

備を指摘されて記入し直す場面も多い。これらを自動的に記入したりチェックする機能があれば便利である。

このような窓口の意味情報とフォームの記入支援機能の定義からなる窓口のメタデータを定義するために、自動処理に適した XML ベースの記述言語を設計した。画面の表示レイアウト定義には HTML(または XHTML) を用い、窓口の検索やフォーム記入支援の定義にこのメタデータ記述言語を用いる。またこれらのメタデータを管理するエージェント間の通信言語として XML ベースの ACL を設計した。

2.1 例題システムのアーキテクチャ

このようなアプリケーションフレームワークを実現するために、まず、研究室内の名簿管理業務を取り上げ、その一連の窓口業務を自動化した名簿管理システムを開発した。例題システムの構成を図 1 に示す。図の各サブシステムは以下のようなものである。

- (a) WWW ブラウザ 本システムと窓口利用者との対話に用いる一般の WWW ブラウザ。
- (b) wwHww クライアント フォームの取寄せ、自動記入、記入チェック、提出を行う代理サーバ。ユーザエージェントは、窓口/フォームの検索を行うインタフェースと自動記入用の個人情報を管理する。フォームエージェントは、記入ナビゲーション機能(自動記入、記入チェック)を実行する。
- (c) wwHww サーバ群 ブローカエージェントは、窓口/フォームのディレクトリ情報を管理し、ユーザエージェントとフォームエージェントを仲介する。フォームエージェントは XML 文書形式で WWW サーバ上で公開する。
- (d) 受付窓口 WWW サーバ 名簿管理システムの受付窓口サーバは、記入フォームを定義する HTML 文書と、記入済文書を受け付ける CGI プログラムからなる一般的な WWW サーバ。

2.2 フレームワークの 2 種類のカスタマイズ方式

アプリケーション開発者は、個々のアプリケーションに固有の処理をホットスポットに定義することで、フレームワークをカスタマイズしてアプリケーションを構築する。フレームワークのカスタマイズ方式は以下の 2 種類に分類できる。

- (A) メタデータの設定 アプリケーションに固有の処理を定義するために、固有のプロパティデータをフレームワークの指定した形式で用意する。
- (B) 差分コンポーネントのプラグイン アプリケーションに固有の機能をフレームワークの指定したインタフェースに沿って用意し、フレームワークにプラグインする。

(A) の方式は業務の専門家による構築に適しているが、この方式のみでは適用範囲がフレームワークで用意された機能に限定され、柔軟性に欠ける。(B) でプラグインする差分コンポーネントは、基本的にアプリケーション開発者が既存のコンポーネントから個々の要求に適したコンポーネントを探し出すことになる。アプリケーションに適したコンポーネントが見つからない場合は、新たに開発する必要がある。

本システムの場合は、業務の専門家がカスタマイズすべきホットスポットは以下のようなものである。

A. メタデータの設定

- (1) 窓口情報の定義
- (2) 電子フォーム定義

B. 差分コンポーネントのプラグイン

- (1) 電子フォーム部品
- (2) 提出書類の処理コンポーネント

A(1) について、ホスト名やファイル名、URL など稼働環境に依存する情報は、メタデータとして用意すべきである。A(2) について、ユーザインタフェース (UI) は個々のアプリケーションに固有であり、また仕様の変更が多い。メタデータとして用意することで、コンパイル作業が不要になりカスタマイズが容易になる。B について、UI 部品やデータ層との接続コンポーネントは個々のアプリケーションで利用する組み合わせが異なるが、各々のコンポーネントは他の業務アプリケーションにも共通に利用できる。したがって、これらのコンポーネントは本来あらかじめ用意しておくべきであると言える。

3. ま と め

一般にホットスポットを多く設定したフレームワークは適用範囲が広がるがアプリケーション開発者の負担が増大する。ホットスポットを限定すると、開発者の負担は少なくなるが適用範囲が限られる。また、コンポーネント開発はプログラミングが必要となり開発者の負担が大きいのに対し、メタデータ方式は構築や変更が容易である。本研究の目的であるエンドユーザ主導開発においては、ホットスポットを限定したメタデータによるカスタマイズ中心の開発手法が有効である。

参 考 文 献

- 1) T.Chusho and K.Fujiwara: FACL : A Form-based Agent Communication Language for Enduser-Initiative Agent-Based Application Development, COMPSAC 2000, IEEE Computer Society, 139-148 (2000).