4. 推論機能の具体例

4.1 画像描画処理における推論機能

本節では、まず画像描画を例にとり推論機能を説明する。画像はWELL-PPP上で動く画像描画システム(CGPAW<sup>[31]</sup>)で描かれている。図3にあるようなオブジェクトが与えられ、TELL-SCPのドキュメント中(ミッション部)に"place A into B."と記述されているものとする。このとき、この動詞"place"を総称的関数(generic function)、その後続くオブジェクト間関係を表わす表現を制約条件としてとらえて処理を行う。この文が実行されると図3右下のようになるが、この処理には一番後ろにくるものから順にオブジェクトを乗せる、すなわち前後関係を認識して処理を行うことが求められる。そこでシステムは推論機能によって、Bが、1)サブオブジェクト二つで構成されているか、2)二つのサブオブジェクトには前後関係があるか、ということを検証し適切な順序の手続きを導く。実際には具体的関数"composite"を複数用いることによって処理が行われる。この関数"composite"は、ユーザー側からは隠べいされる関数である。

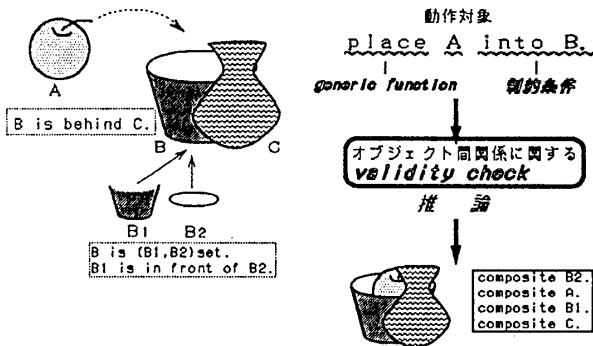


図3. 画像描画処理における推論機能

このような画像関係の処理ではオブジェクトの前後関係を扱う上で、オブジェクト間の関係が変化しないものとの認識も求められる。この例(図3)の場合、処理はオブジェクト単位になされているゆえ、それらを構成するサブオブジェクトには変化はないというこ

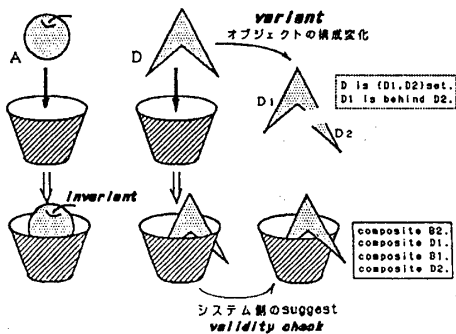


図4. 画像処理におけるオブジェクト構成の変化

とである。この認識は、画像の構成や形状が複雑になったときに特に必要な考え方である。例えば図4のオブジェクトDのような場合、同じ記述を行ってもユーザーの意図する画像が得られない。Dはオブジェクト構成が変化するものとして認識され、サブオブジェクトに分割される必要がある。このオブジェクト構成の変化も推論機能によってユーザー側に提示することもできるであろう。(Dのような図形を考えた場合、前後関係処理の検証事項としてさらに、対象となるオブジェクトの領域に関する検証も加えることとなる。)

4-2. ハフマン符号化問題における推論機能

もう一つの例としてハフマン符号化問題を取り上げる。図5がハフマンツリーのミッションの一例だが、order部に記述された文が仕様を参照しながら処理を行い、図中(\*)を導く。このためには、order部の動詞"construct"を総称的な表現を行う動詞とみなし、先の例と同じ様な一連の具体的関数で表わす必要がある。この場合具体的関数としては、最小値をとるleafを選択する"select"や、その他、実際にnodeを作りpathを付けるといった具体的関数が考えられるであろう。

Huffman tree H(Sleaf) is the binary tree, satisfying H1),H2) and H3)  
 H1) Sleaf = the set of leaves of H.  
 H2) For every x,y and z such that y = the ltree of x and z = rtree of x, value(x) = value(y) + value(z).  
 H3) For every x and y such that x is a subtree of t and y is a subtree of t and value(x) ≤ value(y), length(x) ≤ length(y).

Mission of Huffman tree seeks H using Configuration  
 1) Wstree is a set of binary tree.  
 Order  
 1) Construct the tree having longest length.  
 Timing  
 1) Initially, configuration = Sleaf.  
 2) Construct the tree x by two elements y, z in the Wstree following Order and H3).  
 3) Finally, Wstree becomes a tree.

(\*) The elements have min. value's elements.

図5. ハフマンツリー生成のドキュメント

5. まとめ

推論機能によってもたらされるものを考えてみる。例として取り上げたWELL-PPPのIFO<sup>[4]</sup>の関数サーバーでは、リクエストに応じて関数をレスポンドとして返す等の働きをするが、この関数はすべて貯えられていて負担が重く、新しい関数に応じて新たにサーバーを構築することが必要となる。しかし、この推論機能によってある処理を行う個々の関数を一連の具体的関数の集合と解釈することで、貯えられる関数も整理されてカーネルインターフェイス部の負担を軽くすることができる。推論機能はIFOに組み込まれることで、具体的関数の導出に有効に機能する。またユーザー側から見れば、推論機能によってシステム側の具体的関数を意識せずにかなり自然表記に近い記述が可能となり、情報量を少なく押さえることができる(複数の具体的関数の処理を一つの動詞で表現できる)。そのために、動的プロセスをより明確に、そして階層的に定義できるようにミッション部のConfiguration部での動詞の定義などが求められるであろうと思われる。

文献

[1] 森谷, 村尾, 榎本: "内包的テンプレートによるソフトウェア記述プロセスの実現", 情報処理学会 第42回全国大会, 1991. 3  
 [2] 嶋志田, 丹羽, 榎本: "オブジェクトネットワークによる画像システム記述言語", 情報処理学会 第44回全国大会, 1992. 3  
 [3] 宮本, 関, 嶋志田, 榎本: "前後関係束構造による個別画像のキー抽出描画", 情報処理学会 第44回全国大会, 1992. 3  
 [4] 丹羽, 嶋志田, 榎本: "オブジェクトネットワーク上のインターフェイスとプロトコル", 情報処理学会 第44回全国大会, 1992. 3  
 [5] 榎本 編: "ソフトウェア工学ハンドブック", オーム社, 1986. 4