

マルチエージェントによるスケジュール管理システムのユーザインタフェース*

塩内 正利 市来 宏基 高田 裕志 毛利 隆夫 和田 裕二

3Q-9

(株)富士通研究所 ネットメディア研究センター

E-mail: {shiouchi, iciki, yuji, tmohri, wada}@flab.fujitsu.co.jp

1 まえがき

アプリケーションシステム構築におけるマルチエージェント技術とは、複数のエージェントと呼ばれる単位部品を組織化し、様々な機能を実現するためのアプリケーション構築支援技術であり、近年、分散環境に適したアプリケーション構築技術として注目されている。

筆者等の研究グループでは、分散環境アプリケーションの典型的例題の一つであるスケジュール管理システムの構築にマルチエージェント技術を適用し、実際のアプリケーション構築における問題点や要素技術についての研究を進めている。

本稿ではこのような分散環境におけるアプリケーションシステムにおいて、インタフェースモジュールへのマルチエージェント技術の適用により、拡張性に優れ、分散環境に適したインタフェースが得られることについて述べる。

2 分散スケジュール管理システム IntelliDiary

本研究では、マルチエージェント技術を用いた分散スケジュール管理システム IntelliDiary [1] を構築している。このシステムはそれぞれのユーザに代わってスケジュールを管理するエージェントが相互にメッセージを交換することにより、分散環境において複数ユーザ間でのスケジュールの調整を行なう。このエージェント自身、データベースの管理やスケジュールのナビゲーション [2] など、様々な役割を持った複数のエージェントにより構成されている。

このようなマルチエージェントシステムにおけるインタフェースには、各エージェント間のインタフェースおよびユーザインタフェースの2つのレベルがある。この場合、ユーザインタフェースとは、ユーザの要求を解釈し、エージェント間通信言語を用いて他のエージェントとの通信を行なうことにより必要な情報を集め、得られた情報を加工してユーザの要求に合致した結果を表示する機能であると言える。

An Interface for a Multi-Agent Oriented Schedule Management System; Masatoshi Shiouchi, Hiroki Ichiki, Yuji Takada, Takao Mohri and Yuji Wada; Netmedia Lab., FUJITSU LABORATORIES LTD. 2-2-1 Momochihama, Sawara-ku, Fukuoka 814, Japan

異機種分散環境では、アプリケーションは様々な環境に対応したユーザインタフェースモジュールを備える必要があるが、アプリケーションのコアとなる部分にアクセスするようなモジュールは、コア部分の仕様に非常に依存するため、柔軟性に乏しく、また、新たな環境への対応も容易ではない。

そこで本システムでは、ユーザインタフェース部分を、(1) ユーザからの要求を取り込む本来のユーザインタフェース部分、(2) システムのコア部分のエージェント (以下、Ego と称する) との仲介を行なうディスプレイエージェント、との2つに分ける構成とした (図1)。

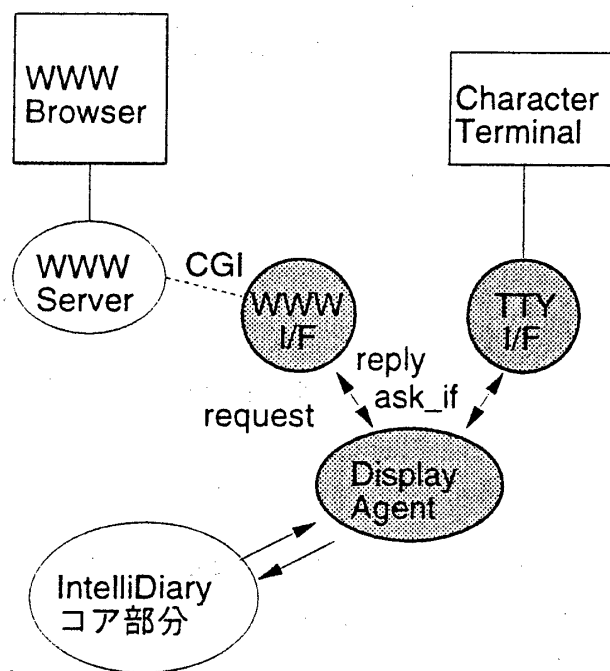


図1: ディスプレイエージェント周辺の構成

このような構成とすることで、ユーザインタフェースモジュールはシステムのEgoの仕様を知る必要がなくなる。また同時に、Egoもユーザインタフェースモジュールの仕様を気にする必要がなくなる。

3 ディスプレイエージェント

ディスプレイエージェントは、様々な環境別に用意されたユーザインタフェースモジュールからのメッセージを解釈し、必要に応じてシステム内のエージェントとの通信によって情報を収集し、それらを加工してユー

ザインタフェースモジュールへと送る。例えば、スケジュールの修正要求を受けたディスプレイエージェントは、ユーザインタフェースモジュールに対し、日付を特定する要求を出す。そして、それに対する回答を受けとり、Ego に対して指定日のスケジュール一覧を要求する。受けとったスケジュールのリストを加工してインタフェースモジュールに送り、修正したいスケジュールの指定を要求する。インタフェースモジュールからスケジュールの指定が得られたならば、そのスケジュールの詳細を Ego から取り寄せ、インタフェースモジュールに送る。そして、修正されたデータを受けとり、Ego に置換えを要求する。

ディスプレイエージェントからユーザインタフェースモジュールへの出力として、データのみを渡すとすると、ユーザインタフェースモジュール側がデータ中の個々の要素がどのような意味を持っているか、そしてそれをどう表示するかを知っていなければならない。特に「～入力画面」「～表示画面」などを複数用いる場合は、全てのモジュール間で表示イメージの構成の同一性を保つ必要があり、一つの環境のインタフェースモジュールでの表示イメージの変更は他の全てのユーザインタフェースモジュールの修正の必要を生じる。

そこで、どのように表示するかを指示する属性の付加機能をディスプレイエージェントに持たせ、インタフェースモジュール側には、データそのものの意味ではなく、表示方法に関する限定された情報だけを解釈する機能を装備することにより、表示イメージの構成の同一性を容易に保持できる。このためのマークアップ言語として、本システムでは HTML を用いている。本来、HTML は SGML のサブセットとして、文書の論理構造を記述するもので、表示方法とは無縁の筈だが論理構造の構成要素と表示方法との対応関係 (SGML における DSSSL に等しい) をユーザインタフェースシステム内に埋め込むことにより、表示方法と密接に関わった文書構造記述言語となっている。そこで、HTML の中で論理構造の記述に適したエレメントを用いて、表示画面を文書と捉えた時の文書構造を記述し、その構造の中にデータを埋め込むことで出力データを合成している。こうして合成された出力データはインタフェースモジュールでレンダリングされ、実際の出力画面が得られる。WWW インタフェースモジュールによる表示例を図 2、3 に示す。ユーザインタフェースモジュールにおいて実際にどのような形で表示されるかはユーザインタフェースモジュールの設計や置かれた環境に依存するが、少なくともインタフェース画面としての

構成は保存されることになる。

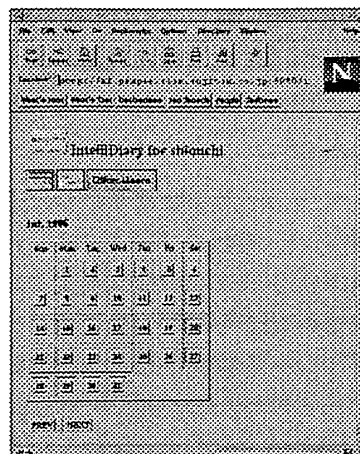


図 2: Netscape での表示例 (カレンダー)

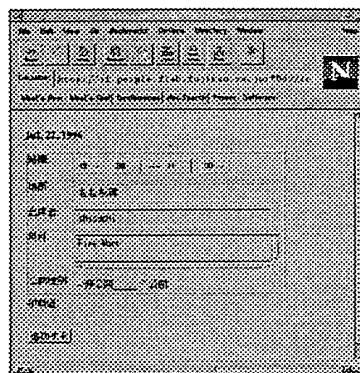


図 3: Netscape での表示例 (スケジュール入力)

4 まとめ

分散環境アプリケーションシステムのユーザインタフェース構築にマルチエージェント技術を適用し、システムのコア部分と本来のユーザインタフェース機能部分との仲介を行なうエージェントを設けることにより、システムとしての拡張性と、ユーザインタフェースの柔軟性を高めることができた。

謝辞

本研究を進める上で、数多くの貴重なご意見を頂きました富士通研究所ネットメディア研究センターの皆様へ感謝致します。

参考文献

- [1] Y. Wada, A. Kawamura, F.G. McCabe, M. Shiouchi, Y. Teramoto and Y. Takada. An Agent Oriented Schedule Management System - IntelliDiary -. In PAAM'96, pages 655-667, Apr. 1996.
- [2] 高田, 市来, 塩内, 毛利, 和田. ユーザの位置情報によるスケジュールナビゲーション. In 情報処理学会第 53 回全国大会, Sep. 1996.