M-36

利用者の役割に基づくワークフローの問い合わせ

Querying Workflows based on Roles of Users

横田裕介 Yusuke Yokota
上林弥彦 Yahiko Kambayashi

1 はじめに

従来のワークフローシステムは通常、業務を遂行する組織内での利用を想定しており、実際に業務に携わる人間がワークフローシステムを利用することが前提となっている。しかし、ワークフローシステムを電子政府のような公的なサービスや電子商取引等で利用することを考えた場合、インターネットのようなオープンなネットワーク上で運用することが要求されると考えられる。このような場面では、業務の発注者が現在の進捗状況を問い合わせるといった、組織の外からのシステムへのアクセスが必要になる。従来は外部の存在として考えられていたユーザも含めてシステム上で扱うことによって、このようなオープンなワークフローシステムの特徴を捉えることができると考えられる。これまでのワークフローに対する問い合わせに関する研究としては Christophides [1] らによるものが挙げられるが、あまり多くは行われていない。さらに、依頼者や管理者、業務担当者といった役割の違いを考慮した問い合わせについてはほとんど考慮されてきていなかったが、この機能の重要性は非常に高いものと考えられる。

このようなシステムにおいては問い合わせを発行したユーザの役割によって得られる結果が異なるべきである。例えば、ある業務を発注したユーザがその進捗状況を問い合わせた場合、他者の発注業務の内容まで知ることは望ましくない。問い合わせの例としては以下のようなものが考えられる。

- **Q1:** 「役割 r_1 として発注されたすべての仕事の進捗状況はどうなっているか」(依頼者)
- **Q2:** 「発注した仕事 w_1 はあとどれくらいで終了するか」(依頼者)
- **Q3:** 「自分が担当する可能性のあるすべての仕事を列挙せよ」(業務の担当者)

Q1 については、指定された役割 r_1 を属性に持つすべてのワークアイテムを列挙し、それぞれのワークアイテムの状態を返す。Q2 については w_1 が現在所属するアクティビティの状態を調べ、次にワークフロースキーマを調べて終了までに通過するアクティビティの定義とそれぞれのアクティビティの実行にかかる時間を見積り、積算した結果を返す。Q3 については、問い合わせを発行したエージェントが仕事を割り振られる可能性のあるアクティビティを列挙し、それぞれのアクティビティにこれから到着する可能性のあるワークアイテムを列挙した結果を返す。

本研究で提案する問い合わせ機能によって、ワークフロー上で複数の仕事を同時に処理している状況を的確に把握することができる。また、管理者による問い合わせによって、あるワークフローにおいて処理が終了した仕事はどの程度存在し、一方順番待ち状態になって流れが悪くなっている仕事はどの程度存在しどのように分布しているかを把握することができる。この機能を発展させることによってワークフローのボトルネックの解析を行い、ワークフロースキーマの改善に役立てることができると考えている。本稿では、まず役割という形でシステムのユーザをモデル化する。次に、ユーザによるワークフローシステムに対する問い合わせの機能について考察する。

京都大学情報学研究科

2 ユーザとその役割

ここではワークフローシステムを利用するあらゆる人間をユーザと呼ぶ。ユーザは一つ以上の役割を持ち、その役割によって権限が定められる。ユーザに直接権限が定められることはなく、またユーザの役割はシステム実行中に動的に変更することが可能であるものとする。さらに、ユーザは自分が持つ役割を他のユーザに委譲することができる。次の三種類の基本的な役割を考える。

管理者 あるワークフローの実行に対する責任を持つユーザ。ワークフロースキーマの作成、ワークフローインスタンスの管理などを行う。あるワークフロースキーマおよびそのインスタンスに関してはスーパーユーザとしての権限を持ち、その範囲内ではすべての情報にアクセスすることが可能である。

エージェント アクティビティを実行する人間[1]。エージェントは下位の役割としてさらに業務ドメインに依存した役割を持ち、それに対応する権限が付与される。基本的に、自分に割り当てられたアクティビティに関する問い合わせを行うことができる。

依頼者 あるワークフローインスタンスに対し仕事を発注する人間。発注した仕事はワークフローインスタンス上のワークアイテムに対応する。基本的に、依頼者は自分が発注した仕事に対応するワークアイテムに関する問い合わせを行うことができる。

役割はクラスとして定義され、そのインスタンスが具体的な役割の表現として用いられる。役割のクラスは階層構造を持つ。通常、エージェントの役割は組織の構造に対応した階層構造を構成する。

3 問い合わせのためのワークフローモデル

ここではワークフローに対する問い合わせを考える上で必要となるモデルを定める。主な要素としてワークフロースキーマ、ワークフローインスタンス、アクティビティ定義、アクティビティ(インスタンス)、ワークアイテム、問い合わせを取り上げ、これらの属性の一部を定義する。

[1] ソフトウェアの場合もありうるがここでは考慮しない

ワークフロースキーマ s

AD(s) アクティビティ定義の集合

adm(s) 役割インスタンス集合。sを管理する権限を持つ役割を定義する。$adm(s)$ の要素はすべて管理者クラスに所属している必要がある。

ワークフローインスタンス i

A(i) iに含まれるアクティビティの集合

W(i) iに含まれるワークアイテムの集合、すなわちワークリスト

アクティビティ定義 ad

def(ad) 具体的な動作の記述

schema(ad) 所属するワークフロースキーマへの参照

read(ad) このアクティビティ定義を参照可能な役割のクラスもしくはインスタンスの集合

exec(ad) このアクティビティのインスタンスを実行すべき役割のクラスもしくはインスタンスの集合

$exec(ad)$ の各要素はエージェントクラスのサブクラスもしくはインスタンスである必要がある。$read(ad)$ は依頼者のクラスもしくはインスタンスを含むことができる。通常、$exec(ad) \subseteq read(ad)$ となる。

役割の指定はインスタンスによる指定およびクラスによる指定の双方が可能である。クラスによる指定の場合、ワークフローシステムが実行時に指定されたクラスに所属するインスタンスを定められた基準に従って自動的に選択し、ワークアイテムを割り付ける。

アクティビティ a

ad(a) このアクティビティの定義への参照

queue(a) そのアクティビティにおける処理中および処理予定のワークアイテムのキュー

read(a) adと同様。初期値は $read(ad(a))$

exec(a) adと同様。初期値は $exec(ad(a))$

$read(a), exec(a)$ の初期値はその $ad(a)$ における定義を引き継ぐが、ワークフローの管理者 $adm(schema(ad))$ が値を上書きすることが可能である。

adに対する問い合わせによってワークフロースキーマに関する情報、aに対する問い合わせによってワークフローの実行状態に関する情報を得ることができる。

ワークアイテム w

def(w) 業務発注時に指定されたワークアイテムの定義

from(w) 仕事を発注したユーザ

role(w) 依頼者が仕事を発注したときに指定した役割インスタンス集合。$role(q)$ と同様、$role(w) \subseteq role(from(w))$ を満たす

問い合わせ q

desc(q) 問い合わせ記述。問い合わせ対象の指定、検索条件などが記述される

from(q) 問い合わせを発行したユーザ。問い合わせ結果の送付先として用いられる

role(q) 役割インスタンス集合。問い合わせを発行する際にユーザが指定した役割であり、発行したユーザが持つ役割の部分集合となる

$role(q)$ は「誰からの問い合わせか」を表現する属性であり、これによって問い合わせ時の参照権限が決定される。したがって、$desc(q)$ が同一であっても $role(q)$ が異なれば得られる結果は変わる。

4 問い合わせの動作

問い合わせ実行時はまず初めに $desc(q)$ によって指定された問い合わせ対象集合の要素それぞれについて $role(q)$ と $read(ad), read(a), role(w)$ とのマッチングが行われる。$role(q)$ のある要素がこれらの属性で指定された役割インスタンス集合に含まれる、もしくは指定された役割クラスもしくはそのサブクラスのインスタンスであるという条件を満たす ad, a, w の集合のみを対象として、問い合わせ結果としてのビューを構成する。ここでのマッチング作業によって、役割の違いによるビューの変化が実現される。

管理者は $read(ad)$ および $read(a)$ を設定することによってアクティビティ定義およびアクティビティの公開の程度を調整することができる。例えば $read(ad)$ にエージェントクラスを指定することによって、すべてのエージェントが定義を参照することができるようになる。一方、エージェントクラスのあるサブクラスを指定することによって、特定のグループに属するエージェントのみ定義を参照することができるようになる。これによって、巨大なワークフロー定義の一部を組織内の他部署から隠蔽することが可能になる。

ワークアイテムに関しては、依頼者は基本的に自分が発注した業務に対応するワークアイテムのみを参照することができ、他者の発注業務を参照することはできない。ただし、グループとしての発注として役割を定義した場合や、他の依頼者から役割を委譲されている場合は、自分以外の者が発注したワークアイテムを参照することができる。

5 まとめ

本稿では、ユーザの役割によってビューが変化するワークフローに対する問い合わせのモデルについて述べた。今後の予定として、モデルを詳細化したのちにXMLによる記述を行い、プロトタイプシステムを構築することを考えている。ワークフローのスキーマ記述にはXML Schemaもしくは RELAX、問い合わせの表現にはXML Queryを利用することを検討している。

References

[1] V. Christophides, R. Hull and A. Kumar: Querying and Splicing of XML Workflows. CoopIS 2001, pp. 386-402.