

電子契約の実行監視を目的とするワークフローの変換

根本 雄介[†] 金 剛^{††} 岩井原瑞穂^{††}

[†] 京都大学工学部情報学科 〒 606-8501 京都市左京区吉田本町
^{††} 京都大学大学院情報学研究科 〒 606-8501 京都市左京区吉田本町
 E-mail: [†]{ynemo,jingang}@db.soc.i.kyoto-u.ac.jp, ^{††}iwaihara@i.kyoto-u.ac.jp

あらまし 現在，ワークフローの利用により企業間電子商取引に伴う業務の自動化が進められている．通常，これらの業務は契約に従って行われるが，ワークフローの実行によりすべての契約規則を遵守できるとは限らない．また，例外が発生した場合は人間の判断により問題解決を行う必要があるため，例外と契約規則の関係を明確にする必要がある．そこで，電子契約を参照することにより問題解決の支援を行う Workflow-Contract-Solution(WCS) モデルが提案されている．本稿では，WCS モデルに基づき，電子契約に記述された条件判断などに従ってワークフローを変換し，その結果を契約の実行監視に用いる手法について考察する．

キーワード 電子商取引，ワークフロー，トランザクション処理，電子契約

Transformation of workflow schema for contract monitoring

Yusuke NEMOTO[†], Gang JIN^{††}, and Mizuho IWAIHARA^{††}

[†] School of Informatics, Faculty of Engineering, Kyoto University
 Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501 JAPAN

^{††} Department of Social Informatics Graduate School of Informatics, Kyoto University
 Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501 JAPAN

E-mail: [†]{ynemo,jingang}@db.soc.i.kyoto-u.ac.jp, ^{††}iwaihara@i.kyoto-u.ac.jp

Abstract Recently, workflows are widely introduced to automate inter-organizational business processes. These business processes are usually executed under certain contracts between the companies. However, a workflow does not necessarily fulfill a contract. When some exceptions occur, the exceptions should be resolved by participants' decisions observing the contract. Thus, it is required to specify the relation between the exception and the contract. Then, Workflow-Contract Solution model (WCS model) was proposed. The WCS model supports implementing solutions to exceptions by making reference to e-contracts. In this paper, we propose some methods to transform workflow schema and to monitor contracts with the results of workflow schema transformation based on the WCS model.

Key words e-commerce, workflow, transaction, electronic contract

1. はじめに

現在，企業間電子商取引に伴う業務の連携と効率化のためにワークフローが広く利用されている．また，ウェブサービスとして様々な業務を連携させる手段が整備されている．具体的には ebXML [2] の Common Business Processes や RosettaNet [8] の PIP (Partner Interface Processes) といった標準化案に従ったビジネスプロセスを用いることが考えられる．

これらの電子商取引業務は，通常はあらかじめ締結された契約に基づいて実行される．業務の実行時にワークフローで処理できないような例外が発生した場合，当事者が契約に記述された規則を参照し，契約を遵守した問題解決を行わなければなら

ない．しかし，人間の判断が必要となる例外処理を扱うための有効な枠組みは，これらの既存の標準化案において提供されていない．そこで，文章による契約を構造化された電子契約に変換し，その電子契約の参照により当事者間の議論や判断を支援することが必要になる．

このような例外処理を伴う電子契約の実行支援を行うために，WCS(Workflow-Contract-Solution) モデル [3] が提案されている．WCS モデルを用いることで契約を簡潔に表現することが可能になる．また，契約の参照により例外発生時の当事者間の議論を支援し，問題解決を行う機能のプロトタイプが実現されている [4]．さらに，WCS モデルを用いて過去に起こった契約事例を蓄積し，それらを問題解決に用いるために事例の検索を行う

手法が提案されている [5]。この他にも電子契約のモデル化手法について議論が行われているが [1] [6]、これらのモデルでは例外発生時の問題解決については検討されていない。

一方、国際的な商取引の現場において、CISG (国際物品売買契約に関する国連条約) [12] に基づいて仲裁機関による調停を必要とするような紛争事例が多数報告されている [11]。そのため、企業間電子商取引が広く普及していく上で、紛争の原因となる契約違反に対処する、すなわち自動化されたワークフローの実行時に契約の実行状態を監視して違反が発生した場合に早期に解決策を当事者間で検討するための枠組みを定めることは、非常に重要である。

企業間ワークフローの実行において、電子契約が正しく実行されているかを監視するモニタ機能は重要である。電子契約の実行監視として、契約に定められた義務等が実行されているかを監視する必要がある。それに加え、実行が将来困難になる状況を早期に警告することも必要である。契約違反の発生の可能性を事前に警告することができれば、当事者は事前に対応策の検討を行うことが可能になる。本稿ではワークフロースキーマを電子契約の実行監視が可能となるように変換する手法について考察する。

以下、本稿では、2. 章で WCS モデルの概要を説明する。3. 章で契約実行監視に必要なワークフローの満たすべき条件を説明し、それを元に 4. 章で契約実行監視を可能にするような形にワークフローを変換する手法について述べる。5. 章で電子契約のモデル化と実行監視に関連する研究について述べ、6. 章で本研究のまとめと今後の課題について述べる。

2. WCS モデル

WCS モデルは、電子契約、ワークフロー、および電子契約に基づいたワークフロー実行上の問題解決の三者の統合管理を行うためのモデルであり、電子契約の実行支援を目的としている。

具体的には、契約に伴う業務を記述するワークフロープロセス、契約規則を記述する契約プロセス、問題解決の手続きを与える問題解決プロセスの 3 つのプロセスと、それらの要素間の関係を記述する意味的リンクにより構成され、それぞれ UML [10] のアクティビティ図により表現される。また、WCS モデルのスキーマとインスタンスが定義されている [5]。

以下、3 種類のプロセスとそれらの関係について説明する。

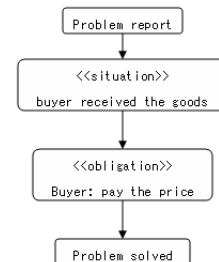
2.1 ワークフロープロセス

ワークフロープロセスは、契約を実行するための組織内及び組織間での作業手順を表現する。通常、ここで定められた順序に従って業務が進められることにより、契約が正常に実行され、関連する業務が完了する。しかし、ここに記述されていない事態が発生した場合、例外として次に挙げる契約プロセスを参照し、問題解決プロセスの実行へと移る。

2.2 契約プロセス

契約プロセスは、電子契約の内容を表現する。通常、契約は文章で書かれた複数の規則により構成されるため、計算機による処理に適するようそれぞれの規則の記述内容を、規則が適用される状況、当事者に与えられる権利、義務といった要素に分

割し、それらの要素を situation, right, obligation といったステレオタイプを付与されたアクティビティとして記述する。要素間の依存関係は制御フローにより記述され、規則の内容によっては条件分岐や並行分岐を含むこともある。以後、契約プロセス中のアクティビティを、ステレオタイプ名を合わせて situation ノード (アクティビティ)、obligation ノード (アクティビティ)、right ノード (アクティビティ) と呼ぶことがある。図 1 は、「買主は商品の受け取り後、代金を支払う」という規則を表現する契約プロセスの例である。



「買主は商品の受け取り後、代金を支払う」

図 1 契約プロセスの例

2.3 問題解決プロセス

問題解決プロセスでは、当事者間の議論や交渉を行う。例外発生時には、契約プロセスを参照し、当事者間の協議により現在の状況に対応する契約事項を定め、それに基づいた解決策を選択する。ここで選択された解決策は、ワークフローの変更として表現される。

2.4 プロセス間関係

上記のプロセス、及びその要素間の意味的な関係は意味的リンクにより記述される。意味的リンクは、意味的に関係のある要素間を結ぶオブジェクトフローとして表現され、要素間の関係の意味を表すステレオタイプをもつ。意味を表すステレオタイプには以下のようなものがある。また、以下に挙げるステレオタイプをもつ意味的リンクは、いずれもワークフローから契約プロセスへ向けて張られていなければならない。

establishes ワークフロー中のアクティビティの実行が、契約プロセスに記述された状況 (situation ステレオタイプをもつアクティビティ) を成立させている。このステレオタイプをもつリンクは situation ノードに対してのみ張ることが許される。以後、このようなリンクを establishes リンクと呼ぶ。

executes ワークフロー中のアクティビティの実行により、契約プロセスに記述された権利や義務 (right, obligation ステレオタイプをもつアクティビティ) を実行している。このステレオタイプをもつリンクは、right, obligation ノードのいずれかに対して張られていなければならない。以後、このようなリンクを executes リンクと呼ぶ。

3. ワークフローによる契約実行監視可能性

電子契約の実行監視は、その契約に従って業務を行うワークフローの実行状態を監視することにより実現される。そのためには、ワークフローと契約プロセスの間に適切な意味的リンク

が張られている必要がある。ここでは、電子契約の実行監視を可能にするために必要なワークフロー、契約プロセス、及び意味的リンクの満たすべき条件を定義する。さらに、それに基づきワークフローの規則に対する無条件遵守可能性、条件付遵守可能性、実行監視可能性といった性質を定義する。

ここで、例として次の規則 A を考える。規則 A を契約プロセスとして表現したものが図 2 である。

「商品検査の完了後、売主は信頼できる検査機関による検査が行われた証拠を買主に示し、不良品率が 30% を超えていたならば商品の代金を規定に従って割り引いて請求しなければならない。不良品率がそれ以下の場合は正規の料金を請求しなければならない。」

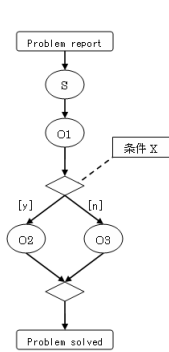


図 2 規則 A を表す契約プロセス P_{CA}

- S (situation ノード):**
検査完了。
- O1 (obligation ノード):**
Seller: 信頼できる検査機関による検査の証拠を提示。
- O2 (obligation ノード):**
Seller: 割引料金を請求。
- O3 (obligation ノード):**
Seller: 通常料金を請求。
- 条件 X (条件分岐ノード):**
不良品率が 30% を超えているか?

3.1 ワークフロー、契約プロセスにおけるノード集合の定義
ワークフローと契約プロセスはともに UML のアクティビティ図により記述される。これらはアクティビティ状態や条件分岐、並行分岐といったノードと状態遷移を表す枝により構成される有向グラフと見なすことができる。そこで、ワークフローをノード集合 V_W と遷移枝集合 E_W からなる有向グラフ $P_W(V_W, E_W)$ と定義する。ノード集合 V_W には開始状態 s_w 、終了状態 f_w が必ず含まれる。

同様に、契約プロセスをノード集合 V_C と遷移枝集合 E_C からなる有向グラフ $P_C(V_C, E_C)$ と定義する。

ノード集合 V_C は、それぞれ situation, right, obligation ノードから成る状況ノード集合 V_S 、権利ノード集合 V_R 、義務ノード集合 V_O 、さらに条件分岐、並行分岐ノード、開始状態 s_c 、終了状態 f_c により構成される。

次に、契約プロセスから生成されるインスタンスに含まれるノードを、各ノードのもつステレオタイプに基づいて分類する。

ある契約プロセスのインスタンスが生成され、開始状態から終了状態に到達しているならば、そのインスタンスは契約プロセス中の条件判断により決定される。そこで、ある契約プロセス中に含まれる条件分岐ノードに記述されている条件 C_C が成立しているときに生成された契約プロセス P_C のインスタンスを $I_C(V_{Ic}(C_C), E_{Ic}(C_C))$ とする。 $V_{Ic}(C_C)$ 、 $E_{Ic}(C_C)$ はそれぞれインスタンス中のノードの集合と遷移枝集合である。

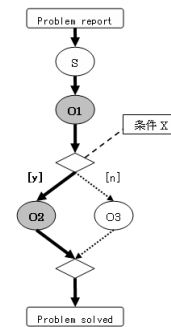
このように I_C は契約プロセスに条件分岐が含まれる場合はその条件判断によって決定されるので、 $V_{Ic}(C_C)$ に含まれる obligation, right ノードの集合を、条件付義務ノード集合 $V_{Io}(C_C)$ 、

条件付権利ノード集合 $V_{Ir}(C_C)$ として次の (1) (2) により定義する。

$$V_{Io}(C_C) \equiv V_{Ic}(C_C) \cap V_O \quad (1)$$

$$V_{Ir}(C_C) \equiv V_{Ic}(C_C) \cap V_R \quad (2)$$

図 2 の契約プロセス P_{CA} の条件 X の下で生成されるインスタンスに対する条件付義務ノード集合は図 3 のようになり (1) により確かにある条件下で生成されるインスタンスに含まれる obligation ノードの集合 (条件付義務ノード集合) を求めることができる。



- $V_O = \{O1, O2, O3\}$
 - $V_{Ic}(X) = \{S, O1, O2\}$
 - $V_{Ic}(\neg X) = \{S, O1, O3\}$
- (1) より
- $V_{Io}(X) = \{O1, O2\}$
 - $V_{Io}(\neg X) = \{O1, O3\}$

図 3 P_{CA} の条件 X 成立時のインスタンスと (1) 式の適用例

さらに、これらの契約プロセス中のノードへの意味的リンクをもつワークフロー中のノード集合を、意味的リンクと契約プロセス中のノードのもつステレオタイプに応じて定義する。

契約プロセス中の situation ノード $v_s \in V_C$ への establishes リンクをもつ、すなわち規則を適用すべき状況を成立させるノードの集合を、状況適用ノード集合 V_{est} として (3) により定義する。

$$V_{est} \equiv \{v | v \langle\langle establishes \rangle\rangle v_s, v \in V_W\} \quad (3)$$

また、obligation ノード $v_o \in V_C$ への executes リンクをもつ、すなわち v_o に記述された義務を実行するノードの集合を義務実行ノード集合 V_{exeo} として (4) により定義する。

$$V_{exeo}(v_o) \equiv \{v | v \langle\langle executes \rangle\rangle v_o, v \in V_W\} \quad (4)$$

同様に、right ノード $v_r \in V_C$ への executes リンクをもつ、すなわち v_r に記述された権利を実行するノードの集合を権利実行ノード集合 V_{exer} として (5) により定義する。

$$V_{exer}(v_r) \equiv \{v | v \langle\langle executes \rangle\rangle v_r, v \in V_W\} \quad (5)$$

3.2 契約実行状態の表現

電子契約は、当事者が契約を構成するそれぞれの規則に従って一連の業務を行うことにより実行される。したがって、電子契約の実行状態を表現するためには、個々の規則の実行状態を表現することが必要である。

ここでは、ある規則を遵守して業務が実行される際に、ワークフローと契約プロセスのインスタンス間に張られる意味的リンクの満たすべき条件として、規則の実行状態を定義する。

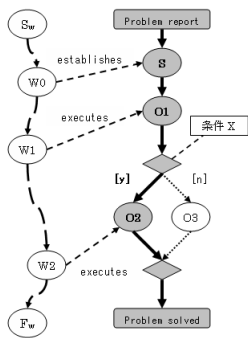
ある規則 C が遵守されている状態とは、一般的に以下の条件 1 が成立しているときに条件 2 も成立している状態である。

- (1) C を適用すべき状況が成立している (状況の成立) .
- (2) C により定められている義務に対応する業務が正しく実行されている (義務の実行) .

状況の成立や義務の実行はワークフロープロセスから契約プロセスへの establishes および executes リンクとして表現される . よって, 規則 C が遵守されている状態は, 規則 C を表現する契約プロセス $P_C(V_C, E_C)$ から条件 C_C の下で生成されたインスタンス $I_C(V_{Ic}(C_C), E_{Ic}(C_C))$ に対して張られる以下のような意味的リンクにより表現される .

- (1) situation ノード $v_s \in V_C$ に対して establishes リンクが張られている (状況の成立) .
- (2) 任意の obligation ノード $v_o \in V_{Io}(C_C)$ に対して executes リンクが張られている (義務の実行) .

このようなリンクを張られているインスタンス I_C の例が図 4 である . I_C へのリンクをもつのは売主のワークフローインスタンスである .



- S (situation ノード):** 検査完了
- O1 (obligation ノード):** Seller: 信頼できる検査機関による検査の証拠を提示 .
- O2 (obligation ノード):** Seller: 割引料金を請求 .
- 条件 X (条件分岐ノード):** 不良品率が 30% を超えているか?
- W0:** 検査実行 .
- W1:** 検査証明書発送 .
- W2:** 割引料金を請求する .

図 4 規則 A を遵守するワークフローと契約プロセスインスタンス

逆に, インスタンス I_C に対してこのようなリンクが張られているならば, 規則 C が遵守されているといえる .

つまり, あるワークフローの実行により規則 C が遵守されているならば, 上記のような意味的リンクがその実行により生成されたワークフローインスタンスから I_C に対して張られていなければならない .

電子商取引に関わる契約は, 主に売主と買主といった 2 者間で結ばれることが主であると考えられる . よって, 以降の議論では, 2 つの異なるワークフローにより契約が実行されることを想定する .

そこで, 契約を実行するためのワークフローを $P_W(V_W, E_W)$, $P'_W(V'_W, E'_W)$ とする . それぞれの実行により, あるインスタンス $I_W(V_{Iw}, E_{Iw})$, $I'_W(V'_{Iw}, E'_{Iw})$ が生成されたとする .

このとき, これらのワークフロー P_W, P'_W の実行により規則 C が遵守されているならば, ワークフローインスタンスのノード集合 V_{Iw}, V'_{Iw} と契約プロセスインスタンス $I_C(V_{Ic}, E_{Ic})$ に対応する状況適用ノード集合 V_{est} , 条件付義務ノード集合 $V_{Io}(C_C)$ に含まれる v_o に対応する義務実行集合 $V_{exeo}(v_o)$ の関係は以下ようになる .

$$V_{est} \cap (V_{Iw} \cup V'_{Iw}) \neq \emptyset \quad (6)$$

$$\forall v_o \in V_{Io}(C_C), V_{exeo}(v_o) \cap (V_{Iw} \cup V'_{Iw}) \neq \emptyset \quad (7)$$

- (6) 式はインスタンス I_W, I'_W による規則の適用状況の成立 (条件 1) に (7) 式はインスタンス I_W, I'_W による規則の遵守に必要なすべての義務の実行 (条件 2) にそれぞれ対応する .

図 4 において, $W0 \in V_{est}, W1 \in V_{exeo}(O1), W2 \in V_{exeo}(O2)$ であることから, $W0, W1, W2$ を含むワークフローインスタンスと契約プロセスは (6) (7) を満たしている .

3.3 無条件遵守可能ワークフロー

ワークフロープロセス P_W, P'_W の開始状態から終了状態へ到達するインスタンスの任意の組が規則 C に違反しないとき, P_W, P'_W を規則 C に対する無条件遵守可能ワークフローとする . このとき, ワークフロー P_W, P'_W の任意の終了状態へ到達するインスタンスの組に対して以下の条件 1, 2 のいずれか一方が成り立つ . また, 実行が終了して開始状態から終了状態へ到達したインスタンスを以後「正常なインスタンス」と呼ぶ .

- (1) 正常なインスタンスの組により規則が遵守される .
- (2) 規則が適用されるべき状況ではない .

条件 1 が成り立つのは, ワークフローと契約プロセスのインスタンスのノード集合が (6) (7) 式を満たす場合である . また, 条件 2 が満たされるとき, 次式が成り立つ .

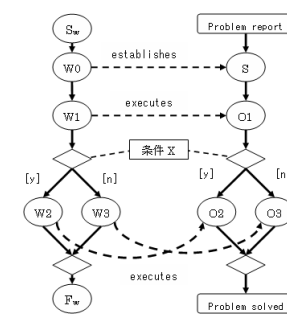
$$V_{est} \cap (V_{Iw} \cup V'_{Iw}) = \emptyset \quad (8)$$

よって, ワークフロー P_W, P'_W が規則 C に対する無条件遵守可能ワークフローであるための必要十分条件は以下ようになる . また, この条件を規則に対するワークフローの無条件遵守可能性とする . ここで, I_W, I'_W はそれぞれ P_W, P'_W の正常なインスタンスである .

無条件遵守可能性

「 $\forall I_C, \forall I_W, \forall I'_W$ に対して, $V_{est} \cap (V_{Iw} \cup V'_{Iw}) = \emptyset$ または $\forall v_o \in V_{Io}, V_{exeo}(v_o) \cap (V_{Iw} \cup V'_{Iw}) \neq \emptyset$ が成り立つ」

具体的には, 規則 A に対して次の図 5 のような無条件遵守可能ワークフローが考えられる .



- S, W0, W1, W2 は図 4 と同じ .
- W3 通常料金を請求する .
- 条件 X が成立, 非成立の場合に, いずれも規則を遵守可能 .
- C_{est} は常に成立 .
- $C_O(O1), C_{exe}(O1)$ は常に真
- $C_O(O2), C_{exe}(O2) = X$
- $C_O(O3), C_{exe}(O3) = \neg X$
- 従って $C_f(O1), C_f(O2), C_f(O3)$ はいずれも常に真である .

図 5 規則 A に対する無条件遵守可能ワークフロー

ワークフローが終了状態まで実行されるとき, そのインスタンスは分岐条件の選択により決定される . 契約プロセスのインスタンスも同様に条件により決定される . そこで, 規則を適用すべき状況を成立させたり義務を実行したりするノードへの遷移条件と上で述べたワークフローの規則に対する無条件遵守可能性の関係を以下に示す .

ここで, あるノード v_1 から v_2 への遷移を確定させる必要十分条件を $tr(v_1, v_2)$ と書くことにする .

規則が適用されるとき、ワークフローの正常なインスタンスは状況適用ノード集合 V_{est} に含まれるいずれかのノードに到達するので、規則が適用される必要十分条件 C_{est} は以下のよう
に書ける。

$$C_{est} = \bigvee_{v \in V_{est}} tr(s_w, v) \quad (9)$$

C_{est} が成立するとき生成されるワークフローの正常なインスタンス I_W, I'_W のいずれか一方は必ず状況適用ノード集合に含まれるノードを含むので、 $V_{est} \cap (V_{I_W} \cup V_{I'_W}) \neq \emptyset$ となる。

また、契約プロセス中のある obligation ノード v_o に記述されている義務が実際に発生するのは v_o が条件付義務ノード集合 V_{I_o} に含まれるときである。従って、義務が発生する必要十分条件 $C_O(v_o)$ は以下のように書ける。

$$C_O(v_o) = tr(s_c, v_o) \quad (10)$$

$C_O(v_o)$ が成立するとき生成される契約プロセスのインスタンス I_C に対する条件付義務ノード集合 V_{I_o} は必ず v_o を含む。よってこのとき $v_o \in V_{I_o}$ である。

さらに、発生した義務 v_o が実行されるとき、 v_o に対する条件付義務実行ノード集合 $V_{exe}(v_o)$ に含まれるいずれかのノードにワークフローのインスタンスが到達しなければならないので、 v_o に記述されている義務が実行されるための必要十分条件 C_{exe} は以下のように書ける。

$$C_{exe}(v_o) = \bigvee_{v \in V_{exe}(v_o)} tr(s_w, v) \quad (11)$$

$C_{exe}(v_o)$ の下で生成されるワークフローの正常なインスタンス I_W, I'_W のいずれか一方は条件実行ノード集合に含まれるノードを必ず含むので、 $V_{exe}(v_o) \cap (V_{I_W} \cup V_{I'_W}) \neq \emptyset$ が成り立つ。

規則を遵守するために義務を実行することが必要になるのは、規則が適用されていてさらに義務が発生している場合である。つまり、「(規則の適用かつ義務の発生) \Rightarrow (義務の実行)」という関係が成立しているとき規則は遵守される。

よって、ある obligation ノード v_o の実行に関して規則に違反しないための必要十分条件 $C_f(v_o)$ は、規則の適用条件 C_{est} 、義務の発生条件 $C_O(v_o)$ 、発生した義務の実行条件 $C_{exe}(v_o)$ を用いて以下のように書ける。

$$C_f(v_o) = \neg C_{est} \vee \neg C_O(v_o) \vee C_{exe}(v_o) \quad (12)$$

ワークフローの任意の正常なインスタンスにより規則が遵守されるならば、契約プロセス中の任意の obligation ノードについて上の条件 $C_f(v_o)$ が満たされなければならない。逆に、任意の条件判断の組に対して $C_f(v_o)$ が真であるならば、与えられたワークフローの組は規則を無条件遵守可能である。

よって、上記の無条件遵守可能ワークフローは、正常に実行されるならば任意の条件下で規則を遵守する。

3.4 条件付遵守可能ワークフロー

ワークフロープロセス P_W, P'_W の少なくとも一つの正常なインスタンスの組が規則 C に違反しないとき、 P_W, P'_W を規則 C に対する条件付遵守可能ワークフローと呼ぶ。これは規則に

違反しないようなインスタンスの組を少なくとも一つ生成することが可能なワークフローであり、以下のような条件を満たす。これを規則に対するワークフローの条件付遵守可能性とする。

条件付遵守可能性
「 $\exists I_C, \exists (I_W, I'_W)$ に対して $V_{est} \cap (V_{I_W} \cup V_{I'_W}) = \emptyset$ または $\forall v_o \in V_{I_o}, V_{exe}(v_o) \cap (V_{I_W} \cup V_{I'_W}) \neq \emptyset$ が成り立つ。」

これは、少なくとも一つの I_C に対応する条件付義務ノード集合 V_{I_o} に含まれるすべての v_o に対して $C_f(v_o)$ が成り立つような条件の組が存在すれば、そのような条件の組を満たすワークフローのインスタンスによってのみ規則 C が遵守されるということを示している。

このような条件付遵守可能ワークフローは、その正常な実行によりある特定の条件が成立している場合に限り規則を遵守するものである。

規則 A に対する条件付遵守可能ワークフローは次の図 6 のようなものが考えられる。

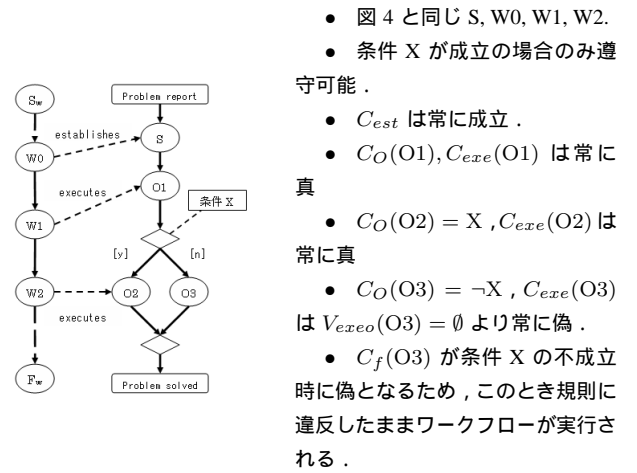


図 6 規則 A に対する条件付遵守可能ワークフロー

- 図 4 と同じ S, W0, W1, W2.
- 条件 X が成立の場合のみ遵守可能。
- C_{est} は常に成立。
- $C_O(O1), C_{exe}(O1)$ は常に真
- $C_O(O2) = X, C_{exe}(O2)$ は常に真
- $C_O(O3) = \neg X, C_{exe}(O3)$ は $V_{exe}(O3) = \emptyset$ より常に偽。
- $C_f(O3)$ が条件 X の不成立時に偽となるため、このとき規則に違反したままワークフローが実行される。

3.5 実行監視可能ワークフロー

電子契約の実行状態はワークフローの実行状態を通して監視されるが、実行監視はワークフローの実行開始から終了まで行われる。アクティビティの失敗などの要因によりワークフローの実行が停止した場合、ワークフローと契約プロセスの間に張られているリンクを参照することで、停止の影響を受ける規則を検出する。その結果、停止により引き起こされる契約違反の予測が可能となる。

しかし、規則違反時にワークフローの実行が停止しなかったならば、当事者は契約違反の発生を検出することなくワークフローに従った業務の実行をそのまま継続することになるので、契約違反に伴って生じる問題の早期解決が困難になる。

このような状況は、規則に対して条件付遵守可能性を満たすワークフローにおいて発生する。なぜなら、条件付遵守可能ワークフローには規則を遵守しないようにその実行を進められる条件の組み合わせが存在するからである。

そこで、violation 状態への遷移を加えることで、次のような実行監視可能ワークフローを定義することができる。ワークフロー P_W, P'_W と契約プロセス P_C のインスタンスのノード間に

次のような関係が成り立つ。

実行監視可能性

「 $\exists I_C, \exists (I_W, I'_W)$ に対して $V_{est} \cap (V_{I_W} \cup V_{I'_W}) \neq \emptyset$ かつ $\exists v_o \in V_{I_o}, V_{exec}(v_o) \cap (V_{I_W} \cup V_{I'_W}) = \emptyset$ ならば、 I_W, I'_W はそれぞれ violation 状態へ遷移する。このような条件が成立しないインスタンス I_W, I'_W は規則に違反しないので、正常に終了状態へ遷移する。」

このような条件が成り立つとき、ワークフローは規則に対して実行監視可能である。実行監視可能ワークフローのインスタンスは、規則に違反する場合は violation 状態へ遷移する。つまり、 $C_f(v_o)$ が成立しないような条件の下で生成されるインスタンスは正常な終了状態 F_W へは遷移せず、violation 状態へ遷移する。

規則 A に対する実行監視可能ワークフローは次の図 7 のようなものが考えられる。

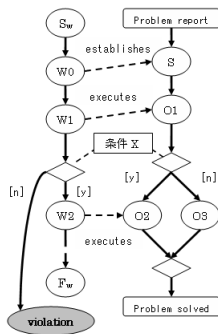


図 7 規則 A に対する実行監視可能ワークフロー

- 図 4 と同じ S, W0, W1, W2.
- 条件 X が成立の場合のみ遵守可能。
- 規則適用条件, 義務発生条件, 義務実行条件は図 6 と同じ,
- $C_f(O3)$ が条件 X の不成立時に偽となるが、このときワークフローの実行は必ず violation 状態へ遷移する。

4. ワークフローの変換

3.3 節で述べた無条件遵守可能性を契約中のすべての規則に対して満たすようなワークフローを事前に設計することは困難である。これは、そのようなワークフローを設計するためには事前に詳細な契約規則に含まれる条件判断等をすべてワークフローに正確に反映することが必要となるからである、そこで、ワークフローと契約プロセスの対応付けに基づいて以下の性質を満たすようにワークフローを変更する。

条件判断による違反の検出 ワークフローの実行時に、規則の遵守を不可能にするような条件が選択された場合は実行を停止して契約違反を検出する。

契約相手の違反の検出 契約相手の義務の不履行により生じる契約違反を検出する。

契約規則中の条件判断がワークフローに正確に反映されていなかったり、契約相手の業務の実行状態を確認するプロセスがワークフローに含まれていなかったりする場合に、これらの性質を満たすように変更を加えることで、ワークフローは 3.5 節で述べた規則に対する実行監視可能性を満たすようになる。

次に、それぞれの性質を満たすようなワークフローの変換手法について述べる。そして、変換後のワークフローを用いた契約実行状態の監視について説明する。

4.1 条件分岐の追加

ある契約プロセス $P_C(V_C, E_C)$ と、それを実行するワークフロー $P_W(V_W, E_W), P'_W(V'_W, E'_W)$ が与えられたときに、ワー

クフローに対して次のような手順でワークフローを追加する。条件分岐追加手続き

(1) V_W の各ノード v に対して、開始状態 s_w から到達することのできるすべての経路を探索し、遷移条件 $tr(s_w, v)$ を求める。

(2) 同様に、 V_O ($V_O : P_C$ に対する義務ノード集合) の各ノード v_o に対する遷移条件、すなわち義務発生条件 C_O を求める。

(3) establishes リンクから P_C に対する状況適用ノード集合 V_{est} を求め、これより 1 で求めた遷移条件から規則の適用条件 C_{est} を求める。

(4) V_O の各ノード v_o に対して、(4a) から (4c) を適用する。

(a) v_o に対する義務実行ノード集合 V_{exec} を executes リンクから求め、これより 1 で求めた遷移条件から義務実行条件 $C_{exec}(v_o)$ を求める。

(b) $C_f(v_o) = \neg C_{est} \vee \neg C_O \vee C_{exec}(v_o)$ を求める。

(c) $C_f(v_o)$ が不成立のとき、 $C_{exec}(v_o)$ も必ず不成立である。そこで、 $C_{exec}(v_o)$ が成立しないようなすべての遷移枝に対して、次のような条件分岐を追加する。

「 $\neg C_{est} \vee \neg C_O$ が成立するならば次のノードへ、成立しないならば violation 状態へ遷移する。」

ただし $C_f(v_o)$ が常に成立するならば変更を行わない。

上記手続きにより、契約プロセス中のすべての obligation ノード v_o に対して、条件 $C_f(v_o)$ が成立する場合にはワークフローの正常な実行が可能になり、不成立の場合にはワークフローのインスタンスは violation 状態へ遷移するようになる。よって、この手続きは 3.4 節で示した条件付遵守可能ワークフローを 3.5 節で示した実行監視可能ワークフローへと変更する。

また、この手続きによる変更が不要なワークフローは、任意の obligation ノード v_o に対して $C_f(v_o)$ が常に成り立っているため、無条件遵守可能ワークフローである。

上記の条件分岐追加手続きの適用が必要となるのは、以下のように条件分岐が存在する場合である。いずれの場合も義務適用条件 C_{est} の成立を仮定している。

(1) ワークフローが条件分岐を含む

(2) 契約プロセスが条件分岐を含む

(3) ワークフロー、契約プロセスともに条件分岐を含む

それぞれの場合について、obligation ノード $O1, O2$ をもつ契約プロセスと、 $O1, O2$ に対して executes リンクを張るアクティビティ $W1, W2$ をもつワークフローを例として変換手法の適用方法を示す。

(1) の場合、図 8 のようなワークフローと契約プロセスの例を考える。義務 $O1$ に対して、義務発生条件が常に成り立ち、義務実行条件 $C_{exec}(O1) = X$ であるから、 $C_f = X$ となる。よって、 X が不成立の場合の遷移枝に条件分岐を追加する。しかし、 C_{est}, C_O ともに常に成立するので手続き (4c) に示した分岐条件は常に成立しない。よって、次の図 9 のように violation 状態への遷移を追加する。

(2) の場合、図 10 のようなワークフローと契約プロセ

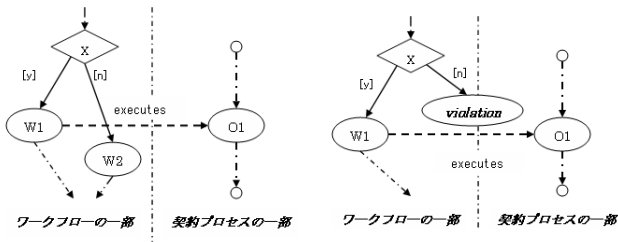


図 8 (1) の例 (変更前)

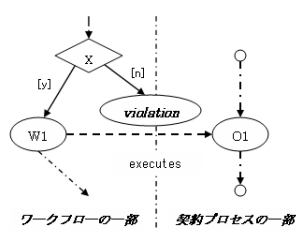


図 9 (1) の例 (変更後)

この例を考える．義務発生条件はそれぞれ $C_O(O1) = X$, $C_O(O2) = \neg X$, 義務実行条件 $C_{exe}(O1)$ は常に成り立つが, $C_{exe}(O2)$ は常に成り立たない．よって $C_f(O1)$ は常に成立するので $O1$ に関する変更は不要である．また, $C_f(O2) = X$ となるので, 義務 $O2$ に対して図 11 のようになる．

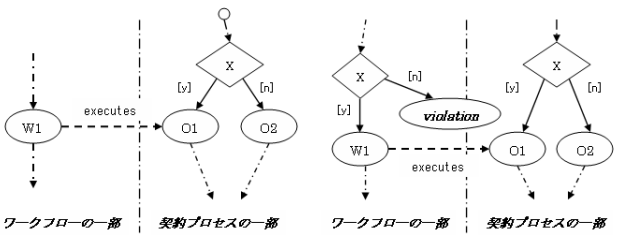


図 10 (2) の例 (変更前)

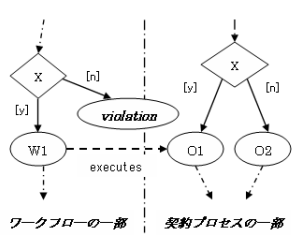


図 11 (2) の例 (変更後)

(3) の場合 (1), (2) を合わせた形の図 12 のようなワークフローと契約プロセスの例を考える．このとき, 義務発生条件はそれぞれ $C_O(O1) = Y$, $C_O(O2) = \neg Y$, 義務実行条件は $C_{exe}(O1) = X$, $C_{exe}(O2) = \neg X$ となる．よって, それぞれの義務に違反しないための条件は $C_f(O1) = \neg Y \vee X$, $C_f(O2) = Y \vee \neg X$ となる．従って義務 $O1$ については $\neg X$ の成立 ($C_{exe}(O1)$ の不成立) により遷移する枝に Y の成立 ($C_O(O1)$ の不成立) により violation 状態へ遷移する条件分岐を追加する．同様に, 義務 $O2$ については X の成立により遷移する枝に $\neg Y$ が成立するときに violation 状態へ遷移する条件分岐を追加する．よって, 変更後のワークフローは図 13 のようになる．

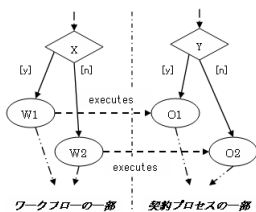


図 12 (3) の例 (変更前)

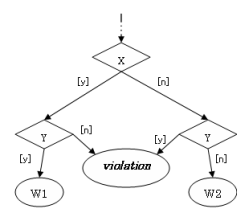


図 13 (3) の例 (変更後)

変更後のワークフローでは, そのインスタンスが規則に違反するような条件が選択されたとき, violation アクティビティに遷移することで違反を検出できる．そのため, ワークフローの実行が正常に終了状態に到達するときは規則が遵守される．

ただし, ここで示したワークフローの変更例は, 規則を適用

する状況の成立が条件付きである場合, すなわち, 契約プロセスの situation ノードに対して establishes リンクを張るアクティビティが条件分岐中にある場合を考慮していない．これは, C_{est} が常に成立すると仮定したことによる．

4.2 確認プロセスの追加

契約プロセス中に義務 $O1, O2$ が並列して存在する場合, 一方の義務を実行しても, もう片方の義務が実行されないと規則が遵守されない．そのため, 自分だけが義務を実行して規則の不履行を知らずにそのまま業務を進めてしまうことが考えられる．そこで, もう一方の義務の実行状態について契約相手に問い合わせを行うプロセスを追加することにより, そのような事態の発生を防ぐ．この問い合わせは相手のワークフローに対して行われ, それに対して相手のワークフローから実行状態を示す返答が出力され, この返答を常に信頼するものとする．

契約プロセス P_C 中にワークフロー P_W により実行される obligation ノード v_{o1} とワークフロー P'_W により実行される obligation ノード v_{o2} が並列して存在したとする．このとき, 以下の手順によりワークフローに確認プロセスを追加する．

確認プロセス追加手続き

(1) $V_{exeo}(v_{o1})$ の各ノード v_1 の後に, 問い合わせ処理を追加する．問い合わせ処理は, 少なくとも 1 つの $v_2 \in V_{exeo}(v_{o2})$ の実行を確認するプロセスと, その問い合わせの結果により実行が確認されたならば次のノードへ, 実行が確認されない場合は violation 状態へ遷移するような条件分岐をからなる．

(2) 同様に $V_{exeo}(v_{o2})$ の各ノード v_2 の後に, 問い合わせ処理を追加．

契約プロセスで義務が並列しているとき, 一方の義務の実行完了と同時に他方の義務の完了も要求されているため, 義務の実行条件は他方の義務も同時に実行される条件を含む．

義務 v_{o1} の実行条件 $C_{exe}(v_{o1})$ は (11) 式によりワークフロー P_W における義務実行ノード集合 $V_{exeo}(v_{o1})$ に含まれるノードへの遷移条件から求められる．しかし, この義務実行条件 $C_{exe}(v_{o1})$ は, 並列して実行されなければならない義務の存在を考慮していない点で不完全な条件である．この点を並列する義務の実行の確認により補うことで, ワークフローが正常に実行される際の義務実行条件を完全なものとしている．

このようなワークフローと契約プロセスの例が図 14 である．ワークフローにより義務 $O1$ を実行しても, $O2$ が実行されなければこの規則は遵守されないが, その場合もワークフロー (1) はそのまま実行を続けることが可能である．そこで, 図 15 のようにワークフロー (1) を変更する．ワークフロー (2) についても同様の変更を行うものとする．

契約相手が義務を実行しないとき, 変更後のワークフローでは, 問い合わせの結果に従って violation 状態に実行が到達し, 規則違反を検出する．

4.3 変換後のワークフローによる契約実行監視

上記の二つの手法により, 規則に違反して実行されうる条件付遵守可能ワークフローを, 規則違反の発生条件の成立を検出可能な実行監視可能ワークフローへと変換することができる．変更後のワークフローの実行により規則は遵守されるため, そ

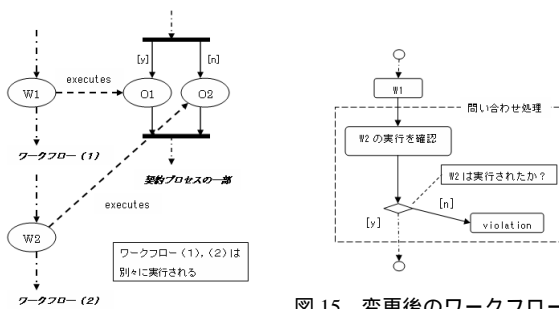


図 15 変更後のワークフロー (1)

図 14 並行する義務とワークフロー

の停止時のみ規則を確認して違反の内容を確認すればよい。

これらの手法によりワークフローの実行を監視することが可能にならない例として、次のような状況が考えられる。

図 16 のように並行する義務が存在するもの以外で実行に複数の当事者が関与するような契約プロセスが存在する場合には、ワークフローにここで述べた変換手法を適用してその実行を監視することはできない。

図 16 は、4.1 節の (1) の例であり、この契約プロセスに対して実行監視可能なワークフローである。ただし、規則の適用状況を成立させるアクティビティと義務を実行するアクティビティが別々のワークフローに存在している。そのため、W1 がの不実行による規則違反はワークフロー (1) においては検出されるが、ワークフロー (2) においては検出されない。

このようなワークフローにも並行する義務が存在する場合と同様に確認プロセスを追加することが考えられる。しかし、この場合に確認プロセスを適切な位置に追加するためには、規則が適用されてから義務が実行されるまでの間に許される猶予期間を考慮しなければならない。従って、このような義務が並行していない場合の契約相手による契約違反をワークフローにより監視することは今回の契約監視手法では扱わない。

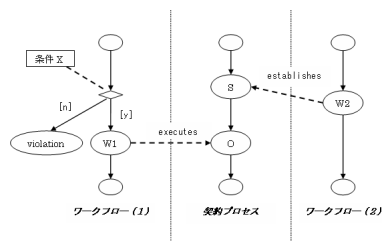


図 16 監視が行われないワークフローの例

5. 関連研究

電子契約の実行を監視する手法としては、BCL (Business Contract Language) により契約を表現し、ワークフロー上で発生する event を監視することで契約の実行を監視する手法が提案されている [7][9]。BCL は、当事者による契約の実行を監視するための言語であり、発生した event に対して内部状態として定められた値を変化させるためのルールとして契約規則を記述する。そして記述された event の発生が検出された際に内部状態が変化し、その結果として契約規則が遵守されているか否かを

確認する。

この手法では、発生する event に対する状態の変化として契約規則を記述している。これに対して WCS モデルは、ある状況の成立時に発生する権利や義務の集合として契約規則を記述している点である。そのため、より抽象的な規則に対して実行の監視を行うのに適している。

6. まとめと今後の課題

本稿では、WCS モデルにおける契約実行状態の表現方法をワークフローと契約プロセスのもつグラフ構造に注目して定式化した。これを基にワークフローの規則に対する無条件遵守可能性、条件付遵守可能性、実行監視可能性といった性質を定義した。そして、条件付遵守可能ワークフローを実行監視可能ワークフローへと変換する手法を、契約規則に条件判断が含まれる場合と契約締結者間で並列して実行されるべき義務が含まれる場合の二つの場合に対して提案した。

今回の変換手法で扱ったのは、あらかじめワークフローと契約プロセスの間に意味的リンクが張られている、すなわちワークフローに記述されている業務と規則に記されている権利や義務との関係が事前に明示されている場合だけであり、これらの関係が契約実行時に動的に変更される場合におけるワークフローの変更を含む契約の実行監視手法を検討することが今後の課題である。また、規則間の意味的な依存関係を考慮して契約の実行監視に役立てる手法を検討することも今後の課題である。

文 献

- [1] S. Angelov, P. Grefen, "A Framework for the Analysis of B2B Electronic Contracting Support," Proceedings 4th Edispuut Conference, Multidisciplinary Perspectives on Electronic Commerce, pp.31-40, Amsterdam, The Netherlands, 2001.
- [2] Electronic Business XML: <http://www.ebxml.org/>.
- [3] M. Iwaihara, H. Jiang, Y. Kambayashi, "An Integrated Model of Workflows, e-Commerce and Solution Implementation," Proc. ACM Symposium on Applied Computing, Organizational Engineering Track, pp.1390-1395, Nicosia, Mar. 2004.
- [4] M. Iwaihara, H. Jiang, Y. Kambayashi, "An Integrated System for Supporting Problem Solution in e-Contracts Execution," Proc. 1st IEEE Int. Workshop on Electronic Contracting (WEC), IEEE CS Press, pp.9-16, July 2004.
- [5] 金剛, 岩井原瑞穂, "ワークフローと電子契約の統合管理における検索機能," TECHNICAL REPORT OF IEICE, Vol.104, No.345, DE2004-116, pp.13-18, Oct. 2004.
- [6] K. Karlapalem, A. R. Dani, P. R. Krishna, "A Frame Work for Modeling Electronic Contracts," Conceptual Modeling - ER 2001: 20th International Conference on Conceptual Modeling, Yokohama, Japan, pp.193-207, Nov. 2001.
- [7] Z. Milosevic, S. Gibson, P. F. Linington, J. Cole, S. Kulkarni, "On design and implementation of a contract monitoring facility," Proc. 1st IEEE Int. Workshop on Electronic Contracting (WEC), IEEE CS Press, pp.62-70, July 2004.
- [8] RosettaNet: <http://rosettanet.org/>.
- [9] R. Tagg, Z. Milosevic, S. Kulkarni, S. Gibson, "Supporting Contract Execution through Recommended Workflows," DEXA2004 Conference, Springer LNCS 3180, pp.1-12, Zaragoza, Spain, Sept. 2004.
- [10] Unified Modeling Language: <http://www.uml.org/>.
- [11] UNILEX on CISG and UNIDROIT Principles: <http://www.unilex.info/>.
- [12] United Nations Convention on Contracts for the International Sale of Goods [CISG]: <http://www.cisg.law.pace.edu/cisg/text/treaty.html>.