ワークフローと電子契約の統合管理における検索機能

金　鋼　　岩井原　瑞穂

京都大学大学院情報学研究科社会情報学専攻
〒606-8501 京都市左京区吉田本町
E-mail: jingang@db.soc.i.kyoto-u.ac.jp, iwaihara@i.kyoto-u.ac.jp

あらまし 企業間の業務提携はWebサービスや企業間ワークフローにより自動化が進んでいるが，これらの業務の実行中に発生した問題は，契約に基づいて解決する必要があり，電子化された契約とワークフローの統合管理が必要となる．ワークフローの実行状態から，関連する電子契約や蓄積された契約実行事例を検索する機能について考察する．

キーワード ワークフロー，電子契約, プロセス，アルゴリズム

Search Functions for Integrated Management of Workflows and Electronic Contracts

Jin Gang and Mizuho Iwaihara

Department of Social Informatics, Kyoto University
Yoshida-Honmachi, Sakyo, Kyoto 606-8501, JAPAN

E-mail: jingang@db.soc.i.kyoto-u.ac.jp, iwaihara@i.kyoto-u.ac.jp

Abstract Automation of inter-organizational businesses is recently advancing, due to the development of web services and inter-organizational workflows. However, in this scenario, contracts between organizations often define rights and obligations of the participants, and shows directions on how to resolve problems occurred during workflow execution. In this paper, we discuss our model of integrating workflow management and e-contract management, putting focus on search functions. We discuss a framework for finding identical or similar instances from case histories which contain records of workflow and contract execution.

Keyword workflow, electronic contract, process, algorithm

1. はじめに

Webサービスなどの近年発達している技術を用いて，企業間の業務を連携させ，企業間ワークフローを構築することが普及しつつある．その場合，企業間の業務提携内容を契約として文書化される.ワークフローや，Webサービスでの契約の履行状況をモニタし，契約に違反した状況にないかを監視する機能が必要とされ，ワークフロー管理システムにこれらのモニタ機能を組み込むことが議論されている[12][13]．また，電子契約の形式的モデルについても議論が行なわれている[1][3][10].

我々がこれまでに提案しているWorkflow-Contract-Solution(WCS)モデル[8][9]では，契約が問題発生時の解決方法の指針を与えていることを踏まえ，契約履行のモニタの他にも，ワークフロー実行における例外発生時にも契約に基づいた問題解決を行なうことを目的としている．WCSモデルではワークフローと電子契約，これらの間の問題解決プロセスのそれぞれの間の関連を記録することにより，契約による問題解決の知識を蓄積し，共有することを目標とする．そのためには，WCSモデルにおける蓄積されたプロセスに高度な検索機能を与えることが必要である．

本稿では，WCSモデルにおける検索機能として，過去の事例から類似したインスタンスを検索したり，統計情報を収集するための基本操作として，ある与えられたプロセスのインスタンスと共通部分を持つインスタンスを求める演算を定義する．さらに，プロセスインスタンスの類似性を求めるための，プロセス抽象化

の手法について述べる．

本稿は以下，2節でWCSモデルの概要を示し，3節ではWCSモデルのスキーマとインスタンスのモデルを定義する．4 節では，WCSモデルに求められる検索機能についてまとめ，プロセスインスタンスの共通部分を求める演算とプロセス抽象化について議論する．5節はまとめである．

2. Workflow-Contract-Solution モデル

2.1. WCS モデルの概要

我々が提案しているWorkflow-Contract-Solutionモデル(WCSモデル)[8] [9]について説明する．**プロセス**とはある目的のために行なわれるアクティビティの集合である．**アクティビティ**は人間または計算機によって実行される作業の単位である．アクティビティを**子プロセス**とすることによりプロセスの階層構造を形成できる．**プロセススキーマ**とは，アクティビティ間の実行順序を定めるものである．プロセススキーマに従うプロセスをそのスキーマの**インスタンス**という．プロセススキーマはパラメータとして，ロール（役割）や子プロセスのスキーマを持ち，スキーマからインスタンスが生成されるときにそれらのパラメータが子プロセスのインスタンスや人を表すオブジェクト等に置換される．

WCS モデルは以下の 3 種類のプロセスを持つ．最初のカテゴリはワークフローであり，2 番目は電子契約，最後は問題解決プロセスと呼ぶものである．

- **ワークフロープロセス**：ワークフロープロセスは作業の手順を定めるプロセスであり，組織内ワークフロー，組織間ワークフローの区別をすることがある．ワークフローの実行により契約の実行が達成される．しかしワークフローの実行を阻害する状況となったとき例外が発生したと認識される．例外には予期されるものと予期していないものが考えられる．例外の解決には契約に定められた解決方法に従う必要がある．予期されない例外は人間によるアドホックな解決方法が考えられ，そこでも契約に関する議論が必要となる．
- **契約プロセス：** 契約プロセスとは，電子契約を構成している要素の間の因果関係を表現したものであり，行為，義務，権利や条件判断とこれらの間の依存関係を表すリンクからなる．参加者の間である契約プロセススキーマについて合意するものとする．例外を電子契約により解決するために，契約プロセステンプレートからインスタンスが生成される．ワークフローとは異なり，契約プロセスは具体的な作業手順を定めるものではなく，義務および権利の抽象的表現である．例外が発生したとき，参加者は例外に対し，議論を通して適切な契約プロセスを選び，その契約プロセスを満たす解決策を決定する．
- **問題解決プロセス：**問題解決プロセスとは，例外を解決するための参加者間の交渉や議論を行なうプロセスである．例外が発生したとき問題解決プロセスが生成され，当事者が電子契約プロセスを参照しながら解決方法を交渉する．解決方法はワークフロープロセスの一時的あるいは恒久的変更である．

2.2. 抽象プロセスと意味的リンク

ワークフローにおいてすべての例外に対する処理をあらかじめ定義することは難しく，ワークフローを中断してアドホックな解決を行なうことが多い．

契約プロセスを導入する主な理由として，ワークフローでは契約の履行を管理する機能が限られることが上げられる．契約には抽象的な表現が含まれ，個々の例外に契約のどの条項が適用されるべきかについては人間の判断が必要となる．しかしそのための有用な依存関係や履歴などの情報を提示する計算機支援が考えられる．WCS モデルでは以下の概念を用いてワークフロー，電子契約，および問題解決プロセスの間の依存関係を表現する．

ワークフロープロセスは作業内容が具体的に示されているアクティビティからなるプロセスである．これに対し，**抽象プロセス(abstract process)**はあるイベントの解釈を与えたり，行為に抽象的意味を割り当てるものである．ここでは UML のアクティビティ図によりワークフロープロセスおよび抽象プロセスを表現する．また**意味的リンク(semantic link)**によりプロセス間またはアクティビティ間の意味的な関連を表現する．各意味的リンクは機能を表現するステレオタイプを持つ．WCS モデルでは，抽象プロセスと意味的リンクを用いて契約プロセス，状況プロセス，問題解決プロセスを表現する．

状況プロセス(situation process)とは，実行中のワークフロープロセスに解釈を与える抽象プロセスである．状況プロセスはワークフローの実行順序を表すものではなく，ある状況が成立していることを宣言しているものである．

問題解決プロセス(solution process)とは，契約プロセスにおける権利/義務アクティビティとワークフロープロセスをリンクする抽象プロセスであり，ある問題解決を実行するプランを表現するものである．問題解決プロセスは，ワークフロープロセスによって実現される．これにはワークフロープロセスを変更したり，断片的なワークフロープロセスの実行によって行なわれる．

上記に述べた各要素を表すクラス図を図 1に示す．各プロセスの振る舞いを記述するためにUMLのアクティビティ図の集合を用いるが，これらをまとめて**WCSスキーマ**と呼ぶ．

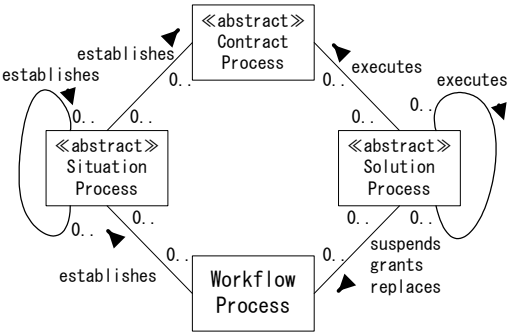

図 1　WCS モデルのクラス図

参加者間の議論や交渉，合意は電子契約に従った問題解決において重要であり，WCS モデルでは抽象プロセスを議論の単位として，抽象プロセスの内容を合意あるいは否決する．

プロセス間の意味的リンクとして以下の型を用いている．establishes はある状況プロセスやワークフロープロセスが電子契約のある条項を成立させているというリンク，executes はある問題解決プロセスの実行により電子契約中のある条項が実行されるというリンク， suspends や grants, replaces はワークフローの実行停止，実行許可，ワークフロープロセスの変更を表すリンクである．

2.3. 抽象プロセスのアクティビティ図による表現

ワークフローの表現として，UML のアクティビティ図を用いることは一般的である．抽象プロセスである契約プロセスや状況プロセスの表現にもアクティビティ図を用いることができる．ここで本稿でのアクティビティ図による表現を述べる．アクティビティ図には開始状態，終了状態，アクティビティ状態（楕円）と，分岐，フォークとジョインからなる同期バー，さらにオブジェクトの値のフローを示すためのオブジェクトフロー状態（長方形）からなる要素がある．アクティビティ図にサブアクティビティ図を定義することで，階層化を行なえる．オブジェクトフロー状態と他の状態間の遷移として，点線の有向枝を用い，それ以外の遷移には実線の有向枝を用いる．１つのフォークで分岐した並行に分岐した状態は１つの同じジョインの同期バーに遷移しなければならないという制約や，その他構造上の制約がある．

抽象プロセスをアクティビティ図で表現するために，以下の方法を用いる．

- 契約プロセスにおける義務,権利, 状況などアクティビティ状態の分類を表すため，ステレオタイプとしてそれぞれ obligation, right, situation を用いる．
- 意味的リンクで他の抽象プロセスとの間にリンクを設定するために，オブジェクトフロー状態を用いる．すなわち１つあるいは複数のオブジェクトを抽象プロセスのインスタンスとし，外部からの参照が必要な制御状態のところに，適宜オブジェクトフロー状態を定義し，意味的リンクをそのオブジェクトに対するリンクとして定義する．ここで用いるオブジェクトを**プロセス状態オブジェクト**あるいは**状態オブジェクト**と呼ぶ．

UML2.0 では，アクティビティ図が拡張されさらに高度な平行システムやアルゴリズムが記述できるようになるが，本稿では現行版の UML1.5 をベースにしている．

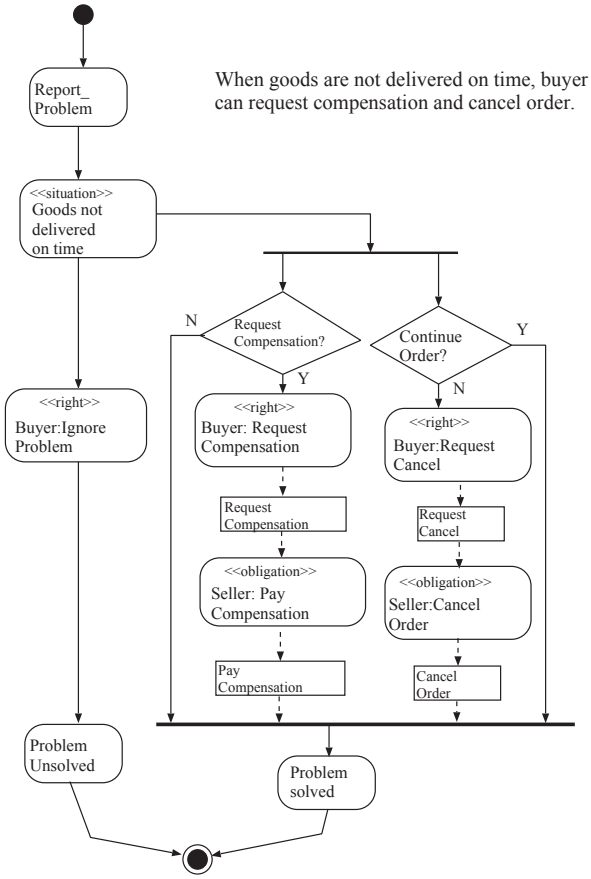

図 2　電子契約プロセスの例

3. WCS モデルのスキーマとインスタンス

WCSモデルは図 1に示したように契約プロセスや状況プロセスなどの抽象プロセス，およびワークフロープロセスがあるが，これらの個々のスキーマはア

クティビティ図で定義される．またプロセス間は指定された型の意味的リンクで接続され，これらにより１つの**WCSスキーマ**が構成される．またWCSスキーマのインスタンスを以下で定義する．

WCS スキーマを構成するプロセススキーマおよび意味的リンクでは，各有向枝に識別子を割り当てておき，その識別子およびその有向枝に割り当てられているステレオタイプや分岐条件などを連結したものを，その有向枝のラベルとする．

また，２つの有向パスが**同型**であるとは，パスの始点から終点までのラベルの列が一致するときとする．

まずアクティビティ図で定義されたプロセスのインスタンスについて定義する．あるプロセス A の**インスタンス** I とは，次の(a)-(e)を満たす有向グラフである．

(a) Aの開始点s_Aに対応するノード$s_{A'}$がIにあり，Iは$s_{A'}$からIのすべてのノードに到達できる有向非巡回グラフである．

(b) $s_{A'}$ を始点とするIのすべての有向パスp'について，Aにp と同型のパスpが存在する．

(c) A'のある枝 e'が A の条件分岐ノード c から出てゆく排他条件を持つ枝と同じラベルならば，A'は排他条件を満たすために e'の始点から出る枝は e'のみである．

(d) Aに含まれる同期バーのフォークfについて，fからラベル$l_1,..,l_n$のn本の枝が出ているならば，A' においてラベル$l_1,..,l_n$のうち１つでもあるノードe'から出ているならば他の$(n-1)$本のラベルもe'から出ていること．

(e) Aのアクティビティ状態 n としてサブプロセスがサブアクティビティ図 A''として定義されているならば，Iにおいては，A'' に含まれるラベルを持つ部分グラフは A''に対するあるインスタンス I''となっており，n に入る枝は I''の始点に接続され，n から出る枝は I''の終点に接続されること．

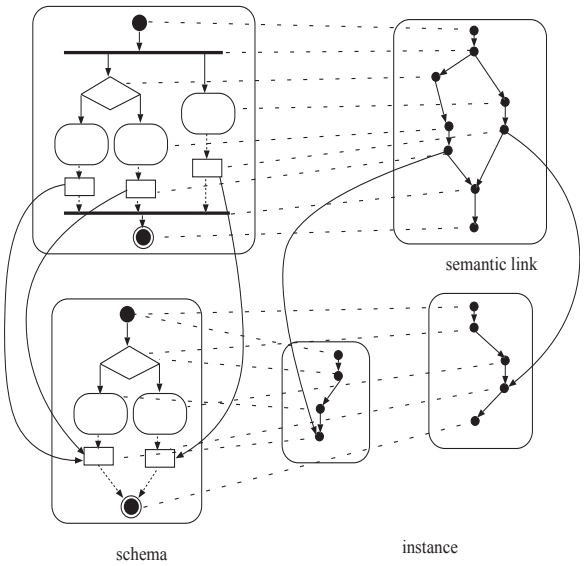

図 3　WCS モデルのスキーマとインスタンス

図 3にWCSモデルのスキーマとインスタンスの関係の例を示す．

上記のプロセスインスタンスの定義では，スキーマではループが含まれていてもよいが，インスタンスでは開始点を根とする有向非巡回グラフであるとしており，またインスタンスはスキーマの一部分に対応するものでよい．

WCSスキーマのインスタンスとは，WCSスキーマに含まれるプロセスに対するインスタンスの集合と，意味的リンクの集合であり，意味的リンクは図 1で指定されたリンクの型に従って，始点と終点のプロセスインスタンスの状態オブジェクトを接続するものである．

プロセスのインスタンスは非巡回有向グラフであるが，WCS スキーマのインスタンスは，意味的リンクを含む閉路が存在してもよい．

上記のインスタンスの定義は，グラフの模倣(simulation)の概念に基づいている．すなわち，２つのグラフG_1からG_2への模倣(simulation)とは，以下を満たすノード間の二項関係Rである：　G_1においてx_1からx_2へのラベルaの有向枝(x_1, a, x_2)がありかつ$(x_1, y_1) \in R$ならば，G_2に有向枝(y_1, a, y_2)が存在する．

模倣は２つのグラフの枝数の積に比例する時間で計算できる．２つの模倣R_1, R_2の和$R_1 \cup R_2$がまた模倣という性質があることから，最大模倣が一意に存在することが分かる．ただしインスタンスとスキーマに閉路が存在し得ることから，インスタンスのあるノードがスキーマのどのノードに対応付けられるかは，模倣では一意には決まらない．

4. WCS モデルにおける検索機能

4.1. 検索機能の応用

WCS モデルの目的とするワークフローと電子契約の統合管理において，電子契約を用いた問題解決のために有用な情報を，蓄積された WCS モデルのインスタンスから検索する機能について議論する．

WCS モデルに対する機能について，以下の利用目的が考えられる：

(a) 類似した事例の検索: ある例外について，それと同様の，あるいは類似したワークフローインスタンスを検索し，その事例がどのように解決されたかを契約プロセスおよび問題解決プロセスのインスタンスを検索する．そして検索キーとしていたワークフローインスタンスと対比させ，一致する部分や相違する部分を求める．あるいは逆に，契約プロセスのある条項について，それが適用されたワークフローインスタンスの事例を調べ，共通する特徴を求める．また,過去のインスタンスを現在実行中のインスタンスに重ね合わせることにより,過去の実行履歴を意思決定の参考にする．

(b) テキストデータベースとの連携：WCSモデルにより蓄積されるインスタンス以外にも，過去の事例や判例などを収集しているテキストデータベースを利用することが考えられる．例えば貿易関係の紛争処理事例を公開しているPace UniversityのCISGデータベースがある[5]． WCSモデルで蓄積されるインスタンスは，ワークフローや契約の実行履歴のレポジトリと考えられる．ある時点でのインスタンスから，その状況に関連するキーワードを求め，それを用いてテキストデータベースを検索することが考えられる．

(c) 統計情報： ワークフロープロセスや契約プロセスの実行履歴の統計を取ることにより，例外の発生しやすい箇所や採用されやすい問題解決プロセスを求め,ワークフロースキーマや契約プロセスのスキーマの改良に役立てる．

4.2. インスタンスの共通部分演算

4.1節で議論した検索機能では， ２つのインスタンスを比較し，同一である部分や類似した部分を求める操作が基本となっている．本節ではそのためのインスタンスの共通部分演算を定義する．これは，WCSスキーマのあるインスタンスについて，指定したノードと，（部分的に）一致する実行履歴を持つノードを求めるものである．まずそのために，最も簡単な，与えられたインスタンスI_1とそのノードn_1,もうひとつのインスタンスI_2およびそのノードn_2について， 等価な部分グラフequiv(I_1,I_2,n_1,n_2)を求める演算を以下に定義する．

インスタンスI_1(同様にI_2)においてノードn_1(同様にn_2)を含む連結成分を$I_1(n_1)$ (同様に$I_2(n_2)$)とする．このとき，以下を満たす$I_1(n_1)$と$I_2(n_2)$のノードの間の二項関係Rを求める．

(1) $(n_1, n_2) \in \mathrm{equiv}(I_1,I_2,n_1,n_2)$.

(2) $I_1(n_1)$に有向枝(x_1, a, x_2)がありかつ$(x_1, y_1) \in \mathrm{equiv}(I_1,I_2,n_1,n_2)$ならば，$I_2(n_2)$に有向枝$(y_1, a, y_2)$が存在する．逆に$I_2(n_2)$に有向枝$(y_1, a, y_2)$がありかつ$(x_1, y_1) \in \mathrm{equiv}(I_1,I_2,n_1,n_2)$ならば，$I_1(n_1)$に有向枝$(y_1, a, y_2)$が存在する

上記の共通部分演算では，２つのインスタンスの間に模倣を双方向に定義しており，双模倣(bisimulation)と呼ばれる.そして指定した２つのノードn_1, n_2と連結な部分に双模倣を制限している．双模倣も最大のものが求まり，枝数m_1とm_2のグラフにおいて，$O((m_1+m_2)\log(m_1+m_2))$で計算する効率の良いアルゴリズムが知られている[14].

上述の共通部分演算により，２つのインスタンスの指定したノードからのパス集合が等しくなるノード間の対応関係が得られ,同じ振る舞いをしている部分や,あるいは異なる振る舞いが始まる部分を重ね合わせて比較できる．

4.3. プロセスの抽象化を用いた共通部分演算

4.2節で定義した共通部分演算は，ワークフロープロセスや契約プロセスの内部の手順まで全て一致するものを求めている．しかし，類似した事例の検索では厳密に同じ振る舞いでなくても，高い類似性を持つ部分が求まれば有用となる場合が考えられる．例えば，契約の同じ条項に関連した例外が，異なる内容のワークフローで起きている場合で，問題解決方法を調べたい場合などである．また,ワークフロースキーマや契約プロセスのスキーマが進化することにより，異なるバージョンのスキーマでのインスタンスの比較が必要になることが考えられる．

以下では，抽象化としてインスタンスの重要な部分を残して簡略化し，より広範囲で等価な部分を求める手法を考察する．

(1) プロセス内部構造の抽象化

これは１つのプロセスにおいて，条件分岐や並行分岐などの分岐ではないアクティビティを除去し，分岐ノードのみを残すものであり，指定したプロセスのインスタンスについてグラフの簡略化を行なう．分岐ノードでの条件判断の結果のみを等価性の判断基準とするものである．

(2) プロセスインタフェースの抽象化

これは指定したプロセスをブラックボックス化し，外部とのインタフェースのみ着目して，インスタ

ンスを簡略化するものである. WCS モデルにおけるプロセスのインタフェースとしては，状態オブジェクトがあり，各状態オブジェクトとそれに接続する意味的リンクのみにインスタンスを簡略化する.

(3) プロセスオントロジによる抽象化

プロセスやアクティビティの機能分類を示すために，WCS モデルでは obligation などのステレオタイプを用いている．これを拡張した分類階層をプロセスオントロジと呼ぶクラス図として用意する．アクティビティのオントロジとして，例えば delivery というクラスを用意し,そのクラスのプロパティとして delivery に伴いやすい例外である delay, missing, returned などが与えられる．インスタンスにおける有向枝のラベルを，その有向枝が接するアクティビティやそれを含むプロセスのプロセスオントロジにおける上位概念の名前に置換する．これによりラベルのプロセス固有の表記をより一般的なものとし，等価と判断できる部分を拡張できる.

図 4に，プロセス内部構造の抽象化（structural abstraction）とプロセスインタフェースの抽象化(interface abstraction)を図式化したものを示す.

上記の(1)-(3)は組み合わせて用いることができ，例えば(2)でプロセスをブラックボックス化したとき，そのプロセスに含まれていたアクティビティを(3)のプロセスオントロジで抽象化を行い，得られた一連の概念名をプロセス自身に付随する概念としてラベルに付加することなどである．またこのようにして得られたプロセスの概念名の集合を，事例データベースのテキスト検索キーワードに用いることができる.

以上の抽象化は適用に任意性がある．そのため,4.2 節の共通部分演算と組み合わせて，等価な部分が見つからないときに，抽象化により等価とみなせるかを試みることなどが考えられる.

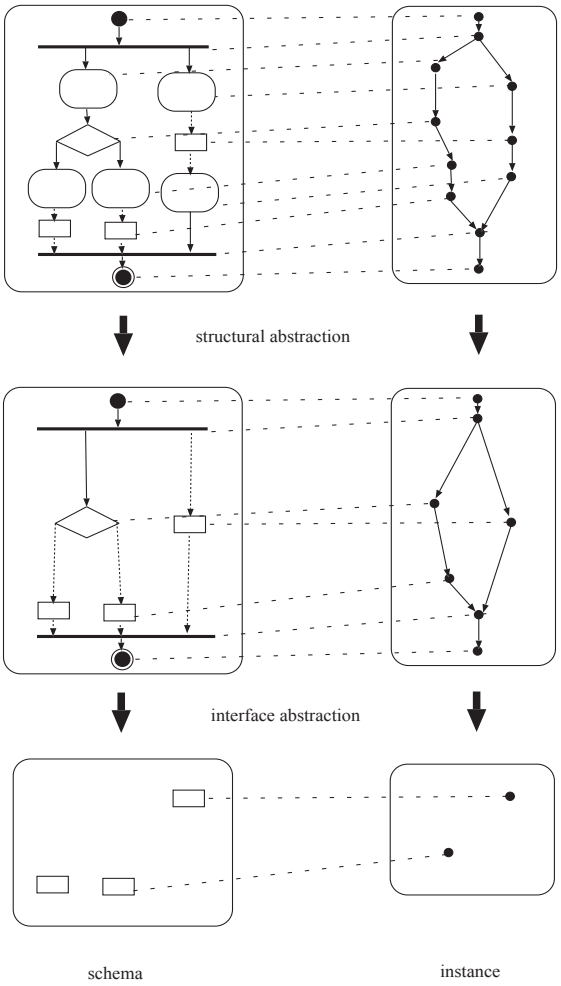

図 4　プロセス抽象化

4.4. 抽象化の実例

4.3 節ではプロセス抽象化の三つの手法について簡単に説明したが、本節ではさらに実例を用いてその中の一つプロセス内部構造抽象化を用いた例外処理のためのワークフローインスタンス検索について記述する。

(1)履歴データベースの構築

ワークフローインスタンスの検索を行う前に、まずワークフローインスタンスのデータがどのように蓄積されるかを検討する。

- インスタンス ID: ワークフローインスタンスを唯一識別するアイテム
- 例外種類：一つのワークフローインスタンスには一つの例外が発生する。例えば、late payment, defect goods のような簡単な単語セットで表現する。
- 適用ルール：例外が発生した場合、それを解決するために適用した契約のルールあるいはルールセットを示す。
- 契約当事者：売買契約でそれぞれの買主、売主を

記述する。

- 役割：契約を実行する会社が契約での役割。同じ会社でも買主になったり、売主になったりすることが可能である。従って、契約で実行するワークフローも異なる。
- 商品：商品の名前である。

(2)プロセス内部構造抽象化

前節でも述べたようにプロセス内部構造抽象化は分岐ノードのみを残すもので、条件判断の結果を透過性の判断基準としているが、そのやり方の正当性について説明しよう。

一つの会社は契約での役割が決まったときには契約を実行するためのワークフロー同じか類似しているわけである。従って、ワークフローの中の分岐のセットも定まることになる。会社が買主である場合のワークフローを見てみると、①不動産の抵当をするか②分割支払いが可能か③最終目的地であるか、などの分岐が存在する。ここで例えば、分割支払いを契約当時に認めたため、後で代金支払いを締め切り以内に全額払えなかったような例外が発生することがしばしばある。また配達の最終目的地であるかどうかにより、商品品質に問題が発生したり、配達の締め切りが送れたりなどの例外が発生する。従って、分岐の条件判断の結果に注目することにより、ワークフローで発生する例外の場所を発見する確率が高くなり、類似したワークフローインスタンスを検索することにおいて効率を上げることができる。

それでは実際のワークフローインスタンスの例を用いてプロセス内部構造の抽象化を説明する。

まず抽象化を利用した類似したワークフローインスタンスの検索のシナリオを説明する。

1. 現在のワークフローインスタンスに対して分岐だけを残す抽象化を行なう。
2. 分岐及びその値をキーとして履歴データベースから検索を行なう。
3. さらに検索結果に対し例外内容をキーとして検索を行なう。

会社 A（買主）と会社 B（売主）の間には手袋という商品の売買契約が締結された。その契約の一部内容は以下とする：

- 配達日は 2005 年 9 月 2 日にする。
- 商品の値段は 1000 円/個とする。
- 最終目的地に配達されたあとに商品の代金支払いを行なう。
- 分割支払いは可能とする。

そして買主のワークフローは図 5 のように表示できる。

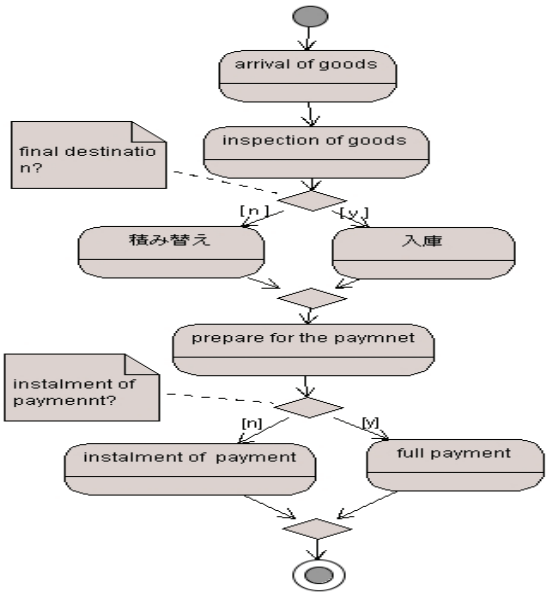

図 5．会社 A のワークフロー

図 5 は会社 A が売買契約で買主という役をするときワークフローである。買主はまず配達会社からの商品の到達を売主に報告し､商品に対し品質検査を行なう。そして現在の港が最後の目的地であるかを判断する。これは買主がさらに小売販売店に商品を売った場合に再配達をする必要があるためである。最終目的地である場合は積み替えを行い、再配達を実行する最終目的地でない場合には入庫させる。つづいて代金支払いの準備を行い、分割支払いか一括払いかを判断しそれぞれに対応する動作を実行する。

図 5 から分かるように、買主のワークフローには二つの分岐が存在するが、今回の取引では買主がさらに小売販売店商品を販売しており、分割支払いが認めたため、これに伴った内部構造抽象化した結果、図 6 のようになっている。

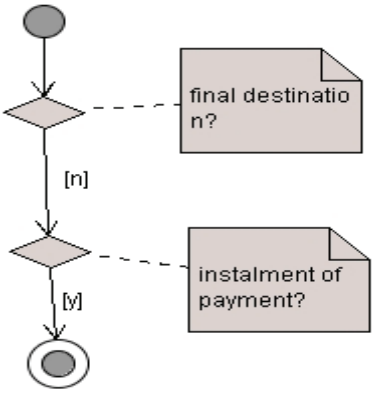

図 6．分岐による抽象化

分岐だけ残した構造を用いてワークフローインスタンスを蓄積した履歴データベースに対し検索を行な

う。[final destination?=n][instalment of payment?=y]と同じ分岐の値を持つワークフローインスタンスがその結果として帰ってくる。

説明の簡単のため、二つの会社 X と Y に契約が締結されたことを X<->Y と記述し、買主を X、売主を Y に予め指定しておく。また X<->Y と P<->Q でワークフローが類似していることは、X の分岐セット⊆P の分岐セットの関係を意味する。

さらに上の取引で「代金支払いの遅延」という例外が発生していると仮定したため、例外内容をキーとして検索を行なった。その結果、A<->D, A<->F という二つのワークフローインスタンスが返ってきた。

A<->D ケースを見てみると、代金支払いの遅延という例外が発生し、会社 A と会社 D の間に締結された契約のルール４を参照し、解決方法ワークフローを適用して問題を解決している。

ルール４：代金支払いが遅延した場合、売主には買主に代金支払いの余裕の時間を与える義務がある。しかし買主が与えたら新しい時間に代金支払いの実行を拒否した場合、売主は他の補償を請求する権利がある。しかも売主は代金支払いの遅延から発生した損害をクレームする権利を失うことなく上の権利を持っている。

また A<->F ケースを見てみると、代金支払いの遅延という例外が発生し、会社 A と会社 F の間に締結された契約のルール５を参照し、解決方法ワークフローを適用して問題を解決している。

ルール５：もし買主が代金支払いの実行に失敗した場合、売主はそれに伴う損害弁償を請求する権利を失うことなく利利子を求める権利がある。

それに対応して A->D ケースでは図７のような解決方法を選択している。つまり代金支払いの遅延に対し、売主は買主に時間の余裕を与える、さらに代金支払いの遅延から発生した損害があれば、買主に損害弁償を行なうことになった。

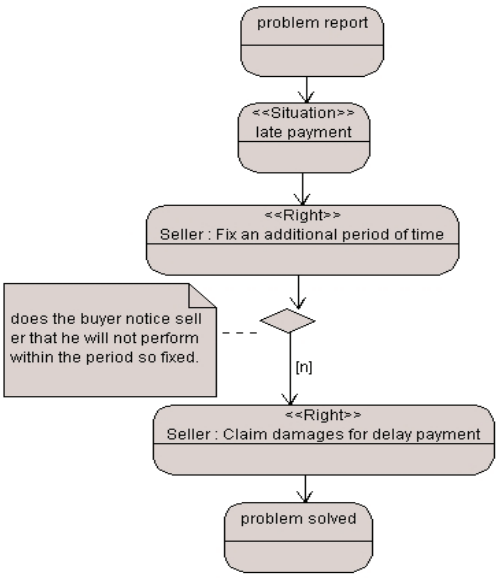

図７．ルールから選択した問題解決

問題解決方法に対応するワークフローが図８のようになる。

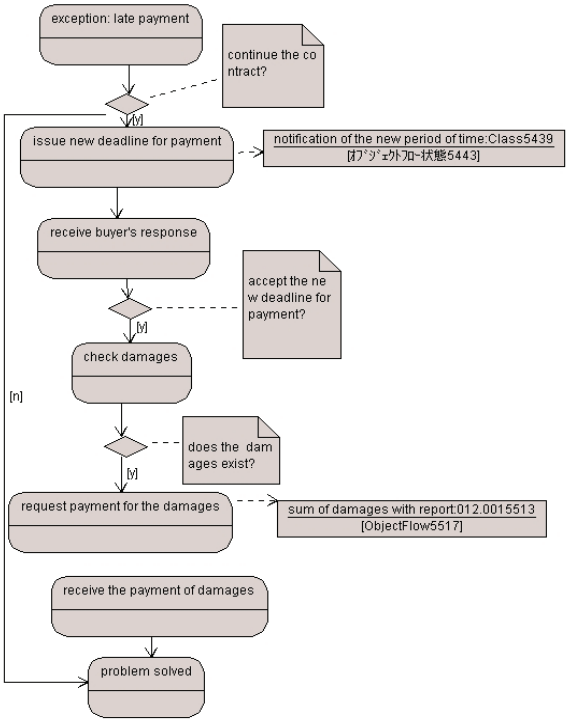

図８．問題解決ワークフロー

5. まとめ

本稿では，ワークフローと電子契約を統合管理するための WCS モデルについて，実行履歴を検索することによりワークフローの例外などの問題解決に役立つ情報を検索する機能について議論した．またそのための基本的手法として共通部分演算とプロセスの抽象化による検索機能を定義した．今後本提案手法の評価を

進めてゆく予定である.

文　献

[1] S. Angelov, P. Grefen; “A Framework for the Analysis of B2B Electronic Contracting Support;” Proc. 4th Edispute Conf., Multidisciplinary Perspectives on Electronic Commerce; Amsterdam, 2001.

[2] Business Process Execution Language for Web Services 1.0. www-106.ibm.com/developerworks/webservices/library/ws-bpel/

[3] M.W.A. Caminada, “Towards a formal model for contract execution,” 4th International Workshop: The Language Action Perspective on Communication Modeling, Copenhagen, 1999.

[4] D. K. W. Chiu, Qing Li, and K. Karlapalem, “Web Interface-Driven Cooperative Exception Handling in ADOME Workflow Management System,” Proc. WISE 2000, pp174-182, Hong Kong, 2000.

[5] CISG Database, Pace University Law School, http://www.cisg.law.pace.edu/cisg/text/cisg-toc.html

[6] Q.Chen, U. Dayal, and M. Hsu, “Conceptual Modeling for Collaborative E-business Processes,” 20th Int. Conf. Conceptual Modeling, pp. 1-16, 2001.

[7] Electronic Business XML. www.ebxml.org/

[8] M. Iwaihara, H. Jiang, and Y. Kambayashi, “An Integrated Model of Workflows, e-Contracts and Solution Implementation,” Proc. ACM Symp. Applied Computing, Organizational Engineering Track, Nicosia, pp. 1390-1395, Mar. 2004.

[9] M. Iwaihara, H. Jiang, and 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.

[10] K. Karlapalem, A.R. Dani, P. R. Krishna, “A Frame Work for Modeling Electronic Contracts”, Conceptual Modeling - ER 2001: 20th Int. Conf. Conceptual Modeling, Nov. 2001.

[11] Konesa. www.canyonblue.com

[12] O. Marjanovic, Z. Milosevic, “Towards Formal Modeling of e-Contracts,” 5th IEEE Int. Enterprise Distributed Object Computing Conf.(EDOC 2001), Sep. 2001.

[13] Z. Milsosevic, S. Gibson, et al., “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.

[14] R. Paige and R. Tarjan, “Three Partition Refinement Algorithms,” SIAM J. Computing, 16:973-988, 1987.

[15] Unified Modeling Language, www.uml.org/

[16] United Nations Convention on Contracts for the International Sale of Goods [CISG], 1980, www.cisg.law.pace.edu/cisg/text/treaty.html