

7L-4

オブジェクト指向分散環境OZ++の トレーディングディレクトリの設計

大西 雅夫* 西岡 利博* 吉田 泰光* 塚本 享治
東洋情報システム 三菱総合研究所 日本ユニシス 電子技術総合研究所

*: 開放型基盤ソフトウェアつくば研究室研究員

1 はじめに

ネットワーク上で提供されるサービスを利用する場合、通信の途絶や複製の存在など、分散システムに特有の事情について考慮する必要がある。しかし、この種のプログラミングはしばしば複雑になり、適切な抽象が提供されることが望ましい。

OZ++は、分散オブジェクト指向プログラミング言語であるOZ++言語で記述されたプログラムをネットワーク上で実行できる環境である。OZ++のオブジェクトであるトレーディングディレクトリ（以下TDと呼ぶ）は、OZ++上のサービスを名前検索するサービスを提供する。すなわち、TDの利用の仕組みとして、OZ++上でサービスの利用のフレームワークが構築されている。本稿では、このTDの設計について報告する。

2 TDの設計方針

TDの設計にあたり、さまざまなサービスで共通に使用できるフレームワークとして、以下の要求項目を設定した。

- サービスを名前でも要求できる。
- 分散環境におけるさまざまな特殊性をクライアントから隠蔽できる。

特殊性とは以下のようなことである。

例えば、複製されているサーバは、故障しても、別のコピーからサービスを継続できるが、その際、サーバの切替をクライアントにプログラミングさせるのは避けたい。また、サーバの事故や機能強化があった場合に、クライアントとは独立に更新したい。TDは、このような透過性を提供する。

The Design of the Trading Directory of OZ++:
an Object-oriented Distributed Systems Environment
Masao Onishi* (Toyo Information Systems, Co., Ltd.),
Toshihiro Nishioka* (Mitsubishi Research Institute, Inc.),
Yasumitsu Yoshida* (Nihon Unisys, Ltd.), and
Michiharu Tsukamoto (Electrotechnical Laboratory);
*: Researcher, Tsukuba Laboratory, Open Fundamental
Software Technology Project

3 TDの機能

TDは、ブローカと呼ばれるオブジェクトを、サービス名をキーとする表形式で管理する（図1）。TDは以下の機能を持つ。

- ブローカを、登録、更新、削除する。
- クライアントからの要求に対してアクセススタブを返す。
- ブローカに対し、適切な時間間隔で、保持している情報の更新を求める。

以下の各節では、これらについて述べる。

3.1 利用の仕組み

サーバは、TDにサービス名とブローカを登録する。

サービスを受けようとするクライアントは、TDに対してサービス名と、受けようとするサービスに対する各種の要望（プリファレンスと呼ぶ）を提示することで、個別の要望に沿ったサービスを要求できる。TDは、サービス名に対応するブローカに要求を委譲する。するとブローカはアクセススタブと呼ばれるオブジェクトを返すので、TDはそれをクライアントに返す。以降、クライアントはアクセススタブに対して処理を要求する。アクセススタブはそれを適切なサーバに委譲する。

3.2 アクセススタブの機能

アクセススタブは、サービスの利用にあたって、各種の透過性を提供する。

- 位置透過性
サーバがどこにあらうと、クライアントは手元のアクセススタブに要求を出せばよい。
- 複製透過性
同等のサービスを提供するサーバが複数ある場合、それらのひとつを選ぶ。
- 障害透過性
サーバ側に障害が発生した際に、クライアントに負担をかけずに、サービスを再開する。

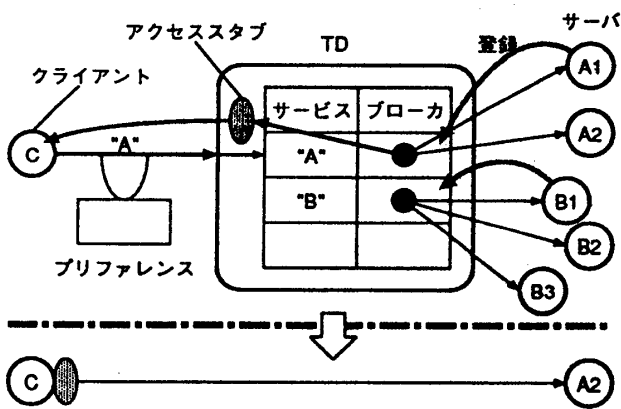


図 1: TD の利用の仕組み

アクセススタブの保持している情報が古くなってしまい、十分に機能しなくなることは考えられる。アクセススタブのプログラミングに際しては、そのような事態がなるべく起こらないことと、万一そうなった場合に、TD から新しいアクセススタブを再取得し、例外を通じてクライアントに引き渡すようにすることを注意しなければならない。この例外は、TD のフレームワークの一部として用意される。

アクセススタブのプログラミングは、サービスの種類に関わらず、複製やトランザクション処理など、共通の要素を持つこととなろう。これらは OZ++ のクラスライブラリとして提供される。

3.3 ブローカの機能

ブローカは、アクセススタブを生成して返すものである。ブローカは、ユーザから指定されたプリファレンスを勘案し、アクセススタブを適切に初期設定して返す。

プリファレンスとは、プリファレンス項目名とプリファレンス値の組の集合である。プリファレンスとして指定される基準には、例えば、ネットワーク的に近い、ロードが低い、など、ある程度各サービスに共通なものや、バージョン 2.2. のプロトコルを理解するネームサービスがよい、隣の部屋のプリンタを使うプリントサーバよりは、同じ部屋のプリンタを使うプリントサーバの方がよい、など、サービスの内容と結びついたものがある。ブローカは、これらの基準を判定する機能を持たねばならないが、そのうち、共通性の高い基準については、それを計算するクラスライブラリが提供される。それ以外の基準については、サービスの種類ごとにサーバの提供者がプログラミングする。

ブローカはアクセススタブに、そのサービスを提供するサーバに関する最新の情報(複製はどれか、そのうち現在サービス中のものはどれか、など)を教える方がよい。それにはブローカ自身、それらの情報を常に最新に保たねばならない。このため、TD は、登録されているブローカをリフレッシュするメソッドを一定時間間隔ごとに呼び出す。TD は、ブローカの登録時に、そのブローカをリフレッシュすべき間隔をブローカに問い合わせしておく。

3.4 TD の配置

OZ++ のオブジェクトは、ドメインという、共通の名称空間を参照しつつ動作する。各ドメインには、一般に複数の TD があるが、そのうちの必ずひとつ、そしてひとつだけが、“trader” という名前でドメインに登録されている。

すべての TD は、そのドメインで管理しているサービスに対するアクセスには、それが例え自身が管理するものでなかったとしても、適切に委譲して応答しなくてはならない。このため、各 TD は、サービスと、それを扱う TD の関係を(高速に検索可能な)表にして保持している。この表に変更があった場合は互いに通知し合い、常に最新に保つ。

TD には、“trade” というサービス名で、TD のブローカが登録されている。クライアントは、ドメインに登録されている TD を得て、その TD から trade のアクセススタブを取得すれば、以降、それを通じて他のすべてのサービスを利用することができ、本稿で述べた利点を楽しむことができる。TD のユーザインタフェースも、このアクセススタブを利用して作ることができる。

4 まとめ

分散システム特有の事情を考慮しつつ、クライアントに適切なサービスを提供するフレームワークとして、OZ++ のトレーディングディレクトリの設計について報告した。

本研究は、情報処理振興事業協会 (IPA) の「開放型基盤ソフトウェア研究開発評価事業」の一環として行なわれたものである。

参考文献

- [1] 西岡他, “オブジェクト指向分散環境 OZ++ の実現”, SWoPP '95 プログラミング研究会にて発表予定, Aug. 1995.