移動計算機環境を支援するインターフェース記述言語*

天野憲樹　渡部卓雄†

北陸先端科学技術大学院大学‡

1　はじめに

近年の技術革新は電子装置の小型化、高性能化をもたらし、その製造技術の進歩はそれらの安価な実現を可能にした。またネットワークはパソコン通信などを通じて急速に個人の生活空間に浸透しつつある。小型化された携帯可能な電子装置とネットワーク、この２つの技術の融合により新しいコンピューティング環境、コンピューティングスタイルが生まれようとしている。このように移動可能な小型計算機をネットワークに接続し、分散環境と融合させた計算機環境を移動計算機環境という。移動計算機環境では移動による計算機を取り巻く環境の動的変化、計算機資源の制約等がシステムレベルの透過性実現を困難にしている。本研究ではこのような環境においてシステムレベルの透過性を実現する枠組として有効なメタレベルアーキテクチャで構成される言語を提案する。

2　移動計算機環境における制約

移動計算機環境は従来の計算機環境と比較して計算機の小型化及び無線ネットワークの物理的問題から生じる避け難い制約を内包している。以下はその主なものである [1]。

- 限られた容量 (メモリ、ＨＤ、バッテリ)
- ネットワークの制約 (バンド幅、回線切断の多発)

これらの制約はいずれも移動計算機環境の持つ本質的かつ致命的な問題である。この種の問題の解決はデバイス技術の革新を持ってしても本質的な解決とは言えない。つまり容量に関する問題については無限量を持てる場合を除き、いずれも制約を解消することはできないからである。このような問題についてソフトウエアによって対処するのは自然な流れであり、実際ソフトウエア的にこれらの負荷の幾らかを軽減することは可能である。

3　従来の解決法による問題点

移動計算機環境における様々な制約、問題をシステムコールや汎用ライブラリで対処するのは一見、一般的かつ妥当であるように見える。UNIX では TCP/IP 等の通信プロトコルはＯＳのカーネル内に含まれており、ネットワークに関する様々な機能はＯＳレベルでサポートされる。それらのプロトコルやサポートルーチンに対して機能の拡張あるいは移動計算機環境に特化したカスタマイズを行うことは妥当な対処であると思える。しかし、これには幾つかの問題がある。

1. ＯＳや汎用ライブラリの肥大化と、そこにかかる負荷の増大。
2. 環境の動的対応には環境の種類に応じた数のサポートルーチンが必要になる。
3. 環境に応じたアプリケーション固有の振舞をサポートすることができない。
4. ＯＳのシステムコールや汎用ライブラリで機能を提供した場合、アプリケーションの記述にシステムレベルの記述が混在することになり、プログラムの可読性、保守性、再利用性を低下させる。
5. カスタマイズが容易でない。

4　言語によるアプローチの有効性

従来の解決法による問題点を踏まえ、本研究では言語の観点からアプローチを試みる。言語が移動計算機環境を支援する機能を含むので、ＯＳがそれらの機能のために肥大化することはなくなる。また、言語レベルでサポートされる為、アプリケーションから見たシステムは透過であり、アプリケーションを修正することなく異なる環境で実行することが可能となる。さらに本研究で提案される言語はメタレベルアーキテクチャ[2] として構成される。環境に応じたアプリケーション固有の振舞についてベースレベルで定義し、環境に基づいてベースレベルの内容を選択する制御部分をメタレベルに持たせ、メタレベルの一部をカスタマイズ可能とする。これによってカーネル等を直に触ることなく、異なる環境へのカスタマイズを容易にし、各環境に相応しいアプリケーション固有の振舞 (最適化) を可能にする。また、これはアプリケーション本来の計算とシステムレベルの計算の明確なモジュール化という点においても利点がある。つまりアプリケーションをベースレベル、それを制御するシステムレベルの計算をメタレベルとして捉えることにより、それぞれ独立のカスタマイズが互いに影響を与えることなく可能になる。

5　インターフェース記述言語の機能

本研究で提案する言語 (言語システム) はアプリケーションとＯＳの間のインターフェースとして位置付けられ、インタープリタとして実現される。そして主に以下の特徴的な機能をサポートする。

1. プログラムのリモート実行
2. QoS(Quality of Service) のサポート

---

*Towards a Linguistic Framework for Transparent Mobile Computing Environment

†Noriki AMANO, Takuo WATANABE, {n-amano, takuo}@jaist.ac.jp

‡Graduate School of Information Science,Japan Advanced Institute of Science and Technology

3. 言語の容易なカスタマイズ

これらの機能を備えることでアプリケーションから見たシステムの透過性を実現することが可能となる。

5.1 プログラムのリモート実行

移動計算機環境で使用される計算機はその目的、利用形態からノート型パソコンもしくはＰＤＡのようなものが一般的である。それらの小型計算機は当然デスクトップ機に比べてメモリサイズ、ハードディスクの容量等が厳しく制限される。その為に通常デスクトップの環境で実行可能なプログラムも実行不可能になることが十分に考えられる。そして一般的にＣＰＵの性能もデスクトップ機と比較してかなり低い。そのような環境ではたとえプログラムが動いたとしても極端にレスポンスが悪くなり使用に耐えないものとなることが予想される。またハードディスクの容量が限られるため多くのファイルを持ち歩くことは現実的でない。そのために必要なデータを遠隔地のＤＢやファイルシステムから取り込む必要性が生じる。つまり

1. デスクトップ環境と同等のプログラム実行
2. 遠隔ホストのデータの効率的アクセス

この 2 点を実現する為の有効な手段としてリモート実行の機能 (図 1 参照) をサポートする。実現方法としては移動計算機から負荷のかかるプログラムのソースコードをリモートホストに送信し、リモートホスト上で実行する。そしてその結果を移動計算機側で表示する。これによりアプリケーションプログラムは環境やデータの有無に関わらず実行可能となり、システムの透過性が実現される。 1 について既存の有効な方法としてＲＰＣを思い浮かべることができるが、ここでのリモート実行はＲＰＣよりも高いフレキシビリティを有する。ＲＰＣではリモートホスト上にあるプロシジャの実行に限定されるが、このリモート実行はプログラムのソースコードをリモートに送る為、リモートホスト上にないプログラムを実行することも可能となる。またＲＰＣではプロシジャをクライアント側が明示的に呼び出す必要があるが、この言語では言語システムがその時の環境を認識し必要に応じて自動的にリモート実行を行うことを可能にする。

5.2 QoS(Quality of Service) のサポート

リモート実行は移動計算機環境においてシステムの透過性を実現する有効な方法であるが、負荷のかかる処理をリモートで実行することができても、その結果の表示は移動計算機上で行わなければならない。つまり移動計算機で結果を表示する際に負荷がかかる場合 (例えば画像データの表示等) を如何にクリアするかがここでの問題である。**QoS** とは環境に応じて動的にデータ等の質を上下に移行し、サービスの質を保証することを意味する。本研究で提案する言語は QoS の機能 (図 1 参照) をサポートする。この実現の為にはプログラムをリモートホストに送信する前にローカルホストの環境情報を取得し、その情報もプログラムと併せてリモートホストに送る必要がある。リモートホスト上でそのプログラムが実行される時、ローカルホストの環境に応じた処理をシステムが選択し、その結果として出力される表示データは移動計算機であるローカルホストの表示レベルに合致した質にクオリティが下げられる。表示データのサイズを抑えることで表示に伴う負荷を軽減し、極度のレスポンスの低下を招くことなくプログラムの実行が可能になり、システムレベルの透過性が実現される。

5.3 言語の容易なカスタマイズ

移動計算機環境は使用する計算機、ネットワークの形態等によって一様ではない。さまざまな環境に合わせて容易にカスタマイズすることが可能でなければ先に述べた従来の方法で移動計算機環境をサポートする場合となんら変わるところがない。逆に言語でこのような環境をサポートする機構を提供するが故に容易なカスタマイズが可能となるのである。既に述べたように本研究で提案する言語はアプリケーションとＯＳの間のインターフェースとして位置付けることができ、その構造はメタレベルアーキテクチャとして実現される。ベースレベルはアプリケーションとのインターフェースとして、またメタレベルはＯＳとのインターフェースとしてそれぞれ独立にカスタマイズすることが可能である。より具体的に言えば、リモート実行させたいプログラムはベースレベルにその定義を登録し、メタレベルではＯＳからの環境情報を取得して、どのようなケースの時にベースレベルで定義したプロシジャをリモート実行するかという環境 (制約) のレベルをディスパッチルールとして自由に設定することができる。

図 1: リモート実行と QoS

6 今後の予定

本研究で提案する、移動計算機環境をサポートするインターフェース記述言語は MOP (メタオブジェクトプロトコル) を持った CLOS のような言語を想定しており、RT-Mach 上に実装を予定している。

参考文献

[1] G.H.Forman and J.Zahorjan. The Challenges of Mobile Computing, IEEE Computer, Vol.27, No.4, 1994.

[2] Maes P. Computational Reflection (Ph. D. Thesis), technical Report 87-2, Artificial Intelligence Laboratory, Vrije Universiteit Brussel, 1987.

[3] 保木本 晃弘, 中島 達夫, 渡部 卓雄. 移動計算機環境に適した拡張可能なオペレーティングシステムの構成技術, 第 11 回ソフトウエア科学会全国大会, 1994.