

オブジェクト指向分散環境OZ++の WWW/CGI インタフェースの設計と実装

3M-5

西岡 利博* 新部 裕* 塚本 享治
三菱総合研究所 三菱総合研究所 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

WWW (World Wide Web)[5] の登場によって、インターネット上で共有または交換される情報の質と量、およびそれらにアクセスするユーザ数は飛躍的に増大している。これは、インターネット上で動作するアプリケーションは、WWW から利用できるインタフェースを整えることによって、専用のクライアントプログラムを配布することなくこれらのユーザ層からの利用に供することができることを意味している。

OZ++ [1] は、ネットワーク上でソフトウェアを自動配送することによりソフトウェアの共有を促進するように設計された分散システムであるが、OZ++ システムで配送されるアプリケーション自体もまたネットワークアプリケーションである場合が多く、それらが共通に使えるネットワーク対応型ユーザインタフェースライブラリの需要は高い。

そこで今回、WWW の CGI (Common Gateway Interface) [2] を利用することにより、WWW から OZ++ アプリケーションを利用するフレームワークを設計し、実装した。本稿では、その概要と、この作業を通じて気づいた、CGI 利用における利点と問題点について述べる。

2 WWW と CGI

WWW はネットワーク上にハイパーメディア(他のハイパーメディアに対するポインタ(ハイパーリンク)を持つようなメディア)形式で蓄積された情報を検索し、閲覧するためのクライアントサーバ型システムである。情報は HTML (HyperText Markup Language)[4]で記述されたテキストであり、URL (Uniform Resource Locator) と呼ばれる階層的な

The design and the implementation of a WWW/CGI interface of OZ++: an Object-oriented Distributed Systems Environment

Toshihiro Nishioka* (Mitsubishi Research Institute, Inc.),
Yutaka Niibe* (Mitsubishi Research Institute, Inc.),
and Michiharu Tsukamoto (Electrotechnical Laboratory);

*: Researcher, Tsukuba Laboratory, Open Fundamental Software Technology Project

称¹で識別される。WWW クライアントが WWW サーバに URL を提示すると、サーバはその URL に対応する情報を取り出し、クライアントに与える。

多くの URL は WWW サーバが稼働しているマシン上のパス名に対応しており、サーバは、単にそのパスにある HTML で記述されたテキストファイルを取り出して返す。しかしクライアントにとっては、URL に対して HTML テキストが返ってくることだけが重要であり、それがファイルとして存在する必要はない。逆に、そのときどきの最新の情報を提供したい場合(株式市況など)や、既存の大規模データベースなどからの情報を提供したい場合など、ファイルとして置くよりも、何らかのプログラムを経由したいことも多い。このために CGI という規格がある。多くの WWW サーバでは、ある種の URL に対しては、ファイルの中身を返すのではなく、CGI 経由で他のプログラムを起動し、その出力を返すように設定できる。

3 OZCGI の設計と実装

WWW クライアントを利用した OZ++ アプリケーションを容易に実現できるように、以下の方針で OZCGI を設計した。

- OZ++ アプリケーションから OZCGI に接続して、WWW クライアントからのそのアプリケーションに対する要求を受け付けられる。
- 複数のアプリケーションが同時にひとつの OZCGI を利用できる。
- OZ++ プログラムから容易に利用できるように、OZ++ のオブジェクトとして実現する。

以上から、図1のような構成を実装した。OZ++ アプリケーションへの要求である URL には、fill-out form 形式[3]の引数(属性-値対のリスト)が付加されており、その引数の一部として“フォーム ID”が付加されているものとする。フォーム ID は、複数の OZ++ アプリケーションのどれに対する要求かを識別するのに用い

¹参考文献の項に URL の例がある。

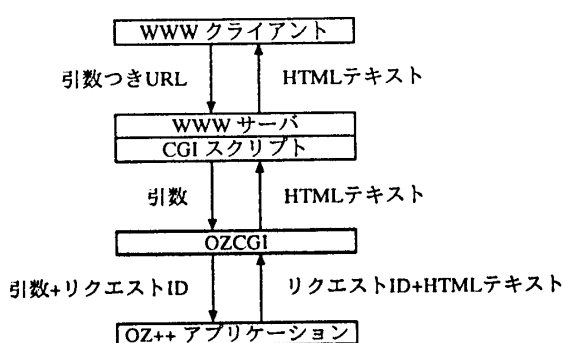


図 1: OZCGI

られる。CGI では WWW サーバ自身が他のサーバと通信することはできないので、そのための CGI スクリプトを用意した。このスクリプトはデータ変換はしない。OZ++ アプリケーションは、あらかじめ、フォーム ID を引数として OZCGI のメソッドを起動しておく。メソッドの実行は OZCGI の内部でブロックされ、フォーム ID の一致する要求が WWW サーバから送られてくると、メソッドの戻り値としてその引数が返される。OZCGI から OZ++ アプリケーションにアクセスするのでなく、逆にしたのは、OZCGI 自体は OZ++ のネームサーバに登録されているサーバであるのに対し、アプリケーションの側は必ずしもそうとは限らないからである。

返信される HTML テキストは、OZ++ アプリケーションが作成したものが、そのまま WWW クライアントにまで達する。このとき、正しくその HTML を要求した WWW クライアントに返信できなければならない。OZCGI では、URL に指定された引数にリクエスト ID という整理番号を付加してから OZ++ アプリケーションに引き渡す。OZ++ アプリケーションは HTML テキストを返信するメソッドの引数にリクエスト ID を含めなくてはならない。これによりクライアントとの対応を取ることができる。

返信される HTML テキストには他の OZ++ アプリケーションや、同じアプリケーションの別の画面への URL を入れられるので、単に情報の表示に使うだけでなく、一般の OZ++ アプリケーションのユーザインタフェース画面として、WWW クライアントの表示内容を自在に制御しつつ処理を進めることができる。

4 まとめ

OZCGI は、OZ++ アプリケーションに WWW からの利用の可能性を提供した点で、OZ++ プログラマにとっては意味のあるものである。

OZCGI を用いて、これまでにふたつのアプリケーションが作成された。そのひとつのアンケートシステムでは、マルチフォームのアンケートの実現に OZCGI の機能が適していることが分かった。回答者の回答によって質問の内容を変えるようなアンケートでは、複数のフォームの表示内容を制御できると便利だが、フォームからの回答を互いに識別できないと、どの回答が同じクライアントからのものなのか分からなくなってしまう。このため、回答者に送信したフォームに ID を付与し、回答についているフォーム ID にしたがってディスパッチするような機能が必要だが、OZCGI では既にそれが用意されているので、アプリケーションはディスパッチされた後の処理だけをすればよい。

今後の課題として、以下の二点を検討している。

● セッションの導入

現状では、リクエスト ID の管理を誤るとクライアントとの対応がとれなくなってしまう危険がある。単純なバグによるこのような危険を防ぐために、クライアントとの間に仮想的なセッションを確立し、アプリケーションからはリクエスト ID を気にすることなく記述できるようなフレームワークを提供することが課題である。

● UI 統合化

OZ++ アプリケーションのためのインタフェースフレームワークとしては、OZCGI の他に、Tcl/Tk の利用を前提とした GUI クラスライブラリと、外部プロセスとの通信を抽象化したクラスライブラリがある。これらはそれぞれ設計の方針も使い方も異なるので、どれを採用するかによってアプリケーションの設計を変えなくてはならない。これらのフレームワークの抽象化と統合が課題である。

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, Aug. 1995.
- [2] <http://www.w3.org/hypertext/WWW/CGI>
- [3] <http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-forms/overview.html>
- [4] <http://www.boutell.com/faq/sgmlhtml.htm>
- [5] <http://www.w3.org/hypertext/WWW/WWW>