

7P-1 ターゲットマシンリンクシステムの異ホストコンピュータへの適応性に関する考察

○澤田 健彦*、三上 理*、須堯 一志**

(*日本電気(株) C&Sシステムインタフェース技術本部、 **日本電気技術情報システム開発(株))

1. はじめに

現在のメインフレームコンピュータにおけるソフトウェア開発は、TSS環境、またはバッチ処理環境で行なわれているが、需要の増加にともない、ホストコンピュータの負荷の増大や、回線による遅延のため、ユーザインタフェースは必ずしも良い状態ではない。

報告者らは、このような問題を処理の分散化によって解決しようと試み、ターゲットマシンリンク(以下、TMLと略す)システムを用いた分散開発環境の研究開発を進めてきた[1][2]。本稿では、このTMLシステムの特徴の一つである、ホストコンピュータ依存情報の独立性(異ホストコンピュータへの適応性)に関する機能について報告する。

2. TMLシステムによるソフトウェア開発[1][2]

TMLシステムは、ホストコンピュータとワークステーションを接続し、利用者があたかもワークステーションの環境で操作しているかのようにホストコンピュータを操作することができる、というシステムである。図1に、TMLシステムのシステム概念図を示す。

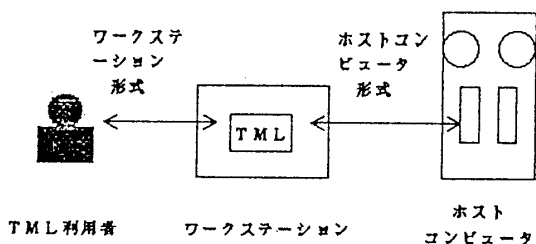


図1 TMLシステム概念図

TMLシステムでは、ワークステーション側のオペレーティングシステムとしてUNIXを採用する。この理由は、

- 1) TMLシステムの利用者として報告者らが現在想定している対象は、ホストコンピュータ上でのソフトウェア開発者である。開発工程としては、コーディング、コンパイル、リンク、実行、及びデバッグ工程を対象としている。
- 2) 一方で、UNIXはソフトウェア開発に適したオペレーティングシステムであり、かつ、既存の多くのワークステーションで利用可能である。

という点である。

ホストコンピュータのオペレーティングシステムとしては、複数のTSS環境を対象とする。

3. ホストコンピュータ依存情報の分離

3.1 TMLシステムのプログラム構成

TMLシステムは、異なるホストコンピュータへの適応性を向上させる点を考慮して設計がなされており、そのプログラムは、ホストコンピュータに依存する部分と、依存しない部分とから構成されている。前者に相当するものは、おもにホストコンピュータのコマンド体系とファイルシステムとに依存する情報が必要となる部分である。図2に、TMLシステムのプログラム構成概念図を示す。

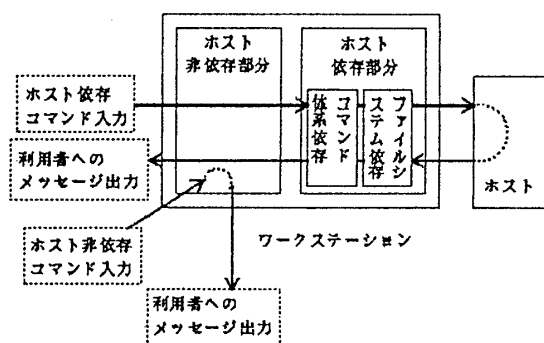


図2 TMLシステムのプログラム構成概念図

ホスト依存部分とホスト非依存部分とが完全に分離されたプログラム構成になっていれば、異ホストコンピュータにTMLシステムを適応させる作業を効率的に行なうことができ、かつ、実現される各TMLシステム間で操作性の統一を図ることができる。

3.2 ホストコンピュータのコマンド体系依存情報

TMLシステムは、1章で述べた目的を実現するため、ホストコンピュータとワークステーションとの通信をできる限り少なくするように設計されている。そのため、利用者がワークステーションに対して行なう入力すべてが、(形式変換後)ホストコンピュータに対して転送されるわけではない。しかし、コンパイル、リンク、実行など、ホストコンピュータの処理に依存するものは必ず、入力コマンドに対応するホストコンピュータのコマンド形式に変換した後、転送されなければならない(図2参照)。そのため、コマンド形式の対応、変換規則の記述が必要となる。

TMLシステムでは、この変換規則をfixと呼ばれるフォーマット変換記述言語[3]を用いて記述する。この記述例(Fortranコンパイラを起動するコマンドの変換規則の記述)の一部を示す。

```
af77(f77):
(MINUS_c)
NCI_set_pr("f")
```

```

NCI_set_com("FORT77")
NCI_set_com(NCI_get_host_name(.args))
NCI_set_com("CULIB=0")
NCI_snd_host()
.....

```

この記述の中で "f77" は UNIX の Fortran コンパイラを起動するコマンドを表わし、"*" はホストコンピュータの TSS でのプロンプトを表わし、"FORT77", "CULIB=0" は同じく TSS での Fortran コンパイラを起動するコマンドとその引数を表わす。また、NCI_xxx は、TML システムで用意されたライブラリ関数である。ライブラリ関数には、ホストコンピュータ側のプロンプト状態のチェック、転送する文字列（コマンド列）の設定、ワークステーション形式名称（ファイル名など）のホストコンピュータ形式名称への変換、ホストコンピュータへのデータの転送などの機能を持ったものが用意されている。ホストコンピュータのコマンド列を生成して転送するための記述は、これらライブラリ関数を用いることによって行なうことができる。

コマンド形式変換のためには、このほかに、利用者が入力するコマンド（文字列）の字句解析部、字句解析の結果を用いた構文解析部が必要であるが、これらは、ホストコンピュータ依存情報を必要としないため、異ホストコンピュータへ TML システムを適応させるときには変更する必要がない。

3.3 ホストコンピュータのファイルシステム依存情報

TML システムには、ワークステーションとホストコンピュータとの間でファイルを共有する機能がある。この機能は、具体的には、ワークステーション上でのファイル名（ディレクトリ名）を指定すれば、それに対応するホストコンピュータ上でのファイル名（ディレクトリ名）を自動的に決定、生成する、という機能である。

この機能を実現するためには、ワークステーションとホストコンピュータ各々のファイルシステム間の対応付けを記述する必要がある。TML システムでは、この情報を簡易に記述する独自の書式が用意されており、システム起動時読み込みファイルにこの記述を行なう。この記述例の一部を示す。

```

chartbl{
/* UNIX code:difference(HOST-UNIX code) */
  0x30-0x39:; /* 0,9 */
  0x41-0x5a:; /* A,Z */
  0x61-0x7a:-0x2d, /* a,z : A,Z */
  .....
}
directory{
/* UNIX file type: HOST file type */
  /directory/file: directory(file),
  /directory/directory: directory
}
.....

```

この記述の中で、chartbl{...} では、ファイル等の名称として使用できる文字コードの、UNIX とホストコンピュータのオペレーティングシステムとの間での違いを吸収するための規則が記述される。例えば、ホストコンピュータのオペレーティングシステムで名

称として英小文字を使用できない場合、UNIX での名称の中に使われている英小文字を英大文字に変換する必要がある。これを定義するため、①で英小文字（コード 0x61 から 0x7a）の各コード値から 0x2d 減じて英大文字に変換する規則が記述されている。

directory{...} では、ファイル等の完全名称を記述する書式の変換規則が記述される。

このような抽象度の高い記述法により、TML システムを異ホストコンピュータに適応させる場合、ホストコンピュータのファイルシステムに関する必要な情報を簡単に記述することができる。

3.4 異ホストコンピュータへの適応性の評価

TML システムの異ホストコンピュータへの適応性を評価するために、報告者らは、ホストコンピュータのオペレーティングシステムとして OS-A を対象とした TML システムを、OS-B に適応させる作業を行なった。その結果、適応のために必要な作業量（書き換えるコード量）は、TML システムの総コード量（fix、lex（字句解析部作成に利用）、C の各言語のソースプログラム行数）の約 14% であった。

4. まとめ

本報告では、TML システムを異ホストコンピュータへ適応させるために提供される機能についての説明と、その評価の報告とを行なった。TML システムでは、この適応性を向上させるために、

- 1) ホストコンピュータのコマンド体系依存情報を記述する fix 記述部の作成を支援するためのライブラリの提供
- 2) ホストコンピュータのファイルシステム依存情報を記述するための簡易な記述規則の提供
- 3) ワークステーションとホストコンピュータとの通信セッションの開設、切断を実現するために必要な情報の記述法の提供

などの機能が用意されている。本報告では、この内の 1)、2) について説明した。

TML システムの適応性を向上させるための今後の課題の一つは、ホストコンピュータのコマンド体系依存情報の記述方法として、まだ記述が煩雑かつ冗長である fix 言語よりも、さらに抽象度をあげた記述規則を提供することである。

TML システム全体としての課題は、適応性向上の他に、ワークステーション、ホスト間の通信手段の改良や、ユーザインタフェース向上のためのウィンドウシステムの導入などがあげられる。

参考文献

- [1] 三上 他：マイクロメインフレームリンクによるソフトウェア開発環境の実現方式，34回情報処全国大会，pp.1171-1172(1987)。
- [2] 三上 他：複数のオペレーティングシステム環境とターゲットマシンリンクシステムの開発，オペレーティングシステム研究会資料，No.37-1(1987)。
- [3] 大竹 他：プロトコルにおけるフォーマット処理記述言語とそのプログラム生成系，35回情報処全国大会，pp.1057-1058(1987)。