

OSI TP/RDAを利用したアプリケーションの設計
(2) アプリケーションプログラムインタフェース

7V-2

阿比留 巖[†] 関根 徹[†] 成田 秀明[†] 本田 邦夫[†] 杉本 浩司[‡]
[†]松下電器産業(株) [‡](株)松下ソフトリサーチ

1. はじめに

我々はOSI TP,RDAを使用し、X/Openの提唱する分散トランザクションモデルに基づいたシステムの設計を行っている¹⁾。そのシステムでは、DTPEの機能をAPが利用するためのインタフェースであるAPIを提供する。一般的に、APIはAPがOSIの知識を必要とせず、容易にAPを作成できるようなものが望ましい。しかしながら、一方ではOSIのサービスを提供するPMのテストやOSIのサービスを全て使用したいというAPのために、OSIのサービスを直接扱えるようなAPIも必要となる。我々は、これらを考慮に入れAPIの設計を行った。

2. APIの構造

上記2つの要求を満たすため、図1に示すようにAPIは3つのコンポーネントに分割した。RM-APIは、設計が終了していないためここでは省略する。残る2つは、OSIのサービスを全て利用できる基本APIと、APが容易にOSIのサービスを利用できるための拡張APIである。APはその要求により、2つのうちのいずれかを使用することができる。図1に示すように、基本APIは、OSI TP,RDAのサービスを扱うものなのでCRMのみを利用する。拡張APIは、基本APIを通してCRMを利用し、またトランザクションを管理するTMは直接利用する。

3. 基本API

基本APIは、OSIのサービスを全て使用するためのインタフェースである。しかしながら、APがなるべく容易にOSIのサービスを利用できるように考慮した。以下に基本APIの特徴、機能について説明する。

(1) サービスプリミティブ対応インタフェース

APがOSIのサービスを全て使用できるようにするため、基本APIはOSIサービスプリミティブと1対1に対応するインタフェース関数を提供する。現在、基本APIが提供するOSIサービスはTP,RDA, ACSE, UDTであり、これらのサービスプリミティブの1つずつに対応した関数を提供する。サービスプリミティブを発行するには、その基本APIのインタフェース関数を実行する。サービスを受信する方法として、基本APIではイベント駆動型を採用した。つまり、サービスを受信するときには、あるAPの関数が基本APIから非同期に呼ばれる。このAPIから呼ばれる関数は、あらかじめAPが基本APIに対して登録しておく。この関数もOSIのサービスプリミティブと1対1に対応している。図2にこのインタフェースを図示する。

(2) パラメータ補完

基本APIのインタフェースのパラメータもOSIのサービスプリミティブのパラメータと基本的には1対1に対応しており、全てのパラメータを設定したり、値を参照することが出来る。しかし、OSIの

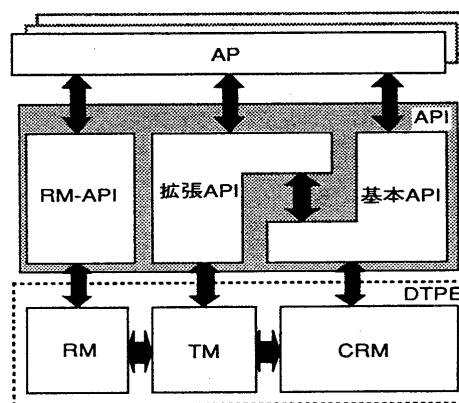


図1 APIの構造

サービスを発行するとき、設定すべきパラメータが多い場合があるので、全てのパラメータを関数の引数として渡すようなインタフェースは適当ではない。そこで必要最小限のパラメータをAPが関数の引数で指定し、その他のパラメータは基本APIが補完することにした。補完の方法は、基本APIがパラメータ補完データの値を参照し補完する。基本APIはAPがパラメータ補完データの値を設定するためのインタフェースを提供しているため、間接的にAPがパラメータを設定できる。また、パラメータ補完データはパラメータ補完ファイルで初期化されるので、必ずしも全てのパラメータをAPが設定する必要はない。また、サービスを受信する場合には、パラメータ抽出データを利用する。基本APIは、必要最小限のパラメータのみ関数の引数でAPに渡し、その他のパラメータはパラメータ抽出データに格納する。APIは、パラメータ抽出データを参照することで全てのパラメータを得ることができる。なお、このパラメータ補完データとパラメータ抽出データはAEI毎に持っている。

また、APが設定せずに、APIが独自に値を設定しても、OSIのサービスを全て使用することに問題ないパラメータもある。このようなパラメータは基本APIが管理し、APは設定しないことで、APの負担を軽くしている。

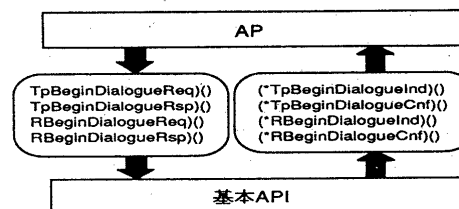


図2 基本APIのインタフェース

Designing of the application for using TP and RDA of OSI

(2) Application Program Interface

Iwao ABIRU[†], Toru SEKINE[†] Hideaki NARITA[†], Kunio HONDA[†], Kohji SUGIMOTO[‡]

[†]Matsushita Electric Industrial Co., Ltd., [‡]Matsushita Soft-Research, Inc.

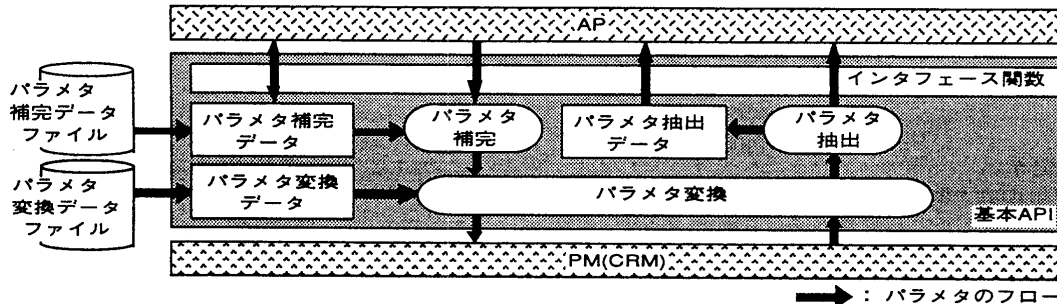


図3 基本API中のパラメータのフロー

(3) パラメータ変換

OSIのパラメータにはAPが直接、扱い難いものがある。例えば、OSI応用層では、通信する相手を識別するのにAEタイトルを使用する。しかしながら、AEタイトルはオブジェクトIDなどで構成されるが、APが直感的に相手を識別できるものではなく、また固定的に割り当てられているものでもある。そこで、APからは文字列で通信相手を指定し、基本APIが文字列からAEタイトルに変換する。この変換により、このような扱い難いパラメータをAPが容易に扱うことが出来る。逆にサービスを受信する場合には、この逆の変換が行われる。このような変換をパラメータ変換と呼ぶ。この変換にはパラメータ変換データが使用され、固定的に変換される。パラメータ変換データはパラメータ変換データファイルの基本APIが読み込むことにより設定される。変換されるパラメータには、この他に应用コンテキストやRDA SQLで使用するSQL適合性レベルなどがある。(2)と(3)で述べたパラメータの補完、変換による基本API中のパラメータのフローを図3に示す。

(4) 应用コンテキスト

基本APIは、OSI TPとRDAを利用するための機能を提供する。現在、これらを使用する应用コンテキストのうち基本APIがサポートするのはTP+UDT, RDA, RDA+TPの3つである。基本APIは、ソースファイルやオブジェクトの管理の面から、1つの基本APIでこの3つの应用コンテキストを同時にサポートすることにした。そのためAPは基本APIを初期化するときに应用コンテキストを指定する。

4. 拡張API

拡張APIには、X/OpenのCM, TXインタフェースを採用した。CM, TXインタフェースは、現在、X/OpenがDTPのAPIとして検討しているものである。これを採用した理由は、システムがX/Openの分散トランザクション処理モデルに基づいていることや、独自のインタフェースを規定するのには比べ、業界標準を採用することでAPの移植性が高くなることを考慮したためである。

TXインタフェースはトランザクション管理を行うためのインタフェースであるが、TMも同様なインタフェースを提供し、拡張APIの処理はそれへの単純なマッピングだけなので拡張APIのTXインタフェースの部分については省略する。以下に、CMインタフェースと基本APIのインタフェースの変換のため、拡張APIが行う処理を示す。

(1) 受信サービスのキューイング

CMインタフェースはイベント駆動型のインタフェースではない。例えば、APはデータを受け取る時に、CMインタフェースのポーリング関数によりデータが通信相手から来ているかどうかを知り、データ受信関数でデータを受け取る。一方、基本APIはイベント駆動型のインタフェースであるため、非同期に拡張APIの

サービス受信関数を呼び出す。このため、APがCMインタフェースを用いてポーリングを行い、APがCMインタフェースのデータ受信関数を実行するまで、サービスの種別やパラメータを拡張API内部にキューイングすることが必要となる。

(2) 受信待ち機構

CMインタフェースは、1つのインタフェース関数で、複数のOSIのサービスプリミティブの発行、受信に対応するものがある。また、CMインタフェースは、インタフェース関数を非ブロッキングで呼び出すことをサポートしている。つまり、その関数の処理が終らないうちに関数を終了することが出来る。その関数の終了と同期を取るには、CMインタフェースが提供する同期関数を利用する。拡張APIは、これらを実現するために、基本APIがサービス受信のための拡張APIの関数を呼び出すのを待つ機構を持つ。

(3) 関数マッピング

CMインタフェースはOSI TPを強く意識しており、OSI TPサービスに類似しているが、1対1には対応していないので、マッピングが必要となる。また、CMインタフェースはデータの転送機能も提供しているが、OSI TPはそれ自体ではデータの転送サービスを提供しない。現在の拡張APIでは、これらをUDTのデータ転送にマッピングしている。

(4) パラメータマッピング

CMインタフェースのパラメータ(関数の引数、関数の返り値、ダイアログ属性値)と基本APIのインタフェースのパラメータ(関数の引数、パラメータ補完データ、パラメータ抽出データ)の変換を行う。

5. おわりに

本稿では、OSI TP/RDAシステムのAPIの設計について報告した。OSI機能をあまり意識せず利用できるAPIとOSIサービスを全て使用できるAPIの2つに分けることで、様々なAPの要求を満たすことができる。今回は触れなかったが、SQL Access GroupはSQL CLI (SQL Call Level Interface)を規定しており、このインタフェースをRMのAPIとして使用することも考えられる。また、基本APIをRPC(Remote Procedure Call)などネットワークを利用するインタフェースに変更することによりAPとDTPEの動作するマシンを分け、クライアント/サーバモデルとすることも可能である。今後は、設計したAPIの実装し、実システムで利用して行くと共に、APIの標準化動向を取り入れ、さらに拡張検討を行う予定である。

[参考文献]

- [1] 関根他, 「OSI TP/RDAを利用したアプリケーションの設計(1) システム概要」, 情処第45回全国大会