

MKng プロジェクトにおけるアプリケーション環境: 3 Z-1 KMSF-NetTop システムアーキテクチャ†

中澤仁¹ 大越匡² 望月祐洋³ 徳田英幸^{2,3}

¹慶應義塾大学総合政策学部 ²慶應義塾大学環境情報学部 ³慶應義塾大学政策・メディア研究科

1 はじめに

計算機の利用者が物理空間内を移動しながら、計算機上の仮想的作業空間での作業を継続する場合、計算機と共に移動する必要がある。しかし、現実的には利用者のみが移動する状況は頻繁に起こり得る。計算機の利用者のみが物理空間を移動する際、計算機上の仮想的作業空間での作業継続性を得るためには仮想的作業空間に移動機能が必要である。

本研究では、計算機の移動を伴わずに仮想的作業空間での作業の継続性を保証する KMSF-NetTop 環境を実現した。利用者は移動に際し、仮想的作業空間 NetTop の移動機能を利用して、使用する計算機の変更とは無関係に作業を継続できる。

本稿では、KMSF-NetTop 環境構築の基礎となる CODE モデル [1] について概要を述べた上で、KMSF-NetTop 環境について述べる。

2 CODE モデル

CODE モデルは、協調機能を持つコンポーネントオブジェクト (Collaborative Object: 以下 CO と呼ぶ) を組み合わせることによって、複合的オブジェクト (HyperObject: 以下 HO と呼ぶ) の容易な生成を実現するソフトウェアモデルである。本節では CODE モデルの概要を述べる。

2.1 CODE モデルの構成

CODE モデルは、CODE サーバ、CODE クライアント、CODE オブジェクトによって成り立つ分散オブジェクト指向ソフトウェアモデルである。

CODE オブジェクトは性質によって、以下のように分類できる。

ソフトウェア型 CO ボタン、スクロールバーなど一般的なアプリケーションソフトウェアの構成要素と類似したもの。

メディア型 CO 静止画、動画、テキスト、音声などメディア情報のデータソース。

HO ソフトウェア型 CO やメディア CO あるいは HO を組み合わせて生成される複合オブジェクト。

図 1 は 4 つのコンポーネント CO を組み合わせて生成された HO の概念図である。

2.2 オブジェクトの内部構造

ソフトウェア型 CO と HO はオブジェクトモデル、オブジェクトビュー、オブジェクトコントローラから成る (図 1)。またメディア CO はオブジェクトモデルのみから成る。オブジェクトモデルはオブジェクトの SGML 定義記述であり、オブジェクトビューはそれを再生するモジュール群、オブジェクトコントローラはオブジェクト

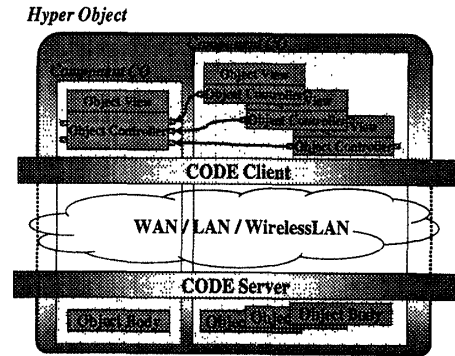


図 1: HyperObject 概念図

が持つ最大 256 個の入出力ハンドラ群およびオブジェクトボディとの同期モジュール群である。

オブジェクトモデルは常に CODE サーバ上に静的に存在する。一つのオブジェクトモデルを複数のオブジェクトビューおよびオブジェクトコントローラが共有することによって、複数の参照者間での共有オブジェクトを容易に生成できる。

入出力ハンドラは、型を持ったモジュールである。ハンドラの型とは入力あるいは出力されるデータの型を指す。複数のオブジェクトを組み合わせて HO を作成する際にハンドラ間を結合し、協調ポリシーを定義する。

3 KMSF-NetTop 環境

計算機、利用者の双方が移動し得るモバイル・コンピューティング環境において、計算機上に存在する仮想的作業空間内での作業の継続性を維持するためには、物理空間における以下の 2 つの状況に対処する必要がある (図 2)。

状況 A 利用者と計算機が共に移動する。

状況 B 利用者のみが移動する。

現在のシステムでは仮想的作業空間と計算機とが 1 対 1 で対応している結果、状況 A においては作業の継続性は保証される。これは利用者が計算機と共に移動することで、作業空間と利用者との透過性が保たれるからである。一方、状況 B においては作業空間を保持する計算機を携帯しないために、作業の継続性が得られない。

本研究では「作業空間」自体に移動機能を持たせた、移動透過型作業空間 (NetTop) を実現した。NetTop を利用することにより、状況 B において別の計算機を使用し始めたとしても、仮想的作業空間をネットワークを通して移動できるので、結果として作業の継続性が得られる。

4 NetTop 動作環境

本研究では、CODE ソフトウェアモデルに基づいて NetTop 環境を設計し、Java 言語を用いたプロトタイプを Solaris2.5.1 上で実装した。本節では、NetTop システムアーキテクチャの構成および特徴について述べる。

4.1 システムの構成

NetTop 環境はクライアント・サーバ型のモデルであり、各利用者の NetTop と 1 対 1 に対応し利用者の位置

Application Environment in the MKng Project:

KMSF-NetTop: its System Architecture

Hitoshi NAKAZAWA¹, Tadashi Okoshi²

Masahiro Mochizuki³ and Hideyuki TOKUDA^{2,3}

¹Department of the Policy Management, Keio University

5322, Endo, Fujisawa, Kanagawa 252, Japan

E-Mail: <jin@sfc.wide.ad.jp>

²Faculty of Environmental Information, Keio University

³Graduate School of Media and Governance, Keio University

†この研究は、情報処理振興事業協会 (IPA) が実施している創造的ソフトウェア育成事業「次世代マイクロカーネル研究プロジェクト」のもとに行われた。

表 1: 関連システムとの機能比較

	NFS	AFS	X Window System	Teleporting	NetTop
保存されたファイル	○	○	×	×	○
インストールされたアプリケーション	○	○	○	×	○
編集中的ファイル	×	×	×	○	○
使用中のアプリケーション	×	×	×	○	○

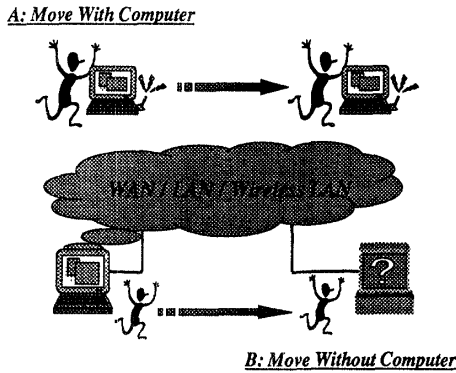


図 2: 作業継続性概念図

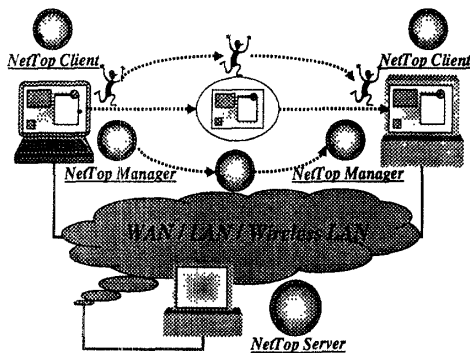


図 3: NetTop システム構成図

情報を管理するマネージャがサーバ・クライアント間を移動することによって仮想的作業空間の移動を実現している。NetTop サーバは利用者の NetTop に関する設定や、利用者が保存した情報などを管理しているサーバプログラムであり、NetTop クライアントは利用者の使用する計算機で、NetTop マネージャを呼び出す際に起動するクライアントプログラムである。

NetTop 環境は CODE モデルに基づいて設計されており、NetTop サーバは CODE サーバの、また NetTop クライアントは CODE クライアントの拡張形である。NetTop 自体は CODE モデルにおける HO として表され、NetTop 上のアイコンや動作中のアプリケーション、利用者が保存した情報などは全て、NetTop を構成するために組み合わせられた CO として実現される。

4.2 移動機能

NetTop には、利用者の要求に応じて別の計算機へ移動する能力が必要となる。利用者が計算機 A から計算機 B へ移動する場合、同時に NetTop を計算機 A から計算機 B へ移動しなければならない (図 3)。利用者と 1 対 1 で対応する NetTop マネージャは、自分がどの計算機上に存

在するかを NetTop サーバに報告している。計算機 B から NetTop を呼び出した場合は、計算機 B 上の NetTop クライアントが計算機 A に存在する NetTop マネージャを移動させ、計算機 A 上での NetTop を再現する。

しかし、計算機 A が常にネットワーク接続を保っているとは限らない点を考慮すると、NetTop 上の全ての操作を NetTop サーバ上の情報へ同期させる必要がある。システムでは、NetTop を CODE モデルにおける共有オブジェクトとして扱うことにより両者の同期は保証される。利用者の NetTop 呼び出しに際して NetTop マネージャが応答しない場合には、NetTop サーバ上に保存された情報から NetTop が再現される。

5 類似システムとの比較

NetTop 環境は仮想的作業空間の移動透過性を実現するが、類似したシステムはいくつか存在する。まず記憶装置に保存された情報に移動透過性を実現するシステムとして NFS[2] や AFS[3] などの分散ファイルシステムが挙げられる。また、特定の計算機にインストールされているアプリケーションを複数の計算機で利用する環境として X Window System が挙げられる。さらに、移動型作業空間を実現しているシステムとしては、Teleporting システム [4] が挙げられる。

これらの関連システムおよび NetTop が実現する移動透過性について比較した結果が表 1 である。保存されたファイルやインストールされたアプリケーションなど、記憶装置に静的に存在するものに関しては既存のシステムでも移動透過性を実現できることがわかるが、編集中的ファイルや使用中のアプリケーションなど、動的にその状態が変化するものまで含めて移動透過性を保証するのは NetTop 環境のみである。

6 おわりに

本稿では、KMSF-NetTop アーキテクチャについて述べた。KMSF-NetTop アーキテクチャは CODE ソフトウェアモデルに基づいて設計、実装され、移動透過型作業環境を提供する。利用者は自分自身の物理的位置に関係なく、計算機さえ存在すれば、NetTop を用いて仮想的作業空間での作業継続性を得られる。

今後の課題としては、耐故障性の確保、NetTop 同期性能の確保などが挙げられる。

参考文献

- [1] 中澤, 岩本, 大越, 永田, “Keio Media Space Family-CODE アーキテクチャ,” コンピュータシステムシンポジウム論文集, (1996).
- [2] Sun Microsystems, Inc., “Network Filesystem Specification”, RFC-1094, DDN Network Information Center, SRI International, Menlo Park, CA, March 1, 1989.
- [3] Morris, J., Satyanarayanan, M., Conner, M.H., Howard, J.H., Rosenthal, D.S., Smith, F.D. (1986), “Andrew: a distributed personal computing environment”, Comms.ACM.
- [4] Frazer Bennett, Tristan Richardson, Andy Harter, “Teleporting - Making Applications Mobile”, Proceedings of 1994 Workshop on Mobile Computing Systems and Applications, December (1994).