

知識情報処理向けスクリプト言語処理系

1 E - 3

宮内 信仁

三菱電機株式会社 情報技術総合研究所

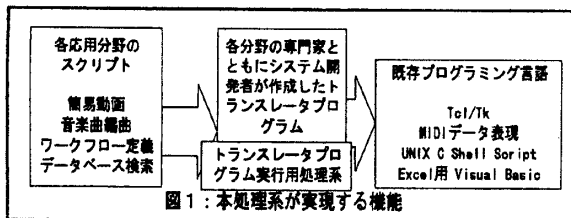
1. はじめに

ソフトウェア生産性技術として多様な研究がなされているが、最近、各専門分野ごとに記述の容易なスクリプト言語の利用が広まっている。これは、応用プログラムの開発が、計算機言語的な要素よりも、各適用業務の知識、経験、規則などに則した記述が重要だからである。このような各適用業務に対応したスクリプト言語の処理系を効率良く開発するために、スクリプト言語から他のプログラミング言語に変換する処理系を、計算機指向のコーディング規則を不要とするプログラミング言語により作成できる処理系開発環境を提案する。

2. 本処理系の果たす役割

本提案では、一般ユーザが使いやすい計算機操作のスクリプト言語で作成したコマンドを従来のプログラミング言語のプログラムに変換するトランスレータを提供する。システム開発者にこのトランスレータの処理記述の作成が容易にできるプログラミング言語を新たに設計した。

従来から、一般的なユーザのためのアプリケーションの作成は、計算機技術の知識をある程度は身につけている技術者によりなされてきた。しかし、ユーザが計算機の世界とは離れた現場の知識を駆使して実現しようとする機能は、直接に計算機指向のプログラミング言語でコーディングするまでもないはずである。本処理系では、知識、規則、経験など



Scripting System for Intellectual Work

Nobuhito Miyauchi

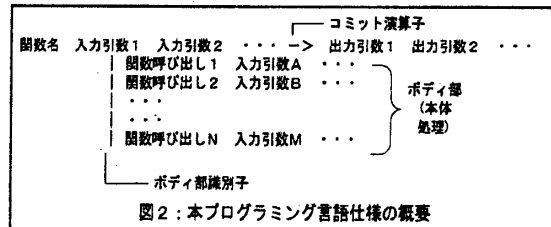
Mitsubishi Electric Corporation

を利用する処理を、論理式に近い形で記述できる。

3. 本プログラミング言語仕様の概要

トランスレータの処理の記述は、論理型プログラミング言語をベースにしたプログラミング言語にて、プログラマが任意に選択した用語を中心に空白文字で区切って並べるだけで行なうことができる。そのプログラム記述も、プログラマが区切り記号の選択を自由に行ない、呼び出し形式に制約を設けないようにして、コーディング規則を低減している。論理型プログラミング言語として、バックトラックは行わないコミットチョイス型を採用した。

必ず使用される特殊記号は、コミット演算子である「->」とボディ部行識別子である「|」の2つだけである。さらに、本プログラミング言語では、変数の意味をもつ文字列と、定数や具体値の意味をもつ文字列とを区別する宣言を行なわない。処理系が矛盾のないように、自動的に判別をして処理を行なう。



4. 代表的な例題プログラム: 動画表示

Tcl/Tk による簡単な動画表示用のスクリプトの作成を行なってみる。Tk には、絵を描くためのウィジェットである canvas が用意されており、これから作成されたユーティリティプログラムとして、canvas 上で行なわれた操作をファイルに記録するものがある。この記録を再生することにより簡単な動画表示が可能になる。例えば、円形を繰り返し数回連続して2次元座標位置を移動しながら表示させる記述例は、以下ようになる。実際に、直接このような記述を行なうと、動画表示が可能であるが、この記述による作成だけでは、負担が大きい。これ

```
{create oval 70 590 90 610 -outline black}
{addtag area withtag 1}
{delete area}
{create oval 71 551 91 571 -outline black}
{addtag area withtag 2}
{delete area}
{create oval 72 514 92 534 -outline black}
{addtag area withtag 3}
{delete area}
:
:
:
```

から、単純に概念的に円形オブジェクトに対する動画効果として指示したいスクリプトとして、「移動の種類 円 始点位置 終点位置 半径 表示コマ番号範囲」のような形態が思いつく。そこで、円形を放物線運動させて、頂点に達してから円形を拡大させる打ち上げ花火のような動画処理用スクリプトを以下のように定義した。

「打ち上げ花火 始点 終点 花火玉半径 炸裂半径 飛行表示コマ数 炸裂表示コマ数」(記述 A)

当スクリプトを実現するトランスレータプログラムを本プログラミング言語で作成したものを示す。

【特徴1】 単語や記号が最も多く一致する記述を選択するコミットチョイスアルゴリズムを採用した。

本処理系では、(記述 A)というスクリプトに対し、登録されているプログラムを調べて、このスクリプトの呼び出しパターンと最も一致するものを捜す。

【特徴2】 組み込み関数(=)での右辺の計算は、変数が全て具体値になるまで保留される。

放物線運動の本体処理行の始めの7行では、ローカル変数の定義として、tagninit はそのまま定数となるが、他は fla に相当する値がこの時点では不定で、計算はされずに式の定義だけがなされる。

【特徴3】 「->」「|」以外に意味のある特殊記号はなく、変数、文字列などの宣言をユーザにさせずに、自動的な置き換えを処理系が可能な限り行う。

同じく8行目には、文字列出力用の組み込み関数が置かれ、この行以降の3行のフォーマットに従って出力を行なう。この文字列出力処理は、fla の値を ka1 から ka2 まで増加させて繰り返される。出力がなされる時点で、fla の値が定まるので、他の座標値も具体的な数値として定まり出力がなされる。

5. プロトタイプの開発

本処理系によるプログラミングの実用性を実証するために、プロトタイプの処理系を Tcl/Tk により

```
打ち上げ花火 (xs,ys) (xg,yg) rmin rmax k1 k2 ->
| 花火玉打ち上げ (xs,ys) (xg,yg) rmin 0 k1
| 花火炸裂 (xg,yg) rmin rmax k1+1 k1+k2+1
```

```
花火玉打ち上げ (xs,ys) (xg,yg) rmin koa1 koa2 ->
| 放物線運動 円 (xs,ys) (xg,yg) rmin koa1 koa2
```

```
花火炸裂 (xg,yg) rmin rmax kob1 kob2 ->
| 拡大運動 円 (xg,yg) rmin rmax kob1 kob2
```

```
放物線運動 円 (xs,ys) (xg,yg) r ka1 ka2 ->
|xt1 =xs+(xg-xs)/(ka2-ka1)*(fla-ka2)
|x1 =xt1-r
|yt1 =yg-(xt1-xs)*(xt1-xs)
|y1 =500-yt1-r
|x2 =xt1+r
|y2 =500-yt1+r
|tagninit =1
|文字列出力 3行 fla(ka1からka2まで)
|{create oval x1 y1 x2 y2 -outline black}
|{addtag area withtag tagninit+fla}
|{delete area}
```

```
拡大運動 円 (xg,yg) rmin rmax kb1 kb2 ->
|r =(rmax-rmin)/(kb2-kb1)*(flb-kb1)+rmin
|xu1 =xg-r
|yu1 =500-yg+r
|xu2 =xg+r
|yu2 =500-yg-r
|tagninit =1
|文字列出力 3行 flb(kb1からkb2まで)
|{create oval xu1yu1xu2yu2 -outline black}
|{addtag area withtag tagninit+flb}
|{delete area}
```

開発した。処理系の適用可能性を評価するために、ワークフロープログラム生成、音楽曲のアレンジデータ生成、データベース検索プログラムなどについて、例題プログラムを作成し動作確認を行った。

6. おわりに

本稿にて提案したプログラミング言語処理システムは、適用業務の知識、経験、規則などに基づいてコーディングが容易に行なえるプログラミング言語により応用プログラムのスクリプト言語の処理系を効率良く開発できることを目指したものである。本提案に基づくスクリプト言語処理系のプロトタイプを実装し、多様なアプリケーションに容易に適用できることが確認できた。現在、人間が認知しやすい形状を適当な用語で指定できる簡易な3次元仮想空間記述スクリプトを設計し、VRMLに変換する処理系への適用の検討を行っている。

参考文献

古川 康一他、「並列論理型言語 GHC とその応用」、共立出版、1987