

分散オブジェクト環境におけるXMLを利用した アプリケーション連携

2N-1

福盛秀雄 天野雅志 赤津雅晴

(株) 日立製作所

1. はじめに

CORBA(Common Object Request Broker Architecture)に代表される分散オブジェクト環境を用いたアプリケーションは近年その相互運用性の高さと柔軟性から注目を集めており、実用的業務への適用も徐々に進みつつある。

CORBA では Interface Repository(IR) および Dynamic Invocation Interface (DII)を用いることによりクライアントが動的にインタフェース情報を取得し、利用することが可能となっている。しかし IR/DII は真の動的なアプリケーション連携環境を実現するにはまだ不十分な点も多い。具体的な問題点をあげると以下ようになる：

- IR/DII は何らかのサービスを提供するサーバ型のオブジェクト(アプリケーション)におけるインタフェースをクライアントへ提供するという機能を主眼に置いているが、データの格納や利用をすることに主眼を置いたオブジェクト(データオブジェクト)における格納データの性質などの情報を提供するには機能的に充分ではない、すなわち、
 - IR/DII において取り扱われる Interface Definition Language(IDL)の機能ではインタフェース情報から得られるようなシンタックス(文法)情報よりも上位のレベルにあるセマンティクス(意味)情報を十分に記述することができない

本発表では、この問題に対し分散オブジェクト間においてより高いレベルの動的な連携を実現させるための機構について提案する。本機構はまず CORBA の上位階層のインタフェースとして実装されるものであるが、将来的には他の分散オブジェクト技術への実装も視野に置いている。

オブジェクト間の情報交換は各オブジェクトがXML(eXtensible Markup Language)によりオブジェクト情報(メタデータ)を自己記述させることにより実現する。これにより、クライアントがサービスに対しそのインタフェース情報をセマンティクスを含め動的に取得することが可能となる。

2. XML を利用したオブジェクト情報の記述

分散オブジェクト環境におけるデータの受け渡しにおいては、そのデータ形式だけではなく意味的な情報(セマンティクス)を含め内容に関するより幅広い情報が必要となるケースが起り得る。データに関するこのような情報を一般にメタデータと呼ぶ。特に今後の分散環

境ではあらかじめ設定された取り決めなしにメタデータの動的な取得、利用が要求されることも考えられる。

ここでは、XML によるメタデータの記述機構を提案する。メタデータの記述にXMLを利用することにより、以下のような利点が発生する：

- 記法の柔軟性により幅広いデータ表現が可能であり、さらにユーザによる定義拡張も容易となる。
- 標準として広く利用されつつある規格であり、分散オブジェクト環境における相互運用性をより確実なものとするのが期待できる

本機構はオブジェクト名、インタフェース、データ型の標準的な基本構造とその記法を定義する。ユーザはこの標準的な基本構造の下位の階層へ新たなタグを追加することにより定義を拡張することができる。

ユーザは XML によるメタデータ記述を定義することにより、オブジェクトのインタフェース、オペレーションにおける入出力パラメタの型、さらにそれらオペレーションおよび入出力パラメタの意味情報を外部に対し明らかにすることが可能となる。

3. 自己記述オブジェクトによるアプリケーション連携

3.1 オブジェクト間連携の一例

オブジェクト間で発生するデータ交換において、両オブジェクトのインタフェースが異なる場合は、データ変換のための中間オブジェクトが必要となる場合がある。このようなケースにおいてオブジェクト間で確実な連携を実現するためには受け渡しされるデータについての意味情報と構造情報がきちんと把握されている必要がある。

図1にその一例を示す。本例では2つの口座間にて通貨の変換を伴う受け渡しを取り扱う。図中「通貨変換」という名で示されたオブジェクトが中間オブジェクトとして働く。

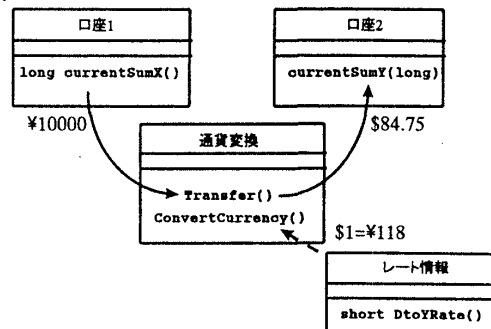


図1. 中間オブジェクトを介したデータ交換の例

本機構に基づくオブジェクトは全て describe()と呼ばれるオペレーションを実装する。このオペレーションはXMLにて記述されたメタデータを呼び出し元へと送る。

CORBA で用いられている Interface Definition Language(IDL)によるデータ表現ではオブジェクト間で受け渡しされるデータはいずれも long 型で表されるのみであり、そのままでは取り扱っている通貨が何であるかを知ることは出来ない。それに対し本例で扱われるメタデータは各オブジェクトについて、それらが持つ属性情報の単位(ここでは通貨単位)が何かを明らかにする(図2)。

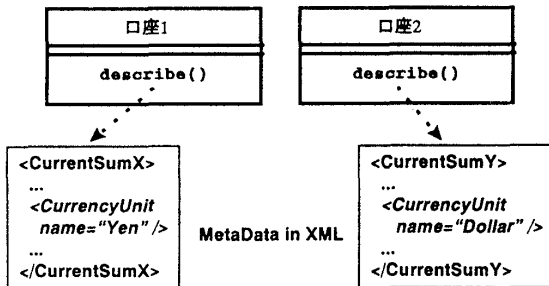


図2. Describe オペレーションによるメタデータの取得

このようにして得られたメタデータを用いて、中間オブジェクトは動的に変換の方法を選択し、実行することが出来るようになる。

多数のオブジェクトからなる大規模なアプリケーション構築においては、動的なサービスを共通のインタフェースで実現する仕組みが今後重要となるものと考えられる。これに対し本機構の提案するメタデータによるオブジェクト記述はXMLを利用することにより相互運用性と柔軟性を大きく高めることを目標としている。

3.2 アプリケーション連携の実現過程

上で述べたような連携を実現する場合においては、実際にはいくつかのレベルに分かれた開発方法が取られることが考えられる。以下おおまかな分類を示す。

レベル1:設計者が連携に必要なすべてのコードを作成する。

レベル2:ジェネリックな連携について、支援ツールを介したコード自動生成機構を用意する

レベル3:既に開発され、利用可能なオブジェクト・コンポーネントをユーザが利用する

レベル4:自動連携機構により、アプリケーションがみずから必要なオブジェクト・コンポーネントを発見し、利用する

理想としてはすべての連携がレベル4のような形式で実現されることが望ましいが、当面のアプリケーション連携は上記のレベルのうち1から3までを組み合わせることにより開発が進められるものと予想される。さらにレベル1~3に基づく連携開発においては、効率的な開発のために適切な支援環境が必要となるものと考えられる。

3.3 アプリケーション連携開発支援

ここでは3.2で述べたレベル1から3までの処理を利用した開発を支援するための環境について述べる。この支援環境は以下の機能を持つ:

- オブジェクト関連付け/データ変換コード生成支援
 - メタデータは実行時連携においてだけでなく、連携機構の設計においても利用することが可能である。支援環境は既存のオブジェクトについて、describe()オペレーションで得られるメタデータを支援環境内で随時参照する機構を備える。(図3)
 - XMLの特徴である人間可読性と機械可読性の両立を活用することにより、手動によるインタフェース関連付け、半自動のインタフェース関連付け、自動化されたインタフェース関連付け等の操作を実現する。
 - インターフェース間の関連付け情報をもとに連携を実現させるためのコードスケルトン、あるいは実装を含むコードを生成する。
- オブジェクトのメタデータ記述の作成支援
 - 本機構の提供するメタデータ定義の拡張による、ユーザによるメタデータ定義の生成、さらにオブジェクトごとのメタデータ記述の作成を支援する。

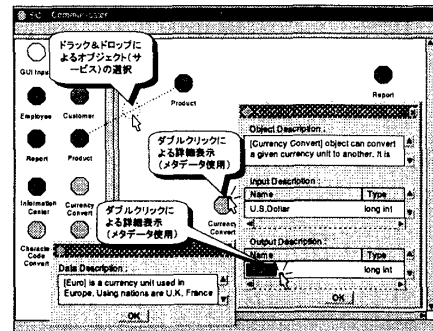


図3. アプリケーション連携開発支援環境

4. まとめ

本稿では、分散オブジェクト環境においてより高度なアプリケーション連携を実現するための機構について提案した。現在基礎部分の実装は終了しており、連携開発支援環境の開発を行っている。

また今後の標準化へ向け、以下の作業を進めている:

- 規格の詳細の検討
- OMG(Object Management Group)への標準化提案に向けた準備

5. 参考文献

- Reaz Hoque, "CORBA3" pp.237-293, IDG Books, 1998
- "The Common Object Request Broker: Architecture and specification Revision 2.2" pp.127-166, Object Management Group (<http://www.omg.org/>), 1998
- Extensible Markup Language (XML) 1.0 (REC-xml-19980210), World Wide Web Consortium, <http://www.w3.org/TR/REC-xml>, 1998