

○ Z : 対象指向開放型分散システムアーキテクチャ

-マルチユーザ化-

3T-3

田中伸明 (松下電器) 塚本享治 (電総研)
 近藤貴士 (シャープ) 水谷功 (住友電工) 中込昌吾 (ABC)

1. はじめに

○ Zシステムは、複数の計算機上にあるオブジェクト間でのメッセージパッシングを実現し、よりユーザが簡単に分散システム上でのプログラミングを行えることを目的とした分散システムである。基本構成についてはすでに発表されているが、今回、複数ユーザ間でのメッセージパッシングができるよう拡張を行ったので報告する。

2. タイプオブジェクト

○ Zシステムでは、継承機能を有する抽象データ型定義を表すオブジェクト (SmalltalkのClassオブジェクトに相当) を、タイプオブジェクトと称している。抽象データ型定義は、大きく2種類に分類され、それぞれをクラス、モニタと称している。(クラス、モニタについては、文献 [1] 参照。)

3. 従来システムの構成とオブジェクトのID

従来の○ Zシステムの構成を図1-aに示す。○ Zシステムでは仮想的なマシンを想定し、1つの仮想マシンをドメインと呼ぶ。それぞれのドメインに

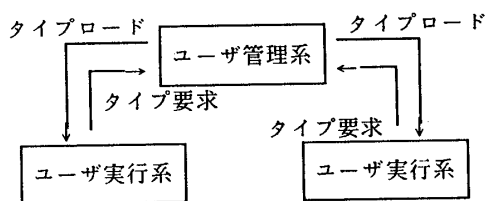


図 1-a 従来のシステム構成

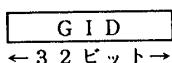


図 1-b GID

は、ドメインidが付与される。このシステムでは、タイプオブジェクト、シンボルオブジェクトは、ユーザ中に唯一の管理ドメインによって生成される。

他ドメインから参照される可能性のあるオブジェクト (グローバルオブジェクトと呼ぶ。) には、図1-bに示すようなID (GIDと呼ぶ。) を付与する。タイプオブジェクトにもGIDは付与される。

○ Zシステムでは、オブジェクト管理に間接テーブル方式を用いており、各ドメイン間でオブジェクトが転送される時はそのオブジェクトのタイプオブジェクトのテーブルエントリのみがいっしょに転送される。そのタイプオブジェクトの本体が必要となった時にフォールトが起こり、管理ドメインに対して、そのタイプオブジェクトの本体のロード要求を行う。このとき、GIDによって、どのタイプオブジェクトをロードするかを指定する。

4. マルチユーザ化における問題点

複数ユーザ間でのメッセージパッシングを行うには、次のことが必要となる。

- 1) 受信側オブジェクトが実行できるメッセージを、送信側が送ること。
- 2) 転送されたオブジェクトへのメッセージを、受信側ドメインが実行できること。

1) は、受信側ユーザが生成したタイプオブジェクトが持つメソッドを示すシンボルオブジェクトを、送信側ドメインが送ればよい。2) は、転送されたオブジェクトのタイプを、受信側ドメインにロードすることが出来ればよい。よって、タイプオブジェクト、シンボルオブジェクトを複数ユーザ間で共有することが必要である。これを実現するため、システム構成を図2-aのように拡張した。従来のシステム構成とは別に、共有されるタイプを管理するドメインを作成した。これをタイプサーバと呼ぶ。

○ Z: Object Oriented Open Distributed System Architecture -- Enhancement for Multi User Environment -- Nobuaki TANAKA⁽¹⁾, Michiharu TSUKAMOTO⁽²⁾, Takashi KONDO⁽³⁾, Isao MIZUTANI⁽⁴⁾, Shogo NAKAGOME⁽⁵⁾

(1)MATSUSHITA Electric Industrial Co.,Ltd. (2)Electrotechnical Laboratory
 (3)SHARP Corporation (4)SUMITOMO Electric Industries, Ltd. (5)ABC Co.,Ltd.

このとき問題となったのは、前章で述べたシステムと違い、タイプオブジェクトのロードを要求する先のドメインが唯一でない、ということである。

5. IDの拡張とロード要求の振り分け

そこで、グローバルオブジェクトが生成されたドメインのドメインidが容易に求められるよう、GUIDを図2-bのように拡張した。これをUID (Unique ID) と呼ぶことにする。

グローバルオブジェクトが生成される時は常にそれを生成したドメインのドメインidがUIDの上位32ビットに付与される。

各ドメインでタイプオブジェクトの本体が必要となった場合は、そのタイプオブジェクトのテーブルエントリからドメインidを求め、そのドメインidを持つドメインに対して、タイプオブジェクト本体のロードを要求する。

6. ブートアップ時のタイプロード

各ドメインを起動するときに必要なタイプオブジェクトは、すべてシステムタイプサーバから獲得することとした。

7. シンボルオブジェクト

タイプオブジェクト中には、シンボルを表すシンボルオブジェクトが含まれる。シンボル名はそれぞれのユーザ、タイプサーバ間で独立であるとした。

例：ユーザaが生成したsym_1という名前のシンボルとユーザbが生成したsym_1という名前のシンボルとタイプサーバcが生成したsym_1という名前のシンボルは、それぞれ別のものである。

他のユーザへのメッセージパッシング、及びタイプサーバが管理するタイプのインスタンス生成時のタイプ名の指定の時に、タイプサーバ中のシンボルを指定する必要が生じた。そこで、タイプサーバで生成されたシンボルを示す場合には、プログラム中でシンボル名を次のようにして表記することとした。

```

srv_name$sym_name
srv_name : サーバ名
$       : 区切り子
sym_name : シンボル名
    
```

8. おわりに

今回の拡張により、タイプオブジェクト、シンボルオブジェクトを複数のユーザ間で共有することが可能となったため、異なるユーザ環境間でオブジェクトを交換することが可能になった。また、タイプオブジェクトの共有によって、システム全体でのタイプオブジェクト数も減少した。

なお、本研究は通産省の大型プロジェクト「電子計算機相互運用データベースシステムの研究開発」で行われたものである

[参考文献]

- 1) 塚本 他, "OZ:対象指向開放型分散システム7-キキチ", 第38回情処全大, pp.1667-1674 (1989)
- 2) 塚本 他, "OZ:オブジェクト指向開放型分散システム7-キキチ -オブジェクト指向型分散プログラミング言語とその実装-", 情報処理学会プログラミング言語研究会資料, 89-PL-21-4 (1989)
- 3) M.Tsukamoto et al, "The Architecture of OZ: Object-Oriented Open Distributed System", Proc. of 2nd ISIIS, pp.153-166 (1988)

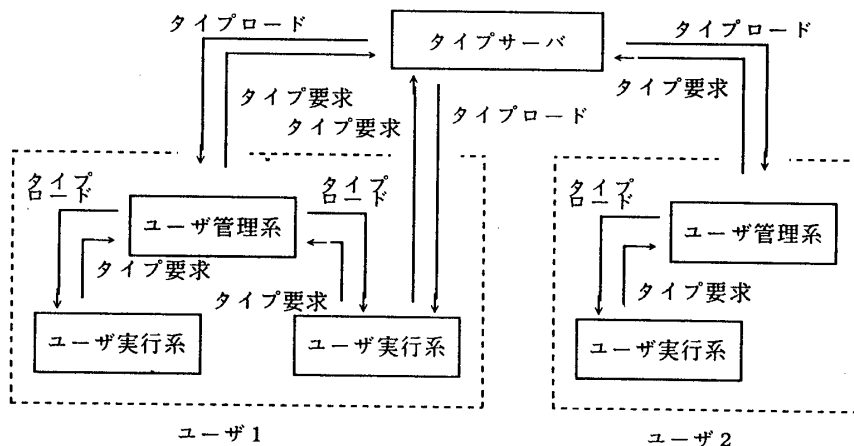


図 2-a マルチユーザ化されたシステム構成

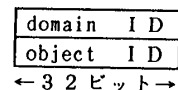


図 2-b UID