電子化文書の動的な回覧機構について*

3V－6

乃木 篤† 中野 篤‡ 黒川 利明§
（株）CSK 技術本部¶‖

1 はじめに

ワークフロー管理システムはオフィスにおける生産性の向上を目指し、ネットワーク環境下での共同作業を支援するグループウェアの一つである。業務における文書の流れを電子化文書の配信に置き換え、その流れに沿って作業者同士をつなぐことにより業務の迅速化や進捗管理を行なう。

今回我々は、電子化文書の回覧機構について考察し、稟議システムの試作を行なった。稟議回覧システムの開発にはScheme言語を拡張したGuile言語を用いた。

2 ワークフローにおける文書の回覧

ワークフローにおける回覧文書の流れは、図1のように表現される。

稟議の場合、回覧はある種の制約のもとで行なわれる。しかもこの制約はあらかじめ固定された静的なものではなく、回覧途中で変更されるかもしれない動的なものである。

文書ファイルのアクセス権限なども文書に対する制約であるが、例えばOSのファイルシステム管理下のファイルのアクセス権限は、通常そのファイルが所属するディレクトリに記録されていて、ファイルシステムがそれを参照し、ユーザからのアクセスを制御する。このような方法はディスク上のファイルのように所在が固定で、施される処理に対する制約が比較的単純な場合に向いていると考えられる。一方、回覧文書のように移動範囲がネットワーク環境全体に渡り、アクセス制御の他に回覧順序など複雑な制約が課せられる場合、文書と制約をパッケージ化して一つにしてしまった方が、制御を汎用的なものにでき、新たな種類の制約も追加しやすい。

今回のシステムでは文書をヘッダと本体からなるオブジェクトとして構成し、文書に対する処理上の制約をヘッダ部分に含めて記述することでこれを実現した。

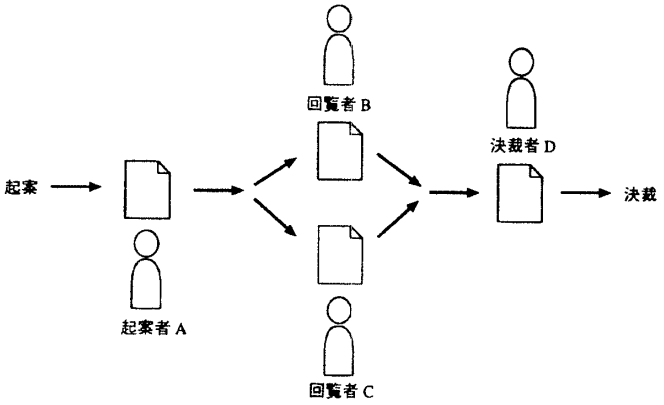

図1: 文書の回覧

3 Guile言語について

本システムの開発にはGuile言語[1]を用いた。Guile言語はScheme言語の一種で、処理系が小さいため、他のアプリケーションに組み込むことができるという特徴がある。プログラムを転送して、他のマシン上の処理系で実行させることもできる。

本システムではクライアント、サーバともにGuile処理系を内蔵し、両者が互いに通信し合う。しかしながら、Guile処理系自信はもともと通信機能をもっていないため、今回新たにsocketインタフェースを追加し、Guile処理系同士が通信できるようにした。その際にGuileが提供するC言語インタフェース機能を活用することで、容易に通信機能を追加することができた。

4 システム構成

本システムは図2に示すように稟議書、ユーザの管理等を行うサーバと、ユーザインタフェースを提供するクライアントで構成されている。

サーバはクライアントに対して稟議システムやユーザ認証などのサービスを提供する。クライアントからの

*Dynamical Circulating Mechanism of an Electronic Document

†Atsushi Nogi
‡Atsushi Nakano
§Toshiaki Kurokawa
¶Technology Division, CSK Corporation
‖Nishi-Waseda Building, 1-12-1 Nishi-Waseda, Shinjuku-ku Tokyo 169, Japan

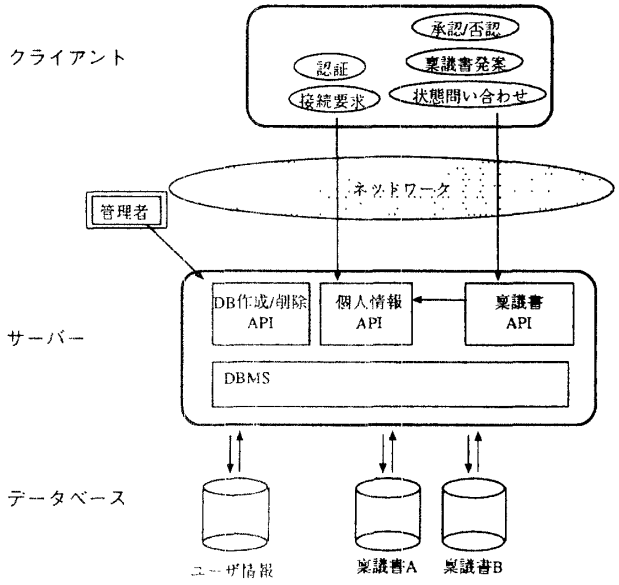

図 2: システム構成

要求により、指定された稟議書のヘッダと本体を配信する。また、稟議書の認可／否認の要求を受け付け、認可の場合は次の回覧者にメイルで通知する。また、クライアントからの要求により稟議書の添付文書のサーバ上のパス名をクライアントに通知する。クライアントは必要に応じてサーバから添付文書をダウンロードする。さらにデータベース管理機能を持ち、稟議書やユーザ情報の登録などを行う。

クライアントは、稟議システムを利用するユーザーがそれらを閲覧したり、起案したりするアプリケーション・プログラムである。できるだけサーバー・クライアント間の通信量を減らすために、クライアント側に独自のデータベースをメモリー上に持ち、キャッシュとして利用している。

将来的にはローカルのキャッシュをメモリ上だけではなくファイルにも格納して、キャッシュを2段階に用意する予定である。問い合わせの優先順位としてはもちろん、キャッシュ・メモリ → キャッシュ・ファイル → サーバー問い合わせの順である。

稟議書の起案はクライアント画面で稟議用のフォームに必要事項を入力することで行なう。項目は入力が必須であるもの、省略できるもの、数値に限られるもの（金額、日付）、などを区別する。

5 回覧ルールの定義

はじめに述べた稟議書の回覧モデルをもとにして、回覧のワークフローを次のように定義した。まず発案者がA、回覧者がB,C、決裁者がDのとき、これまでの書類による回覧では、A→B→C→Dの順序で直列に稟議書が回覧された。しかしこのうちB,Cの部分は本来並列化できるところである。そこで、Aによる発案があった後、稟議書の回覧の通知がB,Cに同時に送られるように回覧ルールを定義する。その際B,Cがともに稟議を承認したときのみ、決裁者に通知が送られるようにする。B,Cのいずれかが否認した場合には、その理由とともに発案者に差し戻される仕組みになっている。

回覧ルールの記述にはGuileのS式を用いた。発案者は稟議書作成の際に回覧者や決議者、回覧順序などを指定する。それがサーバ側でルール形式のS式に変換され、データベースに稟議書とともに保存される。回覧ルールは通常通り直列も指定できる。今後の拡張により、直列、並列の混在や、並列部分の待ち合わせの期間の指定（ある一定期間経過すると、承認／否認に関わらず、次の回覧者に進む）なども可能と考えられる。

6 まとめ

電子会議室や電子稟議などにおける電子化文書の回覧機構について考察し、システムの実装を行なった。文書の回覧は様々な制約の下で行なわれるが、それらをあらかじめ固定された静的なものではなく、回覧途中で変更され得る動的なものと考え、この動的な制約を扱うために文書をヘッダと本体に分け、ヘッダ部に各種の制約を記述し、回覧者が自分の権限に応じて制約の追加や削除、回覧順の変更などを容易に行なえるような枠組を考案した。

今回の開発では動的な制約の変更という点に着目したが、Guileの持つ特性を十分に活かすことはできなかった。今後ヘッダ部分にGuileスクリプトを埋め込むなどより動的な制約の記述が可能かどうか検討していく。

今回試作した稟議システムは、企業内での多様な文書管理のサブシステムとして位置付けられる。今後他のサブシステムとの連携により、オフィス全体の書類回覧業務の効率化を目指していきたい。

参考文献

[1] Free Software Foundation, Inc., Guile User's Manual(1996).