

1F-6

ソフトウェア作成支援のための
 ネームベースアーキテクチャとその実現法

神田陽治^ミ 田中英彦 元岡 達

東京大学 工学部

1° 計画 **ORAGA 1** O**R**iented A**R**chitecture to Govern A**B**stractions

2° 目標 計算機をソフトウェア作成保守援助専用^に設計すること

3° 背景 計算機の規模はいかにあるべきか?

「計算機の実現技術の発展は計算能力を向上させた。今後は単なる処理能力向上に止まらず、ソフトウェア作成を含め積極的に処理を支援すべきである。」

「種々のプログラミングスタイルが提唱された。「人間は階層化・抽象化を駆使して複雑な事柄を把握する。」という事実を基本原理としている。各々のスタイルの違いは自然で強力な分割原理にある。分割原理に従って問題を小さいモジュールに分ければ、それは人間が取扱い易い単位となると同時に、計算機にも扱い易いと期待できる。(XX指向アーキテクチャと呼ばれる)。

「計算機の利用拡大は個人が自らプログラムすることを要求する。専門領域への拡大では、解くべき問題の複雑さ・特殊性が他人に仕事を任せざることをひどく難しくするからであり、一般領域への場合、内容が簡単ならプログラムし易い計算機の助けで自らプログラムできるからである。プログラム作成を頼む際の人件費を考えれば個人に一台の計算機でも引き合ふ。負荷の重い仕事はデバッグ後、大型の計算機に送ればよい。」

まとめれば、

「ソフトウェア作成保守向き・個人用・分散環境も考えたXX指向計算機構成」を求めるべきである。

4° 理念 「EPROM」アーキテクチャによる

定義 = ある時ある所で利用できる概念は、いつでもどこでも通用すべきであり、そのための時間・空間全域に渡る一つの大きな論理空間を持つアーキテクチャのことである。従来の「RAM」と区別してこう呼ぶ。

5° 手段 ネームベースアーキテクチャによる

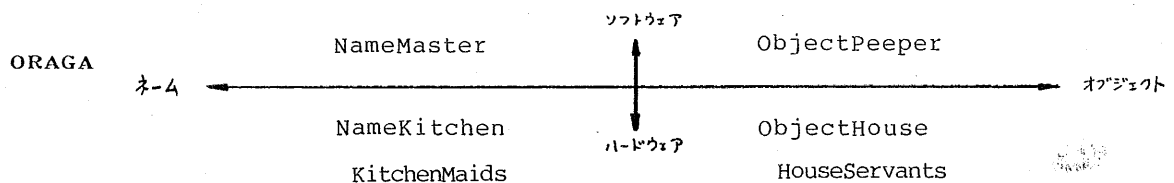
主としてオブジェクトスタイルのプログラミングを「拡張」オブジェクト指向アーキテクチャで支援する。「拡張」とは、

「ケーパビリティ」というアクセス情報をネーム概念へ拡張する。ネームは計算機内で厳重に保護すべき情報の総称であり、ケーパビリティも含まれる。

「シンボリックネーム概念の導入。シンボリックネームは外から見え、計算機内のネームに対応する[3]。例えば Window は、ビットマップ上の仮想スクリーンでありそれ以外ではない情報をユーザに与えるだろう。すなわち適切にシンボリックネームを管理すれば、それは意味と同一性の基準を提供する。」

「一台の計算機を越えて、複数の計算機上で一つの論理空間を維持する。」

6° 構成 全領域に渡るトータル構成で設計する



7° NameKitchen

Name を扱う命令を含むプロセッサアーキテクチャを規定する。Name は Capability, Event, Key の3種を考える[1]。Capability は Object を指し, Event は Object 内の手続き, Key は Object 内のネームスロットを指す。Module は計算の単位であり, そのインスタンスは Context と呼ばれプロセッサへの割当て単位となる。各プロセッサは KitchenMaid と呼ぶ同一構成を持ち[2], NameKitchen はこれら複数台の KitchenMaids からなるマルチプロセッサである。

KitchenMaid への Context の割当ては、実際の起動より早い時期に行なう。(呼ばれ側の) Context が必要とするパラメータは、呼び手側の Context を実行する KitchenMaid から呼ばれ側の Context を保持する KitchenMaid へ、呼び手側が作るそばから送る。呼ばれ側の KitchenMaid は充分パラメータを収集したことを判断し、保持 Context の実行を始める。割当てと起動を分離することにより、割当てのための空き KitchenMaid 探索の時間の吸収、フリキャシュの実行による起動後の実行の高速化を企る。

KitchenMaid は4つの部分へ機能分割される。すなわち、命令・属性・値・補助値のそれぞれに対応する処理装置を持つ[1]。

8° ObjectHouse

Object の所在を管理する sysDirectory と Name のチェーンを管理する sysDictionary からなる[2]。この2つの表は ORAGA-I に一つずつある表であるが、KitchenMaid 単位に分割して保持しアクセス競合を減らすことが望ましい。この分割単位を HouseServant という。ObjectHouse は複数の HouseServant の働きで維持される。

9° NameMaster

SymbolicName と Name との対応を管理する対人インターフェースである。SymbolicName は事物の同一性の判断基準を与える。同一の SymbolicName を持つ物のみか同一の物と判断される。

プログラム中に使われる識別子も SymbolicName である。人間もコンピュータも同じ識別子は同じ物と指すと判断する。人間は識別子を構成する文字列からその意味を推測する。このように SymbolicName は意味を暗示する働きを持つ。NameMaster はより意味を推測し易い SymbolicName 決定の支援システムである。

現在、この基準を求めるために、数多くのソースプログラムでの識別子の付け方を調べる準備を進めている。

10° ObjectPeeper

計算機内の Object を解り易く表示・修正する対人インターフェースである。Object が属すクラスごとに palette と呼ぶ表示型式を作成する。palette はいくつかのフィールドに分かれ、この間に表示指示に関するインヘリタンスがあり、表示型式の指定・変更がたやすくできる。表示は面積の限られたウィンドウに、palette を参考にしつつなるべく効果的に圧縮しておこなう。ObjectPeeper は、palette の指示を解釈実行するインタプリタではなく、それを参照するだけであるので、指示が不十分な palette を持つクラスの Object も解り易く表示できる。

現在、Bitmap Display を用いて製作中である。