

オントロジーに基づくワークフロー展開システムの実現

A Work Flow Composition System based on Ontologies

牛越正弥*¹
Masaya Ushikoshi

高林裕矢*¹
Yuya Takabayashi

和泉憲明*²
Noriaki Izumi

山口高平*¹
Takahira Yamaguchi

*¹ 慶應義塾大学
Keio University

*² 独立行政法人産業技術総合研究所
National Institute of Advanced Industrial Science and Technology

Due to the advancement of Web and computer technology, variety of human tasks can be classified into three types; tasks involving Web-Service, Application in PC, or tasks simply done by hand. Workflow is composed of these tasks. It is becoming more and more difficult to operate using an optimum workflow and with the most recent working style due to remarkable development of Information Technology. In the present study, we propose a system structuring the workflow of the user's operation through ontological descriptions.

1. はじめに

近年のIT技術の進歩により、人間が行う作業のワークフローには、手作業で処理されるタスクの他に、自分の手元の PC アプリケーションによって処理できるタスク、また Web アプリケーションで処理できるタスクも含まれており、これら3つのものによってワークフローは構成される。また、IT の進歩は目覚しく、数年前では PC で行っていた作業が Web 上で処理できるようになっているなど、そのワークスタイルは時々刻々と進化している。そのために、ユーザが最新のワークスタイルを知り、最適なワークフローで業務を処理することは難しくなっている。また、作業とそれを具体的に処理するための方法の一つにまとめて構築しているシステムがあったとしても、新たなタスクを追加する場合や、タスクを変更する場合などに、そのタスク以下全てを書き換えなければならないため、手間が多くなってしまう。

また、作業のタスクを具体的に処理する方法は、手作業、PC アプリケーション、Web アプリケーションの順に多くなっている。例えば、「書類を提出する」を手作業で処理する場合、その方法は「実際に窓口へ提出する」や「ポストに投函する」などしかない。しかし、「レンタカーを手配」を Web アプリケーションで処理する場合、「トヨタレンタカー」「日産レンタカー」など他にも多数の候補があるため、ユーザが最適な処理方法を見つけることが困難になっている。このことが、作業をより複雑化している要因の一つになっている。

そこで、本研究では、オントロジー[1]を用いることによって、ユーザが行う作業と、それに含まれるタスクの関係を記述することにより、ユーザの条件、状況にあった作業のワークフローを生成することができ、また、タスクの追加容易性、変更容易性を考慮したシステムを提案する。

2. 提案システムの概要

提案システムの概要は以下の図 1 のようになる。

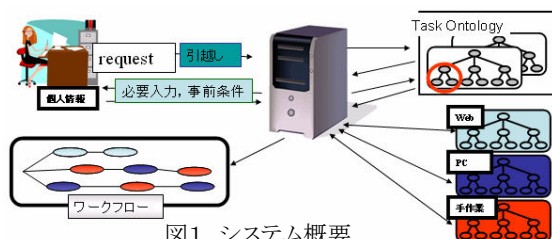


図1 システム概要

システムの流れは、まず最初にユーザが行いたい作業のリクエストを送る(例えば「引越し」)。システムは、そのリクエストをタスクオントロジーから SPARQL[2]を用いることにより検索し、その作業を行うために必要なタスクの発生条件(「引越し」の場合「在学中の子供を伴う」とか「クレジットカードを持っている」等)を抽出する。そして、その条件をユーザが選択することによって、ユーザに合った作業実現のワークフローを作成する。その後、ワークフローに含まれるタスクの実行に際して、それぞれの処理が定義されている Web アプリケーション/PC アプリケーション/手作業オントロジーを参照することによって、実際にタスクが処理されることになる(例えば「引越会社を予約する」という具体的なタスクの場合 Web で実現する場合は「サイトから予約する」、手動で予約する場合には「電話をかける」等)。

以下、使用するオントロジーについて詳細に説明していく。

2.1 使用するオントロジー

本研究における提案システムではタスクオントロジー、Web アプリケーションオントロジー、PC アプリケーションオントロジー、手作業オントロジーの4つのオントロジーを作成した。それぞれのオントロジーの間の関係は図 2 のようになる。

図2の左に示したように、本研究では、「出張」というようなユーザが提示する粒度の大きい行為を「要求」と呼び、出張に含まれる「予約」や「申請」などの行為、またそのサブクラスにあたる「ホテルの予約」や「飛行機の予約」、「出張手続きの申請」などの行為を「タスク」と呼び、タスクの具体的なインスタンスとしての、「楽天トラベルという Web サイトでホテル予約」や「ANA の窓口に行き飛行機予約」などの行為を「処理」と呼ぶことにする。

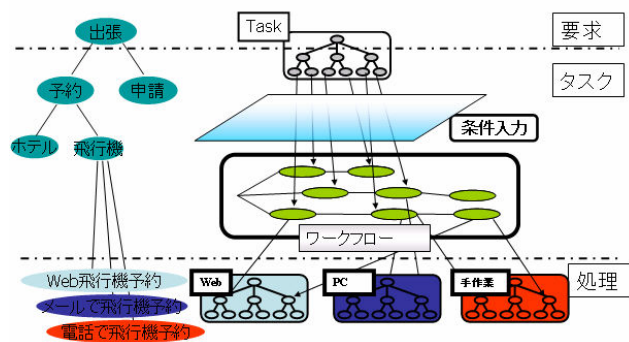


図2 オントロジー関係図

本研究で用いるタスクオントロジー、Web アプリケーションオントロジー、PC アプリケーションオントロジー、手作業オントロジーの4つは、図2のようにタスクオントロジー、Web アプリケーション/PCアプリケーション/手作業オントロジーのように、大きく2つに分けることができ、タスクオントロジーには要求とその要求の持つタスクが Has-A 関係で記述され、Web アプリケーション/PC アプリケーション/手作業オントロジーの3つにはタスクと、そのタスクが持つ処理の Has-A 関係が記述される。

この様に要求とタスクの関係、タスクとその処理の関係の2つに分けて構築することの利点としては、要求から処理レベルまでを一まとめにして構築すると、新たにタスクを追加する場合や、従来のタスクを変更する場合に、そのタスク以下全てを変更する必要があり、多大な手間がかかる。しかし、本提案では、タスクと処理を分けて構築しているため、タスクを追加する際はタスクオントロジーにその項目を追加し、その具体的な処理に関しては、Web アプリケーション / PC アプリケーション / 手作業オントロジーに処理の方法を一部追加すればよいので、本提案システムでは追加容易性、変更容易性が見込まれる。また、処理レベルでワークフローにしてしまうと候補が膨大になりすぎて管理しにくくなるワークフローを、一段抽象化されたタスクレベルで管理することができ、具体的な処理の部分に変更された場合でも、要求やタスクレベルまで変更する必要はなく、ユーザは常にタスクレベルのワークフローから処理に展開することができ、処理方法が変化したときのユーザ負担の軽減を見込むことができる。

次に実際に構築したオントロジーについて具体的な例を挙げて説明していく、本研究においてはオントロジーを引越しの要求に基づき、資料を集め手動で構築した。

(1) タスクオントロジー

タスクオントロジーにはユーザが行う要求とタスクの関係が Has-A の関係で記述されている。また、それぞれのタスクが、その要求の中で発生する事前条件と、行うべき時期、それぞれのタスクの前後関係がプロパティとして定義されている(図3)。

具体的には「引越し」の要求には「レンタカーの手配」「引越し方法の検討」等があり、「レンタカーの手配」が発生する事前条件は「引越会社に頼まない」であり、行うべき時期は「二週間前」であり、「レンタカーの手配」の事前タスクとして「引越し方法の検討」があり、事後タスクとして「引越しを手伝ってもらおう知人に連絡」がある、等のことが OWL-S[3]で記述される

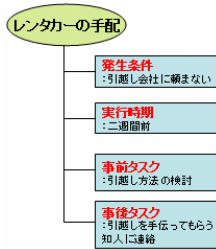


図3 タスクが持つプロパティ

【タスクオントロジー記述の具体例】

```
<予約 rdf:ID="レンタカーの手配">
  <発生条件_yes rdf:resource="#引越し会社に頼まない"/>
  <実行時期 rdf:resource="#二週間前"/>
  <事前タスク rdf:resource="#引越し方法の検討"/>
  <事後タスク rdf:resource="#引越しを手伝ってもらおう知人に連絡"/>
</予約>
```

(2) Web アプリケーション / PC アプリケーション / 手作業オントロジー

Web アプリケーションオントロジー、PC アプリケーションオントロジー、手作業オントロジーはそれぞれ Web アプリケーション、PC アプリケーション、手作業でタスクを処理するために、タスクオントロジーで定義されたタスクに対して、さらにそれを実現するインスタンスを、それぞれのオントロジーごとに定義している。

例えば Web アプリケーションオントロジーには、タスク「レンタカー手配」に対して、その下位概念に「安くレンタカー手配」、「簡単にレンタカー手配」などを記述し、それに対応するインスタンスを作成する(図4)。しかし、「安く」などの感覚はユーザにより異なり、ユーザの個人情報が必要となってくるため、図4のような構造は现阶段では構築中であり、現状のシステムは、タスク「レンタカー手配」の下にインスタンスとして「トヨタレンタカー」などをグラウディングしているだけに留めている。

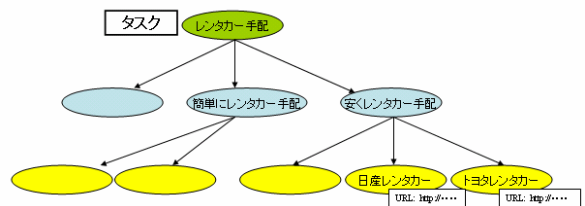


図4 Web アプリケーションオントロジー

3. システムフロー

本研究で提案したシステムのシステムフローを実行画面とともに説明していく。

3.1 作業入力画面

まずユーザは実行したい要求の入力を求められる(図5)。この例ではユーザが行う要求を「引越し」とする。

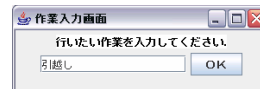


図5 作業入力画面

3.2 条件選択画面

図5にて OK ボタンを押すと、システムはタスクオントロジー上に記述されている入力された要求(この例では、「引越し」)のもつ、要求を実行するために行われるタスクを絞り込むために必要な入力をタスクオントロジー上から取得してきて、ユーザに選択を求める画面が表示される(図6)。この条件を選択することにより、ユーザごとに適したワークフローを生成することができる。

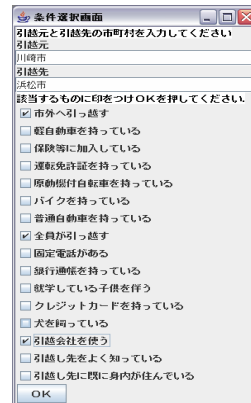


図6 条件選択画面

3.3 ワークフローの出力

タスクにはそのタスクの事前タスクと事後タスクの情報が入っているので、「引越し」要求の全てのタスクとタスクをマッチングさせ、それぞれのタスクの事前タスクと事後タスクがそのタスクにマッチすればタスク同士を結合させる。ここで、図6にてユーザが選択した条件により不要タスクがある場合、その不要タスクの事前タスクは、不要タスクの事後タスクと結合するようにしている(図7)。不要タスクの事後タスクがまた不要タスクの場合は、さらにその事後タスクと結合する。これを繰り返すことでワークフローを有向グラフで出力する(図8)。

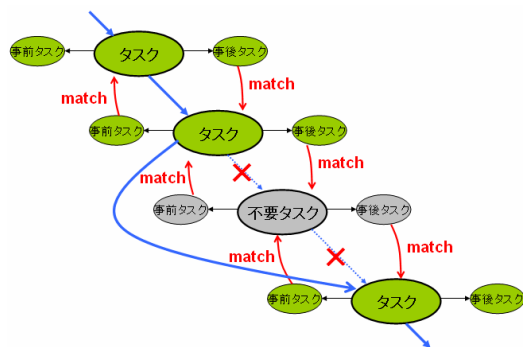


図7 ワークフロー生成方法

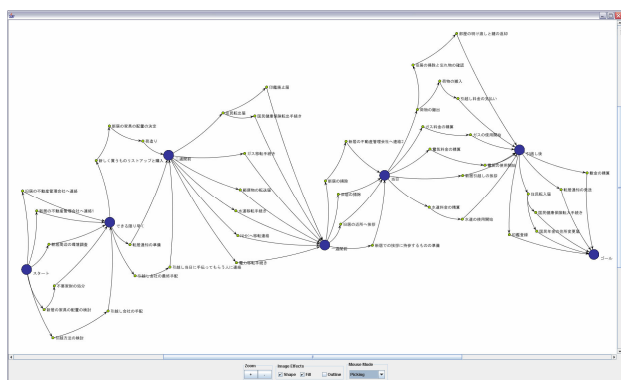


図8 ワークフロー表示画面

このワークフロー上の各タスクを選択すると、それぞれの実行可能な処理方法が現れ、Webであればそのページに、アプリケーションであればその該当アプリケーションを起動し、手作業であれば必要なもの、場所などが表示される。

4. 評価実験

本研究における提案システムの有効性を評価するために、アンケートを用いた被験者による評価実験を行った。実施した評価実験と実験結果について述べた後、考察を行う。

4.1 実験方法

システムを用いることの有用性を示すために、本研究では18名の学生に以下に示す内容のアンケートを行い、実際にシステムを使用してもらい評価を行った。

- ・自分の今の環境において、引越しすることを想定してください
- ・記入表にしたがって、その際に行わなければならないことを入力してください。(Web利用可)
- ・但し、引越先は浜松市、引越元は川崎市としてください。

- ・賃貸物件は契約済みであるとして。
- ・記入表の右上に開始時刻と終了時刻を記入してください。
- ・検索の際、参考にしたサイトがあれば、その名前を記入表の下に記入してください。
- ・記入票への記入がすんだら条件シートをみて合う条件に○をつけてください。

4.2 結果

被験者に川崎市から浜松市に引っ越すという想定で、Web検索を利用してよいという条件でワークフローを作成してもらった。そのとき作成されるワークフローと、システムが条件シートに書かれた条件から生成するワークフローの比較を行うことにより、システムのタスクの網羅性を検証することにより、システムを利用することによる手間の軽減を検証し、システムの有用性を評価する。

アンケートの集計結果より、本システムにより出力されたワークフローに含まれるタスクの数と、ユーザの作成したワークフローに含まれるタスクを比較した。今回の実験ではユーザにより想定した条件が異なるため、システムがそれぞれのユーザに対して出力したタスクの個数と、それぞれのユーザが作成したワークフローに含まれるタスクの個数の平均を取ってみたところ、システムが41.5個出力に対し、ユーザが作成したものは16.2個にとどまった(図9)。

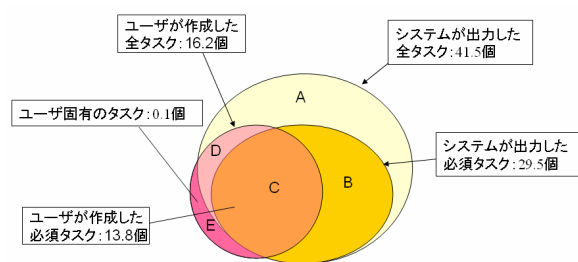


図9 アンケート集計結果ベン図

図9より、ユーザが作成することができなかった必須タスクを、本システムがどれだけカバーしているかは、 $B/(B+C)=53.2\%$ つまり、必須タスク全体の半分以上をシステムがカバーしている結果となった。

次に、各タスクの中でユーザが挙げることが少ないものとして、手続きを含まないユーザのタスクが挙げられる。例えば「近所の人へ引越しの挨拶」や、「部屋の掃除」、「手伝ってもらう人に連絡」など、必須タスクではないが、推奨されるタスクに対して、ユーザはワークフローに組み込まないケースが多かった。

また、ユーザが作成できずシステムが出力した推奨タスク(図9のA)に関して、「新しく購入するモノのリストアップ」といったタスクは必要ないと回答したユーザもいた。つまり、この図9のAの部分に相当するタスクの中には、ユーザにとっては必要でないタスクが含まれている可能性がある。

次に、ユーザは最新のワークスタイルに追従できるかという観点で、実際にどのくらいの人がWebで処理できるタスクをWebではできないと思っているかを調べるため、アンケートでそれぞれのタスクがどのような処理で実行可能かを書いてもらったものを集計し、今回Webで処理できるタスクとして挙げられた、レンタカー・引越し会社の手配、ガス・水道・電力の移転手続き、保険・銀行の住所変更手続きの3つのタスクに関して、ユーザがどの程度Webでできると認識しているかを検証した(図10)。

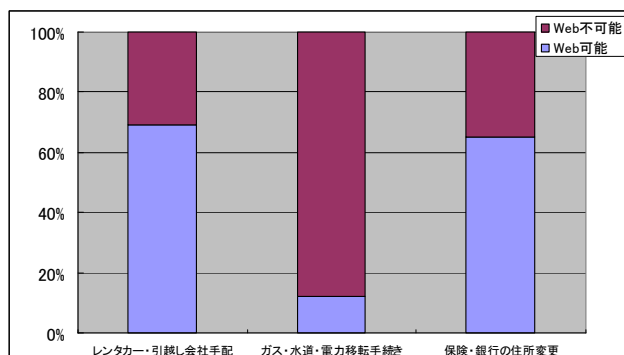


図10 Webで可能なタスクのユーザ認識度

次に実際に数名にシステムを使用してもらい、使いやすさ、よかった点、改善したほうがよい点についてコメントをもらった。

<使いやすさ>

本提案システムではシステムフローが非常に簡単であるので、行いたい要求を入力し、必要な入力と、自分に合う条件を選ぶだけでグラフが出力されるので、使いやすさに関しては、コンピュータの扱いに慣れている人も、そうでない人も、問題はなかった。

<よかった点>

「有向グラフを用いて表示しているので順序と、どこの期間でどのぐらい作業の密度があるかが、わかりやすくてよかった」、「自分が思いつかなかったタスクも提起してくれる」、「自分が行いたいタスクが Web で実行できるのか手作業でなければだめなのか、その候補を提示してくれるので、Web でできると知らなかったものに対しても、実行手段選択により、気づくことができた。」などグラフ表示による把握のしやすさと、今回問題点となっていた候補選択による別の手段の存在の気づきに関して好評を得た。これによって、最新のワークスタイルをユーザが把握することができないという問題に関して効果を示すことができた。

<改善したほうがよい点>

「見やすくするのが面倒である」「どれが Web でできるのか、選択しなくてもわかるほうがよい」、「実行完了したタスクと、そうでないものを区別できるようにしたい」といったグラフの表示・操作面に関する問題と、ワークフローを実行していく過程でのシステム側のフォローに関してコメントを得た。また、「余計なタスク(「手伝ってもらう人に連絡する」等人によっては不要であるもの)が入ることによってフローが見にくかった。」というコメントもあり、どれをデフォルトのタスクとして登録するのかという点にも考慮が必要であると指摘された。

4.3 考察

アンケート結果より、人間が手動でワークフローを作成することは困難であり、本研究の手法と開発したシステムを用いることで、ユーザは自分の行いたい要求を明確に把握することができると示された。また、最新のワークスタイルに気づかないユーザが大多数であると言うことがアンケート結果からわかり、また、「自分が行うタスクが何でできるかわからないとき、実行手段選択により、気づくことができた」というコメントより、最新のワークスタイルに関してユーザに気づかせることをサポートできたと考える。

また、実際にユーザに使用してもらった結果のうち、有向グラフで表示することの視覚的理解のしやすさ、実行手段選択にお

ける最適の選択の補助の部分におけるユーザのコメントにより、タスク実行に関しても十分な補助ができたと考えられる。

また、汎用性としての観点からは、オントロジーを用いることにより、特定のコンテンツに依存しない形式でシステムを構築することができた。このことによって、同じプロパティを用いてオントロジーを構築することによって、引越しに関するワークフローのみを表示するだけでなく、それ以外の要求のワークフローも容易に追加することができるシステムを実現することができた。この点において、今後具体的な検証は必要になってくるが、本システムは汎用的であるといえる。

以上のことから、課題に対する改善は必要となるものの、本システムによりシステムや Web といった近代 IT 技術の発展により複雑になりつつある要求のワークフローを、ユーザにわかりやすく提示し、その要求の補助をするシステムが実現でき、将来、より高度なワークフローの自動処理に向けての可能性を提示することができたと考える。

5. 終わりに

評価実験の結果より、人間が手動でワークフローを生成することは難しく、本研究の手法と開発したシステムを用いて、ユーザは自分の行いたい要求を明確に把握することができ、満足に値する補助ができることが示された。

また、今後の課題として、「引越し」以外のケーススタディを行うことでオントロジーを用いることで生じる汎用性の検証を行っていく必要がある。また、要求の中には Web Service を用いて処理するものもあると考えられる。今後 Web Service が増加すると、Web Service 連携によって処理される要求が増えてくると考えられるので、Web アプリケーションオントロジーと先行研究で行われている Web Service 連携システムとの組み合わせのメカニズムが課題となる。その他には、個人情報はどう扱うかという問題もある。個人情報を必要とする場面は、最初の事前条件入力と Web アプリケーション/PC アプリケーション/手作業オントロジーの2つの場面である。事前条件入力の項目が多い場合に、毎回入力をしなければならないことを考えると、ユーザの情報をなんらかの形で保存し、扱うことが望ましいと考えられる。また、現状ではクラウドに留まっている Web アプリケーション/PC アプリケーション/手作業オントロジーを前述したようなオントロジーに構築する場合に、個人情報をオントロジー化するなどが必要となる。最後に、ワークフローを出力した後でも、その場その場で、ユーザに応じた処理を可能にするメカニズムも必要になると考えられる。

参考文献

- [1] 伊藤英毅:オントロジーを利用した知識の共有/再利用、UNISYS TECHNOLOGY REVIEW 第64号、pp.25-29,2000年
- [2] SPARQL: <http://www.w3.org/TR/rdf-sparql-query/>
- [3] OWL-S: <http://www.w3.org/Submission/2004/07/>