

M-37 数個の処理を実現するためのバッファをもつワークフロー A Workflow Model with Buffers for Processing Concurrent Multiple Tasks

金 鋼[†] 横田 裕介[†] 上林 弥彦[†]
Gang Jin Yusuke Yokota Yahiko Kambayashi

1. まえがき

ワークフローはシステムにおける処理を、コンピュータにも利用者にも分かりやすく表現するための手段として重要である。とくに定型処理の場合は同じような処理が繰り返されるので、ワークフロー上で複数の類似した処理がパイプライン的に実現されることになる。ワークフローは歴史的には人間よりも先にまず機械を対象としていたため、各節点で同時に処理できる仕事の数が1つと規定されていたり、次の節点がまだ処理をしているときは前の節点からは仕事を送られないといったようなモデルが一般に用いられてきており、現在のシステムや人間を対象としたワークフローでも同様に仮定されてきている。しかしながら、実際の場面における人間の仕事の処理を考えると、1つの節点に担当者が複数人いることも多く、複数の処理を行うことが可能である。また処理をする前の時点で複数の処理を待ち行列に入れ待機させておくということも重要となる。ここでは各節点が仕事量に応じた容量を持つようなモデルを考え、さらに各節点に入る部分に対して待ち行列に対応するバッファモデルを提案する。バッファから取り出すときに、各々の処理に対する優先順位を反映させることも可能となる。各仕事についてはそれに対応するデータおよびメタデータがあり、それらを解析することによって全体としての処理の効率化や問題発生回避などが可能となる。本稿ではこの基本的なモデルについて述べる。

2. 基本的な考え方

電子政府における窓口業務などを考えると類似の書類の発行といった仕事があり、これらは受け取られた後、ほぼ同じようなワークフロー処理がなされる。時々問題のある書類があり、その場合には元に差し戻されるといったことも起こりうる。すなわち1つのワークフロー上で常時複数個の処理がなされていることになる。この場合、従来のモデルでは後に依頼された仕事は前に依頼された仕事を追い抜くことができなかつたが、いろいろな問題で優先順位を付ける必要が生じる。例えば仕事が遅れている、すなわち仕事を開始してから非常に遅れているような仕事の場合や、期限があつて終了しなければいけない仕事の期限が迫っているような場合といったことが考えられる。

また、1つの節点に対応する人間が複数人いる場合には、その節点の中で複数の処理を同時に実行することも可能になる。このため、従来のワークフローモデルを一般化して節点では同時に処理できる最大の仕事を定義でき、また節点に入る部分ではバッファがあり、そのバッファの中から次に行う仕事を選ぶ場合に優先順位を利用するといったモデルを考える。これは現実に行われている仕事をワークフロー化するとき有効な拡張であると思われる。

[†]京都大学大学院情報学研究所

3. ワークフローモデル

ワークフローは節点の集合、およびそれらの節点を結ぶ枝集合からなる有向グラフで示される。各節点はアクティビティの単位に対応している。また枝の終点にバッファが用意されており、それはその節点の仕事内容によって決まる最大容量をもつ。また各節点にも容量があり、同時に実行できる仕事量の最大量を示している。枝の先端部分にANDとORの2つの条件があり、ORの場合はそこに入る枝のうち1つの所にある仕事があれば、次に転送されるものである。ANDの場合は、同じ仕事すべての枝に来るという条件となる。対応する仕事が無いような仮想的な節点を許すとANDとORを組み合わせることにより、任意の正関数を扱うことができる。節点から出る枝に対しては条件分岐があり、複数の枝が出ている場合に無条件にすべての枝に出る場合と、特定の条件を満足している場合は特定の枝に出る場合がある。

ワークフロー W は (N, E, C, B, L) で表現される。ここで、 N は各アクティビティに対応する節点、 E は2つの節点を結ぶ有効枝の集合($N \times N$ の上で定義される)、 C は N から正整数 I への関数で各節点の容量を示す。 B は各枝から正整数 I への関数で枝の終点におけるバッファの容量、 L は N から論理 $\{AND, OR\}$ への関数で、各節点の入力枝が複数ある場合に適用される論理関数を示す。

ワークフローの階層化といった一般化も可能であるが、ここでは簡単のために取り扱わない。

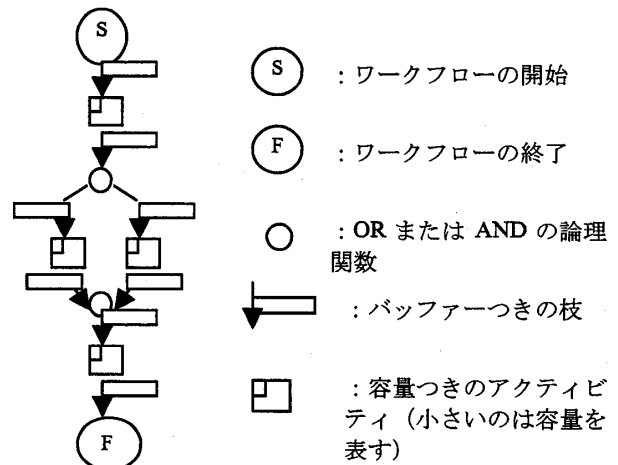


図1: 簡単なバッファ付きのワークフロー

図1はこのモデルを用いた一つの可能なワークフロースキーマである。枝の終点にバッファが用意されており、前のアクティビティから送られた仕事をまずこのバッファの中に入れて、順番に従って処理される。またワークアイテムのバッファ内の順番に対しては制御が可能である。表1のように、バッファにはいくつかの属性がある。

表1: バッファの属性の例

WorkItem ID	Time
005	2002,7,11 12:55:50
008	2002,7,12 10:55:50
010	2002,7,12 12:55:50

WorkItem ID: 一つ一つの仕事を認識するための ID

Time: ワークアイテムがバッファに入った時間

システムはバッファを管理して、場合によってその順番を制御することができる。例えば、締め切りが近づいて期限内に仕事を終わらせなければならない場合には、それに対応するワークアイテムの順番をあげて先に処理する。ワークアイテムごとには優先度(priority) (4章参照) があってそのワークアイテム自身の重要度を表している。最初ワークアイテムがバッファに入る時は自身の優先度の値によって、バッファのどの位置に入るかを定める。しかし場合によっては、あるワークアイテムが長い時間に渡って処理されない状況が発生する。これを防ぐには、Time属性の値からバッファ内の滞在時間を計算し、この時間と優先度を利用して新たなバッファ内の順番を決める。

図1での容量とは一つのアクティビティで同時処理できるワークアイテム数の最大値を表し、その値は動的な変更が可能である。容量の変更およびバッファの働きによって、全体の作業の渋滞を緩和することができる。バッファの大きさは一定であるため、もしあるアクティビティのバッファがいっぱいになったら、そこでの作業が詰まっている可能性が高いと判断することができる。したがって、渋滞が発生した地点のアクティビティの容量を増やすことによって、作業の進行を順調に回復することができる。

4. メタデータと優先順位

ワークアイテムを生成すると、各ワークアイテムに対してメタデータが用意される。その内容は次のようになる。

- 依頼した利用者、ないしは利用者集合
- 仕事の内容
- 仕事の受付時間
- それまでの仕事の経過
- 現在の状態
- 仕事の締め切り時間
- 優先度 (priority)

これらの内容はワークアイテムの状態によって動的に変化するものである。これらの条件から同じバッファ内にある仕事の優先順位を決定することができる。

ワークアイテムの優先順位に影響を与える要素はいろいろ考えられるが、中でも時間、組織全体のポリシーの変更、ワークアイテムのクラスタリングなどが主要な要素となる。例えば、バッファに入った時間から長く経つと優先順位が自動的に上がるようなメカニズムを取ることができる。

5. 利用例

バッファ付きのワークフローモデルを説明するため、簡単な例としてインターネットショッピングを挙げる。図2ではインターネットからユーザの注文が来ると、ワークフローではその注文を一つのワークアイテムとする。商店では注文確認のメールをユーザに送ると同時に、注文書を商品を準備する部門に渡す。その部門では受注した商品を

探して手に入れる。ユーザからの商品の支払いが終わったとき、商品を宅配便などの会社へ送って、ユーザまでの商品の配送を依頼する。

実際の作業では注文書を商品準備の部門にすぐ渡すのではなく、一定量の注文書をまとめてからその部門に送る。そのため、これらの注文書を一時的に格納しなければならない。届いた注文書を複数人の担当者で分担し、商品準備の作業を行う。これがバッファである。そして商品用意の部門では同時に複数もらった注文書のなかで、いくつかの商品は複数人の担当者によって用意される。担当者の人数はワークフローモデルにおける容量に相当する。もし注文書が多ければ、部門の担当者の人数を増やすことによって、同時により多くの商品を用意することができ、全体の作業の効率を上げることができる。商品の届け先までの物理的距離が違うため、商品が顧客に届くまでに掛かる時間は異なる。したがって注文の締め切りも異なる。ここでは単に締め切りを注文の優先度の尺度として挙げて、この優先度とバッファ内の滞在時間と一緒に順番を決める。

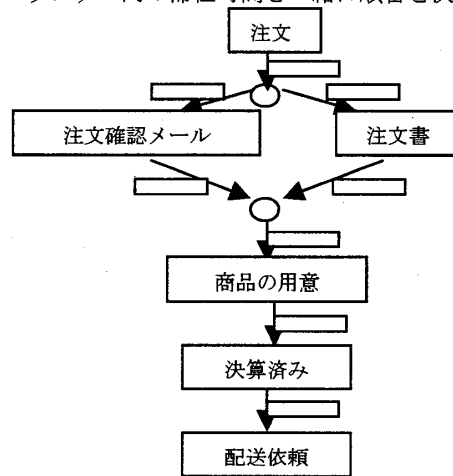


図2: モデルの例—インターネットショッピング

6. まとめ

本稿では、枝ごとにバッファ、節点ごとに容量を持つワークフローモデルを提案した。このモデルにおいては、前の節点から渡された仕事を一度バッファに入れてから処理を行う。これによってバッファ内の仕事の優先順位を制御することができ、仕事の実行時間のより正確な見積もりが可能になる。また、他と比較して待ち時間が長い節点の容量を増やすことによって全体の実行効率を上げるための手段を提供する。今後はこのモデルを基にプロトタイプシステムを実装し、また仕事の締め切りの管理に関するモデルを検討する予定である。

参考文献

- [1] W.M.P. van der Aalst. The application of Petri Nets to Workflow Management. Academic Service, Schoonhoven, 1997.
- [2] David Hollingsworth. Workflow Management Coalition The Workflow Reference Model. 19-Jan-1995.
- [3] Peter Lawrence. Workflow Handbook. John Wiley & Sons, 1997.