

Title	Colored Petri Netsによるワークフローシステムのモデル化と検証
Author(s)	山本, 豊
Citation	
Issue Date	2008-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/4310">http://hdl.handle.net/10119/4310</a>
Rights	
Description	Supervisor:平石邦彦, 情報科学研究科, 修士

修 士 論 文

Colored Petri Nets による  
ワークフローシステムのモデル化と検証

北陸先端科学技術大学院大学  
情報科学研究科情報システム学専攻

山本 豊

2008年3月

修 士 論 文

# Colored Petri Nets による ワークフローシステムのモデル化と検証

指導教官 平石邦彦 教授

審査委員主査 平石邦彦 教授

審査委員 金子峰雄 教授

審査委員 宮地充子 教授

北陸先端科学技術大学院大学  
情報科学研究科情報システム学専攻

0610091 山本 豊

提出年月: 2008 年 2 月

## 概要

本稿ではワークフローシステムのモデル化と検証を行う。ワークフローシステムは幾つかのシステムが連携して動作する。この分散したシステムは時として予定外の動作を引き起こすことがある。それは定義したビジネスフローの問題、ヒューマンエラーによるワークフローシステム又はデータベース上の間違い等が考えられるが、これらの問題を発見、解決するには極めて困難が伴う。それは分散したシステムであるが故に問題がどのシステム上で発生しているかが分かりにくくなるためである。そこでワークフローシステム上の各要素に対して Colored Petri Nets への変換規則を定義することによってモデル変換を行い、理論的計算モデルである Colored Petri Nets を使用することで検証を行い、問題の解決を図ることが可能であるかを検討する。

# 目次

第1章	はじめに	1
1.1	背景	1
1.2	本研究の目的	1
1.3	論文の構成	2
第2章	ワークフローシステム	3
2.1	Cosminexus ワークフローシステム	3
2.1.1	開発環境	4
2.1.2	実行環境	8
第3章	ワークフローシステムの分析	12
3.1	販売業務ビジネスプロセス概要	13
3.2	各プロセス定義内容	15
3.2.1	データベース	15
3.2.2	フロー定義プロセス	16
3.2.3	帳票発生条件定義プロセス	20
3.2.4	帳票処理プロセス (帳票定義内容)	21
第4章	Colored Petri Nets	24
4.1	Colored Petri Nets について	24
4.2	WDF から Colored Petri Nets への形式的変換規則	26
4.2.1	データベースの変換	27
4.2.2	帳票処理プロセスの変換	29
4.2.3	帳票発生条件定義プロセスの変換	41
4.2.4	フロー定義プロセスの変換	45
4.2.5	変換における追加処理	53
第5章	モデルの検証について	58
5.1	シミュレーションによる検証	58
5.2	状態空間生成ツールによる検証	61

第6章 おわりに	64
6.1 まとめ . . . . .	64
6.2 今後の課題 . . . . .	64
参考文献	64
謝辞	66

# 第1章 はじめに

## 1.1 背景

近年, 企業による会計改竄による不祥事の是正やコンプライアンス (法令遵守) が強く求められており, これらの問題に対処するための金融商品取引法 (通称日本版 S O X 法) が 2008 年 4 月 1 日以降開始の事業年度から, 上場企業においては内部統制の整備および評価が義務化されることとなる. 内部統制の達成には I T 統制による不正を起こすことが困難な情報システムの構築や, 企業の業務の透明性の確保, 適切に情報を管理・運営するためのシステムを構築することが必要不可欠であり, 安心電子社会の実現の為には正当性, 公平性, セキュリティ, 進化性, 退耐事故等を考慮し誤りのない電子システムを構築することが必須である. 本研究の特色としてはまず実際に商用のシステムとして地方自治体, 大手銀行などで用いられている日立 Cosminexus Workcoordinator を使用し, これに対しての適応事例であることが挙げられる. またビジネスプロセスにおいては属性情報や担当者の権限を考慮したものを扱う. 属性情報や担当者の権限などは電子社会, 電子行政におけるビジネスプロセスの記述に必須であるが, 従来研究で取り扱ったものはほとんどない.

## 1.2 本研究の目的

本研究では Colored Petri Nets によるワークフローシステム (日立製作所 Cosminexus) のモデル化による検証を行う. Colored Petri Nets は理論的計算モデルであり, その表現力によりワークフローシステムにおける様々な要素を扱うことが可能なモデルである. ワークフローシステムはワークフローエンジン, ビジネス定義ツール, クライアントアプリケーション, データベース等のシステムの要素の他に, それらのシステム上に定義するビジネスプロセスで重要な要素 (担当者の役職, 帳票のアクセス権限, 属性情報等) がある. これら全てを包含して表現するモデルとして Colored Petri Nets は非常に有効であり, 上記のようなワークフローシステムにおいて複数存在する独立した要素を一つのモデルとして表現可能にするものである. 本研究では商用として使用されている (小売, 銀行, 自治体等) 日立製作所製 Cosminexus ワークフローシステムを用い, 販売 ~ 入金におけるビジネスプロセスを定義し, それを実際にワークフローシステムとして動作させることにより, Colored Petri Nets へ変換するために必要な要素の分析をビジネスプロセス, ワークフローシステム両面から行い, 形式的にモデル変換をする方法を検討する. Colored Petri Nets の記述に

はCPNTOOL<sup>1</sup>を用いることにより,シミュレーションを行う事によって,ビジネスプロセス上で発生する問題の発見と解決方法,その有効性を提示する.

### 1.3 論文の構成

第2章では本研究で使用されるワークフローシステムの説明を行う.第3章ではモデル化に必要なビジネスプロセス定義内容の詳細.第4章では定義したビジネスプロセスの変換方法及び,その説明と補足要素.第5章ではモデル化による検証についての説明.第六章では本研究でのまとめを行う.

---

<sup>1</sup><http://wiki.daimi.au.dk/cpntools/cpntools.wiki>



## 第2章 ワークフローシステム

ワークフローシステムとは複数の部署, 人における業務プロセスを分析し明確化, 規定化を行い, これらの業務の流れを自動化することによって, より円滑かつ迅速な業務を達成するためのシステムである. 本稿では Web システムによるワークフローシステムの構築のために, 日立製作所製 Cosminexus ワークフローシステム<sup>1</sup>を用いて, ビジネスプロセス (業務プロセス) の定義を行う. 本研究におけるビジネスプロセスはこの Cosminexus ワークフローシステムの仕様に基づいたものとする.

### 2.1 Cosminexus ワークフローシステム

Cosminexus ワークフローシステムは既存のソフトウェア資産や, 別々に構築された業務システムなどを統合し, 1つの大きな業務システムとして構築することが出来る. 本研究では Cosminexus ワークフローシステムの Web アプリケーション構築システムの一つである, Business Logic Container(BLC) を使用し, この仕様に則った Web アプリケーション環境を構築する. 以下に必要な開発環境と実行環境の説明をする.

---

<sup>1</sup><http://www.hitachi.co.jp/Prod/comp/soft1/cosminexus/>

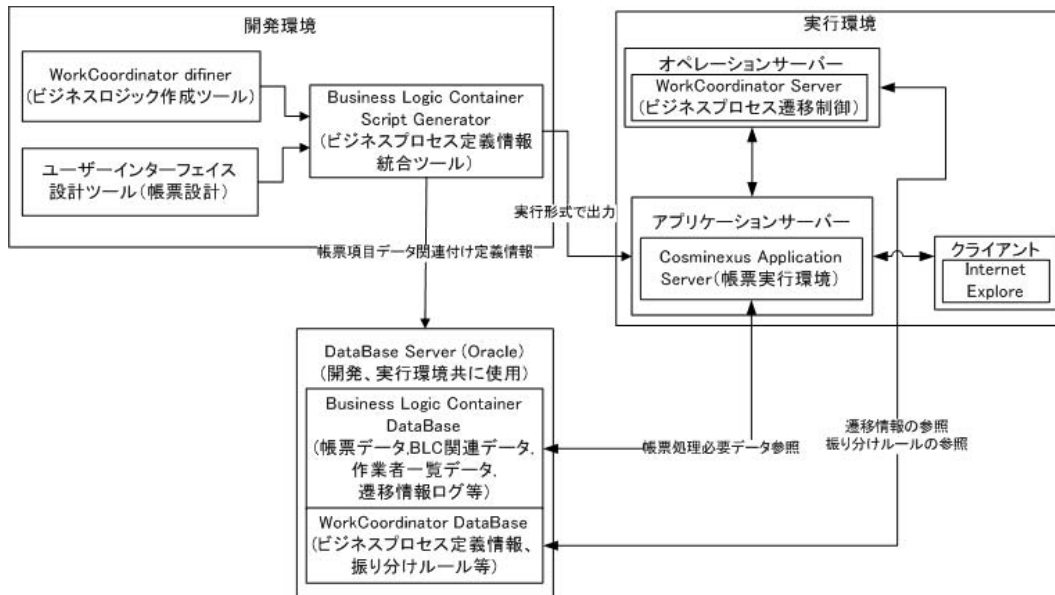


図 2.1: 開発, 実行環境図

### 2.1.1 開発環境

Cosminexus による Web 帳票アプリケーションの開発環境は主に 4 つのソフトウェアから構成される。ビジネスプロセスの定義を行う WorkCoordinator definer. 帳票を設計するユーザーインターフェイス設計ツール. 帳票項目のデータとユーザー情報の管理を行う Database Server(Business Logic Container DataBase). ビジネスプロセス, ユーザーインターフェイス, Database の関連付けを行い, 実行環境で実行できる形に統合する Business Logic Container Script Generator である。

#### WorkCoordinator Definer

WorkCoordinator Definer はビジネスプロセスの定義を行うソフトウェアで, 定義したビジネスプロセスに対して GUI で設計することが出来る。これによりビジネスプロセス構築時, 又は変更時の仕様変更に対して柔軟な対応が取れるようになっている。

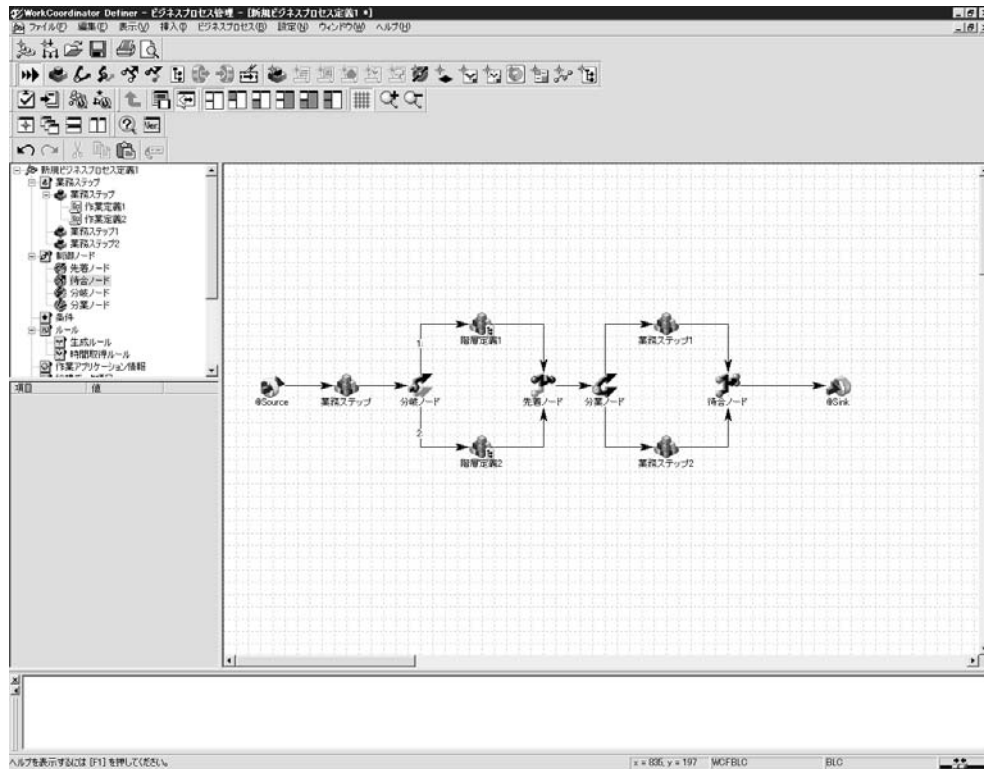


図 2.2: WorkCoordinator Definer

図 2.3 は WorkCoordinator Definer によって定義したビジネスプロセスである。各プロセスで定義される情報について説明する。

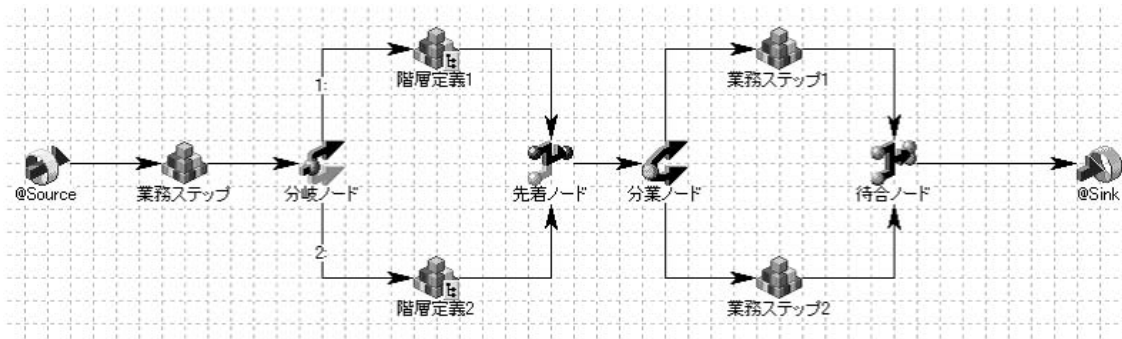


図 2.3: ビジネスプロセス例

- ソースノード



案件の開始を意味し、案件が開始されると、ソースノードから次の業務ステップ又は制御ノードに状態を遷移する。

- シンクノード



案件の終了を意味し、推進された案件は、シンクノードに状態を遷移することで完了する。

- 業務ステップ



業務ステップには作業内容となる作業ステップが登録される。通常、登録された作業は全て実行され、終了すると次のステップに移行する。また発生条件、完了条件を設定することも出来、発生条件はSQLにより指定されデータベースの情報を参照することで実行される。完了条件も同様に実行される。また複数の案件が流れている場合に該当する案件を選択するための振り分けルールを設定を行う。

- 業務ステップ (階層化定義)



ビジネスプロセス定義の一部を階層化したものである。遷移の始点にソースノード、終点にシンクノードが定義される。

- 制御ノード一覧

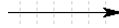
制御ノードは案件の遷移を制御するノードである。幾つかの遷移パターンによって分岐先が決定される。



- 分岐ノード  
次の業務ステップとしてあらかじめ定義された複数の業務ステップから、条件に従って一つの業務ステップを選択し開始する。SQLにより定義された情報により、次の遷移先が決定される。
- 先着ノード  
直前の業務ステップとしてあらかじめ定義された複数の業務ステップのうち、どれか一つが完了した時点で次の業務ステップを開始する。特に定義される情報は無く、自動遷移する。
- 分業ノード  
次の業務ステップとしてあらかじめ定義された複数の業務ステップをすべて開始する。特に定義される情報はなく、自動遷移する。

- 待合ノード  
分業ノードとセットで使用され、複数ある業務ステップのうち全ての業務ステップが完了したタイミングで次のノードへ自動遷移する。

- アロー



上記のプロセス全てから次の遷移先を決定する定義となる。

## Database Server

開発環境では Business Logic Container DataBase を使用する。ビジネスプロセス定義で定義した帳票項目や作業情報(名前, 部署, 所属等)の構築を行い, また Business Logic Container 環境で必要な情報の定義も行う。

## ユーザーインターフェイス設計ツール

Web 帳票となる画面部分を作成するツールである。HTML による記述でユーザーインターフェイス部分を設計する。

## Business Logic Container Script Generator

Business Logic Container Script Generator はユーザーインターフェイス設計ツールによって作成した帳票の項目とデータベースの入出力の関連付けと, WorkCoordinator Definer によって定義したビジネスプロセスの各業務ステップの作業ステップと作成した帳票の登録を行う。これらを統合し実行環境で実行可能な形式 ( JSP ファイル ) として出力する。ファイル出力後, 実行環境で実行させるための登録作業 ( デプロイ ) を行うことによって実行可能となる。

### 2.1.2 実行環境

実行環境は次の 4 つのソフトウェアから成り立つ。ユーザーからの要求に対して Web 帳票を実行する Cosminexus Application Server, ビジネスプロセスの遷移制御を行う WorkCoordinator Server, DataBase Server ( Business Logic Container DataBase 及び WorkCoordinator DataBase ), 完成したシステムに対してアクセスを行うクライアントソフト ( Web ブラウザ ) となる。

## Cosminexus Application Server

Cosminexus Application Server は、クライアントからの要求とデータベースなどの業務システムの処理を橋渡しするためのミドルウェアである。一般的には「Web アプリケーションサーバ」と呼ばれ、Cosminexus ではアプリケーションサーバとして Web サーバ、セッション管理、トランザクション管理などのコンポーネントを提供する。

## WorkCoordinator Server

Business Logic Definer にて定義したビジネスプロセスや振り分けルールの定義に従い、遷移制御を行う。

## DataBase Server

実行環境では Business Logic Container DataBase 及び WorkCoordinator DataBase を使用する。実行の際には帳票の項目データ、ビジネスプロセス遷移の際のログ、担当者一覧が Business Logic Container DataBase に Cosminexus Application Server を介して入出力される。ビジネスプロセス定義情報、振り分けルール等は WorkCoordinator DataBase に WorkCoordinator Server を介して参照される。

## クライアントソフト

Web アプリケーションへのアクセス手段として XML 対応の一般的なブラウザを使用する。本研究には Internet explore を使用する。

## 帳票処理の流れ

Cosminexus ワークフローシステムをユーザーが利用する際には以下の流れになる。これらは、業務プロセス内の作業ステップの条件を満たすまで繰り返され、終了後指定された次の業務ステップ（次の作業）に遷移する。ビジネスプロセスが最終状態（sink ノード）に到着するまでこの処理は繰り返されビジネスプロセス上を遷移していく。

### 1. ログイン

クライアントを使用して Web 帳票にログインを行う。

### 2. 帳票選択

新規に申請を行いたい帳票の選択をするとビジネスプロセスが開始され帳票処理が始まる。処理待ち作業一覧（前のプロセスから送られてきた帳票）より帳票を選択すると帳票処理が行われる。新規申請の場合はプロセス開始時に案件番号が割り振ら



図 2.4: ログイン画面



図 2.5: 処理帳票選択 (新規申請), 処理待ち作業選択



れ、ビジネスプロセス上に複数の案件が発生した場合でも作業者は一意に帳票を処理することが出来る。

### 3. 帳票処理



図 2.6: 帳票処理

選択した帳票内容の処理を行い、次の実行者となる宛先選択を行うことで作業プロセスは終了し、次の作業ステップまたは業務ステップに遷移する。

この一連のプロセス間にはこれだけの処理が行われている。この内容をさらに細かく分けると、ログイン 帳票選択 (又は処理待ち作業選択) 帳票処理 宛先選択 ログアウトとなる。これら各プロセスにおける変換要素の分析を次の章にて行う。

## 第3章 ワークフローシステムの分析

本章では例題として販売業務を想定したビジネスプロセスの定義を行う。Cosminexus ワークフローシステムの各定義部分を分析すると以下のように4つに分類できる。フロー定義プロセスと帳票発生定義プロセスは Business Logic Definer による定義部分となり、帳票処理プロセスはユーザーインターフェイス設計部分とデータベース（帳票項目情報、組織情報）による定義部分となる。データベースは各部分から適宜参照されるデータが格納されている。

これらの定義内容からワークフローシステムの形式的モデル化のために必要な情報の分析を行い、モデル化に必要な情報を記述するためのワークフロー定義フォーマット（WDF: Workflow Definition Format）を提案する。

- フロー定義：ソースノード、シンクノード、業務ステップ、分岐、遷移先定義を定義するプロセスになる。最上位のプロセスとなる。
- 帳票発生条件定義：フロー定義プロセスにおいて定義した業務ステップ内の帳票をどのような順番で発生させるかを定義するプロセスである。中間層のプロセスとなる。
- 帳票処理：フロー定義プロセスに定義された帳票の具体的な処理内容と次プロセス実行者の選択条件を定義するプロセスである。最下層のプロセスとなる。
- データベース：上記のプロセスと連携してデータの入出力を行う。帳票入出力データ項目、及び分岐に関する情報がここに保存され適宜参照される。他にもユーザー情報（部署、役職、名前等）が保存される。

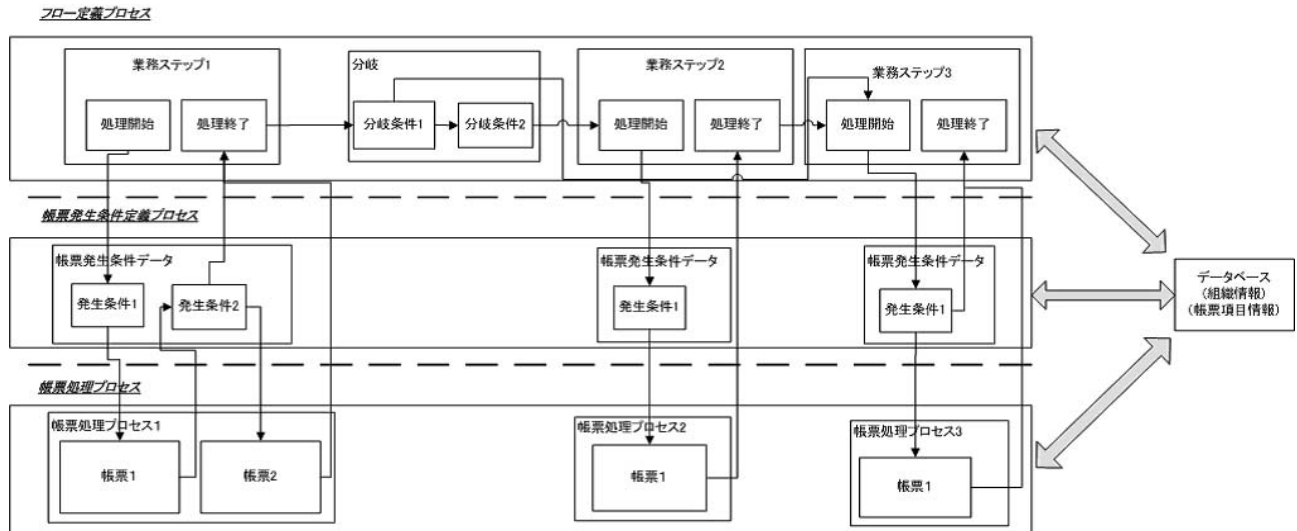


図 3.1: 階層構造図

### 3.1 販売業務ビジネスプロセス概要

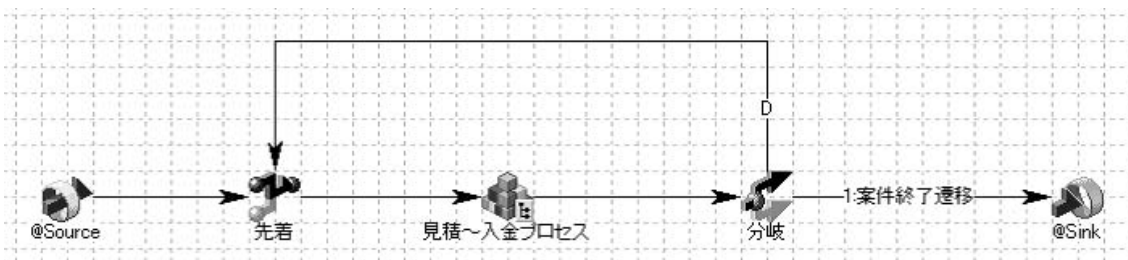


図 3.2: ビジネスプロセス全体図 1

このビジネスプロセスには以下の大きく分けて5つのセクションから構成され、各プロセスの役割は以下のとおりである。

- 見積中セクション

- @source：客先にて注文を取る。その注文内容により一般見積か特別見積、または見積なしの処理であるかを選択する。
- 分岐 (source 後)：見積区分によりどちらかのプロセスへの遷移が決定する。
- 見積中：一般見積、特別見積に対してそれぞれの見積もり処理を行う。一般見積、特別見積共に取った注文に対して納期や見積金額の設定を行う。
- 上位役職承認：見積内容に対して上司がチェックを入れ、許可、不許可の選択を行う。

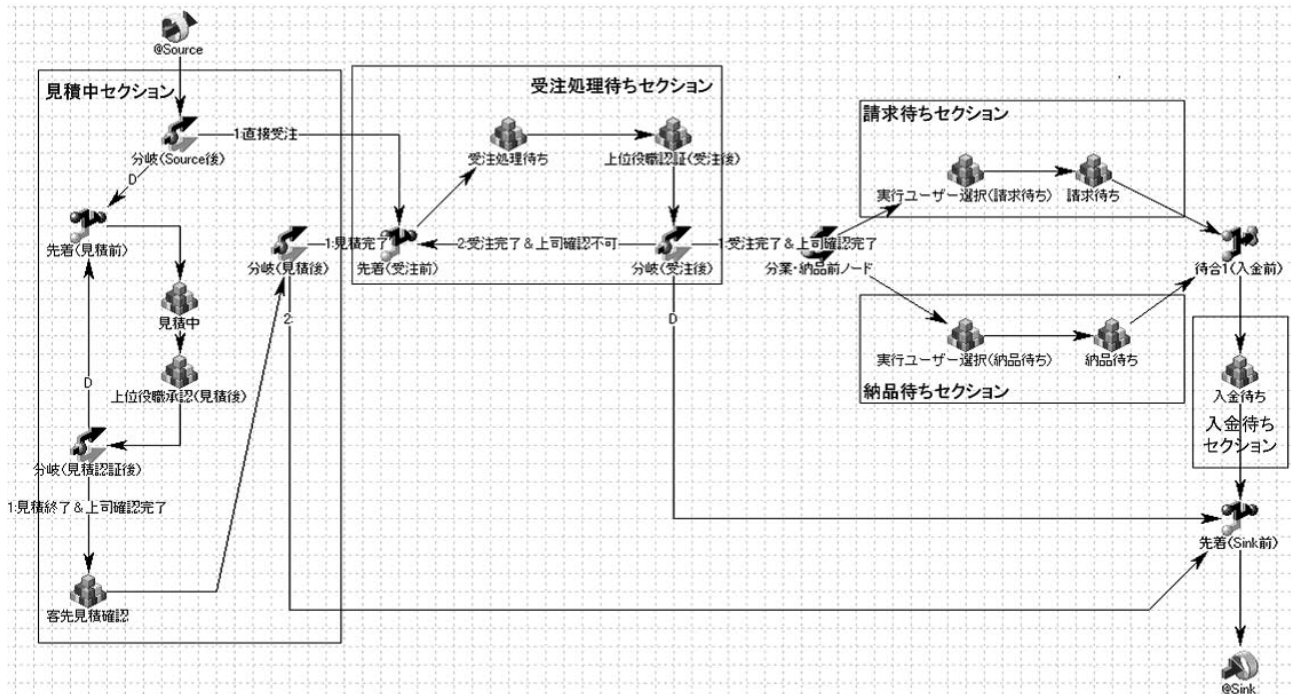


図 3.3: ビジネスプロセス全体図 2

- 分岐 (見積認証後): 上司が許可した場合は次の客先見積確認プロセスへ移行する。不許可の場合は再び見積中プロセスに戻る。
- 客先見積確認: 客先にて見積を提示する。
- 分岐見積後: 客先にて提示した見積の承認があれば、先着 (受注前) に移行する。承認がなければ再び見積処理を行う。
- 受注処理待ちセクション
  - 先着 (受注前): 受注した内容を受けて次のプロセスへ移行する。
  - 受注処理待ち: 受注した注文に対しての処理を行い、受注した注文品が倉庫にあるかどうかの確認を取り、在庫が無ければキャンセルになる。在庫があれば登録作業を行う。
  - 上位役職承認 (受注後) 登録内容について上司が確認をとり許可、不許可の選択を行う。
  - 分岐 (受注後): 登録内容に対して上司が許可した場合は分業納品前ノードへ遷移する。不許可ならば先着 (受注前) に戻り、再び受注処理が行われる。在庫がない場合は終了 (sink ノード) へ遷移する。
  - 分業・納入前ノード: 請求待ち、納入前ノードに同時に自動遷移する前のノードである。

- 請求待ちセクション
  - － 実行ユーザー選択 (請求待ち) : 請求待ちプロセスの作業者の選択を行う。
  - － 請求待ち : 商談内容の売り掛け計上の処理を行い, また商品に対する代金請求の準備作業を行う。
- 納品待ちセクション
  - － 実行ユーザー選択 (納品待ち) : 納品待ちプロセスの作業者の選択を行う。
  - － 納品待ち : 納入待ちは納入日の設定を行い, また納品完了に関する処理も行う。
- 入金待ちセクション
  - － 待合 (入金前) : 請求待ち, 納品待ちの処理の待ち受けを行い, 両方の処理が終了後, 自動遷移する。
  - － 入金待ち : 入金待ちセクションは請求書の発行を行い, 顧客に送付しする。
  - － 先着 (sink 前) : 他のプロセスからの待ち受けを行い, 次のプロセスへ自動遷移する。

## 3.2 各プロセス定義内容

データベースの内容定義 (組織の人物情報 (役職, 部門等)), フロー定義プロセス (遷移先の決定), 帳票発生条件定義プロセス (帳票発生条件), 帳票処理プロセス (帳票定義内容, 宛先人物選択条件) についてそれぞれ説明を行う。全てのプロセスの定義については付録を参照されたい。

### 3.2.1 データベース

データベースには帳票ごとに設定される内容 (帳票項目に関するデータの格納場所) と事前に登録しておく内容 (組織情報に関するデータ) の2つがあるが, ここでは後者の事前登録情報に関するデータベースの説明を行う。前者の内容に関しては各帳票処理の際に定義し説明を行うものとする。また本稿で使用されるすべてのデータベース定義内容は付録に掲載する。

表 3.1 はユーザーの属性に関する定義データ一覧である。これらのデータは帳票作成前にあらかじめ定義を行っておき, ワークフローシステムのログイン時や帳票処理時の宛先参照に使用される。

ID	名字	名前	名字(読み仮名)	名前(読み仮名)	所属(ID表記)	所属部門	役職
SE0000	商談	一郎	SYOUDAN	ICHIROU	EIGYO0000	営業部	平社員
SE0001	商談	二郎	SYOUDAN	JIROU	EIGYO0000	営業部	平社員
SE0002	商談	三郎	SYOUDAN	SABUROU	EIGYO0000	営業部	平社員
SE0003	商談	課長	SYOUDAN	KATYOU	EIGYO0000	営業部	課長
SS0000	商品	一郎	SYOUSHIN	ICHIROU	SYUKKA0000	出荷部	平社員
SS0001	商品	二郎	SYOUSHIN	JIROU	SYUKKA0000	出荷部	平社員
SS0002	商品	三郎	SYOUSHIN	SABUROU	SYUKKA0000	出荷部	平社員
SS0003	商品	課長	SYOUSHIN	KATYOU	SYUKKA0000	出荷部	課長
BK0000	部門	一郎	BUMON	ICHIROU	KEIRI0000	経理部	平社員
BK0001	部門	二郎	BUMON	JIROU	KEIRI0000	経理部	平社員
BK0002	部門	三郎	BUMON	SABUROU	KEIRI0000	経理部	平社員
BK0003	部門	課長	BUMON	KATYOU	KEIRI0000	経理部	課長

表 3.1: ユーザーデータ一覧

### 3.2.2 フロー定義プロセス

フロー定義プロセスに定義されるデータはそのプロセスから次のどのプロセスに遷移するかというデータが定義される。これは WorkCoordinator Definer の仕様で言うと各ノードとアローをセットにしたものになる。各ノードにどのような定義内容が成されるかを説明する。

#### 1. ソースノード

@source(見積～入金プロセス内)

遷移先及び参照番号	備考	参照番号
分岐 (Source 後),1	分岐プロセス一覧へ	1

表 3.2: @source(見積～入金プロセス内)

ソースノードはプロセスが開始されるノードである。このプロセスでは分岐プロセスの分岐 (source 後) に遷移する。通常ソースノードは遷移先しか定義されないが WorkCoordinator Definer の仕様により、一番最初のソースノードのみ以下に記す業務ノードと同等の物が定義される。階層化定義されたプロセスの初めのプロセスに定義されるソースノードのみこの仕様になる。参照番号とは表 3.5 のように遷移した先に複数の条件がある場合に参照するものを指定するものである。

#### 2. シンクノード

シンクノードはそのビジネスプロセスが終了したことを表すノードである。通常は

遷移先は定義されないが、階層化されたプロセスに定義されるシンクノード表 3.4 は上位層への遷移先プロセスが定義され、階層化プロセスが終了したことを意味する。

@sink

遷移先及び参照番号	備考	参照番号
無し	ビジネスプロセス終了	1

表 3.3: @sink

sink(見積 ~ 入金プロセス内)

遷移先及び参照番号	備考	参照番号
分岐,1	分岐プロセス一覧へ, 上位階層へ	1

表 3.4: @sink(見積 ~ 入金プロセス内)

### 3. 業務ステップ

見積中

遷移先及び参照番号	備考	参照番号
見積中,1	帳票発生条件データ一覧へ, 処理開始	1
上位役職承認(見積後),1	業務ステップ一覧へ, 処理終了	2

表 3.5: 見積中

業務ステップには呼び出すべき業務ステップの定義を行う。登録された帳票は帳票発生条件定義プロセスに遷移し定義された帳票を呼び出す

#### 4. 業務ステップ (階層化定義)

階層化定義による業務ステップは新たに業務ステップ下にフロー定義プロセスを生成する。生成したフロー定義プロセスのソースノードに遷移する定義がなされる。階層化されたプロセスから再び戻る場合はシンクノードに到達することによって戻ることが出来る。

見積～入金プロセス

遷移先及び参照番号	備考	参照番号
@source(見積～入金プロセス内),1	ソースノード一覧へ, 下位階層へ	1

表 3.6: 見積～入金プロセス

#### 5. 制御ノード

それぞれの制御ノードに定義されている内容を説明する。

- 分岐ノード

分岐ノードには次のプロセスに遷移するための分岐条件が定義される。まず遷移してきたプロセスがどの参照番号を参照しているのかを確認し、その後分岐条件により遷移先が決定される。

必要 DB 項目一覧

エンティティ:商談

テーブル名: SYOUDAN

分岐条件	遷移先及び参照番号	備考	参照番号
MITSUMORIKUBUN='0'	先着(受注前),1	見積省略='0', 分岐プロセス一覧へ	1
MITSUMORIKUBUN='1'	先着(見積前),1	一般見積='1', 分岐プロセス一覧へ	1
MITSUMORIKUBUN='2'	先着(見積前),1	特別見積='2', 分岐プロセス一覧へ	1

表 3.7: 分岐 (source 後)

分岐(受注後)

必要 DB 項目一覧

エンティティ:受注

テーブル名: JUTYU

分岐条件	遷移先	備考	参照番号
JUTYUKANRYOUBI IS NULL	先着(sink前),1	受注キャンセル, 分岐プロセス一覧へ	1
JUTYUKANRYOUBI IS NOT NULL	分岐(受注後),2	受注OK, 参照順位2へ	1

エンティティ:受注明細

テーブル名: JUTYUMEISAI

分岐条件	遷移先	備考	参照番号
JYOUSHIKAKUNIN_FLAG = 'ON'	分業・納品前ノード,1	受注確認OK='ON', 分岐プロセス一覧へ	2
JYOUSHIKAKUNIN_FLAG = 'OFF'	先着(受注前),1	再受注確認='OFF', 分岐プロセス一覧へ	2

表 3.8: 分岐 (受注後)



- 先着ノード  
先着ノードに分岐条件は定義されず、遷移してきた時に自動的に定義された遷移先に遷移する。

先着（見積前）

分岐条件	遷移先	備考	参照番号
無し	見積中,1	強制遷移, 業務ステップ一覧へ	1

表 3.9: 先着（見積前）

- 分業ノード  
分業ノードは次に定義されている複数のプロセスに同時に遷移する。  
この場合は実行ユーザー選択（請求待ち）、実行ユーザー選択（納品待ち）両方に遷移を行う。

分業（分業・納品前ノード）

分岐条件	遷移先	備考	参照番号
無し	実行ユーザー選択（請求待ち）	同時強制遷移, 業務ステップ一覧へ	1
無し	実行ユーザー選択（納品待ち）	同時強制遷移, 業務ステップ一覧へ	1

表 3.10: 分業（分業・納品前ノード）

- 待合ノード  
待合ノードは直前の複数のプロセスが全て遷移してきた時にのみ、次のプロセスに遷移する。この場合は直前の2つのプロセスが遷移してきた時に入金待ちプロセスに遷移する。

待合1（入金前）

分岐条件	遷移先	備考	参照番号
無し	入金待ち,1	待合ノード, 業務ステップ一覧へ, 直前のプロセス2つ待ち	1

表 3.11: 待合1(入金前)

### 3.2.3 帳票発生条件定義プロセス

見積セクションのノード「見積中」の定義内容について説明を行う。直前のプロセスの参照番号に従い、分岐条件を満たす帳票に遷移を行う。この見積中プロセス表 3.12 には作業プロセスとして一般見積と特別見積の二つのプロセスが登録されている。これらの作業の発生条件として直前の業務プロセス@source によって書き込まれた内容を参照することにより、どちらの帳票が開始されるかが決定される。この場合、SYOUDAN テーブルの BUNKIJYOUKEN が 1 だと一般帳票見積作成の帳票が呼び出され、2 だと特別見積作成の帳票処理が呼び出される。

必要 DB 項目一覧

エンティティ:商談

テーブル名: SYOUDAN

分岐条件	発生帳票名 (又は遷移先)	備考	参照番号
MITSUMORIKUBUN='1'	見積中：一般見積作成	一般見積='1', 帳票データ一覧へ	1
MITSUMORIKUBUN='2'	見積中：特別見積作成	特別見積='2', 帳票データ一覧へ	1

表 3.12: 帳票発生条件データ (見積中)

参照番号が複数ある場合は分岐ノードと同じように直前のプロセスがどの参照番号を参照しているかチェックを行ってから分岐条件を参照し、次のプロセスへ遷移を行う。

必要 DB 項目一覧

分岐条件	発生帳票名 (又は遷移先)	備考	参照番号
無し	受注確認	無条件発生, 帳票データ一覧へ	1
エンティティ:受注 テーブル名: JUTYU			
分岐条件	発生帳票名 (又は遷移先)	備考	参照番号
JUTYUKANRYOUBI IS NOT NULL	受注登録	受注完了日=現日時, 帳票データ一覧へ	2
JUTYUKANRYOUBI IS NULL	上位役職認証 (受注後)	受注完了日=NULL, フロー定義データへ (業務ステップ一覧)	2

表 3.13: 受注処理待ち

### 3.2.4 帳票処理プロセス (帳票定義内容)

表 3.14 見積中ステップに定義する作業ステップ：一般見積帳票のデータ一覧である。これらの定義をデータベースに登録する。

必要 DB 項目一覧  
 エンティティ: 見積  
 テーブル名: MITSUMORI

属性名	カラム名	キー	データ型	入出力属性	備考
商談番号	SYOUDANBANGOU	主キー	文字列	出力	案件キー
見積完了フラグ	MITSUMORIKANRYOUFLAG		文字列	出力	見積完了='ON'
見積納期	MITSUMORINOUKI		文字列	出力	
見積名称	MITSUMORIMEISYOU		文字列	出力	
見積確度	MITSUMORIKAKUDO		文字列	出力	

エンティティ: 見積明細  
 テーブル名: MITSUMORIMEISAI

属性名	カラム名	キー	データ型	入出力属性	備考
見積明細番号	MITSUMORIMEISAIBANGOU	主キー	文字列	出力	案件キー
商談番号	SYOUDANBANGOU		文字列	出力	
商品コード	SYOUSHINCODE		文字列	出力	
見積数量	MITSUMORISUURYOU		文字列	出力	
見積金額	MITSUMORIKINGAKU		文字列	出力	

表 3.14: 帳票項目データ (見積中:一般見積)

次のプロセスにおける実行人物は宛先人物検索条件から選択される。このデータを元にデータベースのユーザーデータ一覧から参照を行う。

部門	役職	次処理帳票	備考
営業部	課長	上位役職帳票確認 (一般見積)	

表 3.15: 宛先人物検索条件

これらの定義内容に基づいた画面の設計をユーザーインターフェイス設計ツールより作成する。

**販売業務プロセス**

■見積中プロセス(商品一覧データ参照用)

◆商品エンティティ

商品一覧	DESKTOPPC0001 DESKTOPPC0002 DESKTOPPC0003 DIGITALCAM0000 DIGITALCAM0001 DIGITALCAM0002 NOTEPC0001 NOTEPC0002 NOTEPC0003
------	---

■商談プロセス(商談中\_見積申請)

◆商談エンティティ

担当者コード	
顧客コード	20071007081303
見積区分選択	1-一般見積
商談名	PC見積
顧客予算	1,000,000

■見積入金プロセス(見積中\_一般見積作成)

◆見積エンティティ

見積納期(顧客希望納期)	2008/1/23
見積名称	PC見積
見積確度	%

■見積入金プロセス(見積中\_一般見積作成)

◆見積明細エンティティ

見積明細番号	
商談番号	
商品コード	NOTEPC0001
見積数量	2
見積金額	300,000

コメント  
記述欄

宛先選択: [宛先を選択してください。]

承認 保留 戻る

図 3.4: 見積中：一般見積帳票

宛先選択: [宛先を選択してください。]

承認 保留

宛先を選択してください。
商談 一郎(営業部 平社員)
商談 二郎(営業部 平社員)
商談 三郎(営業部 平社員)

図 3.5: 見積中：一般見積帳票宛先一覧

次の二つの定義は帳票の作成には使わないが、この帳票における処理者が誰になるか、帳票処理終了後どのプロセスへ遷移するか、ということを確認するために必要になる要素になる。また Colored Petri Nets へ変換する際には必須になる項目でもある。

この帳票にアクセス可能な人物の条件を定義する。

部門	役職	備考
営業部	平社員	

表 3.16: アクセス可能条件（見積中：一般見積作成）

次の遷移先の定義を行う。この場合は他に処理する帳票が無いのでフロー定義プロセスの@source に戻る。もし他にこの業務ステップにおいて処理する帳票がまだ存在する場合は帳票発生条件定義プロセスに戻り再び帳票発生条件の参照を行う。

遷移先及び参照番号	備考
Source,2	フロー定義データ一覧へ（業務ステップ一覧へ）

表 3.17: 遷移先一覧

## 第4章 Colored Petri Nets

定義内容により初期プロセスより順にフロー定義プロセス, 帳票発生条件定義プロセス, 帳票処理プロセスを定義に伴い遷移していくことで最終状態に到達することが出来る. 全ての定義内容に共通するのは遷移方向であり, その中で特に重要なのは分岐情報の参照による遷移方向の決定になる. ペトリネットにおけるデータの参照の方法, そして各定義条件におけるモデル変換の方法を説明する.

### 4.1 Colored Petri Nets について

Petri Nets は離散分散システムを数学的, 視覚的に表現する手法である. トランジション, プレースの2つのノードから成る有向二部グラフであり, ノード同士はアークによって結合される. 初期状態としてプレースにはトークンが与えられ, トランジションが発火することにより, 連結先のプレースに遷移する. このときアークの本数により発生するトークンの数が決定される. Colored Petri Nets は Petri Nets のトークンに対してトークンの型(カラー)を定義し, 属性を持たせることによってトランジションの発火規則を各カラー毎に独立して定めることを可能にする. 本研究では更に階層化定義を可能にした HCPN(Hierarchy Colored Petri Nets) を扱うことが出来る CPN TOOLS<sup>1</sup>を用いることによってモデル化を行う.

---

<sup>1</sup><http://wiki.daimi.au.dk/cpntools/cpntools.wiki>

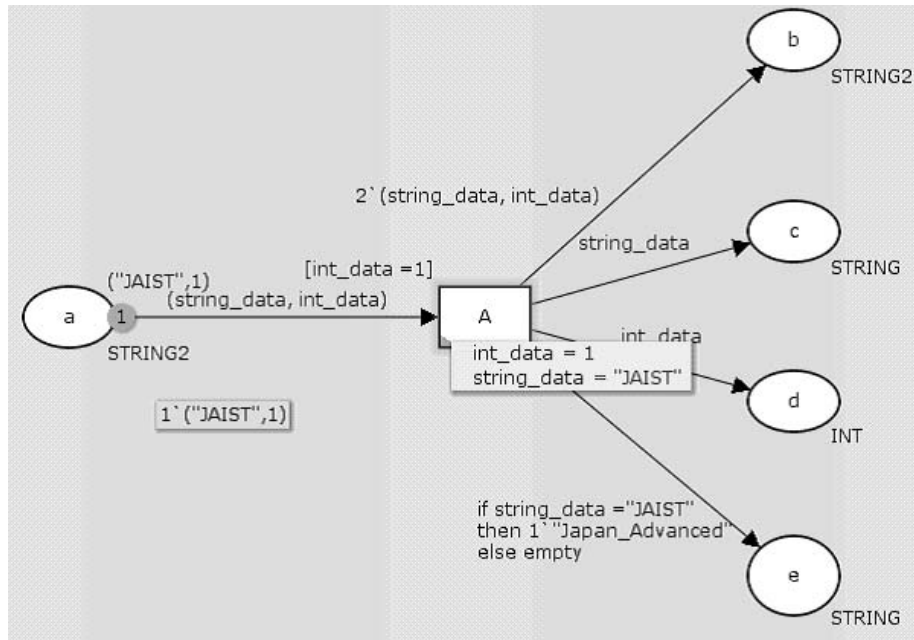


図 4.1: Colored Petri Nets(発火前)

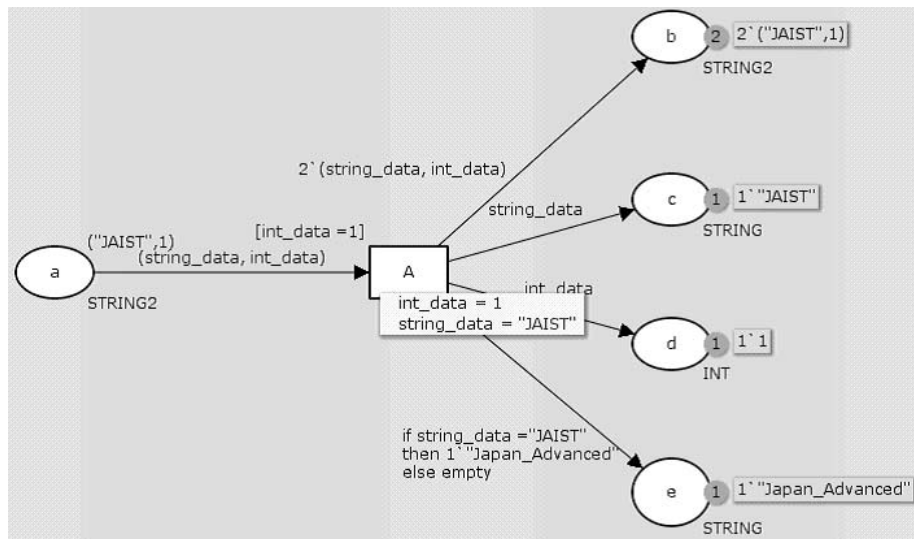


図 4.2: Colored Petri Nets(発火後)

## 4.2 WDF から Colored Petri Nets への形式的変換規則

ビジネスプロセス定義時にフロー定義プロセス、帳票発生条件定義プロセス、帳票処理プロセス、データベースの4つに分け定義を行ったが、それぞれの階層における定義内容の分析を行い Colored Petri Nets への形式的変換規則を検討する。4つのプロセスの各定義内容に対して機械的にモデル化可能になるテンプレートを用意し、適応することで自動変換可能であるという事を示す。

変換作業に伴い最下層である帳票処理プロセスにて行った変換内容(分岐データ)が上位のプロセスでも使用されるため下階層の変換規則から説明する。

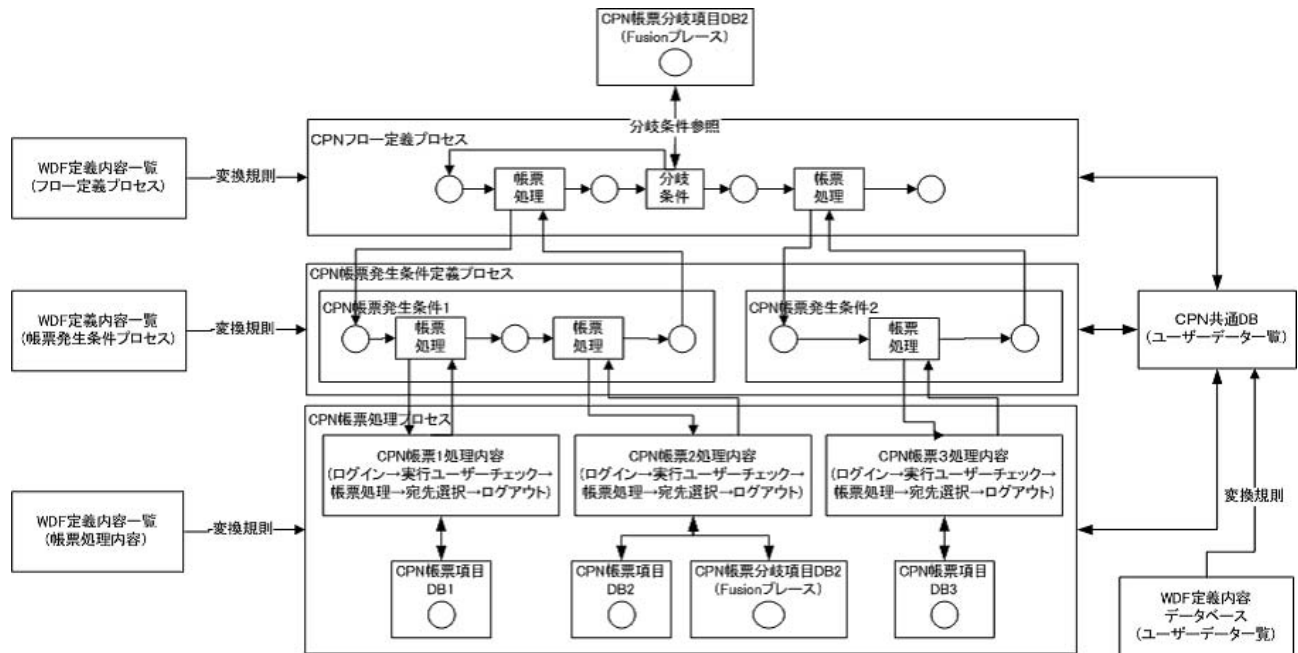


図 4.3: 変換概念図



## 4.2.1 データベースの変換

- ユーザーデータベースの変換

データベースの内容には2種類あり、1つは組織情報に関するデータともう1つは帳票項目に関係するデータになる。ここでは組織情報に関するデータの変換について説明し、帳票項目で定義するデータについては後の帳票処理プロセス時に説明を行う。

ID	名字	名前	名字(読み仮名)	名前(読み仮名)	所属(ID表記)	所属部門	役職
SE0000	商談	一郎	SYOUDAN	ICHIROU	EIGYO0000	営業部	平社員
SE0001	商談	二郎	SYOUDAN	JIROU	EIGYO0000	営業部	平社員
SE0002	商談	三郎	SYOUDAN	SABUROU	EIGYO0000	営業部	平社員
SE0003	商談	課長	SYOUDAN	KATYOU	EIGYO0000	営業部	課長
SS0000	商品	一郎	SYOUDAN	ICHIROU	SYUKKA0000	出荷部	平社員
SS0001	商品	二郎	SYOUDAN	JIROU	SYUKKA0000	出荷部	平社員
SS0002	商品	三郎	SYOUDAN	SABUROU	SYUKKA0000	出荷部	平社員
SS0003	商品	課長	SYOUDAN	KATYOU	SYUKKA0000	出荷部	課長
BK0000	部門	一郎	BUMON	ICHIROU	KEIRI0000	経理部	平社員
BK0001	部門	二郎	BUMON	JIROU	KEIRI0000	経理部	平社員
BK0002	部門	三郎	BUMON	SABUROU	KEIRI0000	経理部	平社員
BK0003	部門	課長	BUMON	KATYOU	KEIRI0000	経理部	課長

表 4.1: ユーザーデータ一覧

見積開始: @Source

この帳票にアクセス可能な条件

部門	役職	備考
営業部	平社員	

表 4.2: 帳票におけるアクセス可能条件例

Fusion プレースは同一の名称の Fusion プレース上にトークンを共有できるプレースである。この Fusion プレース上にユーザー情報一覧と同じトークンを持たせることによって共有するデータベースの内容を表現する。今回の場合は使用されないデータの変換は行っておらず、使用するユーザーデータ (ID, 名前, 所属部門, 役職) のみ変換を行っている。各人物には帳票定義内容時に定義したアクセス可能条件をユーザー情報内に定義している。この場合は@source プロセスにアクセス可能な人物は部門が営業部、役職が平社員になるので SE0000, SE0001, SE0002 がその条件に当てはまる。これよりユーザーデータ (ID, 名前, 所属部門, 役職) + 帳票名称 (この場合 source) のデータ列のトークンを定義している。このアクセス可能条件はログイン時に必要な定義となり、権限を持たない人間によるログインを制限するものである。

```

1`("SE0000","Syoudan_Ichirou","eigyo","hira","source")++
1`("SE0000","Syoudan_Ichirou","eigyo","hira","mitsumori(ippan)")++
1`("SE0000","Syoudan_Ichirou","eigyo","hira","mitsumori(tokubetsu)")++
1`("SE0000","Syoudan_Ichirou","eigyo","hira","kyakusaki_mitsumori_kakunin(ippan)")++
1`("SE0000","Syoudan_Ichirou","eigyo","hira","kyakusaki_mitsumori_kakunin(tokubetsu)")++
1`("SE0000","Syoudan_Ichirou","eigyo","hira","jyutyu_kakunin")++
1`("SE0000","Syoudan_Ichirou","eigyo","hira","jyutyu_touroku")++
1`("SE0000","Syoudan_Ichirou","eigyo","hira","nouhin_sijibi")++
1`("SE0001","Syoudan_Jirou","eigyo","hira","source")++
1`("SE0001","Syoudan_Jirou","eigyo","hira","mitsumori(ippan)")++
1`("SE0001","Syoudan_Jirou","eigyo","hira","mitsumori(tokubetsu)")++
1`("SE0001","Syoudan_Jirou","eigyo","hira","kyakusaki_mitsumori_kakunin(ippan)")++
1`("SE0001","Syoudan_Jirou","eigyo","hira","kyakusaki_mitsumori_kakunin(tokubetsu)")++
1`("SE0001","Syoudan_Jirou","eigyo","hira","jyutyu_kakunin")++
1`("SE0001","Syoudan_Jirou","eigyo","hira","jyutyu_touroku")++
1`("SE0001","Syoudan_Jirou","eigyo","hira","nouhin_sijibi")++
1`("SE0002","Syoudan_Saburou","eigyo","hira","source")++
1`("SE0002","Syoudan_Saburou","eigyo","hira","mitsumori(ippan)")++
1`("SE0002","Syoudan_Saburou","eigyo","hira","mitsumori(tokubetsu)")++
1`("SE0002","Syoudan_Saburou","eigyo","hira","kyakusaki_mitsumori_kakunin(ippan)")++
1`("SE0002","Syoudan_Saburou","eigyo","hira","kyakusaki_mitsumori_kakunin(tokubetsu)")++
1`("SE0002","Syoudan_Saburou","eigyo","hira","jyutyu_kakunin")++
1`("SE0002","Syoudan_Saburou","eigyo","hira","jyutyu_touroku")++
1`("SE0002","Syoudan_Saburou","eigyo","hira","nouhin_sijibi")++
1`("SE0003","Syoudan_Katyoku","eigyo","katyoku","jyoui_yakusyoku_syounin(ippan)")++
1`("SE0003","Syoudan_Katyoku","eigyo","katyoku","jyoui_yakusyoku_syounin(tokubetsu)")++
1`("SE0003","Syoudan_Katyoku","eigyo","katyoku","jyutyu_meisai_kakunin_ninsyou")++
1`("SE0003","Syoudan_Katyoku","eigyo","katyoku","jikkou_user_select(seikyu)")++
1`("SE0003","Syoudan_Katyoku","eigyo","katyoku","jikkou_user_select(nouhin)")++
1`("SE0003","Syoudan_Katyoku","eigyo","katyoku","bungyou_nouhinmae_node")++
1`("BK0000","Bumon_Ichirou","keiri","hira","urikake_seikyuRecord_sakusei")++
1`("BK0000","Bumon_Ichirou","keiri","hira","seikyuusyo_syutsuryoku")++
1`("BK0000","Bumon_Ichirou","keiri","hira","nyuukin_touroku")++
1`("BK0001","Bumon_Jirou","keiri","hira","urikake_seikyuRecord_sakusei")++
1`("BK0001","Bumon_Jirou","keiri","hira","seikyuusyo_syutsuryoku")++
1`("BK0001","Bumon_Jirou","keiri","hira","nyuukin_touroku")++
1`("BK0003","Bumon_Katyoku","keiri","katyoku","urikake_seikyuRecord_sakusei")++
1`("BK0003","Bumon_Katyoku","keiri","katyoku","seikyuusyo_syutsuryoku")++
1`("BK0003","Bumon_Katyoku","keiri","katyoku","nyuukin_touroku")++
1`("SS0000","Syouhin_Ichirou","syukka","hira","nounyu_kosagyoku")++
1`("SS0001","Syouhin_Jirou","syukka","hira","nounyu_kosagyoku")++
1`("SS0002","Syouhin_Saburou","syukka","hira","nounyu_kosagyoku")++
1`("SS0003","Syouhin_Katyoku","syukka","katyoku","nounyu_kosagyoku")++
1`("DAMY","DAMY","DAMY","DAMY","Sink")

```



図 4.4: CPN によるユーザーデータ DB 表現

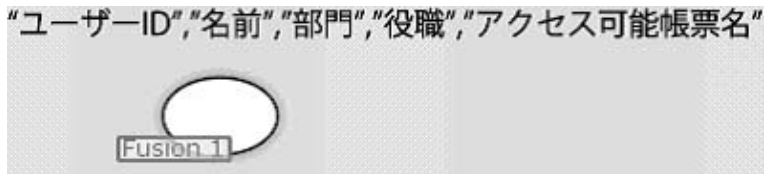


図 4.5: CPN によるユーザーデータ DB 表現内容

## 4.2.2 帳票処理プロセスの変換

帳票処理プロセスの変換では帳票処理のモデル変換を行う,これには帳票の入力表示項目とユーザーによる入力,更にシステムに依存する処理(ログイン,ログアウト等)の処理も含まれる.これによりユーザーの入力項目によって発生する遷移変化をもシミュレートできる.これらの入力項目のモデル化は網羅的に全ての経路を辿るには必須であり重要な項目になる.

### 変換処理

帳票処理が開始されると(ログイン 帳票選択に伴う処理 帳票入出力に関する処理 宛先選択 ログアウト)の順に処理が行われる.これらに対してモデル変換を行うためにビジネスプロセス定義内容がどのように変換されるのかを説明する.

#### 1. CPN 帳票処理全体図

帳票処理が始まると案件番号のトークンが順に階層化された各トランジションに遷移する.トランジションには帳票処理内容が定義されている.

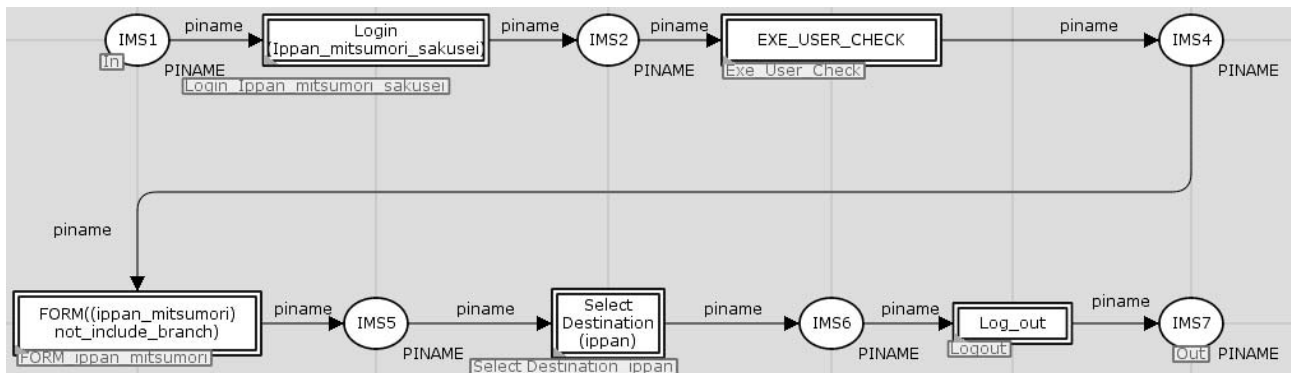


図 4.6: CPN 帳票処理遷移図

## 2. ログイン

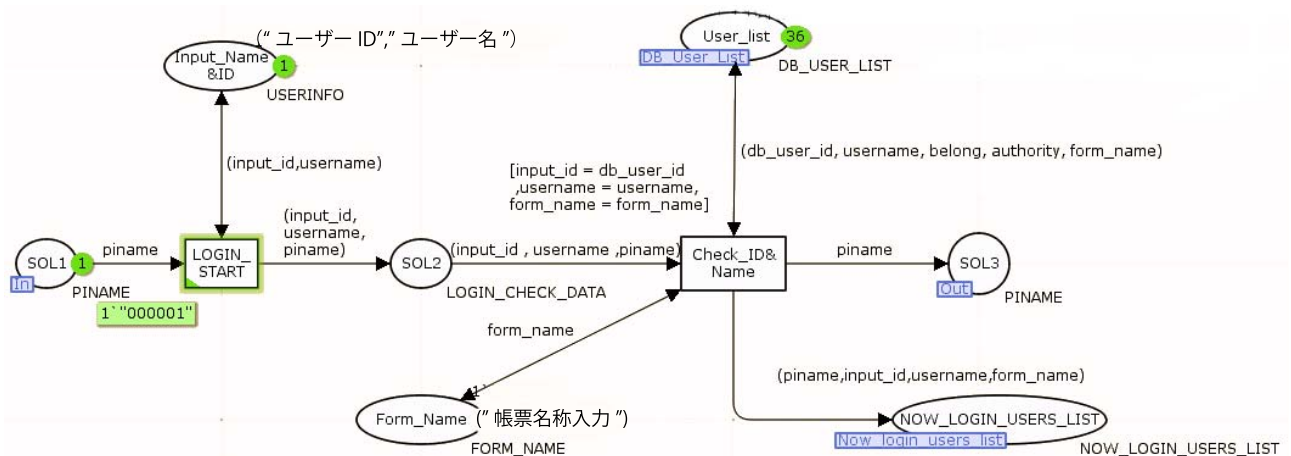


図 4.7: CPN ログインテンプレート全体図

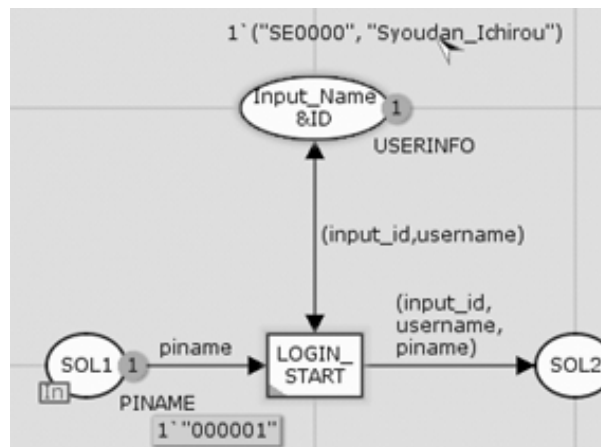


図 4.8: CPN ログイン ID 入力例

ログインはユーザーがIDの入力と名前からの入力で行われる。ログインプロセスに入るとLOGINSTART トランジションの発火により、ユーザー ID、ユーザー名、案件番号を含んだトークンが次のログインチェックの項目に遷移する。

遷移してきたトークンと Form\_Name プレース内の帳票名称（各帳票の変換時に帳票名を入力）の参照により、ユーザーリスト DB からこの帳票にログイン可能な権限者であるかのチェックを行う。DB 内のユーザー情報を比べ同じものが参照できれば、CheckID & Name トランジションが発火しログイン完了となる。ログイン完了と同時に現在ログイン中のユーザーとして一時データ領域 (NOW\_LOGIN\_USERS\_LIST) に案件番号、ユーザー ID、帳票名が保存される。これは次の帳票選択処理に使用される。

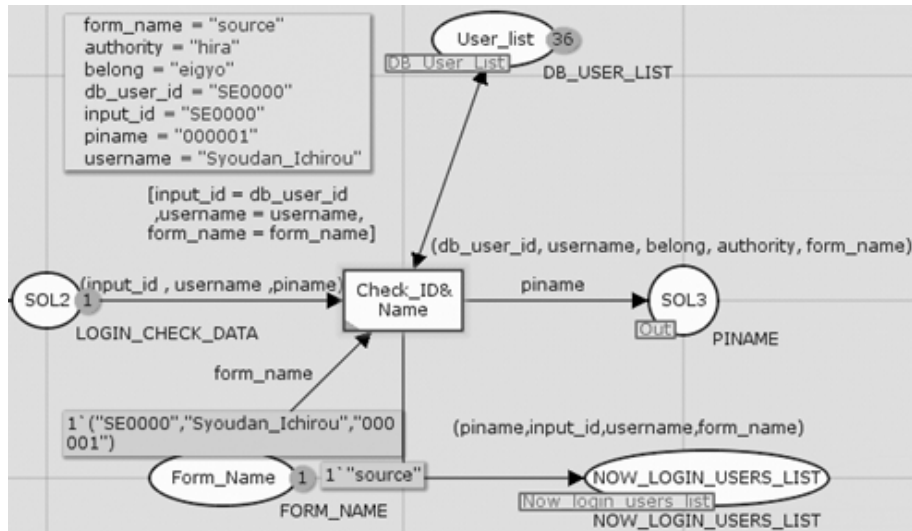


図 4.9: CPN ログイン ID 入力例

3. 帳票選択（処理待ち作業選択）における実行ユーザーチェック処理帳票選択時に行われる内容を定義する。直前の作業プロセスの宛先指定時に自分宛てに送られてきたものであるかのチェックを行う。一番最初の作業プロセスでは宛先の指定がないためこのチェックは行われない。

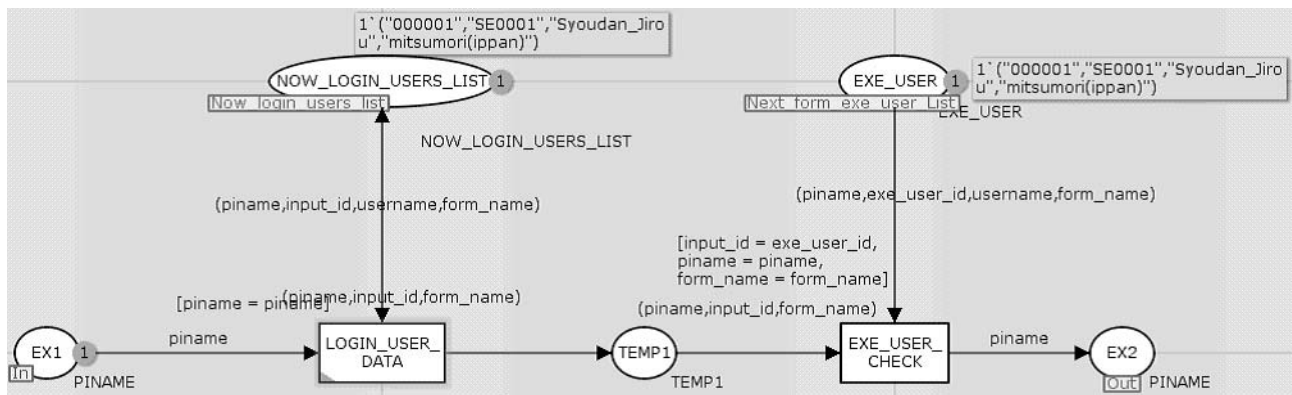


図 4.10: CPN 実行ユーザーチェックテンプレート全体図

まずログインデータの参照を行う。ログインデータのトークンには案件番号、ユーザーID、現在ログイン中の帳票名が含まれている。

EXE\_USER プレースには直前の作業プロセスで指定された宛先のユーザー情報等がトークンとして保存されている。これと先ほどのログインデータとの照合を行うことでチェックを行う。ここでは特にテンプレートに定義する内容は無い。

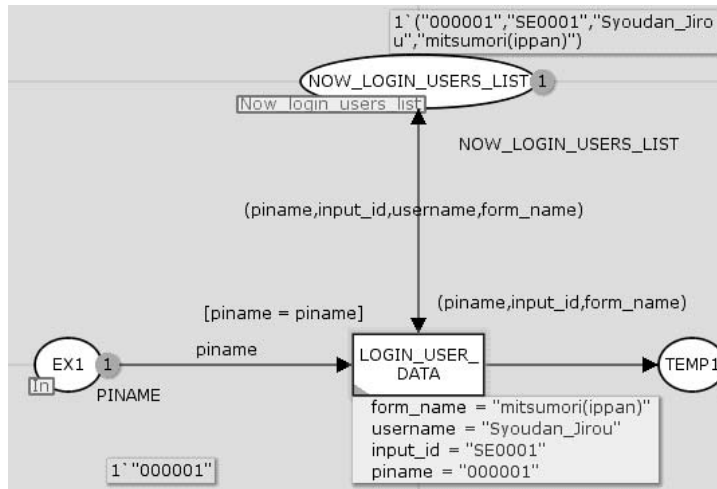


図 4.11: CPN 実行ユーザーチェック例 (ログインデータ参照)

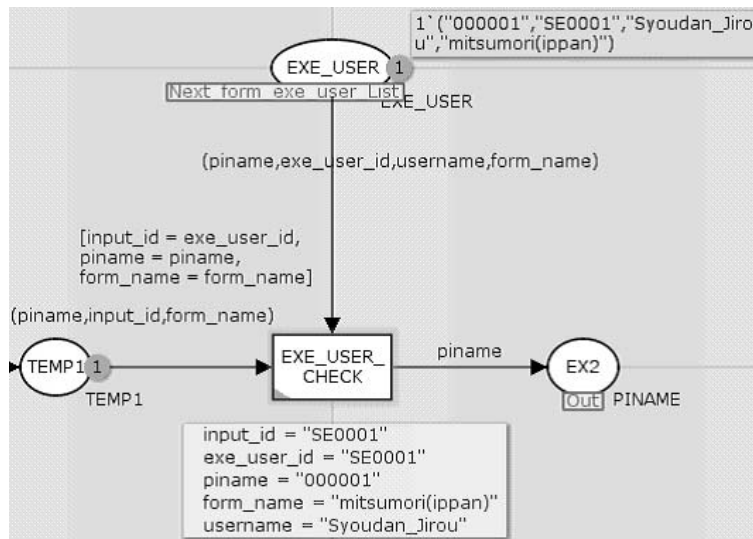


図 4.12: CPN 実行ユーザーチェック例 (実行ユーザーチェック)

#### 4. 帳票処理

帳票処理には2つのパターンがある。処理内容に分岐情報が含まれる場合と含まれない場合である。分岐データとはフロー定義プロセス又は帳票発生条件定義プロセスにおいて遷移先を決定する要素になるデータである。このデータの有無でどのような違いが出るかを説明する。

- CPN 変換帳票処理全体図

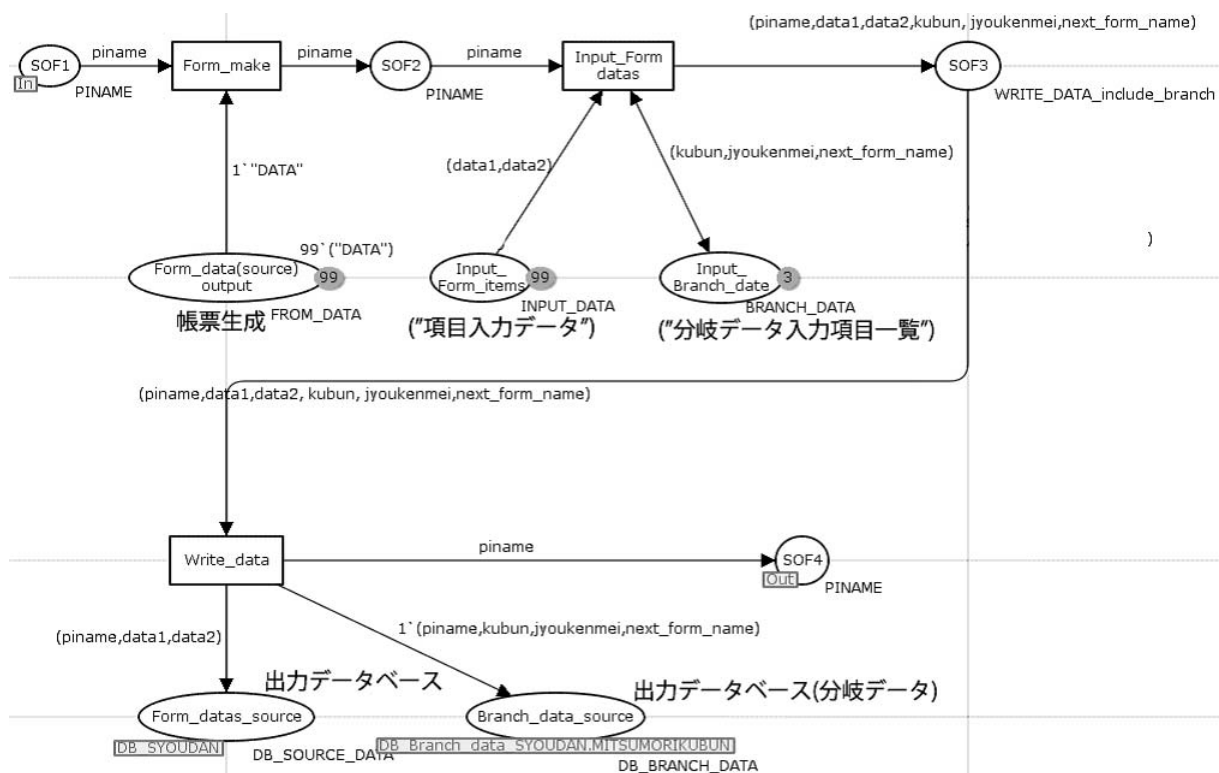


図 4.13: CPN 帳票処理テンプレート 全体図 (分岐データ含む)

分岐情報の有無を問わず、帳票処理時には帳票生成 帳票項目入力 データベース出力の順に遷移する。

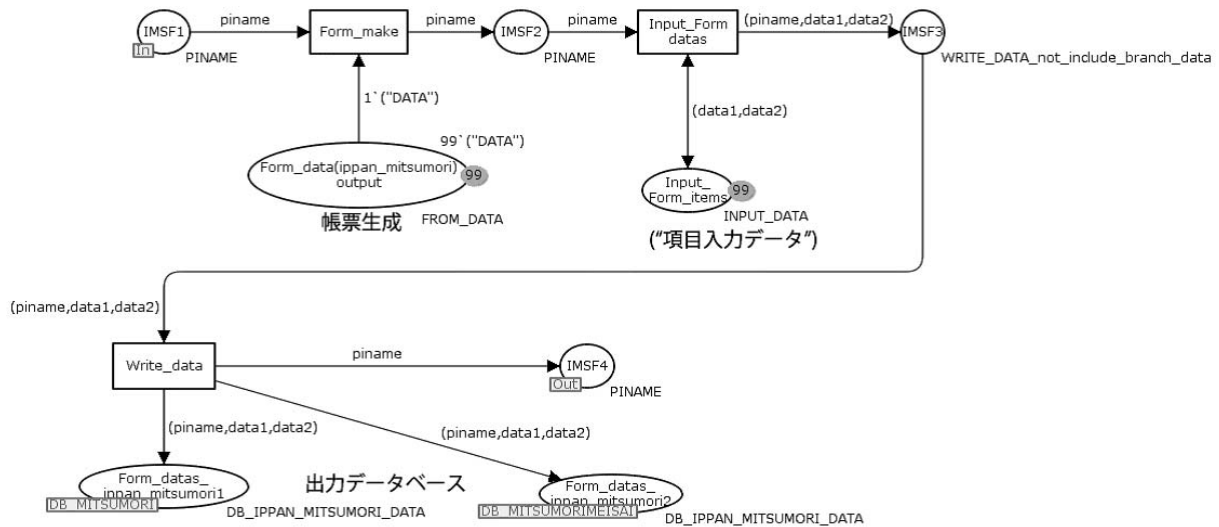


図 4.14: CPN 帳票処理テンプレート 全体図 (分岐データ含まず)

- CPN 帳票生成

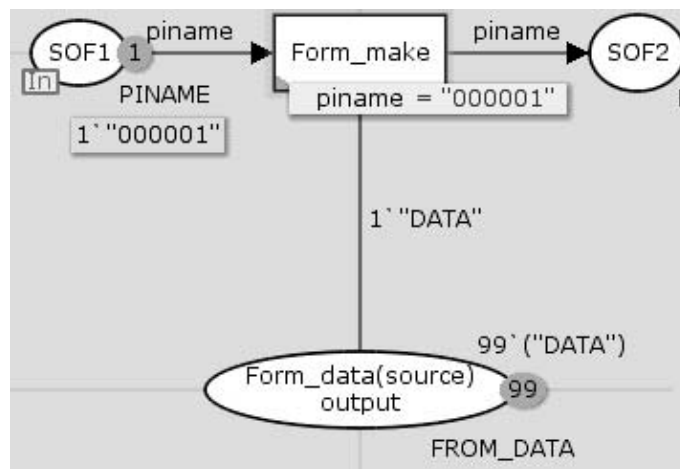


図 4.15: CPN 帳票処理 帳票生成例 (分岐データ含む)

帳票画面生成では帳票画面生成データが読み込まれる. ここでは分岐データの有無による差は無い.

ユーザーインターフェイス作成ツールで作成した帳票データをまとめたものを FROM\_DATA として定義しトークンを持たせてある. 案件番号の遷移により Form\_make トランジションが発火し, これで帳票画面が生成されるものとする.



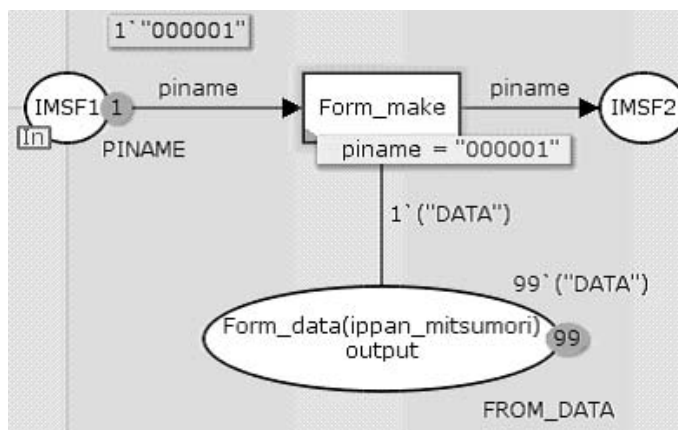


図 4.16: CPN 帳票処理 帳票生成例 (分岐データ含まず)

- CPN 帳票入力部分 (分岐情報含む)

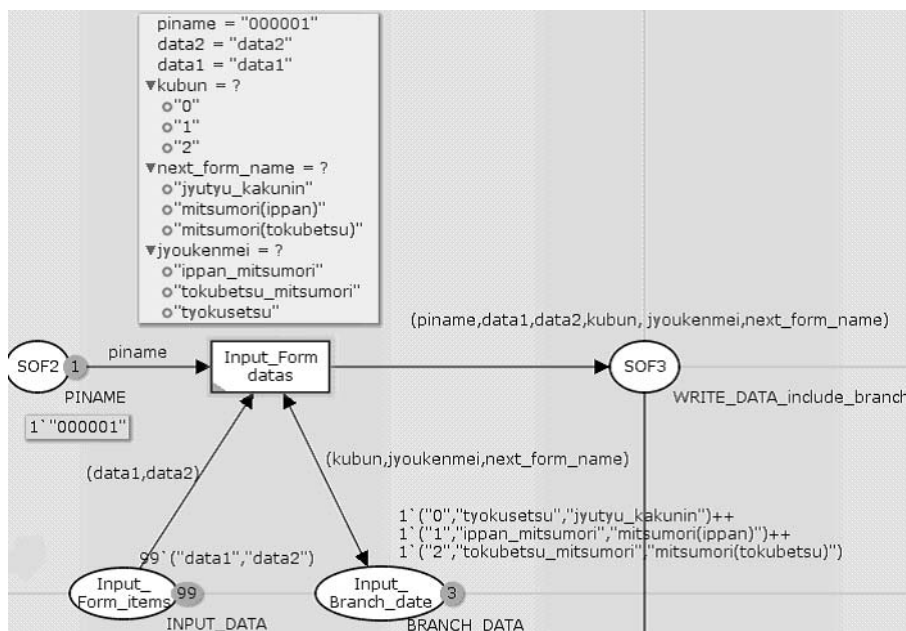


図 4.17: CPN 帳票処理 帳票項目入力例 (分岐データ含む)

帳票内容の中でデータベースを参照して設計を行う。この帳票データの場合は見積区分のデータが分岐データ部分に相当する。モデル化の際にはこのデータ部分を分離して変換する必要があり、分離したデータは Input\_Branch\_Data プレースに見積区分で入力される3つのデータをトークンとして定義し、それ以外の項目データは Input\_Form.items プレース上のトークンとして定義する。帳票項

必要 DB 項目一覧  
 エンティティ:商談  
 テーブル名: SYOUDAN

属性名	カラム名	キー	データ型	入出力属性	備考
商談番号	SYOUDANBANGOU	主キー	文字列	出力	案件キー
担当者コード	TANTOUSYACODE	外来キー	文字列	出力	
顧客コード	KOKYAKUCODE		文字列	出力	
見積区分	MITSUMORIKUBUN		文字列	出力	見積省略='0' or 一般見積='1' or 特別見積='2'
商談名	SYOUDANMEI		文字列	出力	
顧客予算	KOKYAKUYOSAN		文字列	出力	

表 4.3: CPN 帳票項目例 (分岐情報含む)

目の詳細な変換は行ってはならず, 分岐情報の有無で分割して定義してある. 帳票項目の入力 (Input\_Form\_item の data1,data2 がそれに該当する) と分岐データの選択を行い, Input\_Form\_datas トランジションが発火 (入力完了) し入力項目と分岐項目データを持ったトランジションが次のデータ書き込みプロセスに遷移する.

- CPN 帳票入力部分 (分岐情報含まず)

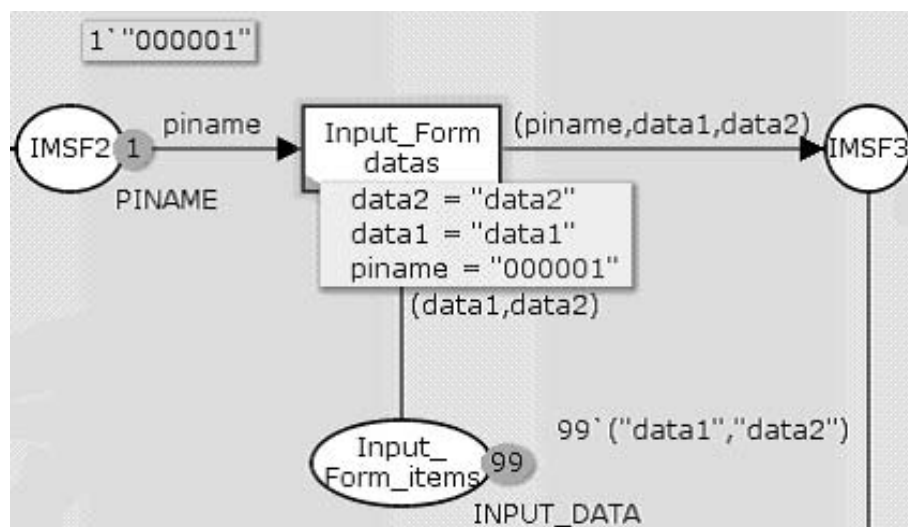


図 4.18: CPN 帳票処理 帳票項目入力例 (分岐データ含まず)

分岐データを含まないデータベースの場合は帳票項目入力部分のみのプレースを定義する. 含む場合と同じように入力が完了すると Input\_Form\_datas トランジションが発火し入力データを含んだトークンが出力処理に遷移する.

必要 DB 項目一覧  
 エンティティ:見積  
 テーブル名: MITSUMORI

属性名	カラム名	キー	データ型	入出力属性	備考
商談番号	SYOUDANBANGOU	主キー	文字列	出力	案件キー
見積完了フラグ	MITSUMORIKANRYOUFLAG		文字列	出力	見積完了='ON'
見積納期	MITSUMORINOUKI		文字列	出力	
見積名称	MITSUMORIMEISYOU		文字列	出力	
見積確度	MITSUMORIKAKUDO		文字列	出力	

エンティティ:見積明細  
 テーブル名: MITSUMORIMEISAI

属性名	カラム名	キー	データ型	入出力属性	備考
見積明細番号	MITSUMORIMEISAIBANGOU	主キー	文字列	出力	案件キー
商談番号	SYOUDANBANGOU		文字列	出力	
商品コード	SYOUHINCODE		文字列	出力	
見積数量	MITSUMORISUURYOU		文字列	出力	
見積金額	MITSUMORIKINGAKU		文字列	出力	

表 4.4: 帳票項目例 (分岐情報含まず)

- CPN 帳票データ出力部分 (分岐情報含む)

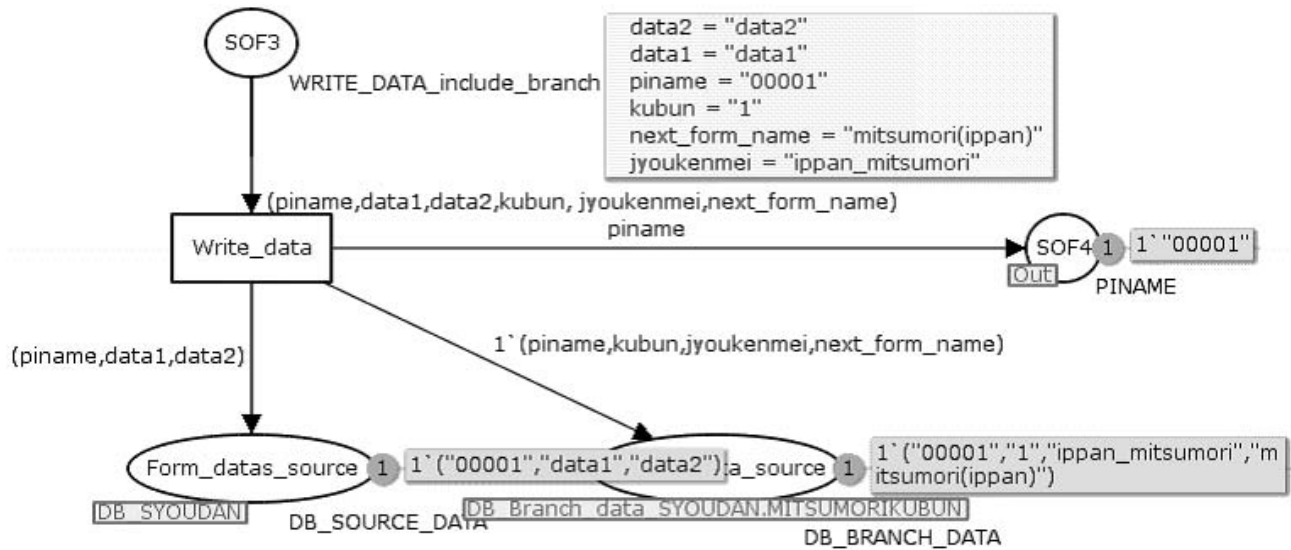


図 4.19: CPN 帳票処理 データ書き込み例 (分岐データ含む)

Write\_data トランジションの発火により入力項目データと分岐項目データを持ったトークンのデータがそれぞれ分離し、分岐データを含まない内容は Form\_datas\_source プレース、分岐データを含んだ内容は Branch\_data\_source プレースへ遷移しデータベースへ出力が行われる。ここで定義するテーブルは帳票定義データと同じものをプレースとして定義し、分岐データの項目のみ分離してある。複数分岐データが有る場合は複数のプレース (テーブル) を定義する。ここで書き込まれ

必要 DB 項目一覧  
 エンティティ:商談  
 テーブル名: SYOUDAN

属性名	カラム名	キー	データ型	入出力属性	備考
商談番号	SYOUDANBANGOU	主キー	文字列	出力	案件キー
担当者コード	TANTOUSYACODE	外来キー	文字列	出力	
顧客コード	KOKYAKUCODE		文字列	出力	
見積区分	MITSUMORIKUBUN		文字列	出力	見積省略='0' or 一般見積='1' or 特別見積='2'
商談名	SYOUDANMEI		文字列	出力	
顧客予算	KOKYAKUYOSAN		文字列	出力	

表 4.5: CPN 帳票項目例 (分岐情報含む)

た分岐データは後に帳票発生条件定義プロセスやフロー定義プロセスにて使用される。

- CPN 帳票データ出力部分 (分岐情報含まず)

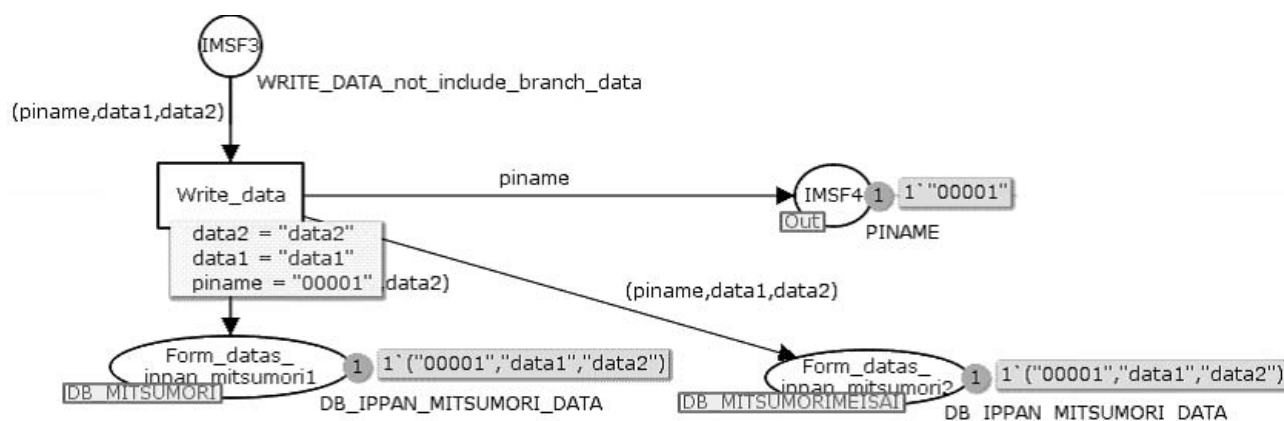


図 4.20: CPN 帳票処理 データ書き込み例 (分岐データ含まず)

Write\_data トランジションの発火により入力データの保存が行われる。分岐データは含まれないので分離せずに出力が行われる。この場合は複数のテーブルが存在するのでその分のプレースを定義してある。

必要 DB 項目一覧  
 エンティティ:見積  
 テーブル名: MITSUMORI

属性名	カラム名	キー	データ型	入出力属性	備考
商談番号	SYOUDANBANGOU	主キー	文字列	出力	案件キー
見積完了フラグ	MITSUMORIKANRYOUFLAG		文字列	出力	見積完了='ON'
見積納期	MITSUMORINOUKI		文字列	出力	
見積名称	MITSUMORIMEISYOU		文字列	出力	
見積確度	MITSUMORIKAKUDO		文字列	出力	

エンティティ:見積明細  
 テーブル名: MITSUMORIMEISAI

属性名	カラム名	キー	データ型	入出力属性	備考
見積明細番号	MITSUMORIMEISAIBANGOU	主キー	文字列	出力	案件キー
商談番号	SYOUDANBANGOU		文字列	出力	
商品コード	SYOUHINCODE		文字列	出力	
見積数量	MITSUMORISUURYOU		文字列	出力	
見積金額	MITSUMORIKINGAKU		文字列	出力	

表 4.6: 帳票項目例 (分岐情報含まず)

## 5. 宛先選択

宛先選択の CPN 変換には宛先人物検索条件に則った変換を行う。

部門	役職	次処理帳票	備考
営業部	課長	上位役職帳票確認 (一般見積)	

表 4.7: 宛先人物検索条件例

SQL プレースに宛先人物検索条件を定義したトークンを定義する。SelectDestinatio(ippan) トランジションにてユーザー情報一覧 (DB\_USER\_LIST プレース上のトークン) から条件に適合する人物を検索し選択する。選択された人物は案件番号, 人物名, 宛先の帳票名のトークンとして EXE\_USER\_LIST プレースに保存され, 次の業務プロセスの帳票選択処理にて利用される。

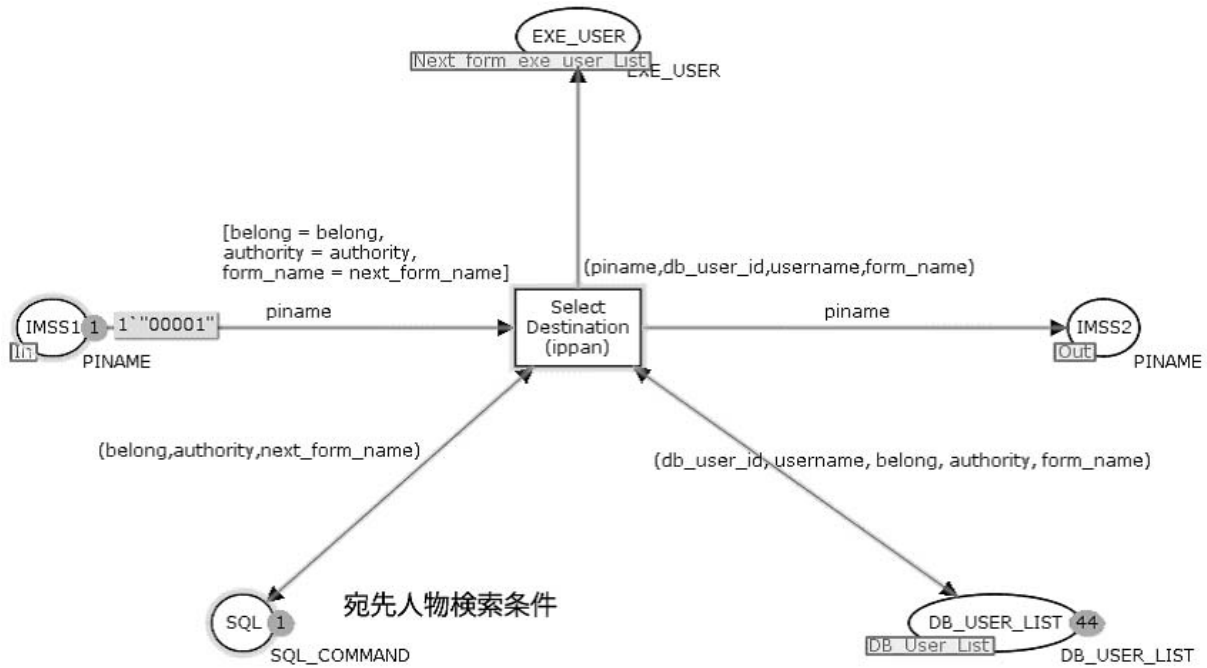


図 4.21: CPN 宛先選択テンプレート

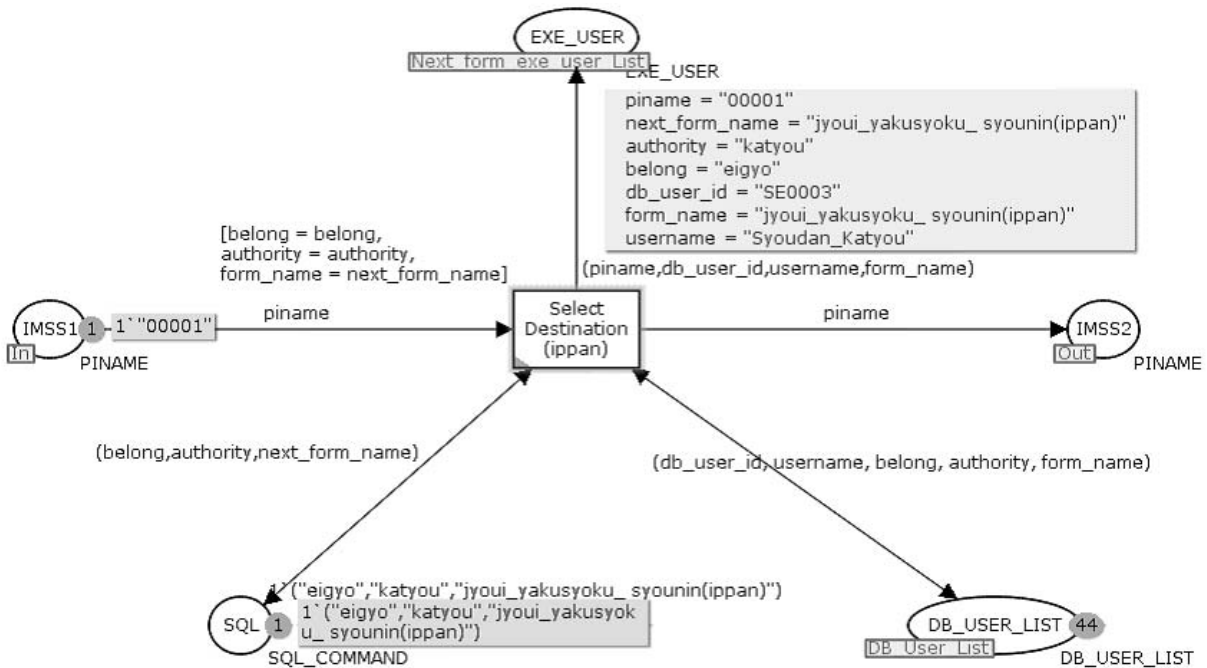


図 4.22: CPN 宛先選択例

## 6. ログアウト

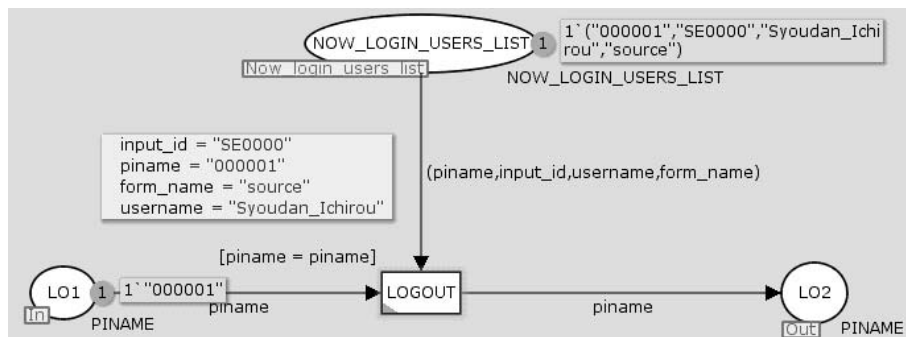


図 4.23: CPN ログアウト

ログイン時に一時保存したログイン情報を除去してログアウト処理は終了する。このプロセスは同一の物が全ての帳票処理に対して適応される。

### 4.2.3 帳票発生条件定義プロセスの変換

帳票発生条件定義プロセスでは帳票定義プロセスで定義された帳票をどのような条件で発生させるかという定義を行うプロセスである。分岐条件によって幾つかのパターンがありそれぞれのパターンについて説明する。

- 発生条件無し

@source

必要 DB 項目一覧

分岐条件	発生帳票名 (又は遷移先)	備考	参照番号
無し	@ source	無条件発生, 帳票データ一覧へ	1

表 4.8: 発生条件無し

発生条件が無い場合は登録した帳票に強制的に遷移し、帳票処理が開始される。CPNでは階層化定義したトランジションに遷移することで帳票発生を表現する。

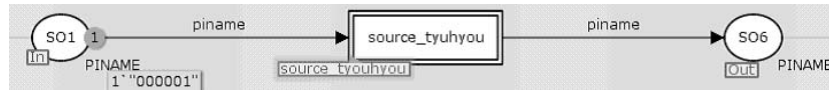


図 4.24: CPN 帳票選択 (条件無し)

- 発生条件有り (参照番号単数)

見積中

必要 DB 項目一覧

エンティティ:商談

テーブル名: SYOUDAN

分岐条件	発生帳票名 (又は遷移先)	備考	参照番号
MITSUMORIKUBUN='1'	見積中: 一般見積作成	一般見積='1', 帳票データ一覧へ	1
MITSUMORIKUBUN='2'	見積中: 特別見積作成	特別見積='2', 帳票データ一覧へ	1

表 4.9: 発生条件有り

条件により発生帳票が変わる場合がある。ワークフローシステムでは DB より分岐条件のデータを参照することによって遷移先が決定される。各帳票処理プロセスを CPN に変換する際、分岐条件をプレースにトークンとして保存したが、そのトークンの分岐情報をトランジションで参照しアロー上でチェックを行うことによって遷移先を決定する。

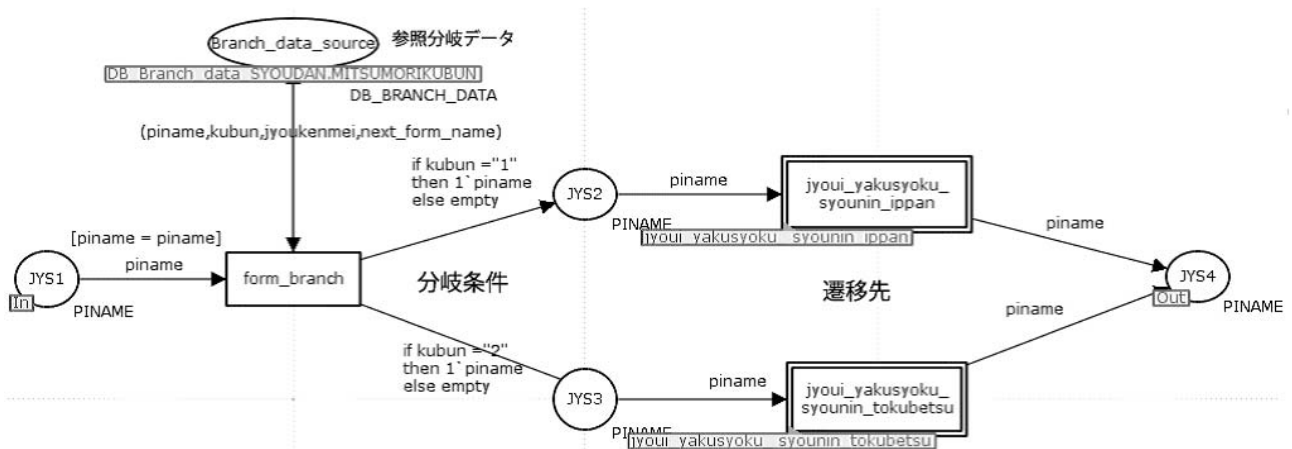


図 4.25: CPN 帳票選択

- 発生条件有り (参照番号複数)

発生条件が複数定義されている場合がある。この場合は参照番号の順に変換を行う。この場合だと参照番号 1 の時は発生条件は定義されていないので初めの帳票は無数



受注処理待ち  
必要 DB 項目一覧  
エンティティ:受注  
テーブル名: JUTYU

分岐条件	発生帳票名 (又は遷移先)	備考	参照番号
無し	受注確認	無条件発生, 帳票データ一覧へ	1
分岐条件	発生帳票名 (又は遷移先)	備考	参照番号
JUTYUKANRYOUBI IS NOT NULL	受注登録	受注完了日=DATE, 帳票データ一覧へ	2
JUTYUKANRYOUBI IS NULL	上位役職認証 (受注後)	受注完了日=NULL, フロー定義データへ (業務ステッパ一覧)	2

表 4.10: 発生条件有り (連結)

件で発生する. つまり発生条件無しの場合と同じ変換を行う. 発生条件 2 は分岐情報の参照により決定されるので分岐情報を持ったプレースを参照することで遷移先が決定する. これは発生条件が単数の場合と同じ変換である. 複数の発生条件が定義されている場合は参照番号毎に連結する形に変換される.

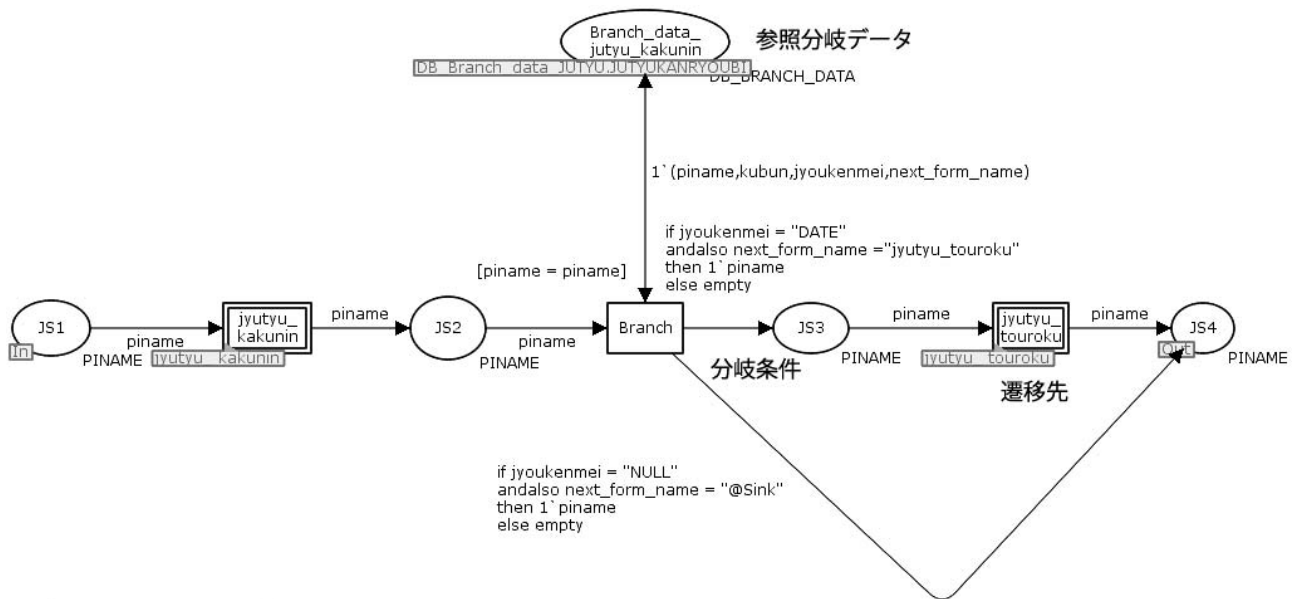


図 4.26: CPN 帳票選択 (複数参照番号)

● 発生条件有り (複数分岐条件参照)

上記の場合は参照する分岐データが 1 つの場合であり, 2 つ以上分岐データの参照が必要な場合は条件が AND, OR で定義される場合がある. この場合は以下の形に変換を行う.

分岐条件	遷移先
条件 1 OR 条件 2	exit

表 4.11: OR 分岐 (参照分岐条件)

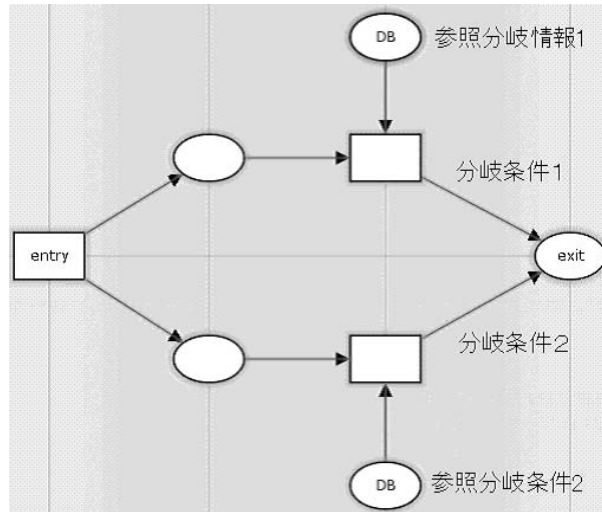


図 4.27: OR 分岐条件例 (参照分岐データ2つ)

分岐条件	遷移先
条件1 AND 条件2	exit

表 4.12: AND 分岐例 (参照分岐データ2つ)

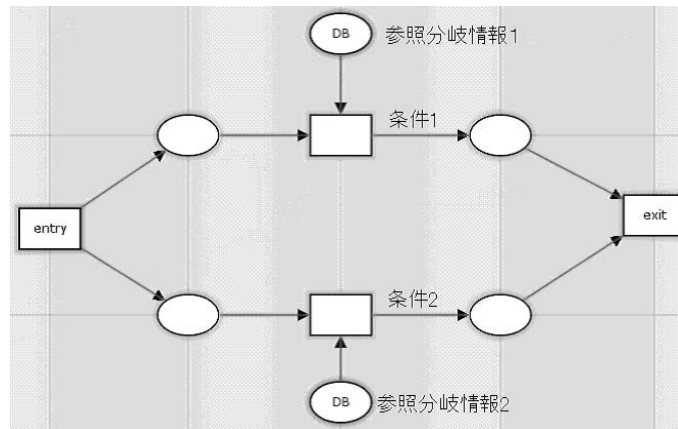


図 4.28: AND 分岐例 (参照データ2つ)

OR の場合は一方でも条件を満たせば,AND の場合は全ての条件を満たす場合に遷移可能になる. 参照分岐データが増えると初期のトランジションからの分岐数がそれだけ増加し, 経路が増えていく.

#### 4.2.4 フロー定義プロセスの変換

ビジネスプロセス定義では各ノードがどのノードに遷移するかを定義した. ここではその遷移先を定義された各ノードに対して変換規則をどの様に適応するかを定義する.

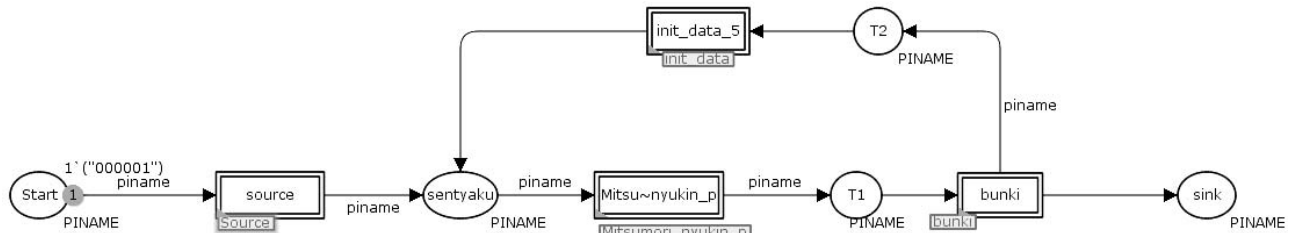


図 4.29: CPN フロー定義全体図 1

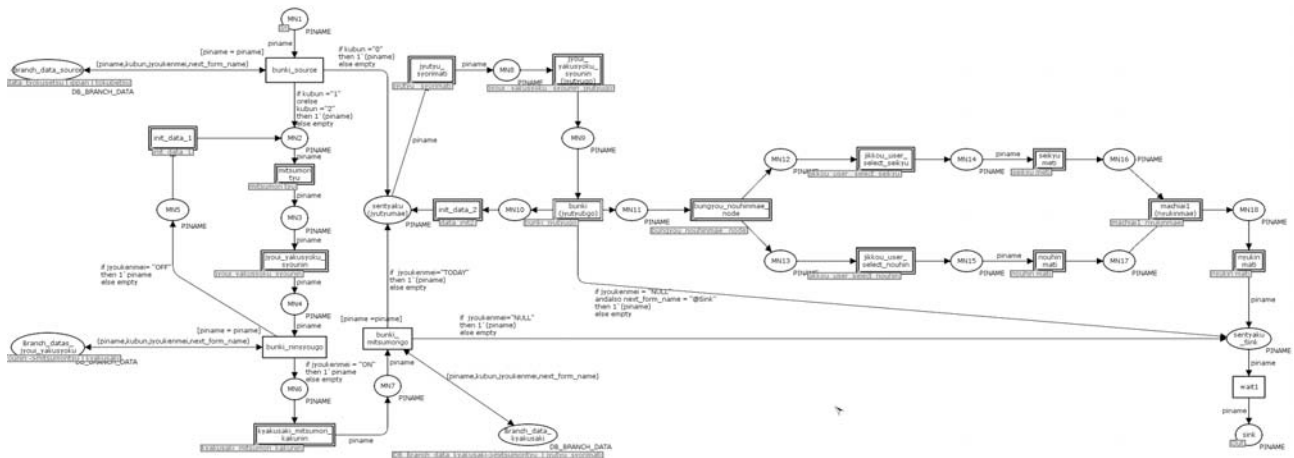


図 4.30: CPN フロー定義全体図 2

#### 1. 業務ステップ

業務ステップには帳票発生条件定義プロセスへの遷移先と次のフロー定義プロセスへの遷移先の2つが定義されている.

一つ目の帳票発生条件定義プロセスへの定義は, 業務ステップ下の帳票発生条件プロセスを表現するために, 階層化したトランジションとして表現する. もう一つのフ

遷移先及び参照番号	備考	参照番号
Source,1	帳票発生条件データ一覧へ	1
先着,1	分岐プロセス一覧へ	2

表 4.13: 業務ステップ source 定義内容

ロー定義プロセスへの遷移先はこの業務ステップから次への遷移先名称が定義されている。この場合は遷移先として先着プロセスが定義されている。

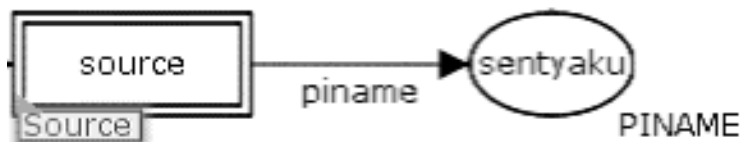


図 4.31: CPN 業務フロー

## 2. 業務ステップ (階層化定義)

階層化定義された業務ステップには下階層への遷移と次の遷移先情報が定義される。

遷移先及び参照番号	備考	参照番号
@source(見積～入金プロセス内),1	ソースノード一覧へ, 下位階層へ	1
分岐,1	シンクノードより遷移, 階層化定義終了	2

表 4.14: CPN フロー定義階層化定義内容

階層化定義は階層化したトランジションによってそれを表現する。階層化定義した場合ソースノードへ遷移する必要があるため、階層下のソースノードへの遷移を定義する。

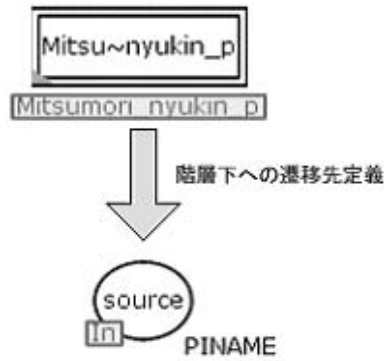


図 4.32: CPN 業務フロー階層化定義 1

次の遷移先は業務ステップと同じようにアローによって示される。

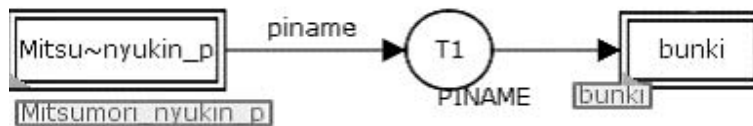


図 4.33: CPN 業務フロー階層化定義 2

### 3. 制御ノード

制御ノードには4種類のノードが有るため各ノードについて説明する。

- 分岐ノード

分岐ノードは階層化(最終的に階層化定義より出る数が3つ以上の場合は必須)したトランジション内にて定義を行う。変換については帳票発生条件定義プロセスと全く同じである。与えられた条件に従ってCPNへの変換を行う。

エンティティ:受注  
テーブル名: JUTYU

分岐条件	遷移先及び参照番号	備考	参照番号
MITSUMORIKANRYOUBI IS NULL	先着,1	見積見直し経路,分岐プロセス一覧へ	1
MITSUMORIKANRYOUBI IS NOT NULL	分岐,2	分岐プロセス一覧へ	1

エンティティ:受注  
テーブル名: JUTYU

分岐条件	遷移先	備考	参照番号
JUTYUKANRYOUBI IS NULL OR SEIKYUSYOHAKKOUFLAG ='ON'	@sink,1	受注キャンセル,シンクノード一覧へ	2

表 4.15: CPN 制御ノード(分岐)定義内容

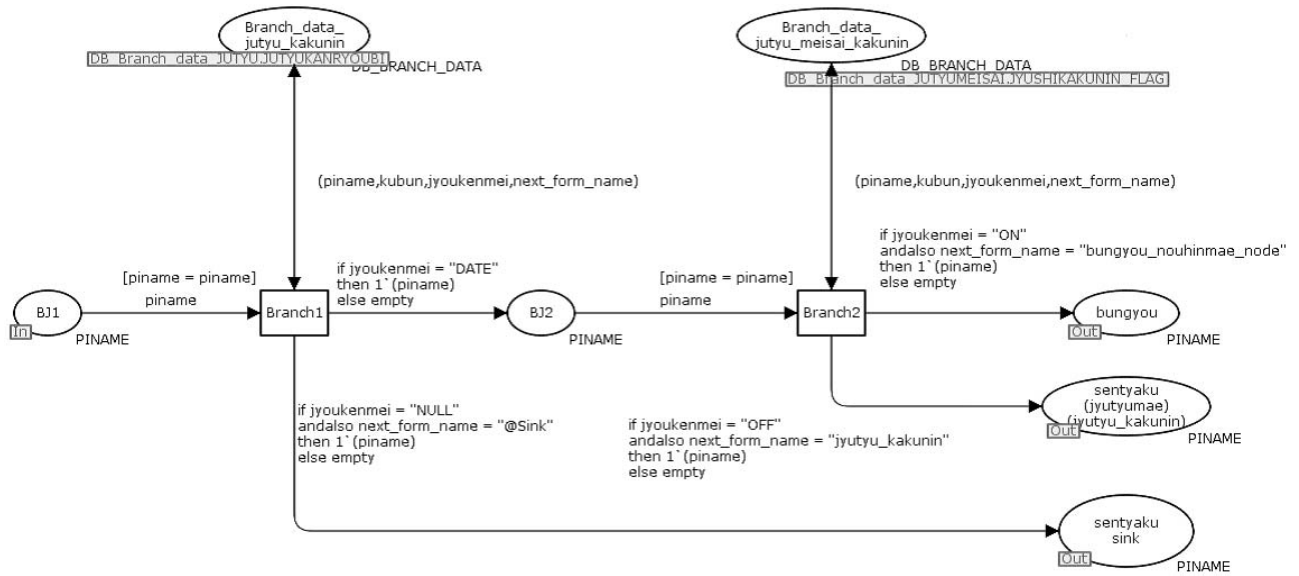


図 4.34: CPN 分岐ノード例

● 先着ノード

分岐条件	遷移先及び参照番号	備考	参照番号
無し	見積～入金プロセス,1	強制遷移, 業務ステップ (階層化定義) 一覧へ	1

表 4.16: CPN 制御ノード (先着) 定義内容

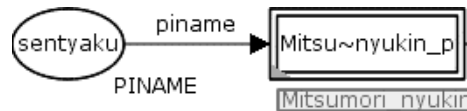


図 4.35: CPN 先着ノード

先着ノードはプレースで表現する. 先着ノードは他のプロセスからの待ち受けを行う. 分岐条件の定義は行われず, 次のプロセスへの遷移先のみ定義される.

- 分業ノード

分岐条件	遷移先	備考	参照番号
無し	実行ユーザー選択 (請求待ち)	同時強制遷移, 分業ノード, 業務ステップ一覧へ	1
無し	実行ユーザー選択 (納品待ち)	同時強制遷移, 分業ノード, 業務ステップ一覧へ	1

表 4.17: CPN 制御ノード (分業) 定義内容

分業ノードは階層化したトランジションによって表現する。分業ノードは処理するルートが複数に変わるので複数に遷移した場合、分岐した経路において業務ステップが複数存在する可能性がある。これに対応するために宛先選択時に EXE\_USER\_LIST プレースに保存されたトークン (実行者) を増加させる必要がある。帳票宛先選択時に増やす方法と、分業ノード時に増やす方法が考えられるが、今回は分業ノードで増加させる方法を採用。これは宛先選択が一つのみになる Cosminexus ワークフローシステムの仕様に則ったものである。

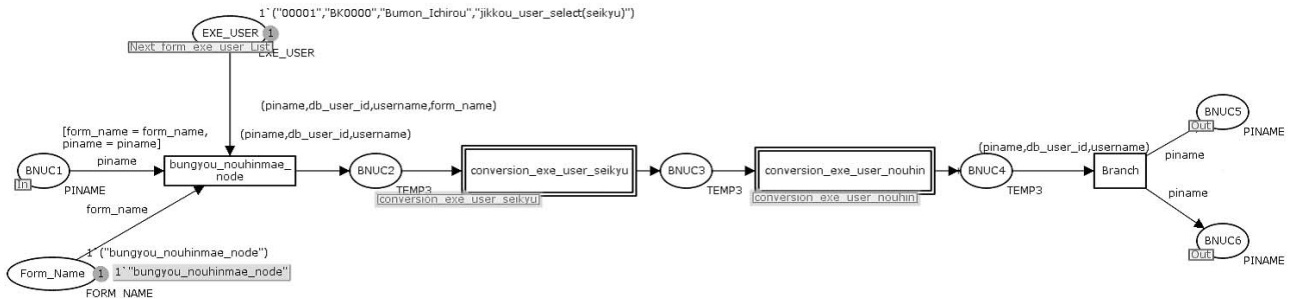


図 4.36: CPN 分業ノード内処理例

図 4.35 は分業ノード内の処理を示した全体図の例である。ここでは分岐後の帳票名称に合わせた変換作業を行っている。これは分岐後の帳票の数に応じて行われる。

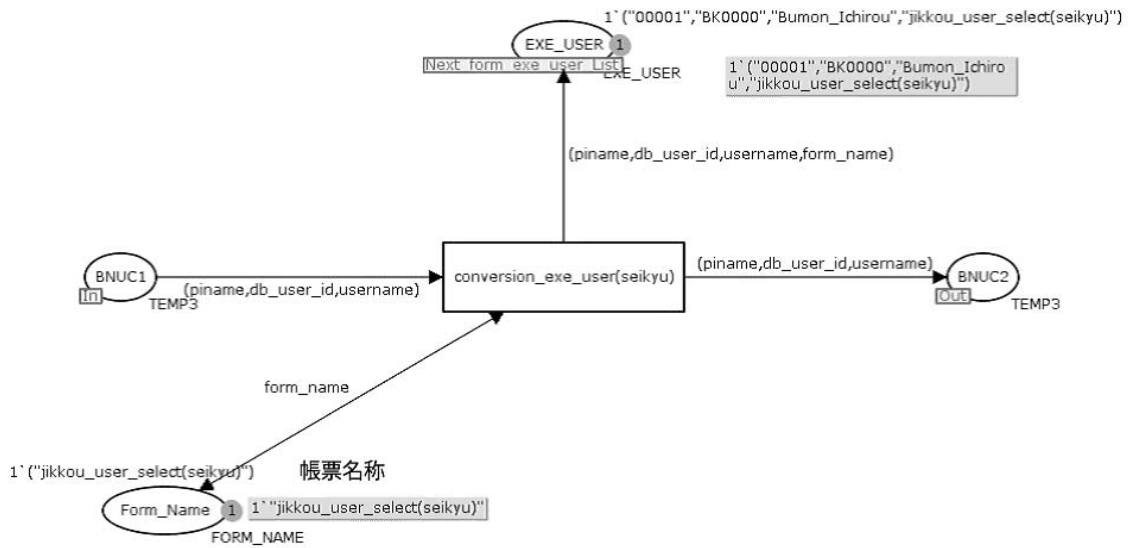


図 4.37: CPN 分業ノード トークン変換作業例

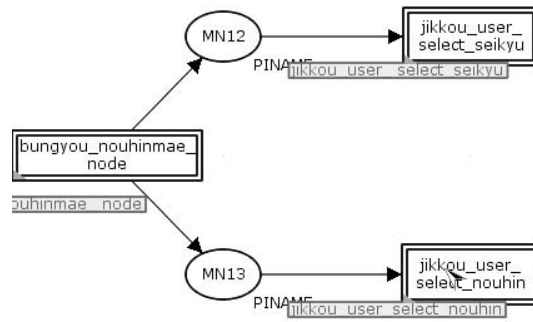


図 4.38: CPN 分業ノード遷移先定義例

変換後は遷移先の各定義に従って分岐先を決定する。



- 待合ノード  
待合いノードは分業ノードとセットで使用され、分業ノードで分かれた複数の経路から待合いノードに全て遷移した場合に次のプロセスに遷移する。

分岐条件	遷移先及び参照番号	備考	参照番号
無し	入金待ち,1	待合ノード,業務ステップ一覧へ,直前のプロセス待ち	1

表 4.18: 待合

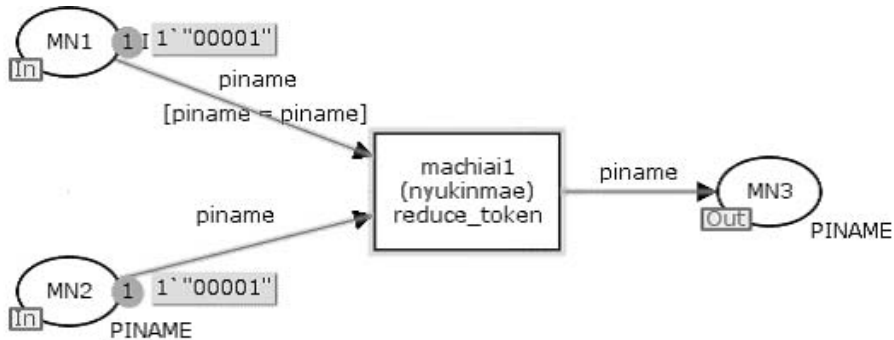


図 4.39: CPN 待合いノード

待合ノードを再現するためトランジションで待ち受けを行う。

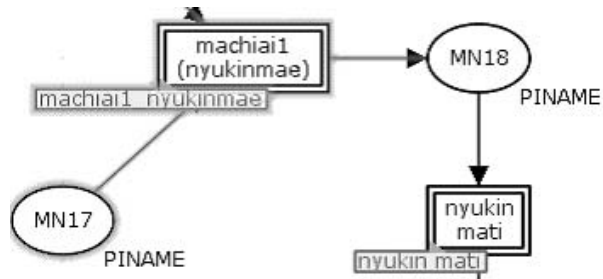


図 4.40: CPN 待合いノード遷移先定義

定義に基づき次の遷移先が決定される。

#### 4. ソースノード

@source(見積～入金プロセス内)

遷移先及び参照番号	備考	参照番号
分岐 ( Source 後 ),1	分岐プロセス一覧へ	1

表 4.19: @source(見積～入金プロセス内)

ソースノードはプレースで表現される. ここで定義されている内容は次のプロセスへの遷移のみであり, 図のように表現する.

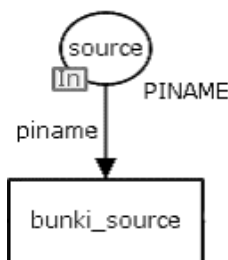


図 4.41: CPN フロー定義ソースノード

#### 5. シンクノード

sink(見積～入金プロセス内)

遷移先及び参照番号	備考	参照番号
見積～入金プロセス,2	階層化定義一覧へ, 上位階層へ	1

@sink

遷移先及び参照番号	備考	参照番号
無し	ビジネスプロセス終了	1

表 4.20: シンクノード定義内容

シンクノードはプレースで表現される. シンクノードには定義される2つのパターンがある.1つは階層化定義より戻る場合であり, もう一つはビジネスプロセスの終了を表す場合である. 階層化定義より戻る場合のシンクノードは階層化定義した業務ステップに遷移するが終了を表すシンクノードはそこでプロセス終了になる.



図 4.42: CPN フロー定義 階層化終了シンクノード



図 4.43: CPN フロー定義 終了シンクノード

#### 4.2.5 変換における追加処理

ペトリネット理論の法則上, 変換したモデルに対して追加処理をしなければならない場合がある. ここではその追加処理の発生条件と内容について説明する.

##### 遷移先が戻るケース

フロー定義プロセス, 帳票発生条件プロセス両方について起こる問題

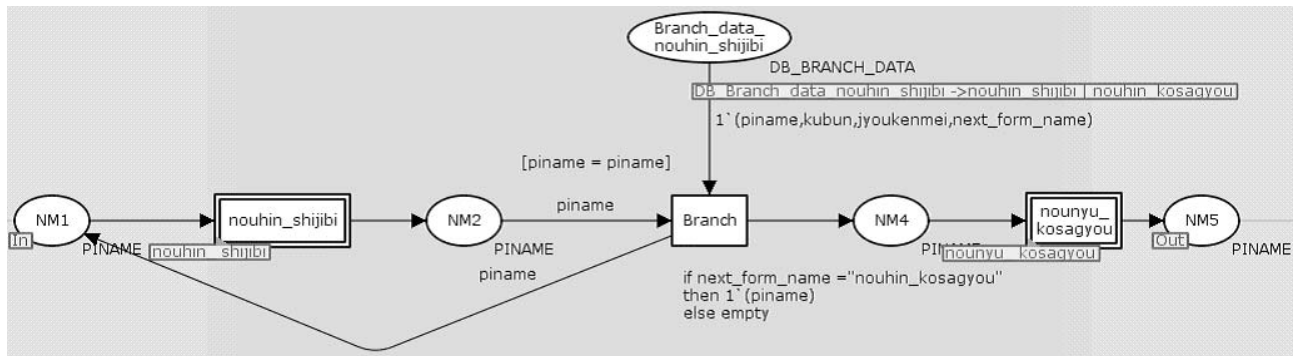


図 4.44: 遷移先が戻る場合 (修正前)

分岐によって遷移先がもう一度通過した経路に戻る場合は, その通過した経路内にある帳票のトークン (DB に書き込んだ帳票項目と分岐情報) を取り除く処理を追加しなけれ

ばならない。

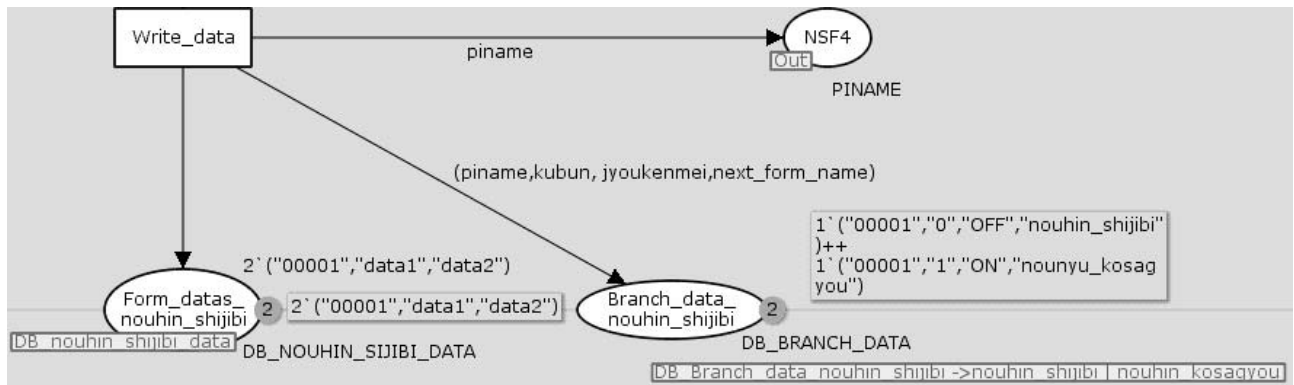


図 4.45: トークンの重複発生例

これは再び同一の経路を通った時に既に帳票項目が書き込まれているため、再び書き込もうとすると帳票項目の重複が発生するためである。

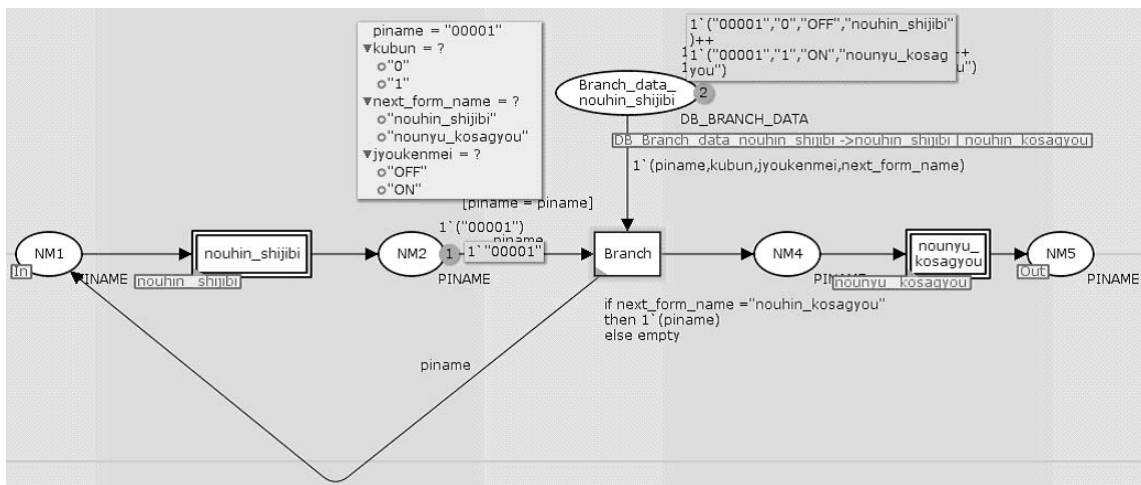


図 4.46: トークンの重複による遷移先不明問題

これにより分岐情報参照の際に遷移先が一意に決定されなくなり、ビジネスプロセスとの不整合が発生する。

これを解消するため DB に書き込んだ帳票項目と分岐情報のトークンを取り除く処理を行う。分岐したトランジションから戻った間に存在する帳票すべてに対して行われる。

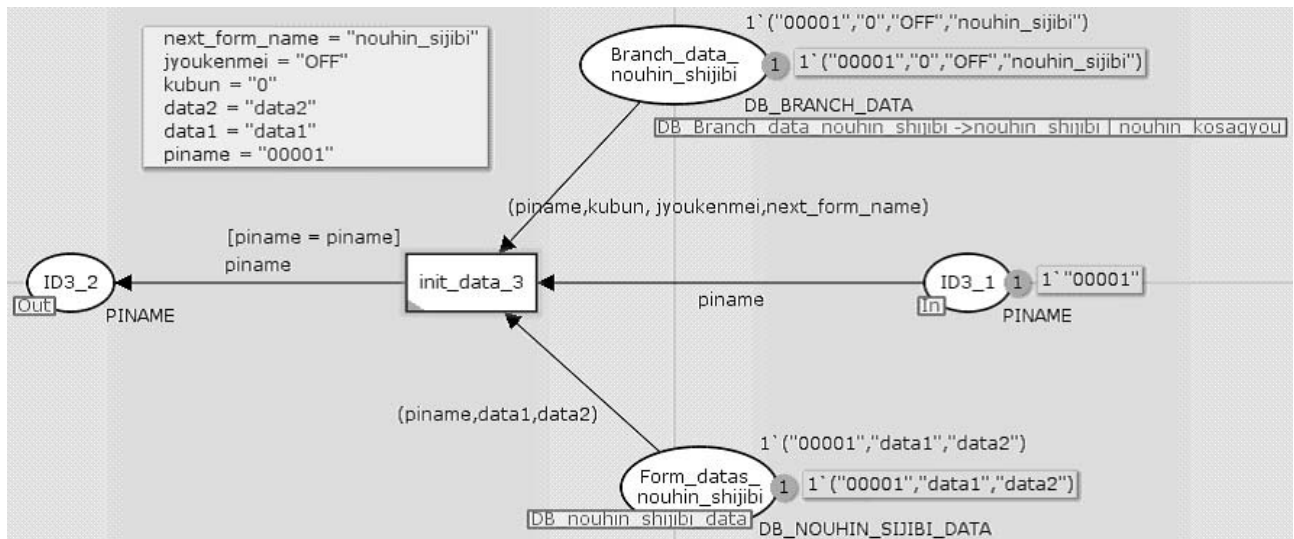


図 4.47: 初期化処理

#### 並行動作時の実行人物チェックの問題

分業ノードにより複数の帳票に遷移した場合、実行ユーザーチェックにて現在ログイン中の人物リストに同一案件番号の人物が複数現れる場合がある。これは同じ時間軸でログインをした場合に発生する現象である。これにより宛先に送られた人物以外の人物が帳票を実行する原因になる。これを解消するために分業ノードから待合いノードまでの間にある帳票処理の実行ユーザーチェック全てに同時に複数のログインが発生した場合でも一意に決定するための処理を入れる必要がある。

実行者リストに保存されるトークンは（案件番号、ユーザー ID、ユーザー名、帳票名称）の情報を持っている。このうち帳票名称は処理中の帳票の名称と同じにしなければならないため、この部分を利用し複数のログイン人物一覧から一意に決定するための処理を入れる。現在処理中の帳票名称のトークンを持っているプレースから名称の参照を行い、同一の名称をもつトークンを取り出し遷移させることで解決可能になる。ログアウト時にも同様の処理を行う。

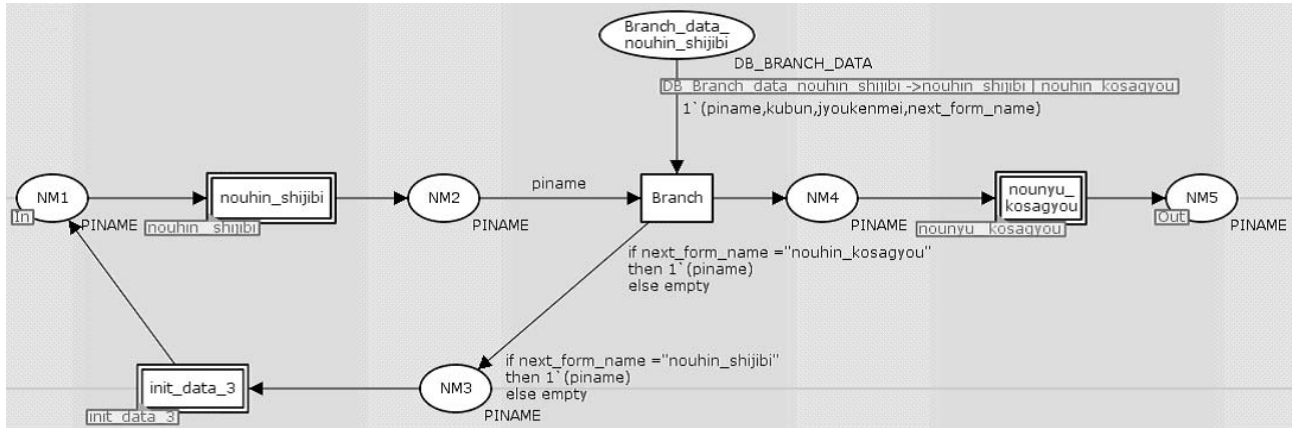


図 4.48: 遷移先が戻る場合 (修正後)

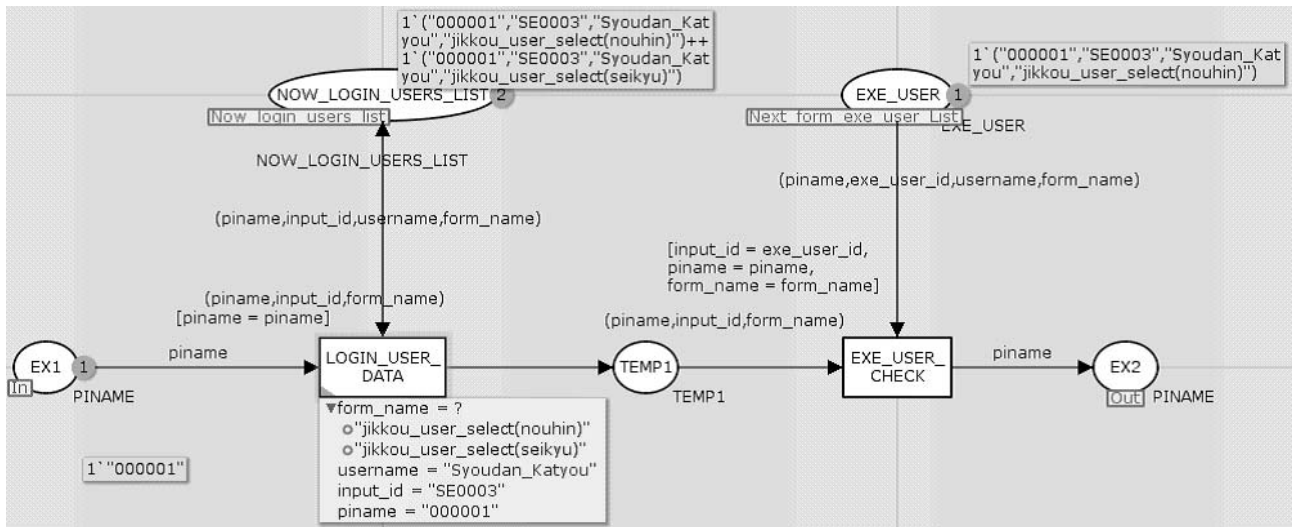


図 4.49: 平行動作時の業務プロセスにおける実行人物不定問題

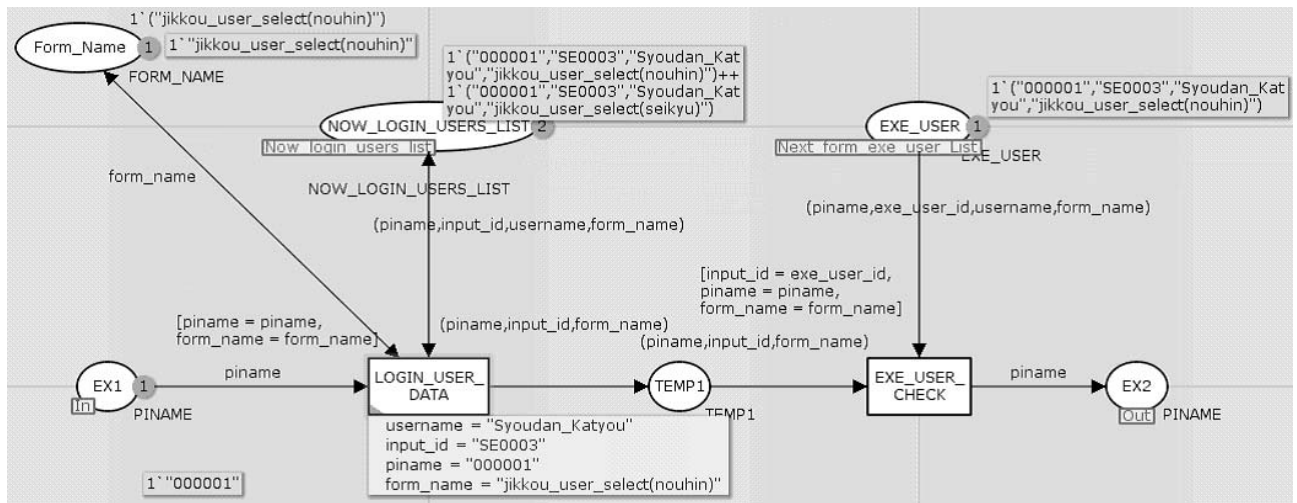


図 4.50: 平行動作時の業務プロセスにおける実行人物不定問題解決例 (実行者チェック)

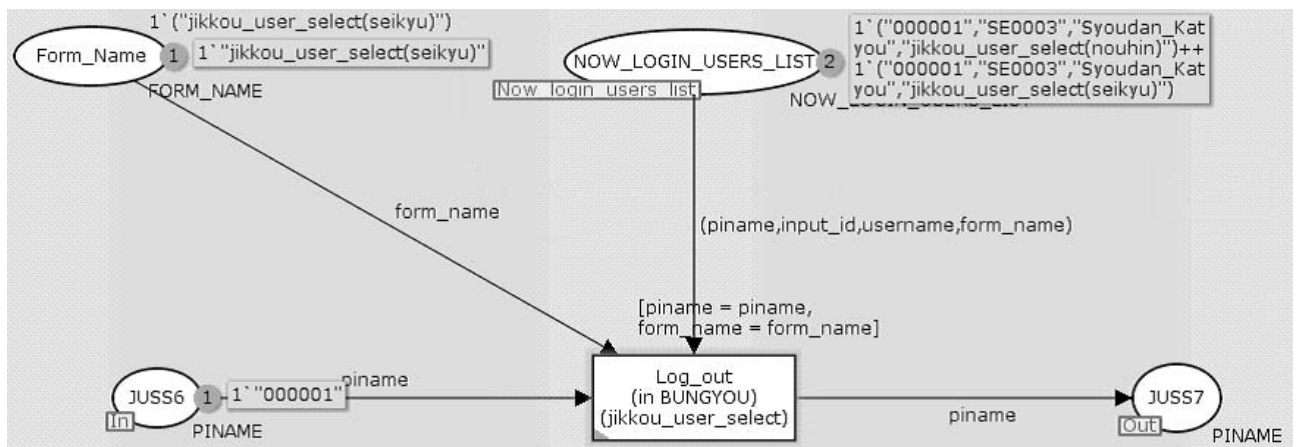


図 4.51: 平行動作時の業務プロセスにおける実行人物不定問題解決例 (ログアウト)

# 第5章 モデルの検証について

## 5.1 シミュレーションによる検証

シミュレーションで変換したモデルを実行した所, 次の問題が発見された.

- 宛先人物と遷移先の不整合問題

通常, 宛先選択時に選んだ人物が次の遷移先の帳票処理における実行者となる. これは帳票に分岐条件が無い場合は問題は発生しない. 遷移先が一つに決定されているためである. しかし帳票項目に分岐条件がある場合は遷移場所がその分岐条件の選択によって変わってしまう. これは宛先人物選択時に問題を発生させる可能性がある. 分岐条件と宛先人物検索条件との関連付けを取ってないためである.

分岐条件によって宛先人物検索条件を拘束してやればこの問題は発生しない.

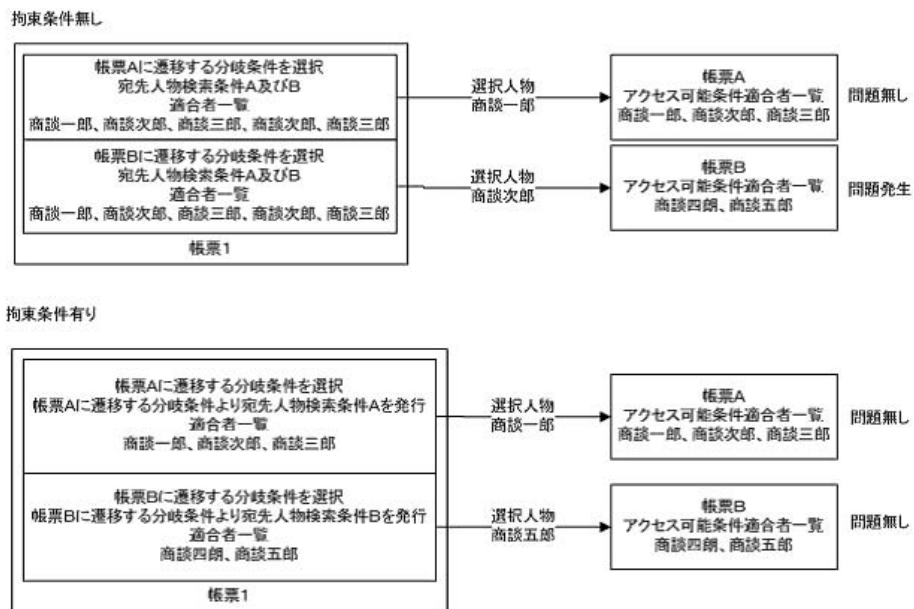


図 5.1: 不整合問題発生例



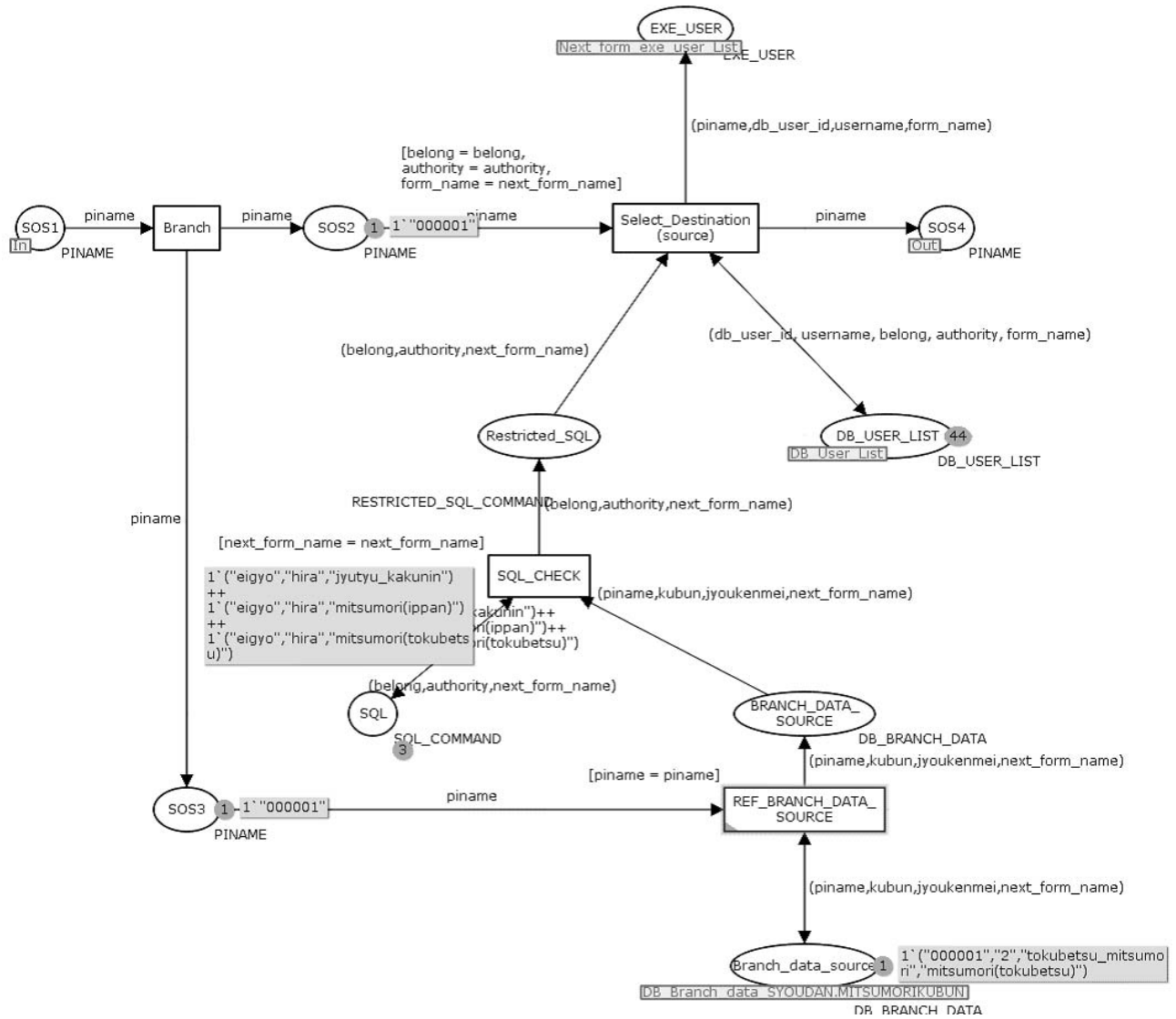


図 5.2: CPN 拘束条件による宛先人物検索

そこで宛先選択処理を行うときに拘束条件を加えたモデル変換を行う。分岐データと関連付けを取るために書き込んだ分岐データを参照し、宛先人物検索条件と照らし合わせることにより、条件を満たした宛先人物検索条件が選り出される。

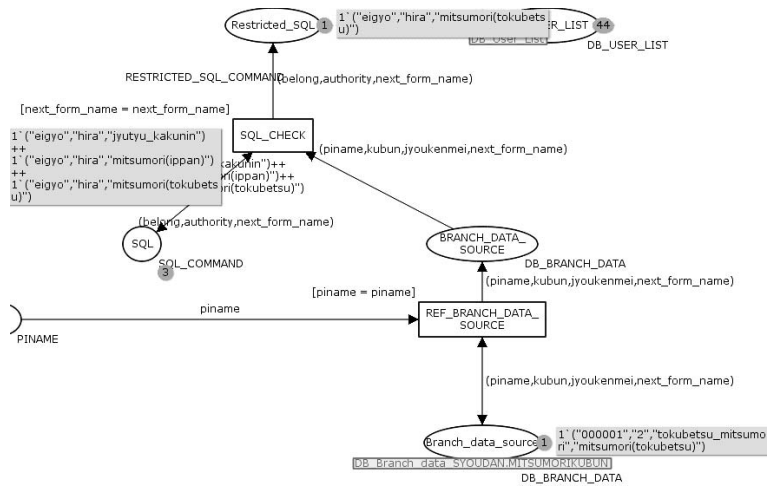


図 5.3: CPN 拘束条件による宛先人物検索

分岐先と宛先人物検索条件が合致した間違いの無い宛先人物が選ばれる。

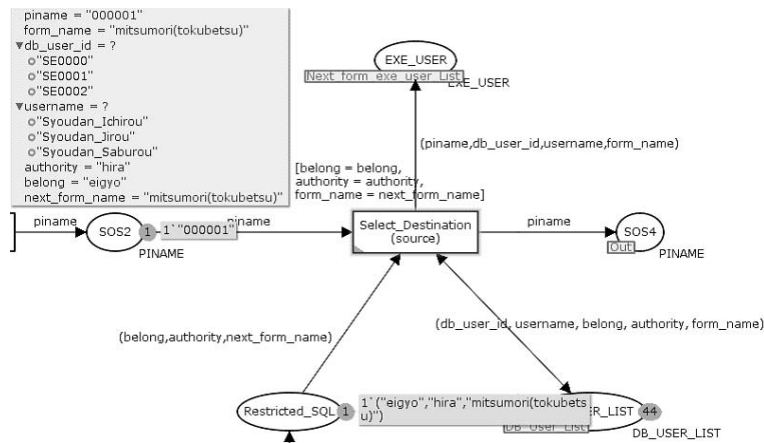


図 5.4: CPN 拘束条件による宛先人物検索

## 5.2 状態空間生成ツールによる検証

状態空間生成ツールでは, 以下のことが調べられる.

- 特定の状態への到達可能性判定.
- 特定のトランジションの発火可能性判定.
- ライブネス検査: 発火できないトランジションのリストを出力する.
- ホーム状態の検査: 任意の状態から到達可能な状態をすべて求める.
- 有界性: 各ブレースに入るトークン数の上限, 下限
- 公平性: 発火可能なトランジションはいつかは必ず発火する等.

さらに, 分岐時間時相論理式により記述された性質の検証を行うモデル検査ツール (ASK-CTL ツール) を備えている. たとえば, ワークフローではソースから開始したフローは必ず何らかの形で終了してシンクノードに到達し, そのときに中間に未処理のジョブを残さないことが要求されるが, このような性質も検証可能である.

状態空間生成では, シミュレーションと異なり, 状態からの分岐先が複数ある場合でもすべての場合を網羅的に調べることができる. ただし, 状態空間を有向グラフとしてメモリ上に展開するため, 検証可能なモデルのサイズに制限がある. 作成したワークフローのモデルでは, ツール上の状態数の上限である 20 万状態を超えてしまった. この制限については, ツールの今後のバージョンアップにより対処されるとのことである. 今回のモデル化ではユーザーのログイン処理など, できるだけ実際のシステム上のふるまいを忠実に再現することを目的としたため, 冗長な記述がかなり含まれている. これらの記述を簡略化するだけで, 状態数はかなり削減されることが考えられる.

ツールでは指定した条件 (時間, 状態数, 状態に関する条件式) により状態空間生成を停止させることができる. それに至るまでは幅優先探索により状態空間を生成するので, 有界モデル検査 (bounded model checking) を行うことになる. 今回, 状態空間生成ツールはモデル作成において, 発火不能なトランジションの検出に用いた.

以下に, 作成段階のモデルに対する CPN TOOLS が出力した検証レポートの一部を示す. これは, 7,200 秒の実行により 115,575 状態まで探索した結果のレポートである.

CPN Tools state space report for:

Report generated: Thu Jan 31 11:15:17 2008

Statistics

---

State Space

Nodes: 115575

Arcs: 182103  
Secs: 7200  
Status: Partial

Scc Graph  
Nodes: 115575  
Arcs: 182103  
Secs: 27

### Boundedness Properties

---

Best Integer Bounds  
Upper Lower  
Branch\_\_next\_user\_conversion'BNUC2 1  
1 0  
Branch\_\_next\_user\_conversion'BNUC3 1  
1 0  
...

### Liveness Properties

---

Dead Markings  
12522 [99974,99972,99884,99883,99880,...]

Dead Transition Instances  
Exe\_User\_Check'EXE\_USER\_CHECK 10  
Exe\_User\_Check'EXE\_USER\_CHECK 12  
Exe\_User\_Check'EXE\_USER\_CHECK 13  
Exe\_User\_Check'EXE\_USER\_CHECK 3  
Exe\_User\_Check'LOGIN\_USER\_DATA 10  
Exe\_User\_Check'LOGIN\_USER\_DATA 12  
Exe\_User\_Check'LOGIN\_USER\_DATA 13  
Exe\_User\_Check'LOGIN\_USER\_DATA 3  
FORM\_nouhin\_shijibi\_include\_branch'Form\_make 1  
FORM\_nouhin\_shijibi\_include\_branch'Input\_Form\_datas 1  
FORM\_nouhin\_shijibi\_include\_branch'Write\_data 1  
...

## Fairness Properties

---

No infinite occurrence sequences.

# 第6章 おわりに

## 6.1 まとめ

ワークフローシステムの分析と Colored Petri Nets へ自動的に変換可能にするために必要なビジネスプロセス定義の内容の分析を行った。それに対してモデル化を行うための変換方法についての検討と、変換した際に発生する問題とその解決方法の提示をすることにより、ワークフローシステム全体のモデル化が可能であることを示した。本研究の内容は入力項目と出力項目、各ノードの遷移先の分析を行うことによりモデル化出来ることを意味し、他のワークフローシステムに対しても汎用的に適用可能である可能性を示している。また帳票の処理内容を完全に独立して定義しているため他のワークフローシステム独自の機能を簡単に追加して定義することも可能である。ワークフローシステムは複数の要素が同調して動作する仕組みであり、これを一律に表現する方法は今までなかったが、これによりワークフローシステム全体の動きを視覚的に確認することを可能にした。これにより早期の問題の発見と解決を図ることを可能にし、更にはペトリネットの解析技法を適応することで問題の発見の自動化を図ることが可能になる。ビジネスプロセス定義から Colored Petri Nets への自動変換を行い、ペトリネット理論に基づいたワークフローシステムの自動生成が可能になれば、信頼性が高く、動作確認の検証のための人員コストを抑えたものが出来る。検証の自動化を実現するための方法論の一つとして本研究はその準備段階を示せるものであれば良いのではないだろうか。

## 6.2 今後の課題

今後の課題としては以下の事が考えられる。

- 変換したモデルがワークフローシステムを正しくモデル化出来てるかという検証
- WDF から CPN への自動変換ツールの作成が可能であるか。
- ワークフロー記述の標準である WfMC<sup>1</sup>や BPMN<sup>2</sup>との対応関係

---

<sup>1</sup><http://www.wfmc.org/>

<sup>2</sup><http://www.bpmn.org/>

## 参考文献

- [1] WorkCoordinator Definer Version3 ユーザーズガイド手引書 日立製作所
- [2] 戸田保一, 飯島淳一, 速水治夫, 堀内正博 ワークフロー ビジネスプロセスの変革に向けて
- [3] CPN TOOL, <http://www.daimi.au.dk/designCPN/>
- [4] 押手俊 Cosminexus ワークフローシステムの Colored Petri Nets によるモデル化と検証
- [5] 遠藤悟, 宮本俊幸, 藤井拓, 熊谷貞俊, Activity hypergraph を使用した BPEL4WS/UML 記述のシミュレーション手法の提案 京都大学工学部長尾研究室, 1991.
- [6] 山口真悟, 葛崎偉, 田中稔 ペトリネットのワークフローへの応用 山口大学工学部 山口大学教育学部

# 謝辞

本研究を進めるにあたり、度重なるご指導をしていただいた平石邦彦教授に心より深い感謝とお礼を申し上げます。また研究室の皆様方には多くの助言または励ましの言葉を頂いたことに深い感謝とお礼を申し上げます。



# Workflow Definition Forma 定義内容一覧

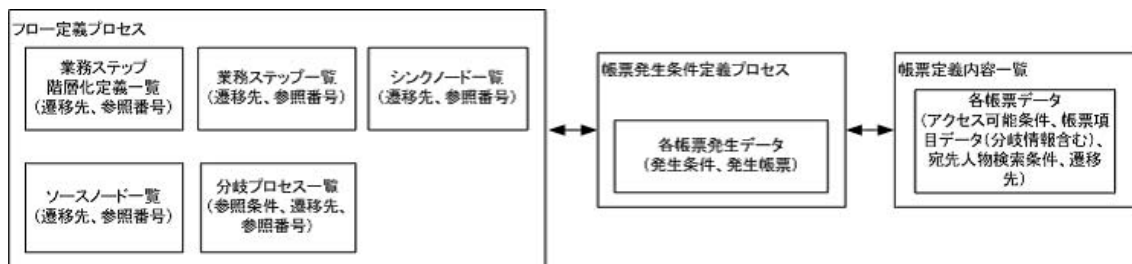


図 6.1: Workflow Definition Format 定義データ構成図

## フロー定義データ一覧

### ソースノード一覧

#### @source(見積～入金プロセス内)

遷移先及び参照番号	備考	参照番号
分岐(Source後),1	分岐プロセス一覧へ,階層化定義開始プロセス	1

### シンクノード一覧へ

#### sink(見積～入金プロセス内)

遷移先及び参照番号	備考	参照番号
見積～入金プロセス,2	階層化定義一覧へ,上位階層へ	1

#### @sink

遷移先及び参照番号	備考	参照番号
無し	ビジネスプロセス終了	1

### 業務ステップ一覧

#### @source

遷移先及び参照番号	備考	参照番号
Source,1	ビジネスプロセス開始、帳票発生条件データ一覧へ	1
先着,1	分岐プロセス一覧へ	2

#### 見積中

遷移先及び参照番号	備考	参照番号
見積中,1	帳票発生条件データ一覧へ、処理開始	1
上位役職承認(見積後),1	業務ステップ一覧へ	2

#### 上位役職承認(見積後)

遷移先及び参照番号	備考	参照番号
上位役職承認(見積後),1	帳票発生条件データ一覧へ、処理開始	1
分岐(見積確認後),1	分岐プロセス一覧へ	2

#### 客先見積確認

遷移先及び参照番号	備考	参照番号
客先見積確認,1	帳票発生条件データ一覧へ、処理開始	1
分岐(見積後),1	分岐プロセス一覧へ	2

**受注処理待ち**

遷移先及び参照番号	備考	参照番号
受注処理待ち,1	帳票発生条件データ一覧へ、処理開始	1
上位役職承認(受注後),1	業務ステップ一覧へ	2

**上位役職認証(受注後)**

遷移先及び参照番号	備考	参照番号
上位役職認証(受注後),1	帳票発生条件データ一覧へ、処理開始	1
分岐(受注後),1	分岐プロセス一覧へ	2

**実行ユーザー選択(請求待ち)**

遷移先及び参照番号	備考	参照番号
実行ユーザー選択(請求待ち),1	帳票発生条件データ一覧へ、処理開始	1
請求待ち,1	業務ステップ一覧へ	2

**実行ユーザー選択(納品待ち)**

遷移先及び参照番号	備考	参照番号
実行ユーザー選択(納品待ち),1	帳票発生条件データ一覧へ、処理開始	1
納品待ち,1	業務ステップ一覧へ	2

**請求待ち**

遷移先及び参照番号	備考	参照番号
請求待ち,1	帳票発生条件データ一覧へ、処理開始	1
待合1(入金前),1	分岐プロセス一覧へ	2

**納品待ち**

遷移先及び参照番号	備考	参照番号
納品待ち,1	帳票発生条件データ一覧へ、処理開始	1
待合1(入金前),1	分岐プロセス一覧へ	2

**入金待ち**

遷移先及び参照番号	備考	参照番号
入金待ち,1	帳票発生条件データ一覧へ、処理開始	1
先着(sink前)	分岐プロセス一覧へ	2

**業務ステップ(階層化定義)****見積～入金プロセス**

遷移先及び参照番号	備考	参照番号
@source(見積～入金プロセス内),1	ソースノード一覧へ、下位階層へ	1
分岐,1	シンクノードより遷移、階層化定義終了	2

## 分岐プロセス一覧

### 先着

分岐条件	遷移先及び参照番号	備考	参照番号
無し	見積～入金プロセス,1	強制遷移、業務ステップ(階層化定義)一覧へ	1

### 分岐 (Source後)

#### 必要DB項目一覧

エンティティ:商談

テーブル名:SYOUDAN

分岐条件	遷移先及び参照番号	備考	参照番号
MITSUMORIKUBUN='0'	先着(受注前),1	見積省略='0'、分岐プロセス一覧へ	1
MITSUMORIKUBUN='1'	先着(見積前),1	一般見積='1'、分岐プロセス一覧へ	1
MITSUMORIKUBUN='2'	先着(見積前),1	特別見積='2'、分岐プロセス一覧へ	1

### 先着(見積前)

分岐条件	遷移先及び参照番号	備考	参照番号
無し	見積中,1	強制遷移、業務ステップ一覧へ	1

### 分岐 (見積認証後)

#### 必要DB項目一覧

エンティティ:見積明細 テーブル名:MITSUMORIMEISAI

分岐条件	遷移先及び参照番号	備考	参照番号
JYOSHIIKAKUNIN_FLAG = 'ON'	客先見積確認,1	見積OK='ON'、業務ステップ一覧へ	1
JYOSHIIKAKUNIN_FLAG = 'OFF'	先着(見積前),1	再見積='OFF'、分岐プロセス一覧へ	1

### 分岐 (見積後)

#### 必要DB項目一覧

エンティティ:見積 テーブル名:MITSUMORI

分岐条件	遷移先及び参照番号	備考	参照番号
MITSUMORIKANRYOUBI IS NOT NULL	先着(受注前),1	見積完了=現時刻、分岐プロセス一覧へ	1
MITSUMORIKANRYOUBI IS NULL	先着(sink前),1	見積再考=NULL、分岐プロセス一覧へ	1

### 先着(受注前)

分岐条件	遷移先及び参照番号	備考	参照番号
無し	受注処理待ち,1	強制遷移、業務ステップ一覧へ	1

**先着(受注前)**

分岐条件	遷移先及び参照番号	備考	参照番号
無し	受注処理待ち,1	強制遷移、業務ステップ一覧へ	1

**分岐(受注後)****必要DB項目一覧**

エンティティ:受注  
テーブル名:JUTYU

分岐条件	遷移先及び参照番号	備考	参照番号
JUTYUKANRYOUBI IS NULL	先着(sink前),1	受注キャンセル、分岐プロセス一覧へ	1
JUTYUKANRYOUBI IS NOT NULL	分岐(受注後),2	受注OK、参照順位2へ	1

エンティティ:受注明細

テーブル名:JUTYUMESAI

分岐条件	遷移先及び参照番号	備考	参照番号
JYOUSHIKAKUNIN_FLAG = 'ON'	分業・納品前ノード,1	受注確認OK='ON'、分岐プロセス一覧へ	2
JYOUSHIKAKUNIN_FLAG = 'OFF'	先着(受注前),1	再受注確認='OFF'、分岐プロセス一覧へ	2

**分業(分業・納品前ノード)**

分岐条件	遷移先	備考	参照番号
無し	実行ユーザー選択(請求待ち)	同時強制遷移、分業ノード、業務ステップ一覧へ	1
無し	実行ユーザー選択(納品待ち)	同時強制遷移、分業ノード、業務ステップ一覧へ	1

**待合1(入金前)**

分岐条件	遷移先及び参照番号	備考	参照番号
無し	入金待ち,1	待合ノード、業務ステップ一覧へ、直前のプロセス待ち	1

**先着(Sink前)**

分岐条件	遷移先及び参照番号	備考	参照番号
無し	@sink(見積～入金プロセス),1	強制遷移、シンクノード一覧へ	1

**分岐****必要DB項目一覧**

エンティティ:受注  
テーブル名:JUTYU

分岐条件	遷移先及び参照番号	備考	参照番号
MITSUMORIKANRYOUBI IS NULL	先着,1	見積見直し経路、分岐プロセス一覧へ	1
MITSUMORIKANRYOUBI IS NOT NULL	分岐,2	分岐プロセス一覧へ	1

エンティティ:受注  
テーブル名:JUTYU

分岐条件	遷移先	備考	参照番号
JUTYUKANRYOUBI IS NULL OR SEIKYUSYOHAKKOUFLAG = 'ON'	@sink,1	受注キャンセル又は受注完了、シンクノード一覧へ	2

## 帳票発生条件データ一覧

### @source

#### 必要DB項目一覧

分岐条件	発生帳票名(又は 遷移先)	備考	参照番号
無し	@source	無条件発生、帳票データ一覧へ	1

### 見積中

#### 必要DB項目一覧

エンティティ:商談

テーブル名:SYOUDAN

分岐条件	発生帳票名(又は 遷移先)	備考	参照番号
MITSUMORIKUBUN='1'	見積中:一般見積作成	一般見積=1、帳票データ一覧へ	1
MITSUMORIKUBUN='2'	見積中:特別見積作成	特別見積=2、帳票データ一覧へ	1

### 上位役職承認(見積後)

#### 必要DB項目一覧

エンティティ:商談 テーブル名:SYOUDAN

分岐条件	発生帳票名(又は 遷移先)	備考	参照番号
MITSUMORIKUBUN='1'	上位役職帳票確認(一般見積)	一般見積=1、帳票データ一覧へ	1
MITSUMORIKUBUN='2'	上位役職帳票確認(特別見積)	特別見積=2、帳票データ一覧へ	1

### 客先見積確認

#### 必要DB項目一覧

エンティティ:商談 テーブル名:SYOUDAN

分岐条件	発生帳票名(又は 遷移先)	備考	参照番号
MITSUMORIKUBUN='1'	客先見積確認(一般)	一般見積=1、帳票データ一覧へ	1
MITSUMORIKUBUN='2'	客先見積確認(特別)	特別見積=2、帳票データ一覧へ	1

### 受注処理待ち

#### 必要DB項目一覧

分岐条件	発生帳票名(又は 遷移先)	備考	参照番号
無し	受注確認	無条件発生、帳票データ一覧へ	1

エンティティ:受注

テーブル名:JUTYU

分岐条件	発生帳票名(又は 遷移先)	備考	参照番号
JUTYUKANRYOUBI IS NOT NULL(JUTYUKANRYOUBI = DATE)	受注登録	受注完了日=現日時、帳票データ一覧へ	2
JUTYUKANRYOUBI IS NULL	上位役職認証(受注後)	受注完了日=NULL、フロー定義データへ(業務ステップ一覧)	2

### 上位役職認証(受注後)

#### 必要DB項目一覧

エンティティ:受注 テーブル名:JUTYU

分岐条件	発生帳票名(又は 遷移先)	備考	参照番号
JUTYUKANRYOUBI IS NOT NULL	受注明細確認認証	受注完了日=現日時、帳票データ一覧へ	1
JUTYUKANRYOUBI IS NULL	分岐(受注後)	受注キャンセル、フロー定義データ一覧へ(分岐プロセス一覧)	1

**実行ユーザー選択(請求待ち)**

**必要DB項目一覧**

分岐条件	発生帳票名(又は遷移先)	備考	参照番号
無し	実行ユーザー選択(請求待ち)	無条件発生、帳票データ一覧へ	1

**実行ユーザー選択(納品待ち)**

**必要DB項目一覧**

分岐条件	発生帳票名(又は遷移先)	備考	参照番号
無し	実行ユーザー選択(納品待ち)	無条件発生、帳票データ一覧へ	1

**請求待ち**

**必要DB項目一覧**

分岐条件	発生帳票名(又は遷移先)	備考	参照番号
無し	売掛・請求レコード作成	無条件発生、帳票データ一覧へ	1

**納品待ち**

**必要DB項目一覧**

分岐条件	発生帳票名(又は遷移先)	備考	参照番号
無し	納入指示日	強制遷移、帳票データ一覧へ	1

エンティティ:納入明細 テーブル名:NOUNYUMESAI

分岐条件	発生帳票名(又は遷移先)	備考	参照番号
NOUHINSIJIBI IS NULL	納入指示日	納入指示保留、帳票データ一覧へ	2
NOUHINSIJIBI IS NOT NULL , NOUNYUBI IS NULL	納品子作業	帳票データ一覧へ	2

**入金待ち**

**必要DB項目一覧**

分岐条件	発生帳票名(又は遷移先)	備考	参照番号
無し	請求書出力	入金待ち中、帳票データ一覧へ	1

エンティティ:受注 テーブル名:JUTYU

SEIKYUSYOHAKKOUFLAG = 'OFF'	請求書出力	入金待ち中、帳票データ一覧へ	2
SEIKYUSYOHAKKOUFLAG = 'ON'	入金登録	帳票データ一覧へ	2

## 帳票データ一覧

**見積開始:@Source**

この帳票にアクセス可能な条件

部門	役職	備考
営業部	平社員	

**必要DB項目一覧**

エンティティ:商談

テーブル名:SYOUDAN

属性名	カラム名	キー	データ型	入出力属性	備考
商談番号	SYOUDANBANGOU	主キー	文字列	出力	案件キー
担当者コード	TANTOUSYACODE	外来キー	文字列	出力	
顧客コード	KOKYAKUCODE		文字列	出力	
見種区分	MITSUMORIKUBUN		文字列	出力	見種省略=0' or 一般見種=1' or 特別見種=2' 分岐データ
商談名	SYOUDANMEI		文字列	出力	
顧客予算	KOKYAKUYOSAN		文字列	出力	

**宛先人物検索条件**

部門	役職	次処理帳票	備考
営業部	平社員	受注確認	
営業部	平社員	一般見種作成	
営業部	平社員	特別見種作成	

**遷移先一覧**

遷移先及び参照番号	備考
Source,2	フロー定義データ一覧へ(業務ステッパー一覧へ)

**見積中：一般見積作成**

この帳票にアクセス可能な条件

部門	役職	備考
営業部	平社員	

**必要DB項目一覧**

エンティティ:見積

テーブル名: MITSUMORI

属性名	カラム名	キー	データ型	入出力属性	備考
商談番号	SYOUDANBANGOU	主キー	文字列	出力	案件キー
見積完了フラグ	MITSUMORIKANRYOUFLAG		文字列	出力	見積完了='ON'
見積納期	MITSUMORINOUKI		文字列	出力	
見積名称	MITSUMORIMEISYOU		文字列	出力	
見積確度	MITSUMORIKAKUDO		文字列	出力	

エンティティ:見積明細

テーブル名: MITSUMORIMEISAI

属性名	カラム名	キー	データ型	入出力属性	備考
見積明細番号	MITSUMORIMEISAI BANGOU	主キー	文字列	出力	案件キー
商談番号	SYOUDANBANGOU		文字列	出力	
商品コード	SYOUIHNCODE		文字列	出力	
見積数量	MITSUMORISUURYOU		文字列	出力	
見積金額	MITSUMORIKINGAKU		文字列	出力	

**宛先人物検索条件**

部門	役職	次処理帳票	備考
営業部	課長	上位役職帳票確認(一般見積)	

**遷移先一覧**

遷移先及び参照番号	備考
見積中2	フロー定義データ一覧へ(業務ステップ一覧へ)

**見積中：特別見積作成**

この帳票にアクセス可能な条件

部門	役職	備考
営業部	平社員	

**必要DB項目一覧**

エンティティ:見積

テーブル名: MITSUMORI

属性名	カラム名	キー	データ型	入出力属性	備考
商談番号	SYOUDANBANGOU	主キー	文字列	出力	案件キー
見積完了フラグ	MITSUMORIKANRYOUFLAG		文字列	出力	見積完了='ON'
見積納期	MITSUMORINOUKI		文字列	出力	
見積名称	MITSUMORIMEISYOU		文字列	出力	
見積確度	MITSUMORIKAKUDO		文字列	出力	

エンティティ:見積明細

テーブル名: MITSUMORIMEISAI

属性名	カラム名	キー	データ型	入出力属性	備考
見積明細番号	MITSUMORIMEISAI BANGOU	主キー	文字列	出力	案件キー
商談番号	SYOUDANBANGOU		文字列	出力	
商品コード	SYOUIHNCODE		文字列	出力	
見積数量	MITSUMORISUURYOU		文字列	出力	
見積金額	MITSUMORIKINGAKU		文字列	出力	

**宛先人物検索条件**

部門	役職	次処理帳票	備考
営業部	課長	上位役職帳票確認(特別見積)	

**遷移先一覧**

遷移先及び参照番号	備考
見積中2	フロー定義データ一覧へ(業務ステップ一覧へ)

**上位役職帳票確認(一般見種)**

この帳票にアクセス可能な条件

部門	役職	備考
営業部	課長	

**必要DB項目一覧**

エンティティ:見種明細

テーブル名: MITSUMORIMESAI

属性名	カラム名	キー	データ型	入出力属性	備考
上司確認フラグ	JYOUSHIKAKUNIN_FLAG		文字列	出力	再見種=OFF or 見種OK=ON 分岐データ

**宛先人物検索条件**

部門	役職	次処理帳票	備考
営業部	平社員	客先見種確認(一般)	
営業部	平社員	見種中:一般見種作成	

**遷移先一覧**

遷移先及び参照番号	備考
上位役職承認(見種後)2	フロー定義データ一覧へ(業務ステップ一覧へ)

**上位役職帳票確認(特別見種)**

この帳票にアクセス可能な条件

部門	役職	備考
営業部	課長	

**必要DB項目一覧**

エンティティ:見種明細

テーブル名: MITSUMORIMESAI

属性名	カラム名	キー	データ型	入出力属性	備考
上司確認フラグ	JYOUSHIKAKUNIN_FLAG		文字列	出力	再見種=OFF or 見種OK=ON 分岐データ

**宛先人物検索条件**

部門	役職	次処理帳票	備考
営業部	平社員	客先見種確認(特別)	
営業部	平社員	見種中:特別見種作成	

**遷移先一覧**

遷移先及び参照番号	備考
上位役職承認(見種後)2	フロー定義データ一覧へ(業務ステップ一覧へ)

**客先見種確認(一般)**

この帳票にアクセス可能な条件

部門	役職	備考
営業部	平社員	

**必要DB項目一覧**

エンティティ:見種

テーブル名: MITSUMORI

属性名	カラム名	キー	データ型	入出力属性	備考
見種完了日	MITSUMORIKANRYOUBI		文字列	出力	
見種完了フラグ	MITSUMORIKANRYOUFLAG		文字列	出力	客先見種完了=ON2' 見種再考=DEL' 分岐データ

**宛先人物検索条件**

部門	役職	次処理帳票	備考
営業部	平社員	見種中:一般見種作成	
営業部	平社員	受注確認	

**遷移先一覧**

遷移先及び参照番号	備考
客先見種確認2	フロー定義データ一覧へ(業務ステップ一覧へ)

**客先見種確認(特別)**

この帳票にアクセス可能な条件

部門	役職	備考
営業部	平社員	

**必要DB項目一覧**

エンティティ:見種明

テーブル名: MITSUMORI

属性名	カラム名	キー	データ型	入出力属性	備考
見種完了日	MITSUMORIKANRYOUBI		文字列	出力	
見種完了フラグ	MITSUMORIKANRYOUFLAG		文字列	出力	客先見種完了=ON2' 見種再考=DEL' 分岐データ

**宛先人物検索条件**

部門	役職	次処理帳票	備考
営業部	平社員	見種中:特別見種作成	
営業部	平社員	受注確認	

**遷移先一覧**

遷移先及び参照番号	備考
客先見種確認2	フロー定義データ一覧へ(業務ステップ一覧へ)



### 受注確認

この帳票にアクセス可能な条件

部門	役職	備考
営業部	平社員	

### 必要DB項目一覧

エンティティ:受注

テーブル名:JUTYU

属性名	カラム名	キー	データ型	入出力属性	備考
商談番号	SYOUDANBANGOU	主キー、外来キー	文字列	出力	案件キー
受注完了日	JUTYUKANRYOUEI		文字列	出力	受注完了=DATA、受注キャンセル=NULL 分岐データ
受注納期	JUTYUNOUKI				
売上計上区分	URIAGEKEIJYOKUBUN				

### 宛先人物検索条件

部門	役職	次処理帳票	備考
営業部	平社員	受注登録	
無し	無し	sink	キャンセルにつき部門、役職ともに指定無し

### 遷移先一覧

遷移先及び参照番号	備考
受注処理待ち2	帳票発生条件データ一覧

### 受注登録

この帳票にアクセス可能な条件

部門	役職	備考
営業部	平社員	

### 必要DB項目一覧

エンティティ:受注明細

テーブル名:JUTYUMESAI

属性名	カラム名	キー	データ型	入出力属性	備考
受注明細番号	JUTYUMESAIBANGOU	主キー	文字列	出力	
商談番号	SYOUDANBANGOU	主キー、外来キー	文字列	出力	
商品コード	SYOUSHINCOD		文字列	出力	
受注数量	JUTYUSUURYOU		文字列	出力	
受注金額	JUTYUKINGAKU		文字列	出力	

### 宛先人物検索条件

部門	役職	次処理帳票	備考
営業部	課長	受注明細確認	

### 遷移先一覧

遷移先及び参照番号	備考
受注処理待ち2	フロー定義データ一覧へ(業務ステッパー一覧へ)

### 受注明細確認

この帳票にアクセス可能な条件

部門	役職	備考
営業部	課長	

### 必要DB項目一覧

エンティティ:受注明細

テーブル名:JUTYUMESAI

属性名	カラム名	キー	データ型	入出力属性	備考
上司確認フラグ	JYOUSHIKAKUNIN_FLAG		文字列	出力	確認OK="ON",確認NG="OFF" 分岐データ

### 宛先人物検索条件

部門	役職	次処理帳票	備考
営業部	課長	分業納品前ノード,受注確認	分業ノードを選択し、自分のトークンを増加

### 遷移先一覧

遷移先及び参照番号	備考
上位役職確認(受注後)2	フロー定義データ一覧へ(業務ステッパー一覧へ)

### 実行ユーザー選択(請求待ち)

この帳票にアクセス可能な条件

部門	役職	備考
営業部	課長	

### 必要DB項目一覧

無し

### 宛先人物検索条件

部門	役職	次処理帳票	備考
経理部	平社員	売掛・請求レコード作成	

### 遷移先一覧

遷移先及び参照番号	備考
実行ユーザー選択(請求待ち)2	フロー定義データ一覧へ(業務ステッパー一覧へ)

**実行ユーザー選択(納品待ち)**

この帳票にアクセス可能な条件

部門	役職	備考
営業部	課長	

**必要DB項目一覧**

無し

**宛先人物検索条件**

部門	役職	次処理帳票	備考
営業部	平社員	納品指示日	

**遷移先一覧**

遷移先及び参照番号	備考
実行ユーザー選択(納品待ち)2	フロー定義データ一覧へ(業務ステップ一覧へ)

**売掛・請求レコード作成**

この帳票にアクセス可能な条件

部門	役職	備考
経理部	平社員	

**必要DB項目一覧**

エンティティ:請求明細

テーブル名:SEIKYUMEISAI	属性名	カラム名	キー	データ型	入出力属性	備考
	請求明細番号	SEIKYUMEISAIKANGOU	主キー	文字列	出力	
	商談番号	SYOUDANBANGOU	主キー, 外来キー	文字列	出力	案件キー
	商品コード	SYOHINC CODE		文字列	出力	
	請求金額	SEIKYUKINGAKU		文字列	出力	

**宛先人物検索条件**

部門	役職	次処理帳票	備考
経理部	平社員	請求書出力	

**遷移先一覧**

遷移先及び参照番号	備考
請求待ち2	フロー定義データ一覧へ(業務ステップ一覧へ)

**納品指示日**

この帳票にアクセス可能な条件

部門	役職	備考
営業部	平社員	

**必要DB項目一覧**

エンティティ:納入明細

テーブル名:NOUNYUMESAI	属性名	カラム名	キー	データ型	入出力属性	備考
	納入明細番号	NOUNYUMESAIKANGOU	主キー	文字列	出力	
	商談番号	SYOUDANBANGOU	主キー, 外来キー	文字列	出力	案件キー
	商品コード	SYOHINC CODE		文字列	出力	
	納品指示日	NOUHINSI JIBI		文字列	出力	納入指示完了=日時, 納入指示保留=NULL' 分岐データ
	納入数量	NOUNYUSUURYOU		数値	出力	

**宛先人物検索条件**

部門	役職	次処理帳票	備考
無し	平社員		遷移先プロセスにあわせて任意選択

備考(実際に実行したい振り分け)

部門	役職	次処理帳票	備考
出荷部	平社員	納品子作業	納品指示日決定, 次のプロセスへ
営業部	平社員	納品指示日	納品指示日未決定, ループ

**遷移先一覧**

遷移先及び参照番号	備考
納品待ち2	帳票発生条件データ一覧

### 納入子作業

この帳票にアクセス可能な条件

部門	役職	備考
出荷部	平社員	

### 必要DB項目一覧

エンティティ:納入明細

テーブル名: NOUNYUMESAI

属性名	カラム名	キー	データ型	入出力属性	備考
納入日	NOUNYUBI		文字列	出力	納入予定日

エンティティ:受注 テーブル名: JUTYU

属性名	カラム名	キー	データ型	入出力属性	備考
納入完了日	NOUNYUKANRYOUBI		文字列	出力	
納品指示完了フラグ	NOUHINSIJIKANRYOUFLAG		文字列	出力	納入完了後='ON',納入前='NULL'

### 宛先人物検索条件

部門	役職	次処理帳票	備考
経理部	平社員	請求書出力	

### 遷移先一覧

遷移先及び参照番号	備考
納品待ち2	フロー定義データ一覧へ(分岐データ一覧)

### 請求書出力

この帳票にアクセス可能な条件

部門	役職	備考
経理部	平社員	

### 必要DB項目一覧

エンティティ:請求明細

テーブル名: SEIKYUMESAI

属性名	カラム名	キー	データ型	入出力属性	備考
請求明細番号	SEIKYUMESAIBANGOU	主キー	文字列	入力	
商談番号	SYOUDANBANGOU	主キー, 外来	文字列	入力	
商品コード	SYOUHINCODE		文字列	入力	
請求金額	SEIKYUKINGAKU		文字列	入力	

### 必要DB項目一覧

エンティティ:受注

テーブル名: JUTYU

属性名	カラム名	キー	データ型	入出力属性	備考
請求書発行フラグ	SEIKYUSYOHAKKOUFLAG	主キー	文字列	出力	請求書出力='ON',請求書未出力='NULL',分岐条件

### 宛先人物検索条件

部門	役職	次処理帳票	備考
経理部	平社員	入金登録	

### 遷移先一覧

遷移先及び参照番号	備考
入金待ち2	帳票発生条件データ一覧

### 入金登録

この帳票にアクセス可能な条件

部門	役職	備考
経理部	平社員	

### 必要DB項目一覧

エンティティ:請求明細

テーブル名: SEIKYUMESAI

属性名	カラム名	キー	データ型	入出力属性	備考
入金金額	NYUKINKINGAKU		文字列	出力	
入金日	NYUKINBI		文字列	出力	

### 宛先人物検索条件

備考	
無し	最終プロセスにつき不必要

### 遷移先一覧

遷移先及び参照番号	備考
入金待ち2	フロー定義データ一覧へ(分岐データ一覧)