

3.2 各要素における個別ルール

3.2.1 インタフェースのルール

3.2.1.1 内部インタフェースのルール

内部インタフェースとは、内部システムの各サブシステム内及びサブシステム間の連携に使用されるインタフェースである。内部システムの連携を以下に示す。

- アクセス先のシステム構成要素が提供するサービスインタフェースを介して連携する。
- ブラウザ又はリッチクライアントからプレゼンテーションロジックに、HTTP等のプロトコルでアクセスする。
- プレゼンテーションロジックから業務アプリケーション(ユーザ)をメソッド呼び出しする。
- 個別データベース又は共有データベースに接続し、SQL文を使用してデータを参照・更新する。

以降に、サービス化の指針、プロトコル及び電文形式のルールを示す。

3.2.1.1.1 サービス化の指針

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.2

指針1： 「(4)サービス化の対象」に示すシステム構成要素の提供機能は、「(3)サービス化対象外の指針」に該当しない場合、サービスインタフェースを設けるよう設計すること。

特例1： 「(3)サービス化対象外の指針」に該当するものについては、サービス化の対象外とすることを許容する。

規約1： サービスインタフェースは、「(5)サービスインタフェースの提供」のとおり提供すること。

SOAを導入して環境変化に対して短期間で迅速、柔軟にシステム対応するため、各システム構成要素は一部の機能をサービスとして提供する。

(1) サービスの条件

サービスは、条件として下記のようなインタフェースを提供しなくてはならないものとする。

- 共通的に利用されることを想定し、インタフェース仕様が管理されている。
- 予め定められた統一的な方法によって利用側からアクセスすることができる。
- 業界標準的なプロトコルを用い、ネットワーク経由でアクセスできる。
- 実装プラットフォームや開発言語の影響を受けずにアクセスできる。

(2) サービス化の指針

サービスとするためにシステム構成要素に1つ以上のサービスインタフェースを設けることを「サービス化」と呼ぶ。

サービス化の指針を以下に示す。

- 他のサブシステムからアクセスされるシステム構成要素をサービス化する。
- サブシステム内で再利用性、交換可能性が求められるシステム構成要素をサービス化する。

(3) サービス化対象外の指針

業務特性や技術的理由により、サービス化しない方がよい場合もある。サービス化の対象外とする指針を以下に示す。

以下の条件に当てはまる場合は、サービス化の対象外とする。

- 「(1)サービスの条件」で示す条件を実現できない場合
- 高度なりアルタイム性が求められるような場合
- 高頻度で利用され、サービスにアクセスするオーバーヘッドが処理時間の多くを占め、オーバーヘッドを許容できない場合
- 高い信頼性が求められ、サービスとした場合の実現コストが非常に高い場合
※例えば、複数のサービスをまたがって処理が実行される場合、アクセスするサービス数が多いほど信頼性は低下する。高い信頼性を確保するためにはハードウェアのコスト増につながる可能性がある。
- 複数のサービスにまたがるトランザクション処理や復旧処理が必要となったり、処理のオーバーヘッドによって要求性能が満たせない場合
- 複数の分かれた操作にまたがってACID特性が求められるような場合

(4) サービス化の対象

サービス化の指針を踏まえたサービス化対象を下表に示す。

ただし、「(3)サービス化対象外の指針」に該当する場合は、特例としてサービスとしないことを許容する。

表 3.2-1 サービス化対象

項番	サービス化対象のシステム構成要素	提供機能	サービスインタフェースの利用を許可するシステム構成要素
1	BPMS	ワークフローの開始, イベント通知等	ESB BPMS BPMS補完機能 業務アプリケーション(システム) 業務アプリケーション(バッチ)
2		ワークフローの状態取得, ユーザタスクの進行等	プレゼンテーションロジック
3	BPMS補完機能	ワークフローの開始, イベント通知等	ESB 業務アプリケーション(システム) 業務アプリケーション(バッチ)
4		ワークフローの状態取得, ユーザタスクの進行等	BPMS プレゼンテーションロジック
5	業務アプリケーション(システム)	BPMSのサービスタスクに対応するビジネスロジックの実行。 ※業務単位のサービス	BPMS ESB
6	※業務単位のサービスについては、「3.2.4.2 サービスインタフェースの粒度 (1) サービスインタフェースの粒度のルール」参照	BPMSのサービスタスクに対応するビジネスロジックの実行。	BPMS補完機能
7	※サービスインタフェースの利用を許可するシステム構成要素については「3.2.4.2 サービスインタフェースの粒度 (3) アクセスパスの制限事項に関するルール」参照。	個別データの取得及び更新のためのビジネスロジック実行。	プレゼンテーションロジック ※業務アプリケーション(システム)とは異なるサブシステムからの利用に制限する。
8	※サービスインタフェースの利用を許可するシステム構成要素については「3.2.4.2 サービスインタフェースの粒度 (3) アクセスパスの制限事項に関するルール」参照。	共通リソースデータの取得及び更新のための共通的に利用するビジネスロジックの実行。	プレゼンテーションロジック ※業務アプリケーション(システム)とは異なるサブシステムからの利用に制限する。 業務アプリケーション(システム) ※業務単位のサービスからの利用に制限する。 業務アプリケーション(ユーザ) 業務アプリケーション(バッチ) ESB
9	BRMS	ビジネスルールの実行	業務アプリケーション(ユーザ) 業務アプリケーション(システム) 業務アプリケーション(バッチ)
10	ESB	外部システムとの連携	BPMS BPMS補完機能 業務アプリケーション(ユーザ) 業務アプリケーション(システム) 業務アプリケーション(バッチ)
11	DBアクセス基盤サービス	共有データベースへの参照・更新 ※1つの操作にトランザクションが閉じている操作。	業務アプリケーション(ユーザ) 業務アプリケーション(システム) 業務アプリケーション(バッチ) ESB 外部システム互換機能

(5) サービスインタフェースの提供

サービスインタフェースを以下のとおり提供すること。

- ステートレスなサービスインタフェースを提供すること(セッションを使用しない¹⁸。)。

¹⁸ ここでいうセッションは、ServletのHttpSession等、特定の実装手段に限らず、同じ送信元からの複数のリクエストであることを識別するために、ある定まった期間に行われる一連の通信をひとまとまりのものとして扱う仕組み全般を指す。

3.2.1.1.2 内部インタフェースの Protokol

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.2

規約1：「表 3.2-2 Protokol及び電文形式(階層定型化サブシステム)」及び「表 3.2-3 Protokol及び電文形式(インタフェース定型化サブシステム)」に示すシステム構成要素は、当該表に示す Protokol及び電文形式を利用するインタフェースを設けること。

推奨1：「表 3.2-2 Protokol及び電文形式(階層定型化サブシステム)」及び「表 3.2-3 Protokol及び電文形式(インタフェース定型化サブシステム)」に示す推奨 Protokol及び電文形式を推奨する。

特例1：「表 3.2-2 Protokol及び電文形式(階層定型化サブシステム)」及び「表 3.2-3 Protokol及び電文形式(インタフェース定型化サブシステム)」に示す特例を許容する。

規約2：FTP(S) Protokolは、「(5)FTP(S) Protokolの利用制限事項」に示すとおり利用すること。

(1) Protokolの適用指針

内部インタフェースの Protokolの適用指針を以下に示す。

- 原則、同期呼び出しでステートレスな Protokolとして最も一般的に利用されている Protokolである HTTP(S)を利用する。
 - 大容量ファイルを効率よく転送したい場合や、一括処理のため複数件数のレコードデータを1つのファイルにまとめてやり取りしたい場合等、HTTP(S)では、効率が悪い場合は、ファイル転送 Protokolとして最も一般的に利用されているFTP(S)の利用を許容する。
 - HTTPSやFTPSは、セキュリティ上、データを保護したい場合に使用する。
- 特定の用途で利用される業界標準的な Protokolが存在する場合は、その Protokolを利用する。
例えば、ユーザ情報を取得する機能(認証機能等)は、業界標準であるLDAPを利用する。
なお、保守性や移植性よりも、効率性、信頼性等の品質特性を優先する場合は、製品が定める Protokolをそのまま利用することを許容する。
ただし、製品が定める Protokolをそのまま利用する場合には、業務アプリケーション内のビジネスロジックが製品の提供する通信ライブラリの独自APIを直接呼び出し、特定の製品に依存してしまうといったことがないようにするため、製品に依存しない汎用的なAPIでラップするものとする。

(2) HTTP(S)の電文形式の適用指針

HTTP(S)の電文形式は、原則、最も一般的な電文形式であり、インタフェース仕様を厳密に定義できることで多業者間の仕様認識齟齬を防ぐ効果が期待できるSOAPで規定される電文形式を利用する。

ただし、以下の場合、他の電文形式も許容する。

- アクセス元がブラウザの場合、Webページの表示のためHTMLやCSSを許容する。また、Ajaxを利用して非同期的にデータを授受する場合はJSON、XMLを許容する。
- リソース指向のインタフェースを採用し利用者に直感的に分かりやすいインタフェースを提供したい場合や、通信のオーバーヘッドを極力減らしたいといった理由がある場合にはREST¹⁹を採用し、電文形式としてJSONやXMLを許容する。
- 複数のサブシステムから共通的に利用されるインタフェースで、データ項目の追加等の際に、既存の利用側サブシステムへの影響を与えにくくするといった理由がある場合には、上記同様RESTを採用する。
- UI層とプレゼンテーション層間の通信では、以下の場合に限りセッションの利用を許容する。
 - 認証・認可に関する情報を保持する場合
 - 一時的に使われる、秘匿性が高い情報をリクエスト間で受け渡す場合
 - 処理の冗長化や性能低下を防ぐ目的で一時的に使われる情報を保持する場合(ユーザの入力値や検索結果の業務処理結果を保持等)

¹⁹ Representational State Transferとして知られる、Webアプリケーションのソフトウェアアーキテクチャのスタイル。本書におけるRESTは、HTTPメソッド(GET, POST, PUT, DELETE等)を用いて対象となる情報(リソース)に対する操作を表現するリソース指向のインタフェースのことを指す。特に断りの無い限りは、XML形式の電文を送受するものとする。

- 業務的要件, システム的要件を実現するためにセッション保持が必要となる場合(不正な画面遷移を防止する目的で, 識別子の保持にセッションを使用する等)

(3) 階層定型化サブシステムが提供するインタフェースのプロトコル及び電文形式

「(1)プロトコルの適用指針」及び「(2)HTTP(S)の電文形式の適用指針」に基づき、階層定型化サブシステムが提供するインタフェースのプロトコル及び電文形式を下表に示す。なお、業務システム(ユーザ)はプレゼンテーションロジックからのメソッド呼び出しとなるためプロトコルを定義する対象外である。

表 3.2-2 プロトコル及び電文形式(階層定型化サブシステム)

項番	インタフェースの提供側システム構成要素	プロトコル	セッション利用可否	電文形式
1	プレゼンテーションロジック (ブラウザからアクセスする場合)	HTTP(S)とする。	○	原則、HTMLやCSSとする。 特例として、Ajaxを利用して非同期にデータを授受する場合は、JSONやXMLを利用することを許容する。
2	プレゼンテーションロジック (リッチクライアントからアクセスする場合)	HTTP(S)とする。	○	原則、SOAPで規定される電文形式とする。 特例として、リソース指向のインタフェースを採用し利用者に直感的に分かりやすいインタフェースを提供したい場合や、通信のオーバーヘッドを極力減らしたい場合、RESTを利用することを許容する。RESTでデータを授受する場合は、JSON又はXMLを利用する。
3	BPMS BPMS補完機能	原則、HTTP(S)とする。 特例として、ヒューマンタスクのステータスを参照・更新する等、BPMS製品が提供するワークフローAPIを使用しアクセスしなければならない場合には、ワークフローAPIが定めるプロトコルとする。	×	XML形式とする。 特例として、BPMS製品が提供するワークフローAPIを使用しなければならない場合には、そのワークフローAPIが使用する電文形式を利用する。
4	業務アプリケーション(システム)	HTTP(S)とする。	×	共通リソースデータの取得及び更新のためのインタフェースの場合はRESTを採用し、データ形式はXMLとする。 上記以外の場合はSOAPで規定される電文形式とする。
5	業務アプリケーション(バッチ) (運用監視サブシステムのジョブ管理システムからアクセスする場合)	運用監視サブシステムのジョブ管理システムの製品が対応するプロトコルとする。	—	運用監視サブシステムのジョブ管理システムの製品が対応する電文形式とする。
6	業務アプリケーション(バッチ) (他サブシステムの業務アプリケーション(バッチ)からアクセスする場合)	FTP(S)を利用する。ただし、以下の場合に限る。 使用条件:大容量ファイルを効率よく転送したい場合や一括処理のため複数件数のレコードデータを1つのファイルにまとめてやり取りしたい場合等、データベース経由では、効率が極めて悪い場合。	—	FTP(S)を利用する場合、データ形式は、テキストデータ(CSV等)とする。 また、特定の用途で利用される業界標準的な形式(イメージデータであればJPEGやTIFF等、圧縮ファイルであればZIP等)を利用する。

セッション利用可否の凡例：○:セッション利用可、×:セッション利用不可、—:該当無し

項番	インタフェースの提供側システム構成要素	プロトコル	セッション利用可否	電文形式
7	個別データベース (業務アプリケーション層の業務アプリケーションからアクセスする場合)	RDBMS製品が使用するプロトコルとする。	—	—
8	個別データベース (情報活用系サブシステムへ情報を提供する目的で、ETLからアクセスする場合)			
9	DBアクセス基盤サービス	HTTP(S)とする。	×	RESTとする。応答データ形式は、XML又はバイナリである。詳細は、『DBアクセス基盤 利用者向けガイドライン 5.基盤サービス』を参照のこと。
10	共有データベース (業務アプリケーション層の業務アプリケーションからアクセスする場合)	RDBMS製品が使用するプロトコルとする。 ※「3.2.1.1.1(3)サービス化対象外の指針」に該当するため、共有データベースへ直接アクセスする場合に採用するプロトコル	—	—
11	共有データベース (情報活用系サブシステムへ情報を提供する目的で、ETLからアクセスする場合)	RDBMS製品が使用するプロトコルとする。	—	—

セッション利用可否の凡例：○:セッション利用可，×:セッション利用不可，—:該当無し

(4) インタフェース定型化サブシステムが階層定型化サブシステムに提供するインタフェースの Protokol 及び電文形式

「(1)Protokolの適用指針」及び「(2)HTTP(S)の電文形式の適用指針」に基づき、インタフェース定型化サブシステムが階層定型化サブシステムに提供するインタフェースのProtokol及び電文形式を下表に示す。

表 3.2-3 Protokol及び電文形式(インタフェース定型化サブシステム)

項番	インタフェースの提供側サブシステム	Protokol	電文形式
1	外部システム連携	HTTP(S)又はFTP(S)とする。 FTP(S)の利用は以下の場合に限る。 使用条件: ● 他サブシステムの業務アプリケーション(バッチ)からESBにアクセスする場合。 ● 大容量ファイルを効率よく転送したい場合や、一括処理のため複数件数のレコードデータを1つのファイルにまとめてやり取りしたい場合等、HTTP(S)では、効率が極めて悪い場合。	HTTP(S)の場合、原則、SOAPで規定される電文形式とする。特例として、リソース指向のインタフェースを採用し利用者に直感的に分かりやすいインタフェースを提供したい場合や、通信のオーバーヘッドを極力減らしたい場合、RESTを利用することを許容する。RESTでデータを授受する場合は、JSON又はXMLを利用する。 FTP(S)を利用する場合、データ形式は、テキストデータ(CSV等)とする。また、特定の用途で利用される業界標準的な形式(イメージデータであればJPEGやTIFF等、圧縮ファイルであればZIP等)を利用する。
2	リソース管理	LDAPとする。	左記に対応する電文形式とする。
3	ビジネスルール管理	原則、HTTP(S)とする。 特例として、性能面の問題を調査し、リスクが高いと判断された場合は、BRMS製品が提供するAPIの呼び出しを許容する。 詳細は「3.2.7.1.2 BRMSの呼び出し方式」を参照のこと。	XML形式とする。SOAPで規定される電文形式を推奨する。 ただし、API呼び出しの場合を除く。

(5) FTP(S)Protokolの利用制限事項

FTP(S)は、「表 3.2-2 Protokol及び電文形式(階層定型化サブシステム)」及び「表 3.2-3 Protokol及び電文形式(インタフェース定型化サブシステム)」に示すシステム構成要素又はインタフェース定型化サブシステムが提供するインタフェースに使用される場合がある。

ToBeアーキテクチャでは、FTP(S)の利用を以下に示すシステム構成要素間の連携に制限する。

- 業務アプリケーション(バッチ)と業務アプリケーション(バッチ)との間の連携
- 業務アプリケーション(バッチ)とESBとの間の連携

3.2.1.2 外部インタフェースのルール

3.2.1.2.1 外部インタフェースの基本的なルール

外部インタフェースと内部インタフェースの連携方式の違い(ギャップ)が発生することが想定されるが、「3.1 技術方式」に示したとおり、外部システム連携層がギャップを吸収する層としての責務を負う。当該層の詳細については、「3.2.5.1.2 外部システム連携層に持たせる機能」を参照のこと。

3.2.1.3 インタフェースの命名ルール

他サブシステムが誤解せず利用（設計）できるようにすることを目的に、以下に対し、論理名の共通的な命名ルールを定める。

- インタフェース名
- 入出力項目名

3.2.1.3.1 インタフェース名の命名ルール

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.2

指針1： インタフェース名は、「(1)インタフェース名の構成」に示す用語の組み合わせに従い、命名すること。

指針2： インタフェース名は、「(2)インタフェース名の重複と同音異義・異音同義語の禁止事項」に示す禁止事項に抵触しないよう、命名すること。

(1) インタフェース名の構成

インタフェース名は、以下の用語の組み合わせで構成すること。

インタフェース名 = 特性名 + 動作名 + 修飾子(0個以上)

※1 特性名や動作名、修飾子等の用語の種類については、「3.2.8.1 用語のルール」を参照のこと。

※2 各用語の区切り文字は、アンダースコア(「_」)する。必要に応じて、追加で区切り文字を定義してもよい。その場合は、サブシステムで利用方法を統一すること。

※3 修飾子を複数使う場合は、「特性名 + 修飾子」を修飾する場合等に用いる。

インタフェース名の例を下表に示す。

表 3.2-4 インタフェース名の例

項番	パターン	インタフェース名の例
1	特性名 + 動作名	「書類チェック」, 「願書受付登録」等
2	特性名 + 動作名 + 修飾子	「書類チェック_一括」, 「願書受付登録_本受付」等

例えば、外部システムにデータを送信し、その後、受信するインタフェースの場合、データ名を特性名としてインタフェース名を定義することで、送受信の関係が分かりやすくなる。インタフェース名の例を下図に示す。

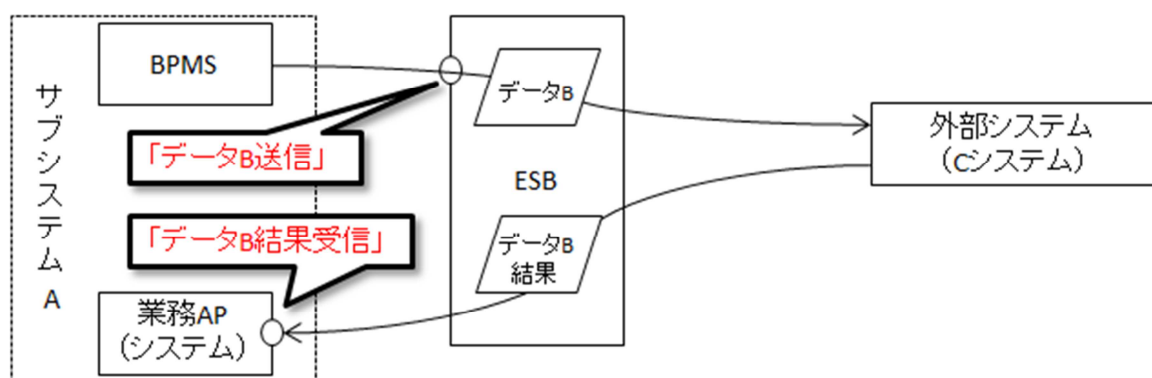


図 3.2-1 インタフェース名の例

(2) インタフェース名の重複と同音異義・異音同義語の禁止事項

インタフェースには以下の2種類があり、インタフェース名の重複や同音異義・異音同義語の禁止の範囲が異なる。命名ルールを定める対象毎に、重複や同音異義・異音同義語の禁止の範囲を以降に示す。

- 公開するインタフェース(他サブシステムからも利用するサービスインタフェース)
- 公開しないインタフェース(サブシステム内のみで利用するサービスインタフェース)

他サブシステムへの公開及び、重複や同音異義・異音同義語の排除のため、各サブシステムが公開するインタフェース名の管理については、「3.2.8.1 用語のルール」を参照のこと。

A. 重複

- サブシステム内では、重複を禁止する。 …(a)
- サブシステム間では、重複を許容する。 …(b)

ただし、インタフェース名とともに、サブシステム名とサービス名を分類として明記することで異なるインタフェースであることが分かるようにすること。

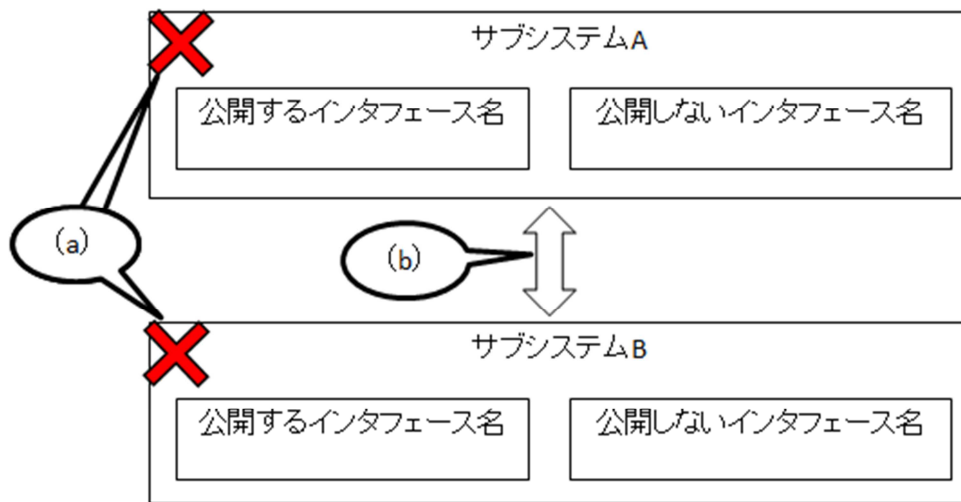


図 3.2-2 インタフェース名の重複禁止の範囲

B. 同音異義・異音同義語

- サブシステム内で、同音異義・異音同義のインタフェース名を禁止する。 …(a)
- サブシステム間では、以下のとおりとする。 …(b)
 - 公開するインタフェースは、他サブシステムが公開するインタフェースも含め、同音異義・異音同義のインタフェース名を禁止する。 …(b1)
 - 公開しないインタフェースは、同音異義・異音同義のインタフェース名を許容する。 …(b2)

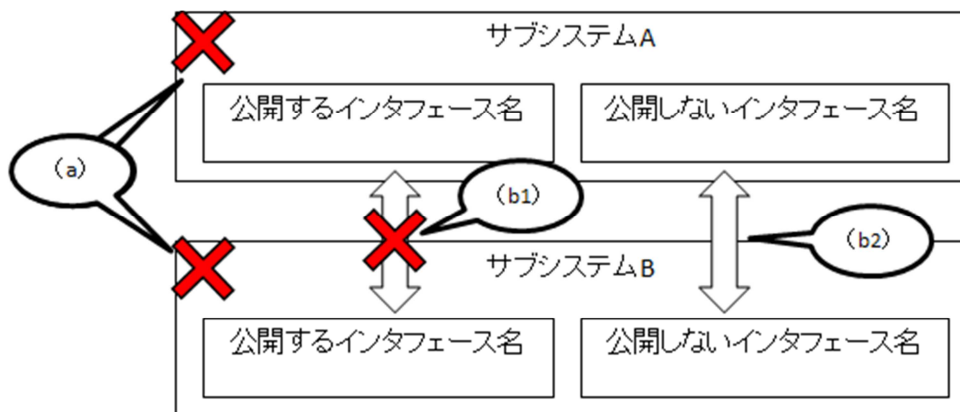


図 3.2-3 インタフェース名の同音異義・異音同義語禁止の範囲

3.2.1.3.2 入出力項目名の命名ルール

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.2

指針1： 入出力項目名は、「(1)入出力項目名の命名ルール」に従って命名すること。

(1) 入出力項目名の命名ルール

入出力項目名の命名ルールを以下に示す。

- データベースのカラム名を利用すること。
- カラムに存在しない名称が必要な場合は、カラム名の命名ルール(「3.2.8.2.2 カラム名の命名ルール」を参照のこと。)に従って名称を定義すること。
- 以下の場合、名称を加工してインタフェース仕様が理解しやすいようにすること。

表 3.2-5 入出力項目名の加工

項番	ケース	付加する用語	付加方法
1	リスト型やマップ型等を入出力項目として指定する場合	データ型を表す用語	名称の最後に「一覧」等のデータ型を表す用語を付与する。
2	入出力項目をグループ化したい場合	グループを表す用語	名称の最初にグループを表す用語を付与する。

3.2.2 UI層及びプレゼンテーション層のルール

(1) UI層

UI層のシステム利用者、利用環境、責務及びシステム構成要素を以下に示す。

A. システム利用者

システム利用者は庁内の職員である。

B. 利用環境

一般的なPC用ディスプレイを持ち、マウス及びキーボードにより画面操作を行う業務用PCである。

C. 責務

UI層の責務は、「ユーザに対して入出力を伴う直接的なインタフェース(GUI)」を提供することである。

UI層の責務を担うために行う主な処理内容を以下に示す。

- システム利用者が入力したデータ内容や操作したイベントの有効性を確認する。また、有効でない場合は画面表示により、システム利用者へ通知する。
- システム利用者の入力したデータ又は操作のイベントに基づき、画面を構成又は電文を組み立て、自サブシステムのプレゼンテーション層へリクエストする。

サブシステムをまたがった情報を1画面に表示する場合においても、自サブシステムのプレゼンテーション層へリクエストする(自サブシステムのプレゼンテーション層が他サブシステムから情報を取得し、1つの画面に編集した情報を返却する。Webブラウザ上よりクロスドメイン通信によって、他サブシステムのプレゼンテーション層から情報取得することを行わないものとする。)

- プレゼンテーション層からのレスポンスに基づき画面を構成する。

D. システム構成要素

UI層は、以下のシステム構成要素を持つ。画面要件に基づきいずれかのシステム構成要素が採用される。

- ブラウザ

ブラウザの責務は、「Webアプリケーションにおいて、ユーザが操作するための画面機能の提供」である。

- リッチクライアント

リッチクライアントの責務は、「Webサービスインタフェースによるリッチクライアントにおいて、ユーザが操作するための画面機能の提供」である。

(2) プレゼンテーション層

プレゼンテーション層の責務及びシステム構成要素を以下に示す。

A. 責務

プレゼンテーション層の責務は、「UI層とワークフロー層・業務アプリケーション層との連携」である。

プレゼンテーション層の責務を担うために行う主な処理内容を以下に示す。

- UI層からのリクエストを受け付ける。
- UI層からのリクエストデータの内容について有効性を確認する。また、有効でない場合はUI層へエラー情報を返却する。
- UI層からのリクエストを処理するために、必要な処理をワークフロー層(自又は他サブシステム)や業務アプリケーション層(自又は他サブシステム)にリクエストする。
- UI層への返却するためのデータを編集し、送信する。

B. システム構成要素

プレゼンテーション層は以下のシステム構成要素で構成される。

- プレゼンテーションロジック

プレゼンテーションロジックの責務は、「UI層からの要求を下位層に振り分け、その実行結果を編集し、UI層へ返却する処理を実行」することである。

3.2.2.1 クライアント構成

スコープ： 階層定型化サブシステム

プロセス： 2.4.2

規約1： クライアント構成は、「(2)クライアント構成の選定ルール」のとおり選定すること。

(1) クライアント構成の分類

ToBeアーキテクチャにおけるクライアント構成を下表に示す。

表 3.2-6 クライアント構成

項番	クライアント構成	説明
1	ブラウザを用いた構成	システム構成要素の「ブラウザ」で動作するクライアントAPを用いた構成。ブラウザからサーバに対しての処理要求の結果として、クライアントAPとなる画面及び処理(HTMLやJavaScript等)がWeb/APサーバから都度返却される。
2	リッチクライアントを用いた構成	システム構成要素の「リッチクライアント」で動作するクライアントAPを用いた構成。ブラウザを介さず、業務用PCのOS上で動作するクライアントAP。ブラウザで扱うことができないソフトウェアの対応が可能。クライアントAPは、事前に業務用PCにアプリケーションが配備される。

クライアント構成のイメージを下図に示す。

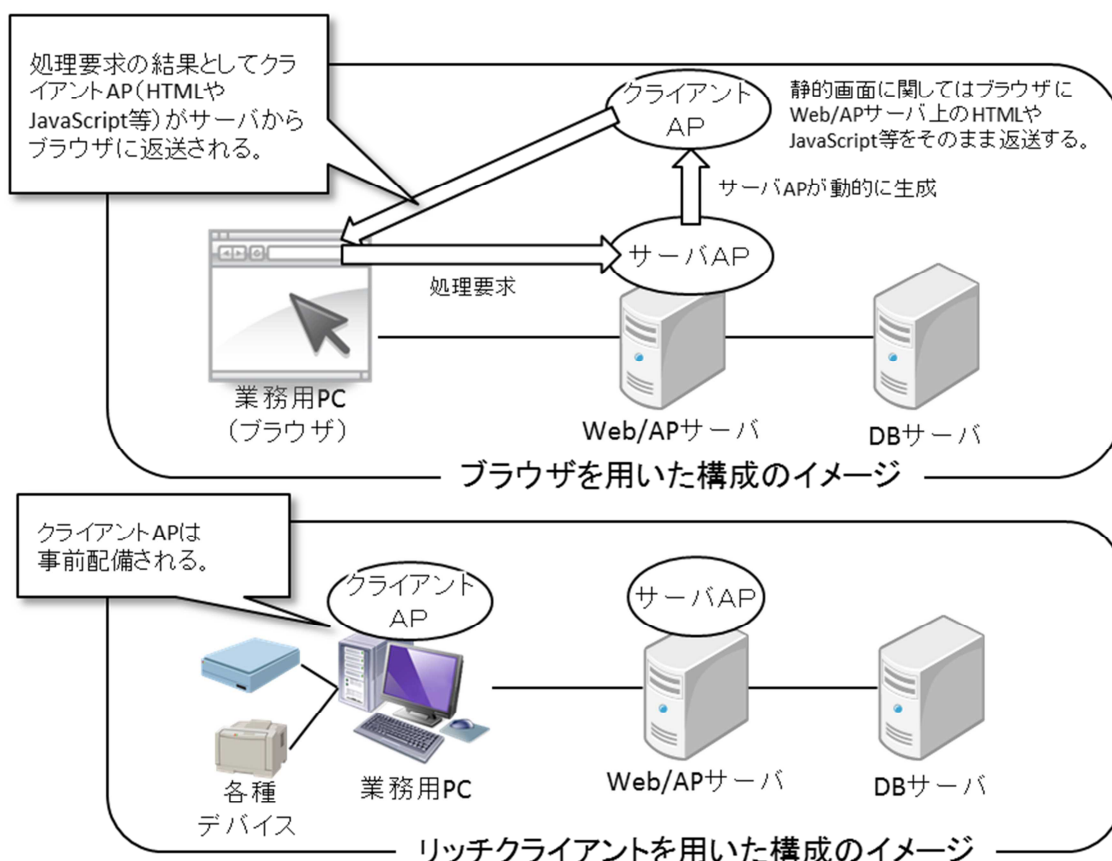


図 3.2-4 クライアント構成のイメージ

(2) クライアント構成の選定ルール

保守性を高めるという設計方針を考慮すると、「ブラウザを用いた構成」には「リッチクライアントを用いた構成」と比べ、以下のメリットがある。

- 幅広い環境で動作し、OSに追加で何らかのプログラムプロダクトをインストールする必要がない。
- アプリケーションは都度サーバから取得し、クライアント側への事前配備を必要としない。

しかしながら、ブラウザで扱うことのできないソフトウェアを利用する場合等、「リッチクライアントを用いた構成」でないと、実現できない場合もある。

以下にクライアント構成の選定ルールを示す。

- 以下に示す採用条件に該当する場合に限り、「リッチクライアントを用いた構成」を採用すること。
「リッチクライアントを用いた構成」の採用条件：
 - 非常に高い入力効率が求められる場合
 - 高度な表示が求められる場合
 - OCRのソフトウェア等、ブラウザで扱うことのできないソフトウェアを利用する場合
 - 「ブラウザを用いた構成」では、特定のブラウザに依存した機能を利用しないと実現できない場合
- 上記以外は、「ブラウザを用いた構成」を採用すること。

3.2.2.2 画面とリクエスト²⁰の作成単位に関するルール

スコープ： 階層定型化サブシステム

プロセス： 2.4.2

指針1： ブラウザからの要求を受けるリクエストとその結果として表示される画面は、BPMSのユーザタスク毎にリクエストと画面のまとまりを割り当てるように作成単位を定めて設計すること。

特例1： 複数のユーザタスクを同時に進めるような業務²¹については、これらのユーザタスクに対し、同一のリクエストと画面のまとまりを割り当てても良い。

(1) 指針に関する補足

UI層は、極めて環境変化に影響しやすく、頻繁に改造が必要となる。したがって、環境変化に対して影響するリクエストと画面を特定できるように、UI層の業務の切れ目となるBPMSのユーザタスク単位にリクエストと画面のまとまりを割り当て、リクエスト毎に処理（プレゼンテーションロジックと業務アプリケーション（ユーザ））を作成する。これにより、影響するリクエストと画面を特定しやすくするだけでなく、あるユーザタスクに対応するリクエストと画面に発生した環境変化による影響を、他のユーザタスクのリクエストと画面に波及させないようにする。

なお、上記単位内の画面遷移においては、静的HTMLを全画面において事前に準備しておいたり、クライアントサイドスクリプティングにより動的に変更させたり、様々な実装方法が採用し得るが、ルール化することによるメリットが特にないため設計の裁量とする。

また、リッチクライアントについては、前述のとおり特殊な業務要件に限定して使用されることから、リクエストと画面の作成単位が業務要件に強く依存する。したがって、リッチクライアントのリクエストと画面の作成単位についてはルール化せず設計の裁量とする。

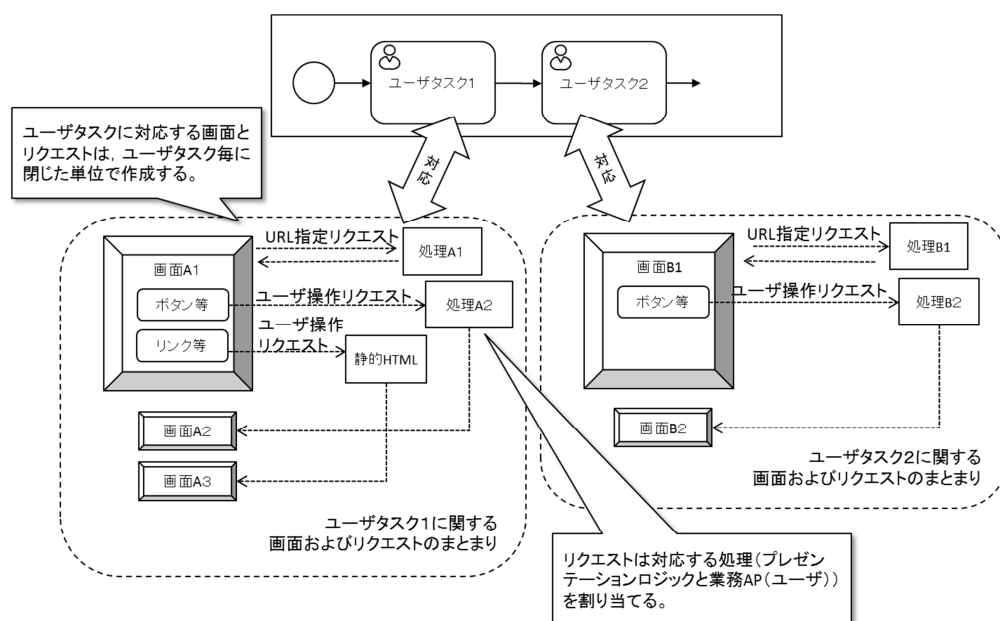


図 3.2-5 ユーザタスクに対するリクエスト及び画面の作成単位

(2) 特例に関する補足

サブシステム分割の結果によっては、例えば、国際調査及び国際予備審査といった異なる業務において生成される起案書類をまとめて起案及び決裁するUIを用意する必要がある。このような、複数のユーザタスクとして分離すべき業務を同時に扱う必要がある場合においては、特例としてこれらユーザタスクの群単位にリクエストと画面のまとまりを割り当てることを許容する。

なお、基本的には指針に従い、ユーザタスクとリクエスト及び画面のまとまりとの対応関係がとれるようにすることが望ましい。したがって、特例を適用する前に、サブシステム分割、機能等の見直しを検討すること。

²⁰ ここでいうリクエストとは、ブラウザからの要求を受けて応答を返すWebサーバのHTTPプロトコルインタフェースであって、プレゼンテーションロジックにおいてルールとして許容されているインタフェース（Ajaxによるアクセス、動的HTMLをロードするためのAPI、単なる静的HTMLのロード等）とする。

²¹ 国際調査と国際予備審査のセット起案等。

3.2.2.3 ブラウザからの要求を受けるアプリケーションの粒度

スコープ： 階層定型化サブシステム

プロセス： 2.4.3

指針1: ブラウザからの要求を受けるアプリケーション(プレゼンテーションロジック及び業務AP(システム)のまとまり)の粒度は、「(1)ブラウザからの要求を受けるアプリケーションの粒度のルール」に従って設計すること。

ブラウザからの要求を受けるアプリケーションはプレゼンテーションロジック及び業務アプリケーション(ユーザ)から構成される。これらの粒度のルールを以下に示す。

ブラウザからの要求を受けるアプリケーションは、リクエストに対する処理(プレゼンテーションロジックと業務アプリケーション(ユーザ))と画面のまとまりを、更に業務のまとまりで束ねたものである。画面とリクエストの作成単位については、「3.2.2.2 画面とリクエストの作成単位に関するルール」を参照のこと。

(1) ブラウザからの要求を受けるアプリケーションの粒度のルール

ブラウザからの要求を受けるアプリケーションは、以下の粒度で構成すること。

- 複数のサブシステムをまたがらないように構成すること。
- サブシステム内においても、「A.分割の観点」に示す観点でアプリケーションを分割すること。

A. 分割の観点

① 業務体系や管理単位で適切に分割すること。

大きなアプリケーションの管理は人間にとって煩雑なものになる。一方、細かいアプリケーションに分割するとアプリケーションの数が多くなり、同様に管理が煩雑になる。業務の責務の切れ目や管理単位でアプリケーションを分割し、アプリケーションの配置が人間にとって、直感的に把握しやすく、管理しやすい大きさにする。

② バージョンアップによるアプリケーションの起動・停止等、運用的な観点で適切な単位で分割すること。

サービス提供時間帯が異なる等、運用上独立しているアプリケーションは資材及びデプロイ先を分け独立させることで、システム運用における柔軟性が向上する場合、分割を行う。

③ 求められる信頼性や性能の度合いが大きく異なる機能が含まれる場合は、アプリケーションを分割すること。

アプリケーションが要求するシステムリソース(CPU、メモリ量、DBコネクションプール数等)を別途与える必要があるアプリケーション群に対しては、アプリケーションを分割しておくこと。物理構成においても、アプリケーションに合わせて分割することで、独立して運用が可能となる。

(2) アプリケーションの粒度のルールを考慮したアプリケーション分割例

サブシステム内でアプリケーションを分割する例としては、実体審査サブシステムにおける実体審査と検索外注業務がある。両者は、業務の責務が分かれており、開発単位や保守・運用における扱いも分けられることから、アプリケーションは分割することが可能である。アプリケーションを分割することで、実体審査と検索外注業務のアプリケーション起動・停止等の運用やバージョンアップ等の保守を独立して管理することができるといったメリットがある。

3.2.2.4 画面設計ルール

画面設計ルールの方針を以下に示す。

(1) 基本方針

画面設計ルールの基本方針を以下に示す。

(方針1) システム利用者に対し、統一した操作感を提供できるルールとする。

異動やシステム更改等で、利用するシステムが変更になった際にシステム利用者が混乱しないよう配慮する。

(方針2) UIに関する部分の汎用化を促進できるルールとする。

画面部品等の仕様を定めることによって汎用化を促進し、個別に作りこむ範囲を業務固有の機能の実現に注力できるようにする。

(方針3) 画面の実現方法に極力依存しないルールとする。

システム更改により画面実現方法(ブラウザを用いた構成／リッチクライアントを用いた構成)が変わったとしても、画面に対する影響を最小限に抑える。

(方針4) 全てのシステム利用者が利用できるよう配慮したルールとする。

システム利用者の利用環境(デバイス層)や利用者属性(色覚障がい等)を踏まえ、一部のシステム利用者が利用できない画面や、誤操作を招くような画面になることを防ぐ。

(2) 適用範囲

画面設計ルールの適用範囲を以下に示す。本適用範囲に当てはまらない場合は、サブシステム毎に別途、ルールを検討するものとする。

- リッチクライアントでのみ実現可能な要件²²が関わる画面は、適用範囲外とする。

ブラウザを用いた構成／リッチクライアントを用いた構成のどちらでも実現可能な画面に関するルールを定める。

²² 非常に高い入力効率や高度な表示が求められる場合等。「3.2.2.1 (2)クライアント構成の選定ルール」を参照。

(3) ルールの全体構成

画面設定ルールの全体構成を下表, 下図に示す。

表 3.2-7 画面設計ルールの構成

項番	ルールの項目	説明
1	アクセシビリティに関するルール	システム利用者の身体条件に伴う対応方針
2	画面定義に関するルール	画面の単位
3		画面間の遷移
4		画面遷移に伴うウィンドウの扱い
5	画面の内容に関するルール (静的設計)	画面の領域
6		画面のレイアウト (領域の組み合わせ)
7		画面部品の扱い
8	画面の動作に関するルール (動的設計)	キー操作等システム利用者に対する動作に関するルール

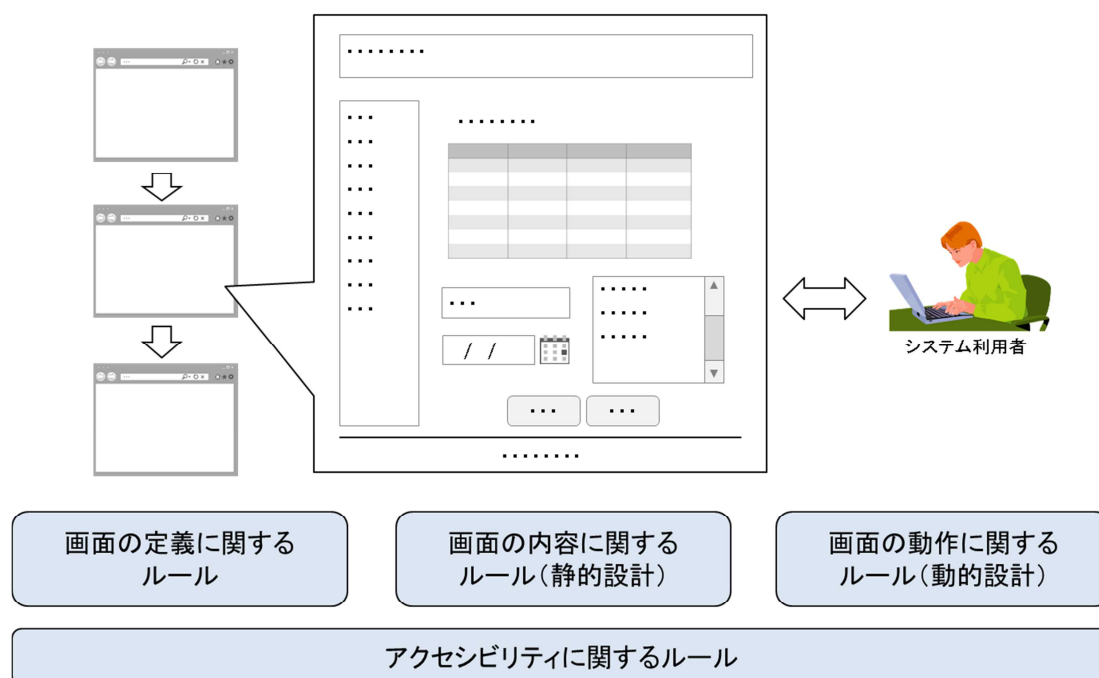


図 3.2-6 画面設計ルールの全体構成のイメージ

3.2.2.4.1 画面のアクセシビリティに関するルール

スコープ：階層定型化サブシステム

プロセス：2.4.2

指針1：画面のアクセシビリティは、「表 3.2-8 アクセシビリティに関する対応方針」に従って設計すること。

システム利用者のアクセシビリティに関する対応方針については、以下のとおりとする。

表 3.2-8 アクセシビリティに関する対応方針

項番	身体条件	アクセシビリティに関する本書における対応方針
1	全盲	対応しない(音声読み上げは対応しない)。
2	聴覚障がい	音だけに依存した設計をしないこととする。
3	知的障がい	対応しない。
4	弱視, 老眼	文字の大きさを変更可能にすることで対応する。
5	色覚障がい	色の組み合わせに配慮する。 ²³
6	肢体不自由者	対応しない(マウスとキーボードの操作を対象とする)。

²³ ツールによるチェックが可能。

ツールの紹介 - NPO法人カラーユニバーサルデザイン機構
<http://www.cudo.jp/colorud/simulation/tools>

3.2.2.4.2 画面の定義に関するルール

スコープ：階層定型化サブシステム

プロセス：2.4.2

指針1：画面の単位は、「(1)画面の単位」に従って設計すること。

指針2：画面間の遷移は、「(2)画面間の遷移」に従って設計すること。

推奨1：「表 3.2-9 代表的な画面遷移のパターン」に示す画面遷移のパターンを推奨する。

指針3：ウィンドウの制御は、「(3)画面遷移に伴うウィンドウの扱い」に従って設計すること。

(1) 画面の単位

業務的な画面の単位を決定する際のルールを以下に示す。

なお、画面を制御するためのビジネスロジック等のコンポーネントを共通化することを制限するものではない。

- BPMNのアクティビティ(ユーザタスク)毎に分けること。

ToBeアーキテクチャでは、BPMSによってアクティビティの実行順序を制御する。ビジネスプロセスの変更によりアクティビティの順序に変更があった場合でも画面構成等への影響を受けにくくするため、画面がアクティビティ間の順序性を持たないよう、アクティビティをまたがる単位では、画面を分ける。

例：起案、決裁

ただし、ダイアログのようにアクティビティが異なっても共通的に利用される画面で、画面構成等の将来的な変更内容がアクティビティ毎に異なることが想定される場合は、同じ画面として定義してもよい。

- 機能毎に分けること。

1つの画面に複数の機能が組み込まれると、画面の目的が曖昧になり操作性が低下するため、参照や更新等の機能毎に分ける。

例：案件照会、書類照会、起案

- 画面遷移パターンに沿って分けること。

機能毎に分割した画面が、「(2)画面間の遷移」で述べる遷移パターンに該当する場合は、そのパターンに従って画面を分ける。

例：入力、確認、完了

- 画面の項目数に応じて分けること。

画面に含まれる項目数が多く画面が縦長になってしまうことで、作業効率の低下等が想定される場合は、画面の項目をいくつかのグループに整理した上で、画面を分ける。

(2) 画面間の遷移

画面間の遷移に関するルールを以下に示す。

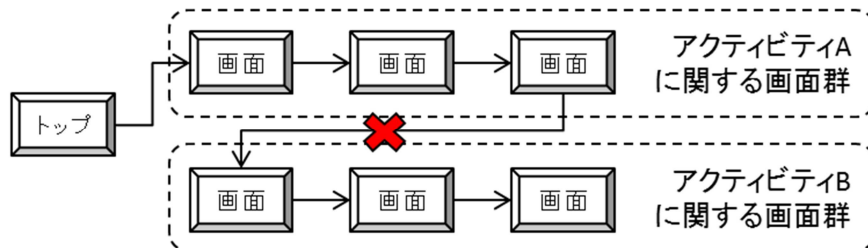
- 他のアクティビティに関する画面群に直接遷移しないこと。

アクティビティの実行順序はBPMSが制御するため、画面側はその順序が変更になっても影響を受けないようにするため、他のアクティビティに関する画面群に直接遷移せず、トップ画面から遷移するようにする。

画面間の遷移の例を下図に示す。

＜悪い例＞アクティビティAが終わるとアクティビティBの画面に遷移する。

※AとBの実行順が変わると、遷移の修正が必要



＜良い例＞アクティビティ間をまたがる遷移はせず、必ずトップへ戻る。

※AとBの実行順が変わっても、遷移の修正は不要

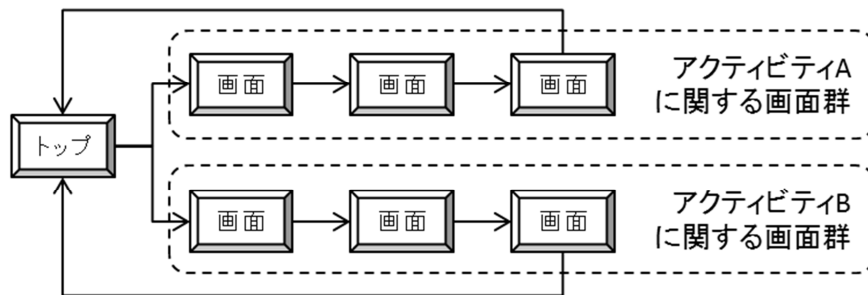


図 3.2-7 画面間の遷移の例

あるアクティビティに対応するヒューマンタスクの状態を先に進めることを例とすると、以下のような制御となる。

- ① 画面群における最初の画面で、あるアクティビティに対応するヒューマンタスクの状態を取得することで、対象の一覧を表示する。
- ② システム利用者が先に進める対象を一覧から選択することで画面遷移し、必要な情報の入力等を行う。
- ③ 画面群の最後の画面のボタンアクション等により、当該アクティビティを完了とし、次に進める。

なお、「アクティビティに関する画面群内」の画面遷移は、別画面へ遷移する方式と画面内の一部分を動的に更新する方式がある。

- 画面の操作感を統一するため、原則、遷移パターンを統一すること。
ただし、特に高い効率性を求める等、サブシステム特有の要件がある場合は、特許庁と協議の上、パターンに含まれる画面の一部を省略することを許容する。
例：登録・更新系遷移における確認画面の省略、等

代表的な画面遷移のパターンを下表に示す。

表 3.2-9 代表的な画面遷移のパターン

項番	遷移パターン	画面遷移	イメージ
1	登録・更新系遷移	以下の3つの画面で遷移する。 ① 情報を入力する画面 ② 入力した内容を確認する画面 ③ 処理の結果を通知する画面	<pre> graph LR A[入力] --> B[確認] B --> C[結果] </pre>
2	検索系遷移 一括表示パターン ※検索条件の項目数が少ない場合	検索条件の入力と、検索結果の表示を1つの画面で表示する。 検索／再検索は、同一画面に遷移する。	
3	検索系遷移 分割表示パターン ※検索条件の項目数が多い場合	検索条件の入力と、検索結果の表示を別画面に分割して表示する。 検索／再検索は、別画面に遷移する。 検索結果表示時は、合わせて検索条件の入力値を表示する。	

(3) 画面遷移に伴うウィンドウの扱い

画面遷移は、同一ウィンドウ内での遷移とする。

ただし、下表に示す場合は、別ウィンドウ等への遷移を行う。

表 3.2-10 別ウィンドウ等への遷移を行うケース

項番	ケース	遷移方法	例	補足説明
1	遷移先画面が補助的役割であり、必ず遷移元画面に戻る場合	画面内ポップアップを利用する。	入力された郵便番号から住所を選択する画面、等	ポップアップは、1つのウィンドウ内に重ねて表示する擬似的なウィンドウである。遷移先画面は遷移元画面よりも小さい任意のサイズで表示し、ポップアップ表示中は、遷移元画面への操作をブロックする(モーダル表示)。 
2	遷移先画面を開いた後も、遷移元画面で作業を継続したい場合	新規ウィンドウを開く。	別システムを開く場合、ヘルプ画面を表示する場合、等	—

3.2.2.4.3 画面の内容に関するルール

スコープ：階層定型化サブシステム

プロセス：2.4.2

指針1： 画面レイアウトは、極力統一したレイアウトになるように設計すること。

推奨1： 画面レイアウトは、「(2)画面のレイアウト」に示すレイアウトを推奨する。

指針2： 画面部品は、「(3)画面部品の扱い」に従って設計すること。

(1) 画面の領域

ToBeアーキテクチャにおいて定義する画面の領域を下表に示す。

表 3.2-11 画面の領域

項番	領域名	定義	領域に含む内容(例)	補足説明
1	共通情報	サブシステム内に共通の情報や共通的な機能を格納する領域	サブシステム名、ログイン者情報、ヘルプ表示、ログアウト、等	サブシステム名(ロゴ含む)を表示する場合、トップ画面へのリンクを兼ねること。
2	個別情報	各画面に関する情報を格納する領域	Copyright表記、連絡先、等	—
3	メイン	画面の主目的となる内容を格納する領域	入力フォーム、参照情報、検索結果、等	—
4	ナビゲーション	他のページへのリンクを格納する領域	他ページへのリンク、等	—
5	アクション	画面の最終的なアクションを決定する手段(ボタン等)を格納する領域	登録ボタン、キャンセルボタン、等	主に登録・更新系画面で使用する。ボタンの並びについては、「(3)画面部品の扱い」を参照。

(2) 画面のレイアウト

- 原則、全ての画面は下図に示すいずれかのレイアウトとすること。

ただし、一般的なPC用ディスプレイとは画面サイズに関する要件が大きく異なる場合は、個別にレイアウトを定めることを許容する。

表示する内容がない場合は、その領域の省略を許容する。

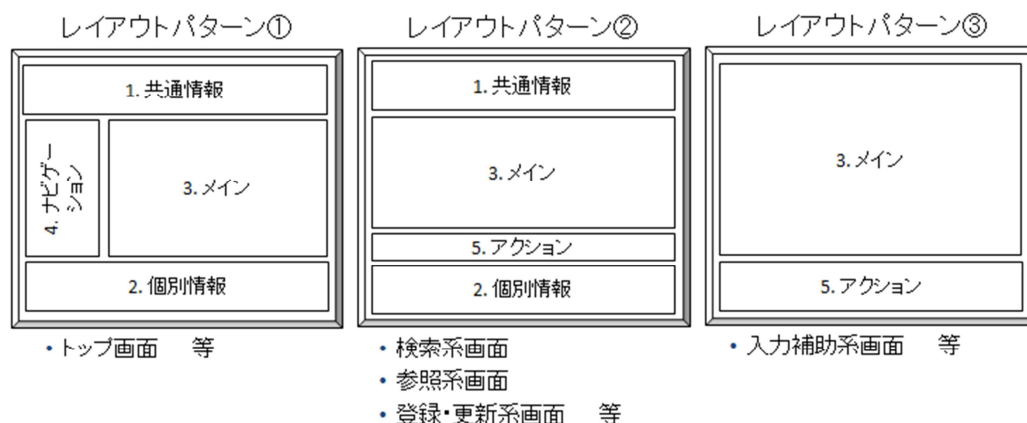


図 3.2-8 画面レイアウトパターン

- 原則、以下のフォーマットに従いウィンドウタイトルを付与すること。

ただし、補足情報を付与する等、フォーマットを変更することを許容する。

「画面名 (画面ID) | サブシステム名」

(3) 画面部品の扱い

ToBeアーキテクチャにおいて定義する画面部品を下表に示す。なお、画面部品を、「基本部品」と、基本部品を組み合わせる動作を拡張した「応用部品」に分類する。

画面を設計する際は、この利用指針に沿って画面部品を選定し、表示や配置に関するルールに従うものとする。

なお、「基本部品」を「応用部品」の一部として使用する場合は「応用部品」のルールに準ずる。

表 3.2-12 基本部品

項番	部品	利用指針	表示や配置に関するルール
1	ブレンテキスト サンプル	情報表示や、説明文の記載に利用する。	文字色は背景色とのコントラストを意識し、利用者によって色の判別が困難としないようにすること。 フォントの種類及び文字の大きさは極力指定せず、OSや画面の実現方式(ブラウザ等)で設定された標準値を利用すること。
2	リンクテキスト サンプル	ユーザが入力した情報の送信を伴わない、ページを移動するだけの画面遷移の契機として利用する。	リンクテキストが画面の実現方式(ブラウザ等)で提供されている場合は、その標準の色や文字装飾(アンダーライン等)を変更しないこと。
3	ボタン サンプル	ユーザが入力した情報の送信を伴う場合や、データ更新等のアクションを伴う画面遷移の契機として利用する。	- ボタンに記載するラベルは「特性名＋動作名」とし、送り仮名は省略すること。 (特性名は、「目的や特性を表す用語」。 動作名は、「ボタン押下後の動作を表す用語」。 悪い例：「案件を登録する」 良い例：「案件登録」 ただし、画面の構成上、特性名が明らかな場合は、特性名を省略してよい。 例：「入力キャンセル」→「キャンセル」 代表的な動作名について、以下に示す。 「登録」、「更新」、「削除」、「検索」、 「完了」、「キャンセル」、「進む」、「戻る」 また、画像によるボタンの代用は許容する。ただし、押したことが分かる等、ボタンと同様の表現を行うこと。 - 複数のボタンを横に並べる場合、登録ボタン等、次の画面に遷移するためのボタンは左側に、キャンセルボタン等、前の画面に遷移するためのボタンは右側に配置すること。 - 誤操作を防ぐため、ボタン自体及びボタン間に十分な幅を取ること。
4	テキストボックス サンプル	1行テキストを入力する場合に利用する。	- 最大入力可能文字数に合わせた幅を確保し、文字が見切れないようにすること。 - パスワードを入力する場合は、「●」に変換する等のマスク処理を施すこと。

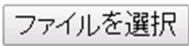
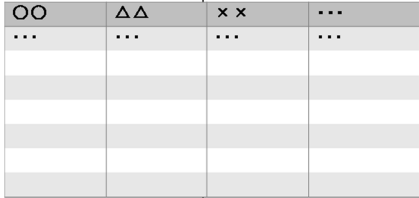
項番	部品	利用指針	表示や配置に関するルール
5	テキストエリア 	複数行テキストを入力する場合に利用する。	● 極力、最大入力可能文字数に合わせた幅及び高さを確保し、文字が見切れないようにすること。ただし、最大入力可能文字数が多い場合、他の画面部品とのバランスを考慮し、高さを小さくし縦スクロールバーを表示すること許容する。 ● 幅を超える文字数を入力した場合は、改行して表示すること(自動で改行コードは付与しない)。また、改行コードが入力された場合は、改行して表示する。
6	ラジオボタン 	選択肢の中から1つを選択する場合、かつ、選択肢が8個程度までの場合に利用する。	● 初期値がないような選択肢の場合でも、選択なしを表す選択肢を用意し初期選択とすること。 ● 選択肢の説明はラジオボタンの右側に記載し、混同しないよう各ラジオボタン間には十分な余白を設けること。
7	ドロップダウンリスト 	選択肢の中から1つを選択する場合で、かつ、上記ラジオボタンに当てはまらない場合に利用する。	● 極力、最大表示文字数に合わせた幅を確保し、文字が見切れないようにすること。 ● 1行表示のみ使用すること。 ※複数行表示は使用しない。
8	チェックボックス 	選択肢の中から複数を選択する場合に利用する。	● 選択肢の説明はチェックボックスの右側に記載し、混同しないようチェックボックス間には十分な余白を設けること。
9	ファイル選択 	ローカルファイルを選択する場合に利用する。	● ファイル選択ダイアログを表示すること。 ● ファイル選択ダイアログに表示するファイル一覧は、極力、ファイル名やファイル形式等でフィルタリングし、ファイルの誤選択を防止すること。

表 3.2-13 応用部品

項番	部品	利用指針	表示や配置に関するルール
1	カレンダ 	日付を選択する場合に利用する。	● テキストボックス及びボタンと合わせて使用し、ボタンを押下するとカレンダーを表示すること。 ● カレンダー上で日付が選択されると、テキストボックスに選択した日付を自動的に挿入すること。 ● テキストボックスに直接日付を入力することも可能とすること。
2	書式付きエディタ 	起案エディタのように、単純なテキストでなく書式付きテキストを入力する際に利用する。	● 書式設定に関するメニューは、書式付きエディタの上部に配置すること。 ● 極力、編集操作のやり直しや、元に戻す機能を提供すること。 ● 幅及び高さ、スクロールバーの表示、改行等については、極力、テキストエリアと同様とすること。
3	グリッド 	表形式で情報を表示したい場合に利用する。	● 1行目に見出し行を設けること。 ● 極力、横スクロールが発生しないよう、表示幅を調整すること。 ● ソートが要件に含まれる場合は、見出し行にソートの切り替え制御を設けること。 ● 一度に表示する件数は、性能要件等を踏まえ妥当な件数に制限すること。 表示したい件数とその件数より多い場合は、以下の2通りの方法から選択すること。 ➢ ボタン押下等のユーザ操作に伴い続きを読み込み、グリッドに追記する。 ➢ ページ送りを設け、指定された範囲のグリッドを再表示する。 ● 列の幅や表示位置の変更をした場合、次回表示時に同じ設定で表示されるようにすること。
4	入力フォーム 	複数の情報を入力する場合に利用する。	● 複数の情報をグルーピングし、タイトルを付与すること。 ● 項目名は、テキストボックス等の入力部品の左側に記載すること。 ● 必須入力項目は、項目名に続けて「(必須)」と記載すること。 ※赤字を用いる等、強調して表示すること。 ● 項目に関する説明や入力例、入力フォーマットを表示したい場合は、項目名に続けてツールチップ表示用アイコンを配置すること。

3.2.2.4.4 画面の動作に関するルール

スコープ：階層定型化サブシステム

プロセス：2.4.3

指針1：画面動作は、「表 3.2-14 画面の動作に関するルール」に従って設計すること。

画面の動作に関するルールを下表に示す。

表 3.2-14 画面の動作に関するルール

項番	ルール項目	ルール
1	入力チェックに関するルール	● 入力チェックは、入力部品からフォーカスが外れたとき等、できるだけ早いタイミングでチェックを行うこと。 ● 入力エラーが発覚した場合は、以下の対処を行うこと。 ➢ 該当する入力部品の背景色を変更する。 ➢ 該当する入力部品の右側又は下側に、エラーの内容を表示する。 ※エラーの内容は「発生したこと」と「対処方法」を記載すること。 ● 「アクション」表示領域に格納されるボタン押下時にしかチェックできない事項については、上記対処を行うとともに、メッセージダイアログ（メッセージを表示するだけのポップアップ）で入力エラーが発生した旨を通知すること。
2	リンクテキスト制御に関するルール	● リンクをクリックした際、該当のリンクの色を変えること。
3	ボタン制御に関するルール	● データ更新を伴うボタンについては、二度押し防止制御を設けること。 ※一度押したボタンは非活性化する。
4	フォーカス制御に関するルール	● 画面表示時の初期フォーカスを定めること。 ● Tabキーを利用した項目間の移動手段を設けること。その際、効率よい移動順に配慮すること。 ● 郵便番号やクレジットカード番号等、形式上複数のテキストボックスに分割した方が望ましい場合は、入力桁数に伴い自動的に次のテキストボックスへフォーカスを移動させること。 ● 作業効率向上のためフォーカス時、入力モードを切り替えることを許容する。
5	キーボード操作に関するルール	● 作業効率向上のため、ショートカットキーを設けることを許容する。 ただし、画面の実現方式（ブラウザ等）によって、ファンクションキー等に既に機能が割り当てられているものは利用しないこと。
6	マウス操作に関するルール	● 右クリックメニューは極力利用しないこと。 ● クリック可能な要素やドラッグ可能な要素については、カーソルの形状を変更すること。 ● ラジオボタン等の選択項目については、選択項目のラベルをクリックすることで選択可能とすること。
7	スクロールに関するルール	● マウスホイール及びキーボードでのスクロールを可能にすること。 ● 同一画面内の別箇所へジャンプする機能を提供する場合は、瞬時に移動するのではなく、スクロールしたことが分かるようアニメーション処理を入れること。
8	入力規制に関するルール	入力規制は、禁止文字や入力桁数制限等といった、不正となる文字を入力できないようにする制御のことである。例えば、電話番号を入力するテキストボックスに対して数字とハイフンのみを入力できるように規制する。 ● 入力規制は、キーボードのキーイベントを検知し、文字が画面に表示される前に規制対象かの判断すること。 ● 規制対象の文字が入力された場合は、文字を表示しないこと。

3.2.2.5 帳票設計ルール

帳票設計のルールについて、外部設計の観点で、画面設計ルールと同様にユーザインタフェースの一部の扱いとして通常同列で扱うことから、並びで章節を配置するものである。実際には、UI層及びプレゼンテーション層内のシステム構成要素のルールではなく、帳票設計に係る業務アプリケーション等のルールである。UI層及びプレゼンテーション層内のシステム構成要素のルールについては、「3.1.2.4 帳票出力方式」を参照のこと。

ユーザにとって理解しやすく、一貫性のある帳票を設計することを目的として、帳票設計の推奨のルールを以降に示す。

なお、以下の場合には、本ルールの適用対象外とする。

- 特許庁における業務規程や法令等によって帳票レイアウトが定められている場合。

3.2.2.5.1 帳票の出力項目に関するルール

スコープ：階層定型化サブシステム

プロセス：2.4.2

指針1： 帳票の出力項目は、見た目を極力、統一するように設計すること。

推奨1：「表 3.2-15 帳票の設定項目の推奨の設定値」に従って設計することを推奨する。

特例1：「表 3.2-15 帳票の設定項目の推奨の設定値」に示す特例を許容する。

帳票の設定項目の推奨の設定値を下表に示す。

表 3.2-15 帳票の設定項目の推奨の設定値

項番	設定項目		推奨の設定値及び理由	補足説明
1	フォント	種類	明朝体 理由：特許庁が外部向けに公開している文書等で一般的に使用しているフォント種類であるため。	「3.2.10 文字コードの扱いのルール」に規定されている符号化文字集合をサポートするフォントを使用すること。 特例： なお、JIS X 0213:2004 ²⁴ で例示字形（デフォルトグリフ）が従前のものから変更されている。このグリフが業務要件を満たさない場合は特例として以下のいずれかの対応を許容する。 ● 規定に準ずるフォントを使用し、従前のグリフを代替グリフとして指定する（使用するフォント及び帳票設計・印刷ソフトウェアがサポートする場合。）。 ● 従前のフォント及びグリフを使用する。
2		サイズ	通常 :10pt タイトル :14pt 理由：通常のフォントサイズは、一般的に使用されるフォントサイズ(10～12pt)のうち、紙面に多くの情報を掲載可能とするため最小のサイズとした。タイトルのフォントサイズは、通常のフォントサイズと比較して違和感のないサイズとして設定。	—
3	表示属性	文字配置	タイトル :中央 表の列タイトル :中央 固定文字数の文字列:中央 可変文字数の文字列:左寄せ 日付, 時刻 :左寄せ 数値 :右寄せ 理由：一般的に使用される文字配置であるため。	項番、ページは数値だが、左寄せとする。
4		文字修飾	設定しない。 理由：見た目の統一を図るため。	文字修飾には、取り消し線、上付き、下付き等がある。

²⁴ 通称「JIS2004」呼ばれる。日本語用の符号化文字集合を規定する日本工業規格(JIS)である。
JIS X 0213はJIS X 0208を包含し、更に第三・第四水準漢字等を加えた上位集合である。

項番	設定項目		推奨の設定値及び理由	補足説明
5		スタイル	設定しない。 理由：見た目の統一を図るため。	スタイルには、下線, 太字, 斜体等がある。
6	表示様式	表示フォーマット	日付: yyyy.MM.dd 例: 2014.01.02 時刻: hh:mm:ss 例: 01:02:03 数値: #,###,##9 例: 1,234 理由: 一般的に使用される表示フォーマットであるため。	項番, ページは数値だが, 以下のように表示する。 項番: カンマなし 例: 1234 ページ: 「P.」+ZZZ9 (ゼロ埋めする) 例: P.0001
7		半角／全角	英字, 数字: 半角 記号: 右記参照 平仮名, 片仮名, 漢字: 全角 理由: 半角／全角の一般的な使い分けであるため。	● 帳票レイアウト定義に含まれる文字を対象とする。 帳票に出力するデータは, 業務要件がない限り, 原則, 加工せずにそのまま表示する。 ● 記号のうち, 全角／半角両方がある場合は, 特に理由のない限り半角を使用する。 例: 「\$」と「\$」の場合は「\$」

3.2.2.5.2 帳票のレイアウトに関するルール

スコープ：階層定型化サブシステム

プロセス：2.4.2

指針1： 帳票のレイアウトは、極力、統一するように設計すること。

推奨1： 共通的に出力する項目の出力位置は、帳票上部とすることを推奨する。

レイアウトが不統一の場合、読解性が低下する。よって、レイアウトを極力、統一するものとする。
ただし、帳票毎に出力する項目が異なるため、全てを統一することはできない。
よって、共通的に出力する項目及びその出力位置の定義するものとする。

帳票には、業務データ以外に一般的に以下のような項目が共通的に出力される。

- タイトル
- 帳票の補足情報(ページ数, 印刷日時等)
- 表の列タイトル

共通的に出力する項目の出力位置は、極力、帳票上部とする。

例として、A4カット紙横方向でのレイアウトイメージを下図に示す。

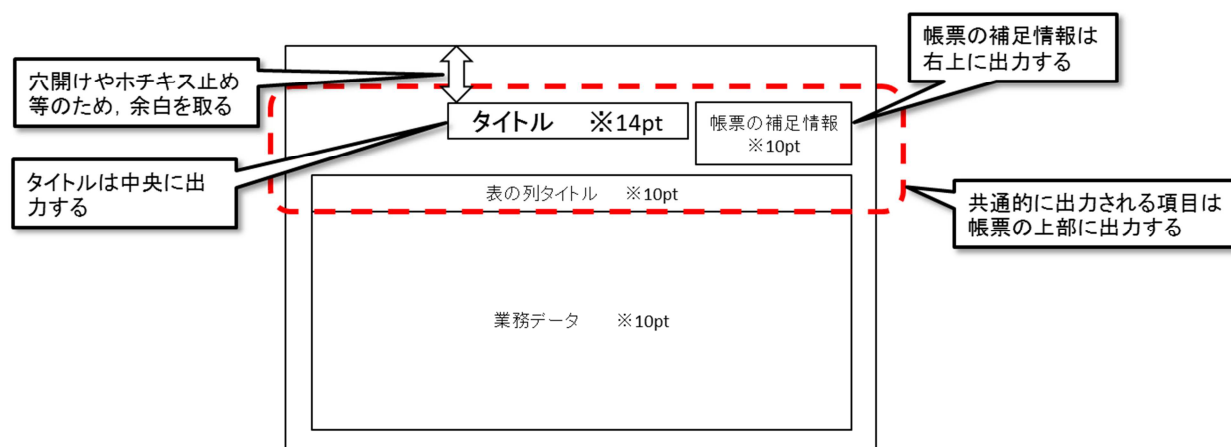


図 3.2-9 帳票レイアウトイメージ

3.2.2.5.3 印刷用紙に関するルール

スコープ：階層定型化サブシステム

プロセス：2.4.2

指針1： 帳票の用紙サイズ, 印刷の向きは, 極力, 統一するように設計すること。

推奨1：「(1)印刷用紙」に示す用紙サイズ, 印刷の向きを推奨する。

(1) 印刷用紙

用紙サイズや印刷方向が混在すると, 読解性の低下や書類の保管等に不便が生じることが考えられる。そのため, 用紙サイズ, 印刷の向きは極力, 統一するものとする。

- 印刷の向きは, 極力, 横方向とする。
- 用紙サイズは, 極力, A4カット紙とする。

ただし, 項目数や出力データの文字数が多く, A4カット紙では入りきらない場合は, A3カット紙も許容する。

3.2.2.5.4 帳票の管理に関するルール

スコープ：階層定型化サブシステム

プロセス：2.4.2

規約1： 帳票をファイルとして管理する必要がある場合のファイル形式は、PDFのファイル形式で出力すること。

帳票をファイルとして管理する必要がある場合、ファイル形式をPDFで統一するものとする。

3.2.3 ワークフロー層のルール

ワークフロー層の責務及びシステム構成要素を以下に示す。

A. 責務

ワークフロー層の責務は、「業務アプリケーションの連携をビジネスプロセスとして管理及び可視化」することである。

ワークフロー層の責務を担うために行う主な処理内容を以下に示す。

- 他層へワークフローを操作するためのサービスインタフェースを提供し、リクエストを受信する。又は、他層に配置されたワークフローAPIからのリクエストを受信する。
- リクエストに従い以下のような処理を行う。
 - ワークフローの生成, 開始処理
 - ワークフローのインスタンスやアクティビティの検索, 状態返却処理
 - ワークフローを構成する要素へのイベント通知
- ワークフローの定義に従ったフロー制御処理, イベント処理, アクティビティ実行を行う。

また、ワークフローの進行状況, 実行履歴の管理を行う。

B. システム構成要素

ワークフロー層は以下のシステム構成要素で構成される。

● BPMS

BPMSの責務は、「ワークフローの制御による業務アプリケーションの連携」である。

BPMSのルールは、以下に示す。

- 「3.2.3.1 ビジネスプロセスのルール」
- 「3.2.3.2 BPMNの記載ルール」

● BPMS補完機能

BPMS補完機能の責務は、「ワークフローの再開箇所の判定等, BPMSにおけるワークフロー制御の補完」である。DBアクセスを行う等, 業務アプリケーション層の責務に該当する処理を行わないこと。

BPMS補完機能のルールは、以下に示す。

- 「3.2.3.3 BPMS補完機能のルール」

3.2.3.1 ビジネスプロセスのルール

3.2.3.1.1 ビジネスプロセス管理として利用する技術

スコープ：階層定型化サブシステム

プロセス：2.4.2

規約1： ビジネスプロセスの実行制御を一元管理する仕組みは、BPMSとすること。

スコープ：階層定型化サブシステム

プロセス：2.3.2

規約1： ビジネスプロセスの表記は、BPMN2.0のプロセス図を使用すること。

ビジネスプロセスの実行制御を一元管理する仕組みとして、BPMSを利用する。²⁵

また、ビジネスプロセスの表記法には、BPMN2.0のプロセス図を使用すること。

²⁵ 業務間の順序関係がない単体で完結する業務はビジネスプロセスとして管理する必要はなく、BPMSの利用を強制するものではない。

3.2.3.1.2 BPMSの適用範囲に関する設計指針

スコープ：階層定型化サブシステム

プロセス：2.3.2, 2.4.3

指針1： BPMSと業務アプリケーションの組み合わせは、「表 3.2-16 BPMSと業務アプリケーションとの組合せの代表的なケース」に示すケースに該当する場合、以下の(1)～(3)に示すBPMSと業務アプリケーションの設計指針に従い、ビジネスプロセスを実現できるよう設計を行うこと。

BPMSと業務アプリケーションとの組合せで一連のビジネスプロセスを実現する上で、保守性の向上のためにBPMSにおけるビジネスプロセスの可視性を確保しなければならないという目的(言い換えると、業務の流れをビジネスプロセスで把握できるようにする目的)から、BPMSと業務アプリケーションを組み合わせるビジネスプロセスを制御する際のBPMS側で行うべき範囲と業務アプリケーションで行うべき範囲の設計指針として、特許庁業務の代表的なケースに基づき以下の表に示す。

表 3.2-16 BPMSと業務アプリケーションとの組合せの代表的なケース

項番	ケース	業務例
1	他のビジネスプロセスとの密接な順序関係を持つビジネスプロセス	予見できないタイミングで発生する異なるビジネスプロセスインスタンスへの割り込み
2		再帰的な構造を持ったビジネスプロセスインスタンス間の実行順序制御
3	アクティビティの割り当てが組織構造と密接な関係を持つビジネスプロセス	アクティビティに対応するヒューマンタスクの動的な絞込み
4		担当者の変更運用
5	BPMNの記述上想定外のイベントが発生するケース	ビジネスプロセスが開始する前に、中間イベントが通知されるケース
6		イベントゲートウェイを利用し、一方のイベントが通知されフローが進行してしまった後に、他方のイベントが通知されるケース
		自発補正
		補正の補正
		実体審査 (予約, 起案, 決裁)
		人事異動時の差戻し
		願書を未受理で、出願審査請求書を受理した場合
		拒理応答の応答期間経過後、意見書を受理した場合

(1) 他のビジネスプロセスとの密接な順序関係を持つビジネスプロセスの設計指針

A. 予見できないタイミングで発生する異なるビジネスプロセスインスタンスへの割り込み

特許庁業務における「自発補正」のように、あるビジネスプロセスインスタンスで発生したイベントにより、既に進行している別のビジネスプロセスインスタンスの処理に割り込み、中断(戻し)したり、待機したりするといった異なるビジネスプロセスインスタンス間での実行順序が必要となるケースがある。このような場合、以下の設計指針に従って、ビジネスプロセスを設計すること。

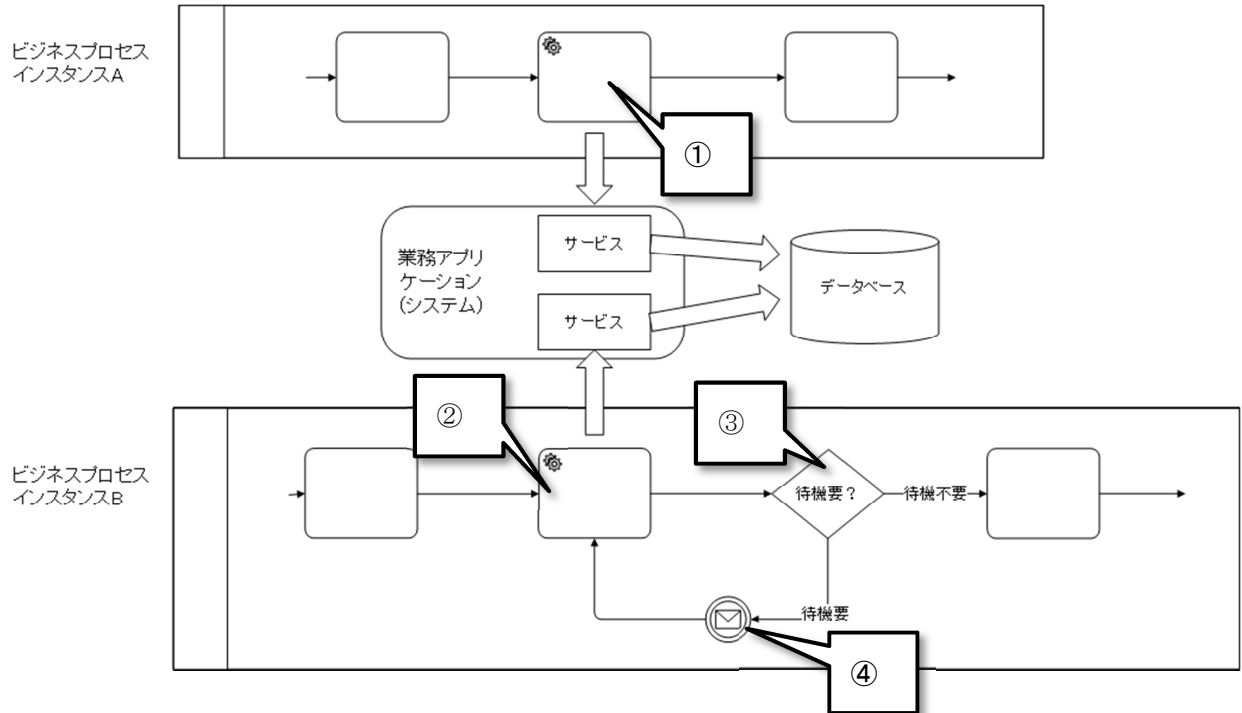
- 全てをBPMNで記載せず、ビジネスプロセスインスタンス間の実行順序制御が必要となる適切なポイントで業務アプリケーション(サービス)を呼び出し、データベースの最新状態から自ビジネスプロセスを待機させるか判断させる。

これは、ビジネスプロセスとしてのアクティビティの順序関係以外に、粒度が細かいシステムの実装上必要な順序制御のフローが混在することで、ビジネスプロセスの可視性・保守性が低下することを防止するためである。

- 判定条件がビジネスルール化の特性に適すると判断できる場合は、ビジネスルールとして業務アプリケーションから切り離す仕組みを利用すること。

ビジネスルールを業務アプリケーションから切り離す仕組みについては、「3.2.7.1.1 ビジネスルールをアプリケーションから切り離すための仕組みとして利用する技術」参照のこと。

例として、待機制御の処理イメージを下图に示す。



- ① ビジネスプロセスインスタンス A が業務アプリケーション(システム)のサービスを介し、データベース上の当該案件情報を更新する。
- ② ビジネスプロセスインスタンスBが業務アプリケーション(システム)のサービスを介し、データベースから当該案件の最新情報から待ち要因があるかを取得し、プロセスデータに反映する。(待ち要因は、他のビジネスプロセスのサービスや業務画面からデータベース上の案件情報の更新が行われことにより更新される)。
待ち要因の判定条件がビジネスルール化の特性に適すると判断できる場合は、ビジネスルールとしてアプリケーションから切り離す仕組みを利用すること。
- ③ ②の結果から排他ゲートウェイで分岐先を判定する。
- ④ 待機要の場合、キャッチ中間メッセージイベントを使い、通知が来るまで(あるいはキャッチ中間タイマーイベントで一定時間)ビジネスプロセスを待機させる。
- ⑤ ②へ戻る。

図 3.2-10 異なるビジネスプロセスインスタンスへの割り込みによる待機制御

なお、排他ゲートウェイと中間イベントを使った待機状態の記載は、ビジネスプロセス上の至るところに発生し、通常シナリオのビジネスプロセスの可視性を低下させてしまう可能性があるため、以下のようにすること。

- 「3.2.3.2.4 ビジネスプロセスパターン事例」の「(1) パターン事例1 他ビジネスプロセス起因の待機処理」のルールに従い、ビジネスプロセスを記載すること。

中断制御の場合は、待機制御とほぼ同様の実現方式となる。「図 3.2-10 異なるビジネスプロセスインスタンスへの割り込みによる待機制御」のプロセスインスタンスBが下図のようになる。

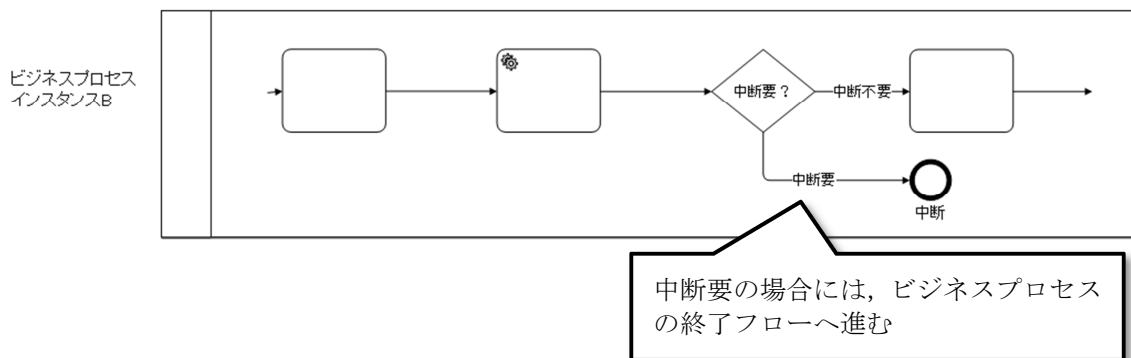


図 3.2-11 異なるビジネスプロセスインスタンスへの割り込みによる中断制御

B. 再帰的な構造を持ったビジネスプロセスインスタンス間の実行順序制御

特許庁業務における「補正の補正」のように、ビジネスプロセスインスタンスが再帰的な構造を持つケースがある。

このような場合、以下の設計指針に従って、ビジネスプロセスを設計すること。

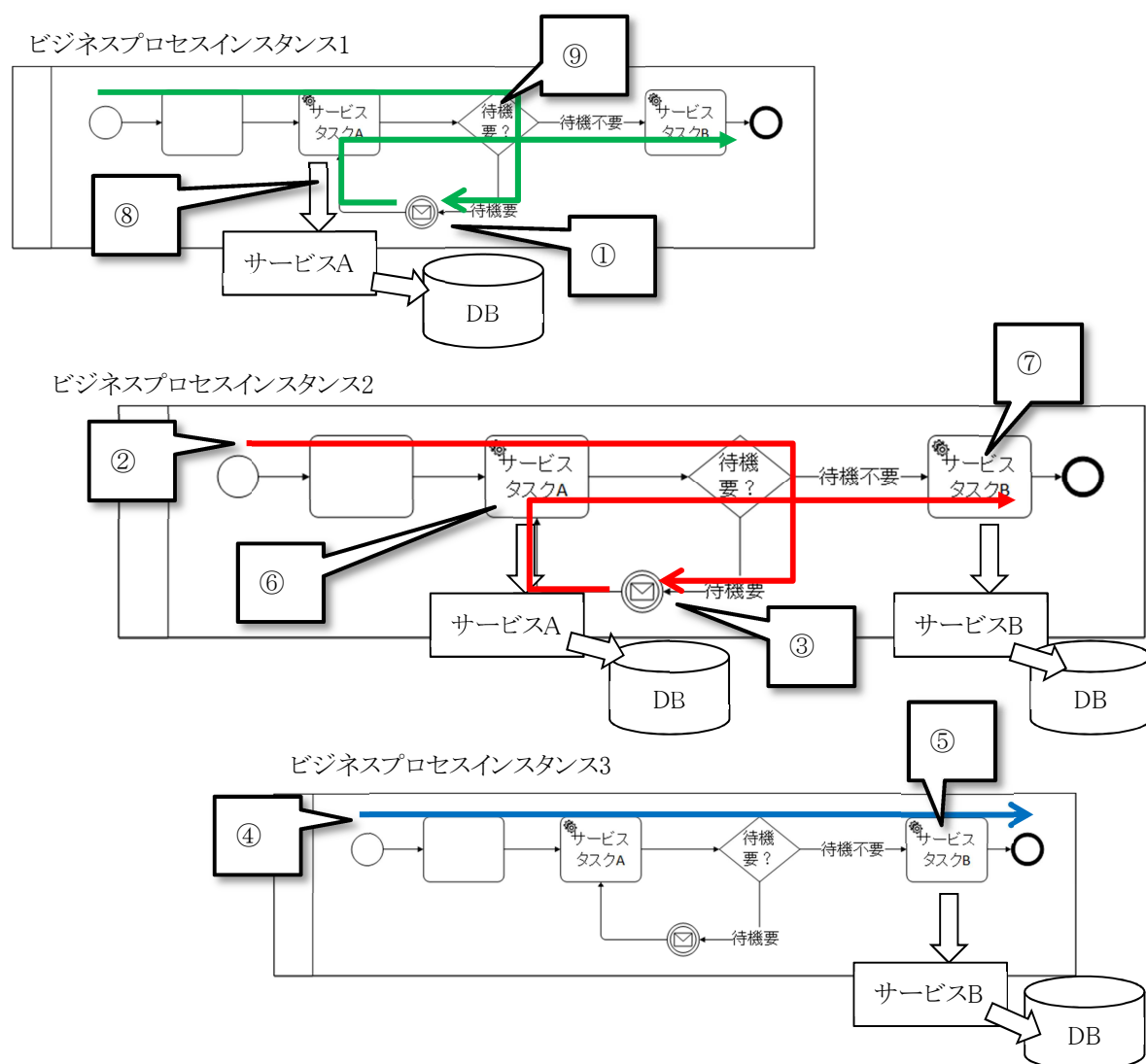
- 全てをBPMNで記載せず、ビジネスプロセスインスタンス間の実行順序制御が必要となる適切なポイントで業務アプリケーション（サービス）を呼び出し、データベースの最新状態から自ビジネスプロセスを待機させるか判断させる。

これは、ビジネスプロセスとしてのアクティビティの順序関係以外に、粒度が細かいシステムの実装上必要な順序制御のフローが混在することで、ビジネスプロセスの可視性・保守性が低下することを防止するためである。

- 判定条件がビジネスルール化の特性に適すると判断できる場合は、ビジネスルールとして業務アプリケーションから切り離す仕組みを利用すること。

ビジネスルールを業務アプリケーションから切り離す仕組みについては、「3.2.7.1.1 ビジネスルールをアプリケーションから切り離すための仕組みとして利用する技術」参照のこと。

再帰的な構造を持ったビジネスプロセスインスタンス間の実行順序制御の処理イメージを下图に示す。



- ① サービスタスク A から呼び出したサービス A の処理結果より判定した結果、待機するフローへ進み、中間イベント(中間タイマーメッセージイベント、キャッチ中間タイマーイベント等)で待機。
- ② 業務的なイベント発生を契機として新たなプロセスインスタンスが開始。
- ③ サービスタスク A から呼び出したサービス A の処理結果より判定した結果、待機するフローへ進み、中間イベント(中間タイマーメッセージイベント、キャッチ中間タイマーイベント等)で待機。
- ④ 業務的なイベント発生を契機として新たなプロセスインスタンスが開始。
- ⑤ サービスタスク A から呼び出したサービス A の結果、待機するフローへ進まず、サービスタスク B のサービス B を実行し、データベースを更新。
- ⑥ 待機状態から抜けた後、再度サービスタスク A からサービス A を実行し、データベースの最新情報を参照した結果、待機要因が解除されていると判定。
- ⑦ 再度サービスタスク A から呼び出したサービス A の結果、待機するフローへ進まず、サービスタスク B からサービス B を実行し、データベース上の案件の情報を更新。
- ⑧ 待機状態から抜けた後、再度サービスタスク A のサービス A を実行し、データベースの最新情報を参照した結果、待機要因が解除されていると判定。
- ⑨ 再度サービスタスク A から呼び出したサービス A の結果、待機するフローへ進まず、次のアクティビティへ進む。

図 3.2-12 再帰的な構造を持ったビジネスプロセスインスタンス間の実行順序制御

なお、可読性を低下することなくビジネスプロセスを管理するため、以下のように詳細なフローを隠蔽する。

- 排他ゲートウェイと中間イベントを使った待機処理が必要なアクティビティをサブプロセス化すること(ただし、ビジネスプロセスの可読性が低下しない場合に限る。)

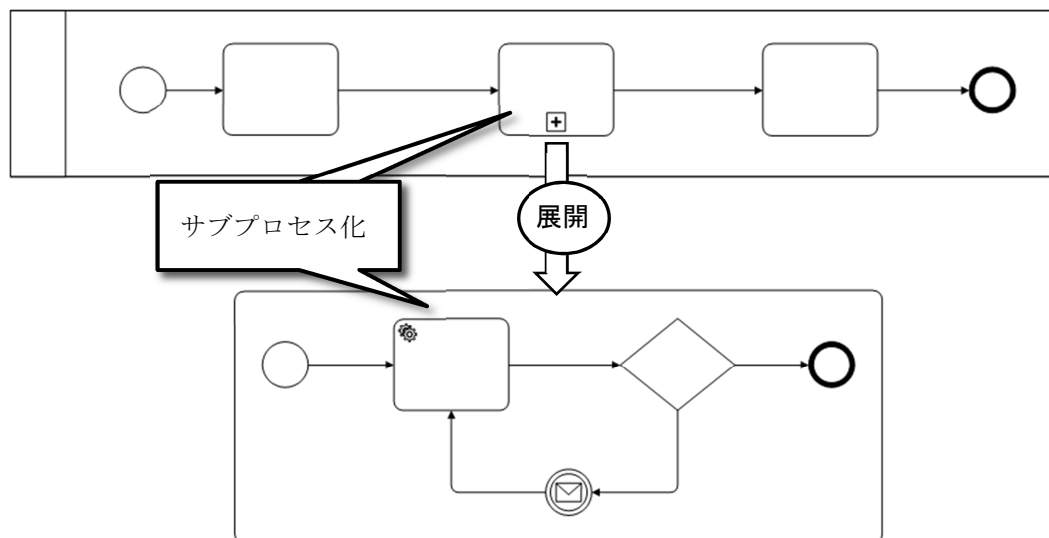


図 3.2-13 再帰的な構造をビジネスプロセスインスタンス間の実行順序制御のサブプロセス化

(2) アクティビティの割り当てが組織構造と密接な関係を持つビジネスプロセスの設計指針

A. アクティビティに対応するヒューマンタスクの動的な絞込み

BPMSでは、BPMNのユーザタスクによって、システム利用者が業務システムの画面を対話的に利用して実行する業務(ヒューマンタスク)を記載できる。ただし、ユーザへのヒューマンタスク割り当て方法は、BPMN仕様だけでは明確に定義されておらず、現状は、製品の実装に依存するところが多い。

BPMS製品の一般的な機能としては、ユーザタスク又はユーザタスクが配置されたBPMNのレーンに対するプロパティ設定で、LDAP等で管理されているユーザやグループをマッピングするといった静的な設定を実施することができる。これにより、製品がサポートする画面機能、又は独自に開発した画面APからワークフローAPIを使ってBPMSを呼び出すことで、起動中のビジネスプロセスインスタンスのうち、対象のユーザ又はユーザが属するグループが実行すべき特定のユーザタスク上で待機しているものを一覧取得することができるようになっている。

しかし、特許庁業務には、ユーザタスクやレーンに設定した静的なロール情報による検索だけでなく、動的に変化する業務的な情報(プロセスデータ)による検索が必要な場合がある。

特許庁業務におけるアクティビティに対応するヒューマンタスクの動的な絞込みの例として、実体審査における「予約」、「起案」、「決裁」がある。それぞれ案件を選択するために「予約対象の出願(案件)」、「起案対象の出願」、「決裁対象の出願」を一覧取得する。例えば、「予約対象の出願(案件)」の一覧取得の場合、出願が各審査室(又は審査グループ)に所属する審査官に振分けられるため、審査官全員でなく、振分け先の審査室(又は審査グループ)に所属する審査官のみに「予約対象の出願」の一覧を画面表示する必要がある。

このような検索条件が動的に変化する場合、以下のいずれかの方式を採用して検索条件に合致するアクティビティを絞り込むこと。²⁶

(A) 方式の概要

方式1)

ワークフローAPIを使ってBPMSが保持する業務データ(プロセスデータ)を検索する。

例：「予約対象の出願(案件)」の一覧取得の場合、一覧画面を利用するユーザが所属している審査室(又は審査グループ)を検索条件にして、ワークフローAPIでプロセスデータに格納されている振分け先の審査室(又は審査グループ)を検索する。

方式2)

ワークフローAPIを使ってBPMSが保持する業務データ(プロセスデータ)を検索した結果に対して、更に業務アプリケーションで対象を絞り込む。

例：「予約対象の出願(案件)」で特定の分類情報を持つ出願(案件)を一覧取得する場合、一覧画面を利用するユーザが所属している審査室(又は審査グループ)を検索条件にして、ワークフローAPIでプロセスデータに格納されている振分け先の審査室(又は審査グループ)を検索する。検索した結果に対して、更に、業務アプリケーションで分類情報に合致した出願(案件)を絞り込む。

方式3)

ヒューマンタスクのステータスを個別データベースで二重管理し、個別データベースのみを検索して対象を絞り込む。

例：「予約対象の出願(案件)」の一覧取得の場合、予め振分け業務の際、出願(案件)に対して「予約待ち」というステータスを個別データベースに格納しておき、一覧取得の際は、出願(案件)の「予約待ち」ステータスとユーザが所属する審査室(又は審査グループ)等を検索条件にして個別データベースを検索する。

²⁶ BPMS製品の機能でアクティビティの絞り込みが可能ならば、BPMS製品の機能を使用してアクティビティの絞り込みを行ってもよい。

(B) 方式の使い分け条件

方式の使い分けの条件は以下のとおりである。

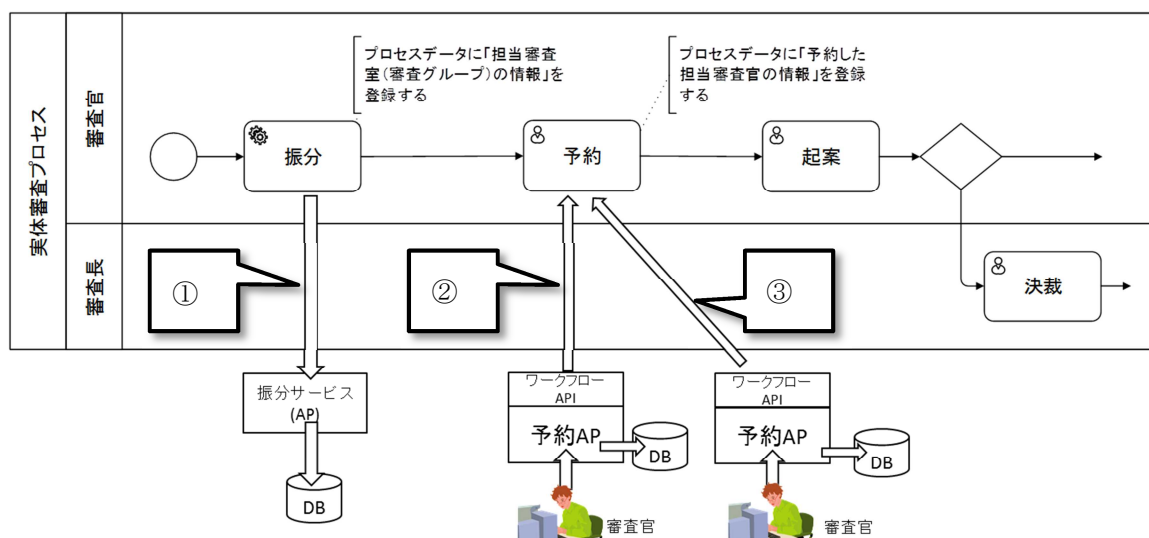
- ビジネスプロセスの保守性、異常時のリカバリの観点から、原則、方式1を採用する。
方式1を採用する場合は、以下の点に留意すること。
 - 製品によっては、ワークフローAPIの返却データ形式がXML等のファイルの場合があり、その構文解析処理に多くの時間がかかる場合がある。
- 方式1のワークフローAPIだけでは、対象のアクティビティを絞り込めない場合、方式2を採用する。
方式2を採用する場合は、以下の点に留意すること。
 - ワークフローAPIの返却件数が大量であった場合、業務アプリケーションによる更なる絞り込み処理に時間がかかる。
 - 製品によっては、ワークフローAPIの返却データ形式がXML等のファイルの場合があり、その構文解析処理に多くの時間がかかる場合がある。
- 対象のアクティビティの絞り込みの際、性能面に懸念がある場合は、方式3を採用する。
例えば、画面に発明の名称等、出願に関する様々な詳細情報を一覧表示するような場合、方式1及び方式2では、ワークフローAPIでアクティビティの一覧を取得後、アクティビティ1つ1つに対して、アクティビティが持つ業務キー(出願番号等)を検索条件にしてデータベースへSQLを発行し、出願の詳細情報を取得する(一般的にBPMSがステータスを管理するデータベースと、BPMS外部のデータベースは結合して検索できないため。)。したがって、一覧表示の件数が多い場合、性能面で懸念が生じる。
また、ワークフローAPIで対象のアクティビティを十分に絞り込むことができない場合、方式2における業務アプリケーションの絞り込み処理に多くの時間がかかる可能性がある。
このように方式1及び方式2では一覧画面表示処理の性能要件を満たせない場合、方式3を採用する。

方式3を採用する場合は、以下の点に留意すること。
 - ビジネスプロセスの変更が発生した場合には、BPMNで記載したビジネスプロセスの定義情報の変更だけでなく、データベースの設計や業務アプリケーションのステータス更新部分の設計にも変更が発生する。
 - 異常時、データベースとBPMS上のステータスに不整合が発生した場合のリカバリが必要である。

(C) 方式の詳細

方式1)

ワークフローAPIを使ってBPMSが保持する業務データ(プロセスデータ)を検索する例として、「予約対象の出願(案件)」の一覧取得の処理イメージを下図に示す。



- ① BPMS より、サービス呼び出し、対象の出願に対して、振分け先の担当審査室(又は審査グループ)を BPMS に返却。BPMS が保持するプロセスデータに担当審査室の情報を保存。
- ② ユーザからの要求により、AP はワークフローAPI を使って、プロセスデータとして保存された担当審査室に対し、ユーザの属する審査室を検索条件として予約対象のアクティビティを取得。アクティビティ情報に含まれる業務キー(出願番号等)をもとにデータベースへアクセスし出願の詳細情報を取得し、一覧画面を表示。
- ③ ユーザが出願の予約を要求すると、AP はワークフローAPI を使って、業務として担当の審査官情報を送付しプロセスインスタンスを更新することで、BPMS が保持するプロセスデータに担当審査官が保存されるとともに、起案のユーザタスクへ進む。

図 3.2-14 アクティビティに対応するヒューマンタスクの動的な絞込み(方式1)

方式2)

ワークフローAPIを使ってBPMSが保持する業務データ(プロセスデータ)を検索し、更に業務アプリケーションで対象を絞り込む。

ワークフローAPIを利用する点は方式1と同様であるが、方式2では、ワークフローAPIが返却したアクティビティやプロセスデータの情報を、業務アプリケーションで更に絞り込みを行う。

「予約対象の出願(案件)」の一覧取得を例にすると、業務アプリケーションの絞り込み処理は、「図 3.2-14 アクティビティに対応するヒューマンタスクの動的な絞込み(方式1)」②の「予約AP」にて行う。

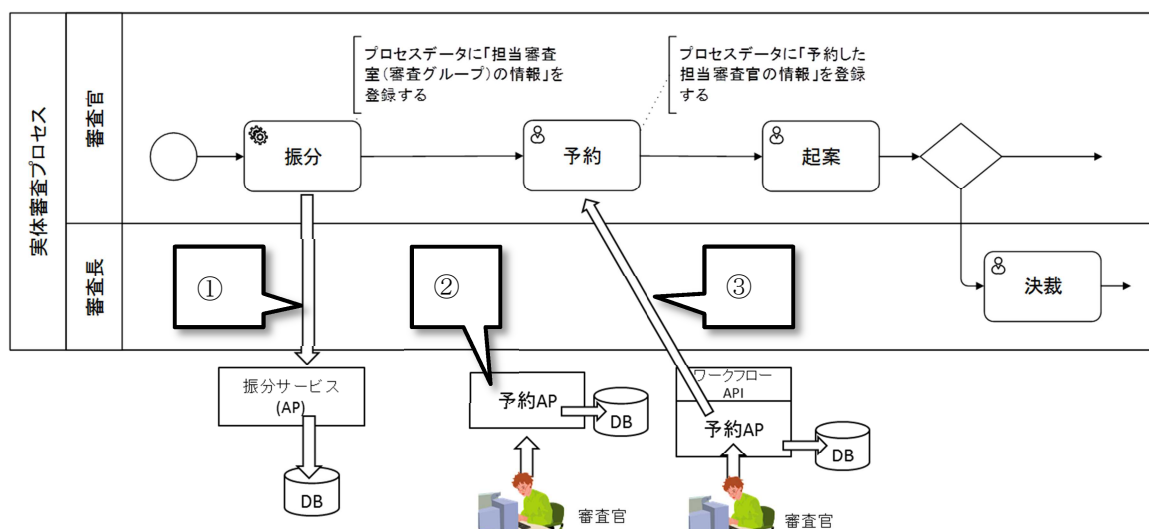
方式3)

ヒューマンタスクのステータスを個別データベースで二重管理し、アクティビティの一覧を取得するときは個別データベースのみを検索する。

BPMSと個別データベースの二重管理となり、保守性のデメリットがあるため、ヒューマンタスクのステータスに関してのみ、個別データベースで二重管理する。

一覧取得時にはBPMSにアクセスせず、個別データベースだけ検索できるようにすることで、性能要件に対処する。アクティビティを更新しプロセスを進行させる際には、方式1と同様ワークフローAPIを使ってBPMSを呼び出す。このとき、個別データベースで管理するプロセスの進捗状況も更新する。

例として、「予約対象の出願(案件)」の一覧取得の処理イメージを下図に示す。



- ① BPMS からサービスを呼び出し、対象の出願に対して、振分け先の担当審査室を決定。また、個別データベースに予約待ちであることを保存。
- ② ユーザからの要求により、AP は予約待機状態かつユーザの属する審査室を検索条件として個別データベースを検索し、対象の出願を取得。出願の詳細情報も同時に取得し、一覧画面を表示。
- ③ ユーザが出願の予約を要求すると、AP はワークフローAPI を使ってプロセスインスタンスを更新。個別データベースに当該ユーザを担当審査官として保存するとともに起案待ちであることを保存。

図 3.2-15 アクティビティに対応するヒューマンタスクの動的な絞込み(方式3)

B. 担当者の変更運用

ユーザの実行待ちとなっているアクティビティについて、人事異動等の理由により担当者の変更をしなければならないケースがある。例えば、通常、差戻し時は起案者である担当審査官に割り当ててが、起案者が審査室外へ異動していた場合に指定分類の代表担当官に差戻したり、審査グループ案件で審査グループ外に異動していた場合に審査グループ長へ差戻したりする等、起案者とは別の審査官に差戻す必要がある。

このような場合、以下のいずれかの方式を採用すること。

- 「A. アクティビティに対応するヒューマンタスクの動的な絞込み」にて、方式1及び方式2を採用する場合には、担当者異動時、ワークフローAPIを使ってBPMSで担当者を保持するプロセスデータを更新する方式を採用する。

担当者の情報をBPMS上のプロセスデータに持つため、担当者異動時にワークフローAPIを使用して、プロセスデータの値を変更する必要があるためである。

- 「A. アクティビティに対応するヒューマンタスクの動的な絞込み」にて、方式3を採用する場合には、担当者異動時、個別データベースに保持する担当者情報を変更する方式を採用する。

担当者の情報はBPMS上のプロセスデータには保持せず、個別データベース上のみに持っているため、担当者異動時に個別データベースの値を変更する必要があるためである。

A. ビジネスプロセスが開始する前に、中間イベントが通知されるケース

この場合、対象のビジネスプロセスインスタンスが存在しないため、BPMSがエラー又は誤動作してしまう。

このようなケースでは、以下の設計指針に従って、設計すること。

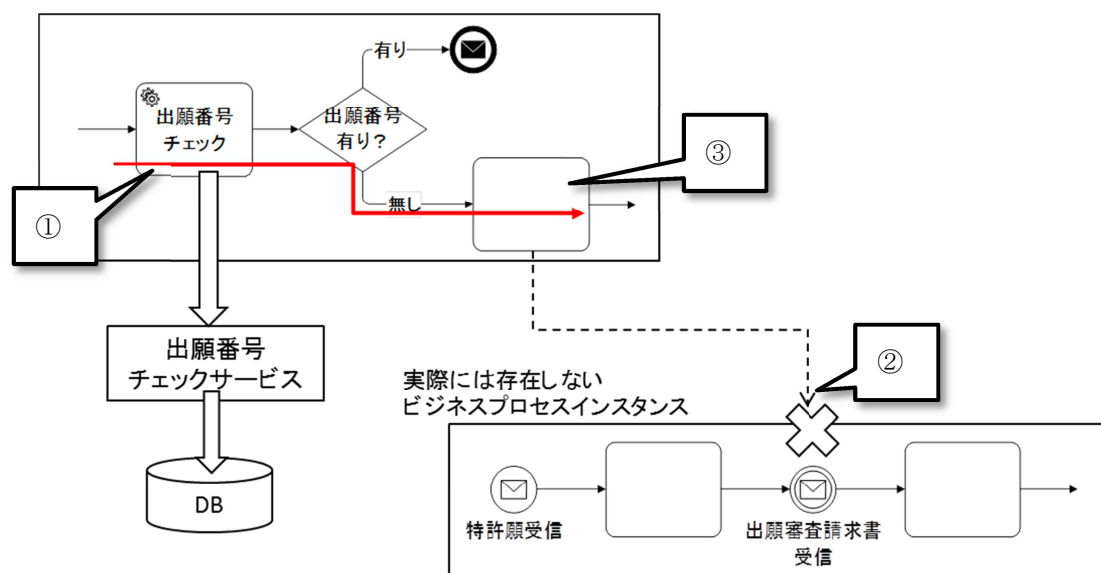
- BPMSを呼び出す前に、業務的なチェックをし、ビジネスプロセスインスタンスが存在しないと判断される場合には、BPMSへ通知しないように設計する。

例えば、出願審査請求書を受理した場合、呼び出し元のビジネスプロセス等で、事前に特許願を受理しているか(出願番号が存在するか)といった業務的なチェックを行い、存在しない場合には、BPMSを呼び出さないようにする。

BPMSの呼び出し元は、以下の5つのパターンがある。パターン毎の事前の業務チェックの処理箇所を以下に示す。

項番	BPMS呼び出し元	事前チェック箇所
1	ビジネスプロセス	中間イベントを通知するアクティビティの直前。
2	BPMS補完機能	BPMS補完機能にてBPMSを呼び出す直前。
3	業務アプリケーション(ユーザ)	事前チェックは、BPMS補完機能に委譲する。
4	業務アプリケーション(バッチ)	事前チェックは、BPMS補完機能に委譲する。
5	ESB	事前チェックは、BPMS補完機能に委譲する。

例として、出願審査請求書受理時のビジネスプロセスが呼び出し元だった場合の処理イメージを下图に示す。



- ① 出願番号が存在するか(願書を受理済か)の業務的なチェックを実施し、ビジネスプロセスインスタンスが起動していることを事前にチェック。
- ② 呼び出し元のチェック結果の判定により、BPMS に中間イベントは通知しない。
- ③ 必要に応じて、想定外のケースへ対処するフローへ進む。

図 3.2-16 ビジネスプロセス開始前での中間イベント通知防止の制御

B. イベントゲートウェイを利用し、一方のイベントが通知されフローが進行してしまった後に、他方のイベントが通知されるケース

例えば、拒絶理由通知応答待ちについて、意見書の受理のイベントと期限経過のイベントに対するBPMNのイベントゲートウェイで表現することができる。しかし、応答期限経過後に意見書を受理した場合、期限経過のイベント発生によるフローへ進んでしまうにもかかわらず、期限ぎりぎりで受理した書類だったため受付業務実施中に応答期間経過チェックのバッチ処理が先に動いてしまうケースや、戻し止めの運用のように期間内の受理として認めるケースが想定される。

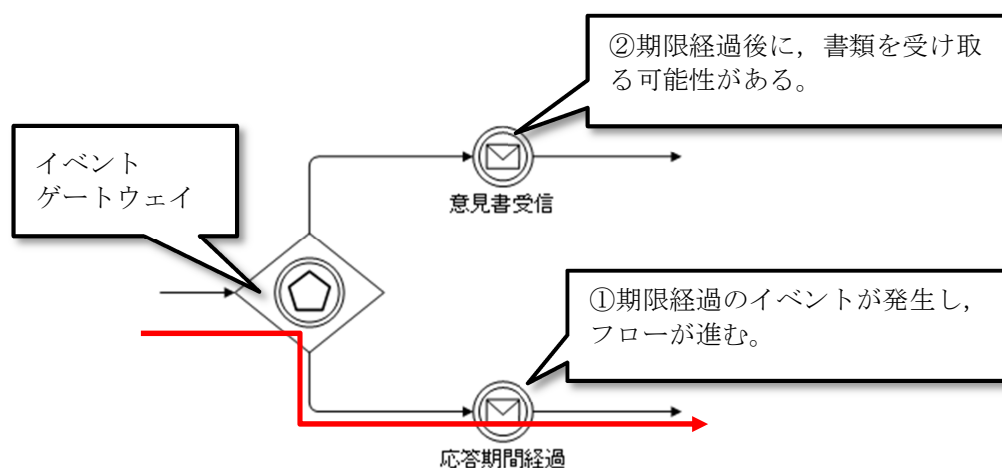


図 3.2-17 イベントゲートウェイによる排他処理

応答期限を経過した場合、BPMN上は既に期限経過のフローへ進んでしまっているため、意見書を受け取ったことによるイベントをビジネスプロセスに通知したとしても、元のフローに戻すことはできない。また、イベントが通知されてしまった場合、製品によってはエラーになったり、誤動作の原因になったりするおそれがある。

このようなケースでは、以下の設計指針に従って、設計すること。

- 呼び出し元のビジネスプロセス及び業務アプリケーションで、業務的なチェックを実施し、既に他方のイベントが発生しフローが進行していると判断される場合にはBPMSに通知しないようにする。

例えば、意見書受理のイベントをBPMSに通知する前に、当該出願が期限経過しているか必ず事前にチェックし、既に期限経過している場合には、BPMSに通知しないように設計する。

BPMSの呼び出し元別の事前の業務チェックの処理箇所は、「表 3.2-17 BPMSの呼び出し元と事前チェック箇所」と同様である。

例として、意見書受理時のビジネスプロセスが呼び出し元だった場合の処理イメージを下图に示す。

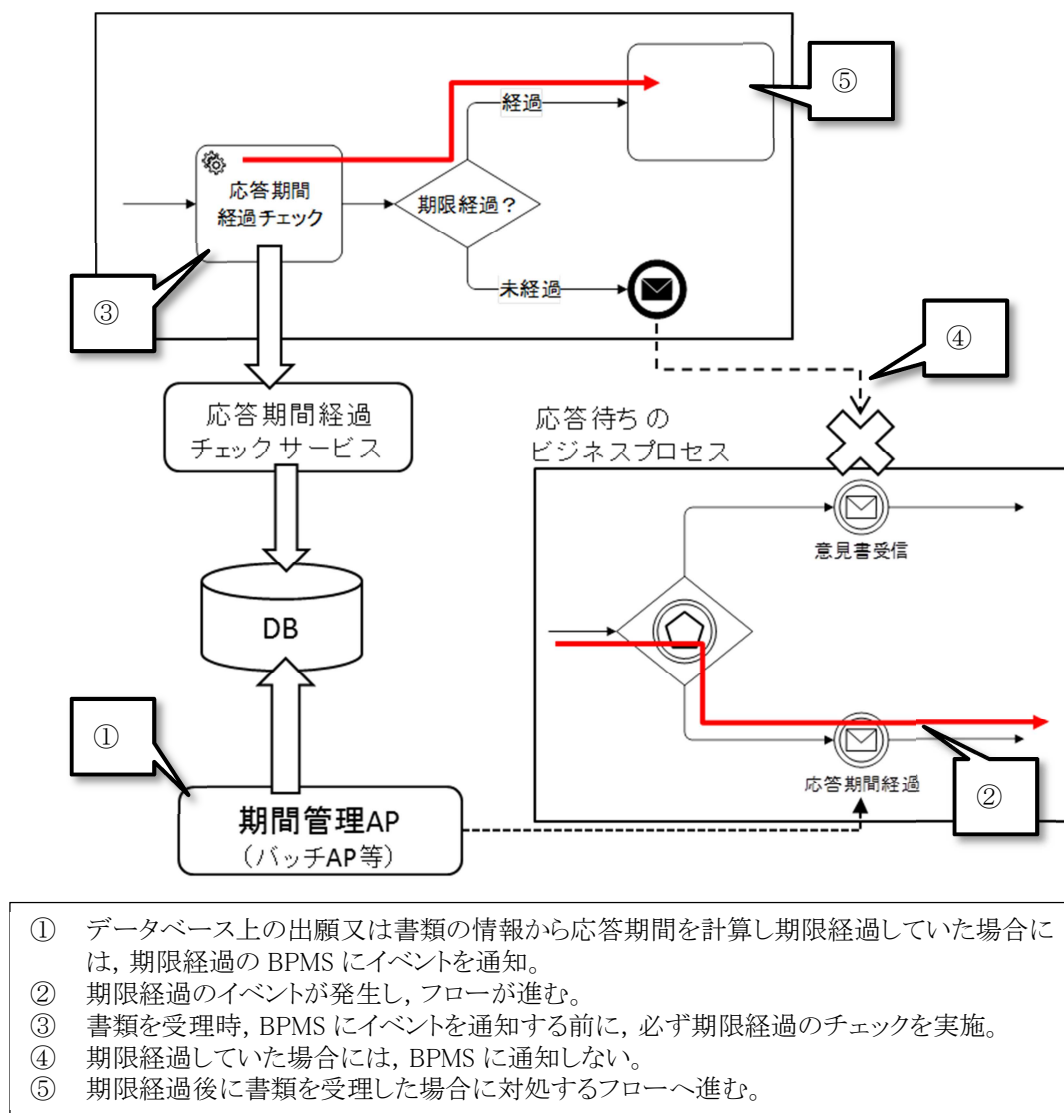


図 3.2-18 イベントゲートウェイのフロー進行後のイベント通知防止の制御

3.2.3.1.3 ビジネスプロセスのステータス管理の設計指針

スコープ：階層定型化サブシステム

プロセス：2.4.3

指針1： ビジネスプロセスのプロセスデータは、BPMSではキー情報だけを保持するように設計すること。

指針2： ビジネスプロセスのステータスの管理は、「(1)ステータス管理のルール」に従って、設計すること。

特例1： ヒューマンタスクにおけるステータスの一覧取得時の性能面に懸念がある場合には、ステータスの一部をデータベースにも保持し二重管理することを許容する。

指針3： 待機要因の管理は、「(2)待機要因の管理ルール」に従って、設計すること。

ビジネスプロセスをBPMSで制御する際に使用するデータは、二重管理の防止のため、BPMSではキー情報だけを持つようにする。特に、LDAP上で定義されている組織とユーザに関する情報はBPMSで二重管理しない。キー情報の例として、ビジネスプロセスインスタンスを一意に特定するキー情報、ビジネスプロセスインスタンスに対応する事件や書類のキー情報、アクティビティ(ユーザタスク)の担当者のキー情報、アクティビティの実行責務を負うロール情報、ビジネスプロセスの分岐先を判断する情報、ビジネスプロセスのステータスがある。

特許庁業務におけるビジネスプロセスのステータスは、ステータスを確認する人の視点から2つに分類される。

① 申請人又は他の課室の職員から見た大きなプロセスの進行状況。

例えば、方式係属中、方式完、審査係属中、査定、審判係属中、登録係属中といった書類や出願(事件)の大きな審査状況

② 課室内における職員レベルでの詳細なプロセスの進行状況。

例えば、目視審査中、起案書決裁中といった方式審査室内の申請書類の審査状況

(1) ステータス管理のルール

上記①、②のいずれも、以下のルールに従い、ステータスを管理する。

- 原則、BPMSのステータス管理機能を利用して管理すること。

BPMSのステータス管理機能は、ビジネスプロセスがどのアクティビティまで進んだかといったビジネスプロセスの進行状況、また、アクティビティの実行者、実行日時といった実行履歴を管理する機能を指す。

- ヒューマンタスクの状態も原則、BPMSの機能を利用して管理し、WS-HumanTask²⁷で定義される状態のいずれかに対応付けができるよう設計すること。BPMS製品が管理するヒューマンタスクの状態が必ずしもWS-HumanTask準拠である必要はないが、WS-HumanTaskで定義されるいずれかの状態(「Created」、「Ready」、「Reserved」等)へと対応付けが取れるようにすること。
- 特例として、ヒューマンタスクにおけるステータスの一覧取得時の性能面に懸念がある場合には、ステータスの一部をデータベースにも保持し二重管理することを許容する。ただし、以下の点に留意して利用すること。
 - ビジネスプロセスの変更が発生した場合には、BPMNで記載したビジネスプロセスの定義情報の変更だけでなく、データベースの設計や業務アプリケーションのステータス更新部分の設計にも変更が発生する。
 - 異常時、データベースとBPMS上のステータスに不整合が発生した場合のリカバリが必要である。

²⁷ 『Web Services Human Task (WS-HumanTask) Specification』 <http://docs.oasis-open.org/bpel4people/ws-humantask-1.1.html>

(2) 待機要因の管理ルール

特許庁システムで扱う「待機要因」は、ビジネスプロセスがなぜ待機しているのかの要因を示すものであり、BPMSのステータス管理で管理できない。申請書類の受理等の他のビジネスプロセスのイベントによって、待機要因の内容が変更されたり、新たな待機要因が増えたりする等、刻々と変化していくこともある。したがって、業務アプリケーションによる導出が必要となる。

待機要因の管理ルールについて、以下に示す。

- 待機要因は、データベースの情報や、関連するビジネスプロセスインスタンスのBPMS上のステータス等を利用して、業務アプリケーションで導出すること。

3.2.3.1.4 ビジネスプロセスを介したデータの授受ルール

スコープ：階層定型化サブシステム

プロセス：2.4.3

指針1： ビジネスプロセスを介して授受するデータは、「(1)ビジネスプロセスを介して授受するデータのルール」に従って設計すること。また、「(2)ビジネスプロセスを介して授受するデータの設計時の留意事項」に示す留意事項を遵守すること。

(1) ビジネスプロセスを介して授受するデータのルール

ビジネスプロセスを介して授受するデータは以下の3つとすること。

- 業務対象を識別するID

出願番号や書類ID等, ビジネスプロセスインスタンスの業務対象を一意に示すID又はIDの組み合わせを指す。BPMSや他のシステム構成要素から対象のビジネスプロセスインスタンスを特定するために利用される。また, ビジネスプロセスが利用するサービスへのパラメータとして利用される。

- ビジネスプロセスの分岐条件を判定するために必要なデータ

審査結果や決裁結果等, ビジネスプロセスが利用するサービスのレスポンス等がそれにあたる。

- データベースに保管すべきでないデータ

一時的に生成され利用後に破棄される, 永続化の必要のないデータを指す。

(2) ビジネスプロセスを介して授受するデータの設計時の留意事項

上述「(1)ビジネスプロセスを介して授受するデータのルール」で示したデータ以外にビジネスプロセスが利用するサービスのパラメータとして必要なデータがあった場合は, サービスインタフェースが正しい粒度で設計されていない可能性がある。「3.2.4.2 サービスインタフェースの粒度」を参考にサービスインタフェースの再設計を検討すること。

3.2.3.2 BPMNの記載ルール

(1) BPMN記載ルールの方針

BPMN記載ルールの方針を以下に示す。

(方針1) 統一的な記載が可能で、ビジネスプロセスの可読性、保守性の向上に寄与するルールとする。

以下のような問題が発生しない記載ルールとする。

- プロセスモデリングの担当者毎に、同じ意味を持つビジネスプロセスが、異なった形で記述され、統一的でない。
- 一般的でない記法を使い、難解なプロセスで一部の有識者しか理解できない。
- 適切な粒度でプロセスが分割されておらず、保守性が低い。
- 必要な情報が記載されていない、又は不必要で冗長的な記載がある。

(方針2) 開発工程の進行に伴うプロセスの詳細化に対してトレーサビリティを保持できるルールとする。

以下のような問題が発生しない記載ルールとする。

- 工程によってプロセス図が大きく変更され、正しく詳細化されているか判断しにくい。
- 上流工程にさかのぼって修正が必要な場合、その修正箇所の特定制がしにくい。

(方針3) 特定のBPMN製品に限定されるようなルールにしない。

- 「2 ルールの考え方」に基づき、製品固有の機能に依存したルールにしない。
- BPMN標準ではあっても一般的にサポートされていない機能に依存したルールや実装上の差異が記載方法に影響しやすいルールにしない。
- プロセス図の表記以外の実装部分(プロセスデータ設定、サービスインタフェース定義等)は製品固有部分が多いため、ルール化の対象外とする。

(2) ルールの適用範囲

ルールの適用範囲を下表に示す。

表 3.2-18 BPMN記載ルールの適用範囲

項番	項目	対象
1	BPMN仕様	BPMN2.0
2	BPMNモデリングレベル ²⁸	分析モデル 実行可能モデル
3	図	プロセス図(Process diagrams)

²⁸ 記述モデルはルールの適用対象外だが、記述モデル作成の際は、後続の分析モデルや実行可能モデルによる成果物との対応付けが取りやすくなるよう留意すること。

3.2.3.2.1 BPMN要素の使用ルール

スコープ：階層定型化サブシステム

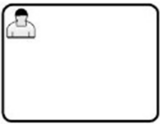
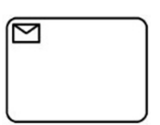
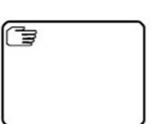
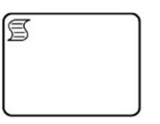
プロセス：2.3.2, 2.4.3

規約1： ビジネスプロセスの記載に使用するBPMN要素は、「表 3.2-19」～「表 3.2-28」に示す使用可否、及び制約事項のとおり使用すること。

BPMN要素の使用制限と使用する際の制約事項を以下に示す。

(1) アクティビティ

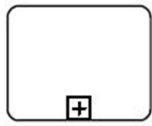
表 3.2-19 BPMN要素の使用ルール(アクティビティ・タスクタイプ)

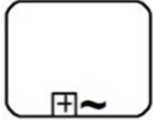
項番	図形名称	図形	説明	使用可否	制約事項	補足説明
1	タスク (タスクタイプなし)		タスクタイプを指定しないタスクを表す。	×	—	使用不可理由： 記述レベルで使用されるが、分析レベル以降では使用しない図形であり、トレーサビリティを損なわないため。 代替方法： なし
2	ユーザタスク		人が介入して実行する業務処理を表す。	○	無	制約事項： なし
3	サービスタスク		人の介入なしに自動で実行される業務処理を表す。	○	無	制約事項： なし
4	送信タスク		メッセージの送信処理を表す。	×	—	使用不可理由： スロー中間メッセージイベントと同じ意味であるため。 代替方法： スロー中間メッセージイベントを使用する。
5	受信タスク		メッセージの受信処理を表す。	×	—	使用不可理由： キャッチ中間メッセージイベントと同じ意味であるため。 代替方法： キャッチ中間メッセージイベントを使用する。
6	マニュアルタスク		人だけで実行される「手作業」を表す。	×	—	使用不可理由： システム化されないアクティビティであり、ビジネスプロセスモデリング対象外であるため。 代替方法： なし
7	スクリプトタスク		スクリプトによる処理を表す(プロセスデータのデータ変換等を行う)。	×	—	使用不可理由： プロセスデータは、キー情報で構成されており、スクリプトによるデータ変換が必要とされる可能性が低い。

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
						代替方法: なし
8	ビジネスルールタスク		ビジネスルールによる処理を表す。	×	—	使用不可理由: ToBeアーキテクチャでは、BPMSからBRMSを直接呼び出しせず、サービスを介して呼び出すため。 代替方法: サービスタスクからBRMS上のビジネスルールを呼び出す。

使用可否の凡例：○は使用可，×は使用不可を表す。

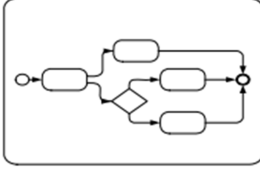
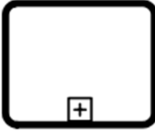
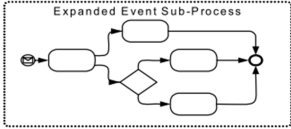
表 3.2-20 BPMN要素の使用ルール(アクティビティ・マーカ)

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
1	タスク (アクティビティ・マーカなし)		アクティビティ・マーカを指定していないタスクを表す。	×	—	使用不可理由: 記述レベルで使用されるが、分析レベル以降では使用しない図形であり、トレーサビリティを損なわないため。 代替方法: なし
2	折りたたまれたサブプロセス		サブプロセスを包含するアクティビティを表す。	○	無	制約事項: なし
3	ループ		繰り返し実行するアクティビティを表す。	×	—	使用不可理由: 代替方法があり、現時点で特許システムにてループタスクにするべきケースが見つからないため。必要性が判明した時点で使用可とする。 代替方法: タスクの後続に、繰り返し処理をするか否かを判定するゲートウェイを配置し、繰り返し処理をする場合にはシーケンスフローで対象処理に戻るようビジネスプロセスを作成する。
4	パラレルマルチインスタンス		プロセスデータ内の配列データを並行処理するアクティビティを表す。	×	—	使用不可理由: ビジネスプロセス途中でインスタンス単位が異なる場合は、別ビジネスプロセスとして定義するため。 代替方法: 別ビジネスプロセスで定義し、BPMS補完機能等経由し

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
						て、呼び出しを行う。
5	シーケンシャルマルチインスタンス		プロセスデータ内の配列データを1データ毎に逐次処理するアクティビティを表す。	×	—	使用不可理由： ビジネスプロセス途中でインスタンス単位が異なる場合は、別ビジネスプロセスとして定義するため。 代替方法： 別ビジネスプロセスで定義し、BPMS補完機能等経由して、呼び出しを行う。
6	アドホック	 折りたたまれたサブプロセスの例	サブプロセス内のアクティビティを実行順、実行数を予め定義せず、サブプロセスの実行者によって判断するアクティビティを表す。	×	—	使用不可理由： 現時点で特許システムにて使用するケースが見つからず、要否が不明であるため。 必要性が判明した時点で使用可とする。 代替方法： なし
7	補償		アクティビティの実行結果が意図しない結果になった場合に処理前の状態に戻すアクティビティを表す。	×	—	使用不可理由： BPMS製品により対応状況が異なる。対応していない場合、ビジネスプロセスを大幅に変更する必要があり、分析モデルと実行可能モデルで大きな乖離が発生し、トレーサビリティが損なわれるため。 代替方法： 補償処理を表す、ビジネスプロセスを別途サブプロセス等で定義する。

使用可否の凡例：○は使用可，×は使用不可を表す。

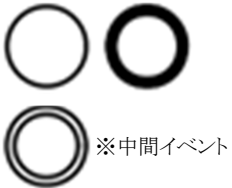
表 3.2-21 BPMN要素の使用ルール(アクティビティ・その他)

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
1	展開されたサブプロセス		「折りたたまれたサブプロセス」で省略されたサブプロセスの詳細を表す。	○	無	制約事項: なし
2	コールアクティビティ	 ユーザタスクの例  折りたたまれたサブプロセス ※境界が太線	グローバルに定義された、ビジネスプロセスあるいはタスクを表す。実行中のビジネスプロセスで再利用することができる。	○	無	制約事項: なし
3	イベント・サブプロセス	 展開されたイベント・サブプロセスの例 ※境界が点線	ビジネスプロセス又はサブプロセスの中に置かれ、スタート・イベントのトリガでアクティブになるアクティビティを表す。所属するビジネスプロセスを並行して(非中断で)実行することができる。	○	無	制約事項: なし
4	トランザクション・サブプロセス	 ※境界が二重線	ロングトランザクションが必要とされる複数のアクティビティの集合を表す。	×	—	使用不可理由: 補償タスクと同様に、BPMS製品により対応状況が異なる。対応していない場合、ビジネスプロセスを大幅に変更する必要があり、分析モデルと実行可能モデルで大きな乖離が発生し、トレーサビリティが損なわれるため。 代替方法: なし

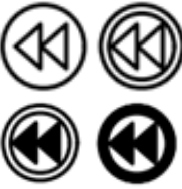
使用可否の凡例: ○は使用可, ×は使用不可を表す。

(2) イベント

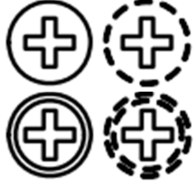
表 3.2-22 BPMN要素の使用ルール(イベントタイプ)

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
1	イベント(イベントタイプなし)		タイプの指定されないイベントを表す。	○	有	制約事項: ● 中間イベントは使用不可とする。 理由: 中間イベントは、トリガとなるイベントが明確であるため、イベントタイプを持つイベントを利用する。
2	メッセージ		メッセージの送受信のイベントを表す。	○	有	制約事項: ● イベント・サブプロセス(親ビジネスプロセス中断)開始イベント、境界(中断)イベントは使用不可とする。 理由: 現時点で特許庁業務にて使用するケースが見つからず、要否が不明であるため。必要性が判明した時点で使用可とする。 ● 境界(非中断)イベントは使用不可とする。 理由: イベント・サブプロセスで代替可能であり、表記の統一を図るため。 ● メッセージフローを使用しない場合はメッセージ送信先又は受信元の情報をテキスト注釈で記載すること。 理由: メッセージの送信先又は受信元を明確にするため。 なお、メッセージフローは実行可能モデルでは使用できないため、分析モデルでメッセージフローを使用する場合、実行可能モデルではメッセージ送信先/受信元の情報を表す代替方法として、テキスト注釈を使用すること。

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
3	タイマー		周期的なイベント, 時間の経過やタイムアウトを表す。	○	有	スケジューリング処理は, BPMS外のシステム構成要素で対応する場合があります, 実行可能モデルでは, メッセージイベントに置き換えが必要になるケースもある。 制約事項: ● イベント・サブプロセス (親ビジネスプロセス中断) 開始イベント, 境界 (中断) イベントは使用不可とする。 理由: 現時点で特許庁業務にて使用するケースが見つからず, 要否が不明であるため。必要性が判明した時点で使用可とする。 ● 境界 (非中断) イベントは使用不可とする。 理由: イベント・サブプロセスで代替可能であり, 表記の統一を図るため。 ● 極力, 経過時間指定等で閉塞時間帯に稼動しないように設定すること。 理由: 業務閉塞した場合でも, BPMSのバックアップ等が正常に実施できない可能性があるため。詳細は, 「3.2.13.3 システム運転管理方式設計のルール」参照のこと。
4	エラー		業務継続できない重大なエラーのキャッチ, 又はスローのイベントを表す。	×	—	使用不可理由: 重大なエラー発生によるエラーイベントを契機に実行される業務的ビジネスプロセス処理は, 現時点で特許庁業務に見つからず, 要否が不明であるため。必要性が判明した時点で使用可とする。 代替方法: なし

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
5	エスカレーション		業務継続できる軽度なエラーのキャッチ, スローを表す。	×	—	使用不可理由: 現時点で特許庁業務にて使用するケースが見つからず, 要否が不明であるため。必要性が判明した時点で使用可とする。 代替方法: なし
6	キャンセル		トランザクションを表すサブプロセスの中で使用され, トランザクションのキャンセルを表す。	×	—	使用不可理由: 使用不可としているトランザクション・サブプロセスとともに使用されるイベントであるため。 代替方法: なし
7	補償		補償の実行や, 補償のトリガのイベントを表す。	×	—	使用不可理由: 補償タスクと同様に, BPMS製品により対応状況が異なる。対応していない場合, ビジネスプロセスを大幅に変更する必要がある, 分析モデルと実行可能モデルで大きな乖離が発生し, トレーサビリティが損なわれるため。 代替方法: 補償処理を表すビジネスプロセスを別途サブプロセス等で定義する。
8	条件		予め規定している条件を満たしたことを示すイベントを表す。	×	—	使用不可理由: 条件を満たしていることをモニタする場合, 条件を満たすまで待機する業務としてサービス化するため。 代替方法: 条件を満たしていることをモニタするサービスを設け, メッセージやシグナルイベントで通知を行う。

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
9	リンク		プロセス図の離れた場所にジャンプすることを表す。可読性を保つために利用する。	○	有	実行可能レベルでは使用できないBPMS製品が一部存在するが、トレーサビリティへの影響は少ないため、使用可とする。 制約事項: ● 同じページの離れた場所につながるシーケンスフローに対して、可視性を保つために使用する場合と複数ページにまたがるビジネスプロセスをつなげるために使用する場合があるが、後者は、ビジネスプロセス自体が長く複雑すぎる場合があるので、極力、使用せず、ビジネスプロセスの分離やサブプロセス化を検討すること。
10	シグナル		ブロードキャストされるイベントを表す。	○	有	制約事項: ● イベント・サブプロセス(親ビジネスプロセス中断)開始イベント、境界(中断)イベントは使用不可とする。 理由: 現時点で特許庁業務にて使用するケースが見つからず、要否が不明であるため。必要性が判明した時点で使用可とする。 ● 境界(非中断)イベントは使用不可とする。 理由: イベント・サブプロセスで代替可能であり、表記の統一を図るため。
11	停止		ビジネスプロセスの即時停止を表す(同じビジネスプロセス内の全インスタンスが停止。)	○	無	制約事項: なし

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
12	マルチプル		複数のイベントのうち、いずれかを受信又は複数のイベントを送信することを表す。	×	—	使用不可理由： 現時点で特許庁業務にて使用するケースが見つからず、要否が不明であるため。必要性が判明した時点で使用可とする。 代替方法： なし
13	パラレルマルチプル		複数のイベントの全てを受信することを表す。	×	—	使用不可理由： 現時点で特許庁業務にて使用するケースが見つからず、要否が不明であるため。必要性が判明した時点で使用可とする。 代替方法： なし

使用可否の凡例：○は使用可，×は使用不可を表す。

イベントを配置可能な場所と配置するイベント図形は、下表のとおりである(図形が記載されている箇所が使用可能を示す)。

表 3.2-23 BPMN要素の使用ルール(イベント配置可能な場所)

イベント タイプ	配置場所							
	開始			中間				終了
	トップ レベル	イベント・ サブプロセ ス (親ビジネ スプロセス 中断)	イベント・ サブプロセ ス (親ビジネ スプロセス 非中断)	キャッチ	境界 (中断)	境界 (非中断)	スロー	
タイプなし		—	—	—	—	—	—	
メッセージ		—			—	—		
タイマー		—			—	—	—	—
リンク	—	—	—		—	—		—
シグナル		—			—	—		
停止	—	—	—	—	—	—	—	

上表のとおり、以下は配置を不可としている。理由は、「表 3.2-22 BPMN要素の使用ルール(イベントタイプ)」の制約事項を参照のこと。

- 中間イベント(タイプなし)の配置は不可とする。
- イベント・サブプロセス(親ビジネスプロセス中断)、境界(中断)、境界(非中断)のイベントの配置は不可とする。

(3) ゲートウェイ

表 3.2-24 BPMN要素の使用ルール(ゲートウェイ)

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
1	排他ゲートウェイ		フローが排他的に分岐又は分岐したフローが結合することを表す。	○	無	製品により、対応している図形が異なるため、製品で対応している図形を使用する。製品が両方を対応している場合は、項番1を使用する。 制約事項： なし
2						
3	並列ゲートウェイ		フローが並列に分岐又は並列処理されていたフローが同期的に結合することを表す。	○	無	制約事項： なし
4	内包ゲートウェイ		定義された条件を全て評価し、適合する全ての分岐を実行することを表す。	×	—	使用不可理由： 現時点で特許庁業務にて使用するケースが見つからず、要否が不明であるため。必要性が判明した時点で使用可とする。 代替方法： なし
5	複合ゲートウェイ		排他、内包、並列の3種のゲートウェイによって表現できない複雑な分岐処理を行う場合に使用する。	×	—	使用不可理由： 現時点で特許庁業務にて使用するケースが見つからず、要否が不明であるため。必要性が判明した時点で使用可とする。 代替方法： なし
6	排他イベントゲートウェイ		後続の複数の受信イベントのうち、最初に受信したイベントのフローを選択することを表す。	○	無	制約事項： なし
7	排他イベントゲートウェイ (開始イベント)		開始イベントが複数存在する場合に、最初に受け取ったイベントのフローを選択することを表す。	×	—	使用不可理由： 現時点で特許庁業務にて使用するケースが見つからず、要否が不明であるため。必要性が判明した時点で使用可とする。 代替方法： なし
8	並列イベントゲートウェイ (開始イベント)		開始イベントが複数存在し、発生したイベントのフローを全て実行することを表す。	×	—	使用不可理由： 現時点で特許庁業務にて使用するケースが見つからず、要否が不明であるため。必要性が判明した時点で使用可とする。 代替方法： なし

使用可否の凡例： ○は使用可，×は使用不可を表す。

(4) 接続オブジェクト

表 3.2-25 BPMN要素の使用ルール(接続オブジェクト)

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
1	シーケンスフロー		フローの遷移を表す。	○	無	制約事項: なし
2	条件付きシーケンスフロー		フローの遷移に条件がある場合を表す。	×	—	使用不可理由: ゲートウェイで表記できるため。 代替方法: ゲートウェイを使用する。
3	デフォルト・シーケンスフロー		複数のシーケンスフローのうち、デフォルトのシーケンスフローを表す。	○	有	制約事項: ● ゲートウェイの分岐に対して2つ以上使用しない。
4	メッセージフロー ²⁹		ビジネスプロセス間でのメッセージの送受信の流れを表す。	○	有	制約事項: ● 実行可能モデルでは削除すること。 理由: 実行可能モデルでは使用できないため。
5	関連		関連を表す。	○	有	制約事項: ● BPMN要素とテキスト注釈との関連に使用する。 理由: その他の関連先(データオブジェクト等)は, BPMN要素が使用不可となっているため。
6	データ関連		データの関連を表す。	×	—	使用不可理由: データオブジェクトを使用不可としているため。 代替方法: テキスト注釈で表記する。

使用可否の凡例: ○は使用可, ×は使用不可を表す。

²⁹ 当該図形は実行可能モデルでは使用できないが, 分析モデルでの表現力が上がり, かつ分析モデルから実行可能モデルにBPMNモデリングレベルを詳細化する際, 図形の削除等, 影響の小さい修正が想定されるという理由により分析モデルでは使用可能とする。

(5) データ

表 3.2-26 BPMN要素の使用ルール(データ)

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
1	データオブジェクト		アクティビティが要求,あるいは生成するデータを表す。	×	—	使用不可理由: 実行可能モデルでは,サポートしていない。また,プロセスデータは,キーにあたるデータのみで構成されるため,分析モデルでの設計上の補助的な情報としても活用することがないため。 代替方法: 記載が必要になった場合は,テキスト注釈で表記する。
2	コレクション・データオブジェクト		複数のインスタンスを持つデータオブジェクトを表す。	×	—	「使用不可理由」,「代替方法」は, No1.データオブジェクトと同じ。
3	データ入力		入力として使用されるデータを表す。	×	—	「使用不可理由」,「代替方法」は, No1.データオブジェクトと同じ。
4	データ出力		出力されるデータを表す。	×	—	「使用不可理由」,「代替方法」は, No1.データオブジェクトと同じ。
5	データストア		ビジネスプロセス外に保持しているデータを表す。	×	—	「使用不可理由」,「代替方法」は, No1.データオブジェクトと同じ。

使用可否の凡例: ○は使用可, ×は使用不可を表す。

(6) スイムレーン

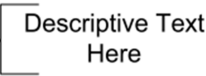
表 3.2-27 BPMN要素の使用ルール(スイムレーン)

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
1	プール		ビジネスプロセス全体を包含する区画を表す。	○	有	制約事項: ● ブラックボックスプール ³⁰ (外部関係者を表し、ビジネスプロセスを記述しないプール)については、実行可能モデルでは削除すること。 理由: ブラックボックスプールは実行可能モデルでは使用できないため。
2	レーン		役割や部署等によってプール内を整理・分割するために使用される区画を表す。	○	無	制約事項: なし

使用可否の凡例: ○は使用可, ×は使用不可を表す。

(7) その他

表 3.2-28 BPMN要素の使用ルール(その他)

項番	図形名称	図形	説明	使用可否	制約事項	補足説明
1	グループ		構成要素のグルーピングを表す。	×	—	使用不可理由: 実行可能モデルでは、使用できない図形であり、使用しない場合でも、可読性に影響ないため。 代替方法: なし
2	テキスト注釈		読み手に補足情報を提供するために使用する。	○	有	制約事項: ● 「関連」を使用して注釈対象と接続すること。

使用可否の凡例: ○は使用可, ×は使用不可を表す。

³⁰ 当該図形は実行可能モデルでは使用できないが、分析モデルでの表現力が上がり、かつ分析モデルから実行可能モデルにBPMNモデリングレベルを詳細化する際、図形の削除等、影響の小さい修正が想定されるという理由により分析モデルでは使用可能とする。

3.2.3.2.2 命名ルール

スコープ：階層定型化サブシステム

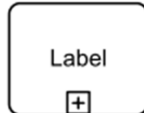
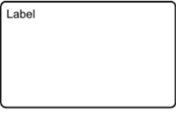
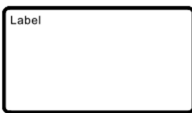
プロセス：2.3.2, 2.4.3

指針1： BPMN要素のラベル名は、「(1)ラベル命名ルール」、「(2)用語に関するルール」に従って付与すること。

BPMN要素のラベルの命名ルールを以下に示す。

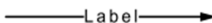
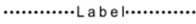
(1) ラベル命名ルール

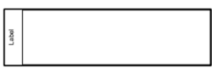
表 3.2-29 ラベル命名ルール

項番	図形名称	図形	命名条件	命名ルール	例
1	ユーザタスク, サービスタスク	 サービスタスク  ユーザタスク	なし	● フォーマット, 内容は以下のよう にすること。 「動作の対象を表す名詞」+「動作 を表す名詞」 ● 「動作の対象を表す名詞」に該当 するものがない場合は, 省略して もよい。 ● 同じプロセス図内に同じラベルが あってはいけない。³¹	書類起案
2	折りたたまれたサ ブプロセス, 展開されたサブ プロセス	 折りたたまれた サブプロセス  展開された サブプロセス	なし	● フォーマット, 内容は以下のよう にすること。 「動作の対象を表す名詞」+「動作 を表す名詞」 ● 「動作の対象を表す名詞」に該当 するものがない場合は, 省略して もよい。 ● 対応する折りたたまれたサブプロ セスと展開されたサブプロセスの ペアは, 同じラベルにすること。 それ以外は, 同じプロセス図内に 同じラベルがあってはけない。³¹	書類起案
3	折りたたまれたコ ールアクティビ ティ, 展開されたコール アクティビティ	 折りたたまれたコ ールアクティビ ティ  展開されたコ ール アクティビティ	なし	● フォーマット, 内容は以下のよう にすること。 「動作の対象を表す名詞」+「動作 を表す名詞」 ● 「動作の対象を表す名詞」に該当 するものがない場合は, 省略して もよい。 ● 対応する折りたたまれたコールア クティビティと展開されたコールア クティビティのペアで, 同じラベル にすること。 ● 異なるコールアクティビティは, 異 なるラベルにすること。	書類起案

³¹ 同じプロセス図内に同じアクティビティが2つ以上記載される場合でも, それぞれのアクティビティが実行される文脈に応じて異なるラベルを付与すること。サブプロセス等の他の図形要素で同じラベルとしない場合も同様とする。

項番	図形名称	図形	命名条件	命名ルール	例
4	折りたたまれたイベント・サブプロセス, 展開されたイベント・サブプロセス	 折りたたまれたイベント・サブプロセス  展開されたイベント・サブプロセス	なし	● フォーマット, 内容は以下のよう にすること。 「動作の対象を表す名詞」+「動作 を表す名詞」 ● 「動作の対象を表す名詞」に該当 するものがない場合は, 省略して もよい。 ● 対応する折りたたまれたイベント・ サブプロセスと展開されたイベン ト・サブプロセスのペアで, 同じラ ベルにすること。それ以外は, 同 じプロセス図内で同じラベルがあ ってはいけない。³¹	待ち合わせ 解除
5	開始イベント	 Label	なし	● ラベルは記載しない。	—
6	終了イベント	 Label	なし	● フォーマット, 内容は以下のよう にすること。 「終了状態を表す名詞」 ● 同じプロセス図内で終了イベント が複数ある場合, 同じラベルがあ ってはいけない。³¹	方式完
7	メッセージイベン ト (開始, キャッチ 中間)	 Label	なし	● フォーマット, 内容は以下のよう にすること。 「メッセージ名」+“受信” ● 対応するスロー中間, 終了メッ セージイベントと同じ「メッセージ 名」にすること。	審査開始通 知受信
8	メッセージイベン ト (スロー中間 終了)	 Label	なし	● フォーマット, 内容は以下のよう にすること。 「メッセージ名」+“送信” ● 対応する開始, キャッチ中間メッ セージイベントと同じ「メッセージ 名」にすること。	審査開始通 知送信
9	タイマーイベン ト (開始, キャッチ 中間)	 Label	なし	● 内容は以下のよう にすること。 「時刻, 時間間隔, 経過時間等ト リガとなる内容を表す名称」 ● 時間を表す場合は, 24時間表記 で記述すること。	毎週月曜2 0:00 5分毎 15分間 締切りの1日 前
10	リンクイベン ト (キャッチ, スロ ー)	 Label	なし	● フォーマット, 内容は以下のよう にすること。 「大文字アルファベット」 (アルファベット順に付与する。Zま で使ったら, その後はAA~AZ, BA ~BZ...と付与する。) ● 対応するスロー, キャッチのペア で同じラベル名にすること。	A

項番	図形名称	図形	命名条件	命名ルール	例
11	シグナル (開始, キャッチ 中間)	 Label  Label  Label	なし	● フォーマット, 内容は以下のよう にすること。 「シグナル名」+“受信” ● 対応するスロー中間, 終了シグナ ルイベントと同じ「シグナル名」に すること。	發送待機解 除通知受信
12	シグナル (スロー中間 終了)	 Label  Label	なし	● フォーマット, 内容は以下のよう にすること。 「シグナル名」+“送信” ● 対応する開始, キャッチ中間シグ ナルイベントと同じ「シグナル名」 にすること。	發送待機解 除通知送信
13	停止	 Label	なし	● フォーマット, 内容は以下のよう にすること。 「停止状態を表す名詞」 ● 同じプロセス図内で停止イベント が複数ある場合, 同じラベルがあ ってはいけない。³¹	中断
14	排他ゲートウェイ	 Label  Label	分岐に利 用する場 合	● フォーマット, 内容は以下のよう にすること。 「条件の対象名称」+“?”	書類ステー タス?
			結合に利 用する場 合	● ラベルは記載しない。	—
15	排他イベントゲー トウェイ	 Label	なし	● ラベルは記載しない。	—
16	並列ゲートウェイ	 Label	なし	● ラベルは記載しない。	—
17	シーケンスフロー	 Label	排他ゲー トウェイ の分岐 に利用 される場合	● 内容は以下のよう にすること。 「条件値の名称」 条件の対象名称が複数ある場合: 「条件の対象名称」+ “=” + 「条件値」	方式完
			その他	● ラベルは記載しない。	書類ステー タス=方式完
18	デフォルト・シー ケンスフロー	 Label	なし	● ラベルは記載しない。	—
19	関連	 Label	なし	● ラベルは記載しない。	—

項番	図形名称	図形	命名条件	命名ルール	例
20	プール		ホワイトボックスプールの場合	● フォーマット, 内容は以下のよう にすること。 「ビジネスプロセスを表す名称」+” プロセス”	実体審査プロセス
21	レーン		なし	● フォーマット, 内容は以下のよう にすること。 「ビジネスプロセス上の登場人物の 名称」(部門, 役職等)	審査官 指導審査官 審査長

(2) 用語に関するルール

- ラベルの名称に使用する用語は, インタフェース, データベースの命名と同様の用語集を使うこと。
用語集の詳細については, 「3.2.8.1 用語のルール」参照のこと。

3.2.3.2.3 ビジネスプロセス記載ルール

スコープ:	階層定型化サブシステム
プロセス:	2.3.2, 2.4.3
指針1:	ビジネスプロセスは、「(1)ビジネスプロセスの記載単位ルール」に示す単位で設計すること。
規約1:	ビジネスプロセスは、「(2)同じ意味であるが表記上異なるビジネスプロセスの記載統一ルール」の とおり記載すること。
指針2:	ビジネスプロセスのアクティビティは、「(3)アクティビティの単位のルール」に示す単位で設計する こと。
規約2:	BPMN要素は、「(4)レイアウトルール」の「A.時系列にあった配置ルール」、「C.BPMN要素の重なり 防止のルール」、「D.プールの配置ルール」、「E.イベント・サブプロセスの配置ルール」の とおり配 置すること。
指針3:	レーン配置順は、「(4) レイアウトルール B.レーンの配置ルール」に従い、登場人物の分類毎に 可読性を考慮した配置順序を決定し、各ビジネスプロセスで統一した配置順にすること。
推奨1:	役職のレーンを配置する場合、「下層の役職から上層の役職の順」に上から下へレーンを配置す ること。

BPMN要素を組み合わせることでビジネスプロセスを記載する上でのルールを以下に示す。

(1) ビジネスプロセスの記載単位ルール

可視性、保守性を考慮し、以下のようにビジネスプロセスを記載する。

A. 下表の観点でビジネスプロセスを独立させること。

制度改正や移行を考慮すると、ビジネスプロセスインスタンスのライフサイクルが短いことが好ましい。
したがって、業務の責務の切れ目においてなるべくビジネスプロセスが終了するようにし、後続のビジネ
スプロセスへはメッセージイベントで連携させる。

表 3.2-30 ビジネスプロセスの粒度

項 番	ビジネスプロセス の粒度	概要	備考
1	業務の責務	業務の責務の切れ目でプロセス を分割する(業務の責務は、業務 作業期限が存在したり、業務パフ ォーマンスを測定する単位となる 場合もある。)	※業務の責務の切れ目の例としては以下のと おり。 ● 起案／決裁／発送準備 等 ※業務の責務の切れ目における処理の特徴と しては、以下のとおり。 ● 共有データベースのイベントリソース等の データを更新した後 ※業務の責務で分割することで、プロセスの移 行の際に仕掛かり中の業務の移行を容易にす る効果を期待する。
2	ビジネスプロセス インスタンスの単 位	ビジネスプロセスインスタンスの 単位が異なる場合は、プロセスを 分割する。 ビジネスプロセスインスタンスの 単位が同じ場合は、同じ責務レ ベルの階層となるように階層化を 行う。階層化については後述す る。	※ビジネスプロセスインスタンスの単位の例は 以下のとおり。 ● 出願受付～方式審査は申請書類単位 ● 分類付与～実体審査は出願(事件)単位 ● 起案は起案種類単位
3	共通プロセス	複数のプロセスから利用される共 通的なプロセスは、共通プロセス として分割する。	—

B. 業務の責務レベルに応じてビジネスプロセスを階層化すること。

上位ビジネスプロセスと下位ビジネスプロセスの関係は、業務の責務を横断する上位ビジネスプロセスと1つの業務の責務内の下位ビジネスプロセスとなる。

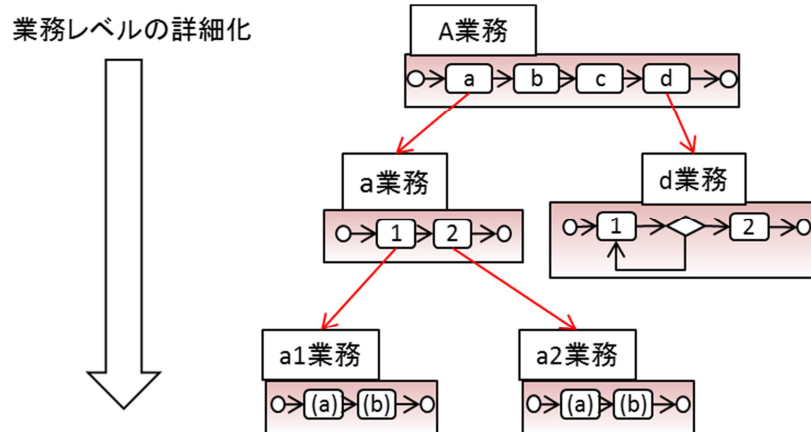


図 3.2-19 ビジネスプロセス階層化の概念図

C. 業務の責務が同じでもフローが大きく異なるビジネスプロセスは分けて定義すること。また、ビジネスプロセスの呼び分けは、ゲートウェイを用いず、BPMS補完機能で行うこと。

例えば、対象書類の出願日等によって適用される法律の新旧が変わり、それによってフローが大きく異なる場合は、ビジネスプロセスを分けて定義し、ビジネスプロセスの呼び分けはBPMS補完機能で行う。ビジネスルールに基づいた呼び分けの判断が必要な場合は、業務アプリケーション(システム)にアクセスし、その結果を用いて行う。これにより、ビジネスプロセス単位の追加や削除を可能とし、新たな法律への対応や、不要になった旧法に基づくビジネスプロセスの破棄を容易にする。

D. ビジネスプロセスの可視性、保守性を考慮して、適宜サブプロセス化すること。

大きすぎるビジネスプロセスは可視性、保守性が低下するからである。

以下のような観点でサブプロセス化すること。

- 同じ業務の責務内で分類ができる単位
- 業務色の薄いフロー制御

例えば、下図のようなビジネスプロセス連携時のメッセージ送受信のフロー等、業務色の薄い手続きにあたるフローをサブプロセス化することにより、業務色の薄いフローを隠蔽し、ビジネスプロセスの可視性、保守性が向上する。

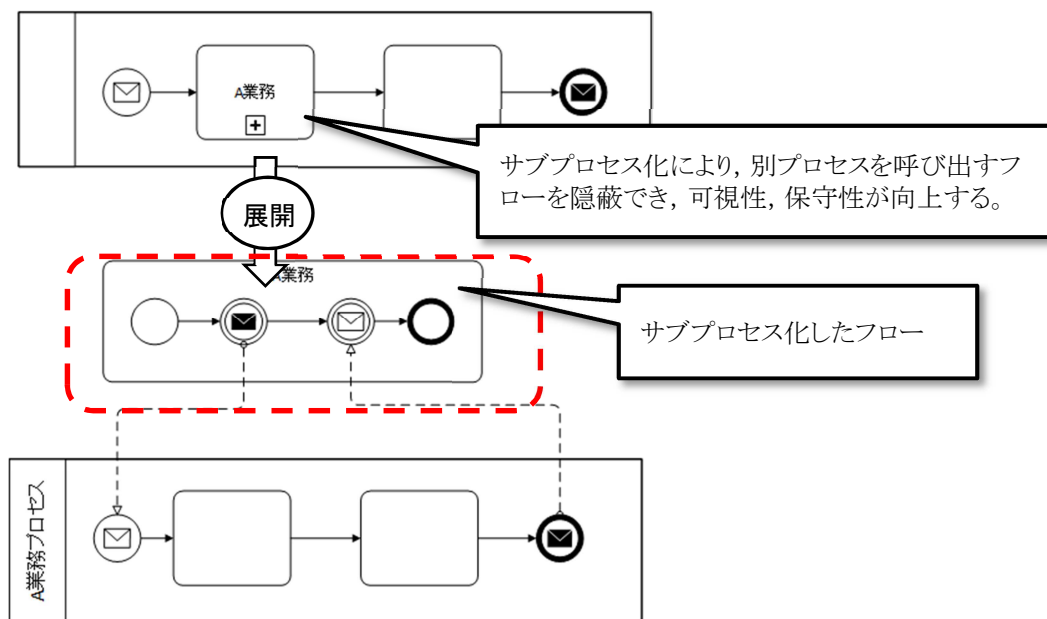


図 3.2-20 ビジネスプロセスのサブプロセス化の例

(2) 同じ意味であるが表記上異なるビジネスプロセスの記載統一ルール
 BPMNの仕様上、複数の記載方法が許されているものに対する統一的な記載ルールを以下に示す。

- A. 排他的分岐経路の合流における排他ゲートウェイの記述
- 排他ゲートウェイで分岐された経路を合流する際、排他ゲートウェイを使用する方法と使用しない方法の2通りの記載方法がある。余分なBPMN要素の記載を避け、シンプルに表現するため、以下のルールとする。
- 排他ゲートウェイで分岐された経路を合流する際、排他ゲートウェイを使用せず、シーケンスフローを直接アクティビティに接続する。
- ただし、合流後、別のゲートウェイに接続する場合は、本ルールは適用されない。必ず、合流の排他ゲートウェイを記述すること。

○ 良い例	合流時に排他ゲートウェイを使用しない。
✕ 悪い例	合流時に排他ゲートウェイを使用する。

図 3.2-21 排他的分岐経路の合流における排他ゲートウェイの記述

- B. 並列処理の分岐における並列ゲートウェイの記載
- 並行処理の分岐には、並列ゲートウェイを使用する方法と使用しない方法の2通りの記載方法がある。余分なBPMN要素の記載を避け、シンプルに表現するため、以下のルールとする。
- 並列処理の分岐には、並列ゲートウェイを使用しない。

○ 良い例	並列処理の分岐に、並列ゲートウェイを使用しない。
✕ 悪い例	並列処理の分岐に、並列ゲートウェイを使用する。

図 3.2-22 並列処理の分岐における並列ゲートウェイの記載

C. 終了状態が複数ある場合の終了イベントの記載

終了状態が複数ある場合は、1つの終了イベントに集約する方法と別々にする方法の2通りの記載方法がある。ビジネスプロセスの終了状態が明確になり、可読性が向上するため、以下のルールとする。

- 終了状態が複数ある場合は、別々の終了イベントで記載すること。また、終了イベントのラベルには、それぞれの終了状態を表す名称を記載すること。

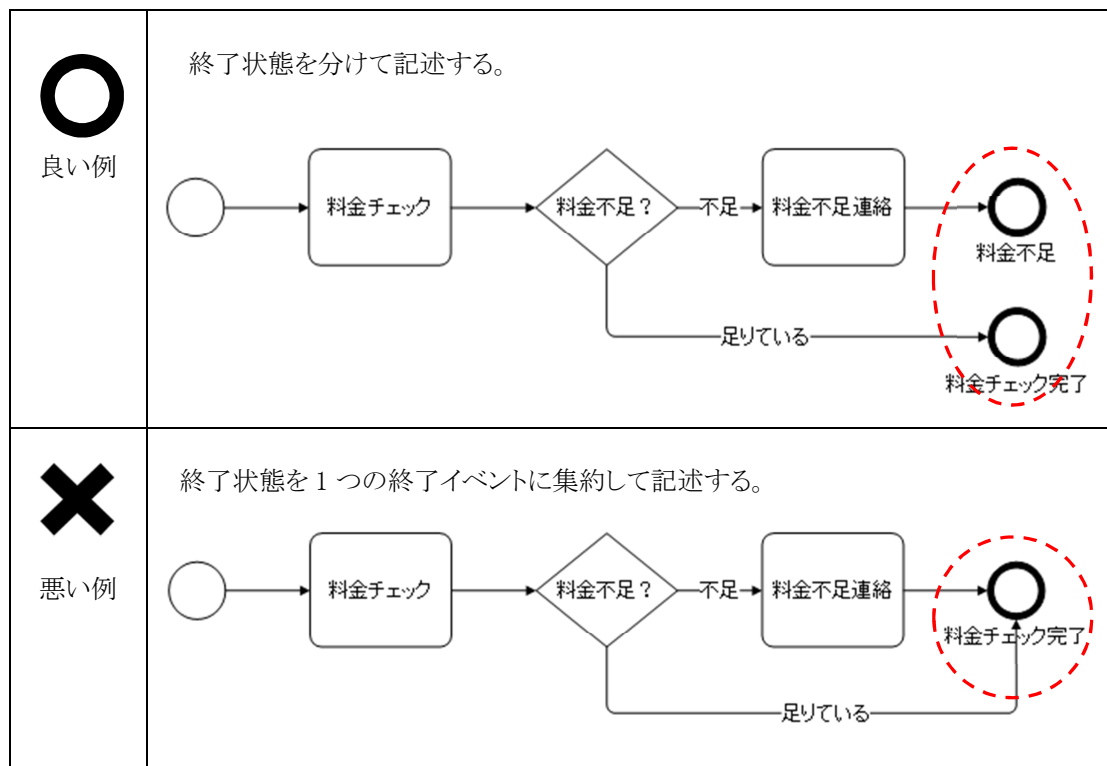


図 3.2-23 終了状態が複数ある場合の終了イベントの記載

(3) アクティビティの単位のルール

BPMNのアクティビティの単位についてのルールを以下に示す。

- ビジネスプロセスのアクティビティの単位は、業務として意味のある最小の単位とすること。
- 業務として意味のある単位とは、何らかの仕事が完了し、仕掛中の状態ではなく他の担当者に受け渡すことのできるような単位であり、ビジネスプロセスの流れや進行状況を可視化する上でも意味のある単位のことである。
- なお、システムタスクの単位を設計するにあたっては、対応するサービスインタフェースの単位と同一であるため、「3.2.4.2サービスインタフェースの粒度」を参照すること。

(4) レイアウトルール

BPMN要素の配置についてのルールを以下に示す。

A. 時系列にあった配置ルール

- BPMN要素の配置は、時系列に合わせて左から右へ記述すること(「図 3.2-24 時系列にあった配置例1」の①)。
※差戻し等ビジネスプロセスの前工程に戻る場合を除く。
※「3.2.3.2.4 ビジネスプロセスパターン事例」でパターンとして標準化されているものに関しては、パターンの記述方法に従うこと。
- アクティビティへ入力するシーケンスフローは、基本、左側中央(付近)に接続すること(「図 3.2-24 時系列にあった配置例1」の②)。
ただし、可読性に影響がある場合は、上部中央(付近)又は、下部中央(付近)に接続すること。
- アクティビティから出力するシーケンスフローは、基本、右側中央(付近)に接続すること(「図 3.2-24 時系列にあった配置例1」の③)。

- シーケンスフローを曲げるのは最大2箇所にとどめること（「図 3.2-25 時系列にあった配置例2」の④）。

※差戻し等ビジネスプロセスの前工程に戻る場合を除く。

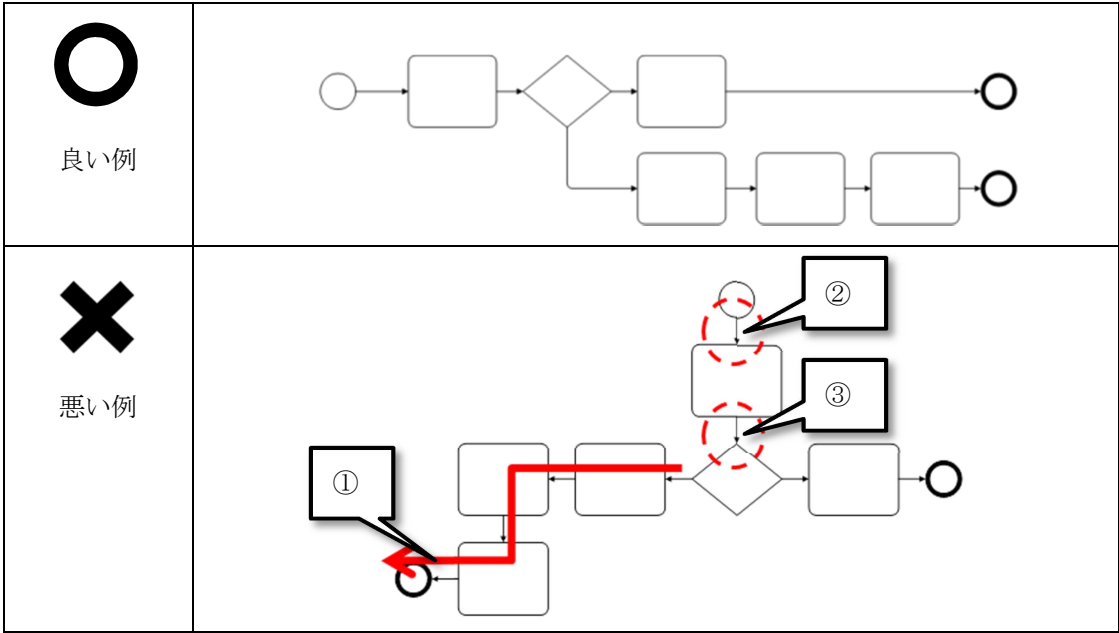


図 3.2-24 時系列にあった配置例1

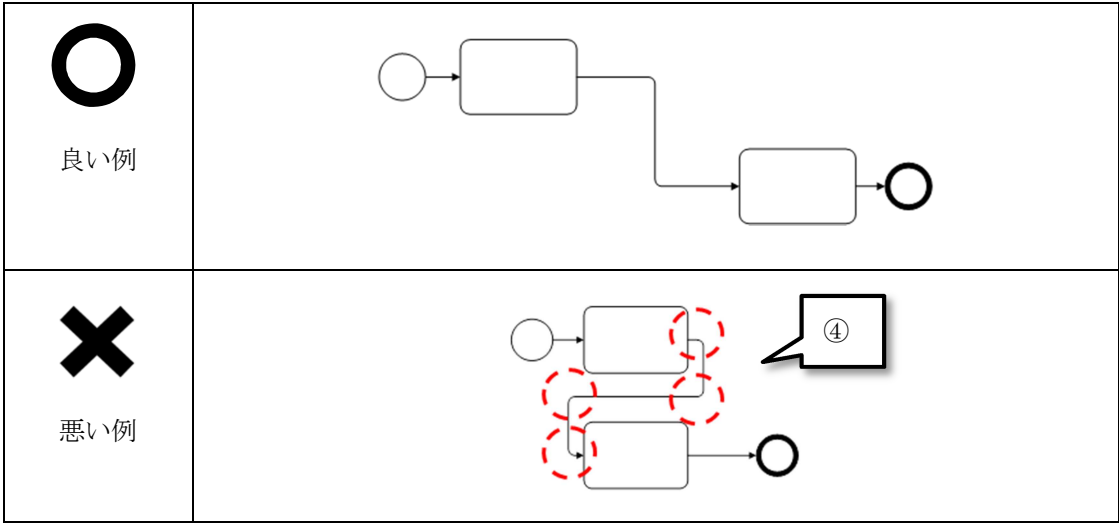


図 3.2-25 時系列にあった配置例2

B. レーンの配置ルール

登場人物を表すレーンのルールを以下に示す。

- 登場人物の分類毎に可読性を考慮した配置順序を決定し、各ビジネスプロセスで統一したレーン配置順にすること。ただし、複数レーンを縦断するようなシーケンスフローが多くなり、可読性が悪くなる場合は、配置順を変更することを許容する。

配置順序の例として、役職の場合、「下層の役職から上層の役職の順」に上から下へレーンを配置する。これは、承認フローを考慮すると、ビジネスプロセス進行順にレーンの上から下へBPMN要素配置でき、可読性の向上が見込めるためである。

システムを表すレーンのルールを以下に示す。

- システム用のレーンを設けないこと。
- サービスタスクは、以下のルールに従って、レーンに配置すること。
 - 直前に配置されているタスク、イベントと同じレーンに配置する。
 - サービスタスクの業務内容が、上記ルールで配置したレーンにふさわしくない場合は、ふさわしいレーンに配置する。

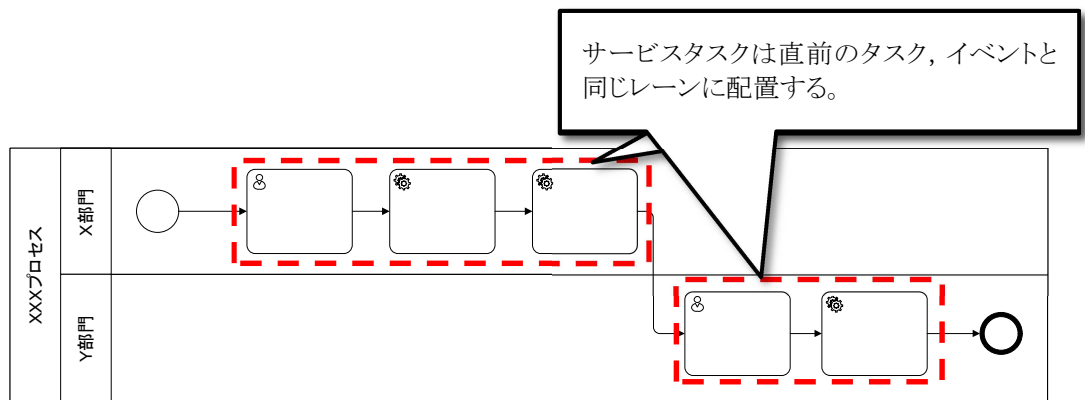


図 3.2-26 サービスタスクのレーンの配置例

C. BPMN要素の重なり防止のルール

- BPMN要素が重ならないように配置すること。

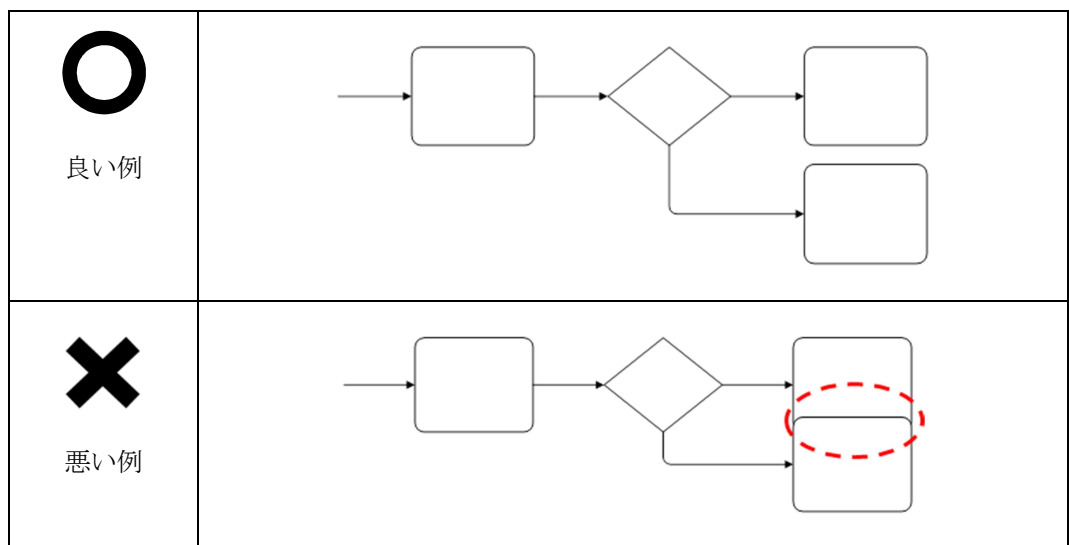


図 3.2-27 BPMN要素の重なり防止のルール1

- シーケンスフローがBPMN要素の上を横切らないように配置すること。

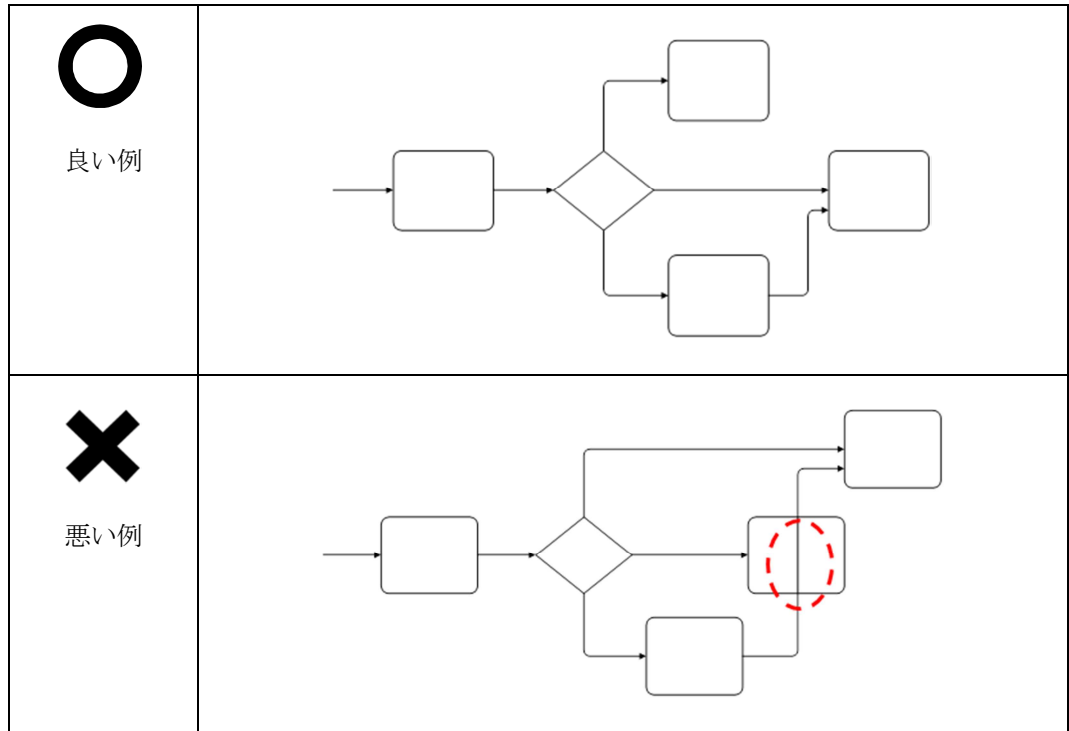


図 3.2-28 BPMN要素の重なり防止のルール2

- シーケンスフローの交差は、可読性が低下しないよう最小限にすること。

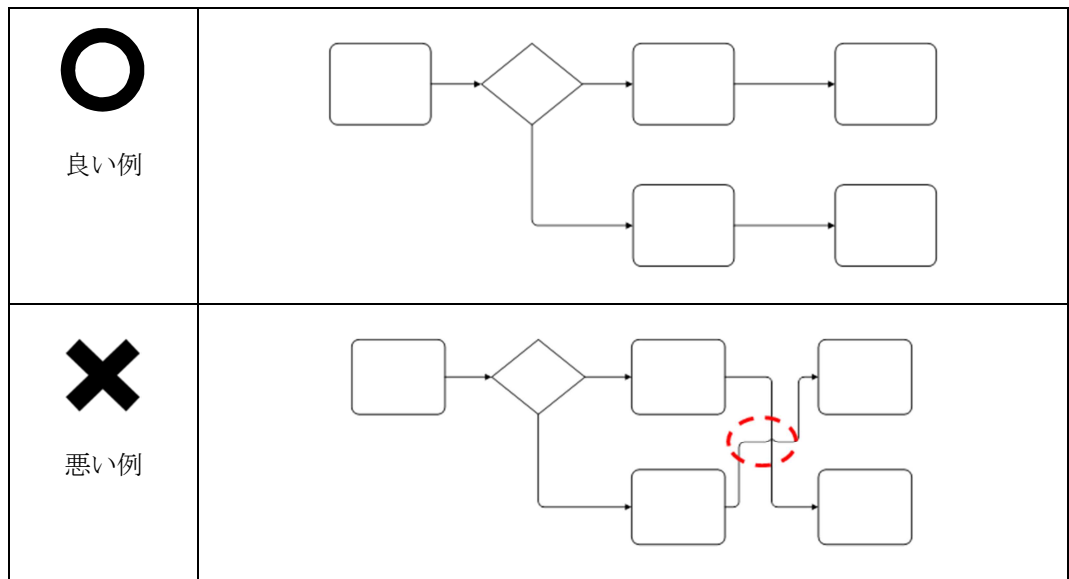


図 3.2-29 BPMN要素の重なり防止のルール3

- シーケンスフローをゲートウェイに接続する場合、シーケンスフローが重ならないよう、ゲートウェイの同じ頂点に複数接続しないこと。
ただし、ゲートウェイの全ての頂点にシーケンスフローが接続されている場合は、同じ頂点に複数接続することを許容する。

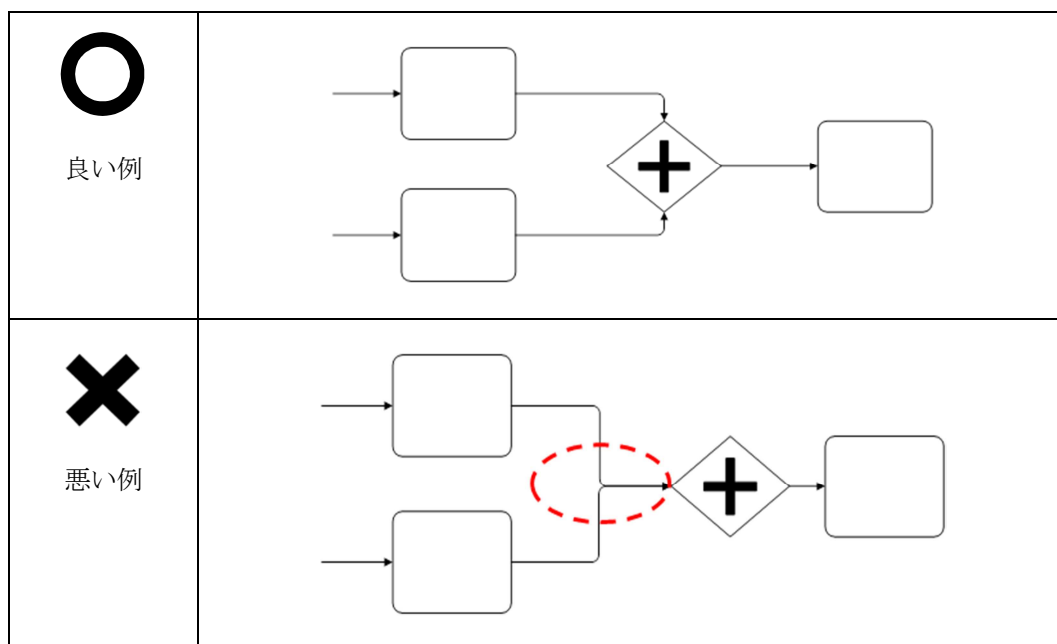


図 3.2-30 BPMN要素の重なり防止のルール4

- 1つのBPMN要素に複数のシーケンスフローが接続される場合、シーケンスフローが重ならないように、基本、BPMN要素の同じ場所で接続しないこと。

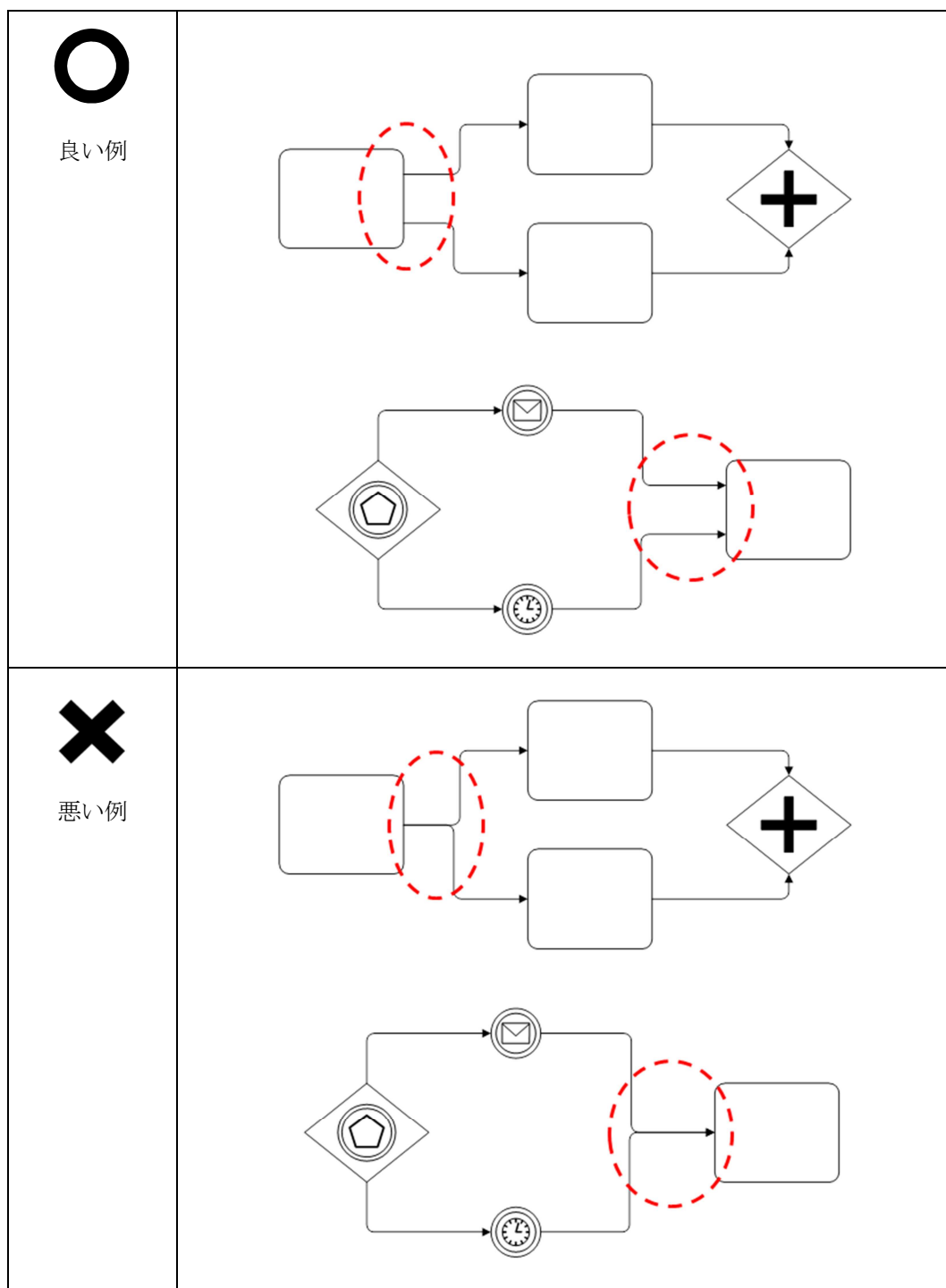


図 3.2-31 BPMN要素の重なり防止のルール5

D. プールの配置ルール

- プールは横向きに配置すること。

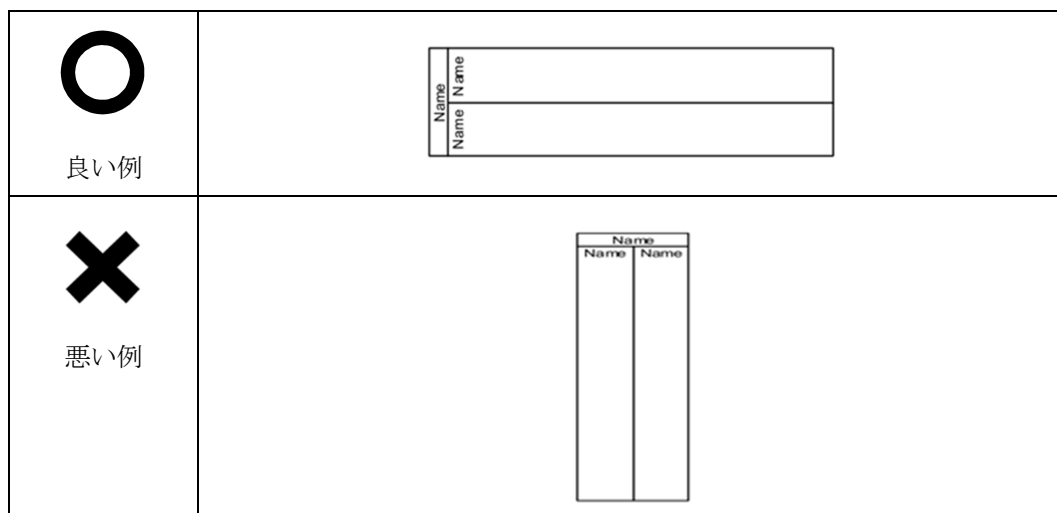


図 3.2-32 プールの配置ルール

E. イベント・サブプロセスの配置ルール

- イベント・サブプロセスは親ビジネスプロセスの下に配置すること。

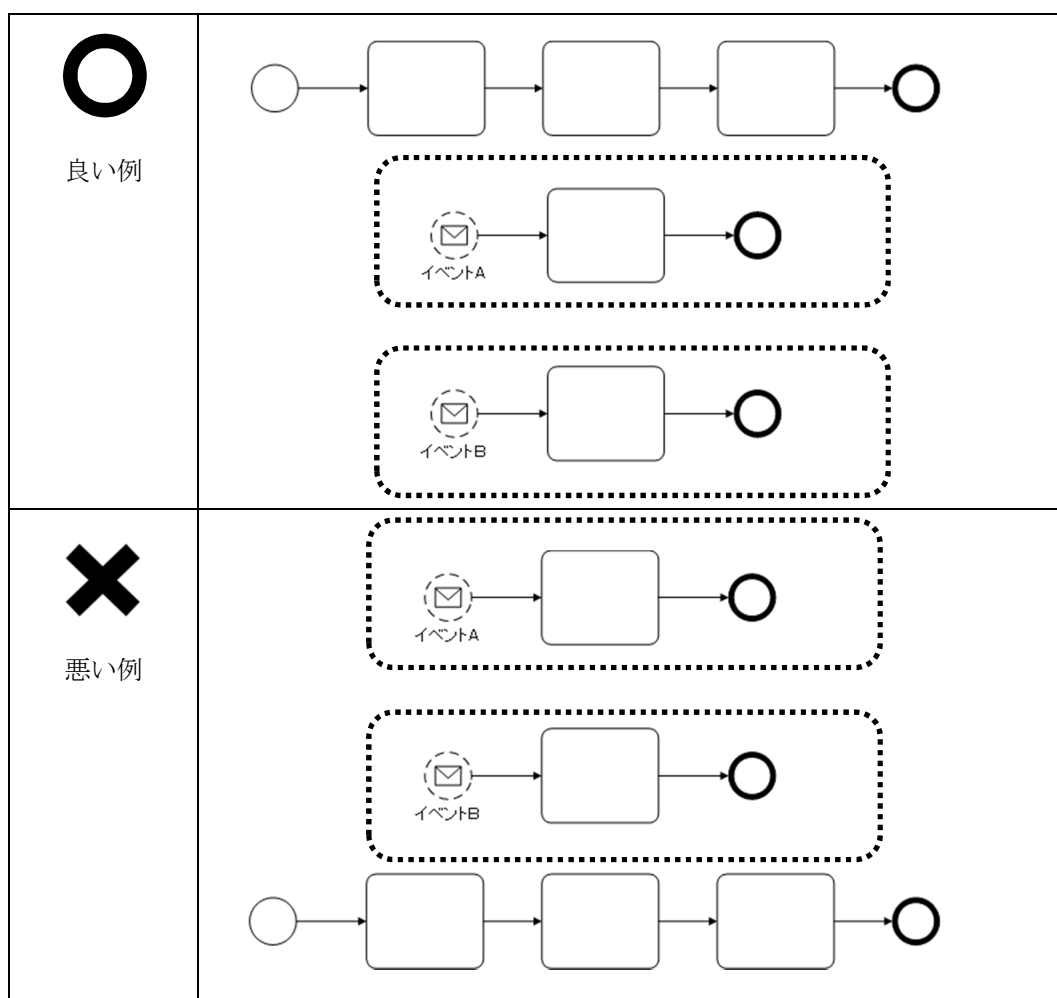


図 3.2-33 イベント・サブプロセスの配置ルール

3.2.3.2.4 ビジネスプロセスパターン事例

スコープ：階層定型化サブシステム

プロセス：2.3.2, 2.4.3

規約1: ビジネスプロセスは、ビジネスプロセス設計時に「表 3.2-31 ビジネスプロセスパターン事例一覧」に示す適用シーンに合致した場合、当該パターン事例を適用すること。

業務でよく出現するビジネスプロセスをパターン事例として定義する。ビジネスプロセス設計時に適用シーンに合致したものがあった場合、パターン事例を適用すること。

パターン事例を下表に示す。

表 3.2-31 ビジネスプロセスパターン事例一覧

項番	パターン事例名	適用シーン	備考
1	他ビジネスプロセス起因の待機処理	予見できないタイミングで他のビジネスプロセスインスタンスから待機を要求される場合	業務例： 自発補正

(1) パターン事例1 他ビジネスプロセス起因の待機処理

A. 適用シーン

予見できないタイミングで他のビジネスプロセスインスタンスから待機を要求される場合。

B. 適用条件

- 待機要求を発生させる他のビジネスプロセスインスタンスが、データベースに管理されているステータスを“待機を意味するステータス”に更新すること(本パターン事例は、データベースのステータスを取得することで、待機が必要かを判定する。)
- 他のビジネスプロセスインスタンスから待機解除をメッセージで通知すること。

C. プロセス図

可読性を考慮し、本パターン事例は、サブプロセス又は、コールアクティビティ内に表記する。

※下図の“○○”には、適用するビジネスプロセスにあった業務名称が入る。

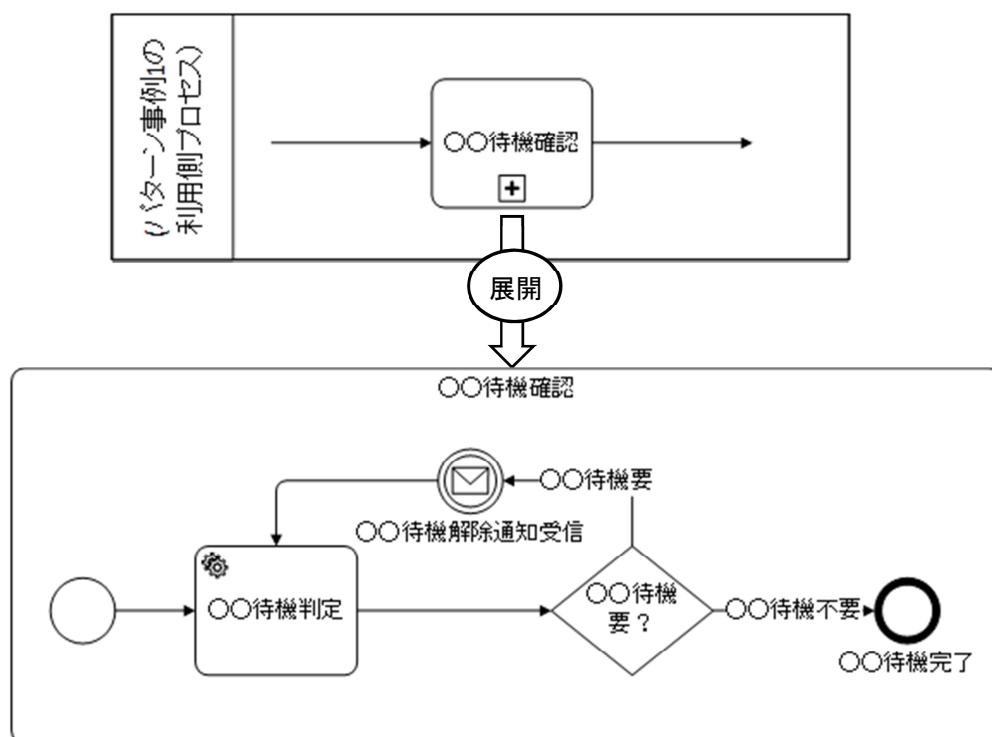


図 3.2-34 パターン事例1 プロセス図

D. 主な構成要素と適用ルール

① サブプロセス又はコールアクティビティ

表 3.2-32 パターン事例1 主な構成要素と適用ルール
(サブプロセス又はコールアクティビティ)

項番	ラベル名	主なBPMN要素	説明	適用ルール
1	ー	開始イベント	サブプロセスを開始する。	ー
2	〇〇待機判定	サービスタスク	サービス呼び出しを行い、待機を要求しているかを判定する。	ラベル名の“〇〇”は、命名ルールに従い、適切な名称を付与すること。
3	〇〇待機要？	排他ゲートウェイ	「〇〇待機判定」タスクの戻り値から待機が必要かを判定し、以下のように分岐する。 ● 必要(Yes)の場合、「〇〇待機解除通知受信」へ。 ● 必要ない(No)場合、終了イベントへ。	ラベル名の“〇〇”は、命名ルールに従い、適切な名称を付与すること。
4	〇〇待機解除通知受信	キャッチ中間メッセージイベント	「〇〇待機解除通知」を受信し、「〇〇待機確認」タスクへ遷移する。	ラベル名の“〇〇”は、命名ルールに従い、適切な名称を付与すること。
5	〇〇待機完了	終了イベント	サブプロセスを終了する。	ラベル名の“〇〇”は、命名ルールに従い、適切な名称を付与すること。

② 利用側ビジネスプロセス

表 3.2-33 パターン事例1 主な構成要素と適用ルール
(利用側ビジネスプロセス)

項番	ラベル名	主なBPMN要素	説明	適用ルール
1	〇〇待機確認	折りたたまれたサブプロセス又は、コールアクティビティ	本パターン事例のビジネスプロセスを呼び出す。	ラベル名の“〇〇”は、命名ルールに従い、適切な名称を付与すること。

E. 適用例

方式審査システムでの結果に応じた実体審査システムの審査待機処理の例を示す。

方式審査システムの結果は、データベースに審査待機を表すステータスが格納されているものとする。

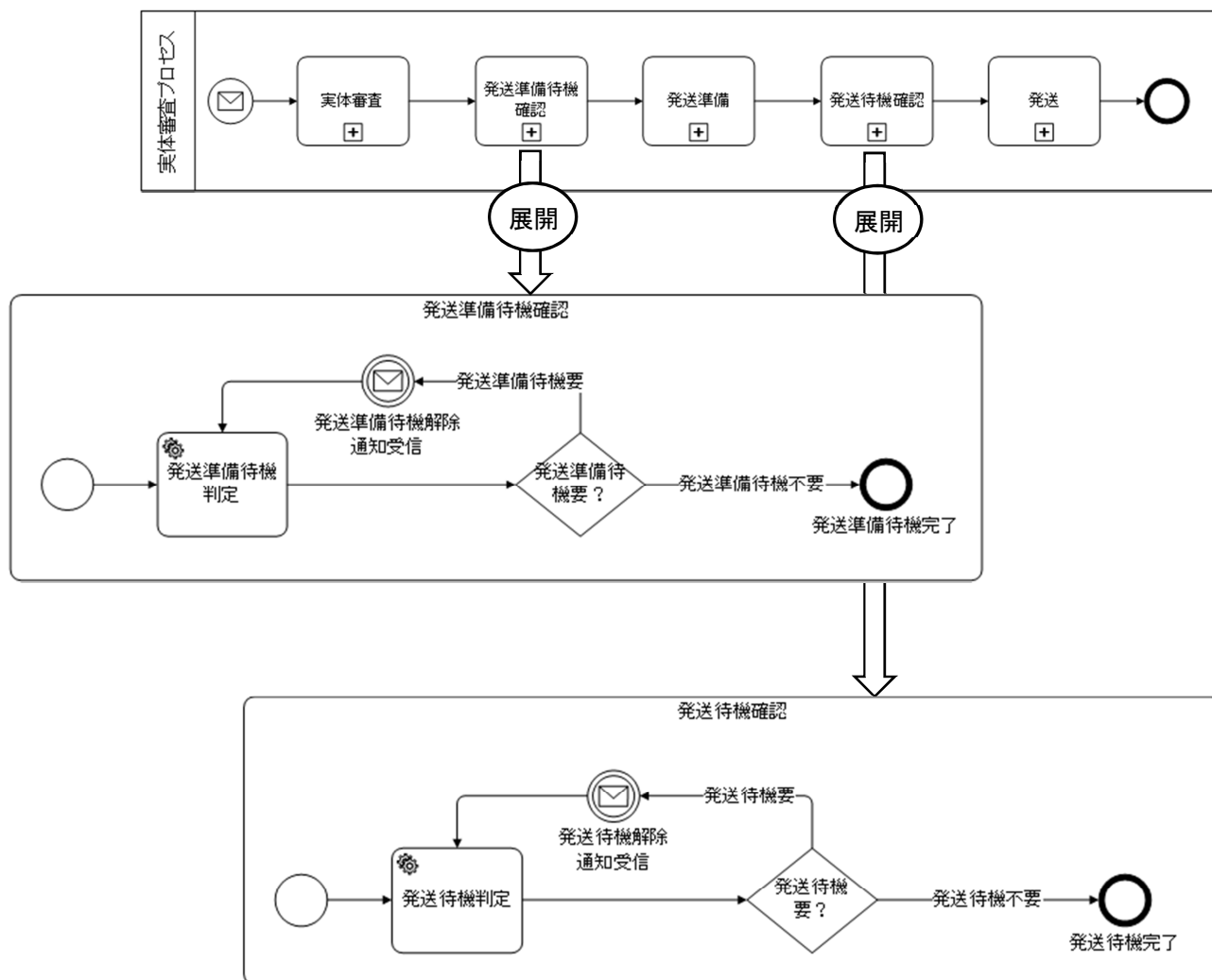


図 3.2-35 パターン事例1 適用例

F. 補足

「E.適用例」にあったように、ビジネスプロセス中に複数の待機箇所がある場合、シグナルを利用することにより待機解除通知受信のイベントを1つにすることができる。

具体的な方式は、以下のとおりである。

- ① 待機解除通知受信のメッセージ中間キャッチイベントをシグナル中間キャッチイベントに変更する。
- ② 待機解除通知メッセージで駆動するイベント・サブプロセスを設ける。
- ③ イベント・サブプロセス内では、スロー中間シグナルイベントを使って、キャッチ中間シグナルイベントへブロードキャストする。

「E.適用例」をこの方式にするとプロセス図は以下のとおりである。

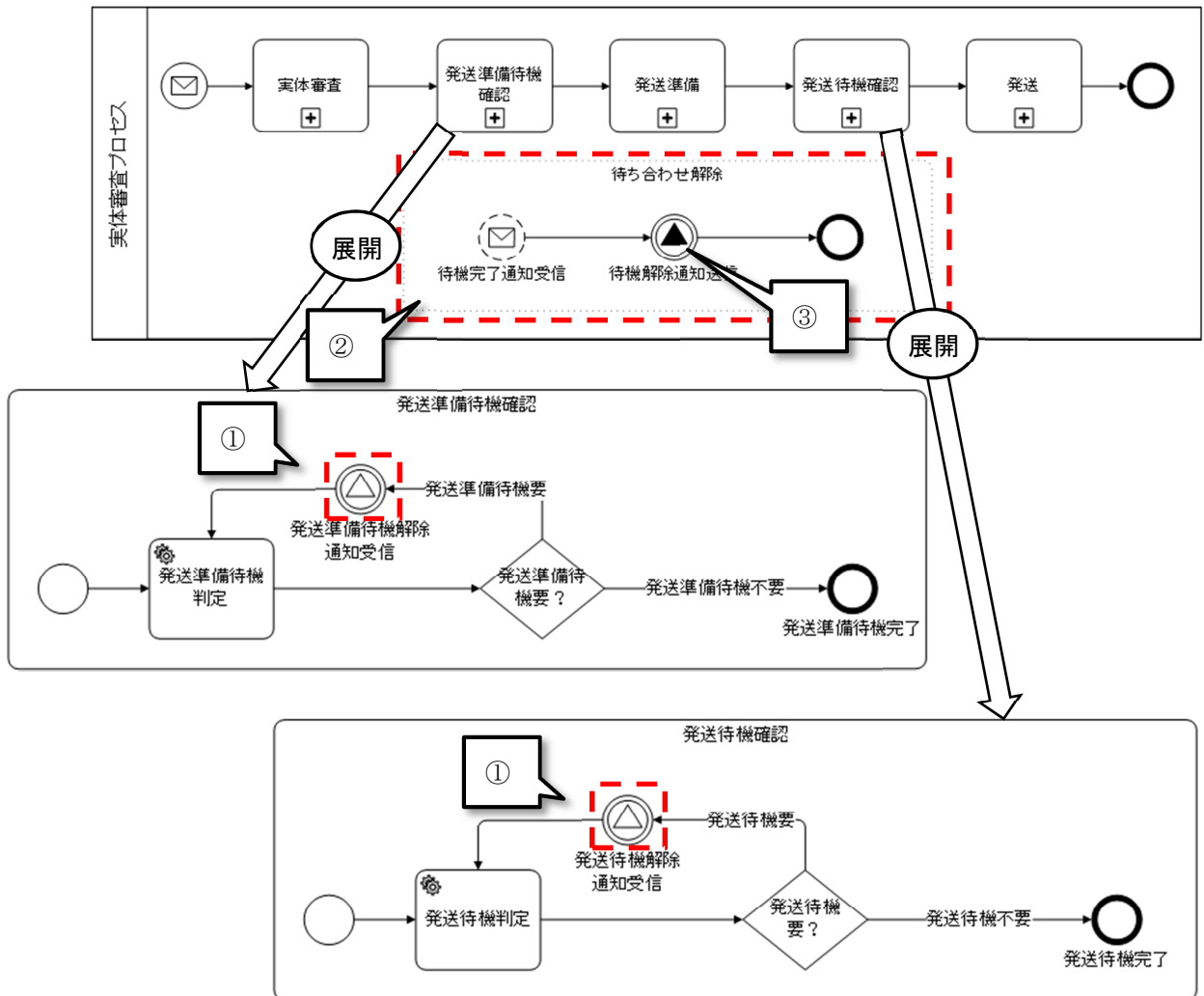


図 3.2-36 パターン事例1 適用例(シグナル利用)

3.2.3.3 BPMS補完機能のルール

3.2.3.3.1 BPMS補完機能で実現する機能

スコープ：階層定型化サブシステム

プロセス：2.4.3

規約1： BPMS補完機能で実現する機能は、「(1)BPMS補完機能で実現する機能のルール」に示す機能とすること。

連携するビジネスプロセスインスタンスを業務的な条件により動的に判断・特定する等、本来BPMSにて実現されるべきものであるもののBPMSの標準機能だけでは実現することが難しく、別途アプリケーションを介する必要がある場合がある。このような連携処理を実現するためにBPMSを補完する処理を行うアプリケーションがBPMS補完機能であり、ワークフロー層に配置される。

BPMS補完機能は、様々な補完処理に対応するため、補完が必要な対象毎に作成する。また、補完処理はビジネスプロセスの定義内容に強く依存したものとなるため、ビジネスプロセスの変更に伴い、対応するBPMS補完機能の修正が必要になる可能性が高い。

(1) BPMS補完機能で実現する機能のルール

BPMS補完機能が実現する機能のルールを以下に示す。

- BPMS単体で実現できない機能に限り、BPMSと連携し補完する処理を実装すること。
- 業務アプリケーション層の責務となる業務ロジックは持たせず、ワークフロー層の責務として、起動するビジネスプロセス又は連携するビジネスプロセスインスタンスの特定やイベント通知の要否判定といった制御を行うこと。

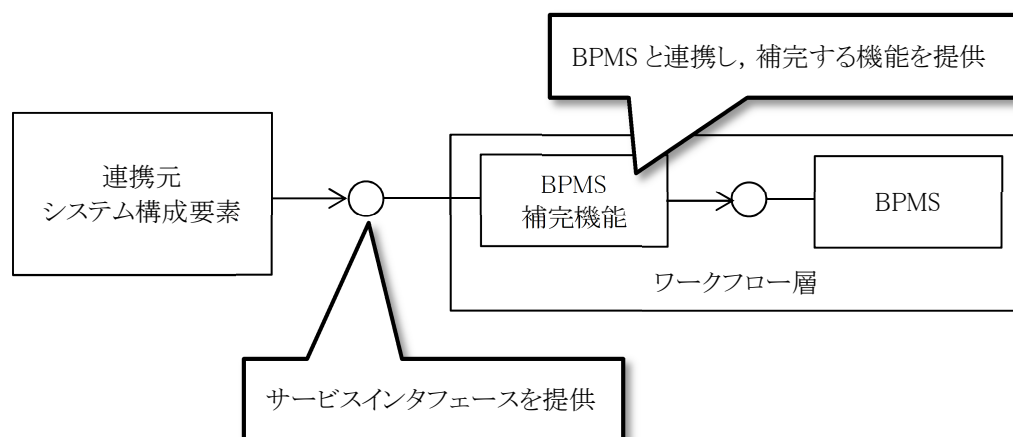


図 3.2-37 BPMS補完機能が実現する機能³²

³² 図中の「連携元システム構成要素」の詳細は、「3.1.1.3.1.2 システム構成要素間のアクセスパス」を参照のこと。

3.2.3.3.2 BPMS補完機能が提供する補完処理例

以下にBPMS補完機能が提供する補完処理の例を示す。

- (1) 起動するビジネスプロセスやイベント通知するビジネスプロセスインスタンス及びビジネスプロセス上の対象イベントを業務的な条件で動的に特定する処理

BPMS補完機能は、連携元からのパラメータ又はデータベースの格納データから、連携対象のビジネスプロセスやビジネスプロセスインスタンス及びビジネスプロセス上の対象イベントを特定し、ビジネスプロセス起動やビジネスプロセスインスタンスへイベント通知を行う。

例えば、自発補正の方式審査完了に伴う実体審査プロセスへの中断解除通知³³は、中断解除通知するビジネスプロセスインスタンスが静的に特定できない。このようなときにBPMS補完機能にて中断解除通知をするべきビジネスプロセスインスタンス及びビジネスプロセス上の対象イベントを特定する処理を実装する。更に、特定した単一又は複数のビジネスプロセスインスタンスに中断解除通知を行う。

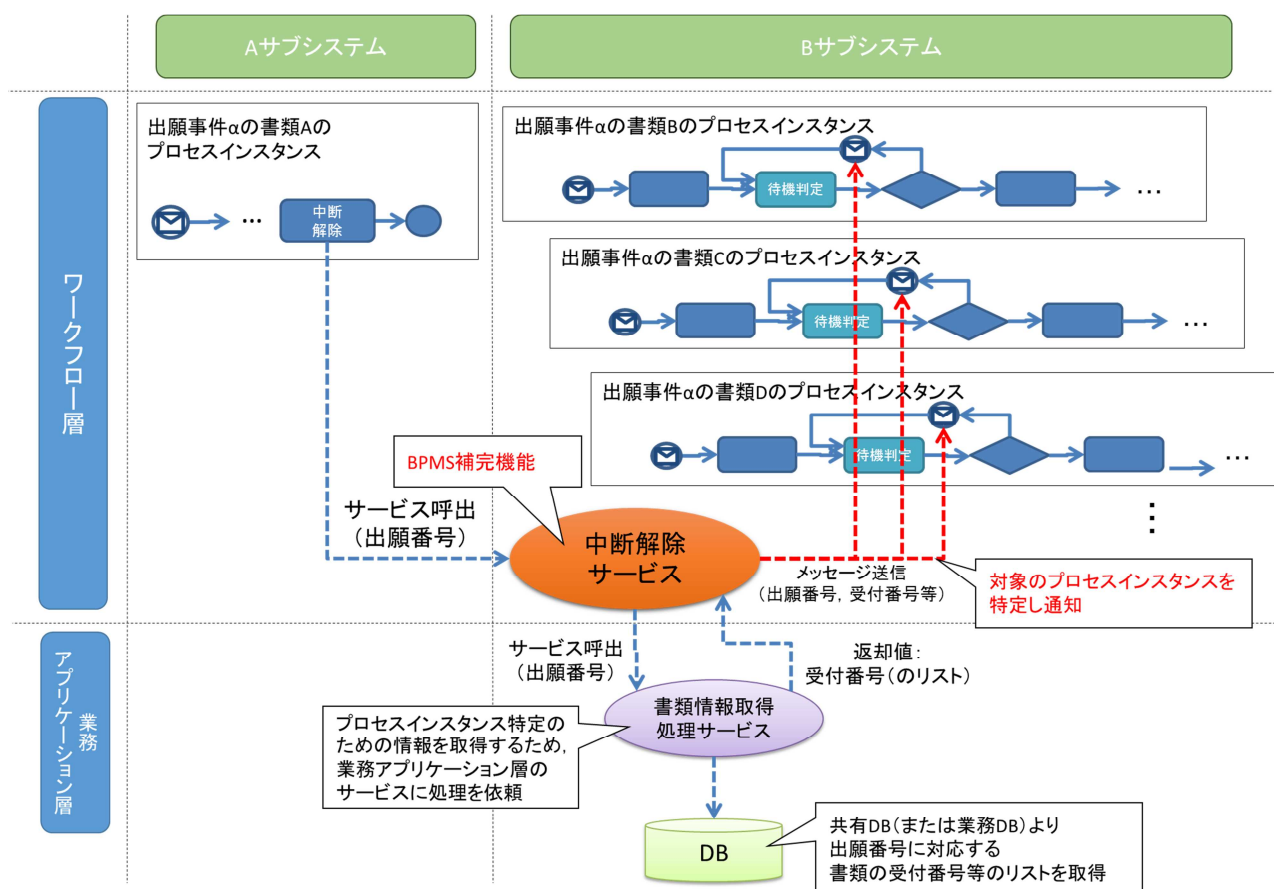


図 3.2-38 BPMS補完機能が提供する補完処理例1

³³ 本処理の詳細については、「3.2.3.1.2 BPMSの適用範囲に関する設計指針」の「(1)他のビジネスプロセスとの密接な順序関係を持つビジネスプロセスの設計指針」を参照のこと。

(2) 並行運用しているビジネスプロセスの対象選定処理

制度改正により制度改正前と改正後のビジネスプロセス両方を並行運用する必要がある場合においても、BPMS補完機能を利用して並行運用する複数のビジネスプロセスから起動させるものを選択する。

これにより、ビジネスプロセス内に条件分岐のためのサービスタスク及びゲートウェイを追加したり、ビジネスプロセス内に改正前後のものを併記するの必要がなくなるため、ビジネスプロセスの可視性を向上させるとともに、改正前のビジネスプロセスが不要になった際に、ビジネスプロセスの修正を行うことなく破棄することが可能となる。³⁴

(3) BPMS連携時の事前又は事後処理

例えば、イベント待ちの状態でない場合のイベント通知は、製品によっては、挙動が異なる。イベント待ち状態か否かの判定等、製品によって事前の処理が必要な場合がある。このような場合、BPMS補完機能にて、補完処理を実装する。³⁵

³⁴ 本処理の事例として、「3.2.3.2.3 ビジネスプロセス記載ルール」の「(1)ビジネスプロセスの記載単位ルール C.」を参照のこと。

³⁵ 本処理の事例として、「3.2.3.1.2 BPMSの適用範囲に関する設計指針」の「(3)BPMNの記述上想定外のイベントが発生するケースの設計指針」を参照のこと。

3.2.3.3.3 配置位置

スコープ：階層定型化サブシステム

プロセス：2.4.3

規約1： BPMS補完機能の配置は、「(1)BPMS補完機能の配置位置のルール」に示すとおり配置すること。

(1) BPMS補完機能の配置位置のルール

BPMS補完機能の配置位置のルールを以下に示す。

- 補完対象のBPMSサービスを提供するサブシステムに配置すること。
利用側のサブシステムからBPMS補完機能が実現する処理を隠蔽し、サブシステム間のインタフェース変更影響の局所化をするためである。

3.2.4 業務アプリケーション層のルール

(1) 責務

業務アプリケーション層の責務は、サブシステムが担う「業務処理の実行」である。業務アプリケーション層の責務を担うための主な処理内容を以下に示す。

- プレゼンテーション層、ワークフロー層又は外部システム連携層からリクエストを受け取り、業務処理を実行する。
- 業務処理にてデータ処理を行う場合、基盤機能層又はデータベース層にアクセスし、データ取得及び更新を行う。
- 業務処理にてビジネスルールの判定を行う場合、BRMSにアクセスし、判定結果を取得する。
- 業務処理にて外部システムへ連携する場合、外部システム連携層にアクセスし、連携を行う。
- 業務処理にてワークフローの実行又は状態取得等を行う場合、ワークフロー層にアクセスする。

(2) システム構成要素

業務アプリケーション層は、「業務処理の実行」を行うにあたり、処理方式³⁶の特性に合わせた3つのアプリケーションとデータベースのシステム構成要素で構成される。業務アプリケーション層を構成するシステム構成要素を以下に示す。

- 業務アプリケーション(ユーザ)
- 業務アプリケーション(システム)
- 業務アプリケーション(バッチ)
- 個別データベース

A. 業務アプリケーション(ユーザ)

- 責務は「ユーザからの入力情報に基づく業務処理又はユーザが必要とする情報の取得」である。
- 画面とのインタラクティブなやり取りに特化した業務処理を行う(例えば、情報の検索や閲覧等)。
- プレゼンテーションロジックからメソッドアクセスされることにより、業務処理を実行する。
- 以下に列挙する場合に限り、他システム構成要素にアクセスしてもよい。
 - データ取得又は更新のために、DBアクセス基盤サービス、共有データベース、自サブシステムの個別データベースにアクセスする。
 - 外部システム連携のためにESBへアクセスする。
 - ビジネスルールの判定を得るためにBRMSへアクセスする。
 - 共通リソースデータの取得又は更新のために自サブシステム又は他サブシステムの業務アプリケーション(システム)にアクセスする。

B. 業務アプリケーション(システム)

- 責務は、以下に示す責務のいずれかである。
 - BPMSのサービスタスクに対応し業務として意味のある一連の処理の実行。
 - 他のシステム構成要素で必要とするデータの取得又は更新。
- 上記の責務を実行するサービスを提供する。
- ステートレスなサービスインタフェースを提供する(セッションを使用しない。)
- プレゼンテーションロジック、BPMS、BPMS補完機能、ESB、業務アプリケーション層の3つ業務アプリケーションからサービスインタフェース経由でアクセスされることにより、業務処理を実行する。
- 以下に列挙する場合に限り、他システム構成要素にアクセスしてもよい。
 - データ取得又は更新のために、DBアクセス基盤サービス、共有データベース、自サブシステムの個別データベースにアクセスする。
 - 外部システム連携のためにESBへアクセスする。
 - ビジネスルールの判定を得るためにBRMSへアクセスする。
 - 業務アプリケーション(システム)のビジネスロジックで、別のビジネスプロセスインスタンスのステータス情報に基づいた処理を行うためのステータス取得のために自サブシステム又は他サブシステムのBPMS又はBPMS補完機能へアクセスする。
 - 共通リソースデータの取得又は更新のために自サブシステム又は他サブシステムの業務アプリケーション

³⁶ 「3.1.1.3.1.1 システム構成要素の処理方式による分類」を参照。

ション(システム)にアクセスする。

C. 業務アプリケーション(バッチ)

- 責務は「BPMSのサービスタスクに対応せず所定のタイミング(期間, 時間又はディレード処理要求の発生等)で処理の実行」を行うことである。
- バッチジョブに特化した業務処理を行う。
- ジョブ管理マネージャからのアクセス又はディレード処理要求の登録が確認されることにより業務処理を実行する。
- 以下に列挙する場合に限り, 他システム構成要素にアクセスしてもよい。
 - データ更新, 取得のために, DBアクセス基盤サービス, 共有データベース, 自サブシステムの個別データベースにアクセスする。
 - 外部システム連携のためにESBへアクセスする。
 - ビジネスルールの判定を得るためにBRMSへアクセスする。
 - 他サブシステムへの連携のためにFTPで業務システム(バッチ)へアクセスする(データベース経由では, 効率が極めて悪い他サブシステム連携に限る。)
 - 人事異動のためビジネスプロセスインスタンスに割り当てられている担当者をバッチにて一括で変更する等の目的でアクセスするために, 又はバッチからビジネスプロセスを開始するために自サブシステム又は他サブシステムのBPMS又はBPMS補完機能へアクセスする。
 - 共通リソースデータの取得又は更新のために自サブシステム又は他サブシステムの業務アプリケーション(システム)にアクセスする。

D. 個別データベース

- 責務は「サブシステム固有の業務データの管理」である。
- 管理するデータは, サブシステムの固有データ(個別リソースデータ及び個別業務イベントデータ)と共通リソースデータである。
- 自サブシステムの業務アプリケーション層の3つ業務アプリケーションからデータベースがサポートするプロトコルでアクセスされる。

3.2.4.1 サービスの粒度

スコープ： 階層定型化サブシステム

プロセス： 2.4.2, 2.4.3

指針1： サービスの粒度は、「(1)サービスの粒度のルール」に従って設計すること。

業務アプリケーション層のシステム構成要素のうち、サービスを提供するのは、業務アプリケーション(システム)のみである。ここでは、業務アプリケーション(システム)が提供するサービスの粒度のルールを以下に示す。

サービスは、1つ以上のサービスインタフェースを業務のまとまりで束ねたものである。サービスインタフェースについては、「3.2.4.2 サービスインタフェースの粒度」、サービス化の指針については、「3.2.1.1.1 サービス化の指針」を参照のこと。

(1) サービスの粒度のルール

サービスは、以下の粒度で構成すること。

- 複数のサブシステムをまたがらないように構成すること。
- サブシステム内においても、「A.サービス分割の観点」に示す観点でサービスを分割すること。

A. サービス分割の観点

① 業務体系や管理単位で適切に分割すること。

大きなサービスの管理は人間にとって煩雑なものになる。一方、細かいサービスに分割するとサービス数が多くなり、同様に管理が煩雑になる。業務の責務の切れ目や管理単位でサービスを分割し、サービスの配置が人間にとって、直感的に把握しやすく、管理しやすい大きさにする。

② バージョンアップによるサービスの起動・停止等、運用的な観点で適切な単位で分割すること。

サービス提供時間帯が異なる等、運用上独立しているサービスは資材及びデプロイ先を分け、サービスを独立させることで、システム運用における柔軟性が向上する場合、分割を行う。

③ 求められる信頼性や性能の度合いが大きく異なる機能が含まれる場合は、サービスを分割すること。

サービスが要求するシステムリソース(CPU、メモリ量、DBコネクションプール数等)を別途与える必要があるサービス群に対しては、サービスを分割しておくこと。物理構成においても、サービスに合わせて分割することで、独立して運用が可能となる。

(2) サービスの粒度のルールを考慮したサービス分割例

サブシステム内でサービスを分割する例としては、実体審査サブシステムにおける実体審査と検索外注業務がある。両者は、業務の責務が分かれており、開発単位や保守・運用における扱いも分けられることから、サービスは分割することが可能である。サービスを分割することで、実体審査と検索外注業務のサービス起動・停止等の運用やバージョンアップ等の保守を独立して管理することができるといったメリットがある。

3.2.4.2 サービスインタフェースの粒度

スコープ:	階層定型化サブシステム
プロセス:	2.4.2
指針1:	サービスインタフェースの粒度は、「(1)サービスインタフェースの粒度のルール」に従って設計すること。
規約1:	サービスインタフェース呼び出しは、「(3)アクセスパスの制限事項に関するルール」に示すとおりとすること。

業務アプリケーション(システム)のサービスインタフェースの粒度のルールを以下に示す。

(1) サービスインタフェースの粒度のルール

ビジネスプロセスのシステムタスクに対応してビジネスロジックを実行する業務アプリケーション(システム)のサービスインタフェースの粒度は、以下に従うものとする。

- 業務可視化の促進のため、業務として意味のある最小の単位を原則とする。

以降、システムタスクに対応した業務アプリケーション(システム)が提供するサービスインタフェースを「業務単位」のサービスインタフェースと呼ぶ。

(2) サービスインタフェースの粒度の設計時の留意点

業務単位のサービスインタフェースを設計する際は、粒度のルールに従い、必要に応じて、サービスインタフェースを統合又は分割すること。例えば、サービスインタフェースの出力データに他のサービスインタフェースにしか利用されない制御用データが存在する場合、その2つのサービスインタフェースは依存性がある状態と判定できる。このような場合は、粒度が小さすぎるため、サービスインタフェースの統合を行うこと。また、環境変化によりコンポーネントの特定部分が繰り返し修正されるような場合は、サービスインタフェースに複数の業務が混在しており、業務として意味のある最小単位になっていない可能性がある。このような場合は、サービスの分割を検討すること。分割したサービスインタフェースの連携フローはBPMSのビジネスプロセスに記述する。

(3) アクセスパスの制限事項に関するルール

業務アプリケーション(システム)へのアクセスは、アプリケーション同士の複雑な連携を排除しお互いを疎の関係とする目的から、提供するサービスに応じてアクセスを許容するシステム構成要素を制限する。下表に示すアクセスパスの制限事項に従うこと。

表 3.2-34 アクセスパスの制限事項

項番	業務アプリケーション(システム)が提供するサービスの概要	サービス利用側のシステム構成要素
1	ビジネスプロセスのシステムタスクに対応するビジネスロジックを実行するためのサービス (業務単位のサービスインタフェースを提供)	BPMS 他サブシステムのプレゼンテーションロジック ※1つのアクティビティのみに相当する処理で、ビジネスプロセスとして意味を成さないため、BPMSを経由するオーバーヘッドを考慮し、直接システムタスク相当の処理を呼び出す場合 ESB ※外部システムから業務単位のビジネスロジック実行要求を受け付ける場合
2	共通リソースデータの取得及び更新のためのサービス	業務アプリケーション(ユーザ) 業務アプリケーション(システム) ※項番1のサービスから利用する場合のみ 業務アプリケーション(バッチ) 他サブシステムのプレゼンテーションロジック ESB ※外部システムから共通リソースデータの操作要求を受け付ける場合

3	サブシステムが管理する個別データの取得及び更新のためのサービス	他サブシステムのプレゼンテーションロジック ※更新においては、サブシステムをまたがった業務を同時に扱う必要がある場合のみ(例えば、国際調査と国際予備審査のセット起案)
4	BPMSサービスの補完処理に必要なデータ取得及び更新のためのサービス	BPMS補完機能

項番1のシステムタスクに対応するビジネスロジックを実行するためのサービスと、項番2の共通リソースデータのアクセスのためのサービスについて、アクセスパスを下図に示す。

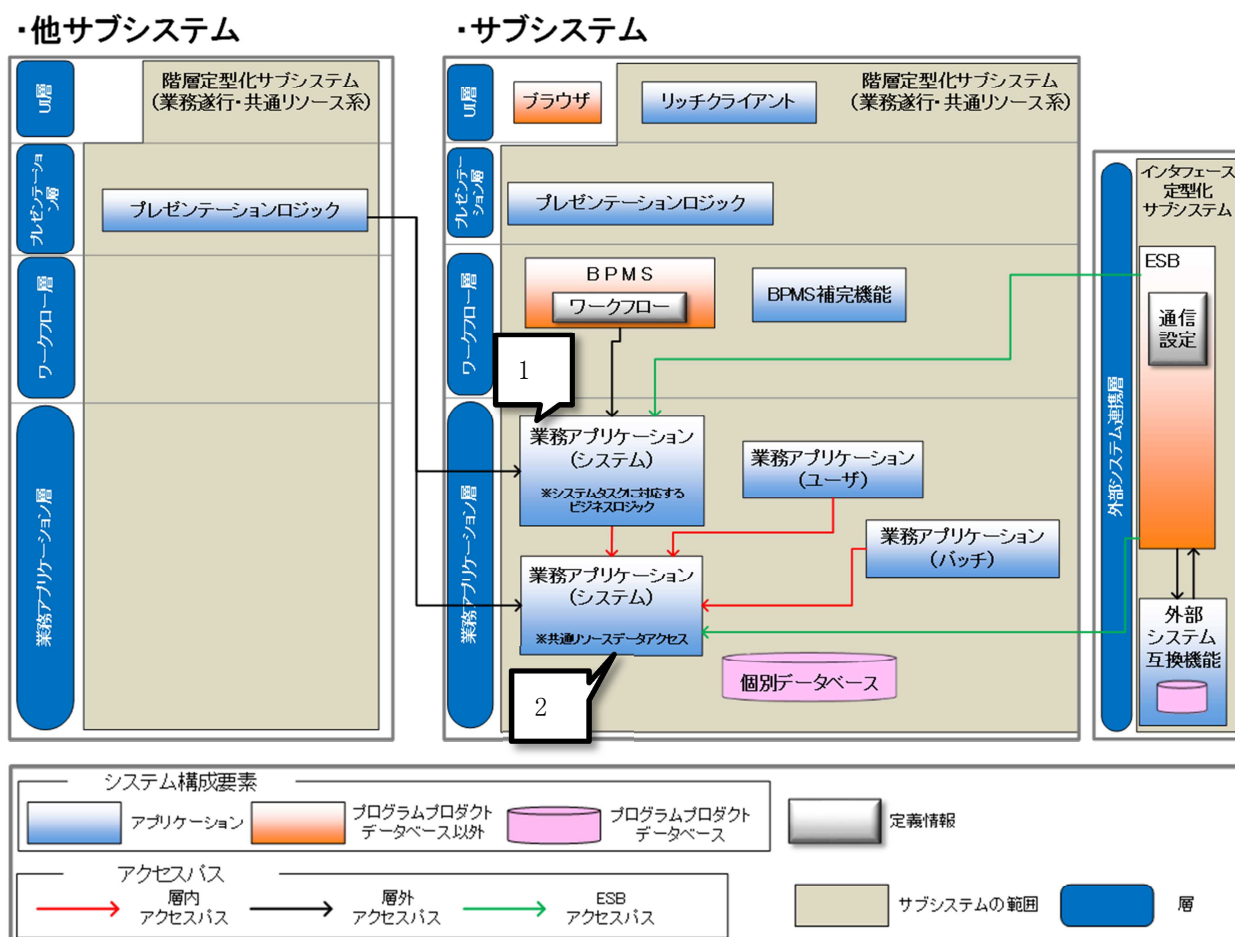


図 3.2-39 業務アプリケーション(システム)のアクセスパスの制限事項(項番1, 2)³⁷

³⁷ 図中の番号は、「表 3.2-34 アクセスパスの制限事項」の項番を示す。

項番3のサブシステムが管理する個別データの取得及び更新のためのサービスについて、アクセスパスを下図に示す。

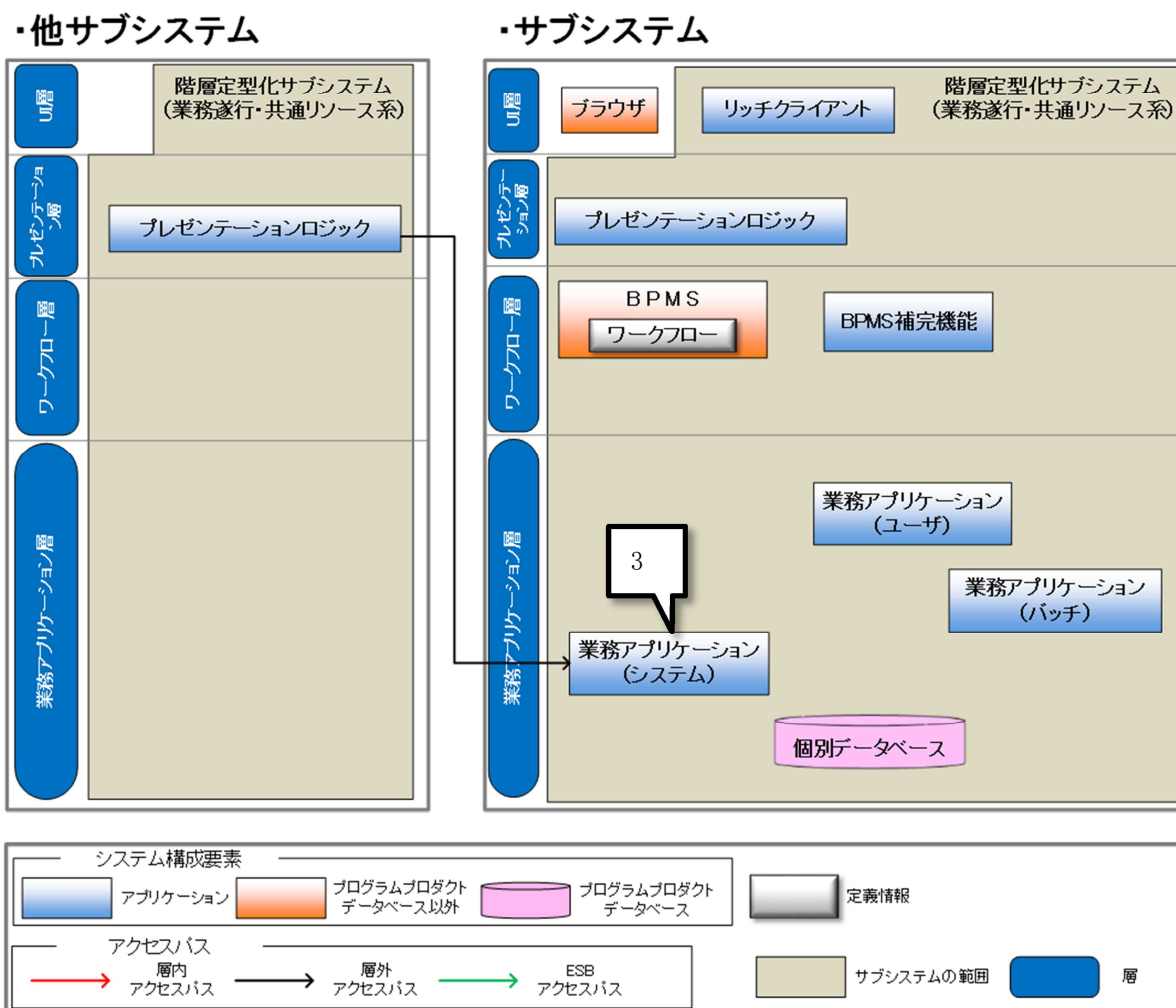
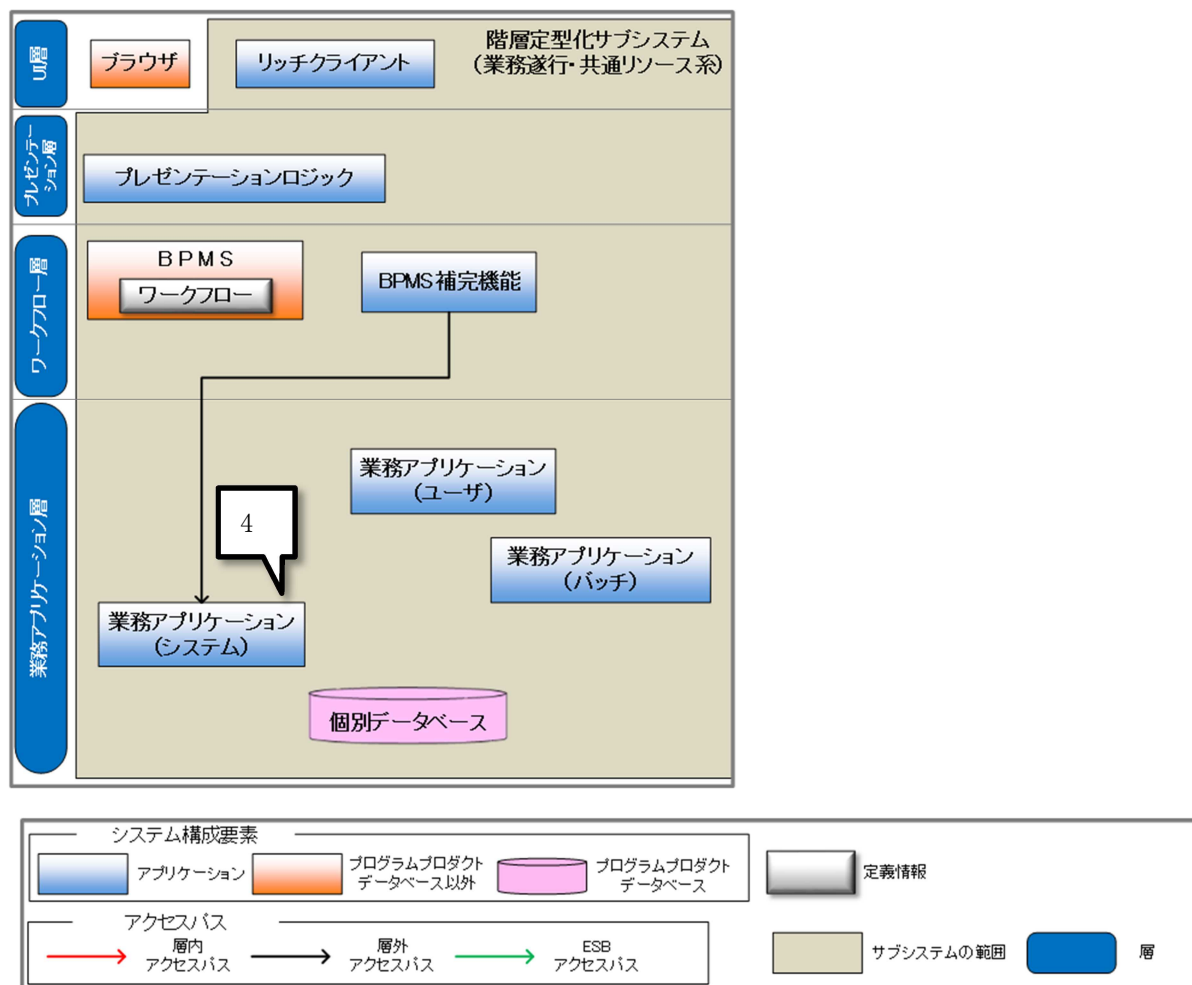


図 3.2-40 業務アプリケーション(システム)のアクセスパスの制限事項(項番3)³⁷

項番4のBPMSサービスの補完処理に必要なデータ取得及び更新のためのサービスについて、アクセスパスを下図に示す。



(4) アクセスパスの制限事項に関する補足

「表 3.2-34 アクセスパスの制限事項」で示した通り、業務アプリケーション(システム)同士のアクセスは、項番1のシステムタスクに対応するビジネスロジックを実行するためのサービスから、項番2の共通リソースデータの取得及び更新のサービスへのアクセスのみに制限するルールとしている。

下図のように、項番1の業務アプリケーション同士や、項番2の業務アプリケーション同士のアクセスは、アプリケーションを疎の関係とする目的から禁止されていることに留意すること。

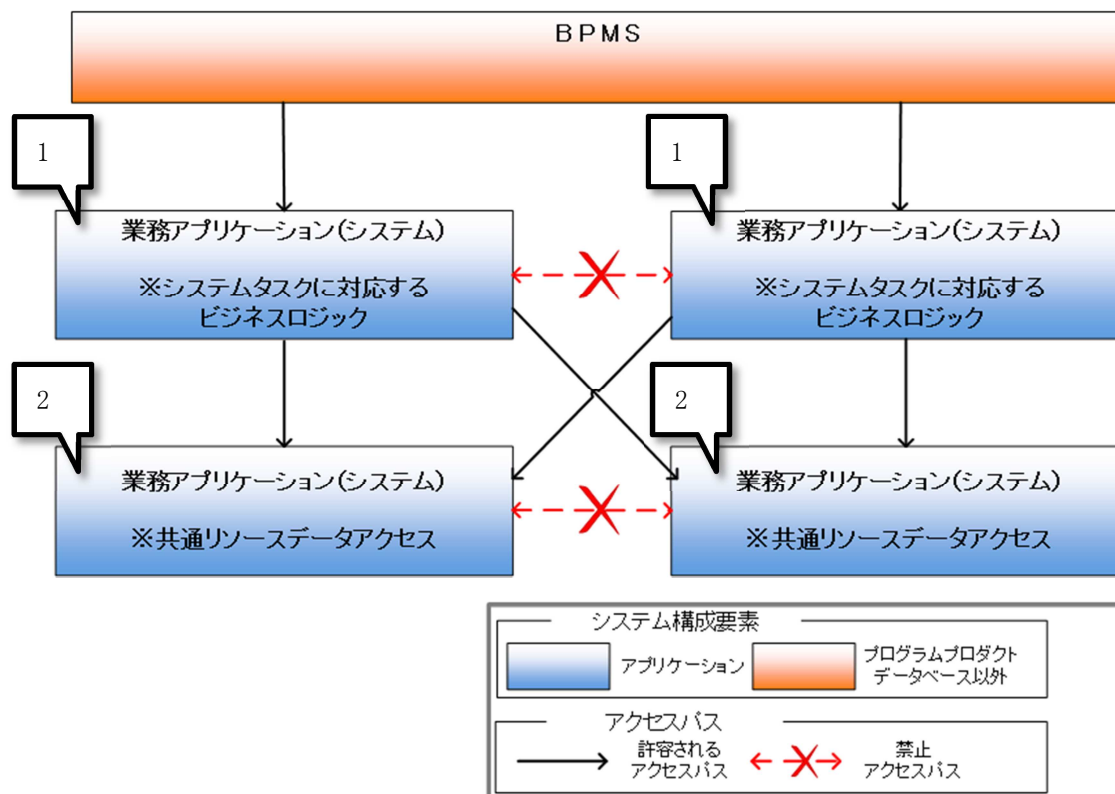


図 3.2-42 業務アプリケーション(システム)のアクセスパスの制限事項補足³⁷

3.2.4.3 サブシステム間共通機能

(1) サブシステム間共通機能の定義

サブシステム間共通機能の定義を以下に示す。

サブシステム間共通機能は、サブシステム間で共通化すべきと特許庁が判断した機能とする。
以下の条件を全て満たす業務的な機能を、サブシステム間共通機能の候補とする。

- 将来的に、不変であると考える機能
- 機能の変更のときに、業務によらず、処理ロジックを一括変更すべきと考える機能
- 複数のサブシステムから利用する機能
- 以下のいずれかに該当する機能
 - 法令等で定められた重要な項目を導出する機能
 - データの整合性を保つための機能

(2) サブシステム間共通機能の分類

ToBeアーキテクチャにおけるサブシステム間共通機能の分類を下表に示す。

表 3.2-35 サブシステム間共通機能の分類

項番	分類	詳細
1	サービス	詳細は、「表 3.2-1 サービス化対象」を参照のこと。
2	共有コンポーネント	● 共有ビジネスロジック(業務特有機能) ● 共有DBアクセス(基盤API) 詳細は、「3.1.1.4.2 オンライン処理方式」、「3.1.1.4.3 バッチ処理方式」及び「3.1.1.4.4 ディレードバッチ処理方式」を参照のこと。

3.2.4.3.1 サブシステム間共通機能の提供形態

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.3

規約1： サブシステム間共通機能の提供は、「(2)サブシステム間共通機能提供時の留意事項」に示す事項に留意し、提供すること。

(1) サブシステム間共通機能の提供形態

サブシステム間共通機能の提供形態を下表に示す。

表 3.2-36 サブシステム間共通機能の提供形態

項番	分類	提供形態
1	サービス	サービスインタフェースを公開する。
2	共有コンポーネント	コンポーネントをサブシステムに配付する。

(2) サブシステム間共通機能提供時の留意事項

サブシステム間共通機能を提供する際に留意すべき事項を以下に示す。

表 3.2-37 サブシステム間共通機能提供時の留意事項

項番	分類	提供時の留意事項	理由
1	共有コンポーネント	自サブシステム内の独自コンポーネントに依存しないこと。	自サブシステム内の独自コンポーネントに依存した状態であると、他サブシステムでその共有コンポーネントを利用するために独自コンポーネントまで必要になってしまうため。

3.2.4.4 特許庁開発フレームワーク

特許庁開発フレームワークは、ToBeアーキテクチャに対応した業務アプリケーションを構築する際に共通的に必要となる機能を提供し、構造定型化を図り、保守性及び移植性を高めるためのアプリケーション基盤である。特許庁開発フレームワークは、原則としてサブシステム毎に設計・開発するものであるが、複数のサブシステム間で同一のフレームワーク部品を利用する等して、開発効率を向上させることを禁止するものではない。また、特許庁開発フレームワークは、完全な作り込みではなくOSS等のフレームワーク製品を利用して実現する想定だが、特許庁固有の要件により、一般的なフレームワーク製品又はその組み合わせてだけでは実現できない場合は、特許庁固有の要件に対応したフレームワークの機能追加を行う。

(1) 特許庁開発フレームワークの設計方針

特許庁開発フレームワークは以下を実現できるように設計開発すること。

- コンポーネント内の個別の作り込みを小さくすること。
- 共通的に利用する機能を提供すること。
- コンポーネント間の依存性を排除すること。
- サブシステム毎にバージョンアップができること。

(2) 特許庁開発フレームワークの分類

特許庁開発フレームワークは、処理方式(詳細は、「3.1.1.3.1.1 システム構成要素の処理方式による分類」を参照のこと。)に合わせて、以下に分類される。

表 3.2-38 特許庁開発フレームワークの分類

項番	分類	対応する処理方式
1	画面オンライン・フレームワーク	● オンライン処理方式(画面系) ※サーバで動作する部分が対象 ● 帳票出力方式
2	サービス・フレームワーク	● オンライン処理方式(サービス系) ● 帳票出力方式 ● 連携処理方式
3	バッチ・フレームワーク	● バッチ処理方式 ● ディレイドバッチ処理方式 ● 帳票出力方式 ● 連携処理方式
4	リッチクライアント・フレームワーク	● オンライン処理方式(画面系) ※リッチクライアントで動作する部分が対象

帳票出力方式及び連携処理方式は、オンライン処理方式とバッチ処理方式の組み合わせであることから、画面オンライン・フレームワーク、サービス・フレームワーク又はバッチ・フレームワークに含まれる。

3.2.4.4.1 特許庁開発フレームワークの適用範囲

スコープ：階層定型化サブシステム

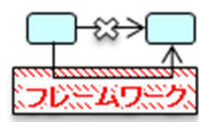
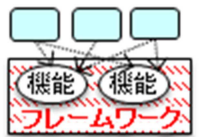
プロセス：2.4.2

規約1： 開発対象のアプリケーションに適用する特許庁開発フレームワークは、「(3)業務アプリケーションに適用するフレームワーク」に示すとおりとすること。

(1) コンポーネントにおける適用パターン

業務アプリケーションを構成するコンポーネントにおいて、特許庁開発フレームワークの適用パターンは以下のとおりである。

表 3.2-39 コンポーネントにおける特許庁開発フレームワークの適用パターン

項番	適用パターン	目的	対象コンポーネント	フレームワーク機能の例
1	コンポーネントそのものをフレームワークが提供するもの 	共通的な機能を提供する。	ロジック層、インフラストラクチャ層の一部のコンポーネント(共有コンポーネント)	- 共有DBアクセス(基盤API) - 共有ビジネスロジック(業務特有機能)
2	コンポーネントの基盤部分をフレームワークが提供するもの 	個別の作り込みを小さくする。	- インタフェース層のコンポーネント(共有画面部品を含む) - インフラストラクチャ層のコンポーネント	- SOAPサービス - 個別DBアクセス等
3	基本的に個別に作り込むもの 	個別の作り込みを小さくする(フレームワークへ依存させずに、コンポーネントの独立性を高めるべきもの)。	ロジック層のコンポーネント(個別に作成するコンポーネント)	—
4	コンポーネント間の呼び出しを制御するもの 	依存関係を制御する。	全てのコンポーネント	依存関係制御等
5	コンポーネント横断的に利用される共通的な機能を提供するもの 	個別の作り込みを小さくする。	全てのコンポーネント	- ログ - ユーティリティ - パラメータ設定値提供等

APレイヤ(「3.1.1.4.1 APレイヤの定義及びコンポーネントとの関係」を参照)におけるコンポーネントの適用パターンに基づく各コンポーネントの配置を下図に示す。

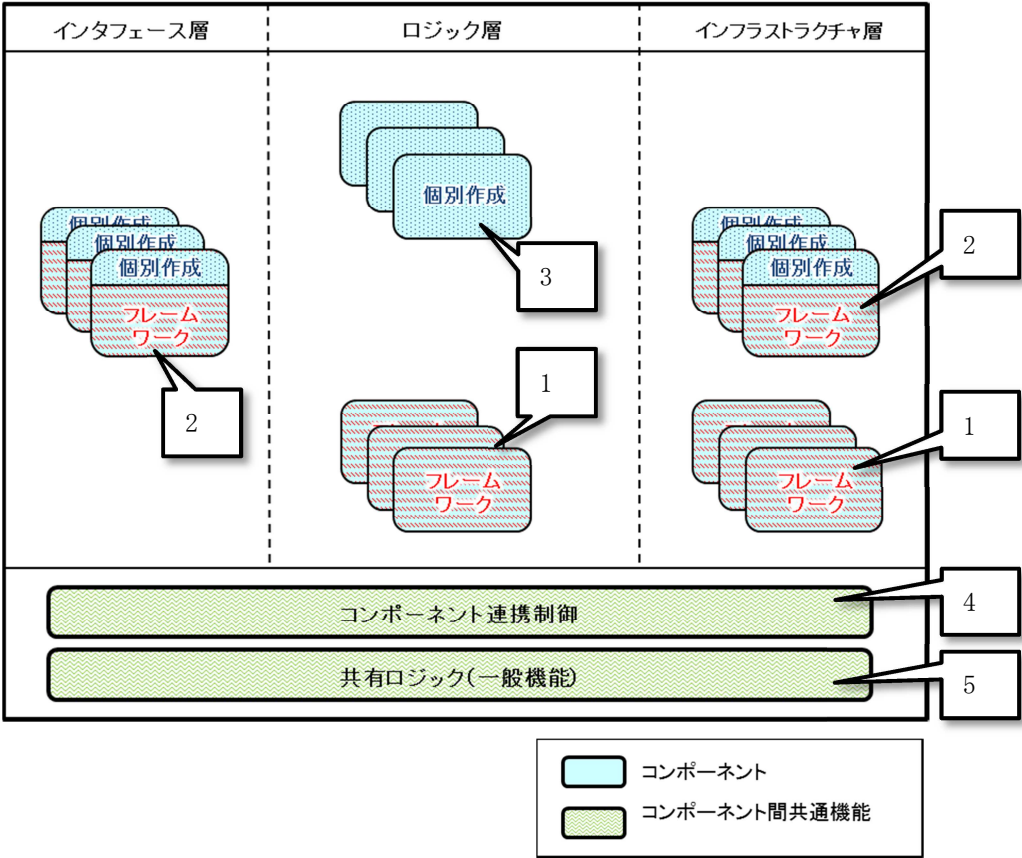


図 3.2-43 特許庁開発フレームワークにおける各コンポーネントの配置³⁸

³⁸ 図中の番号は、「表 3.2-39 コンポーネントにおける特許庁開発フレームワークの適用パターン」の項番を表す。

(2) 業務アプリケーションにおける特許庁開発フレームワークの適用範囲

業務アプリケーションは、特許庁開発フレームワークとそれを利用して構築するサブシステム固有の個別作成プログラムで構成する。

業務アプリケーションにおける特許庁開発フレームワークの適用範囲を下图に示す。

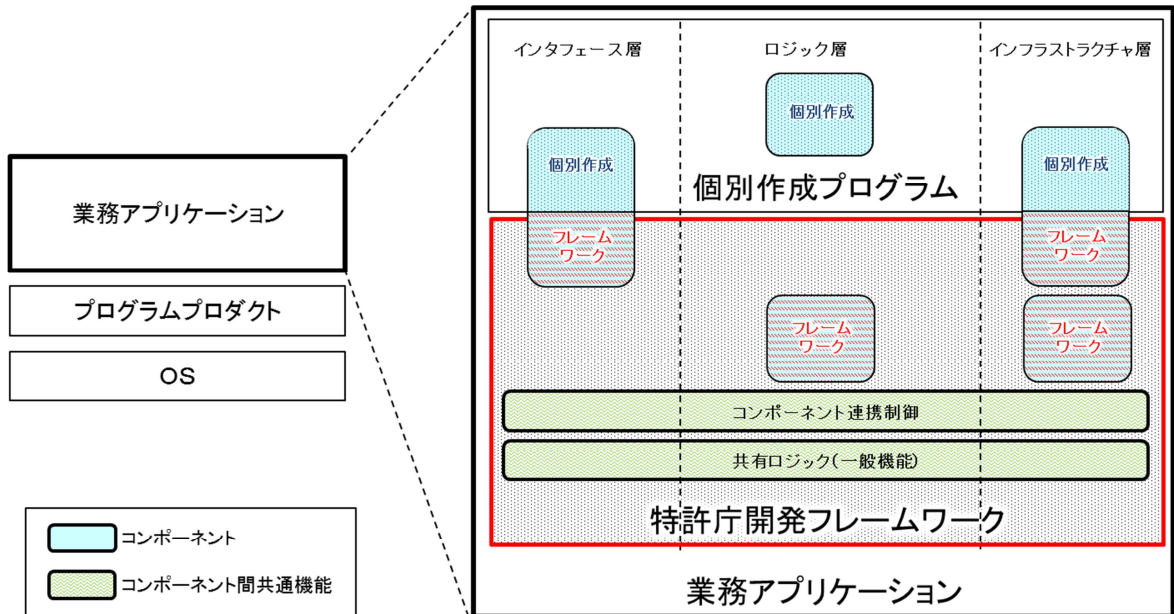


図 3.2-44 業務アプリケーションにおける特許庁開発フレームワークの適用範囲

(3) 業務アプリケーションに適用するフレームワーク

開発対象の業務アプリケーション(「表 3.1-8 開発対象のアプリケーション」を参照のこと。)に対して適用する特許庁開発フレームワークを下表に示す。

表 3.2-40 フレームワークの適用対象

項番	開発対象の業務アプリケーション	適用する特許庁開発フレームワーク
1	UI層アプリケーション(リッチクライアント)	リッチクライアント・フレームワーク
2	プレゼンテーションロジック 及び 業務アプリケーション(ユーザ) 及び 動的Webコンテンツ	画面オンライン・フレームワーク
3	静的Webコンテンツ (UI層アプリケーション含む)	画面オンライン・フレームワーク
4	BPMS補完機能	サービス・フレームワーク
5	業務アプリケーション(システム)	サービス・フレームワーク
6	業務アプリケーション(バッチ)	バッチ・フレームワーク
7	DBアクセス基盤サービス	サービス・フレームワーク
8	外部システム互換機能	対象なし ※階層定型化サブシステムではないため、対象としない。

3.2.4.4.2 特許庁開発フレームワークの機能

スコープ:	階層定型化サブシステム
プロセス:	2.4.2
規約1:	特許庁開発フレームワークの機能は、「(2)機能概要」に示す機能に準拠したものであること。
規約2:	フレームワーク機能の配置は、「(3)フレームワークに対する配置」のとおりとすること。
指針1:	サブシステム固有機能は、「(2)E.サブシステム固有フレームワーク機能」に従い、設計開発すること。

(1) 機能の分類

特許庁開発フレームワークを構成する機能には、以下の分類がある。

表 3.2-41 特許庁開発フレームワークを構成する機能の分類

項番	特許庁開発フレームワークの分類	フレームワークを構成する機能の分類
1	画面オンライン・フレームワーク	● サーバ汎用機能 ● クライアント汎用機能(ブラウザ) ● 共有コンポーネント ● サブシステム固有機能
2	サービス・フレームワーク	● サーバ汎用機能 ● 共有コンポーネント ● サブシステム固有機能
3	バッチ・フレームワーク	● サーバ汎用機能 ● 共有コンポーネント ● サブシステム固有機能
4	リッチクライアント・フレームワーク	● クライアント汎用機能(リッチクライアント) ● サブシステム固有機能

特許庁開発フレームワークを構成する機能分類の詳細を以下に示す。

表 3.2-42 特許庁開発フレームワークの機能分類

項番	分類	説明
A	サーバ汎用機能	ToBeアーキテクチャのシステム的な要件から導出されたフレームワーク機能のうち、サーバで実行される機能。
B	クライアント汎用機能(ブラウザ)	ToBeアーキテクチャのシステム的な要件から導出されたフレームワーク機能のうち、ブラウザで実行される機能。
C	クライアント汎用機能(リッチクライアント)	ToBeアーキテクチャのシステム的な要件から導出されたフレームワーク機能のうち、リッチクライアントで実行される機能。
D	共有コンポーネント	サブシステム間共通機能の共有コンポーネントである共有ビジネスロジック(業務特有機能)と共有DBアクセス(基盤API)を指す。
E	サブシステム固有機能	サブシステムの業務要件に応じて作成するフレームワークの機能。「3.2.4.4 (1)特許庁開発フレームワークの設計方針」に従い、必要に応じてサブシステム毎に設計・開発する。

特許庁開発フレームワークへの機能分類の配置イメージを下図に示す。

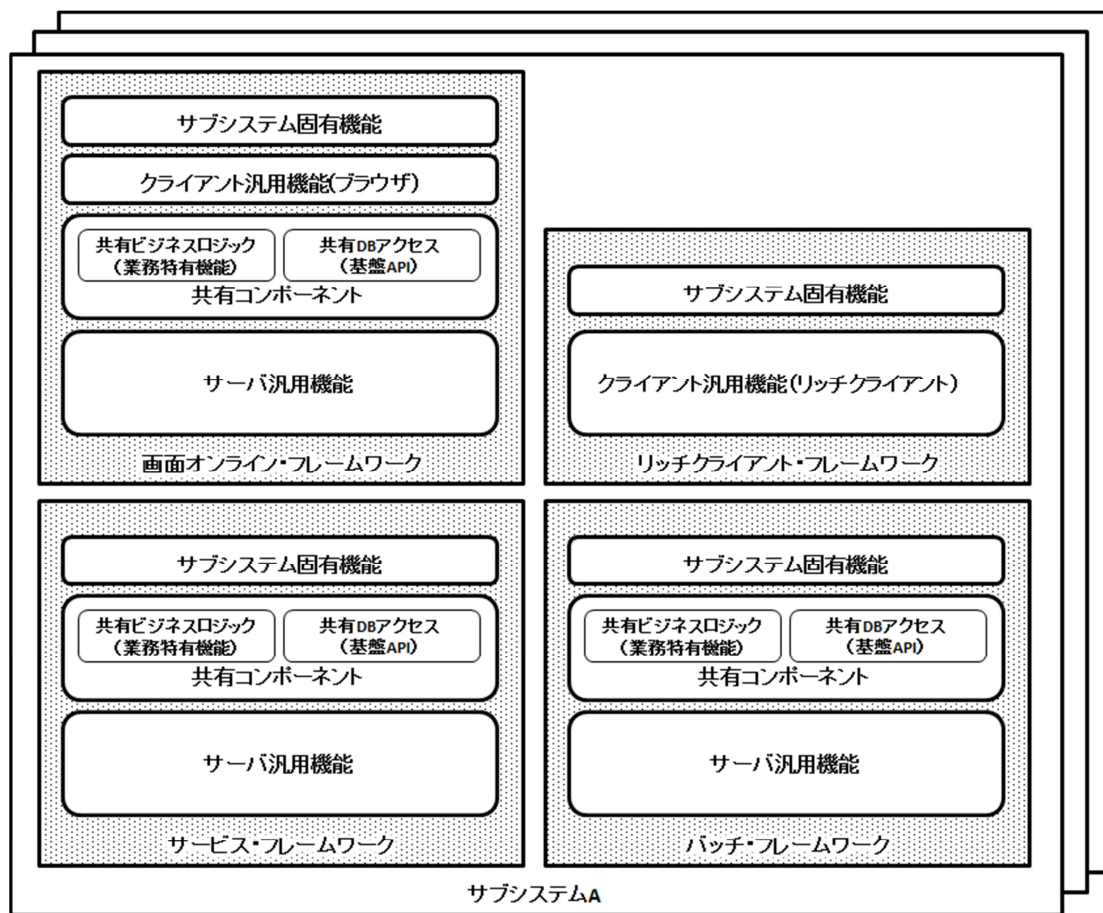


図 3.2-45 フレームワーク機能の配置イメージ

(2) 機能概要

A. サーバ汎用機能

サーバ汎用機能をAプレイヤ別に示す。

(A) インタフェース層

インタフェース層におけるサーバ汎用機能を以下に示す。

表 3.2-43 サーバ汎用機能(インタフェース層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	SOAPサービス	SOAPの形式に従ったサービスを作成するための機能 ● インタフェース定義 ● 電文の解析 ● 応答電文作成 等	プレゼンテーションロジック	リクエストハンドラ
			業務アプリケーション(システム)	サービスインタフェース
			BPMS補完機能	サービスインタフェース
2	RESTサービス	RESTの形式に従ったサービスを作成するための機能 ● インタフェース定義 ● 電文の解析 ● 応答電文作成 等	プレゼンテーションロジック	リクエストハンドラ
			DBアクセス基盤サービス	基盤サービスインタフェース
3	LDAPアクセス	LDAPサーバへのアクセス機能 LDAPサーバを用いた認証・認可については、「3.2.13.1 (2)認証・認可」を参照。	プレゼンテーションロジック	リクエストハンドラ
4	認証・認可チェック	ユーザ情報を利用し、認証を行う。取得した認可情報で必要に応じて、認可チェックをする。 詳細は、「3.1.2.1 オンライン処理方式」、「3.2.13.1 (2)認証・認可」を参照。	プレゼンテーションロジック	リクエストハンドラ
			業務アプリケーション(システム)	サービスインタフェース
			BPMS補完機能	サービスインタフェース
			DBアクセス基盤サービス	基盤サービスインタフェース
5	入力チェック	サーバでの入力値のチェック(必須チェック、フォーマットチェック、範囲チェック、関連性チェック、文字種チェック等)を行う機能。 入力チェック機能が利用されるタイミングや利用方法については、「3.1.2.1 オンライン処理方式」、「3.1.2.2 バッチ処理方式」を参照。文字種チェックについては、「3.2.10 文字コードの扱いのルール」を参照。	プレゼンテーションロジック	リクエストハンドラ
			業務アプリケーション(システム)	サービスインタフェース
			業務アプリケーション(バッチ)	バッチインタフェース
			BPMS補完機能	サービスインタフェース
			DBアクセス基盤サービス	基盤サービスインタフェース
6	ディレード処理要求制御	定期的に「ディレード処理要求管理情報」を確認し、登録された要求がある場合、要求内容に応じたビジネスロジックを実行する。詳細は、「3.1.1.4.4 ディレードバッチ処理方式」を参照。	業務アプリケーション(バッチ)	バッチインタフェース

(B) ビジネスロジック層

ビジネスロジック層におけるサーバ汎用機能を以下に示す。

表 3.2-44 サーバ汎用機能(ビジネスロジック層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	帳票出力	帳票出力処理に関わる機能 ● 帳票ファイル作成 ● 帳票出力情報更新 ● 帳票印刷要求 等 帳票出力処理方式については、「3.1.2.4 帳票出力方式」を参照。	業務アプリケーション(ユーザ)	共有ビジネスロジック
			業務アプリケーション(システム)	共有ビジネスロジック
			業務アプリケーション(バッチ)	共有ビジネスロジック

(C) インフラストラクチャ層

インフラストラクチャ層におけるサーバ汎用機能を以下に示す。

表 3.2-45 サーバ汎用機能(インフラストラクチャ層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	個別DBアクセス	個別データベースのデータ取得及び更新機能。汎用的なAPIを提供するために製品固有のデータアクセス技術、永続化手段を隠蔽する。	業務アプリケーション(ユーザ)	個別DBアクセス
			業務アプリケーション(システム)	
			業務アプリケーション(バッチ)	
2	SOAPクライアント	SOAPサービスを呼び出すための機能。SOAPによる通信技術や手順を隠蔽する。	業務アプリケーション(ユーザ)	業務AP(システム)アクセス
			業務アプリケーション(システム)	
			業務アプリケーション(バッチ)	
			BPMS補完機能	
3	ルールエンジンアクセス	BRMSヘルールの実行を要求する機能。原則、ビジネスルール管理サブシステムのサービス呼び出しを行うが、高度な性能要件によってはBRMSのAPIライブラリを利用したコンポーネントにする。ロジック層に対してBRMSの呼び出し方式を隠蔽する。詳細は、「3.2.7.1.1 ビジネスルールをアプリケーションから切り離すための仕組みとして利用する技術」を参照。	業務アプリケーション(ユーザ)	ルールエンジンアクセス
			業務アプリケーション(システム)	
			業務アプリケーション(バッチ)	
4	外部システムアクセス	ESBが提供する外部システム連携インタフェースを呼び出す機能。上述のSOAPクライアントをそのまま使用するか、それをラップした形態とする。	業務アプリケーション(システム)	外部システムアクセス
			業務アプリケーション(バッチ)	
			BPMS補完機能	
5	ワークフローエンジンアクセス	BPMSのサービスを呼び出す機能。上述のSOAPクライアントやBPMS製品が提供するAPIを利用してアクセスするが、アクセス方式をロジック層に対して隠蔽する。	プレゼンテーションロジック	ワークフローエンジンアクセス
			業務アプリケーション(システム)	
			業務アプリケーション(バッチ)	
			BPMS補完機能	
6	FTPクライアント	他サブシステムの業務アプリケーション(バッチ)に対し、ファイルでデータを連携するためにFTP通信を行う。FTP呼び出しを行う際の通信技術や手順を上位層から隠蔽する。	業務アプリケーション(バッチ)	FTPクライアント

(D) APLレイヤ横断

APレイヤを横断して利用するサーバ汎用機能を以下に示す。

表 3.2-46 サーバ汎用機能 (APレイヤ横断)

項番	機能名	機能概要	システム構成要素
1	依存関係制御	変化の多いコンポーネントの交換可能性を高めることを目的とし、コンポーネントの依存関係を制御する。 DIコンテナ、MVCフレームワーク、AOPフレームワーク、ファクトリ等	● UI層以外のアプリケーションが配置されるシステム構成要素
2	ログ出力	デバッグやシステム監視等の目的でアプリケーションの動作をファイルに記録する機能。 ログ出力機能を利用したログ管理方式については、「3.2.13.7 ログ管理方式設計のルール」を参照。	● UI層以外のアプリケーションが配置されるシステム構成要素
3	メッセージ管理	画面に出力するメッセージをアプリケーションから切り離された定義情報を使用して管理する機能。メッセージのKeyやID等を指定してメッセージを取得する。	● UI層以外のアプリケーションが配置されるシステム構成要素
4	パラメータ設定値提供	設定ファイルやDBに設定された値を読み取り、アプリケーションで使用するパラメータ値を応答する機能。 ビジネスルールをアプリケーションから切り離す仕組みとしても利用する。「3.2.7.1.1 ビジネスルールをアプリケーションから切り離すための仕組みとして利用する技術」を参照。	● UI層以外のアプリケーションが配置されるシステム構成要素
5	ユーティリティ	日付・年号操作、文字列操作、CSV、XMLの解析等といったユーティリティ機能。	● UI層以外のアプリケーションが配置されるシステム構成要素

B. クライアント汎用機能(ブラウザ)

(A) インタフェース層

インタフェース層におけるクライアント汎用機能(ブラウザ)を以下に示す。

表中の「システム構成要素」及び「コンポーネント」は、機能の実行場所を示す。機能の配信場所は、プレゼンテーションロジックのリクエストハンドラ・コンポーネントである。

表 3.2-47 クライアント汎用機能(ブラウザ) (インタフェース層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	共通画面部品	テキストボックス, ボタン等の基本部品や, カレンダーや書式付きエディタ等といった応用部品のスタイルや機能。 画面設計ルール画面部品の実現する。 画面部品については, 「3.2.2.4.3 (3)画面部品の扱い」を参照。	ブラウザ	画面
2	イベント制御	画面操作によって発生するクリック等のイベントを制御し, 対応する処理を実行する機能。 画面設計ルール画面の動作を実現する。 画面の動作については, 「3.2.2.4.4 画面の動作に関するルール」を参照。	ブラウザ	イベントハンドラ
3	入力規制	禁止文字や入力桁数制限等といった, 不正となる文字を入力できないように制御する機能。入力済みの値をチェックする入力チェックとは異なり, 文字自体の入力を正当な入力値のみを許可するよう制限する機能である。不正な入力を入力規制で防止するか, クライアント汎用機能の入力チェックで防止するかは, ユーザ利便性や使用する技術などの観点から設計すること。	ブラウザ	イベントハンドラ

(B) ロジック層

ロジック層におけるクライアント汎用機能(ブラウザ)を以下に示す。

表中の「システム構成要素」及び「コンポーネント」は、機能の実行場所を示す。機能の配信場所は、プレゼンテーションロジックのリクエストハンドラ・コンポーネントである。

表 3.2-48 クライアント汎用機能(ブラウザ) (ロジック層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	入力チェック	ユーザの入力値チェック(必須チェック, フォーマットチェック, 範囲チェック, 関連性チェック等)を行う機能。 画面設計ルール画面の入力チェックを実現する。 入力チェックについては, 「3.2.2.4.4 画面の動作に関するルール」を参照。 入力チェック機能が利用されるタイミングについては, 「3.1.2.1 オンライン処理方式」を参照。	ブラウザ	画面ロジック

(C) インフラストラクチャ層

インフラストラクチャ層におけるクライアント汎用機能(ブラウザ)を以下に示す。

表中の「システム構成要素」及び「コンポーネント」は、機能の実行場所を示す。機能の配信場所は、プレゼンテーションロジックのリクエストハンドラ・コンポーネントである。

表 3.2-49 クライアント汎用機能(ブラウザ) (インフラストラクチャ層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	サーバ通信	サーバと通信するための機能。通信技術や手順を隠蔽する。	ブラウザ	プレゼンテーションロジックアクセス

(D) APLレイヤ横断

APレイヤを横断して利用するクライアント汎用機能(ブラウザ)を以下に示す。

表中の「システム構成要素」は、機能の実行場所を示す。機能の配信場所は、プレゼンテーションロジックのリクエストハンドラ・コンポーネントである。

表 3.2-50 クライアント汎用機能(ブラウザ) (APレイヤ横断)

項番	機能名	機能概要	システム構成要素
1	ユーティリティ	日付・年号操作, 文字列操作, CSV, XMLの解析等といったユーティリティ機能。	ブラウザ

C. クライアント汎用機能(リッチクライアント)

(A) インタフェース層

インタフェース層におけるクライアント汎用機能(リッチクライアント)を以下に示す。

表 3.2-51 クライアント汎用機能(リッチクライアント) (インタフェース層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	共通画面部品	テキストボックス, ボタン等の基本部品や, カレンダや書式付きエディタ等といった応用部品のスタイルや機能。 画面設計ルール画面部品の実現する。 画面部品については, 「3.2.2.4.3 (3)画面部品の扱い」を参照。	リッチクライアント	画面
2	イベント制御	画面操作によって発生するクリック等のイベントを制御し, 対応する処理を実行する機能。 画面設計ルール画面の動作を実現する。 画面の動作については, 「3.2.2.4.4画面の動作に関するルール」を参照。	リッチクライアント	イベントハンドラ
3	入力規制	禁止文字や入力桁数制限等といった, 不正となる文字を入力できないように制御する機能。入力済みの値をチェックする入力チェックとは異なり, 文字自体の入力を正当な入力値のみを許可するよう制限する機能である。不正な入力を入力規制で防止するか, クライアント汎用機能の入力チェックで防止するかは, ユーザ利便性や使用する技術などの観点から設計すること。	リッチクライアント	イベントハンドラ

(B) ロジック層

ロジック層におけるクライアント汎用機能(リッチクライアント)を以下に示す。

表 3.2-52 クライアント汎用機能(リッチクライアント) (ロジック層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	入力チェック	ユーザの入力値チェック(必須チェック, フォーマットチェック, 範囲チェック, 関連性チェック等)を行う機能。 画面設計ルール画面の入力チェックを実現する。 入力チェックについては, 「3.2.2.4.4画面の動作に関するルール」を参照。 入力チェック機能が利用されるタイミングについては, 「3.1.2.1 オンライン処理方式」を参照。	リッチクライアント	画面ロジック

(C) インフラストラクチャ層

インフラストラクチャ層におけるクライアント汎用機能(リッチクライアント)を以下に示す。

表 3.2-53 クライアント汎用機能(リッチクライアント) (インフラストラクチャ層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	サーバ通信	Webサービスを呼び出すための機能。通信技術や手順を隠蔽する。	リッチクライアント	プレゼンテーションロジックアクセス

(D) APLレイヤ横断

APレイヤを横断して利用するクライアント汎用機能(リッチクライアント)を以下に示す。

表 3.2-54 クライアント汎用機能(リッチクライアント) (APレイヤ横断)

項番	機能名	機能概要	システム構成要素
1	依存関係制御	コンポーネントの依存関係を制御する。 DIコンテナ, MVCフレームワーク, AOPフレームワーク, ファクトリ等	リッチクライアント
2	ログ出力	デバッグやシステム監視等の目的でアプリケーションの動作をファイルに記録する機能。 ログ出力機能を利用したログ管理方式については,「3.2.13.7 ログ管理方式設計のルール」を参照。	リッチクライアント
3	メッセージ管理	画面に出力するメッセージをアプリケーションから切り離された定義情報を使用して管理する機能。メッセージのKeyやID等を指定してメッセージを取得する。	リッチクライアント
4	パラメータ設定値提供	設定ファイルやDBに設定された値を読み取り, アプリケーションで使用するパラメータ値を応答する機能。 ビジネスルールをアプリケーションから切り離す仕組みとしても利用する。「3.2.7.1.1 ビジネスルールをアプリケーションから切り離すための仕組みとして利用する技術」を参照。	リッチクライアント
5	ユーティリティ	日付・年号操作, 文字列操作, CSV, XMLの解析等といったユーティリティ機能。	リッチクライアント

D. 共有コンポーネント

共有コンポーネントは、インタフェース層の機能は持たない。

(A) ロジック層

ロジック層における共有コンポーネントを以下に示す。

表 3.2-55 共有コンポーネント(ロジック層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	共有ビジネスロジック(業務特有機能)	ビジネスロジックのうち、複数の業務アプリケーションで共有し、かつ特許庁業務特有の機能(期間計算等)。サブシステム間共通機能の共有コンポーネントである共有ビジネスロジック(業務特有機能)を指す。 共有ビジネスロジック(業務特有機能)は、コンポーネントが各サブシステムに配付されるため、ラップした形で、ロジック層に対して隠蔽した形で利用すること。	業務アプリケーション(ユーザ) 業務アプリケーション(システム) 業務アプリケーション(バッチ) BPMS補完機能	共有ビジネスロジック(業務特有機能)

(B) インフラストラクチャ層

インフラストラクチャ層における共有コンポーネントを以下に示す。

表 3.2-56 共有コンポーネント(インフラストラクチャ層)

項番	機能名	機能概要	システム構成要素	コンポーネント名
1	共有DBアクセス(基盤API)	共有データベースのデータ取得及び更新機能。 『DBアクセス基盤 利用者向けガイドライン』の基盤APIを指す。	業務アプリケーション(ユーザ) 業務アプリケーション(システム) 業務アプリケーション(バッチ) DBアクセス基盤サービス	共有DBアクセス(基盤API)

E. サブシステム固有フレームワーク機能

「3.2.4.4 (1)特許庁開発フレームワークの設計方針」に従い、サブシステムの業務要件に応じてサブシステム毎に設計開発すること。

(3) フレームワークに対する配置

フレームワーク機能の各フレームワークへの配置を以下に示す。

表 3.2-57 フレームワーク機能の配置

項番	機能分類	Aレイヤ	機能名	特許庁開発フレームワーク			
				画面オンライン	サービス	バッチ	リッチクライアント
1	サーバ汎用機能	インタフェース層	SOAPサービス	○ ³⁹	○	×	×
2			RESTサービス	○ ³⁹	○	×	×
3			LDAPアクセス	○	×	×	×
4			認証・認可チェック	○	○	×	×
5			入力チェック	○	○	○	×
6			ディレード処理要求制御	×	×	○	×
7		ビジネスロジック層	帳票出力	○	○	○	×
8		インフラストラクチャ層	個別DBアクセス	○	○	○	×
9			SOAPクライアント	○	○	○	×
10			ルールエンジンアクセス	○	○	○	×
11			外部システムアクセス	×	○	○	×
12			ワークフローエンジンアクセス	○	○	○	×
13			FTPクライアント	×	×	○	×
14		Aレイヤ横断	依存性管理	○	○	○	×
15			ログ出力	○	○	○	×
16			メッセージ管理	○	○	○	×
17			パラメータ設定値提供	○	○	○	×
18			ユーティリティ	○	○	○	×
19	クライアント汎用機能(ブラウザ)	インタフェース層	共通画面部品	○	×	×	×
20			イベント制御	○	×	×	×
21			入力規制	○	×	×	×
22		ロジック層	入力チェック	○	×	×	×
23		インフラストラクチャ層	サーバ通信	○	×	×	×
24		Aレイヤ横断	ユーティリティ	○	×	×	×
25	クライアント汎用機能(リッチクライアント)	インタフェース層	共通画面部品	×	×	×	○
26			イベント制御	×	×	×	○
27			入力規制	×	×	×	○
28		ロジック層	入力チェック	×	×	×	○
29		インフラストラクチャ層	サーバ通信	×	×	×	○
30		Aレイヤ横断	依存関係制御	×	×	×	○
31			ログ出力	×	×	×	○
32			メッセージ管理	×	×	×	○
33			パラメータ設定値提供	×	×	×	○
34			ユーティリティ	×	×	×	○
35	共有コンポーネント	ビジネスロジック層	共有ビジネスロジック(業務特有機能)	○	○	○	×
36		インフラストラクチャ層	共有DBアクセス(基盤API)	○	○	○	×

凡例：○:配置する(ただし業務要件がない場合は配置しない)，×:配置しない。

³⁹ UI層がリッチクライアントの場合に、SOAPサービス、RESTサービス的一方又は両方が配置される。詳細は「3.2.1.1.2 内部インタフェースのプロトコル」参照。UI層がブラウザの場合には配置されない。

3.2.4.4.3 特許庁開発フレームワークのサブシステム間の共有

スコープ：階層定型化サブシステム

プロセス：2.4.3

指針1： 特許庁開発フレームワークをサブシステム間で共有する場合、サブシステム毎に独立してバージョンアップができるよう、資材管理の仕組みを整備すること。

特許庁開発フレームワークは、原則としてサブシステム毎に設計・開発するものであるが、複数のサブシステム間で同一のフレームワーク部品を利用する等して、開発効率を向上させることを禁止するものではない。しかしながら、フレームワーク機能への変更要求の内容やサブシステムの刷新のタイミングはサブシステム毎に異なる可能性があるため、サブシステム毎に独立して特許庁開発フレームワークを更新できる必要がある。あるサブシステムの刷新が特許庁開発フレームワークの更新を伴うときに、同じフレームワーク部品を利用している別のサブシステムに影響を及ぼすような状況を防ぐため、サブシステム毎に独立してバージョンアップができるよう、資材管理の仕組みを整備することとする。

特に特許庁開発フレームワークは、完全な作り込みではなくフレームワーク製品を利用して実現する想定であるため、製品のバージョンや特定バージョンに依存した作り込み部分の資材管理をサブシステム毎に行う必要がある。

3.2.4.5 個別データベースの構成及びアクセスルール

3.2.4.5.1 個別データベースの構成

スコープ：階層定型化サブシステム

プロセス：2.4.2

規約1： 個別データベースの構成は、「(1)個別データベースの構成ルール」のとおりとすること。

(1) 個別データベースの構成ルール

個別データベースの構成ルールを以下に示す。

- 個別データベースは、「サブシステムに1つ」とする。
- 『データ統合方針書』の「2.3.2 データの具体的な配置位置(情報活用系システム用データを除く)」のとおり、個別データベースに配置する対象データは共通リソースデータ、個別リソースデータ及び個別業務イベントデータである。各対象データの範囲に従って設計すること。

3.2.4.5.2 個別データベースへのアクセスルール

スコープ：階層定型化サブシステム

プロセス：2.4.2

規約1： 個別データベースへのアクセスは、「(1)個別データベースへのアクセスルール」に準拠すること。

スコープ：階層定型化サブシステム

プロセス：2.4.3

指針1： 情報活用系サブシステムへのデータ提供は、「(3)情報活用系サブシステムへのデータ提供のルール」に準拠して設計すること。

(1) 個別データベースへのアクセスルール

個別データベースへのアクセスルールを以下に示す。

- 同一サブシステム内の業務アプリケーションからのアクセスを許可する。
- 情報活用系サブシステムへデータ提供する目的以外では、他サブシステムの個別データベースを直接アクセスすることを禁止する。

個別データベースにアクセスするイメージを下図に示す。

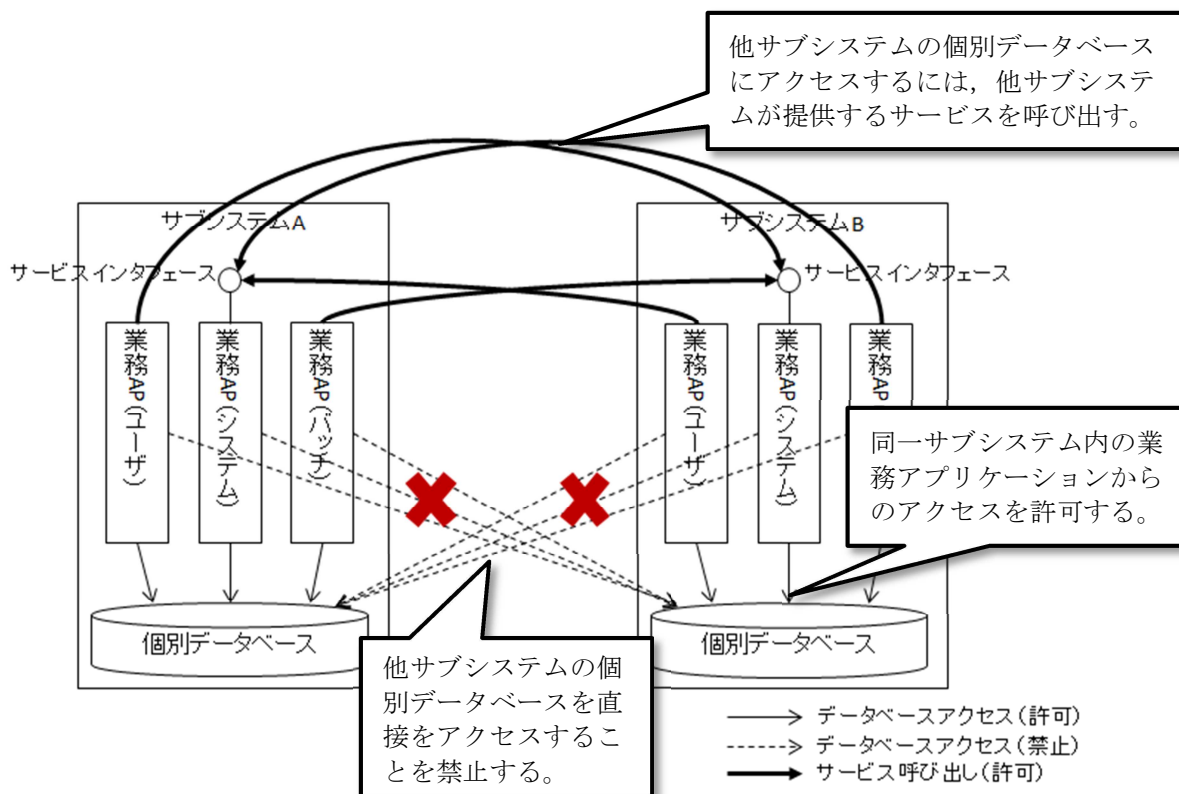


図 3.2-46 個別データベースへのアクセスイメージ

(2) 個別データベースへのアクセスルールの補足

個別データベースへのアクセスルールに従い、BRMS、BPMSから個別データベースへのアクセスは禁止する。

他サブシステムの個別データベースにアクセスするには、他サブシステムが提供するサービス呼び出すものとする。ただし、業務アプリケーション(システム)同士の呼び出し関係は一定のルールに従う必要がある。詳細は、「3.2.4.2 サービスインターフェースの粒度 (3)アクセスパスの制限事項に関するルール」を参照のこと。

(3) 情報活用系サブシステムへのデータ提供のルール

情報活用系サブシステムへのデータ提供に関してのルールを以下に示す。

- 個別データベースは情報活用系サブシステムへのデータ提供に対応するため、一般的なETLの機能によってデータ取得ができる方法を提供すること。

3.2.5 外部システム連携層のルール

(1) 責務

外部システム連携層の責務は、「内部システムと外部システム間で通信する際のギャップの吸収」である。外部システム連携層の責務を担うための主な処理内容を以下に示す。

A. 内部システムから外部システムへの連携

- 定型化された内部インタフェースとして、サービスインタフェースを内部システムに対し提供する。
- 内部システムから受信したデータの形式、文字コード等を連携先外部システムのインタフェース仕様に合わせて変換する(連携方式のギャップ吸収を行う。)
- 連携先外部システムのインタフェース仕様に合ったプロトコルで外部システム接続し、連携データを送信する。
- 連携先外部システムからのレスポンスのデータ形式、文字コード等を内部システムのインタフェースに合わせて変換する(連携方式のギャップ吸収を行う。)
- 内部インタフェースを利用してレスポンスを内部システムへ返却する。

B. 外部システムから内部システムへの連携

- 各外部システムがサポートする通信仕様に則ったインタフェースを外部システムに対し提供する。
- 外部システムから受信したデータの形式、文字コード等をToBeアーキテクチャで標準としている電文形式、文字コード等に変換する(連携方式のギャップ吸収を行う。)
- ToBeアーキテクチャで標準としているプロトコルで連携先内部システムへ接続し、連携データを送信する。
- 連携先内部システムからのレスポンスのデータ形式、文字コード等を外部システムの通信仕様に合わせて変換する(連携方式のギャップ吸収を行う。)
- 外部システムがサポートする通信仕様に則り、レスポンスを外部システムへ返却する。

(2) システム構成要素

外部システム連携層を構成するシステム構成要素を以下に示す。以下のシステム構成要素が連携することで、外部システム連携層の責務を実現する。

- ESB
- 外部システム互換機能

A. ESB

- ESBの責務は、「内部システムと外部システムの連携方式の違い(ギャップ)の吸収」である。
- 内部システムから外部システムへの連携においては、ワークフロー層、業務アプリケーション層のシステム構成要素(個別データベースを除く)からサービスインタフェース経由でアクセスされることにより、業務処理を実行する。
- 外部システムから内部システムへの連携においては、外部システムから外部システムのインタフェース仕様にあったインタフェースでアクセスされることにより、業務処理を実行する。
- 以下に列挙する場合に限り、他システム構成要素にアクセスしてもよい。
 - 内部システムから受信したデータを連携するために外部システムへアクセスする。
 - 内部システムと外部システムとの連携方式のギャップを吸収する処理を行うために、外部システム互換機能へアクセスする。
 - 外部システムから受信したデータを連携するためにBPMS、BPMS補完機能、業務アプリケーション(システム)、DBアクセス基盤サービスへアクセスする。

B. 外部システム互換機能⁴⁰

- 外部システム互換機能の責務は、「ギャップのうちESBで吸収しない部分の補完」である。
- ESB又は外部システムからアクセスされることにより、業務処理を実行する。
- 外部システム互換機能は、ESBで処理できない連携方式のギャップを吸収する処理で、データ蓄積が必要な場合等のために、システム構成要素の内部にデータベースを持つ。
- 以下に列挙する場合に限り、他システム構成要素にアクセスしてもよい。
 - 外部システムからのESBがサポートしていないプロトコルによる電文受信後、内部システムへ連携するために、ESBへアクセスする。
 - ESBで処理できない連携方式のギャップを吸収する処理で、共有データベースへのデータ取得又は更新のために、DBアクセス基盤サービス又は共有データベースへアクセスする。

⁴⁰ 外部システム互換機能の実現手段 (ESBのアダプタとするか、サーバ上のアプリケーションとするか等) は定めていない。

3.2.5.1 外部システム連携層の管理ルール

3.2.5.1.1 外部システム連携層に利用する技術

スコープ： インタフェース定型化サブシステム(外部システム連携)

プロセス： 2.4.2

規約1： 外部システム連携層は, ESBを含めること。

規約2： 外部システム連携層の機能は, 必要となる機能のうち, ESBのみで実現できないものについて, 外部システム互換機能により補完すること。

外部システム連携層に利用するプログラムプロダクトは, ESBとする。

ただし, ESBの機能を補完する機能として外部システム互換機能を開発するものとする。

3.2.5.1.2 外部システム連携層に持たせる機能

スコープ： インタフェース定型化サブシステム(外部システム連携)

プロセス： 2.3.3

規約1： 外部システム連携層の機能は、「(1)外部システム連携層の機能」に示す機能とすること。

指針1： 外部システム連携層の機能は、「(2)外部システム連携層が提供することを禁止する機能」に従い設計すること。

スコープ： インタフェース定型化サブシステム(外部システム連携)

プロセス： 2.4.2

指針1： 外部システム連携層で吸収するギャップは、「(3)外部システム連携層が吸収するギャップ」に記載したものを対象とするように設計すること。

指針2： ESBと外部システム互換機能の責務の分担は、「(4)ESBと外部システム互換機能の責務の分担」に従うように設計すること。

(1) 外部システム連携層の機能

外部システム連携層に基本的に必要な機能を下表に示す。なお、刷新過渡期のToBe対象システムとの連携においてのみ利用する機能もある。これらの機能は、刷新後、他の機能等で実現する。

表 3.2-58 外部システム連携層が持つ機能

項番	大分類	小分類	項目	外部システムとの連携 (右記以外)	外部システムのうち、 刷新過渡期のToBe対象システムとの連携	
				利用目的	利用目的	刷新後の 実現方法
1	通信機能	メッセージング	メッセージの送受信	外部システムとのメッセージの送受信を行う。	ToBe対象システムとのメッセージの送受信を行う。	内部システム間の連携は、単件リアルタイム化する。
2			システム間連携ログ	外部システムとの連携における使用状況の確認や問題発生時の解析に利用する。	ToBe対象システムとの連携における使用状況の確認や問題発生時の解析に利用する。	連携を実現する他のシステム構成要素のログを使用する。
3			再送や重複排除	利用しない。	原則、利用しない。 ただし、設計工程での検討の結果、必要性が認められる連携に限り、個別に使用することを許容する。	原則、実施しない。 ただし、必要な場合、内部システムが実施する。

項番	大分類	小分類	項目	外部システムとの連携 (右記以外)	外部システムのうち、 刷新過渡期のToBe対象システムとの連携	
				利用目的	利用目的	刷新後の 実現方法
4	変換機能	データ変換		外部システムとの差異(ギャップ)を吸収するため、データ変換や通信プロトコル変換等を行う。 詳細は「(3)外部システム連携層が吸収するギャップ」を参照のこと。	ToBe対象システムとの差異(ギャップ)を吸収するため、データ変換や通信プロトコル変換等を行う。 詳細は「(3)外部システム連携層が吸収するギャップ」を参照のこと。	内部システムは、データ形式等を統一する。
5		通信プロトコル変換				
6		形式チェック		電文の形式をチェックする。	利用しない。 ※特許庁システム内での連携のため。	各システム構成要素のインタフェース層で形式チェックを行う。
7		リソースアクセス (共有データベースへのアクセス)		外部システムに連携する元データを共有データベースから取得する。又は、外部システムから連携されたデータを保存する。	ToBe対象システムに連携する元データを共有データベースから取得する、又はToBe対象システムから連携されたデータを保存する。 また、ToBe対象システムが共有データベースアクセスを必要とする場合、外部システム連携層を介してアクセスする。	基盤機能層を利用する。
8		アダプタ (パッケージ固有のプロトコルやホストと接続するための製品固有な機能)		ベンダロックインとなるため積極的には利用しないが、外部システムとの連携に必要な場合は、利用する。	原則、利用しない。 ただし、ToBe対象システムのプログラムプロダクトとの連携に必要な場合は、利用する。	内部システムは、データ形式等を統一する。

(2) 外部システム連携層が提供することを禁止する機能

以下に示す機能の提供は禁止する。

- BPMSのステータス取得機能を、外部システム連携層を介したデータサービスとすることを禁止する。

理由： BPMS製品によってインタフェースは異なり、汎用的なインタフェースを開発するのは困難であり、外部システム連携層を介した場合でも交換可能性には寄与しないと考えられるため。

- 内部システム向けに共有データベースへのアクセスをデータサービスとすることを禁止する。

理由： 共有データベースへのアクセスは、基盤機能層で行うため。

- (3) 外部システム連携層が吸収するギャップ
外部システム連携層が吸収するギャップを下表に示す。

表 3.2-59 外部システム連携層が吸収するギャップ

項番	ギャップ	吸収するギャップの内容
1	プロトコル	内部システムはアーキテクチャ標準仕様で定めるプロトコルを利用するが、外部システムでは多様なプロトコルが利用されるため、プロトコルの変換を行う。 内部システムのプロトコルの詳細は、「3.2.1.1.2 内部インタフェースのプロトコル」を参照のこと。
2	文字コード	内部システムはアーキテクチャ標準仕様で定める文字コードを利用するが、外部システムでは多様な文字コードが利用されるため、文字コードの変換を行う。 内部システムの文字コードの詳細は、「3.2.10 文字コードの扱いのルール」を参照のこと。
3	電文形式	内部システムはアーキテクチャ標準仕様で定める電文形式を利用するが、外部システムでは多様な電文形式が利用されるため、電文形式の変換を行う。 内部システムの電文形式の詳細は、「3.2.1.1.2 内部インタフェースのプロトコル」を参照のこと。
4	連携データ	内部システムはESBとの通信には基本的にキー情報のみを受け渡し多量のデータは共有データベースを介して受け渡す。 外部システムとデータをまとめて受け渡す場合は、外部システム連携層が連携データの分解又は統合を行う。
5	連携タイミング	内部システムで扱うデータは、基本的にリアルタイム連携による業務単件データである。 外部システムと連携するにあたり、業務単件データをまとめて連携する場合、外部システム連携層では以下の処理を行う。 ● 内部システムから外部システムへ連携する場合 外部システム連携層内に一時的に連携データを保持しておき、処理単位毎(時間等)に連携データを統合して連携する。⁴¹ ● 外部システムから内部システムへ連携する場合 外部システム連携層で連携データを単件単位に分割し、業務単件データ毎に順次、内部システムと連携する。
6	通信データ量	内部システムは連携時に受け渡すデータ量が小さい(基本的に連携データはデータベースに格納する)。 外部システムは連携時に受け渡すデータ量が大きい(連携に必要なデータを全て連携時の引数で受け渡す)。また、大量データ(メガバイトオーダー)の連携がある。 そのため、外部システム連携層では以下の処理を行う。 ● 内部システムから外部システムへの連携する場合 共有データベースに格納されている連携データを統合し、外部システムに送信する大量データを生成する。 ● 外部システムから内部システムへ連携する場合 外部システムから受信した大量データを共有データベースへ格納する。

⁴¹ 外部システム連携層から外部システムに連携を行う際の実行の契機は、運用監視システムが担ってもよい。

(4) ESBと外部システム互換機能の責務の分担

外部システム連携層で吸収するギャップに対して、ESBと外部システム互換機能のそれぞれの責務を下表に示す。

表 3.2-60 ESBと外部システム互換機能の責務

項番	ギャップ	ESB	外部システム互換機能
1	プロトコル	採用するESB製品が保有する機能で対応できるギャップの吸収処理を行う。	採用するESB製品のアダプタが対応していないプロトコルの受送信の機能を提供する。
2	文字コード		採用するESB製品が対応していない文字コードの変換機能を提供する。
3	電文形式		採用するESB製品の機能だけでは変換できない電文形式の変換機能を提供する。
4	連携データ		採用するESB製品の機能だけでは過不足を補完できない連携データの補完機能を提供する。
5	連携タイミング	—	連携タイミングのギャップに対応する機能を提供する。
6	通信データ量	—	通信データ量のギャップに対応する機能を提供する。

3.2.6 基盤機能層及びデータベース層のルール

基盤機能層とデータベース層の責務及びシステム構成要素を以下に示す。なお、本節のシステム構成要素を持つ共有データベース管理サブシステムの外部仕様及び利用方法等については、『DBアクセス基盤利用者向けガイドライン』に従うこと。

(1) 基盤機能層

A. 責務

基盤機能層の責務は、「データベース層へのアクセス」である。ビジネスロジックは基盤機能層の責務に含まれない。ここで言うビジネスロジックとは、業務毎にデータベースへの取得や応答の方法が異なるような処理⁴²のロジックのことであり、そういった性質の処理は、業務アプリケーション層の責務となる。基盤機能層の責務を担うための主な処理内容を以下に示す。

- 共有データベースのデータ取得及び更新を行うためのサービスインタフェースを提供する。
- 業務アプリケーション層及び外部システム連携層からデータベース層への処理リクエストを受け付ける。
- データベース層へアクセスし、要求されたデータの取得及び更新処理を実行する。
- データベース層へアクセスした結果を業務アプリケーション層及び外部システム連携層へ返却する。

B. システム構成要素

基盤機能層は、以下のシステム構成要素を持つ。

- DBアクセス基盤サービス
 - DBアクセス基盤サービスの責務は、「共有データベースにアクセスするためのサービスの提供」である。
 - DBアクセス基盤サービスは、業務アプリケーション層の3つのアプリケーション、外部システム連携層のESB及び外部システム互換機能からアクセスされることにより、業務処理を実行する。
 - 以下に列挙する場合に限り、他システム構成要素にアクセスしてもよい。
 - ✓ 業務アプリケーション層の3つのアプリケーション、外部システム連携層のESB及び外部システム互換機能から受信した電文に応じたデータベースアクセスを実行するために、共有データベースへアクセスする。

(2) データベース層

データベース層の責務及びシステム構成要素を以下に示す。

A. 責務

データベース層の責務は、「特許庁システム全体で共有すべきデータの管理」である。

データベース層の責務を担うために行う主な処理内容を以下に示す。

- 業務アプリケーション層、基盤機能層又は外部システム連携層(外部システム互換機能)からデータベース層への処理リクエストを受け付ける。
- データベース層からのレスポンスを業務アプリケーション層、基盤機能層又は外部システム連携層(外部システム互換機能)へ返却する。

B. システム構成要素

データベース層は以下のシステム構成要素で構成される。

- 共有データベース
 - 共有データベースの責務は、「特許庁システム全体で共有すべきデータの管理」である。
 - 管理するデータは、事件・書類データ及び特定サブシステム間共有データ⁴³である。
 - 業務アプリケーション層の3つのアプリケーション、DBアクセス基盤サービス、又は外部システム連携層の外部システム互換機能からデータベースがサポートするプロトコルでアクセスされる。

⁴² 例えば、未公開データを取得できないようにしたり、秘匿情報をマスキングするような処理は、業務によって異なる処理であり、ビジネスロジックに相当するため基盤機能層の責務には含まれない。

⁴³ 詳細は、『データ統合方針書』の「2.3 データ配置の考え方」を参照のこと。

3.2.6.1 共有データベースの構成及びアクセスルール

スコープ： 階層定型化サブシステム

プロセス： 2.4.2

規約1： 共有データベースの構成は、「(1)共有データベースの構成とデータ管理のルール」で示すデータの単位とすること。

規約2： 共有データベースへのアクセスは、「(2)共有データベースへのアクセスルール」に準拠すること。

スコープ： 階層定型化サブシステム

プロセス： 2.4.3

規約1： 共有データベースに一時的に保存される受け渡しデータは、受領するサブシステムが使用後に削除すること。

(1) 共有データベースの構成とデータ管理のルール

共有データベースに管理すべき共有データは、事件及び書類データを対象とする。また、事件及び書類データの単位で一元管理する。

事件及び書類データの単位には、以下のものがある。

- 出願事件
- 登録事件
- 国際出願事件
- 審判事件

また、上記の構成に基づいて、サブシステム間で受け渡しする必要がある場合、その受け渡しデータも共有データベースで管理する。共有データベースに一時的に保存される受け渡しデータは、特定サブシステム間共有データ⁴⁴に保持される。これは、連携のための一時的なデータであって、永続的に管理すべきデータではないため、受領するサブシステムが使用後に削除すること。

(2) 共有データベースへのアクセスルール

共有データベースへのアクセスルールを以下に示す。

- 以下のシステム構成要素からのアクセスのみを許可する。
 - 業務アプリケーション(ユーザ)
 - 業務アプリケーション(システム)
 - 業務アプリケーション(バッチ)
 - DBアクセス基盤サービス
 - 外部システム互換機能
- 以下に示す基盤APIを利用し、共有データベースにアクセスする。
 - 基盤API(データベース版)： 直接アクセスする。
 - 基盤API(サービス版)： DBアクセス基盤サービスを介してアクセス(データ取得のみ)する。
性能に問題がある場合は、特許庁と協議の上、基盤API(データベース版)を利用することを許容する。
- データの管理単位(出願事件、登録事件等)が異なる複数の共有データベースに対し、同一コネクションを使用したアクセスを禁止する。
- 情報活用系システムがETLを利用しアクセスすることを許容する。

なお、共有データベースへのアクセスルールの詳細は、『DBアクセス基盤 利用者向けガイドライン』に従うものとする。

⁴⁴ 詳細は、『データ統合方針書』の「2.3 データ配置の考え方」を参照のこと。

3.2.7 ビジネスルール管理層のルール

(1) 責務

ビジネスルール管理層の責務は、「ビジネスルールの管理, 実行」である。ビジネスルール管理層の責務を担うための主な処理内容を以下に示す。

- ビジネスルールを実行するためのサービスインタフェースを提供する。
- 業務アプリケーション層の3つの業務アプリケーションからアクセスされることにより, ビジネスルールの実行を行い, その結果を返却する。

(2) システム構成要素

ビジネスルール管理層は以下のシステム構成要素で構成される。

- BRMS

BRMSの責務は、「ビジネスルールの管理, 実行」である。なお, ビジネスルールを実行するためのサービスインタフェースの提供において性能面のリスクが高いと判断された場合は, BRMSのライブラリを業務アプリケーション(ユーザ), 業務アプリケーション(システム), 業務アプリケーション(バッチ)に配付し, BRMS製品が提供するAPIを利用することにより, 通信によって発生するオーバーヘッドを回避する。

3.2.7.1 ビジネスルールのルール

ビジネスルールとは、手続的な処理は持たず実行順序によらない宣言的な記述が可能なものである。ビジネスルールには、以下のようなものがある。

- 「～の場合、…とみなす」というような推論を表すもの
- 「～から、…を算出する」というような計算を表すもの
- 「～の場合、…をする」というような振分を表すもの
- 「～の場合のみ…できる」というような制約を表すもの

データベースやファイルへのアクセス、ネットワークを介した他サブシステムへアクセス等、システムリソースを扱う処理はビジネスルールの実現上必要な手続きであり、ビジネスルールではない。保守性の観点から原則、ビジネスルールとは区別して管理する。

ただし、ごく簡単なビジネスルール（計算式等）とデータベースアクセス等が入り混じった（そのような形にせざるを得ない）手続きも存在し得る。このような手続きに関して無理にビジネスルールとデータベースアクセス等を分けて管理するとかえって保守性が悪くなる（規模の増加や解析性が低下する）ため、同一の実装手段で統一するものとする。

ビジネスルールを組み合わせた手続的な処理をビジネスロジックという。例を以下に示す。

- ① データベースから取得したデータを変数に代入
- ② ①の変数を利用し、「～の場合、…とみなす」というような推論を表すビジネスルールを実行
- ③ ②の結果から、「～の場合、…をする」というような振分を表すビジネスルールで処理を分岐
- ④ 処理結果を基にデータベースを更新

③のように簡単な振分を表すビジネスルールは、ビジネスロジックから切り離さない方がよいケースもある。

「振分」、「制約」のビジネスルールについては、フロー図での分岐で表現されるもので、ビジネスロジックで扱うレベルのビジネスルールであれば、処理の流れで出現する分岐（プログラムコードであればif文）で表現されるものである。一方、ビジネスプロセスで扱うレベルの「振分」、「制約」のビジネスルールは、業務フロー図の分岐（BPMNであれば排他ゲートウェイ）として表現され、BPMSで管理されるものである。

ビジネスルールの条件部又は判定部（例えば、「～の場合、…とみなす」の「～」又は「…」の部分）には、数値、文字列、日付等の固定値が現れる。これらをパラメータと呼び、以下の特徴を持つ。

- ビジネスルール管理層、業務アプリケーション層だけでなく、プレゼンテーション層、ワークフロー層等業務アプリケーション全体に現れる。
- 業務処理の実行によって変更されず、運用業者又は保守業者によって変更される。
- 数値、文字列、日付等の単純な値や、それらの組み合わせがパラメータの候補となる。

ビジネスルールと、ビジネスロジック、ビジネスプロセス、パラメータの関係を概念的に図示化したものを以下に示す。

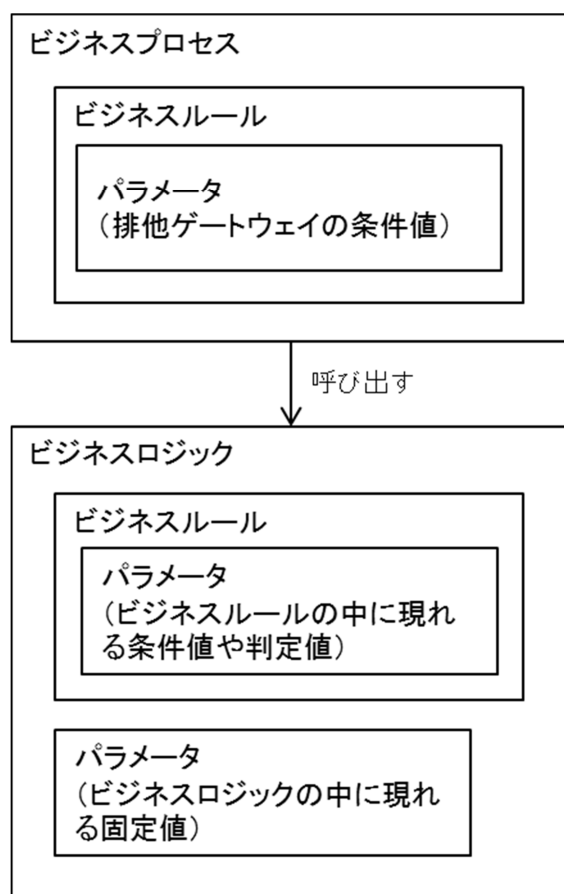


図 3.2-47 ビジネスルールと、ビジネスロジック、ビジネスプロセス、パラメータの関係

3.2.7.1.1 ビジネスルールをアプリケーションから切り離すための仕組みとして利用する技術

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム(ビジネスルール管理)
プロセス：2.4.2
規約1： ビジネスルールをアプリケーションから切り離すための仕組みは、BRMSを用いること。

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム(ビジネスルール管理)
プロセス：2.4.3
規約1： BRMSによるビジネスルールの記述は、「(3)BRMSによるビジネスルールの記述ルール」に示すとおり記述すること。
指針1： ビジネスルールをアプリケーションから切り離すための仕組みの適用方法は、「(4)コンポーネント機能による依存関係の制御」に従って設計すること。
指針2： BRMSの適用は、「(2)BRMSを用いたビジネスルールの記述対象」に従って設計すること。
特例1： ビジネスルールは、「(2)BRMSを用いたビジネスルールの記述対象」に示す特例を許容する。

(1) ビジネスルールをアプリケーションから切り離すための仕組み

ビジネスルールをアプリケーションから切り離すための仕組みとして「BRMS」を利用する。

(2) BRMSを用いたビジネスルールの記述対象

BRMSを利用すると保守性が下がるケースもあるため、以下の2つの特性を合わせ持つビジネスルールを対象に「推論」、「計算」のビジネスルールを記述する。

- 複数の「定型的な判定条件群と、結果」の組み合わせが多数現れるビジネスルール
- 変更が発生しやすいビジネスルール、又は複数箇所から利用されるビジネスルール

例えば、入力チェック処理における、入力データ単体で完結するチェック処理(データの性質に基づくチェック等)、あるいは入力データ間の相関関係に基づく単純なチェック処理には、BRMSを適用しない。

「振分」、「制約」のビジネスルールについては、アプリケーションから切り離さず、業務アプリケーションやBPMSで記述するものとする。

● 業務アプリケーション

ビジネスロジックレベルの「振分」及び「制約」は、プログラムコード上の分岐(if文等)で記載する。

特例として、BRMS製品によってはルールのフロー制御機能を持っている場合、ルールの可視性を低下せず、各ルールの再利用性が妨げられない範囲においてのみ、フロー制御機能の使用を許容する。

● BPMS

ビジネスプロセスレベルの「振分」及び「制約」は、BPMNの排他ゲートウェイで記載する。

排他ゲートウェイの詳細は「3.2.3.2 BPMNの記載ルール」を参照のこと。

BRMS及び業務アプリケーション、BPMSの使い分けのイメージを下図に示す。

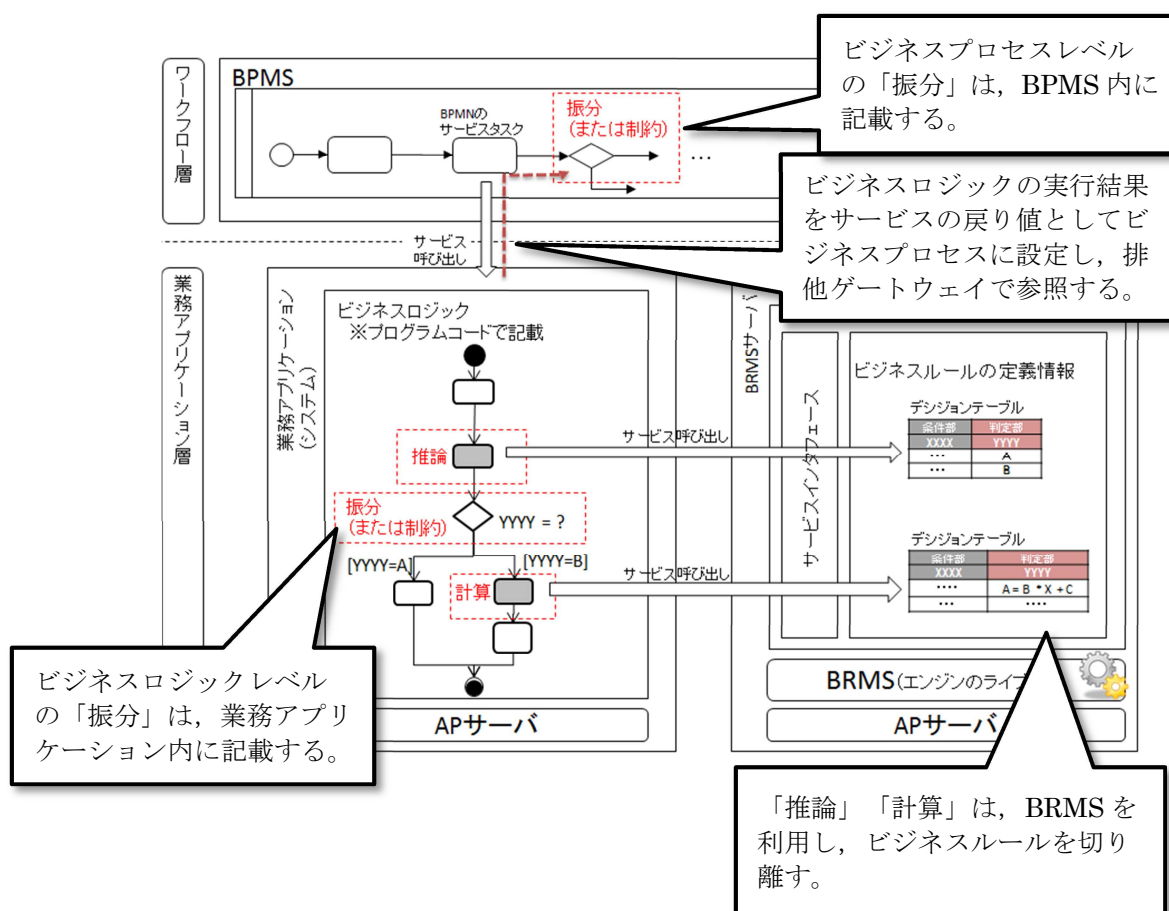


図 3.2-48 BRMS及び業務アプリケーション、BPMSの使い分けのイメージ

(3) BRMSによるビジネスルール記述ルール

BRMSによるビジネスルールの記述ルールを以下に示す。

- BRMSから直接データベースにアクセスしデータを取得することを禁止する。データベースからのデータ取得は、BRMSの呼び出し元である業務アプリケーションで行い、BRMSへ渡すこと。

(4) コンポーネント機能による依存関係の制御

- BRMSで管理しないビジネスルールであっても、インタフェースを変えずにビジネスルールを実装するコンポーネントを差し替え可能にして(DIコンテナの利用等), ある程度、保守性を担保できるような設計上の工夫は実施すること(DIコンテナの利用等の詳細は、「3.2.4.4.2 特許庁開発フレームワークの機能」を参照のこと)。
- 特に、対象書類の出願日等によって適用される法律の新旧が変わるビジネスルールについては、旧法に基づくビジネスルールが不要になったときに、設定ファイルの変更等、他のビジネスルールに影響を与えない方法で破棄できるようにすること。

3.2.7.1.2 BRMSの呼び出し方式

スコープ：階層定型化サブシステム

プロセス：2.4.3

規約1： BRMSの呼び出し方式は、「(2)BRMSの呼び出し方式の選択ルール」のとおり選定すること。

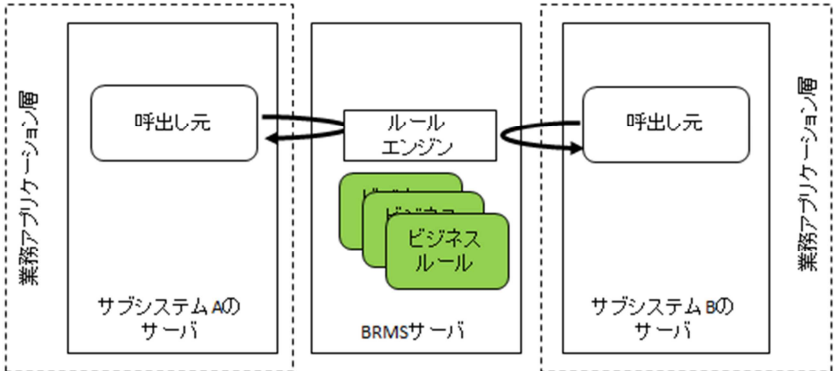
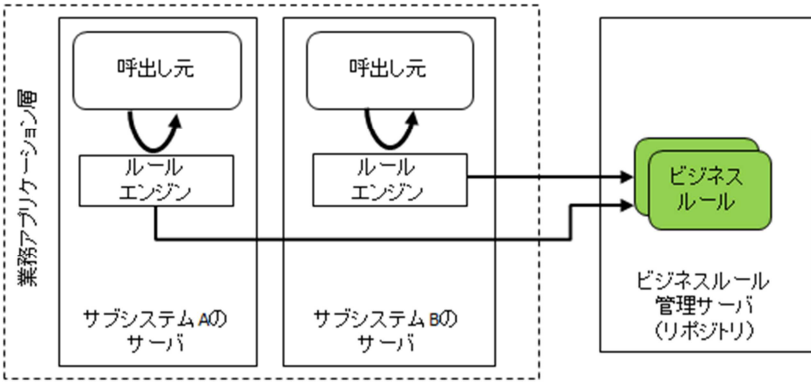
特例1： 性能面のリスクが高いと判断した場合は、「ライブラリ配備＋リポジトリ方式」を許容する。

規約2： BRMSを利用する階層定型化サブシステムの設計に参与するステークホルダ(設計・開発ベンダ等)の開発対象は、「(3)BRMSの開発対象」に示すとおりとすること。

(1) BRMSの呼び出し方式

BRMSの呼び出し方式を下表に示す。

表 3.2-61 BRMSの呼び出し方式

項番	呼び出し方式	説明
1	サービス方式	ビジネスルールの実行をサービスとして提供する。 多階層構造上, BRMSはインタフェース定型化サブシステムに位置づける。 
2	ライブラリ配備方式 ＋リポジトリ方式	ルールエンジンをサブシステムのAPサーバ毎に配備し, ビジネスルールはリポジトリとしてまとめて管理する。 多階層構造上, BRMSは業務アプリケーション層の各システム構成要素のコンポーネントとして位置づける。 

(2) BRMSの呼び出し方式の選択ルール

レスポンス低下といった性能面で懸念があるものの移植性と費用面のメリットを重視し、ToBeアーキテクチャでは、原則、「サービス方式」とする。

ただし、要件定義及び設計工程において性能面の問題を調査し、リスクが高いと判断した場合は、「ライブラリ配備＋リポジトリ方式」を許容する。

(3) BRMSの開発対象

BRMSサーバ又はビジネスルール管理サーバは、階層定型化サブシステム内に個別に設けず、特許庁システム内で共通的に使用されるものであるため、各階層定型化サブシステムの設計に関与するステークホルダ(設計・開発ベンダ等)は、以下の機能を開発するものとする。

- BRMSサーバ又はビジネスルール管理サーバに登録するビジネスルール
 - BRMSサーバが提供するルールエンジン呼び出すためのインタフェース定義
- ※インタフェースを利用しビジネスルールを実行するためのフレームワークは、BRMSサーバ又はビジネスルール管理サーバの設計に関与するステークホルダ(設計・開発ベンダ等)が開発するものとする。

3.2.8 データ設計に関するルール

3.2.8.1 用語のルール

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.2

指針1：用語は、「(1)用語の種類」に示す種類を使用すること。

指針2：用語の表記は、「(2)用語の表記」に示す表記ルールに従い、表記すること。

規約1：用語の整備は、「(3)A.整備規約」に示す規約のとおり整備すること。

指針3：用語の整備は、「(3)B.整備指針」に示す指針を遵守すること。

データベースのテーブル名やインタフェース名等の論理名に使用する用語のルールを以下に示す。

(1) 用語の種類

用語の種類を下表に示す。

表 3.2-62 用語の種類

項番	用語の種類	概要	例
1	特性名	データの目的や特性を表す用語。	「先願」、「拒絶理由」、「書類」、「願書受付」等。
2	区分語	「コード」、「番号」、「区分」、「フラグ」	左記のとおり。
3	修飾子	修飾の対象となる特性名や区分語、値ドメイン名(後述)についてその意味内容を詳細に説明するために用いられる用語。	● コードの種類を表す「四法」等。 ● 補足説明する「手運用」や「担当職員」等。 ● グループを表す「変更前」、「変更後」等。 ● データ型を表す「一覧」等。
4	動作名	業務的な動作を表す用語。	「生成」、「チェック」等。
-	値ドメイン名 ⁴⁵	特性名のうちデータが持つ意味や性質、値の制約(取り得る値)が規定されている用語。コード系と非コード系からなる。	-
5	コード系	「修飾子＋区分語」で構成される用語。	「四法コード」、「納付方法区分」等。
6	非コード系	業務的に重要な意味や利用制約を有する用語。	「出願日」、「決裁日」、「優先権主張日」等。

(2) 用語の表記

用語の表記ルールを以下に示す。

- アルファベット、算用数字、アンダースコア(「_」)は、半角とすること。それ以外は、全て、全角とすること。
- 業務として意味がある場合を除き、送り仮名は省略すること。
例：「取り扱い」→「取扱」

⁴⁵ 値ドメインについては、『データ統合方針書』の「3.1.1.2.3 値ドメイン管理」及び「3.1.1.2.4 データ辞書構築」参照のこと。

(3) 用語の整備

A. 整備規約

用語の整備規約を以下に示す。

- 共有データベースや公開されるインタフェースで利用される用語は、特許庁全体で一元管理される用語集を確認し、重複や同音異義・異音同義語の排除すること。
- サブシステム内で固有の用語は、サブシステムの責任において、サブシステム内で一元管理し、サブシステム内の重複や同音異義・異音同義語を排除すること。

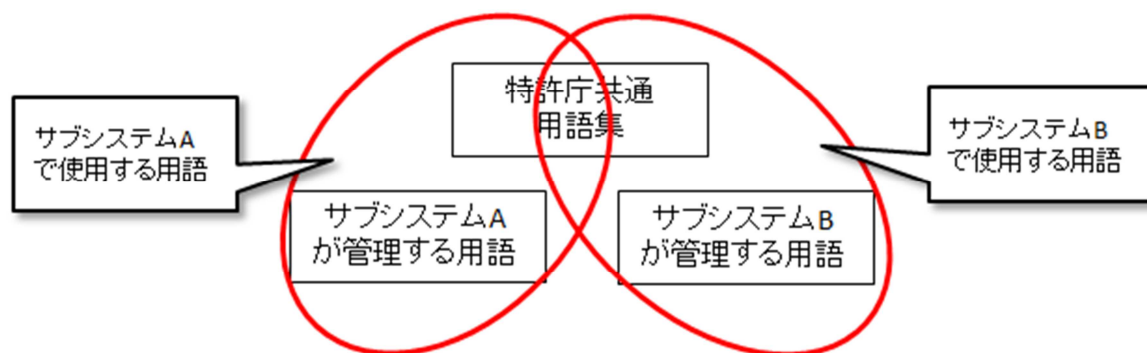


図 3.2-49 サブシステムで使用する用語の関連

B. 整備指針

用語の整備指針を以下に示す。

- 用語は、意味を定義し、誤った意味で用語が利用されないようにすること。
用語定義の出典や参照先として、法令や標準規格等があれば合わせて記載すること。
- 外部環境の変化に影響されにくい用語を定義すること。
外部環境の変化に影響されにくい用語には、「四法コード」や「出願日」等、値ドメインとして管理される名称等が考えられる。また、外部環境の変化に影響されやすい用語には、プログラムプロダクト名等が考えられる。

3.2.8.2 論理データベース設計の際の名称の標準化ルール

サブシステムにおいて誤解せず概念データモデル定義、論理データベース設計できるようにすることを目的に、以下に対し、論理名の共通的な命名ルールを定める。

- テーブル名
- カラム名

3.2.8.2.1 テーブル名の命名ルール

スコープ:	階層定型化サブシステム及びインタフェース定型化サブシステム(外部システム連携)
プロセス:	2.4.2, 2.4.3
指針1:	テーブル名は、「(1)テーブル名の構成」に示す用語の組み合わせに従い、命名すること。
指針2:	テーブル名は、「(2)テーブル名の重複と同音異義・異音同義語の禁止事項」に示す禁止事項に抵触しないよう、命名すること。

(1) テーブル名の構成

テーブル名は、以下の用語の組み合わせで構成すること。

テーブル名 = 特性名 + 修飾子(0個以上)

※1 特性名や修飾子等の用語の種類については、「3.2.8.1 用語のルール」を参照のこと。

※2 各用語の区切り文字は、アンダースコア(「_」)する。必要に応じて、追加で区切り文字を定義してもよい。その場合は、サブシステムで利用方法を統一すること。

※3 修飾子を複数使う場合は、「特性名 + 修飾子」を修飾する場合等に用いる。

テーブル名の例を下表に示す。

表 3.2-63 テーブル名の例

項番	パターン	テーブル名の例
1	特性名	「代理人」、「書類」、「手続補正」等
2	特性名 + 修飾子	「書類_中間手続」、「手続補正_方式審査」等
3	特性名 + 修飾子 + 修飾子	「手続補正_方式審査_担当職員」等

(2) テーブル名の重複と同音異義・異音同義語の禁止事項

サブシステムに対しての共有データベース、個別データベースの関係性を考慮し、テーブル名の重複と同音異義・異音同義語の使用を下表に示すとおり、禁止する。

表 3.2-64 重複と同音異義・異音同義語の禁止部分(テーブル名)

項番	配置関係	重複	同音異義・異音同義語
1	同一のデータベース内	禁止	禁止
2	異なる共有データベース間		
3	共有データベースと個別データベース間		
4	異なる個別データベース間	許容	許容

3.2.8.2.2 カラム名の命名ルール

スコープ:	階層定型化サブシステム及びインタフェース定型化サブシステム(外部システム連携)
プロセス:	2.4.2, 2.4.3
指針1:	カラム名は、「(1)カラム名の構成」に示す用語の組み合わせに従い、命名すること。
指針2:	カラム名は、「(2)カラム名の重複と同音異義・異音同義語の禁止事項」に示す禁止事項に抵触しないよう、命名すること。

(1) カラム名の構成

カラム名は、以下の用語の組み合わせで構成すること。

カラム名 = テーブル名(※1) + 特性名(任意) + 値ドメイン名 + 修飾子(任意)

※1 テーブル名は、カラム名の重複を排除する目的で、テーブル名をつけることで、カラムの意味を適切に表現できる場合に設定する。

※2 テーブル名については、「3.2.8.2.1 テーブル名の命名ルール」を参照のこと。

※3 特性名や値ドメイン名、修飾子等の用語の種類については、「3.2.8.1 用語のルール」を参照のこと。

※4 各用語の区切り文字は、アンダースコア(「_」)とする。必要に応じて、追加で区切り文字を定義してもよい。その場合は、サブシステムで利用方法を統一すること。

カラム名の例を下表に示す。

表 3.2-65 カラム名の例

項番	パターン	カラム名の例
1	値ドメイン名	「四法コード」,「起案日」,「IPC」,「記部連番」等
2	特性名 + 値ドメイン名	「先願有無フラグ」,「拒絶理由記部連番」等
3	値ドメイン名 + 修飾子	「IPC_サブクラス」,「受付番号_手運用」等

(2) カラム名の重複と同音異義・異音同義語の禁止事項

サブシステムに対しての共有データベース、個別データベースの関係性を考慮し、カラム名の重複と同音異義・異音同義語の使用を下表に示すとおり、禁止する。

表 3.2-66 重複と同音異義・異音同義語の禁止部分(カラム名)

項番	配置関係	重複	同音異義・異音同義語
1	同一テーブル内	禁止	禁止
2	同一データベース内の異なるテーブル間		
3	異なる共有データベース間		
4	共有データベースと個別データベース間		
5	異なる個別データベース間	許容	許容

3.2.8.3 XMLとSGMLのデータの扱いのルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.3

指針1： XMLやSGMLの書類形式を意識した業務処理，データ構造は，「図 3.2-50 書類形式を意識する層の範囲」に示す範囲に配置するよう設計すること。

指針2： 書類形式に依存した処理は，SGML形式を扱うコンポーネントが，将来，不要となる可能性を考慮して，XMLとSGMLに分けてコンポーネント化されるよう設計すること。

推奨1： XML形式及びSGML形式を扱うコンポーネントの切り替え，切り離しには，DI(Dependency Injection: 依存性注入)等を用いて，依存関係を制御する仕組みを利用することを推奨する。

特許庁システムで扱う書類形式は，XML及びSGMLがあるが，SGMLの書類は，将来XML化する方針である。ただし，既に蓄積済みのSGMLはXML化せず，そのまま用いることが考えられる。文書形式の移行方針については，『特許庁システム移行方針書』を参照のこと。

ToBeアーキテクチャにおいては，以下に示すXML及びSGMLのデータの扱いのルールに従うこと。

- XMLやSGMLの書類形式を意識した業務処理（書類形式の判定処理，スキーマチェック，パース処理等）及びデータ構造は，原則，多階層構造における以下の層に配置し，上位層に対して書類形式を隠蔽すること。

- 業務アプリケーション層
- 基盤機能層
- データベース層
- 外部システム連携層（外部システムと内部システムとの連携時に扱う書類形式が異なる場合のみ）
- ビジネスルール管理層

ただし，書類形式を保持したまま，画面での処理を行う業務要件がある場合には，UI層，プレゼンテーション層での処理を許容する。

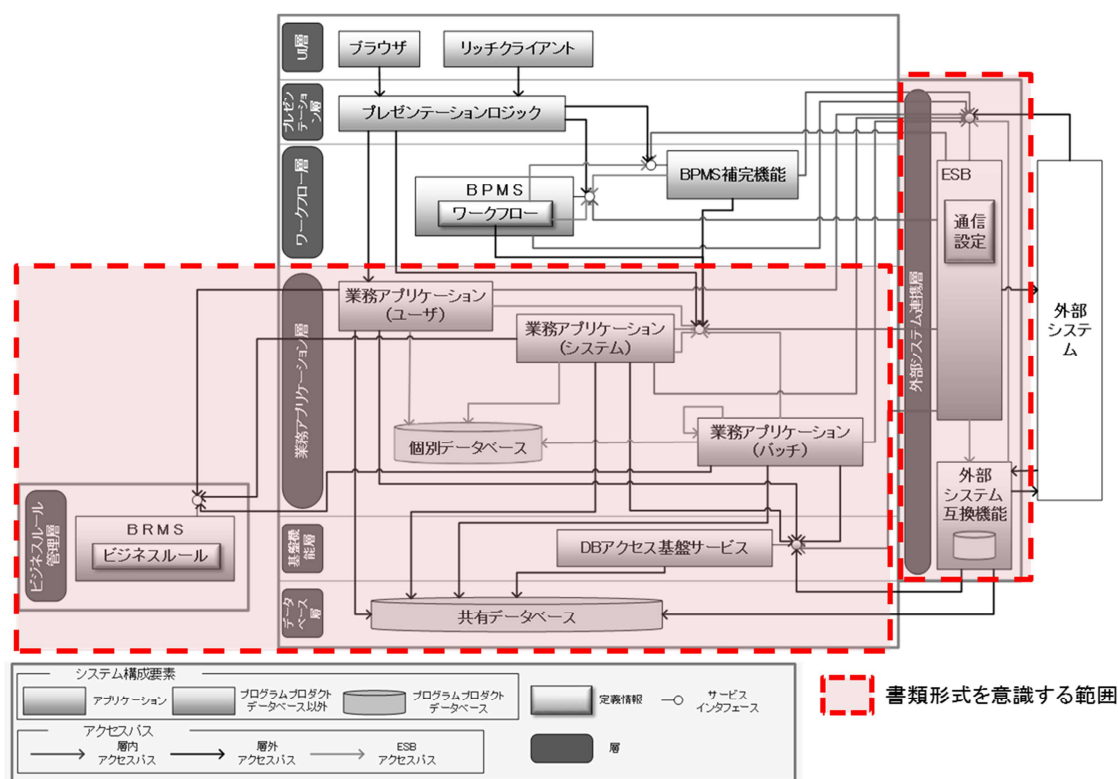


図 3.2-50 書類形式を意識する層の範囲

- 書類形式に依存した処理は、XMLとSGMLに分けてコンポーネント化すること。

SGML形式を扱うコンポーネントは、将来、不要となる可能性がある。また、XMLとSGMLの処理を容易に切り替え、切り離しが可能なように、XMLやSGMLの書類形式に依存した処理は、XMLやSGML混在したコンポーネントにしないこと。

また、切り替え、切り離しには、DI(Dependency Injection:依存性注入)等を用いて、依存関係を制御する仕組みを利用することを推奨する(依存関係を制御する仕組みについては、「3.2.4.4 特許庁開発フレームワーク」を参照のこと)。

同じ書類の処理等、コンポーネントを共通化できる場合、以下の方法で共通化が可能である。

- ① サブシステム間で使用される場合、サブシステム間共通機能の共有ロジックである共有ビジネスロジック(業務特有機能)として共通化する。詳細は、「3.2.4.3 サブシステム間共通機能」参照のこと。
- ② サブシステム内で使用される場合、特許庁フレームワークのサブシステム固有機能として共通化する。詳細は、「3.2.4.4.2 特許庁開発フレームワークの機能」参照のこと。

3.2.8.4 特別な構造を持つデータの取り扱い方法

特許庁業務が持つ特性や、新しいタイプの出願への対応等により、扱うデータによっては、データ設計にあたって特別な考慮が必要な場合も考えられる。今後想定される環境変化の影響を受け、既存のデータ構造から見直す必要があることが予想されるものがある。

本章節では、以下の個別の三例を想定し、エンティティの設計において考慮すべきデータの扱い方を定める。

- 多件一通の手続に関するデータの扱い
- デジタルコンテンツに関するデータの扱い
- 主キーが決定していないエンティティの扱い

なお、テーブルの設計に関しては、本章節にて定めるエンティティの設計におけるデータの扱い方を踏まえ、処理都合や非機能要件等の制約を加味して行う。

3.2.8.4.1 多件一通の手続に関するデータの扱い

スコープ：階層定型化サブシステム

プロセス：2.4.2

指針1：一つの書類が複数の事件等に関連する手続となる業務要件の実現は、その手続きを表すエンティティについて、「表 3.2-67 多件一通の手続に関するデータの扱い方針」に示す方針に従い、データ構造を設計すること。

多件一通の手続とは、一通の手続書類が複数の事件等を対象とした手続を指す。このような、各事件を特定する番号に一意に紐付かない多件一通の手続を管理するデータの構造については下表に示すとおりとする。

表 3.2-67 多件一通の手続に関するデータの扱い方針

項番	要件	方針
1	多件一通の手続に関するデータの扱い	手続書類を管理するエンティティと対象事件等を管理するエンティティを紐付ける、手続の明細を管理するエンティティを設け、それぞれ適切な粒度で管理できるデータ構造とすること。

データ構造のイメージを下図に示す。図中において、手続書類を管理するエンティティを「中間手続」、対象事件を管理するエンティティを「出願事件」、両者を紐付け手続の明細を管理するエンティティを「中間手続__対象出願事件」として表している。

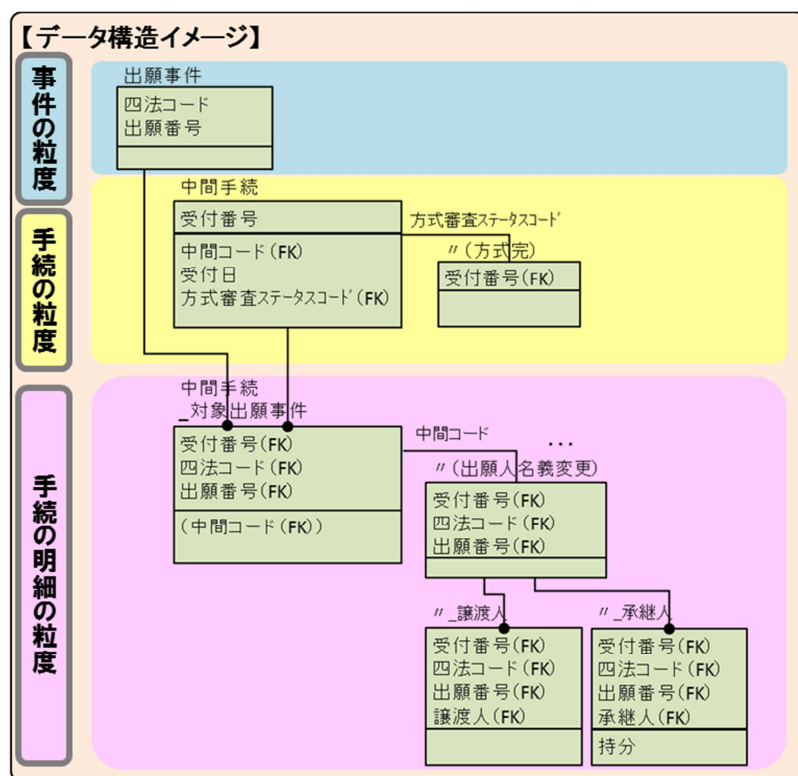


図 3.2-51 多件一通の手続のデータ構造(出願人名義変更届の例)

3.2.8.4.2 デジタルコンテンツに関するデータの扱い

スコープ：階層定型化サブシステム

プロセス：2.4.2

指針1： デジタルコンテンツを管理するエンティティは、「表 3.2-68 デジタルコンテンツに関するデータの扱い方針」に示す方針に従い、データ構造を設計すること。

新しいタイプの商標（動き、音等）等のデジタルコンテンツを出願時に受け付けることが想定されるため、扱いを決めておく必要がある。デジタルコンテンツに関するデータの管理ルールについては、下表に示すとおりとする。

表 3.2-68 デジタルコンテンツに関するデータの扱い方針

項番	要件	方針
1	デジタルコンテンツに関するデータの扱い	- 画像データや音源データ等のデジタルコンテンツデータは、特許庁の行政サービスを遂行する上で各業務から共通的に使用される共通リソースデータとして捉え、エンティティとして一元管理するものとする。 - 画像データや音源データ等を表すエンティティは、一元管理されたエンティティのサブタイプとして定義するものとする。

データ構造のイメージを下図に示す。図中において、デジタルコンテンツを一元管理するエンティティとして「コンテンツ」、「コンテンツ」エンティティのサブタイプとして「図形商標」、「音源」を表している。

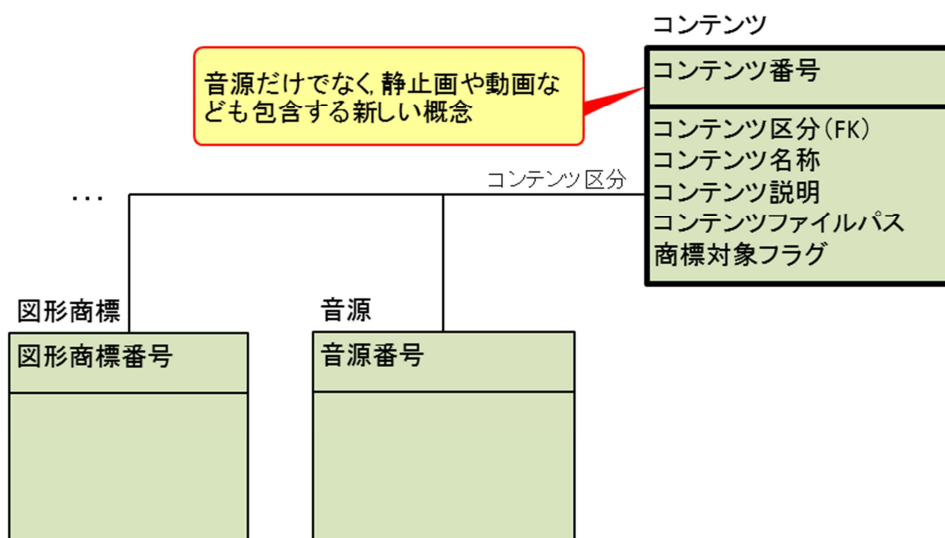


図 3.2-52 デジタルコンテンツを管理するデータ構造例

3.2.8.4.3 主キーが決定していないエンティティの扱い

スコープ：階層定型化サブシステム

プロセス：2.4.2

規約1： エンティティは、主キーが決定していない場合、必ず主キーを設定すること。

指針1： 主キーの設定は、「表 3.2-69 主キーの設定条件」に示す条件を満たすように設計すること。

主キーが決定していないエンティティに対しては、必ず主キーを設定すること。主キーについては下表に示す条件を満たすこと。

表 3.2-69 主キーの設定条件

項番	主キーの設定要件	説明
1	一意性	対象を一意に識別できること。
2	安定性	長い年月で一度設定した名称と値が変化しないこと。 例えば、体系に意味を含めない連番等。
3	簡潔性	より少ない項目で構成されていること。 例えば、顧客ID、受注ID等。
4	平易性	必要以上に長くないこと。 例えば、連番や固定桁数の文字列であること。

3.2.8.5 処分が決定した事件データの扱いのルール

スコープ： 階層定型化サブシステム(共有データベース管理)

プロセス： 2.4.2

指針1： 処分が決定した事件データは、「表 3.2-70 処分が決定した事件データの扱い方針」に示す方針に従い設計すること。

利用頻度が低い又は全く利用しないデータを、未整理のままデータベースで管理することによって、データ量の増加に伴うレスポンスの低下や維持管理コストの増加を招くおそれがあるため、それらのデータを適切に管理するためのルールを下表に定める。

表 3.2-70 処分が決定した事件データの扱い方針

項番	要件	方針
1	処分が決定した事件データの扱い	● データの種類毎に保存期間を設け、保存期間を超過したデータをデータベースから削除する仕組みを用意すること。 ● 事件の権利消滅後の保存期間内において、当該事件に対して遡及手続されることを考慮し、権利消滅前と同様の配置とすることでアプリケーションからのデータアクセス方法を統一すること。

なお、処分が決定した事件データとは、例えば以下のものを指す。

- 最終処分が決定した事件のデータ
 - 権利の有効期限が過ぎた事件のデータ。
- 関連する制度が終了したデータ
 - 「出願公告制度」に関するデータ(平成8年に廃止)等。

3.2.9 プログラムプロダクトのルール

プログラムプロダクトのうち、ルールを定める必要があるプログラムプロダクトとシステム構成要素の関係を下表に示す。⁴⁶

プログラムプロダクトの論理構成における配置は、「3.1.1.3.2 システム構成要素のソフトウェア構成」を参照のこと。

表 3.2-71 プログラムプロダクト一覧

項番	プログラムプロダクト	システム構成要素
1	BPMS	BPMS
2	BRMS	BRMS
3	ESB	ESB
4	Webブラウザ	ブラウザ
5	Webサーバ	プレゼンテーションロジック 業務アプリケーション(ユーザ)
6	APサーバ	業務アプリケーション(ユーザ) 業務アプリケーション(システム) DBアクセス基盤サービス
7	FTPサーバ	業務アプリケーション(バッチ)
8	ジョブ管理エージェント	業務アプリケーション(バッチ)
9	帳票設計・印刷ソフトウェア	業務アプリケーション(ユーザ) 業務アプリケーション(システム) 業務アプリケーション(バッチ)
10	DBMS	個別データベース 共有データベース
11	認証プラグイン/ 個人認証用ソフトウェア	ブラウザ リッチクライアント

※個別の業務要件に応じて採用するプログラムプロダクトは上記には含めていない。

例：OCRソフトウェアやOCR向けスキャナ制御ソフトウェア等

※外部システムで利用している製品に依存するため上記には含めていない。

例：MQやTP1等

⁴⁶ ルールを定める必要があるプログラムプロダクトとシステム構成要素の関係であり、表に含まれていないプログラムプロダクトの採用を排除するものではない。ただし、システム構成要素を実現する上で、本書に示すアーキテクチャ標準仕様を逸脱してはならない。

3.2.9.1 BPMSのルール

スコープ： 階層定型化サブシステム

プロセス： 2.4.3

指針1： BPMSは、「(1)BPMSの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。

推奨1： ビジネスプロセスの進行状況と実行履歴を抽出及び加工して、リアルタイムでグラフや表形式等で画面表示するための実現手段については、BPMSが保有するBAM(Business Activity Monitoring)を推奨する。

指針2： BPMSは、「(3)BRMSの適用範囲」に示す範囲で適用するように設計すること。

(1) BPMSの機能要件

A. TRM⁴⁷の定義から必要とされる機能要件(基本事項)

- ビジネスプロセスのデータや制御フローを定義するために必要なデータ操作機能を提供していること。
- サービスの呼び出し・データ操作・障害通知・例外処理・プロセスの終了・サブプロセス化等を組み合わせることでプロセスを定義できること。
- サービスとのやり取りを行うための標準(Web サービスプロトコル等)又は実装規約に基づいて管理が行えること。

B. TRMの定義から必要とされる機能要件(加点事項)

加点事項無し。

C. 特許庁固有の機能要件

- BPMN2.0に準拠していること。
理由：「3.2.3.1.1 ビジネスプロセス管理として利用する技術」を参照。
- ビジネスプロセスの進行状況が、BPMNのプロセス図で確認できること(例えば、BPMNのプロセス定義がBPELに変換して実行される場合でも、プロセスの進行状況が変換後のBPELではなく、対応するBPMNのプロセス図上で確認できること)。
理由： 開発工程の進行に伴うプロセスの詳細化に対してトレーサビリティを保持できるようにするため。「3.2.3.2 BPMNの記載ルール」を参照。
- ワークフローのバージョン管理機能を持ち、新旧並行運用や、起動中のビジネスプロセス変更が可能であること。
理由： ワークフローの開始から終了までの期間は、長期にわたる場合があるため。
- 外部から指示によりフロー実行可能であること。
理由： 他システム構成要素からのアクセスを実現するため。アクセスパスについては、「3.1.1.3.1.2 システム構成要素間のアクセスパス」を参照。
- 「表 3.2-2 プロトコル及び電文形式(階層定型化サブシステム)」及び「表 3.2-3 プロトコル及び電文形式(インタフェース定型化サブシステム)」に記載されているプロトコル、電文形式に対応していること。
理由： ToBeアーキテクチャで利用するプロトコル及び電文形式を標準化するため。
- ビジネスプロセスの進行状況と実行履歴を管理し、それらの情報を取得することが可能なこと。
理由： 「3.2.13.4 システム監視方式設計のルール」を参照。
- ビジネスプロセスの進行状況と実行履歴を抽出及び加工して、リアルタイムでグラフや表形式等で画面表示することが可能であること。
理由： 「3.2.13.4 システム監視方式設計のルール」を参照。
推奨： BPMSが保有するBAM(Business Activity Monitoring)の機能を利用することを推奨する。

⁴⁷ 技術参照モデル(Technical Reference Model) のこと。詳細は以下を参照のこと。

<http://www.ipa.go.jp/osc/trm/>

また、「TRMの定義から必要とされる機能要件」については、『情報システム調達のための技術参照モデル(TRM)平成24 年度版 物品調達編』から引用している。

- ビジネスプロセスの停止・再開に時間がかかり、システム運転時間内にバックアップ等のシステム運転が完了しない可能性がある場合、オンラインバックアップが可能であること。
理由：「3.2.13.3 システム運転管理方式設計のルール」を参照。

(2) BPMSの機能要件の補足

無し。

(3) BPMSの適用範囲

- サブシステムにビジネスプロセスがある場合、BPMSを適用する。ビジネスプロセスがない場合は、BPMSを適用しない。
- 「3.2.3.1.2 BPMSの適用範囲に関する設計指針」に示すケースに該当するビジネスプロセスは全てに対してBPMSを適用せず、業務アプリケーションと連携して実現する。
- 複雑な条件に基づくビジネスプロセスの分岐判定は、BPMSを適用せず、ビジネスルール化し、BRMSを適用する。

3.2.9.2 BRMSのルール

スコープ： インタフェース定型化サブシステム(ビジネスルール管理)

プロセス： 2.4.3

指針1： BRMSは、「(1)BRMSの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。

指針2： BRMSは、「(3)BRMSの適用範囲」に示す範囲で適用するように設計すること。

特例1： BRMSは、「(3)BRMSの適用範囲」に示す特例を許容する。

(1) BRMSの機能要件

A. TRMの定義から必要とされる機能要件(基本事項)

TRMに該当する技術ドメインがないため、基本事項無し。

B. TRMの定義から必要とされる機能要件(加点事項)

TRMに該当する技術ドメインがないため、加点事項無し。

C. 特許庁固有の機能要件

- ビジネスルールを定義する情報とその実行エンジンが分離されていること。

理由： ビジネスルールの定義情報を一元管理するため。詳細は「3.2.7.1.2 BRMSの呼び出し方式」を参照。

- 外部仕様として記載したビジネスルールがプログラムとして実行可能なものであること。実行にあたって追加情報が必要な際は、ビジネスルールの記載とは別の形で定義できるようになっていること。

理由： 外部仕様としてのビジネスルールと、実行のために必要な設計情報との対応付けを取りやすくするため。

- 実行時において、一度の実行で複数個のビジネスルールの定義を実行することが可能なこと。

理由： 設計時に人間が管理しやすい単位に分割したビジネスルールの定義を実行するため。

- 「表 3.2-3 プロトコル及び電文形式(インタフェース定型化サブシステム)」に記載されているプロトコル、電文形式で通信が可能であること。

理由： ToBeアーキテクチャで利用するプロトコル及び電文形式を標準化するため。

(2) BRMSの機能要件の補足

無し。

(3) BRMSの適用範囲

- 以下の2つの特性を合わせ持つビジネスルールを対象に「推論」、「計算」のビジネスルールの定義及び実行にBRMSを適用する。

➢ 複数の「定型的な判定条件群と、結果」の組み合わせが多数現れるビジネスルール。

➢ 変更が発生しやすいビジネスルール、又は複数箇所から利用されるビジネスルール。

- 「振分」、「制約」のビジネスルールの定義及び実行については、BRMSを適用せず、業務アプリケーションやBPMSを適用する。特例として、BRMS製品によってはルールのフロー制御機能を持っており、そのような製品を用いる場合、ルールの可視性を低下せず、各ルールの再利用性が妨げられない範囲においてのみ、フロー制御機能の使用を許容する。

詳細は、「3.2.7.1.1 ビジネスルールをアプリケーションから切り離すための仕組みとして利用する技術」を参照のこと。

- チェック処理における、入力データ単体で完結するチェック処理(データの性質に基づくチェック等)、あるいは入力データ間の相関関係に基づく単純なチェック処理の定義及び実行には、BRMSを適用せず、特許庁フレームワークの機能を利用して実現する。

- ビジネスルールの判定に用いる情報をデータベース等にアクセスして取得する処理は、業務アプリケーションを適用する。業務アプリケーションからビジネスルールの判定に用いる情報を受け取り、ビジネスルールの判定を行う処理の定義及び実行にBRMSを適用する。

3.2.9.3 ESBのルール

スコープ： インタフェース定型化サブシステム(外部システム連携)

プロセス： 2.4.3

指針1： ESBは、「(1)ESBの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。
なお、「A.TRMの定義から必要とされる機能要件(基本事項)」に示す機能要件は、外部システム互換機能と組み合わせて実現してもよいものとする。

指針2： ESBは、「(3)ESBの適用範囲」に示す範囲で適用すること。

ESBを補完するシステム構成要素として、外部システム互換機能がある。ESBの機能要件は、外部システム互換機能と組み合わせて実現する。

(1) ESBの機能要件

A. TRMの定義から必要とされる機能要件(基本事項)

- 特定のネットワークトランスポート技術に依存しない構造で異なるシステム間をつなぐ共通バックボーンを提供すること。
- 異なるプロトコル間の変換が可能なこと。
- 異なるデータ形式の間で変換が可能なこと。
- メッセージ内容によるルーティングが可能なこと。
- エンタープライズ・サービス・バスを利用する他の要素の処理を変えることなく、独立して使用できること。

B. TRMの定義から必要とされる機能要件(加点事項)

加点事項無し。

C. 特許庁固有の機能要件

- 「表 3.2-2 プロトコル及び電文形式(階層定型化サブシステム)」及び「表 3.2-3 プロトコル及び電文形式(インタフェース定型化サブシステム)」に記載されているプロトコル、電文形式に対応していること。

理由： ToBeアーキテクチャで利用するプロトコル及び電文形式を標準化するため。

- 特殊なプロトコル変換を行う等のため、ESBの一連の処理(変換やルーティング等)の中で外部システム互換機能と連携することが可能なこと。

理由： 外部システム互換機能にてESBが対応しない連携ギャップの吸収処理を補完するため。

(2) ESBの機能要件の補足

TRMの基本事項に定義されている要件のうち、以下の事項は必須としない。

- 分散アプリケーション間において、アプリケーションやネットワークで障害が発生した場合、トランザクションをロールバックしてエラー通知を行ったり、エラー解決後トランザクションを再送信するための、メッセージを確実に伝達したりする仕組みを提供すること。

理由： 送達保証機能は、ある多重度を境に急激にスループットの劣化が発生するため、使用しない。

(3) ESBの適用範囲

ESBと外部システム互換機能の適用範囲の分担は、採用するESBの製品に大きく左右される。そのため、ESBの適用範囲としてはESBの適用の最小範囲と、ESBと外部システム互換機能の組み合わせの適用範囲を規定するに留め、それ以外は設計に関与するステークホルダ(設計・開発ベンダ等)の裁量に委ねることとする。

外部システム連携層におけるESBの適用の最小範囲を以下に示す。

- 内部システム向けにインタフェースを提供する用途で適用する。
- 外部システム互換機能に対し、サービスの物理位置を隠蔽し、リクエストに応じた実行すべきサービスの特定、実行を含むようなシステムのルーティングを行う場合に適用する。

ESBと外部システム互換機能の組み合わせを適用する範囲を以下に示す。

- 内部システムと外部システムの連携する際、連携方式のギャップ(プロトコル、文字コード、電文形式、連携データ、連携タイミング及び通信データ量等)を吸収する仲介役として適用する。内部システムと外部システムの連携がない場合は、適用しない。

なお、ESBと外部システム互換機能を組み合わせる際には、アプリケーションの作り込みを極力避け、ESB製品の設定情報の変更機能等のプログラムプロダクトの機能を利用すること。

3.2.9.4 Webブラウザのルール

スコープ： 階層定型化サブシステム

プロセス： 2.4.3

指針1： Webブラウザは、「(1)Webブラウザの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。

指針2： Webブラウザは、「(3)Webブラウザの適用範囲」に示す範囲で適用すること。

(1) Webブラウザの機能要件

A. TRMの定義から必要とされる機能要件(基本事項)

- Webサーバと通信して指定されたURIから情報を取り出すことができること。
- W3C HTML標準, ECMAスクリプト規格に規定された機能のみを用いて作成されたウェブコンテンツの表示及びフォームを通じての入力ができること。
- JPEG, GIF及びPNG形式の画像データを表示できること。
- 画像の自動ダウンロードの制御ができること。
- ポップアップの制御ができること。
- フィッシングサイトURLへのアクセス制御ができること。
- エンコードの選択ができること。

B. TRMの定義から必要とされる機能要件(加点事項)

加点事項無し。

C. 特許庁固有の機能要件

- システム利用者が使用する業務用PCのOS上で動作すること。
理由： システム利用者が使用する環境で動作する必要があるため。

(2) Webブラウザの機能要件の補足

無し。

(3) Webブラウザの適用範囲

- 多階層構造におけるUI層のクライアント構成として、「ブラウザを用いた構成」の選定条件に適合した場合は、Webブラウザを適用する。リッチクライアントの選定条件に適合した場合は、Webブラウザは適用しない。クライアント構成の選定ルールは、「3.2.2.1 クライアント構成」を参照。
- 「ブラウザを用いた構成」を採用した際に、プレゼンテーションロジックから受信した「(1)Webブラウザの機能要件」に示す静的コンテンツを表示する処理にWebブラウザを適用する。
- 「ブラウザを用いた構成」を採用した際に、「表 3.1-10 ブラウザ又はリッチクライアントの責務」を実現するためにWebブラウザを適用する。

3.2.9.5 Webサーバのルール

スコープ： 階層定型化サブシステム

プロセス： 2.4.3

指針1： Webサーバは、「(1)Webサーバの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。

指針2： Webサーバは、「(3)Webサーバの適用範囲」に示す範囲で適用すること。

(1) Webサーバの機能要件

A. TRMの定義から必要とされる機能要件(基本事項)

- Webブラウザ等のクライアントPCからの要求に対して、Webサーバ上に格納されたコンテンツをHTTP又はそれに準ずるプロトコルにより返送できること。加えてプログラムを実行することによって動的にHTMLデータを作成して返送する機能を有すること。
- ログイン状態や画面遷移を管理することのできるセッション管理機能を有すること。
- IP認証やユーザ認証等により、Webサイトに対する認証と認可(承認)が行えること。またサーバ外部のディレクトリサービスと連携した認証・認可を行えること。
- SSLやTLS等のプロトコルを使用してWWWの通信データを暗号化できること(データ盗聴・改ざん・なりすましを防御することができること)。

条件： サブシステムの要件に通信データの暗号化が含まれる場合のみ、機能要件に加えるものとする。

B. TRMの定義から必要とされる機能要件(加点事項)

採用無し。

C. 特許庁固有の機能要件

無し。

(2) Webサーバの機能要件の補足

TRMの基本事項に定義されている要件のうち、以下の事項は必須としない。

- ネットワークを介してサーバに送られる処理要求を一括して受け、管理し、複数のサーバ群に分散して中継・送信する負荷分散処理機能を有すること
理由： 他の手段(負荷分散装置等)と組み合わせて、必要に応じて実現する機能要件であるため、必須としない。
- 負荷分散装置の分散方式は、ラウンドロビンと、応答時間を考慮した負荷分散とが選択できること。また、どちらの方式の場合でも、事前に設定したレスポンス時間以上応答がないサーバに対して負荷分散処理をやめる機能を有すること
理由： 他の手段(負荷分散装置等)と組み合わせて、必要に応じて実現する機能要件であるため、必須としない。

(3) Webサーバの適用範囲

A. WebサーバとUI層のシステム構成要素との境界

- UI層のシステム構成要素とHTTPに則って電文の送受信を行う用途で適用する。

B. WebサーバとAPサーバとの境界

以下の機能については、APサーバと連携して実現するようWebサーバを適用する。

- 動的にHTMLデータを作成して返送する機能については、APサーバと連携して業務アプリケーションを実行し、その結果を受け取って動的コンテンツを提供するような構成で適用する。
- セッション管理機能について、APサーバと組み合わせて実現されるような構成で適用する。
- 認証と認可の機能要件について、APサーバと組み合わせて実現されるような構成で適用する。

以下の機能については、APサーバと連携せず、Webサーバのみを適用する。

- 静的コンテンツを提供する場合には、Webサーバのみで処理し、APサーバとは連携しない。

3.2.9.6 APサーバのルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム(ビジネスルール管理)

プロセス： 2.4.3

指針1： APサーバは、「(1)APサーバの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。

指針2： APサーバは、「(3)APサーバの適用範囲」に示す範囲で適用すること。

(1) APサーバの機能要件

A. TRMの定義から必要とされる機能要件(基本事項)

- SOAPやWSDL等のWebサービス技術を利用したサービスを提供できること。
- Javaコンポーネント, .NETコンポーネント等のコンポーネントアプリケーションを構築・実行するための基盤を有すること。
- APサーバ上で実行されるアプリケーションプログラムがDBサーバ上のデータベースに接続してデータのアクセスを行うためのインタフェースと機能を有すること。
- 遠隔地のサーバ間でプログラム同士がメッセージ通信を行うことのできる機能を有すること。
- アプリケーションプログラム群, サービス群をAPサーバ上に配信・配付して利用／提供可能な状態にできること。
- APサーバへの処理要求に対する認証と認可(承認)が行えること。その際にAPサーバ外部のディレクトリサービスと連携した認証・認可が行えること。

B. TRMの定義から必要とされる機能要件(加点事項)

加点事項無し。

C. 特許庁固有の機能要件

無し。

(2) APサーバの機能要件の補足

TRMの基本事項に定義されている要件のうち、以下の事項は必須としない。

- 複数のデータベース／サーバにまたがるデータの更新を単一のトランザクションとして処理できること。

理由： 検討結果から、データベースの更新にはグローバルトランザクションを用いて複数のデータベースを更新(2フェーズコミット)するような方法を採用しないため、使用しない。

- 以下の脅威からAPサーバ及びアプリケーションを防御するためのセキュリティ機能を有すること。

➤ ネットワーク盗聴(WebサーバとAPサーバ間, APサーバとDBサーバ間)

➤ 不正アクセス

防御のための機能は、APサーバ単独の機能として実現していなくとも、ネットワーク機器等の機能を用いてシステム全体の構成として実現できていればよい。ネットワーク盗聴防止は通信の暗号化を必ず求めるものではなく、AP通信パケットがWebサーバとAPサーバ間, APサーバとDBサーバ間以外に流れなければよい。不正アクセス対策はIPアドレス制限等の設定により、APアクセスをWebサーバからのアクセスだけに制限できればよい。

理由： 当該機能は、APサーバだけでなく、他の手段(上記に記載されているような内容のもの)と組み合わせる必要に応じて実現する機能要件であるため、必須としない。

- DBサーバの構成要素の性能(トランザクション処理スループット, レスポンスタイム等)を監視し、サービスが適切に提供されていることを監視する機能を有すること。性能の劣化の兆候を検出した場合には、自動又は手動で性能を最適化することができること。

理由： APサーバだけでなく、他の手段(システム運用管理プロダクト等)と組み合わせる必要に応じて実現する機能要件であるため、必須としない。

(3) APサーバの適用範囲

A. APサーバと業務アプリケーションとの境界

- HTTPやSOAPといった通信プロトコルによる通信を伴って実行する業務処理のうちプログラムプロダクトとして汎用的に実現すべき機能を実行し、業務アプリケーションと連携し処理を実行するための実行環境として適用する。

APサーバと業務アプリケーションの適用範囲の分担は、採用するAPサーバ製品が持つ機能により異なり、将来的にも変遷があると考えられるため、上記以外のルールは規定せず、設計に関与するステークホルダ(設計・開発ベンダ等)の裁量に委ねる。

B. APサーバとWebサーバとの境界

- 「3.2.9.5 (3)B.WebサーバとAPサーバとの境界」を参照のこと。

3.2.9.7 FTPサーバのルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム(外部システム連携)

プロセス： 2.4.3

指針1： FTPサーバは、「(1)FTPサーバの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。

指針2： FTPサーバは、「(3)FTPサーバの適用範囲」に示す範囲で適用すること。

(1) FTPサーバの機能要件

A. TRMの定義から必要とされる機能要件(基本事項)

TRMに該当する技術ドメインがないため、基本事項無し⁴⁸。

B. TRMの定義から必要とされる機能要件(加点事項)

TRMに該当する技術ドメインがないため、加点事項無し⁴⁸。

C. 特許庁固有の機能要件

- FTPサーバにFTP(S)プロトコルを用いて接続でき、ファイルの送受信ができること。

理由： FTPサーバの基本的な機能であるため。

- システム利用者及び情報システムの運用管理者を特定し、それぞれの権限に合わせて許可される操作や、その対象となるファイルやフォルダへの操作の許可や拒否を行う仕組みを有すること。

理由： 『ポリシー実施手順(開発編別紙「設計基準」)』に従ったセキュリティ方式を採用するため。

(2) FTPサーバの機能要件の補足

無し。

(3) FTPサーバの適用範囲

- 以下に示す連携に限り、大容量ファイルを効率よく転送したい場合や、一括処理のため複数件数のレコードデータを1つのファイルにまとめてやり取りしたい場合に適用する。
 - 業務アプリケーション(バッチ)と他サブシステムの業務アプリケーション(バッチ)との間の連携
 - ESB又は外部システム互換機能と業務アプリケーション(バッチ)との間の連携

⁴⁸ 『情報システム調達のための技術参照モデル(TRM) 平成24 年度版 物品調達編』の「2.11.3.FTPサービス」は、機能要件として、公開サービス向けの機能要件の定義がされており、ToBeアーキテクチャで想定しているシステム内部でFTPプロトコルを用いてファイル送受信を行う用途と異なるものであるため、FTPサーバの技術ドメインとしていない。

3.2.9.8 ジョブ管理エージェントのルール

スコープ： 階層定型化サブシステム

プロセス： 2.4.3

指針1： ジョブ管理エージェントは、「(1)ジョブ管理エージェントの機能要件」に示す機能要件を満たすジョブ管理マネージャのエージェントとして動作すること。

指針2： ジョブ管理エージェントは、「(3)ジョブ管理エージェントの適用範囲」に示す範囲で適用すること。

(1) ジョブ管理エージェントの機能要件

以下の機能要件を満たすジョブ管理マネージャのエージェントとして動作することをジョブ管理エージェントの機能要件とする。

A. TRMの定義から必要とされる機能要件(基本事項)

- システム運用カレンダーの定義が可能であること。
- システム運用カレンダーに従ってプログラム・ジョブ実行がスケジューリングでき、自動実行が可能であること。
- ジョブ運行制御の一元管理が可能であること。
- ジョブの実行状況、実行履歴はログ情報として記録可能であること。
- ジョブの実行状況がモニタリング可能であること。
- 1日を48時間で管理できること。

※運用日1日を24:00を超えた表記で扱えること。

例： 1日の運用開始時間が「9:00」の場合、9:00～33:00(翌日9:00)として扱えること。

B. TRMの定義から必要とされる機能要件(加点事項)

- イベント(タイマやファイル受信、ファイル変更等)を契機にジョブの実行が可能であること。

C. 特許庁固有の機能要件

無し。

(2) ジョブ管理エージェントの機能要件の補足

無し。

(3) ジョブ管理エージェントの適用範囲

- ジョブ管理と連動して業務アプリケーション(バッチ)に実行契機を与える役割として適用する。

3.2.9.9 帳票設計・印刷ソフトウェアのルール

スコープ： 階層定型化サブシステム

プロセス： 2.4.3

指針1： 帳票設計・印刷ソフトウェアは、「(1)帳票設計・印刷ソフトウェアの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。

指針2： 帳票設計・印刷ソフトウェアは、「(3)帳票設計・印刷ソフトウェアの適用範囲」に示す範囲で適用すること。

(1) 帳票設計・印刷ソフトウェアの機能要件

A. TRMの定義から必要とされる機能要件(基本事項)

TRMに該当する技術ドメインがないため、基本事項無し。

B. TRMの定義から必要とされる機能要件(加点事項)

TRMに該当する技術ドメインがないため、加点事項無し。

C. 特許庁固有の機能要件

- 帳票に埋め込まれるデータとレイアウトが分離されており、レイアウトの変更が容易であること。
理由： ToBeアーキテクチャでは、保守性の観点から帳票データとレイアウトファイルを分離して管理する方式をとるため。「3.1.2.4 帳票出力方式」参照のこと。
- 新規出力帳票の作成、及び作成したレイアウトに対する雛型文書の編集等がGUIを用いて可能であること。
理由： 開発効率性や保守性の観点からレイアウトの作成にGUIを用いるため。
- 作成した帳票をPDFファイルとして出力するための機能を有すること。
理由： 「3.2.2.5 帳票設計ルール」の「3.2.2.5.4 帳票の管理に関するルール」を参照。

(2) 帳票設計・印刷ソフトウェアの機能要件の補足

無し。

(3) 帳票設計・印刷ソフトウェアの適用範囲

- 各業務アプリケーションからの実行依頼を受けて帳票を生成・出力するための用途として適用する。

3.2.9.10 DBMSのルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム(外部システム連携)

プロセス： 2.4.3

指針1： DBMSは、「(1)DBMSの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。

指針2： DBMSは、「(3)DBMSの適用範囲」に示す範囲で適用すること。

(1) DBMSの機能要件

A. TRMの定義から必要とされる機能要件(基本事項)

- リレーショナルなデータ表現形式のデータベースを管理するシステム(リレーショナルデータベース管理システム)、又は、XMLデータを取り扱うことのできるXMLデータベースシステムであること。

除外：「又は、XMLデータを取り扱うことのできるXMLデータベースシステムであること。」については、特許庁システムの要件にないため、機能要件から除外する。リレーショナルデータベース管理システムであることを必須の機能要件とする。

- 問い合わせ言語(データ定義言語:DDL, データ操作言語:DML, データ制御言語:DCL)を使って記述したプログラムを実行することにより、データベースの定義、データのアクセス、アクセス制御を行えること。
- トランザクション処理機能を有すること。
- データベースにアクセスする利用者を、識別子を通して特定し、許可されている利用者ならばアクセスを許可し、それ以外は拒否する機能を有すること。ユーザの役割毎に、データベース上のオブジェクト(テーブル、ビュー、列)毎に行うことのできるアクションを限定できること。

B. TRMの定義から必要とされる機能要件(加点事項)

加点事項無し。

C. 特許庁固有の機能要件

- ETL⁴⁹製品に対応し、連携することが可能なこと。

理由：情報活用系サブシステムへ情報を提供する目的で、ETLを経由してDBMSに保持する情報を抽出するため。

- ANSI/ISOのSQL標準のSQLを使用してアクセスが可能なこと。

理由：「3.2.11 開発言語の選定ルール」を参照。

(2) DBMSの機能要件の補足

TRMの基本事項に定義されている要件から以下の事項は必須としない。

- データベースの監査を行える機能を有すること。利用者が実行したSQL文の実行履歴をログの形式で記録できること。また権限を越えたアクセスの事象についてはその情報を管理ツール等に通知できること。また、ログの情報をサマリーとして取りまとめた報告書の形式(レポート形式)で閲覧できること。

理由：DBMSだけでなく、他の手段(システム運用管理プロダクトやレポート製品等)と組み合わせる必要に応じて実現する機能要件であるため、必須としない。

- DBサーバの構成要素の性能(トランザクション処理スループット、レスポンスタイム等)を監視し、サービスが適切に提供されていることを監視する機能を有すること。性能の劣化の兆候を検出した場合には、自動又は手動で性能を最適化することができること。

理由：DBMSだけでなく、他の手段(システム運用管理プロダクト等)と組み合わせる必要に応じて実現する機能要件であるため、必須としない。

⁴⁹ TRMに記載されている「ETL」の定義を参照のこと。

(3) DBMSの適用範囲

- 業務アプリケーションやツール等で生成したデータを永続化する際に、永続化したデータをSQLで高速に検索することが求められる場合又はデータの更新の際にACID特性を求められる場合にDBMSを使用する。ただし、大量データのため永続化等にDBMSを利用すると効率が極めて悪い場合はDBMSを使用しない。DBMSを使用しない場合は、ファイル等で永続化する。

3.2.9.11 認証プラグイン／個人認証ソフトウェアのルール

スコープ： 階層定型化サブシステム

プロセス： 2.4.3

指針1： 認証プラグイン／個人認証ソフトウェアは、「(1)認証プラグイン／個人認証ソフトウェアの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。

指針2： 認証プラグイン／個人認証ソフトウェアは、「(3)認証プラグイン／個人認証ソフトウェアの適用範囲」に示す範囲で適用すること。

(1) 認証プラグイン／個人認証ソフトウェアの機能要件

A. TRMの定義から必要とされる機能要件⁵⁰(基本事項)

- 端末上で稼動するエージェントが、利用者に代わってユーザ認証を行うことにより、Web アプリケーションやグループウェア、C/S型アプリケーション等に自動的にログインし、シングルサインオンを実現すること。
- 利用者はデスクトップに一度ログインすれば、端末上の利用可能なアプリケーションをログインの手間をかけずに利用することができること。デスクトップOSのログイン認証と連携するアプリケーションの採用、又は、予めユーザIDとパスワードを登録して利用するアプリケーションに合わせてエージェントが認証情報を自動的に入力することにより、シングルサインオンを実現する。

B. TRMの定義から必要とされる機能要件⁵⁰(加点事項)

採用無し。

C. 特許庁固有の機能要件

無し。

(2) 認証プラグイン／個人認証ソフトウェアの機能要件の補足

TRMの基本事項に定義されている要件のうち、以下の事項は必須としない。

- シングルサインオンの対象とするアプリケーションや自動入力されるユーザIDとパスワードは、システム管理者が一元的に管理できること。端末の設定を集中管理できること。
理由： 認証プラグイン／個人認証ソフトウェアだけでなく、他の手段と組み合わせて必要に応じて実現する機能要件であるため、必須としない。

(3) 認証プラグイン／個人認証ソフトウェアの適用範囲

- 『LDAPアクセス運用』で規定するシングルサインオンの仕組みに適用する。

⁵⁰ 認証プラグイン／個人認証ソフトウェアのTRMの技術ドメインは、『情報システム調達のための技術参照モデル(TRM) 平成24年度版 物品調達編』の「2.13.5.デスクトップ・シングルサインオンの機能要件・非機能要件・標準技術」である。

3.2.10 文字コードの扱いのルール

(1) 文字コードの扱いのルールの方針

文字コードの扱いのルールにおける方針を以下に示す。

なお、文字コードは文字の集合である「文字セット」(符号化文字集合)とバイトコードのマッピングを示した「エンコーディング」(符号化方式)の2つを指す。

- ToBe対象システムでは、システム固有に様々な文字コードを利用している。保守性を考慮し、内部システムで扱う文字コードを統一する。
- ToBe対象システムが扱っている文字は、内部システムでも利用できるようにする。
- 外部システムに極力、改修が発生しないように考慮する。
- ToBeアーキテクチャが長期にわたって適用されることを想定し、将来的に必要となると考えられる文字セットも扱えるようにする。

(2) ToBe対象システムが使用する文字

ToBe対象システムが使用する文字は以下に示す規定内文字及び特許庁外字に分類される。

A. 規定内文字

特許庁システムの文書仕様⁵¹における文字種の規定に基づき、以下を規定内文字とする。

- JIS X 0201-1976 (1byte)
- JIS X 0208-1997 (2byte)

ただし、半角カナ、丸文字、外字を除く。

B. 特許庁外字

規定内文字以外で、特定の文字を独自に規定し、特許庁システムにおいて運用を行っている文字種を「特許庁外字」とする。

特許庁外字の分類を下表に示す。

表 3.2-72 特許庁外字の分類

項番	外字分類	説明	発生起因
1	システム独自要件に基づく文字	特許庁システムにおいて、サブシステム独自に利用要件があり、個別に登録又は特別な処理を行っている文字種	特許庁内要望
2	マドプロ用のラテン文字	IBから送付されるマドプロ出願に含まれるラテン文字	特許庁外要望
3	人名・地名用の外字	職員等の姓名、及び地名表示のために個別登録している外字	事務的要望

上記を踏まえ、ToBe対象システムが使用する文字の構成を下図に示す。

規定内文字	特許庁外字
JIS X 0201-1976 JIS X 0208-1997 (半角カナ、丸文字、外字を除く)	(1)システム利用外字 (2)マドプロ用ラテン文字 (3)人名・地名用外字

図 3.2-53 ToBe対象システムが使用する文字の構成

⁵¹ 以下のXML及びSGMLの文書仕様を指す。(2015年1月時点)

XMLの文書仕様:『日本国特許庁電子文書交換標準仕様 XML編 第5.41版』の「1.1.5 使用文字種」

SGMLの文書仕様:『日本国特許庁電子文書交換標準仕様 特定書類編 第5.3版』の「6.1 文字セット」

3.2.10.1 使用する文字コードのルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.2, 2.4.3

規約1： 内部システムで扱う文字コードは、「(1)使用する文字コード」に示す文字コードを採用すること。

規約2： 内部システムで使用する文字は、「(2)使用可能な文字」に示す文字とすること。

規約3： 内部システムで扱う文字コードの仕様への対応は、「(3)文字コードの仕様への対応」のとおりとすること。

規約4： 文字コードは、「(5)文字コードを扱う対象」に示す対象に適用すること。

特例1： プログラムプロダクトのログや設定ファイル等、業務処理に直接影響しないものは、「(1)使用する文字コード」に示した文字コードをサポートしないことを許容する。

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.2

規約1： 特許庁外字の扱いは、「(4)特許庁外字の扱い」のとおりとすること。

(1) 使用する文字コード

内部システムで使用する文字コードのルールを以下に示す。

- 文字セット／エンコードは、Unicode/UTF-8とする。

(2) 使用可能な文字

Unicodeで規定する文字の中から、内部システムが使用可能な文字を制限する。

特許庁システムの文書仕様で規定される使用可能な文字がJIS第三、第四水準を含むJIS X 0213に拡張されることを想定し、内部システムで使用可能な文字(以降、ToBe使用可能文字と呼ぶ)を以下とする。

- JIS X 0213で規定する文字(以降、ToBe規定内文字と呼ぶ。)
- JIS X 0213で規定外の特許庁外字(以降、ToBe特許庁外字と呼ぶ。)

ToBe対象システムと内部システムにおける規定内文字と特許庁外字の関係を下図に示す。

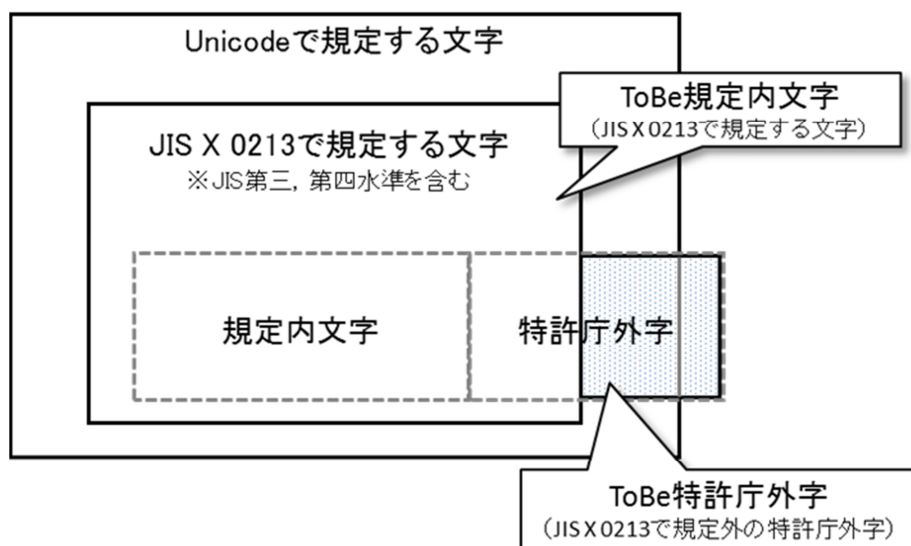


図 3.2-54 ToBe対象システムと内部システムにおける規定内文字と特許庁外字の関係

(3) 文字コードの仕様への対応

JIS第三, 第四水準, Unicodeが持つ仕様への対応を以下に示す。

- 「サロゲートペア」(Unicode3.2以上)は使用する。
- 「結合文字」は使用しない。
- 「異体字セクタ」⁵²(Unicode5.1以上)は使用しない。

(4) 特許庁外字の扱い

特許庁外字の扱いのルールを以下に示す。

- 各ToBe対象システムで使用している特許庁外字をUnicodeへマッピングし, ToBe特許庁外字を一元化する。

マッピングの結果, 特許庁外字は以下のように扱う。

- ① Unicode及びJIS X 0213に含まれる特許庁外字は, ToBe規定内文字として扱う。
- ② Unicodeに含まれるが, JIS X 0213に含まれない特許庁外字は, ToBe特許庁外字として扱う。JIS X 0213以外に追加で使用する文字を明確にすること。
- ③ Unicodeに含まれない特許庁外字は, Unicodeの私用領域にマッピングし, ToBe特許庁外字として扱う。JIS X 0213以外に追加で使用する文字と対応する符号位置を明確にすること。

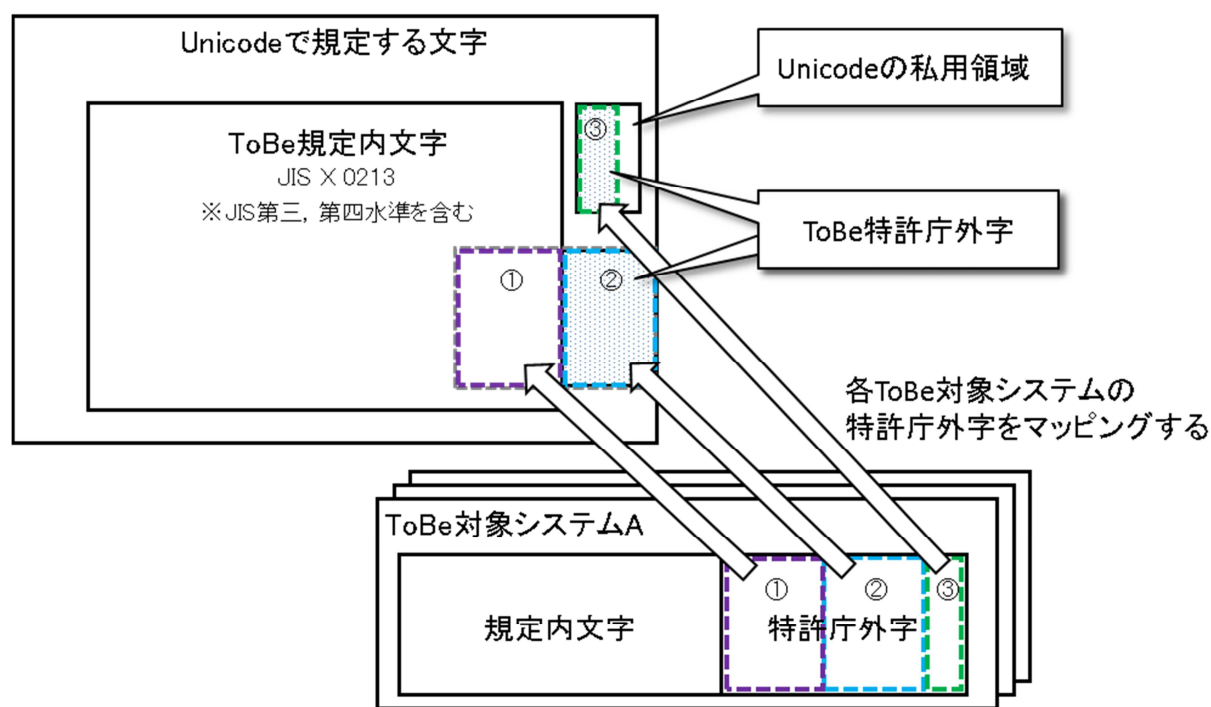


図 3.2-55 特許庁外字のマッピング

⁵² Unicodeにおいて異体字のグリフ(字形)を指定するためのセクタ(選択子)。

異体字は, 意味が同じで字形(グリフ)が異なる文字であり, Unicodeの符号位置は親文字と同じである。

文字符号の後に異体字セクタを置くことで, どの異体字を表すか指定する。

漢字の異体字のコレクションは複数存在し, それぞれで定義する文字が異なり重複も存在するため, 異体字セクタを使用する場合は適用するコレクションを決定しておく必要がある。また, 対応するフォントもそれぞれ異なる。

- プログラムプロダクトは、ToBe規定内文字、ToBe特許庁外字を扱えること⁵³。
- ToBe特許庁外字の管理を行うこと。
 - サブシステム間連携の全体調整・管理を担うベンダが担当すること。
 - 管理対象は文字のマッピング及び私用領域に登録するグリフ(字形)とする。

(5) 文字コードを扱う対象

文字コードを扱う対象を下表に示す。

表 3.2-73 文字コードを扱う対象

項番	対象	ルールの適用範囲
1	プログラムプロダクト	プログラムプロダクト内の文字関連処理 ※プログラムプロダクトはUnicode/UTF8をサポートしていること。 ※ただし、ログや設定ファイル等、業務処理に直接影響しないものはサポートしていなくてもよい。
2	アプリケーション	プログラムや設定ファイル、ログファイル等。
3	データベース値	データベースのカラム内に格納されているデータ等。
4	ファイル	連携ファイル(CSV等)等。
5	電文	SOAPのXMLや電文データ等。
6	Webページ	テキストリソース(HTML) 等。

⁵³ ToBe特許庁外字のうちUnicodeに含まれない特許庁外字においては、Unicodeの私用領域の内容を勝手に変換しない(プログラムプロダクトによってこれらの文字列の入力が妨げられたり、意図せず違う文字列に変換されたりしない)、画面や帳票に出力するプログラムプロダクトでは外字フォントを指定し出力する方法があることを指す。

3.2.10.2 文字コードに関する制御のルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.3

規約1： 内部システムにおける文字コードに関する制御は、以下に示す(1)～(4)のとおり扱うこと。

規約2： 特許庁システムの文書仕様における使用可能な文字の変更前後の対応は、(5)のとおりとすること。

指針1： JIS X 0213:2004のグリフの扱いは、「(6)JIS X 0213:2004のグリフの扱いについて」に従い設計すること。

(1) 外部システム連携における文字コードの扱い

外部システム連携における文字コードの扱いにおけるルールを以下に示す。

- 外部インタフェースで文字コード変換や外字変換が必要な場合、外部システム連携層で文字コード変換を行うこと。
- 外部システムから内部システムへ送信された文字種のチェックを行い、ToBe使用可能文字以外の場合は、エラーを返却すること。

(2) 画面における文字コードの扱い

画面における文字コードの扱いにおけるルールを以下に示す。

A. サーバ側の対処

- 画面入力又はファイルアップロードで内部システムへ送信された文字種のチェックを行い、ToBe使用可能文字以外の場合は、エラーを返却すること。

B. 業務用PCの対処

- ToBe使用可能文字に対応したフォントを使用すること。
- Unicodeの私用領域に格納されている外字のグリフ(字形)を業務用PCに予め登録しておくこと。

(3) 帳票及び出力装置における文字コードの扱い

帳票及び出力装置における文字コードの扱いにおけるルールを以下に示す。

- プリンタ出力時、プリンタフォントを使用する場合、ToBe使用可能文字をサポートするフォントを用意すること。
- プリンタフォントを使用しない場合、画像にビットマップ化して出力すること。
- 画像にビットマップ化して出力する場合、Unicodeの私用領域に格納されている外字のグリフ(字形)を帳票出力するサーバに予め登録しておくこと。

(4) 業務アプリケーションにおける文字コードの扱い

業務アプリケーションにおける文字コードの扱いにおけるルールを以下に示す。

- サロゲートペアが含まれても正しく機能する文字列処理をアプリケーションに実装すること。
- 結合文字、異体字セレクトは使用しないものとし、文字種チェックにおいて検出し、エラーを返却すること。

(5) 文書仕様における使用可能な文字の変更前後の対応

特許庁システムの文書仕様で規定される使用可能な文字がJIS第三，第四水準を含むJIS X 0213に拡張されるまでは，規定内文字，特許庁外字のみを扱う必要がある。使用可能な文字の規定が変更される前後の対応のルールを以下に示す。

- 使用可能な文字の規定が変更される前後ともに，下表に示すシステム構成要素で内部システムに送信されるデータの文字種チェックを行うこと。

表 3.2-74 文字種チェックの対象データとチェック箇所

項番	データ	文字種チェックを行うシステム構成要素
1	画面からの入力データ	プレゼンテーションロジック
2	画面からのアップロードされたファイル	業務処理と同じタイミングで文字種チェックを行う必要がある場合（例えば，ファイルのストリームから順次データを取得し，業務処理を行う場合）は業務アプリケーション（ユーザ）。 それ以外は，プレゼンテーションロジックとする。
3	外部システムからの電文又は連携ファイル	ESB又は外部システム互換機能

- 使用可能な文字の規定が変更される前後で，上記の文字種チェックの内容を切り替えること。文字種チェックの内容を下表に示す。

表 3.2-75 文字種チェックの内容

使用可能な文字の規定の変更前／後	文字種チェックの内容
変更前	規定内文字及び特許庁外字に該当する文字以外の場合，エラーを応答する。
変更後	ToBe規定内文字及びToBe特許庁外字に該当する文字以外の場合，エラーを応答する。

(6) JIS X 0213:2004のグリフの扱いについて

JIS X 0213:2004（以降，「JIS2004」と呼ぶ）では文字セットに加え，デフォルトグリフ（例示字体）が変更になっている。そのため，JIS2004をサポートするフォントを使用すると従来のフォントとは異なる字体の文字が表示・印刷されることになる。

特に商標においては影響が大きいため，JIS2004を適用する際には扱いを整理しておく必要がある。以下のような対応方法が考えられる。

- 包摂基準に従い新旧の文字を同じものとして捉える。
 メリット： システム的な対応が必要ない。
 デメリット： 外部仕様の変更を要する。法令や運用規則等の改正又は解釈の修正が求められる可能性がある。
- DB等のデータを変換（Unicodeの私用領域に外字として割り当てを）する。
 メリット： 外部仕様の変更が必要ない。
 デメリット： 比較的規模の大きなデータ変換作業が必要となる。

上記を踏まえ，サブシステム毎に対応方法を検討すること。

3.2.11 開発言語の選定ルール

スコープ:	階層定型化サブシステム及びインタフェース定型化サブシステム
プロセス:	2.4.3
指針1:	「表 3.2-76 システム構成要素毎の開発言語と選定指針」で推奨とするシステム構成要素は、同表の「選定指針」に従い開発言語を採用すること。
推奨1:	「表 3.2-76 システム構成要素毎の開発言語と選定指針」で示す開発言語を推奨する。
規約1:	「表 3.2-76 システム構成要素毎の開発言語と選定指針」で規約とするシステム構成要素は、同表に示す開発言語を使用すること。

(1) 開発言語選定における共通指針

各システム構成要素の開発言語を選定する上での共通指針を以下に示す。

開発言語は、以下の特性を持つ言語を選定すること。

- 開発・保守の容易性
 - 市場のシェアを多く占めており、技術者の確保が容易であること。
 - 開発環境が提供される等、開發生産性が高い開発言語であること。
- 移植の容易性
 - プログラムプロダクトの変更、バージョンアップ等の環境変化が発生しても、対応が容易であること。
- 動作環境に対する対応性
 - 開発したプログラムが動作するプログラムプロダクトがサポートしていること。
- 利用部品に対する対応性
 - 基盤API等のサブシステム間共通機能呼び出すことができること。
 - コンポーネントが利用するライブラリ呼び出すことができること。

ライブラリの例：BPMSのワークフローAPI、BRMSライブラリ、OCRソフトウェア等

(2) 開発言語の選定

前述の「(1)開発言語選定における共通指針」に従い、開発言語を選定する。下表にシステム構成要素毎の開発言語を示す。

表 3.2-76 システム構成要素毎の開発言語と選定指針

項番	システム構成要素	規約／推奨	開発言語	選定指針
1	ブラウザ	規約	● HTML ● JavaScript ※文書構造はW3C勧告に従うものとし、特定のブラウザ製品でのみ動作する機能は利用しない ⁵⁴ 。	—
2	リッチクライアント	推奨	● .NET Frameworkがサポートする開発言語。	● 「(1)開発言語選定における共通指針」に準ずる。
3	プレゼンテーションロジック	推奨	● Java	● 業務アプリケーション(ユーザ)と同一とする。
4	業務アプリケーション(ユーザ)	推奨	● Java	● 「(1)開発言語選定における共通指針」に準ずる。
5	業務アプリケーション(システム)	推奨	● Java	● 業務アプリケーション(ユーザ)と同一とする。

⁵⁴ HTMLは、『HTML文書規約』に従うこと。ただし、HTML5以上を導入する場合、HTML5以上に適合した『HTML文書規約』の策定が必要となる。

項 番	システム構成要素	規約/ 推奨	開発言語	選定指針
6	業務アプリケーション (バッチ)	推奨	● Java	● 業務アプリケーション(ユーザ)と 同一とする。
		推奨	ジョブ管理エージェントからシェ ルスクリプトを経由してバッチを 起動する場合 ● シェルスクリプト	● 「(1)開発言語選定における共通 指針」に準ずる。
7	BPMS	—	—	—
8	BPMS補完機能	推奨	● Java	● BPMSがサポートしている開発言 語とする。 ● 極力、業務アプリケーション(ユ ーザ)と同一とする。
9	BRMS	—	—	—
10	ESB	—	—	—
11	外部システム互換機 能	推奨	● Java	● ESBがサポートしている開発言語 ● 極力、業務アプリケーション(ユ ーザ)と同一とする。
12	個別データベース ※個別データベース へアクセスする開発 言語を示す。	規約	● ANSI/ISOのSQL標準のS QL ● 製品固有の拡張SQL(性 能要件を満たす等の目的 での使用に制限する。)	—
13	DBアクセス基盤サー ビス	推奨	● Java	● 「(1)開発言語選定における共通 指針」に準ずる。
14	共有データベース ※個別データベース へアクセスする開発 言語を示す。	規約	● ANSI/ISOのSQL標準のS QL ● 製品固有の拡張SQL(性 能要件を満たす等の目的 での使用に制限する。)	—

3.2.12 特許庁システムが標準とする非機能要求グレードのルール

(1) システムの分類

「権利者及び出願人等から産業財産権の情報に関する申請を受け付けて、審査官が審査し、当該情報を長期にわたって管理する」という特許業務の流れを踏まえると、特許庁システムは以下の3つに分類できる。

- 出願をサポートする窓口系システム
- 権利付与を担う基幹システム
- 安定的に膨大な情報を保持する原本系システム

(2) 品質項目

IPA(情報処理推進機構)が公開している『システム基盤の非機能要求に関する項目一覧』の『非機能要求グレード活用シート(2013年4月)』(全236項目)から特許庁システムにおいて重要な品質項目を下表に示す。

表 3.2-77 品質項目

項番	カテゴリ	項目	説明
1	可用性	サービス切替時間	想定し得る障害に対して必要な対策を施した状態で、業務再開までに要する時間。
2		業務継続の要求度	障害に対する業務継続の要求度。
3		RPO(目標復旧地点)	業務データの復旧地点。
4		RTO(目標復旧時間)	災害時等、業務継続対策を実施していないケースでの復旧時間。 ※サービス切替時間とは異なる。
5		稼働率	利用条件下でシステムが要求されたサービスを提供できる時間の割合。
6	性能・拡張性	レスポンス遵守率	定められたレスポンスタイムを遵守すべき割合。
7	運用・保守性	バックアップ保存期間	データ保全の観点でバックアップデータを保存すべき期間。 ※可用性の観点によるバックアップとは異なる。
8	セキュリティ	管理権限を持つ主体の認証	資産を利用する主体を識別するための、認証の度合い。

3.2.12.1 非機能要求レベルの設定に関するルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.3.2

指針1： 非機能要求レベルは、システムの分類別に示す「表 3.2-78」、「表 3.2-79」及び「表 3.2-80」の非機能要求レベルの基準値を参考に設計すること。

推奨1： 各サブシステムの非機能要求レベルは、「表 3.2-78」、「表 3.2-79」及び「表 3.2-80」の基準値を推奨する。

システムの分類毎に、重要品質項目の非機能要求レベルの推奨される基準値を示す。

(1) 出願をサポートする窓口系システム

出願をサポートする窓口系システムの非機能要求は、可用性要件が高いのが特徴である。

出願をサポートする窓口系システムに分類されるシステム例を以下に示す。

- オンライン受付システム
- インターネット出願受付システム
- オンライン発送システム
- 申請人登録システム
- 電子現金納付システム 等

下表に出願をサポートする窓口系システムの非機能要求レベルの基準値を示す。

表 3.2-78 非機能要求レベル基準値(出願をサポートする窓口系システム)

項番	カテゴリ	項目	該当レベル		
			最高レベル	レベル	概要
1	可用性	サービス切替時間	5	4	10分未満
2		業務継続の要求度	2	1	単一障害時は業務停止を許容せず、処理を継続させる。
3		RPO(目標復旧地点)	3	3	障害発生時点 (日次バックアップ+アーカイブからの復旧)
4		RTO(目標復旧時間)	4	3	6時間以内
5		稼働率	5	4	99.99%
6	性能・拡張性	レスポンス遵守率	5	0	遵守率を規定しない。
7	運用・保守性	バックアップ保存期間	5	5	永久保存
8	セキュリティ	管理権限を持つ主体の認証	3	3	複数回、異なる方式による認証

(2) 権利付与を担う基幹システム

権利付与を担う基幹システムの非機能要求は、性能保証要件が高いのが特徴である。

権利付与を担う基幹システムに分類されるシステム例を以下に示す。

- 振分システム
- DE料管理システム
- 特実方式審査システム
- 意匠・商標方式審査システム
- 特実審査周辺システム
- 料金納付管理システム 等

下表に権利付与を担う基幹システムの非機能要求レベルの基準値を示す。

表 3.2-79 非機能要求レベル基準値(権利付与を担う基幹システム)

項番	カテゴリ	項目	該当レベル		
			最高レベル	レベル	概要
1	可用性	サービス切替時間	5	4	10分未満
2		業務継続の要求度	2	1	単一障害時は業務停止を許容せず、処理を継続させる。
3		RPO(目標復旧地点)	3	3	障害発生時点 (日次バックアップ+アーカイブからの復旧)
4		RTO(目標復旧時間)	4	0	1営業日以上の停止を許容する。
5		稼働率	5	3	99.9%
6	性能・拡張性	レスポンス遵守率	5	2	80%
7	運用・保守性	バックアップ保存期間	5	1	1年未満
8	セキュリティ	管理権限を持つ主体の認証	3	3	複数回、異なる方式による認証

(3) 安定的に膨大な情報を保持する原本系システム

安定的に膨大な情報を保持する原本系システムの非機能要求は、データ保全要件が高いのが特徴である。

安定的に膨大な情報を保持する原本系システムに分類されるシステム例を以下に示す。

- 記録原本管理システム
- 記録ファイル管理システム
- 審判紙出力システム⁵⁵ 等

下表に安定的に膨大な情報を保持する原本系システムの非機能要求レベルの基準値を示す。

表 3.2-80 非機能要求レベル基準値(安定的に膨大な情報を保持する原本系システム)

項番	カテゴリ	項目	該当レベル		
			最高レベル	レベル	概要
1	可用性	サービス切替時間	5	4	10分未満
2		業務継続の要求度	2	1	単一障害時は業務停止を許容せず、処理を継続させる。
3		RPO(目標復旧地点)	3	3	障害発生時点 (日次バックアップ+アーカイブからの復旧)
4		RTO(目標復旧時間)	4	0	1営業日以上の停止を許容する。
5		稼働率	5	3	99.9%
6	性能・拡張性	レスポンス遵守率	5	0	遵守率を規定しない。
7	運用・保守性	バックアップ保存期間	5	5	永久保存
8	セキュリティ	管理権限を持つ主体の認証	3	3	複数回、異なる方式による認証

⁵⁵ データの永久保存という特性はないが、情報保持を補助するシステムであることから当分類とする。

3.2.13 システム機能に関する共通的なルール

本章では、システム機能(セキュリティやバックアップ等)に関する設計開発時に遵守すべきルールを定める。
本章にルールを記載するシステム機能の方式を以下に示す。

- (1) セキュリティ方式
- (2) 冗長化処理方式
- (3) システム運転管理方式
- (4) システム監視方式
- (5) バックアップ方式
- (6) リリース管理方式
- (7) ログ管理方式

特許庁内にはシステム機能に関するガイドライン等が既にあるため、原則、ガイドライン等に従うものとするが、ToBeアーキテクチャとの関係や、補足事項等を本章に記載する。

本章で参照するガイドライン等を以下に示す。

- 『ポリシー実施手順(開発編別紙「設計基準」)』
- 『LDAPアクセス運用』
- 『運用管理エージェント登録ガイドライン』
- 『バックアップ設計指針』
- 『特許庁システム リリースポリシー』
- 『パッチ適用方針』

3.2.13.1 セキュリティ方式設計のルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.3

指針1： セキュリティ方式設計は、『ポリシー実施手順(開発編別紙「設計基準」)』を遵守して設計すること。
また、「表 3.2-81 『ポリシー実施手順(開発編別紙「設計基準」)』とToBeアーキテクチャの関係」に示す事項は、同表の「ToBeアーキテクチャとの関係」に従い設計すること。

指針2： 仮想化ソフトのセキュリティは、「(3)仮想化ソフトのセキュリティ」に従い設計すること。

(1) 『ポリシー実施手順(開発編別紙「設計基準」)』への対応

セキュリティ方式設計は、『ポリシー実施手順(開発編別紙「設計基準」)』に準ずるものとする。

『ポリシー実施手順(開発編別紙「設計基準」)』の記載内容のうち、ToBeアーキテクチャとの関係を下表に示す。

表 3.2-81 『ポリシー実施手順(開発編別紙「設計基準」)』とToBeアーキテクチャの関係

章節	タイトル	概要	ToBeアーキテクチャとの関係
2.1	本人確認機能	全ての情報システムについて、本人確認を行う必要性の有無を検討すること。	「システム利用者(庁職員等)」の識別情報を利用し認証・認可を行う。 詳細は、「(2)D.認証・認可の処理方法」を参照のこと。
2.2	アクセス制御機能	全ての情報システムについて、情報システム及びそれに保存されている情報へのアクセス制御を行う必要性の有無を検討すること。	他サブシステムの個別データベースにアクセスするには他サブシステムが提供するサービスを経由してアクセスする。 共通的に利用するデータベースにアクセスするには、DBアクセス基盤サービス又は配付される基盤APIからアクセスする。 詳細は、「3.2.4.5 個別データベースの構成及びアクセスルール」及び「3.2.6.1 共有データベースの構成及びアクセスルール」を参照のこと。
2.5	権限管理機能	全ての情報システムについて、権限管理を行う必要性の有無を検討すること。	「システム利用者(庁職員等)」や「システム」毎に、アクセス可能なシステム構成要素を制限する。 詳細は「(2)E.認証・認可を実施するシステム構成要素間のアクセス」を参照のこと。
2.6	共有識別子(グループID)の利用	権限管理を行う必要があると認めた情報システムにおいて、共有識別子(ID等)の利用許可については、情報システム毎にその必要性を判断すること。	サブシステム毎に、アクセス可能なシステム構成要素を制限する。また、「システム」の識別情報を利用し認証・認可を行う。 詳細は、「(2)D.認証・認可の処理方法」を参照のこと。
2.9	証跡管理	全ての情報システムについて、証跡管理を行う必要性の有無を検討すること。	ToBeアーキテクチャで導入されるプログラムプロダクトも、証跡管理のため、ログを出力するように設定する。 また、不正アクセス等のセキュリティ監査のため、ログを出力する。 詳細は「3.2.13.7 ログ管理方式設計のルール」を参照のこと。

章節	タイトル	概要	ToBeアーキテクチャとの関係
2.10	監視機能	情報システムについて、情報セキュリティの侵害又はそのおそれのある事象の発生を監視する必要性の有無を検討し、必要があると認めた場合には、監視のために必要な措置を定めること。	ToBeアーキテクチャで導入されるプログラムプロダクトも、監視の対象とする。 詳細は「3.2.13.4 システム監視方式設計のルール」を参照のこと。
2.18	入出力データの妥当性確認機能	開発するソフトウェアにおいて処理するデータ及び入出力されるデータの情報セキュリティに関する妥当性を確認する機能の必要性の有無を検討し、必要と認めたときは、その方法を適切に（例えば、HTMLタグやスクリプト等として機能する不正な文字列や通信過程において生じたデータ誤り等、データ処理の障害になる情報がデータ内に含まれない状態であること等）設計しなければならない。	ToBeアーキテクチャにおいても同様に入出力データの妥当性の確認を行う。 具体的には以下のようなものがある。 ● 入力されたパラメータの長さや内容を検査し、無害化する機能を設けること。 ● OSの関数、SQLコマンド等の呼び出しといった出力情報に不正なデータの混入を排除すること。 ● 製品名及びそのバージョン、登録されているユーザID等、攻撃の糸口となり得る不必要な情報は出力しないこと。 ● なりすましによるアクセスを防止するため、適切なセッション管理を行うこと。

(2) 認証・認可

A. 目的

認証・認可の目的を以下に示す。

認証：本人性を確認し、なりすまし等を防止する。

認可：サービスの利用やリソースへのアクセス等に対する権限を制御し、権限を持たない者からのアクセスを防ぐ。

B. 対象

認証・認可の対象を下表に示す。

表 3.2-82 認証・認可の対象

項番	対象	目的
1	システム利用者 (庁職員等)	権限を持たない利用者からのアクセスを防ぐ。 認証・認可は、原則、『LDAPアクセス運用』に従うものとする。
2	システム	特許庁のシステム全体で共通的に利用可能な機能が、意図していないシステムからアクセスされることを防ぐ。

C. 識別情報の管理方法

認証・認可のための識別情報の管理方法を下表に示す。

表 3.2-83 識別情報の管理方法

項番	対象	識別情報の管理場所	管理する情報	取得方法
1	システム利用者 (庁職員等)	リソース管理サブシステム(既存システムでは、共通テーブル管理システム)	利用者、部署、権限等	LDAPより取得する。
2	システム	認証・認可する側のシステム内	システムの認証・認可情報	－

D. 認証・認可の処理方法

認証・認可の処理方法を下表に示す。

表 3.2-84 認証・認可の処理方法

項番	対象	認証・認可される側	認証・認可する側
1	システム利用者 (庁職員等)	① 職員等の識別情報(IDやパスワード等)を送信する。利便性向上のため、操作端末上でシングルサインオン(SSO)を実現するための認証基盤の利用を推奨する。	② 認証対象の識別情報を用い、リソース管理サブシステム(既存システムでは、共通テーブル管理システム)から職員等の情報(利用者、部署、権限等)をLDAPより取得することにより認証を行う(認証)。 ③ 取得した職員等の情報を用いてアクセス権限の制御を行う(認可)。
2	システム	① 自システムの識別情報(IDやパスワード等)を送信する。	② 認証対象のシステムの識別情報が正しいか照合し認証を行う(認証)。 ③ 認証と同様に、予め管理されている認可情報と照合し、アクセス権限の制御を行う(認可)。

(A) BPMSに対する認証認可の処理方法

BPMSに対する認証認可の対象は、以下とする。

- 認証認可の対象に製品の制約がない場合は、BPMSに対して「システム」での認証・認可とする。
- 「システム利用者(庁職員等)」による認証認可を必要とする製品の場合、「システム利用者(庁職員等)」での認証を行う。

例えば、BPMSに渡された「システム利用者(庁職員等)」の識別情報を使用して、BPMSがLDAPサーバで認証を行う場合等がある。

BPMSがLDAPサーバから「システム利用者(庁職員等)」の情報を取得するために「システム」での認証を必要とする場合がある。この場合は、BPMSからLDAPサーバに対しても「システム」での認証を許容する。このようなケースにおける認証・認可の手順の例を以下に示す。

- ① BPMSへの認証・認可は「システム」で行い、「システム利用者(庁職員等)」の識別情報はサービスの入力項目として渡す。ここでは「システム利用者(庁職員等)」自身に対する認証・認可を行うのではなく、役割に関する情報を取得することが目的であるため「システム利用者(庁職員等)」のIDを渡すだけでよく、パスワードを渡す必要はない。
- ② 次に、BPMSはLDAPサーバにアクセスし(ここで「システム」での認証を行う)、システム利用者(庁職員等)の役割に関する情報を取得し、「システム利用者(庁職員等)」の役割に応じたアクティビティの検索等に使用する。

E. 認証・認可を実施するシステム構成要素間のアクセス

認証・認可を実施するシステム構成要素間のアクセスを下表に示す。

表 3.2-85 システム構成要素間のアクセス

項番	対象	識別情報送信元システム構成要素	認証する側のシステム構成要素
1	システム利用者 (庁職員等)	● ブラウザ ● リッチクライアント	● プレゼンテーションロジック
2		● ブラウザ ● プレゼンテーションロジック ⁵⁶	● BPMS ⁵⁷
3	システム	● BPMS ● BPMS補完機能 ● プレゼンテーションロジック ● 業務アプリケーション(ユーザ) ● 業務アプリケーション(システム) ● 業務アプリケーション(バッチ) ● ESB	● 業務アプリケーション(システム)
4		● プレゼンテーションロジック ⁵⁸ ● BPMS ● BPMS補完機能 ● ESB ● 業務アプリケーション(システム) ⁵⁹ ● 業務アプリケーション(バッチ) ⁵⁹	● BPMS

⁵⁶ 「D.(A)BPMSに対する認証認可の処理方法」に示す「「システム利用者(庁職員等)」での認証を必要とする製品の場合」のみに使用するアクセスである。

⁵⁷ ブラウザからBPMSへのアクセスは原則、行わない。ただし、製品が提供する機能(BAM等)をブラウザから直接利用するような場合は庁職員等の認証を行うものとする。

⁵⁸ 当該職員に割り当てられているタスクを取得する等の処理を実現する場合、「システム利用者(庁職員等)」の識別情報をワークフローAPIに指定することで実現するものとする。

⁵⁹ アクセスパスの特例において認証・認可を実施するアクセスを指す。アクセスパスの特例については、「3.1.1.3.1.2 システム構成要素間のアクセスパス」を参照のこと。

項番	対象	識別情報送信元システム構成要素	認証する側のシステム構成要素
5	システム(前ページからの続き)	● プレゼンテーションロジック ● BPMS ● ESB ● 業務アプリケーション(システム)⁵⁹ ● 業務アプリケーション(バッチ)⁵⁹	● BPMS補完機能
6		● 業務アプリケーション(ユーザ) ● 業務アプリケーション(システム) ● 業務アプリケーション(バッチ)	● BRMS
7		● BPMS ● BPMS補完機能 ● 業務アプリケーション(ユーザ) ● 業務アプリケーション(システム) ● 業務アプリケーション(バッチ) ● 外部システム互換機能	● ESB
8		● 業務アプリケーション(ユーザ) ● 業務アプリケーション(システム) ● 業務アプリケーション(バッチ) ● ESB ● 外部システム互換機能	● DBアクセス基盤サービス ⁶⁰
9		● 業務アプリケーション(ユーザ) ● 業務アプリケーション(システム) ● 業務アプリケーション(バッチ)	● 個別データベース
10		● 業務アプリケーション(ユーザ) ● 業務アプリケーション(システム) ● 業務アプリケーション(バッチ) ● DBアクセス基盤サービス ● 外部システム互換機能	● 共有データベース

(3) 仮想化ソフトのセキュリティ

仮想化技術を導入した場合は、攻撃者になりすましを行い仮想化ソフトの特権権限を不正に取得した場合の全仮想マシンへの影響が懸念される。

そのため、仮想化ソフトについても他のプログラムプロダクトと同様にセキュリティ方式設計を行うこと。

⁶⁰ 基盤機能層は責務としてビジネスロジックを持たないため、業務に依存するシステム利用者を対象とした認証・認可ではなく、システムを対象とした認証・認可とする。基盤機能層の責務については「3.2.6基盤機能層及びデータベース層のルール」を参照。

3.2.13.2 冗長化処理方式設計のルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.3

規約1： クラスタ構成は、「(1)冗長化の方法(クラスタ構成)と選定ルール」のとおり選定すること。

特例1： より費用対効果の高い構成が実現できる場合は、選定ルールを逸脱することを許容する。

指針1： 仮想化における冗長化処理は、「(3)仮想化に対する考慮」に示す事項を考慮して設計すること。

指針2： セッションにおける冗長化処理は、「(4)セッションにおける考慮」に示す事項を考慮して設計すること。

推奨1： セッションレプリケーション等によるセッション保護機能は利用しないことを推奨する。

本節ではSLAに基づき冗長化する場合に、遵守すべき技術的なルールについて記載する。

(1) 冗長化の方法(クラスタ構成)と選定ルール

冗長化の方法(クラスタ構成)と選定ルールを下表に示す。

表 3.2-86 冗長化の方法

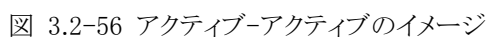
項番	クラスタ構成	選定ルール	補足説明
1	アクティブ-アクティブ	スケールアウトに適した論理ノード	負荷分散装置、又は高可用性ソフトウェアを用いて冗長化する。 例えば、Webサーバ等。
2	アクティブ-スタンバイ	スケールアウトに適していない論理ノード	例えば、DBサーバ等。

論理ノード毎にスケールアウトに適しているか否かは、「図 3.1-12 ソフトウェア構成図」を参照のこと。

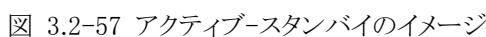
特例として、プログラムプロダクトの機能や構成の仕方によって、より費用対効果の高い構成がある場合には、上記の選定ルールを逸脱することを許容する。

A. アクティブ-アクティブの例

アクティブ-アクティブのイメージを下図に示す。



アクティブ-スタンバイのイメージを下図に示す。



(3) 仮想化に対する考慮

A. 仮想化対象外のサーバ

仮想化の対象外のサーバを以下に示す。以下のサーバは、仮想環境上に配置しないこと。

- ActiveDirectoryのドメインコントローラ
- 仮想マシンの監視サーバ
- NTPサーバ
- 仮想環境上での動作を保証していないソフトウェアを搭載するサーバ

B. 仮想化における冗長化技術について

仮想化技術のうち、冗長化に関わる技術を下表に示す。

表 3.2-87 仮想化における冗長化技術

項番	技術	概要	補足説明
1	ライブマイグレーション	ある物理マシン上で稼動する仮想マシンを、停止させることなく丸ごと別の物理マシン上に移動させること。又はそのような機能。	● 仮想化ソフトが提供する。 ● 通常、ハードウェアのメンテナンス時に使用する。 ● 厳密には切り替え時に瞬断(製品にもよるが一般的にはミリ秒単位以上)が発生するが、セッションやメモリ情報を全て引き継ぐことにより、ユーザに使用中の仮想マシンが移動したことは分からないようにすることも可能。
2	HAクラスタ	あるゲストOSを同一物理マシン上の別空間、別物理マシン上の仮想環境又は別物理マシン上の物理環境に移動させること。又はそのような機能。	● クラスタリングソフトが提供する。 ● 通常、障害発生時に使用する。 ● セッションやメモリ情報の引継ぎは、使用するクラスタリングソフトの機能に依存する。

障害発生時、ライブマイグレーションだけでも業務継続は可能であるが、仮想マシンの数が多かったり信頼性に対する要件が特に高い仮想マシンが含まれる場合には切り替え時間が要件を満たさない可能性もあるため、HAクラスタと組み合わせて使用することも視野に入れ、サブシステム毎に検討するものとする。

(4) セッションにおける考慮

画面系のオンライン処理では、複数ページにまたがる一連の処理を実現するため、UI層とプレゼンテーション層との間の通信においてセッションを利用する。通常、冗長化されたAPサーバのいずれか1つにセッションが管理される。

そのため、サーバをアクティブ-アクティブにしている場合、負荷分散装置で対象となるセッションを管理しているAPサーバへ振分けを行う。

ただし、セッションを管理しているサーバが故障した場合、セッションにアクセスできなくなり、故障したサーバのセッションを利用して業務を行っていたクライアントにはエラーが返却されることになる。対策としてはセッションレプリケーション機能等で、セッションをクラスタリングするか、セッション情報を別のサーバ上のRDBMSで管理する方法がある。

セッションレプリケーションのイメージを下図に示す。ただし、製品により実現方式は異なる。

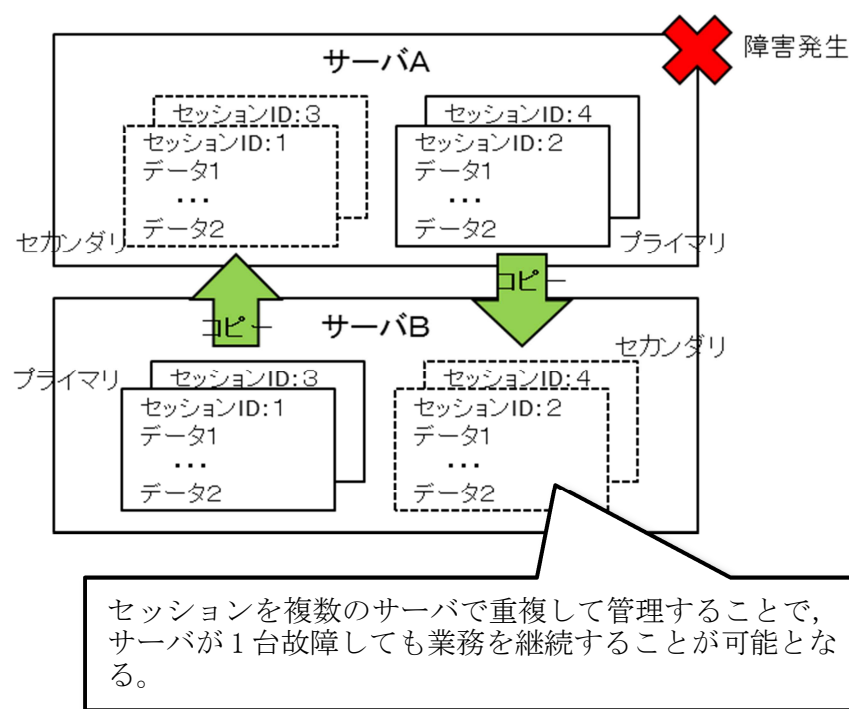


図 3.2-58 セッションレプリケーションのイメージ

しかし、これらの機能は、多くのアクセスが集中した場合に性能が劣化したり、レプリケーションの処理が新たな障害発生の原因になったりする場合がある。また、サーバの障害以外にも、Webブラウザのフリーズやユーザの誤った終了操作によってセッションが失われることもあるため、画面からあまり多くの情報を入力させないようにしたり、途中のページで固有DBに情報を保存し、後に再開可能とする等して、やり直しによる業務継続が可能であるような仕様にするべきである。

よって、ToBeアーキテクチャとしては、セッションレプリケーション等によるセッション保護機能は利用しないことを推奨する。

ただし、ユーザ利便性やSLAを遵守する上で、セッションレプリケーション機能等の導入が必要かサブシステム毎に検討するものとする。

3.2.13.3 システム運転管理方式設計のルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.3

指針1： システム運転管理方式は、『運用管理エージェント登録ガイドライン』を遵守して設計すること。

指針2： 業務閉塞は、「(1)B.業務閉塞のルール」に示すルールに従い設計すること。

推奨1： オンライン運転時間だけでなく、システム的な運転時間も特許庁全体で極力、統一することを推奨する。

特例1： やむを得ず、BPMS内部から経過時間指定での実行が必要な場合は、エラー終了状態になることを想定して開局時の処理を設計すること。

システム運転管理方式は、『運用管理エージェント登録ガイドライン』に準ずるものとする。

(1) 業務閉塞

ToBeアーキテクチャにおける業務閉塞に関わるルールを以下に示す。

A. 閉塞状態と運転時間

閉塞状態のステータスを下表に示す。

表 3.2-88 閉塞状態のステータス

ステータス	状態
開放	クライアントからのリクエストを受け付ける状態
予閉塞	クライアントからの新規リクエストを受け付けない状態
閉塞	クライアントからのリクエストを受け付けず、仕掛かり中処理がない状態

ToBeアーキテクチャでは、他サブシステムに公開されるサービスがあるため、オンライン運転時間が過ぎ、画面からの操作が閉塞された後もBPMS上のシステムタスクが完了するまでサービスの提供が必要となる。このようにオンライン運転時間後も、システムが動作している時間帯を「システム的な運転時間」と呼ぶ。運転時間のイメージを下図に示す。

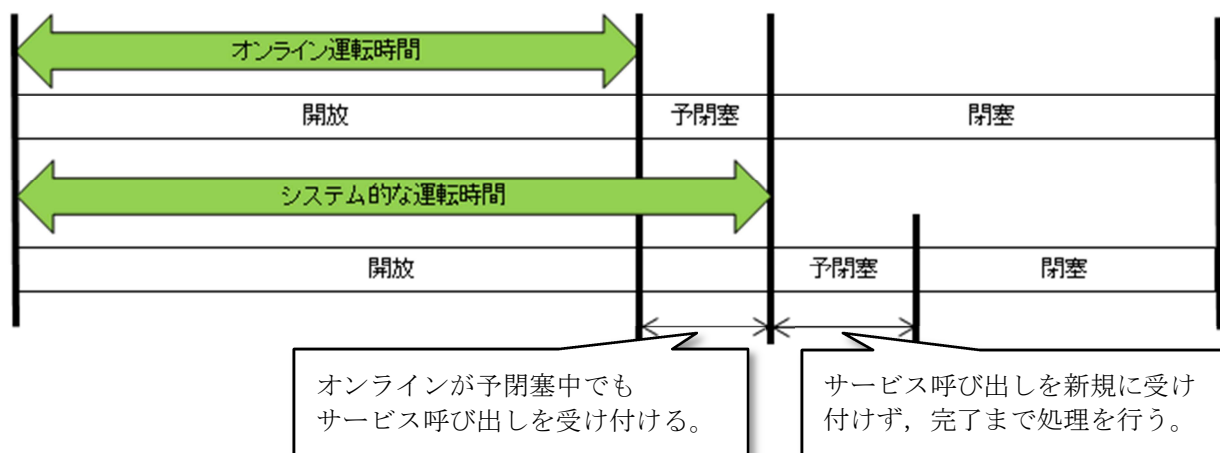


図 3.2-59 運転時間のイメージ

B. 業務閉塞のルール

業務閉塞のルールを以下に示す。

- 予閉塞から閉塞までの時間(仕掛かり中処理が正しく終了するまでの時間)については、サーバや業務処理によってリクエストの滞留量が異なるため、サブシステム毎に設計するものとする。ただし、上記を考慮しているに関わらず、閉塞時に仕掛かり中処理がある場合には、全てのオンライン処理を強制的に終了し、閉塞状態に入るものとする。
- 閉塞状態の切り替えは自動的にを行うため、ジョブ管理システムにスケジュール登録して自動で切り替えられるように設計すること。
- 他サブシステムに公開されるサービスを考慮し、オンライン運転時間だけでなく、システム的な運転時間も特許庁全体で極力、統一することを推奨する。
- BPMS内部から経過時間指定等での実行は行わないように設計すること(ビジネスプロセス上のタイマーイベントによる経過時間等の指定を行わないようにする。)
理由: BPMSでは、前回の実行からn時間後というように経過時間等を指定してBPMS内部からビジネスプロセスインスタンスへのアクセスをすることが可能である。このような設定をしている場合、オンライン処理が業務閉塞していてもBPMSのバックアップ等が正常に実施できない可能性があるためである。
特例: やむを得ず、経過時間指定等での実行が必要な場合は、エラー終了状態になることを想定して開局時の処理を設計すること。
- BPMSのビジネスプロセスインスタンスの停止／再開に時間がかかる場合は、オンラインバックアップが可能な製品を採用すること。
理由: 特許庁システムでは、多くのビジネスプロセスインスタンスが起動されるため、ビジネスプロセスインスタンスの停止や再開には時間がかかると想定される。BPMSを停止してバックアップする場合、業務閉塞中にはバックアップが完了しない可能性があるからである。

3.2.13.4 システム監視方式設計のルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.2

指針1： KPI等の測定、可視化、定期的分析は、「(2)稼動統計等の監視 A」に従って設計すること。

推奨1： KPI等を測定、可視化し定期的にKPI達成の分析・評価を支援する仕組みは、BI(Business Intelligence)、DWH(Data Warehouse)を使用して実現することを推奨する。

指針2： KPI等のリアルタイム確認は、「(2)稼動統計等の監視 B」に従って設計すること。

推奨1： 業務の実施状況をリアルタイムに確認し、ビジネスプロセス遂行上の問題発見や予防措置を支援する仕組みは、BPMS製品が持っているBAM(Business Activity Monitoring)を使用して実現することを推奨する。

指針3： SLA達成状況確認は、「(2)稼動統計等の監視 C」に従って設計すること。

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.3

指針1： ハードウェアやソフトウェア等の監視は、「(1)ハードウェアやソフトウェア等の監視」に従って設計すること。

(1) ハードウェアやソフトウェア等の監視

ToBeアーキテクチャにおいても、『運用管理エージェント登録ガイドライン』に従い、システム監視を行うものとする。

ToBeアーキテクチャで導入されるプログラムプロダクト⁶¹も、監視の対象とする。

なお、画面操作でレスポンスが悪い場合やバッチが想定時間内に完了しない等、パフォーマンスに問題が見つかった場合、原因箇所を特定できるよう対処を行うこと。以下に例を示す。

- ログファイルに処理の開始日時と終了日時が分かるように処理名や日時を出力する。
- プログラムプロダクトにパフォーマンス等を確認する機能がある場合は、それを利用する。

(2) 稼動統計等の監視

稼動統計等の監視に関するルールを以下に示す。

A. KPI等の測定、可視化、定期的分析

- KPI等を測定、可視化し定期的にKPI達成の分析・評価を支援する仕組みを構築する場合は、KPIの変更、分析データの変更等の環境変化に対して、容易に対応できる方式を採用すること。
 - BI(Business Intelligence)、DWH(Data Warehouse)

B. KPI等のリアルタイム確認

- KPI達成のための更なる取り組みとして、日々の業務の中で、業務の実施状況をリアルタイムに確認し、ビジネスプロセス遂行上の問題発見や予防措置について迅速な対応を行うことで、ビジネスプロセスの効率を向上させることが有効である。このような活動を支援する仕組みを構築する場合は、KPIの変更、分析データの変更等の環境変化に対して、容易に対応できる方式を採用すること。また、以下のプログラムプロダクトを利用することを推奨する。
 - BPMS製品が持っている、BAM(Business Activity Monitoring)

C. SLA達成状況確認

- SLAの達成状況を報告できるようにするため、応答時間を計測可能な箇所でログ出力し、機械的にSLA測定が可能な情報を蓄積するようにすること。ログについては、「(7)ログ管理方式」を参照のこと。

⁶¹ 詳細は、「表 3.2-71 プログラムプロダクト一覧」を参照のこと。

3.2.13.5 バックアップ方式設計のルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.3

指針1： バックアップ方式は、『バックアップ設計指針』に遵守して設計すること。

指針2： BPMSのバックアップ方式は、オフラインバックアップにする場合、ビジネスプロセスインスタンスの停止及び再開に時間がかかることを考慮して設計すること。

ToBeアーキテクチャにおいても『バックアップ設計指針』に従い、バックアップを行うものとする。

ただし、BPMSのバックアップ方式をオフラインバックアップにする場合、ビジネスプロセスインスタンスの停止及び再開に時間がかかることを考慮すること。

3.2.13.6 リリース管理方式設計のルール

スコープ： 階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス： 2.4.3

指針1： リリース管理方式は、『特許庁システム リリースポリシー』及び『パッチ適用方針』に遵守して設計すること。

ToBeアーキテクチャにおいても、以下に従いリリースやパッチ適用等が実現できるよう設計すること。

- 『特許庁システム リリースポリシー』
- 『パッチ適用方針』

3.2.13.7 ログ管理方式設計のルール

3.2.13.7.1 ログ出力の目的

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.3

指針1： OS, プログラムプロダクト, アプリケーションが出力するログを使用して「表 3.2-89 ログ出力の目的」に示す目的に沿ったログが出力されるよう設計すること。また、サブシステムの要件として、「表 3.2-89 ログ出力の目的」に示すログ出力の目的以外がある場合には、サブシステム毎に設計すること。

ログ出力の目的を下表に示す。

OS, プログラムプロダクト, アプリケーションが出力するログを使用して、下表の目的を達成できるように設計すること。また、サブシステムの要件として下表以外のログ出力の目的がある場合には、サブシステム毎に設計すること。

表 3.2-89 ログ出力の目的

目的		説明
エラー, 障害の対処	エラー, 障害検知	● 運用監視システムによるログ監視でエラー, 障害の検知に使用する。
	原因解析	● システム管理者がエラー, 障害時の原因解析情報として使用する。
	リカバリ	● システム管理者がエラー, 障害時のリカバリ作業に使用する。
キャパシティ管理	障害の予兆察知	● 障害の予兆察知のための情報(処理性能, リソース残量等)として使用する。
	設備条件整理	● 設備をスケールアウト又はスケールアップする際のサイジングの元データとして使用する。
監査, セキュリティ		● システムへのリクエストや処理内容を記録し, 監査の証跡として使用する。 ● セキュリティ違反の検知や処理内容把握, 経路追跡のために使用する。
SLA遵守確認	可用性管理 (稼働実績測定)	● 稼働率, 障害回復時間の非遵守回数等を計算するための元データとして使用する。
	性能管理 (応答時間測定)	● 処理の応答時間遵守率を計算するための元データとして使用する。
	ジョブ正常稼働 非遵守件数測定	● ジョブが正常に動作しなかった件数の取得のための元データとして使用する。
システム改善		● 機能の利用状況を確認し, 機能改善や廃止の元となる情報として使用する。
デバッグ		● プログラムが正しく動作しているかを開発者が確認するために使用する。

3.2.13.7.2 ログ出力情報

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.3

指針1： OS, プログラムプロダクト, アプリケーションがログ出力する情報は, 「表 3.2-90 ログ出力情報」の「ログに出力すべき情報」を参考に設計すること。

ログに出力すべき情報を下表に示す。

これらのログ情報は, OS, プログラムプロダクト及びアプリケーションを組み合わせで出力する。

表 3.2-90 ログ出力情報

目的		ログに出力すべき情報
エラー, 障害の対処	- エラー, 障害検知 - 原因解析 - リカバリ	- エラー情報 エラー, 障害発生時の状況, 発生箇所, 等 - 処理トレース情報 プログラムプロダクト, アプリケーションにおける処理の履歴, 等 - サーバ間通信処理情報 サーバ間 (内部サブシステム間連携及び外部連携) の通信履歴, 等 - データベース処理情報 データベースの処理履歴, トランザクションの成功/失敗, 等 - データベース更新ログ DBMSが管理する更新ログ, 等
キャパシティ管理	- 障害の予兆察知 - 設備条件整理	- サーバ間通信処理情報 サーバ間 (内部サブシステム間連携及び外部連携) の通信履歴, 処理時間, 等 - データベース処理情報 データベースの処理履歴, 処理時間, 等 - リソース情報等 ディスクやメモリ等の残量等の情報, 等
監査, セキュリティ		- サーバ間通信処理情報 サーバ間 (内部サブシステム間連携及び外部連携) の通信履歴, 等 - サーバ機器のイベント情報 サーバの起動及び停止, OS/プログラムプロダクトへのログイン及びログアウト, 特定リソースへのアクセス等, セキュリティポリシー上, 取得保管が義務付けられるログ, 等
SLA遵守確認	- 可用性管理 (稼動実績測定) - 性能管理 (応答時間測定) - ジョブ正常稼動非遵守件数測定	- サーバ機器のイベント情報 サーバの起動及び停止, 等 - サーバ間通信処理情報 サーバ間 (内部サブシステム間連携及び外部連携) の通信履歴, 処理時間, 等 - 運用監視システムのジョブ管理エラー情報 ジョブ処理のエラー履歴, 等
システム改善		利用状況履歴 画面アプリケーションの機能利用履歴, サービスの利用履歴, 等
デバッグ		- デバッグ情報 開発者がプログラムの動作確認をするためにプログラムに組み込んで出力した情報, 等 「エラー, 障害の検知」, 「原因解析」の目的で出力したログ

3.2.13.7.3 ログレベル

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.3

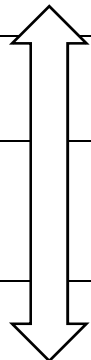
指針1： 出力するログのログレベルは、「(2)A.ログレベルに関わる指針」に従って設計すること。

規約1： 出力するログのログレベルは、「(2)B.ログレベルに関わる規約」のとおりとすること。

(1) ログレベルの定義

ログの重要度を示すログレベルの定義を下表に示す。

表 3.2-91 ログレベル

ログレベル	説明	ログ出力 文字列	レベル の高さ	通常運用時に 出力する対象
異常	システム管理者に通知し、エラーの調査や復旧のために操作が必要な例外・エラー（システムエラー、業務処理異常）が発生した場合に使用する。	ERROR		○
警告	利用者の再実行等で復旧可能な例外・エラー（タイムアウト等の業務的な警告等）が発生した場合に使用する。	WARN		○
情報	本番環境での通常運用時に、システムの動作状況に関する情報（システムの開始・終了時、読み込んだ設定ファイルの情報等）を出力する際に使用するもの。	INFO		○
デバッグ	開発環境や本番環境での動作確認時や異常が発生した際の解析時に、システムの動作状況に関する詳細な情報（プログラムのデバッグ情報としての、メソッドの入出力情報、処理の分岐の情報等）を出力する。	DEBUG		×

(2) ログレベルに関わるルール

A. ログレベルに関わる指針

ログレベルに関わる指針を以下に示す。

- 出力するログは、「(1)ログレベルの定義」に示すログレベルに分類して設計すること。
- 「情報」レベルのログは、ログ出力量が多くなりやすい傾向にあるため、ログ出力の目的を達成する必要最小限の出力量になるよう設計すること。

B. ログレベルに関わる規約

ログレベルに関わる規約を以下に示す。

- デバッグ情報をエラー解析に利用したい等、出力するログのレベルを切り替えたい場合、設定ファイル等を変更することで、出力するログレベルを切り替えられるようにすること。

3.2.13.7.4 運用監視システムとの連携

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.3

指針1： OS, プログラムプロダクト, アプリケーションが出力するログと運用監視システムとの連携は, 「(2)運用監視システムとの連携ルール」に従って設計すること。

OS, プログラムプロダクト, アプリケーションが出力するログが運用監視の対象となる。運用監視システムとの連携ルール等を以下に示す。

(1) 運用監視システムとの連携方法と重要度

運用監視システムは, 重要度に応じて予め設定された特定のキーワードがログに出力されたことを検知し, システム管理者へ異常等の情報を通知する。

運用監視における重要度は, 『運用管理エージェント登録ガイドライン』に定義されている。下表に抜粋を示す。

表 3.2-92 運用監視における重要度

重要度	説明	例
重大	複数システム又は複数端末へ影響があり即座に復旧(リカバリ)を実施する必要があるエラー	● ノードダウン ● クラスタ切替発生
大	単一システム内又は特定の端末のみで影響があるエラーで即日中に対処を実施する必要があるエラー	● プロセスダウン ● プログラムプロダクトエラー ● アプリケーションエラー
中	業務影響のないエラーで対処要否を判断する必要があるエラー	● リソース閾値超過 ● 管理者ログイン
小	業務影響のないエラーで対処の必要がないエラー, 又は, システム状態の通知	● 業務量多の警告 ● 故障回復通知(リンクアップ) ● サービス起動通知
無	エラーではないメッセージ	● ジョブの起動・正常終了

(2) 運用監視システムとの連携ルール

運用監視システムとの連携ルールを以下に示す。

- ログレベル等のログ内容を使用することで, 重要度に応じたキーワードを運用監視システムに設定できるようにログ出力内容を設計すること。
- 運用開始後における監視対象の変更は, 運用監視システムのキーワード設定を変更することで, 柔軟に対応し, 極力, アプリケーション側のログ出力機能を変更しなくてもよいものとする。

3.2.13.7.5 ログ出力項目とフォーマット

スコープ:	階層定型化サブシステム及びインタフェース定型化サブシステム
プロセス:	2.4.3
規約1:	ログ出力項目及びフォーマットは、「(1)ログ出力項目及びフォーマットのルール」のとおりとすること。
指針1:	個別ヘッダ及びメッセージは、「(2)個別ヘッダ及びメッセージのルール」に従って設計すること。

(1) ログ出力項目及びフォーマットのルール

ログ出力項目及びフォーマットのルールを以下に示す。

- 運用監視システムとの連携を考慮し、1行で出力すること。運用監視システムが監視しない情報で、かつ可読性が向上する場合、改行を行うこと。
- 下表に示すログ出力項目とフォーマットに従い、ログ出力すること。

表 3.2-93 ログ出力項目とフォーマット

種別	項目	説明とフォーマット	例
ヘッダ		ログの属性を表す項目を出力する。 フォーマット: ● ヘッダの1項目毎に角括弧(“[”及び“]”)で囲むこと。 ● 項目間はタブを入れること。	—
	ログ出力日時	ログ出力日時(年月日時分秒ミリ秒)を示す。 フォーマット: yyyy/MM/dd△HH:mm:ss.SSS	[2014/10/20△01:02:03.456]
	ログレベル	「表 3.2-91 ログレベル」で定義したログレベルを示す。同表の「ログ出力文字列」に示す文字列を出力する。 フォーマット: ● 5文字で出力すること。 ● 文字長が5文字未満の場合、語尾をスペースで埋めること。	[INFO△]
	個別ヘッダ (1つ又は複数)	エラーや統計等、対象のログを特定するための情報。ログ出力の目的毎に出力する項目が異なる。項目名と内容で構成される。定義済みの個別ヘッダの項目名は「表 3.2-94 定義済み個別ヘッダ」を参照のこと。 フォーマット: 「項目名」:「内容」	[txid:1234567890]
メッセージ		ログ出力の目的に応じて、適切な内容を出力する。	処理が開始されました。

※△は半角スペースを表す。

出力されるログのイメージを以下に示す。

[2014/10/20△01:02:03.456] [INFO△] [txid:1234567890] [proc:方式審査処理] 処理が開始されました。

※△は半角スペースを表す。また、項目間の区切りはタブである。

図 3.2-60 出力されるログのイメージ

予め定義済みの個別ヘッダを下表に示す。

表 3.2-94 定義済み個別ヘッダ

個別ヘッダ	項目名	説明	例
トランザクション識別子	txid	「エラー、障害の対処」、「監査、セキュリティ」等の目的でログ解析する際、複数のサーバのログを横断して、処理の履歴が追跡できるように出力した文字列のこと。同じ処理開始契機で動作したログに同一の識別子を出力する。 トランザクション識別子の発生方式と引き継ぎ方式を設計する必要がある。「エラー、障害の対処」、「監査、セキュリティ」を目的とするログで複数のログに横断的に出力される処理の場合は、必須で出力すること。	[txid:1234567890]
処理名／ID	proc	どこで発生した事象かを示す。 処理名称、処理モジュール名、画面名やそのID等	[proc:方式審査処理]
起動契機情報	ユーザIDの場合:uid ジョブIDの場合:jid	起動契機になった処理の情報を示す。 画面の場合はユーザID、バッチの場合はジョブIDやディレード処理要求ID等	[uid:user001]
通信元／先情報	通信元の場合:src 通信先の場合:dst	受信時の通信元や送信時の通信先の情報を示す。 URLやホスト名等	[dst:http://xxxx/yyyy]
接続先データベース情報	db	接続先のデータベースを示す。 接続先のデータベース名やID等	[db:共有DB01]

(2) 個別ヘッダ及びメッセージのルール

個別ヘッダ及びメッセージのルールを以下に示す。

- 個別ヘッダ及びメッセージの内容は、セキュリティの観点から、個人情報、非公開情報は出力しないように設計する。
- 定義済み個別ヘッダ以外が必要な場合、ログ出力の目的に応じて、サブシステム毎に設計すること。

3.2.13.7.6 ログ運用

スコープ：階層定型化サブシステム及びインタフェース定型化サブシステム

プロセス：2.4.3

規約1： ログ出力媒体は、「(1)ログ出力媒体」に示す媒体とすること。

規約2： ログ出力単位は、「(2)ログ出力単位」に示す単位とすること。

特例1： 「(2)ログ出力単位」に示す特例を許容する。

指針1： ログローテーションは、「(3)ログローテーション」に従い、設計すること。

指針2： バックアップは、「(4)バックアップ」に従って設計すること。

指針3： ログの参照用保持は、「(5)参照用保持」に従って設計すること。

指針4： ログの参照用権限は、「(6)ログ参照権限」に従って設計すること。

(1) ログ出力媒体

ファイルに出力すること。

(2) ログ出力単位

ログの出力単位は、以下のとおりとする。

- ログ出力の目的ごとに、異なるファイルに出力すること。また、可読性や、後述する「(4)バックアップ」の頻度、「(5)参照用保持」の期間を考慮して、より細かい単位に出力先ファイルを区別してもよい。
- 複数のプロセスが1つのファイルに対して同時にログ出力しないこと。

複数プロセスから1ファイルへの出力について、特例を以下に示す。

- ジョブ処理等、アプリケーションの動作時間帯が重ならない等の理由で、ログ出力処理が競合しない場合、複数プロセスで1ファイルに出力することを許容する。
- 複数プロセスで同時にログ出力する場合でも、1ファイルに出力する要件があった場合は、以下の条件を満たす場合は、許容する。
 - 排他制御等を設け、ログの損失なく、ログ出力、ローテーション等ができる。
 - 性能要件を満たすことができる。

(3) ログローテーション

バックアップの周期やログ参照の利便性から、ログローテーションは以下のように行う。

- ログファイルの容量上限値を設け、容量上限値を超過したら、ローテーションを行うこと。
- ログファイルの容量上限値は、運用設計の結果を反映できるよう、設定値で変更できるようにしておくこと。
- 原則、ログのバックアップ周期に合わせて、適切な周期でローテーションを行うこと。
ただし、他の契機(OS/プログラムプロダクトの起動・停止時等)でローテーションを行うことを許容する。
- ローテーションしても運用監視サーバによる監視が継続できるよう、最新ログのファイル名は固定とし、ローテーション後の古いファイルはファイル名をリネームすること。
- ローテーション後の古いファイルは、日付や連番等を付与し、最新ファイルとの対応付けと時系列が分かるファイル名にすること。

例：最新ファイル名： logfile.log

ローテーション後のファイル名： logfile.log.20140102102030

(4) バックアップ

『バックアップ設計指針』に従い、バックアップを行うものとする。

(5) 参照用保持

ログの参照保持については以下のように設計する。

- 「エラー，障害の対処」を目的としたログについては，エラー解析のしやすさを考慮し，参照しやすい保存箇所，保存形式で保持する期間を設けること。

例：3週間分のログはログ出力フォルダに保持する 等

- ディスク容量を考慮し，参照保持期間を過ぎたログを削除する仕組みを設けること。

(6) ログ参照権限

「監査，セキュリティ」を目的にしたログは，不当な消去，窃取，改ざん，業務上の必要なくアクセス等されないよう，ログには適切なアクセス制御を行うこと。

項番	チェック項目				執筆者チェック欄				レビューでのチェック欄				
	頁	大項目 (プロセス)	中項目(スコープ)		強制力	ID	チェック項目	反映箇所	反映内容	反映日	反映者	評価	確認日
			定型化分類	システム構成要素等									
1	508	2.3.2	階層定型化サブシステム及びインタフェース定型化サブシステム		指針	23201450-DG1-N1	非機能要求レベルは、システムの分類別に示す「表 3.2-78」、「表 3.2-79」及び「表 3.2-80」の非機能要求レベルの基準値を参考に設計すること。						
					推奨	23201450-DG1-R1	各サブシステムの非機能要求レベルは、「表 3.2-78」、「表 3.2-79」及び「表 3.2-80」の基準値を推奨する。						
2	131	2.3.2	階層定型化サブシステム		規約	23200560-RG1-N1	ビジネスプロセスの表記は、BPM2.0のプロセス図を使用すること。						
3	132	2.3.2	階層定型化サブシステム		指針	23200570-DG1-N1	BPMsと業務アプリケーションの組み合わせは、「表 3.2-16 BPMsと業務アプリケーションとの組合せの代表的なケース」に示すケースに該当する場合、以下の(1)～(3)に示すBPMsと業務アプリケーションの設計指針に従い、ビジネスプロセスを変更できよう設計を行うこと。						
4	149	2.3.2	階層定型化サブシステム		規約	23200620-DG1-N1	ビジネスプロセスの記述に使用するBPM要素は、「表 3.2-19」～「表 3.2-28」に示す使用可否、及び制約事項のとりを使用すること。						
5	163	2.3.2	階層定型化サブシステム		指針	23200630-DG1-N1	BPM要素のラベル名は、「(1)ラベル命名ルール」、「(2)用語に関するルール」に従って付与すること。						
6	167	2.3.2	階層定型化サブシステム		指針	23200640-DG1-N1	ビジネスプロセスは、「(1)ビジネスプロセスの記載統一ルール」に示す単位で設計すること。						
					規約	23200650-RG1-N1	ビジネスプロセスは、「(2)同じ意味であるが表記上異なるビジネスプロセスの記載統一ルール」のとり記載すること。						
					指針	23200660-DG2-N1	ビジネスプロセスのアクティビティは、「(3)アクティビティの単位のルール」に示す単位で設計すること。						
					規約	23200670-RG2-N1	BPM要素は、「(4)レイアウトルール」の「A.時系列にあった配置ルール」、「C.BPM要素の重なり防止のルール」、「D.プールの配置ルール」、「E.イベント・サブプロセスの配置ルール」のとり配置すること。						
					指針	23200680-DG3-N1	レーン配置順は、「(4)レイアウトルール B.レーンの配置ルール」に従い、登場人物の分類別に可読性を考慮した配置順序を決定し、各ビジネスプロセスで統一した配置順にすること。						
					推奨	23200680-DG3-R1	役割のレーンを配置する場合、「下層の役割から上層の役割の順」に上から下へレーンを配置すること。						
7	177	2.3.2	階層定型化サブシステム		規約	23200690-RG1-N1	ビジネスプロセスは、ビジネスプロセス設計時に「表 3.2-31 ビジネスプロセスパターン事例一覧」に示す適用シーンに合致した場合、当該パターン事例を適用すること。						
8	219	2.3.3	インタフェース定型化サブシステム	外部システム連携	規約	23200860-DG1-N1	外部システム連携機能は、「(1)外部システム連携機能」に示す機能とすること。						
					指針	23200870-DG1-N1	外部システム連携機能は、「(2)外部システム連携機能が提供することを禁止する機能」に従い設計すること。						
9	142	4.2	階層定型化サブシステム及びインタフェース定型化サブシステム		規約	24200020-RG1-N1	サブシステムの機能は、「表 3.1-2 システム構成要素の種類及び責務」に示された責務を担うシステム構成要素から実現すること。また、「3.1.1.3.1.3 サブシステムの範囲」に示される分類毎のサブシステムの機能は、それぞれ「表 3.1-5 サブシステムの分類とシステム構成要素」に掲載されたシステム構成要素から実現すること。						
					規約	24200030-RG2-N1	システム構成要素間のアクセスは、「3.1.1.3.1.2 (1)多階層構造の概要に基づくアクセス」に示すアクセスパスとすること。また、「3.1.1.3.1.2 (3)アクセスパスの制限事項」に示すとり制限すること。						
					特例	24200030-RG2-S1	「3.1.1.3.1.2 (2)アクセスパスの特例」に示すアクセスパスは、下位層のシステム構成要素から上位層のシステム構成要素へのアクセスを許可すること。						
					規約	24200040-RG3-N1	全システム構成要素のソフトウェア構成は、「図 3.1-12 ソフトウェア構成図」に示す構成とすること。						
					規約	24200050-RG1-N1	サブシステムの処理方式は、「表 3.1-3 処理方式による分類」に示す処理方式とすること。						
10	76	2.4.2	階層定型化サブシステム及びインタフェース定型化サブシステム		指針	24200270-DG1-N1	BPMsを介したサブシステム間連携方式は、「3.1.2.5.1.1 (1)連携方式」、「3.1.2.5.1.1 (2)設計・開発時の考慮点」に従って設計すること。						
					指針	24200280-DG2-N1	BPMsとジョブ管理システム間の連携方式は、「3.1.2.5.1.2 (1)連携方式」、「3.1.2.5.1.2 (2)設計・開発時の考慮点」に従って設計すること。						
					指針	24200290-DG3-N1	バッチ処理方式のサブシステム間連携方式は、「3.1.2.5.1.3 (1)連携方式」、「3.1.2.5.1.3 (2)設計・開発時の考慮点」に従って設計すること。						
					指針	24200300-DG4-N1	外部システム連携を介した外部システム連携方式は、「3.1.2.5.1.4 (1)連携方式」、「3.1.2.5.1.4 (2)設計・開発時の考慮点」に従って設計すること。						
11	95	2.4.2	階層定型化サブシステム及びインタフェース定型化サブシステム		指針	24200340-DG1-N1	「(4)サービス化の対象」に示すシステム構成要素の提供機能は、「(3)サービス化対象外の指針」に該当しない場合、サービスインタフェースを設けるよう設計すること。						
					特例	24200340-DG1-S1	「(3)サービス化対象外の指針」に該当するものについては、サービス化の対象外とすることを許容する。						
					規約	24200350-RG1-N1	サービスインタフェースは、「(5)サービスインタフェースの提供」のとり提供すること。						
12	98	2.4.2	階層定型化サブシステム及びインタフェース定型化サブシステム		指針	24200360-RG1-N1	「表 3.2-2 プロトコル及び電文形式 (階層定型化サブシステム)」及び「表 3.2-3 プロトコル及び電文形式 (インタフェース定型化サブシステム)」に示すシステム構成要素は、当該表に示すプロトコル及び電文形式を利用するインタフェースを設けること。						
					推奨	24200360-RG1-R1	「表 3.2-2 プロトコル及び電文形式 (階層定型化サブシステム)」及び「表 3.2-3 プロトコル及び電文形式 (インタフェース定型化サブシステム)」に示す階層プロトコル及び電文形式を推奨する。						
					特例	24200360-RG1-S1	「表 3.2-2 プロトコル及び電文形式 (階層定型化サブシステム)」及び「表 3.2-3 プロトコル及び電文形式 (インタフェース定型化サブシステム)」に示す特例を許容する。						
					規約	24200370-RG2-N1	FTP(S)プロトコルは、「(5)FTP(S)プロトコルの利用制限事項」に示すとり利用すること。						
13	104	2.4.2	階層定型化サブシステム及びインタフェース定型化サブシステム		指針	24200380-DG1-N1	インタフェース名は、「(1)インタフェース名の構成」に示す用語の組み合わせに従い、命名すること。						
					指針	24200390-DG2-N1	インタフェース名は、「(2)インタフェース名の重複と同名異義・異音同義語の禁止事項」に示す禁止事項に抵触しないよう、命名すること。						
14	106	2.4.2	階層定型化サブシステム及びインタフェース定型化サブシステム		指針	24200400-DG1-N1	入出力項目名は、「(1)入出力項目名の命名ルール」に従って命名すること。						
15	232	2.4.2	階層定型化サブシステム及びインタフェース定型化サブシステム		指針	24200990-DG1-N1	用語は、「(1)用語の種類」に示す種類を使用すること。						
					指針	24201000-DG2-N1	用語の表記は、「(2)用語の表記」に示す表記ルールに従い、表記すること。						
					規約	24201010-RG1-N1	用語の整備は、「(3)A.整備規約」に示す規約のとり整備すること。						
					指針	24201020-DG3-N1	用語の整備は、「(3)B.整備指針」に示す指針を遵守すること。						
16	280	2.4.2	階層定型化サブシステム及びインタフェース定型化サブシステム		規約	24201350-RG1-N1	内部システムで扱う文字コードは、「(1)使用する文字コード」に示す文字コードを採用すること。						
					規約	24201360-RG2-N1	内部システムで使用する文字は、「(2)使用可能な文字」に示す文字とすること。						
					規約	24201370-RG3-N1	内部システムで扱う文字コードの仕様への対応は、「(3)文字コードの仕様への対応」のとりとすること。						
					規約	24201380-RG4-N1	文字コードは、「(5)文字コードを扱う対象」に示す対象に適用すること。						
					特例	24201390-RG4-S1	プログラム・プログラムのログや設定ファイル等、業務処理に直接影響しないものは、「(1)使用する文字コード」に示した文字コードを指定しないことを許容する。						
					規約	24201390-RG1-N1	特許庁外字の扱い、「(4)特許庁外字の扱い」のとりとすること。						

【●●システム構築】

特許庁アーキテクチャ標準仕様書 第1.0版 適合確認チェックリスト

項番	頁	大項目 (プロセス)	中項目(スコープ)		小項目	チェック項目		反映箇所	執筆者チェック欄		レビューでのチェック欄					備考	
			定型化分類	システム構成要素等		強制力	ID		反映内容	反映日	反映者	評価	確認日	確認者	コメント票No.		
17	283	2.4.2	階層定型化サブシステム及び インタフェース定型化サブシ ステム		3.2.13.4 システム監視方式設計の ルール	指針	E4201530-DG1-N1	KPI等の測定、可視化、定期的分析は、「(2)稼働統計等の監視 A」に従って設計すること。									
						推奨	E4201530-DG1-R1	KPI等を測定、可視化し定期的にKPI達成の分析・評価を支援する仕組みは、BI (Business Intelligence) 、 DWH (Data Warehouse) を使用して実現することを推奨する。									
						指針	E4201540-DG2-N1	KPI等のリアルタイム確認は、「(2)稼働統計等の監視 B」に従って設計すること。									
						推奨	E4201540-DG2-R1	業務の実施状況をリアルタイムに確認し、ビジネスプロセス遂行上の問題発見や予防措置を支援する仕組み は、BPM製品が持っているBAM (Business Activity Monitoring) を使用して実現することを推奨する。									
18	228	2.4.2	階層定型化サブシステム及び インタフェース定型化サブシ ステム	ビジネスルール管理	3.2.7.1.1 ビジネスルールをアプリ ケーションから切り離すための仕組 みとして利用する技術	規約	E4200930-RG1-N1	ビジネスルールをアプリケーションから切り離すための仕組みは、BPMを用いること。									
19	234	2.4.2	階層定型化サブシステム及び インタフェース定型化サブシ ステム	外部システム連携	3.2.8.2.1 テーブル名の命名ルール	指針	E4201030-DG1-N1	テーブル名は、「(1)テーブル名の構成」に示す用語の組み合わせに従い、命名すること。									
						指針	E4201040-DG2-N1	テーブル名は、「(2)テーブル名の重複と同音異義・異音同義語の禁止事項」に示す禁止事項に抵触しないよう 、命名すること。									
20	235	2.4.2	階層定型化サブシステム及び インタフェース定型化サブシ ステム	外部システム連携	3.2.8.2.2 カラム名の命名ルール	指針	E4201050-DG1-N1	カラム名は、「(1)カラム名の構成」に示す用語の組み合わせに従い、命名すること。									
						指針	E4201060-DG2-N1	カラム名は、「(2)カラム名の重複と同音異義・異音同義語の禁止事項」に示す禁止事項に抵触しないよう 、命名すること。									
21	108	2.4.2	階層定型化サブシステム		3.2.2.1 クライアント構成	規約	E4200410-RG1-N1	クライアント構成は、「(2)クライアント構成の選定ルール」のとおり選定すること。									
22	110	2.4.2	階層定型化サブシステム		3.2.2.2 画面とリクエストの作成単 位に関するルール	指針	E4200420-DG1-N1	ブラウザからの要求を受けるリクエストとその結果として表示される画面は、BPMのユーザタスク毎にリクエ ストと画面のまとまりを割り当てるように作成単位を定めて設計すること。									
						特例	E4200420-DG1-S1	複数のユーザタスクを同時に進めるような業務上については、これらのユーザタスクに対し、同一のリクエ ストと画面のまとまりを割り当てても良い。									
23	114	2.4.2	階層定型化サブシステム		3.2.2.4.1 画面のアクセシビリティ に関するルール	指針	E4200440-DG1-N1	画面のアクセシビリティは、「表 3.2-8 アクセシビリティに関する対応方針」に従って設計すること。									
24	115	2.4.2	階層定型化サブシステム		3.2.2.4.2 画面の定義に関するル ール	指針	E4200450-DG1-N1	画面の単位は、「(1)画面の単位」に従って設計すること。									
						指針	E4200460-DG2-N1	画面間の遷移は、「(2)画面間の遷移」に従って設計すること。									
						推奨	E4200460-DG2-R1	「表 3.2-9 代表的な画面遷移のパターン」に示す画面遷移のパターンを推奨する。									
						指針	E4200470-DG3-N1	ウィンドウの制約は、「(3)画面遷移に伴うウィンドウの扱い」に従って設計すること。									
25	119	2.4.2	階層定型化サブシステム		3.2.2.4.3 画面の内容に関するル ール	指針	E4200480-DG1-N1	画面レイアウトは、極力統一したレイアウトになるように設計すること。									
						推奨	E4200480-DG1-R1	画面レイアウトは、「(2)画面のレイアウト」に示すレイアウトを推奨する。									
						指針	E4200490-DG2-N1	画面部品は、「(3)画面部品の扱い」に従って設計すること。									
26	125	2.4.2	階層定型化サブシステム		3.2.2.5.1 帳票の出力項目に関する ルール	指針	E4200510-DG1-N1	帳票の出力項目は、見た目を極力、統一するように設計すること。									
						推奨	E4200510-DG1-R1	「表 3.2-15 帳票の設定項目の推奨の設定値」に従って設計することを推奨する。									
						特例	E4200510-DG1-S1	「表 3.2-15 帳票の設定項目の推奨の設定値」に示す特例を許容する。									
27	127	2.4.2	階層定型化サブシステム		3.2.2.5.2 帳票のレイアウトに関 するルール	指針	E4200520-DG1-N1	帳票のレイアウトは、極力、統一するように設計すること。									
						推奨	E4200520-DG1-R1	共通的に出力する項目の出力位置は、帳票上部とすることを推奨する。									
28	128	2.4.2	階層定型化サブシステム		3.2.2.5.3 印刷用紙に関するルール	指針	E4200530-DG1-N1	帳票の用紙サイズ、印刷の向きは、極力、統一するように設計すること。									
						推奨	E4200530-DG1-R1	「(1)印刷用紙」に示す用紙サイズ、印刷の向きを推奨する。									
29	129	2.4.2	階層定型化サブシステム		3.2.2.5.4 帳票の管理に関するル ール	規約	E4200540-RG1-N1	帳票をファイルとして管理する必要がある場合のファイル形式は、PDFのファイル形式で出力すること。									
30	131	2.4.2	階層定型化サブシステム		3.2.3.1.1 ビジネスプロセス管理 として利用する技術	規約	E4200550-RG1-N1	ビジネスプロセスの実行制御を一元管理する仕組みは、BPMとすること。									
31	187	2.4.2	階層定型化サブシステム		3.2.4.1 サービスの粒度	指針	E4200720-DG1-N1	サービスの粒度は、「(1)サービスの粒度のルール」に従って設計すること。									
32	188	2.4.2	階層定型化サブシステム		3.2.4.2 サービスインタフェース の粒度	指針	E4200730-DG1-N1	サービスインタフェースの粒度は、「(1)サービスインタフェースの粒度のルール」に従って設計すること。									
						規約	E42007400-RG1-N1	サービスインタフェース呼び出しは、「(3)アクセスバスの制限事項に関するルール」に示すとおりとすこ と。									
33	196	2.4.2	階層定型化サブシステム		3.2.4.4.1 特許庁開発フレームワー クの適用範囲	規約	E4200760-RG1-N1	開発対象のアプリケーションに適用する特許庁開発フレームワークは、「(3)業務アプリケーションに適用 するフレームワーク」に示すとおりとすること。									
34	199	2.4.2	階層定型化サブシステム		3.2.4.4.2 特許庁開発フレームワー クの機能	規約	E4200770-RG1-N1	特許庁開発フレームワークの機能は、「(2)機能概要」に示す機能に準拠したものであること。									
						規約	E4200780-RG2-N1	フレームワーク機能の配置は、「(3)フレームワークに対する配置」のとおりとすること。									
						指針	E4200790-DG1-N1	サブシステム固有機能は、「(2)E.サブシステム固有フレームワーク機能」に従い、設計開発すること。									
35	213	2.4.2	階層定型化サブシステム		3.2.4.5.1 個別データベースの構成	規約	E4200810-RG1-N1	個別データベースの構成は、「(1)個別データベースの構成ルール」のとおりとすること。									
36	214	2.4.2	階層定型化サブシステム		3.2.4.5.2 個別データベースへのア クセスルール	規約	E4200820-RG1-N1	個別データベースへのアクセスは、「(1)個別データベースへのアクセスルール」に準拠すること。									
37	224	2.4.2	階層定型化サブシステム		3.2.6.1 共有データベースの構成 及びアクセスルール	規約	E4200900-RG1-N1	共有データベースの構成は、「(1)共有データベースの構成とデータ管理のルール」で示すデータの単位と すること。									
						規約	E4200910-RG2-N1	共有データベースへのアクセスは、「(2)共有データベースへのアクセスルール」に準拠すること。									
38	239	2.4.2	階層定型化サブシステム		3.2.8.4.1 多件一連の手続に関する データの扱い	指針	E4201000-DG1-N1	一つの書頭が複数の事件等に関連する手続となる業務要件の実現は、その手続を表すエンティティについ て、「表 3.2-67 多件一連の手続に関するデータの扱い方針」に示す方針に従い、データ構造を設計するこ と。									
39	240	2.4.2	階層定型化サブシステム		3.2.8.4.2 デジタルコンテンツに関 するデータの扱い	指針	E4201100-DG1-N1	デジタルコンテンツを管理するエンティティは、「表 3.2-68 デジタルコンテンツに関するデータの扱い方 針」に示す方針に従い、データ構造を設計すること。									
40	241	2.4.2	階層定型化サブシステム		3.2.8.4.3 主キーが決定してい ないエンティティの扱い	規約	E4201110-RG1-N1	エンティティは、主キーが決定していない場合、必ず主キーを設定すること。									
						指針	E4201120-DG1-N1	主キーの設定は、「表 3.2-69 主キーの設定条件」に示す条件を満たすように設計すること。									
41	242	2.4.2	階層定型化サブシステム	共有データベース管理	3.2.8.5 処分が決定した事件デー タの扱いのルール	指針	E4201130-DG1-N1	処分が決定した事件データは、「表 3.2-70 処分が決定した事件データの扱い方針」に示す方針に従い設計 すること。									

項番	頁	大項目 (プロセス)	中項目(スコープ)		小項目	チェック項目		チェック項目	執筆者チェック欄				レビューでのチェック欄					備考
			定型化分類	システム構成要素等		強制力	ID		反映箇所	反映内容	反映日	反映者	評価	レビューでのチェック欄				
														確認日	確認者	コメント票No.		
42	218	2.4.2	インタフェース定型化サブシステム	外部システム連携	3.2.5.1.1 外部システム連携層に利用する技術	規約	E4200840-RG1-N1	外部システム連携層は、ESBを含めること。										
						規約	E4200850-RG2-N1	外部システム連携層の機能は、必要となる機能のうち、ESBのみで実現できないものについて、外部システム互換機能により補完すること。										
43	219	2.4.2	インタフェース定型化サブシステム	外部システム連携	3.2.5.1.2 外部システム連携層に持たせる機能	指針	E4200880-DG1-N1	外部システム連携層で吸収するギャップは、「(3)外部システム連携層が吸収するギャップ」に記載したものを対象とするように設計すること。										
						指針	E4200890-DG2-N1	ESBと外部システム互換機能の責務の分担は、「(4)ESBと外部システム互換機能の責務の分担」に従うように設計すること。										
44	142	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.1.1.3 多層層構造の詳細	規約	E4300050-RG1-N1	サブシステムの処理方式は、「表 3.1-3 処理方式による分類」に示す処理方式とすること。										
45	31	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.1.1.4 APレイヤ・コンポーネント定義	規約	E4300060-RG1-N1	コンポーネント間のアクセスは、「3.1.1.4.1 APレイヤの定義及びコンポーネントとの関係」に示すアクセスとすること。										
						規約	E4300070-RG2-N1	オンライン処理方式のコンポーネント間のアクセスは、「図 3.1-13から図 3.1-17」に示すアクセスパスとすること。										
						規約	E4300080-RG3-N1	オンライン処理方式のコンポーネントは、「表 3.1-10～表 3.1-14」に定義される単位で作成し、同表に定義されている責務を全うするようコンポーネントの機能を実現すること。										
						規約	E4300090-RG4-N1	バッチ処理方式のコンポーネント間のアクセスは、「図 3.1-18」に示すアクセスパスとすること。										
						規約	E4300100-RG5-N1	バッチ処理方式のコンポーネントは、「表 3.1-15」に定義される単位で作成し、同表に定義されている責務を全うするようコンポーネントの機能を実現すること。										
						規約	E4300110-RG6-N1	ディレードバッチ処理方式のコンポーネント間のアクセスは、「図 3.1-20」に示すアクセスパスとすること。										
46	45	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.1.2.1 オンライン処理方式	規約	E4300120-RG7-N1	ディレードバッチ処理方式のコンポーネントは、「表 3.1-16」に定義される単位で作成し、同表に定義されている責務を全うするようコンポーネントの機能を実現すること。										
						指針	E4300130-DG1-N1	オンライン処理方式の処理シーケンスは、「3.1.2.1.2 処理シーケンス」に示す例を参考に設計すること。										
						指針	E4300140-DG2-N1	オンライン処理方式のデータ保護方式は、「3.1.2.1.3 データの保護方式」に従って設計すること。										
						指針	E4300150-DG3-N1	オンライン処理方式のエラー処理方式は、「3.1.2.1.4 エラー処理方式」に従って設計すること。										
47	56	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.1.2.2 バッチ処理方式	指針	E4300160-DG4-N1	オンライン処理方式の業務継続方式は、「3.1.2.1.5 業務継続方式」に従って設計すること。										
						指針	E4300170-DG1-N1	バッチ処理方式の処理シーケンスは、「3.1.2.2.2 処理シーケンス」に示す例を参考に設計すること。										
						指針	E4300180-DG2-N1	バッチ処理方式のデータ保護方式は、「3.1.2.2.3 データの保護方式」に従って設計すること。										
						指針	E4300190-DG3-N1	バッチ処理方式のエラー処理方式は、「3.1.2.2.4 エラー処理方式」に従って設計すること。										
48	63	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.1.2.3 ディレードバッチ処理方式	指針	E4300200-DG4-N1	バッチ処理方式の業務継続方式は、「3.1.2.2.5 業務継続方式」に従って設計すること。										
						指針	E4300210-DG1-N1	ディレードバッチ処理方式の処理シーケンスは、「3.1.2.3.2 処理シーケンス」に示す例を参考に設計すること。										
						指針	E4300220-DG2-N1	ディレードバッチ処理方式のデータ保護方式は、「3.1.2.3.3 データの保護方式」に従って設計すること。										
						指針	E4300230-DG3-N1	ディレードバッチ処理方式のエラー処理方式は、「3.1.2.3.4 エラー処理方式」に従って設計すること。										
49	76	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.1.2.5 連携処理方式	指針	E4300240-DG4-N1	ディレードバッチ処理方式の業務継続方式は、「3.1.2.3.5 業務継続方式」に従って設計すること。										
						指針	E4300320-DG2-N1	連携処理方式のデータ保護方式は、「3.1.2.5.3 データの保護方式」に従って設計すること。										
						指針	E4300330-DG3-N1	連携処理方式の業務継続方式は、「3.1.2.5.4 業務継続方式」に従って設計すること。										
50	194	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.2.4.3.1 サブシステム間共通機能の提供形態	規約	E4300750-RG1-N1	サブシステム間共通機能の提供は、「(2)サブシステム間共通機能提供時の留意事項」に示す事項に留意し、提供すること。										
51	236	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.2.8.3 XMLとSQLのデータの扱いのルール	指針	E4301070-DG1-N1	XMLやSQLの書型形式を意識した業務処理、データ構造は、「図 3.2-50 書型形式を意識する層の範囲」に示す範囲に配置するよう設計すること。										
						指針	E4301080-DG2-N1	書型形式に依存した処理は、SQL形式を扱うコンポーネントが、将来、不変となる可能性を考慮して、XMLとSQLに分けてコンポーネント化されるよう設計すること。										
52	260	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.2.10.1 使用する文字コードのルール	推奨	E4301080-DG2-R1	XML形式及びSQL形式を扱うコンポーネントの切り替え、切り離しには、DI (Dependency Injection: 依存性注入) 等を用いて、依存関係を制御する仕組みを利用することを推奨する。										
						規約	E4301350-RG1-N1	内部システムで扱う文字コードは、「(1)使用する文字コード」に示す文字コードを採用すること。										
						規約	E4301360-RG2-N1	内部システムで使用する文字は、「(2)使用可能な文字」に示す文字とすること。										
						規約	E4301370-RG3-N1	内部システムで扱う文字コードの仕様への対応は、「(3)文字コードの仕様への対応」のとおりとすること。										
53	263	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.2.10.2 文字コードに関する制御のルール	規約	E4301380-RG4-N1	文字コードは、「(5)文字コードを扱う対象」に示す対象に適用すること。										
						時例	E4301380-RG4-S1	プログラムプロダクトのログや設定ファイル等、業務処理に直接影響しないものは、「(1)使用する文字コード」に示した文字コードをサポートしないことを許容する。										
						規約	E4301400-RG1-N1	内部システムにおける文字コードに関する制御は、以下に示す(1)～(4)のとおり扱うこと。										
						規約	E4301410-RG2-N1	特許庁システムの文書仕様における使用可能な文字の変更前後の対応は、(5)のとおりとすること。										
54	265	2.4.3	階層定型化サブシステム及びインタフェース定型化サブシステム		3.2.11 開発言語の選定ルール	指針	E4301430-DG1-N1	JIS X 0213:2004のグリフの扱いには、「(6)JIS X 0213:2004のグリフの扱いについて」に従い設計すること。										
						推奨	E4301430-DG1-R1	「表 3.2-76 システム構成要素毎の開発言語と選定指針」で推奨とするシステム構成要素は、同表の「選定指針」に従い開発言語を採用すること。										
						規約	E4301440-RG1-N1	「表 3.2-76 システム構成要素毎の開発言語と選定指針」で規約とするシステム構成要素は、同表に示す開発言語を使用すること。										

【●●システム構築】

特許庁アーキテクチャ標準仕様書 第1.0版 適合確認チェックリスト

項番	頁	大項目 (プロセス)	中項目(スコープ)		小項目	チェック項目		反映箇所	執筆者チェック欄				レビューでのチェック欄				備考	
			定型化分類	システム構成要素等		強制力	ID		チェック項目	反映箇所	反映内容	反映日	反映者	評価	確認日	確認者		コメント票No.
55	272	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.1 セキュリティ方式設計の ルール	指針	24301460-DG1-N1	セキュリティ方式設計は、『ボリシー実施手順（開発編別紙「設計基準」）」を遵守して設計すること。また、『表 3.2-84「ボリシー実施手順（開発編別紙「設計基準」）」とTobeアーキテクチャの関係』に示す事項は、同表の「Tobeアーキテクチャとの関係」に従い設計すること。 仮想化アプットのセキュリティは、『(3)仮想化アプットのセキュリティ』に従い設計すること。										
						指針	24301470-DG2-N1											
						規約	24301480-RG1-N1											
						特例	24301480-RG1-S1											
						指針	24301490-DG1-N1											
56	277	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.2 冗長化処理方式設計の ルール	指針	24301500-DG2-N1	クラスタ構成は、『(1)冗長化の方法（クラスタ構成）と選定ルール』のとおり選定すること。 より費用対効果の高い構成が実現できる場合は、選定ルールを逸脱することを許容する。 仮想化における冗長化処理は、『(3)仮想化に対する考慮』に示す事項を考慮して設計すること。 セッションにおける冗長化処理は、『(4)セッションにおける考慮』に示す事項を考慮して設計すること。 セッションレプリケーション等によるセッション保護機能は利用しないことを推奨する。										
						指針	24301510-DG1-N1											
						指針	24301520-DG2-N1											
						推奨	24301520-DG2-R1											
						特例	24301520-DG2-S1											
57	281	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.3 システム運転管理方式設計の ルール	指針	24301510-DG1-N1	システム運転管理方式は、『運用管理エージェント登録ガイドライン』を遵守して設計すること。 業務閉塞は、『(1)B.業務閉塞のルール』に示すルールに従い設計すること。 オンライン運転時間だけでなく、システム的な運転時間も特許庁全体で権力、統一することを推奨する。 やむを得ず、BMS内部から経過時間指定での実行が必要な場合は、エラー終了状態になることを想定して開局等の処理を設計すること。										
						指針	24301520-DG2-N1											
						指針	24301520-DG2-N1											
						推奨	24301520-DG2-R1											
						特例	24301520-DG2-S1											
58	283	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.4 システム監視方式設計の ルール	指針	24301560-DG1-N1	ハードウェアやソフトウェア等の監視は、『(1)ハードウェアやソフトウェア等の監視』に従って設計すること。										
59	284	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.5 バックアップ方式設計の ルール	指針	24301570-DG1-N1	バックアップ方式は、『バックアップ設計指針』に遵守して設計すること。										
60	285	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.6 リリース管理方式設計の ルール	指針	24301580-DG2-N1	BMSのバックアップ方式は、オフラインバックアップにする場合、ビジネスプロセスインスタンスの停止及び再開に時間がかかることを考慮して設計すること。										
						指針	24301590-DG1-N1		リリース管理方式は、『特許庁システム リリースポリシー』及び『パッチ適用方針』に遵守して設計すること。									
61	286	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.7.1 ログ出力の目的	指針	24301600-DG1-N1	OS、プログラムプロダクト、アプリケーションが出力するログを使用して『表 3.2-89 ログ出力の目的』に示す目的に沿ったログが出力されるよう設計すること。また、サブシステムの要件として、『表 3.2-89 ログ出力の目的』に示すログ出力の目的以外がある場合には、サブシステム側に設計すること。										
62	287	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.7.2 ログ出力情報	指針	24301610-DG1-N1	OS、プログラムプロダクト、アプリケーションがログ出力する情報は、『表 3.2-90 ログ出力情報』の「ログに出力するべき情報」を参考に設計すること。										
63	288	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.7.3 ログレベル	指針	24301620-DG1-N1	出力するログのログレベルは、『(2)A.ログレベルに関わる指針』に従って設計すること。 出力するログのログレベルは、『(2)B.ログレベルに関わる規約』のとおりとすること。										
						規約	24301630-RG1-N1											
64	289	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.7.4 運用監視システムとの 連携	指針	24301640-DG1-N1	OS、プログラムプロダクト、アプリケーションが出力するログと運用監視システムとの連携は、『(2)運用監視システムとの連携ルール』に従って設計すること。										
65	290	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.7.5 ログ出力項目とフォー マット	規約	24301650-RG1-N1	ログ出力項目及びフォーマットは、『(1)ログ出力項目及びフォーマットのルール』のとおりとすること。 個別ヘッダ及びメッセージは、『(2)個別ヘッダ及びメッセージのルール』に従って設計すること。 ログ出力媒体は、『(1)ログ出力媒体』に示す媒体とすること。 ログ出力単位は、『(2)ログ出力単位』に示す単位とすること。 「(2)ログ出力単位」に示す特例を許容する。 ログローテーションは、『(3)ログローテーション』に従い、設計すること。 バックアップは、『(4)バックアップ』に従って設計すること。 ログの参照用保持は、『(5)参照用保持』に従って設計すること。 ログの参照用権限は、『(6)ログ参照権限』に従って設計すること。										
						指針	24301660-DG1-N1											
						指針	24301670-RG1-N1											
						指針	24301680-RG2-N1											
						指針	24301690-DG1-N1											
66	292	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム		3.2.13.7.6 ログ運用	規約	24301670-RG1-N1	ログ出力媒体は、『(1)ログ出力媒体』に示す媒体とすること。 ログ出力単位は、『(2)ログ出力単位』に示す単位とすること。 「(2)ログ出力単位」に示す特例を許容する。 ログローテーションは、『(3)ログローテーション』に従い、設計すること。 バックアップは、『(4)バックアップ』に従って設計すること。 ログの参照用保持は、『(5)参照用保持』に従って設計すること。 ログの参照用権限は、『(6)ログ参照権限』に従って設計すること。										
						指針	24301680-RG2-N1											
						指針	24301690-DG1-N1											
						指針	24301700-DG2-N1											
						指針	24301710-DG3-N1											
67	228	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム	ビジネスルール管理	3.2.7.1.1 ビジネスルールをアプリ ケーションから切り離すための仕組 みとして利用する技術	規約	24300940-RG1-N1	BMSによるビジネスルールの記述は、『(3)BMSによるビジネスルールの記述ルール』に示すとおり記述すること。 ビジネスルールをアプリケーションから切り離すための仕組みの適用方法は、『(4)コンポーネント機能による依存関係の制御』に従って設計すること。 BMSの適用は、『(2)BMSを用いたビジネスルールの記述対象』に従って設計すること。 ビジネスルールは、『(2)BMSを用いたビジネスルールの記述対象』に示す特例を許容する。										
						指針	24300950-DG1-N1											
						指針	24300960-DG2-N1											
						指針	24300960-DG2-N1											
						特例	24300960-DG2-S1											
68	251	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム	ビジネスルール管理	3.2.9.6 APサーバのルール	指針	24301240-DG1-N1	APサーバは、『(1)APサーバの機能要件』に示す機能要件を満たしているプログラムプロダクトを採用すること。 APサーバは、『(3)APサーバの適用範囲』に示す範囲で適用すること。										
						指針	24301250-DG2-N1											
69	234	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム	外部システム連携	3.2.8.2.1 テーブル名の命名ルール	指針	24301030-DG1-N1	テーブル名は、『(1)テーブル名の構成』に示す用語の組み合わせに従い、命名すること。 テーブル名は、『(2)テーブル名の重複と同音異義・異音同義語の禁止事項』に示す禁止事項に抵触しないよう、命名すること。										
						指針	24301040-DG2-N1											
70	235	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム	外部システム連携	3.2.8.2.2 カラム名の命名ルール	指針	24301050-DG1-N1	カラム名は、『(1)カラム名の構成』に示す用語の組み合わせに従い、命名すること。 カラム名は、『(2)カラム名の重複と同音異義・異音同義語の禁止事項』に示す禁止事項に抵触しないよう、命名すること。										
						指針	24301060-DG2-N1											
71	253	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム	外部システム連携	3.2.9.7 FTPサーバのルール	指針	24301260-DG1-N1	FTPサーバは、『(1)FTPサーバの機能要件』に示す機能要件を満たしているプログラムプロダクトを採用すること。 FTPサーバは、『(3)FTPサーバの適用範囲』に示す範囲で適用すること。										
						指針	24301270-DG2-N1											
72	256	2.4.3	階層定数化サブシステム及び インタフェース定数化サブシ ステム	外部システム連携	3.2.9.10 DBMSのルール	指針	24301320-DG1-N1	DBMSは、『(1)DBMSの機能要件』に示す機能要件を満たしているプログラムプロダクトを採用すること。 DBMSは、『(3)DBMSの適用範囲』に示す範囲で適用すること。										
						指針	24301330-DG2-N1											

項番	頁	大項目 (プロセス)	中項目(スコープ)		小項目	チェック項目		反映箇所	執筆者チェック欄		レビューでのチェック欄		備考			
			定型化分類	システム構成要素等		強制力	ID		反映内容	反映日	反映者	評価		確認日	確認者	コメント票No.
73	68	2.4.3	階層定型化サブシステム		3.1.2.4 帳票出力方式	指針	24300250-DG1-N1	帳票出力方式の処理シーケンスは、「3.1.2.4.2 処理シーケンス」に示す例を参考に設計すること。								
						指針	24300260-DG2-N1	帳票出力方式の業務継続方式は、「3.1.2.4.3 業務継続方式」に従って設計すること。								
74	111	2.4.3	階層定型化サブシステム		3.2.2.3 ブラウザからの要求を受けるアプリケーションの粒度	指針	24300430-DG1-N1	ブラウザからの要求を受けるアプリケーション（プレゼンテーションロジック及び業務AP（システム）のまゝり）の粒度は、「(1)ブラウザからの要求を受けるアプリケーションの粒度のルール」に従って設計すること。								
75	123	2.4.3	階層定型化サブシステム		3.2.2.4.4 画面の動作に関するルール	指針	24300500-DG1-N1	画面動作は、「表 3.2-14 画面の動作に関するルール」に従って設計すること。								
76	132	2.4.3	階層定型化サブシステム		3.2.3.1.2 BPMSの適用範囲に関する設計指針	指針	24300570-DG1-N1	BPMSと業務アプリケーションの組み合わせは、「表 3.2-16 BPMSと業務アプリケーションとの組合せの代表的なケース」に示すケースに該当する場合、以下の(1)～(3)に示すBPMSと業務アプリケーションの設計指針に従い、ビジネスプロセスを実現できるよう設計を行うこと。								
77	145	2.4.3	階層定型化サブシステム		3.2.3.1.3 ビジネスプロセスのステータス管理の設計指針	指針	24300580-DG1-N1	ビジネスプロセスのプロセステータスは、BPMSではキー情報だけを保持するように設計すること。								
						指針	24300590-DG2-N1	ビジネスプロセスのステータスの管理は、「(1)ステータス管理のルール」に従って、設計すること。								
						特例	24300590-DG2-S1	ヒューマンタスクにおけるステータスの一覧取得時の性能面に懸念がある場合には、ステータスの一部をデータベースにも保持して二重管理することを許容する。								
						指針	24300600-DG3-N1	待機要因の管理は、「(2)待機要因の管理ルール」に従って、設計すること。								
78	147	2.4.3	階層定型化サブシステム		3.2.3.1.4 ビジネスプロセスを介したデータの授受ルール	指針	24300610-DG1-N1	ビジネスプロセスを介して授受するデータは、「(1)ビジネスプロセスを介して授受するデータのルール」に従って設計すること。また、「(2)ビジネスプロセスを介して授受するデータの設計時の留意事項」に示す留意事項を遵守すること。								
79	149	2.4.3	階層定型化サブシステム		3.2.3.2.1 BPMS要素の使用ルール	規約	24300620-RG1-N1	ビジネスプロセスの記載に使用するBPMS要素は、「表 3.2-19」～「表 3.2-28」に示す使用可否、及び制約事項のとり使用すること。								
80	163	2.4.3	階層定型化サブシステム		3.2.3.2.2 命名ルール	指針	24300630-DG1-N1	BPMS要素のラベル名は、「(1)ラベル命名ルール」、「(2)用語に関するルール」に従って付与すること。								
81	167	2.4.3	階層定型化サブシステム		3.2.3.2.3 ビジネスプロセス記載ルール	指針	24300640-DG1-N1	ビジネスプロセスは、「(1)ビジネスプロセスの記載単位ルール」に示す単位で設計すること。								
						規約	24300650-RG1-N1	ビジネスプロセスは、「(2)同じ意味であるが表記上異なるビジネスプロセスの記載統一ルール」のとり記載すること。								
						指針	24300660-DG2-N1	ビジネスプロセスのアクティビティは、「(3)アクティビティの単位のルール」に示す単位で設計すること。								
						規約	24300670-RG2-N1	BPMS要素は、「(4)レイアウトルール」の「A.時系列にあった配置ルール」、「C. BPMS要素の重なり防止のルール」、「D. プールの配置ルール」、「E. イベント・サブプロセスの配置ルール」のとり配置すること。								
						推奨	24300680-DG3-N1	レーン配置順は、「(4)レイアウトルール B.レーンの配置ルール」に従い、登場人物の分類毎に可読性を考慮した配置順序を決定し、各ビジネスプロセスで統一した配置順にすること。 役職のレーンを配置する場合、「下層の役職から上層の役職の順」に上から下へレーンを配置すること。								
82	177	2.4.3	階層定型化サブシステム		3.2.3.2.4 ビジネスプロセスパターン事例	規約	24300690-RG1-N1	ビジネスプロセスは、ビジネスプロセス設計時に「表 3.2-31 ビジネスプロセスパターン事例一覧」に示す適用シーンに合致した場合、当該パターン事例を適用すること。								
83	181	2.4.3	階層定型化サブシステム		3.2.3.3.1 BPMS補完機能で実現する機能	規約	24300700-RG1-N1	BPMS補完機能で実現する機能は、「(1)BPMS補完機能で実現する機能のルール」に示す機能とすること。								
84	184	2.4.3	階層定型化サブシステム		3.2.3.3.3 配置位置	規約	24300710-RG1-N1	BPMS補完機能の配置は、「(1)BPMS補完機能の配置位置のルール」に示すとおり配置すること。								
85	187	2.4.3	階層定型化サブシステム		3.2.4.1 サービスの粒度	指針	24300720-DG1-N1	サービスの粒度は、「(1)サービスの粒度のルール」に従って設計すること。								
86	212	2.4.3	階層定型化サブシステム		3.2.4.4.3 特許庁開発フレームワークのサブシステム間の共有	指針	24300800-DG1-N1	特許庁開発フレームワークをサブシステム間で共有する場合、サブシステム毎に独立してバージョンアップができるよう、資料管理の仕組みを整備すること。								
87	214	2.4.3	階層定型化サブシステム		3.2.4.5.2 個別データベースへのアクセスルール	指針	24300830-DG1-N1	情報活用系サブシステムへのデータ提供は、「(3)情報活用系サブシステムへのデータ提供のルール」に準拠して設計すること。								
88	216	2.4.3	階層定型化サブシステム		3.2.6.1 共有データベースの構成及びアクセスルール	規約	24300920-RG1-N1	共有データベースに一時的に保存される受け渡しデータは、受領するサブシステムが使用後に削除すること。								
89	230	2.4.3	階層定型化サブシステム		3.2.7.1.2 BRMSの呼び出し方式	規約	24300970-RG1-N1	BRMSの呼び出し方式は、「(2)BRMSの呼び出し方式の選択ルール」のとり選択すること。								
						特例	24300970-RG1-S1	性能面のリスクが高いと判断した場合は、「ライブラリ配属＋リポトリ方式」を許容する。								
						規約	24300980-RG2-N1	BRMSを利用する階層定型化サブシステムの設計に関与するステークホルダ（設計・開発ベンダ等）の開発対象は、「(3)BRMSの開発対象」に示すとおりとすること。								
90	244	2.4.3	階層定型化サブシステム		3.2.9.1 BPMSのルール	指針	24301140-DG1-N1	BPMSは、「(1)BPMSの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。								
						推奨	24301140-DG1-R1	ビジネスプロセスの進行状況と実行履歴を抽出及び加工して、リアルタイムでグラフや表形式等で画面表示するための実現手段については、BPMSが保有するBAM（Business Activity Monitoring）を推奨する。								
						指針	24301150-DG2-N1	BPMSは、「(3)BRMSの適用範囲」に示す範囲で適用するように設計すること。								
91	249	2.4.3	階層定型化サブシステム		3.2.9.4 Webブラウザのルール	指針	24301200-DG1-N1	Webブラウザは、「(1)Webブラウザの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。								
						指針	24301210-DG2-N1	Webブラウザは、「(3)Webブラウザの適用範囲」に示す範囲で適用すること。								
92	250	2.4.3	階層定型化サブシステム		3.2.9.5 Webサーバのルール	指針	24301220-DG1-N1	Webサーバは、「(1)Webサーバの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。								
						指針	24301230-DG2-N1	Webサーバは、「(3)Webサーバの適用範囲」に示す範囲で適用すること。								
93	254	2.4.3	階層定型化サブシステム		3.2.9.8 ジョブ管理エージェントのルール	指針	24301280-DG1-N1	ジョブ管理エージェントは、「(1)ジョブ管理エージェントの機能要件」に示す機能要件を満たすジョブ管理マネージャのエージェントとして動作すること。								
						指針	24301290-DG2-N1	ジョブ管理エージェントは、「(3)ジョブ管理エージェントの適用範囲」に示す範囲で適用すること。								
94	255	2.4.3	階層定型化サブシステム		3.2.9.9 帳票設計・印刷ソフトウェアのルール	指針	24301300-DG1-N1	帳票設計・印刷ソフトウェアは、「(1)帳票設計・印刷ソフトウェアの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。								
						指針	24301310-DG2-N1	帳票設計・印刷ソフトウェアは、「(3)帳票設計・印刷ソフトウェアの適用範囲」に示す範囲で適用すること。								
95	258	2.4.3	階層定型化サブシステム		3.2.9.11 認証プラグイン/個人認証ソフトウェアのルール	指針	24301340-DG1-N1	認証プラグイン/個人認証ソフトウェアは、「(1)認証プラグイン/個人認証ソフトウェアの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。								
						指針	24301350-DG2-N1	認証プラグイン/個人認証ソフトウェアは、「(3)認証プラグイン/個人認証ソフトウェアの適用範囲」に示す範囲で適用すること。								
96	246	2.4.3	インタフェース定型化サブシステム	ビジネスルール管理システム	3.2.9.2 BRMSのルール	指針	24301160-DG1-N1	BRMSは、「(1)BRMSの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。								
						指針	24301170-DG2-N1	BRMSは、「(3)BRMSの適用範囲」に示す範囲で適用するように設計すること。								
						特例	24301170-DG2-S1	BRMSは、「(3)BRMSの適用範囲」に示す特例を許容する。								
97	247	2.4.3	インタフェース定型化サブシステム	外部システム連携	3.2.9.3 ESBのルール	指針	24301180-DG1-N1	ESBは、「(1)ESBの機能要件」に示す機能要件を満たしているプログラムプロダクトを採用すること。なお、「A.TRMの定義から必要とされる機能要件（基本事項）」に示す機能要件は、外部システム互換機能と組み合わせず実現してもよいものとする。								
						指針	24301190-DG2-N1	ESBは、「(3)ESBの適用範囲」に示す範囲で適用すること。								