

異機種分散環境における UIMS の構成

5 G-6

岩崎 未知

日本電気(株) C&C システム研究所

1 はじめに

ワークステーション(WS)や高性能なパーソナルコンピュータの普及と共に、アプリケーション・プログラムに対してビットマップ・ディスプレイやマウスを使ったグラフィカルなユーザ・インタフェース(GUI)を用意することが一般的になりつつある。一方、そのようなUIの実現が困難なハードウェア上でUIの品質に無頓着に開発されたソフトウェアも「資産」として数多く存在し、システムとして今後も利用され続けて行くであろうと思われる。そこで、GUIを備えたマシンからネットワークを通してGUIを持たないマシンを利用する形態が増加し、計算機システムがUI担当マシン(UIサーバ)と各種アプリケーション(AP)本来の機能を実現するマシン(APサーバ)へと分化していくことが予想される。

この様な異機種分散環境でのユーザ・インタフェース・マネージメント・システム(UIMS)構築においては、アプリケーション・インタフェース(API)とセマンティック・フィードバックの実現手法が問題となる。本稿では、これらを実現する手法として現在試作中のUIMSで用いている枠組みを説明し、また、その枠組のUIカスタマイズ手段としての効用について述べる。

2 フィードバック実現上の問題点

WS等の上にダイレクト・マニピュレーションを用いたUIを構築しようとする場合、セマンティック・フィードバックを如何に実現するかが重要な問題となる[1]。

ユーザの操作に対するセマンティック・フィードバックの実現に必要な制約情報(セマンティック・コンストレイント)は通常AP部のみが知っており、UI部はその情報なしにはフィードバックを実現できない。緻密なフィードバックを実現しようとする場合、AP部がフィードバックを実現する枠組では、情報交換の頻度と速度を高くする必要がある。また、APがUIに深く立ち入らなければならず両者の依存関係が強くなってしまい、UIをAPから独立させるというUIMS本来の目的の達成が困難となりがちである。これに対してUI部でフィードバックを実現する枠組では、必要な制約情報を如何にしてAP部から受け取るかが問題となる。その他、AP部に存在する情報の一部をUI部で重複して保持することにより、データ転送のコストと複数のデータ間の一貫性維持の問題が発生する。以下では、これらの問題を回避するための枠組みについて説明する。

3 分散環境での UIMS の枠組み

3.1 構造

概観 全体は、[図1]の構造をとる。2つのAPI部の

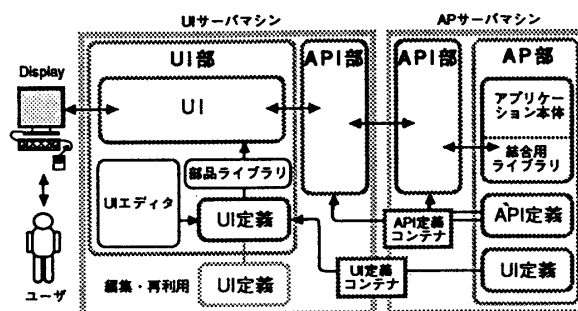


図1: 全体の構造

間の通信部分の構造は対称であり、相互にメッセージを送り合うことにより対等に動作する。UI部とAP部は並列に動作し、両者の間には同期メカニズムを備える。UI部内部は並列性を備え、特定のAPの処理に関連して対応する部分が「待ち」に入った場合も、UI部内の他の部分は停止しない。

インタフェース定義コンテナ アプリケーション起動時に、UI部とAPI部はそれぞれUI定義とAPI定義をAP部からダウンロードする。この時の形態をコンテナと呼ぶことにする。この時使用するプロトコルは、機種に依存しないよう予め定めたものを使用する。UI部は、UI定義コンテナ内部のUI定義により実際のUIの構成を決定し、UIサーバ内部に存在する標準部品ライブラリからUIオブジェクト(後述)を生成することによりUIを実現する。実現されたUIにユーザは変更を加えることができる。変更はユーザ毎にUI定義に反映され、UIサーバ上に保存される。次回以降の起動ではこちらが使用される。API部は、API定義コンテナ内のAPI定義からアプリケーション実行中にUI部とAP部が交換するメッセージのフォーマット等を獲得し、APIオブジェクト(後述)を生成する。この部分で異機種間の違いを吸収すると共に制約情報を保持する。

3.2 フィードバックのUI部での実現

部品の利用 UI部に、フィードバックにおける代表的な振舞いを備えた部品、あるいは代表的な振舞いをパラメータとして指定可能な部品をUIオブジェクトとして

用意しておけば、制約定義情報としてフィードバックの実現方法を細かく指定する代わりに部品に対するパラメータを準備するだけで済み、通信量の減少が可能となる。特殊な振舞いを要するものの実現に関しては、UI定義コンテナ中にマシンに依存しない(UI部が解釈可能な)形式で記述された「仕様」を内蔵する。

制約定義コンテナ セマンティック・フィードバックの実現は、応答を指定した部品をUI部に用意し、これにパラメータを与えることで行なわれる。一般にここで必要となる情報としては、AP部内でアトミックに実行される実行単位に対するパラメータのリスト、各パラメータの取り得る値の範囲、フィードバックの仕方等がある[2][3]。AP側のAPI部が各実行単位のインタフェース仕様をAP部から獲得する。こうして得られた制約情報は制約定義コンテナに格納されUI側のAPI部へ転送される。転送された情報は、API定義から生成されたAPIオブジェクトに格納される([図2]参照)。

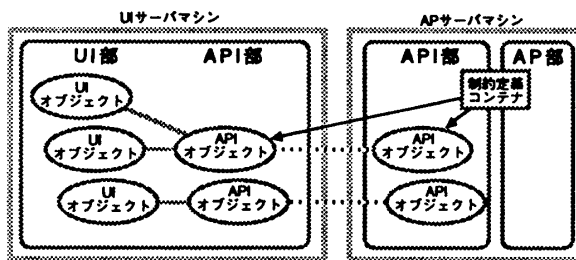


図2: UIオブジェクトとAPIオブジェクト

フィードバックの生成 ユーザの操作に対するフィードバック生成の枠組みについて説明する。[図2]において、ユーザの操作に対するフィードバックは、部品としてのUIオブジェクトによって実現される。この時UIオブジェクトはAPIオブジェクトに保持されている情報を用いる。APIオブジェクトには、制約定義コンテナから得たセマンティック・フィードバックに必要な情報が保持されている(現在、APIオブジェクトはUI側とAP側に重複して存在している)。AP部においてフィードバックに必要な情報に変更が起きた場合、AP部は新しいパラメータを制約定義コンテナに格納して適切なAPIオブジェクトに向けて送り出す。これを受け取ったAPIオブジェクトは、必要に応じて自分の保持する制約情報を更新する。この枠組みによってUI部での高速なセマンティック・フィードバックが実現される。

コンテナ授受のタイミング 制約定義コンテナ授受のタイミングとしては次の選択が考えられる。

- UI側で必要となった時点で転送する。
- AP側で変化があった時点で転送する。

前者ではレスポンスが遅くなる可能性が存在する。一方、後者では不要な転送の発生が起こり得る。以上の点から、UI側API部の制約情報が古くなった場合、AP側API部はUI側の情報の無効化(invalidate)のみを行ない、UI側がフィードバックに要求されるレスポンスの程度を分類し、その制約情報の重要度や最近の使用状況を見ながら転送を行なうかどうかを判断するのが良いと思われる。この部分の実現に関しては、定量的な評価が必要となろう。

3.3 UIのカスタマイズ

このモデルの構造は、UIをカスタマイズする場面でも利用可能である。前述の様にAPIオブジェクトはAP中の実行単位の「インタフェース仕様」として機能するので、これに表示形態を与えることによりUIとアプリケーションとの結合部分においてアプリケーションの構造を見せることが可能となる。UI部ではこれらに対してUI実現部品(UIオブジェクト)を結合すれば良い。これにより、一種のアプリケーション・インタフェースの視覚化が実現できUIとアプリケーションとの結合関係の視覚的な編集が可能となる。

また、部品自体をオブジェクトとすることにより、それら部品を編集するエディタが部品の詳細について知る必要がなくなるため、UIエディタの構築が容易になることも期待される。

4 おわりに

異機種分散環境において、WS上から異機種上のアプリケーションを利用する際に良好なセマンティック・フィードバックの実現を可能とするためのUIMSの枠組みを提案した。また、この枠組みがUIのカスタマイズ(UIとAPとの結合)の手段としても有効であることを示した。現在、この枠組を用いたUIMSを試作中であり、UI部とAP部との結合の視覚化手法、セマンティック・コンストレイントの保持手法を中心に検討を継続する予定である。

最後に、本研究に対して有益な助言を頂いた日本電気(株)C&Cシステム研究所の小池部長、久保主管研究員に感謝致します。

参考文献

- [1] Hudson, Scott E., *UIMS Support for Direct Manipulation Interfaces*, ACM Computer Graphics, vol. 21, no. 2, pp. 120-124, (1987)
- [2] Szekely, Pedro, *Modular Implementation of Presentations*, CHI+GI'87, pp. 235-240, (1987)
- [3] Szekely, Pedro, *Separating the User Interface from the Functionality of Application Programs*, no. CMU-CS-88-101, Technical Report, Computer Science Dept., Carnegie Mellon University, (1988)