

DX 実践手引書

IT システム構築編

レガシーシステム刷新ハンドブック



独立行政法人 情報処理推進機構
社会基盤センター

このドキュメントは下記の URL からダウンロード可能です。

https://www.ipa.go.jp/ikc/our_activities/dx.html#section7

目 次

第1章 はじめに.....	4
1.1. 背景	4
1.2. DX 実践手引書 IT システム構築編 レガシーシステム刷新ハンドブックの 想定読者について	7
第2章 現行 IT システムの全体把握	8
2.1. 現行 IT システムの全体把握の目的.....	8
2.2. IT システムの粒度	9
2.3. 「全体システム構成図」を構成する要素、記法.....	10
2.4. 「全体システム構成図」作成手順の概要	15
2.5. 「全体システム構成図」作成のための各 STEP の内容.....	16
第3章 現行 IT システムの仕様復元	34
3.1. 現行 IT システムの仕様復元とその必要性	34
3.2. 仕様復元の各工程での段階的情報遡及.....	36
3.3. 現行 IT システムの仕様復元の概要.....	38
3.4. 概要設計の遡及の考え方	39
3.5. 現行 IT システムの仕様復元(バッチ型).....	41
3.6. 現行 IT システムの仕様復元(オンライン型).....	43
3.7. 現行 IT システムの仕様復元(WEB 型)	44
3.8. 現行 IT システムの仕様復元(ゲートウェイ型)	46

第1章 はじめに

1.1. 背景

あらゆる産業において、新たなデジタル技術を使ってこれまでにないビジネスモデルを展開する新規参入者が登場し、ゲームチェンジが起ころうとしている。こうした中で、各企業は、競争力維持・強化のために、デジタルトランスフォーメーション（DX：Digital Transformation）をスピーディに進めていくことが求められている。DXとは「企業がビジネス環境の激しい変化に対応し、データとデジタル技術を活用して、顧客や社会のニーズを基に、製品やサービス、ビジネスモデルを変革するとともに、業務そのものや、組織、プロセス、企業文化・風土を変革し、競争上の優位性を確立すること」であり、企業がグローバル競争で勝ち残るために必須であるといわれている。

DXを実現するには、ビジネスの戦略を立てる経営（CEO）とビジネスを支えるITシステム（CIO）が密に対話して進めることが重要である。その論点は、以下の3点である。（1）デジタル技術がビジネスにどのようなインパクトや効果を与えるか。（2）デジタル技術を用いたビジネスを支えるように、現行のITシステムをどのように変革していくか。（3）変革したITシステムが継続してビジネスの変化に対応し続けるようにどのように維持し続けていくか。（1）はそれぞれの企業のビジネス戦略によって個別に議論する必要があるが、（2）（3）に関しては、企業横断的に解決すべき共通の課題が存在する。このため、本DX実践手引書ITシステム構築編 レガシーシステム刷新ハンドブックではこの共通課題を解決することを目指す。

DXを進めて行く上での原動力となるのは、デジタル技術の活用である。デジタル技術の技術革新は、ビジネスモデルの変革を含めたビジネス革新を実現可能にするいくつかの技術を生み出している。それらを活用しビジネス革新を現実のものとするためには、企業のITシステムが次の3つの技術要件を満たしておく必要がある。（1）データ活用性：活用したいデータが必要なタイミング鮮度、精度で容易にとりだせるようになっていること。（2）マーケット変化に対する柔軟性：ビジネス要件や環境変化などの変更に対応できるようになっていること。

（3）マーケット変化に追従可能なスピーディな対応：ITシステムが素早く高い頻度で機能を提供できるようになっていること、である。

しかしながら、多くの企業が抱えているITシステムには、前述の要件を十分に満たしていないものが多く存在し、それがDXの推進を阻害している。経済産業省が2018年に発表した「DXレポート」¹では、この状況を「2025年の崖」と称して、「日本企業に共通の課題として、事業部門ごとに構築、複雑化・ブラックボックス化された企業システムがDXを阻んでおり、2025

¹ 「DX デジタルトランスフォーメーション レポート ～ITシステム「2025年の崖」の克服とDXの本格的な展開～」経済産業省 デジタルトランスフォーメーションに向けた研究会

年以降、これが最大 12 兆円/年（現在の約 3 倍）の経済損失を生じうる」と警鐘をならしている。また、同じく経済産業省が 2019 年に発表した DX 推進指標を、同年に 287 社の企業が自己評価したものを IPA が分析した結果²によると、DX 指標の上位 5%企業は自社の IT 資産の評価・分析は行ったうえで、評価・分析の結果を踏まえて不要資産の廃棄などの具体的な施策につなげているが、残りの 95%の企業は DX にまったく取り組んでいないレベルにあるか、DX の散発的な実施に留まっているに過ぎない段階である傾向が明らかになった。

評価・分析がそれだけで終わらないようにするために、先行企業は上記 DX の要件を満たす自社の IT システムのあるべき姿を描き、あるべき姿に向けて継続的に IT システムの変革をすすめている。その他の多くの企業では、次の 2 つの大きな課題が存在している。すなわち、（１）自社のビジネスに沿った IT システムのあるべき姿を描くことが難しく、新たなシステム開発技術の適用方法も明確になっていないこと。そして、（２）既存の IT システムをリスクとコストをかけて変革することが難しいことである。

このうち、特に（２）の課題が解決できない理由として、次に示す 2 つの要因が特に大きな障害となっていると考えられる。

1 つめは既存 IT システムが大規模化し、マネジメントが困難になるレベルになっているものが存在していることである。多くの企業では、開発時から繰り返し機能拡張をすることで超大規模化した IT システムが存在している。一般にマネジメントは、その IT システムの規模に対し指数関数的に困難になると言われている。このレベルのマネジメントが可能なプロジェクトマネージャは数えるほどしか存在せず、現実問題として、大手 IT ベンダーですら超大規模な IT システムの開発に失敗した例が多くある。かかる状況下で、多くの企業では自社の大規模化した IT システムの変革難易度の高さ故、ごまかしながら使っているのが現状である。このような超大規模システムをコントロール可能な状態に分割し、全体をマネジメントする P2M（Program & Project Management）レベルの IT システム開発向けの考え方は確立していない。

2 つめは企業の IT システムのブラックボックス化が進んでいることである。過去に開発した IT システムの中には、設計書をメンテナンスしないままに変更が加えられてしまっているものや、当初想定したアーキテクチャのポリシーに反した変更（コピー＆ペーストなど）が行われてしまっているなどの理由により、その構造や仕様が誰にも分からなくなっているものが存在している。プログラム解析技術だけでは「処理」の仕様を復元することはできるが、なぜその処理が必要なのかという「業務」の仕様を復元することはできない。事実、現行仕様が不明となっている IT システムの部分再構築のプロジェクトは問題化しやすく、IT ベンダーの問題プロジェクトの多くがこのケースであるといわれている。いわゆる「現行機能保証」を求められるとプロジェクトの難易度が高くなるということが、失敗事例として語られている。このような、IT システムは、これを DX の要件を満たすように変革したいと考えても、現行システム要件のうち必要な要件を明確にしないと、正しい構築ができない。

² 「DX 推進指標 自己診断結果 分析レポート」 独立行政法人情報処理推進機構

上記の2つの要因のいずれか、もしくは両方のため、多くの企業ではITシステムの全体像を把握されていないのではないかと懸念される。このことは、一方では保守や維持のコストを肥大化させ、かけているコストの妥当性を経営に説明できず、企業の経営層に企業のITシステム部門への不信感を抱かせる原因になっている。そして、他方では、ITシステム革新のコストをいたずらに大きく、そしてその効果を不十分なものとさせ、経営が求めるニーズにIT部門が十分に応えられないことにもつながっている。こうした事情の中で、経営層はITシステムをブラックボックスで巨大なコストがかかっていると見るようになり、ITシステム部門から見ると経営層はITのことをわかっていないと感じ、両者が分かりあえず不幸な状況になっている企業が散見される。そして、このことが適切なITシステムの投資判断を難しくし、全体像の把握できないITシステムを温存させることで、対立がいつまでたっても改善されない悪循環構造ができてしまう。このような状況では、ビジネス環境の激しい変化に追随していくことは困難である。

本DX実践手引書 ITシステム構築編 レガシーシステム刷新ハンドブックは、上記の技術課題に対して、ITベンダー各社の知恵を結集し次の観点でITシステム変革を行うための基本的な考え方を示すものとして作成される（（1）からスタートし、随時提供していく）。

- （1）現状の超大規模システムを分割し、コントロール可能な粒度に整理する（今回公開分）
現行の超大規模システムをマネジメントが容易な規模（機能システム）に分割整理
新しい開発方式を踏まえた品質マネジメント方式（今後作成）
- （2）現行ITシステムの概要仕様を復元する（今回公開分）
ブラックボックス化されているITシステムの概要仕様を復元するステップを示す
- （3）DXの要件を満たすITシステムのあるべき姿を描く（今後作成）
デジタル技術を活用してビジネスを変革するためにDXの要件を満たすITシステムのあるべき姿を描く
- （4）新しい開発方法でITシステムの変革を進める（今後作成）
ビジネス要件できめられた目指す姿と優先度に従って、アジリティ、スピード、品質を向上させる考え方をを用いてITシステムを変革する

本ハンドブックを活用することで、ITシステム部門からテクノロジー駆動で経営に提言するための材料になると幸いである。

1.2. DX 実践手引書 IT システム構築編 レガシーシステム刷新ハンドブ

ックの想定読者について

本ハンドブックは、DX、すなわちデジタル技術を活用してビジネスの変革を行いたいと考えている組織自身が IT システムの変革およびその DX の要件の維持を行う際において活用されること想定して作成されている。この組織とは、事業の遂行や、IT システムの維持、管理、開発を担う担当者はもちろんのこと、次節で述べるが、ハンドブックの概要については、DX を現実のものとするための戦略をたて、上記活動の報告を受ける立場から、組織の経営者も含まれる。

かかる所以は、DX は、事業、業務、IT システムの3つの立場の人々が三位一体となってデジタル技術の活用を経営層に提言し、これを実現し、さらに継続的にあるべき姿に向けて変革を続けていくことがビジネスの成功につながると考えられるからである。これらを実現することがデジタルエンタープライズ（DX が行えている企業）たる必須条件である。

また、この組織をサポートし、アドバイスを行う立場である IT コンサルタントや IT ベンダーも、本ハンドブックについて理解しておくべきである。

2.2. IT システムの粒度

本章以降、様々な IT システムの規模について記載するため、ここで下表のように用語の定義をしておく。以下に示す規模は目安であり、実際には業務を行う組織との関係で再定義する。

表 2-1 IT システムの規模

規模レベル	本書での呼称	ITシステムのイメージ	規模の目安
レベル I	社会システム	業態全体を包含する IT システム 規模レベル。 国民へのサービスなど社会全体で使われているような IT システムが該当する。	
レベル II	全体システム	1つの企業全体が保有する IT システムすべてを包含する規模レベル。P2Mレベルのマネジメントが求められる。	10万FP以上
レベル III	機能システム	1つの企業の一事業部門が保有する IT システムすべてを包含する規模レベル。(*)	1～2万FP
レベル IV	サブシステム	実質的に人が管理・認識し扱える、再構築などのプロジェクト単位として扱いやすい IT システム 規模。オンライン／バッチ／Web／ゲートウェイなどの特性を持つ IT システムに分類される。	1,000～3,000FP

(*) 複数のサブシステムから構成される部門システム。

総合テストを同時に実施する単位、サブシステム構成図を作成する単位に相当。

FP (ファンクションポイント)

2.3. 「全体システム構成図」を構成する要素、記法

「全体システム構成図」における要素とその記法について、仮定の船舶会社を例として、以下、説明していきたい。この事例における「全体システム構成図」(例)を示すと下記の通りである。

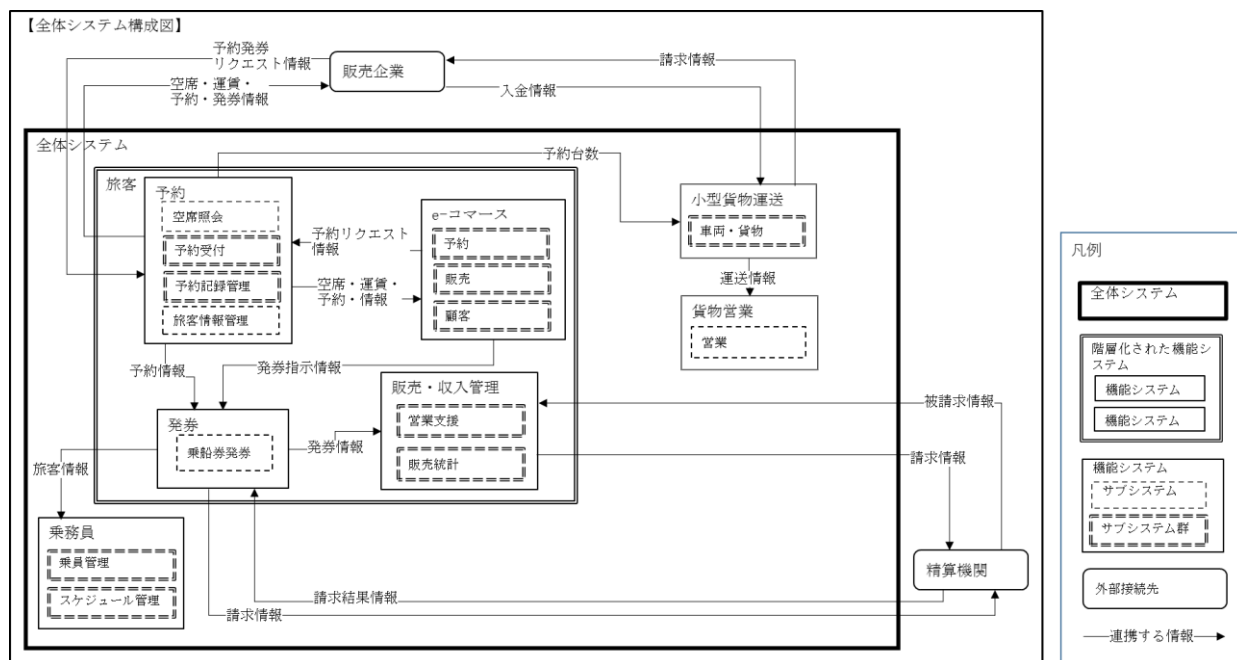


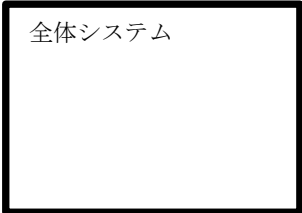
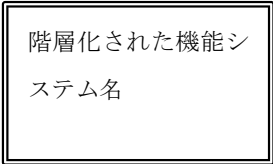
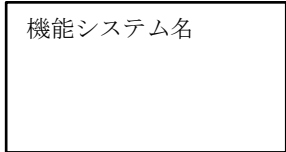


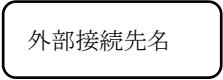
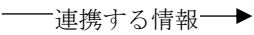
図 2-2 「全体システム構成図」(例)

「全体システム構成図」は、上記記載例のように、基本的には1枚で作成し、全体を俯瞰できるようにする。

ただし、「全体システム構成図」の項目が多くなって、図が複雑化し、可読性が損なわれる場合には、階層化された機能システムをそれぞれ別紙に記載してもよい。

この事例を使いながら、「全体システム構成図」を作成するプロセスを以降で解説する。

本ハンドブックでは、「全体システム構成図」内の要素について、以下の記法を用いる。

No	要素	表記	説明
1	全体システム		<ul style="list-style-type: none"> 全体システムを表す。 評価の対象とする全 IT システムをこの中に含む。
2	階層化された機能システム		<ul style="list-style-type: none"> 階層化された機能システムを表し、大きな機能システムを階層化して表現し中に機能システムを含む。 階層化された機能システム名を枠内に記載する。
3	機能システム		<ul style="list-style-type: none"> 機能システムを表し、その中に、サブシステム、または、サブシステム群を含む。 機能システム名を枠内に記載する。
4	サブシステム		<ul style="list-style-type: none"> サブシステムを表す。 サブシステム名を枠内に記載する。
5	サブシステム群		<ul style="list-style-type: none"> 複数のサブシステムを含んでいることを表す。 サブシステム群名を枠内に記載する。
6	外部接続先		<ul style="list-style-type: none"> 外部接続先を表す。機能システムと連携している外部の接続先、利用している外部のサービスを表す。 外部接続先名を枠内に記載する。
7	機能システム間、機能システムと外部接続先間の連携		<ul style="list-style-type: none"> 機能システム間、機能システムと外部接続先間の連携を表す。 連携概要が分かるように、連携する情報の流れと、主要な連携内容を記載する。

各要素の記載例を、以下に示す。

1. 全体システム

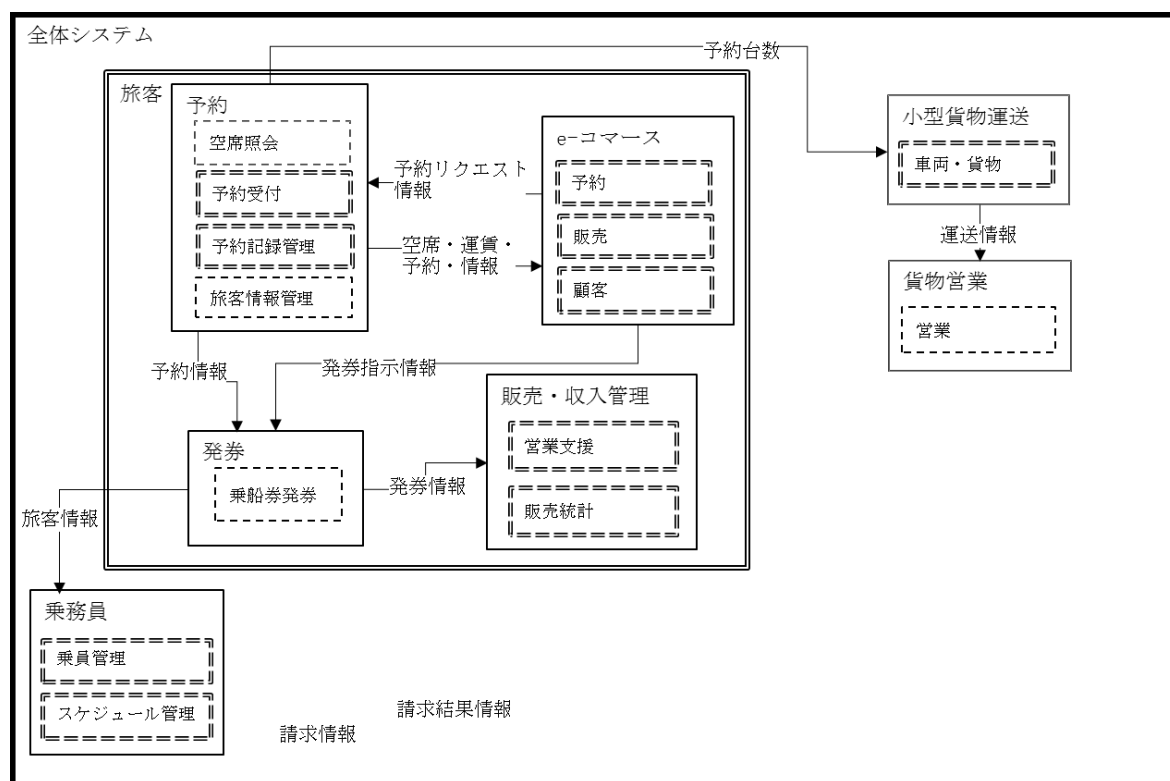


図 2-3 「全体システム構成図」の記載例

2. 階層化された機能システム

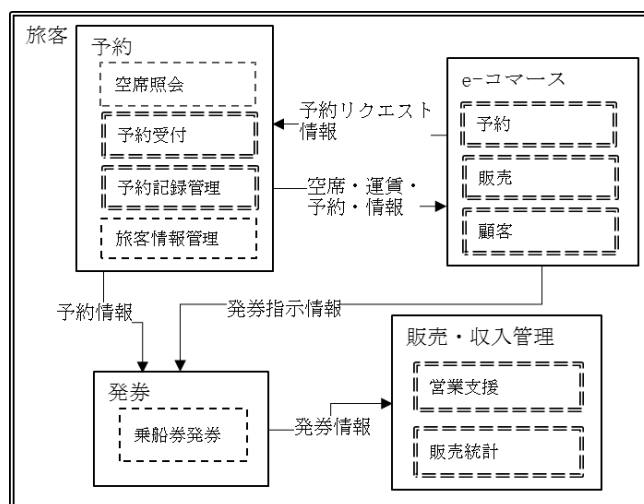


図 2-4 「階層化された機能システム」の記載例

3.機能システム

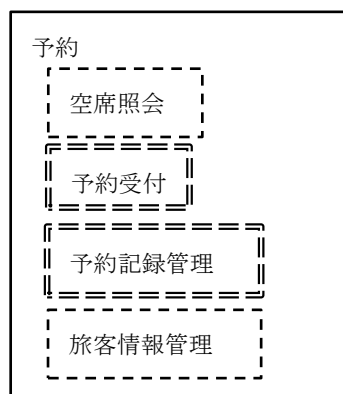


図 2-5 「機能システム」の記載例

4.サブシステム

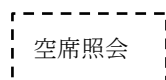


図 2-6 「サブシステム」の記載例

5.サブシステム群

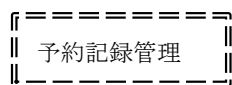


図 2-7 「サブシステム群」の記載例

6.外部接続先

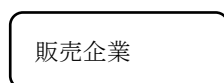


図 2-8 「外部接続先」の記載例

7. 機能システム間、機能システムと外部接続先間の連携

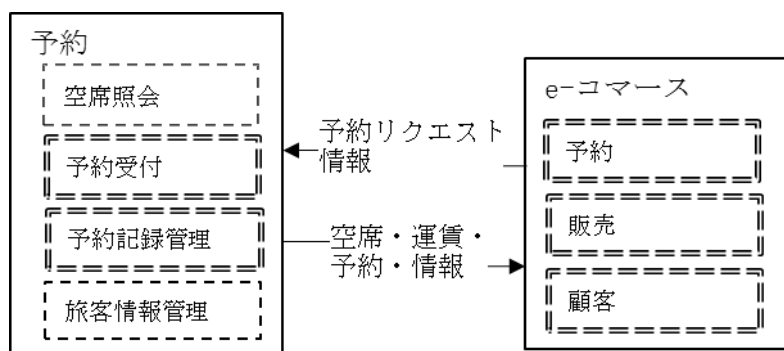


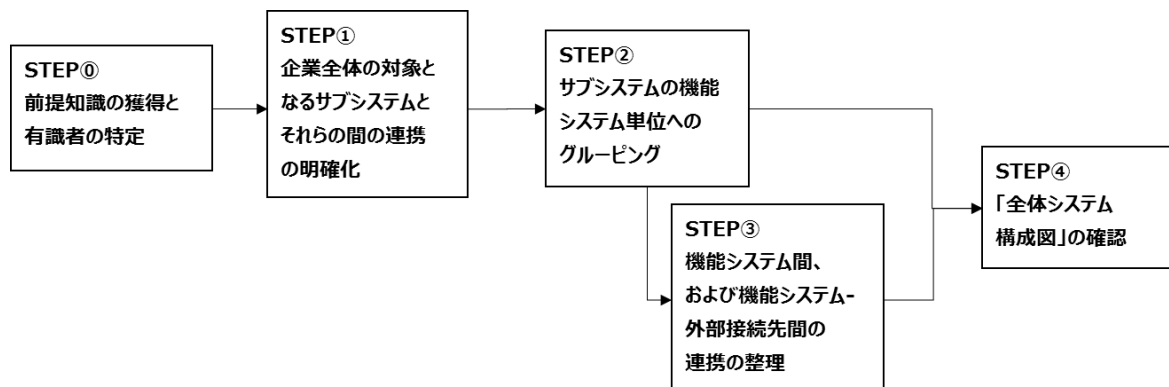
図 2-9 「機能システム間の連携」の記載例

連携に関して、以下で補足する。

No	属性	説明
7-1	矢印	<ul style="list-style-type: none"> 機能システム間、機能システムと外部接続先との間の連携において、連携元から連携先へ受け渡す情報の流れを表し、連携元、連携先それぞれの枠をつなぐ。同じ方向に受け渡す情報について、1本程度に集約して表す。 同じ連携元、連携先から、反対の情報の流れがある場合は、連携先から連携元へ向かう矢印を追加する。
7-2	連携する情報	<ul style="list-style-type: none"> 連携元から連携先へ受け渡す情報の内容を矢印の線上に記載する。
7-3	連携元	<ul style="list-style-type: none"> 機能システム間、機能システムと外部接続先との間の連携について、情報の起点となる送信元を表す。 連携元として記載する対象は、機能システム、外部接続先。
7-4	連携先	<ul style="list-style-type: none"> 機能システム間、機能システムと外部接続先との間の連携について、情報の終点となる送信先を表す。 連携先として記載する対象は、機能システム、外部接続先。

2. 4. 「全体システム構成図」作成手順の概要

以下に示す STEP で、「全体システム構成図」を作成する。



STEP①は、「全体システム構成図」作成の工程に着手する前の準備を行う。

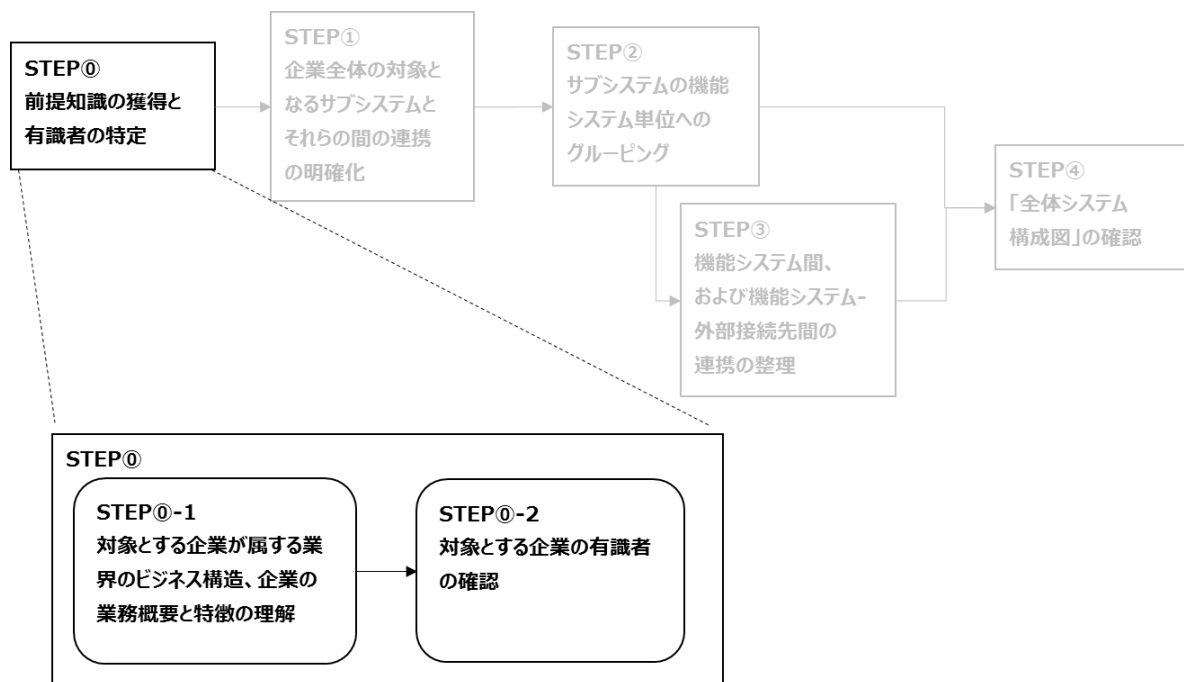
STEP①～③で、「全体システム構成図」に含める各要素、及び各要素間の連携についての情報収集と整理を行い、段階的に「全体システム構成図」を作成する。

STEP④で、作成した「全体システム構成図」の検証を行い、完成させる。

2.5. 「全体システム構成図」作成のための各 STEP の内容

各 STEP で実施する内容、作成する対象について、以下に記載する。

□STEP① 前提知識の獲得と有識者の特定



STEP①の目的：「全体システム構成図」の作成に向けた、前提知識の獲得と有識者の特定。

入力情報：業務概要を事前に理解する資料(例えば、企業内に通常整備されている、用語集や基本業務のマニュアル)、組織の構造や業務の分担を理解する資料(例えば、組織図、業務分掌)

手順：

STEP①-1 対象とする企業が属する業界のビジネス構造、企業の業務概要と特徴の理解

タスク内容：前提知識を得るため、その企業が属する業界についてのビジネスの構造、企業の業務概要と特徴を理解する。

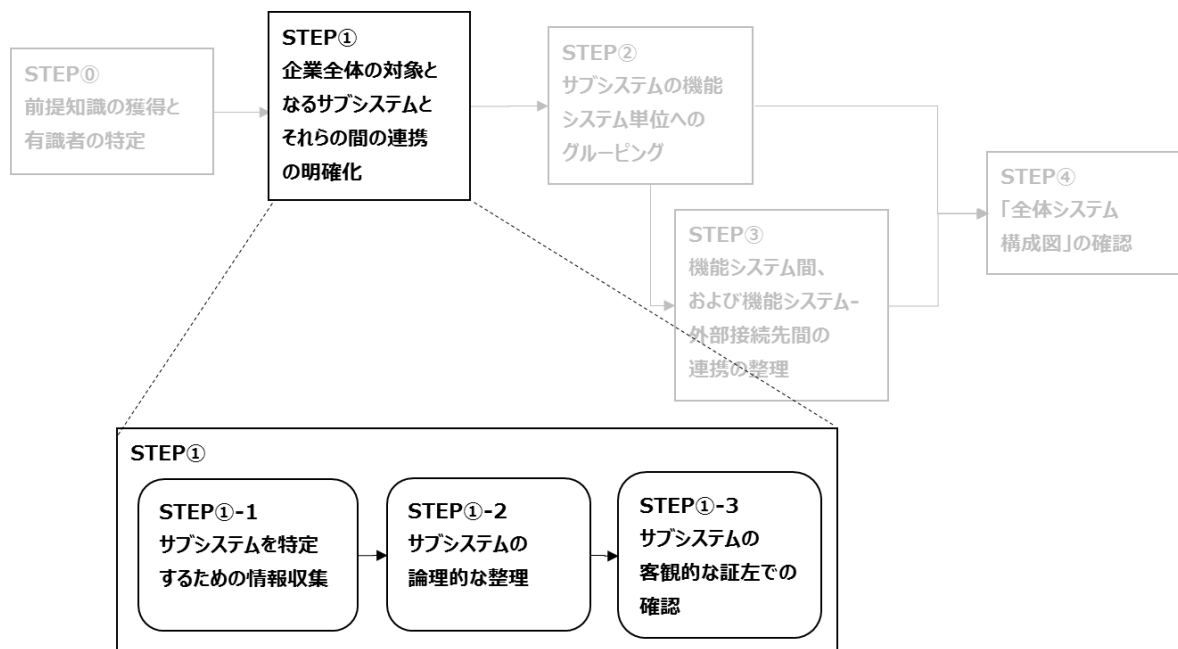
STEP①-2 対象とする企業の有識者の確認

タスク内容：対象とする企業の中で複数の IT システムを押さえている有識者、業務領域を広く理解している有識者を確認する。

留意点：

- ・有識者としては、10 名を限度として業務全体をカバーできるくらいのメンバーを特定する。まずは、IT 部門のメンバーから確認を行い、必要に応じて業務部門の有識者に確認を行う。

□STEP① 企業全体の対象となるサブシステムとそれらの間の連携の明確化



STEP①の目的：全体システムを構成するサブシステムを過不足なく特定し、全体システムの対象範囲を明らかにするとともに、サブシステム間の関連を明確にする。

関係者：複数のサブシステムを把握している IT システム有識者

入力情報：現状稼働しているサブシステム、それらの関連を確認するための資料(例えば、サブシステム構成図、サブシステム一覧、システムの利用ガイド、ジョブ一覧、サブシステムの情報が出力されている帳票)。

出力情報：

「サブシステム一覧」

‘主管部門’、‘サブシステム番号’、‘サブシステム名’に関しては、必須項目とする。‘サブシステム概要’は、当 STEP①では、簡略な内容のみを記載し、すべての STEP が終わるまでに、内容を完成させる。

(図のイメージは、以下の手順 STEP①-2 内に記載)

「サブシステム構成図」

‘サブシステム’、‘外部の接続先’、‘サブシステムおよび、外部との連携’について、記載。

(図のイメージは、以下の手順 STEP①-2 内に記載)

手順：

STEP①-1 サブシステムを特定するための情報収集

タスク内容：企業内で稼働していると思われるサブシステムとそれらの関連について情報を収集する。

留意点：

- ・情報を収集する上では、「サブシステム構成図」、「サブシステム一覧」が主な資料となるが、サブシステムに重複があってもよく、不足することがないように、できるかぎり多くの資料を集める。
- ・「サブシステム構成図」は、収集した後で、組織図の情報と対応付けて、すべてのサブシステム構成図が集まっているかを確認する。

STEP①-2 サブシステムの論理的な整理

タスク内容：収集した資料から、サブシステムを洗い出し、整理をする。また、サブシステム間や外部との関連について、整理する。

留意点：

- ・対象となるサブシステムに関して、情報収集した内容を、「サブシステム一覧」として整理し、重複の確認を行う。以下が作成例。

No	階層化された機能システム名	機能システム名	主管部門	サブシステム番号	サブシステム名	サブシステム概要
A01			旅客販売	01-010	空席照会	空席照会リクエストに応じて、空席情報を提供する。
A02			旅客販売	01-021	予約（一般）	予約リクエストに応じて、座席の予約・変更・取消を行う。
A03			旅客販売	01-022	予約（団体）	団体の予約・変更・取消を行う。
A05			旅客販売	01-030	事前座席指定	予約時に旅客の希望するランクの船室や特定座席を確保する。
A06			旅客販売	01-040	予約記録管理	予約記録を作成し管理する。
A07			旅客販売	01-050	旅客情報管理	旅客の情報を管理する。
A09			旅客販売	01-060	予約管理帳簿作成	予約記録を検索し、帳簿を作成する。
A28			Web販売	07-010	旅客総合Webサービス	予約システム・運賃システムと連動して、空席照会・予約・予約確認・取消・運賃照会・チケットレスサービスを取扱う。

図 2-10 「サブシステム一覧」の例

- ・整理する上では、サブシステムの管理情報(例：サブシステム番号)を活用して、抜け漏れを防ぐことを考慮する。ただし、全体システムとして見た時に、サブシステム番号が重複するなど不整合な状態となっていないか、留意の上、活用する。
- ・収集した「サブシステム構成図」は、粒度や記載の仕方を統一し、整理を行う。その際、「サブシステム一覧」も参照し、「サブシステム構成図」にサブシステムの過不足がないか確認する。また、サブシステム間、サブシステムと外部の間の連携に関しても、最新状態に更新する。「サブシステム構成図」は、以降のSTEPで、サブシステムに関連した連携情報を集約する際にそのインプットとなる。以下が作成例。

【貨物】サブシステム構成図

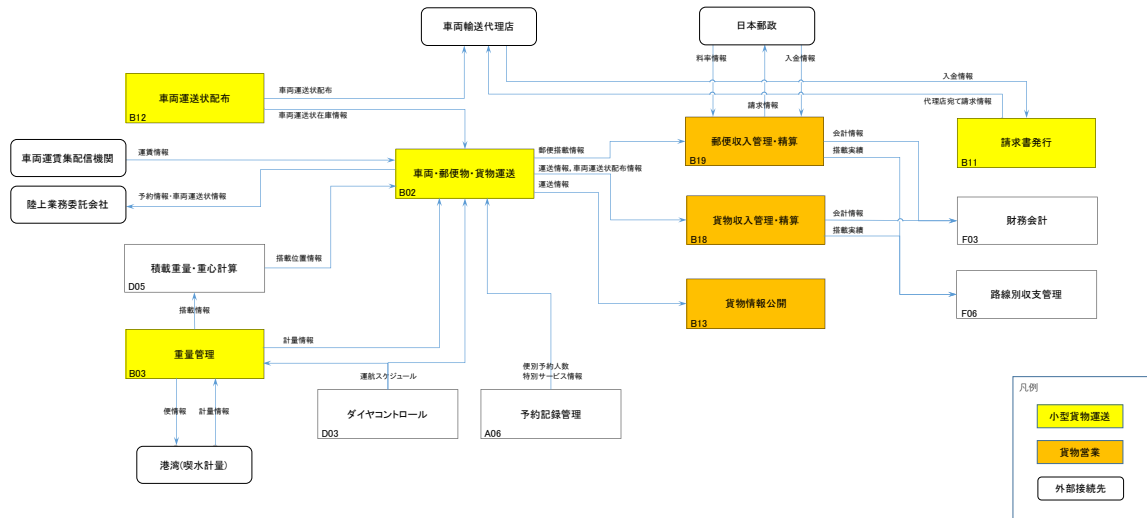


図 2-11 「サブシステム構成図」(例)

STEP①-3 サブシステムの客観的な証左での確認

タスク内容：論理的に整理した「サブシステム一覧」に関し、客観的な証左を用いて、対象とするサブシステムに抜け漏れや重複がないかを確認する。

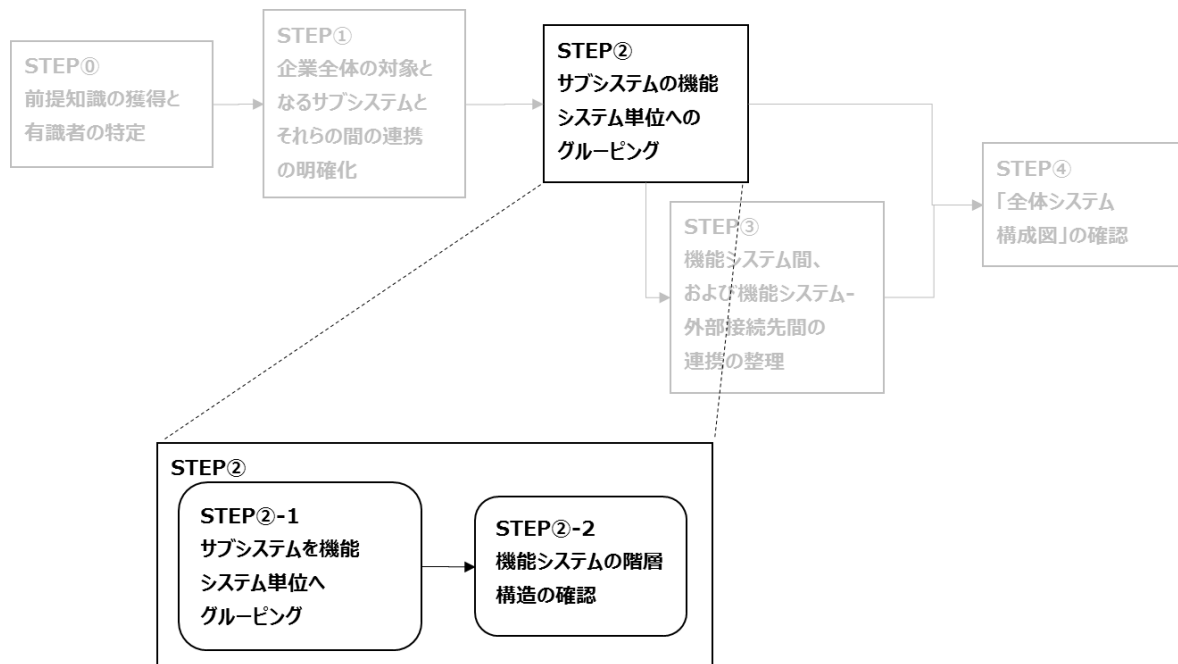
留意点：

- ・当 STEP①-3 は、網羅性の担保が難しい場合、IT システムに関する情報の整備の状態など、状況に合わせて実施する。ただし、当 STEP①で全体システムの対象範囲の確定を行っているため、ここで漏れが発生することは、後に再構築などを行う際に、大きなトラブルへつながることになるため、十分な検討が必要。
- ・対象を精査する方法として、例えば、一年分のシステム運用ログなどを用いて、対象となるサブシステムを確認する。それぞれの IT システムの状況を踏まえ、より確実性の高い方法を検討の上、実施する。確認した結果は、「サブシステム一覧」の確認結果欄へ記載する。また、客観的な証左での確認でサブシステムに漏れが見つかれば「サブシステム一覧」へ追加し、稼働の実績が見つからなければ、「サブシステム一覧」からサブシステムの削除を行う。

No	階層化された機能システム名	機能システム名	主管部門	サブシステム番号	サブシステム名	サブシステム概要	確認結果
A01			旅客販売	01-010	空席照会	空席照会リクエストに応じて、空席情報を提供する。	
A02			旅客販売	01-021	予約（一般）	予約リクエストに応じて、座席の予約・変更・取消を行う。	
A03			旅客販売	01-022	予約（団体）	団体の予約・変更・取消を行う。	
A05			旅客販売	01-030	事前座席指定	予約時に旅客の希望するランクの船室や特定座席を確保する。	
A06			旅客販売	01-040	予約記録管理	予約記録を作成し管理する。	
A07			旅客販売	01-050	旅客情報管理	旅客の情報を管理する。	
A09			旅客販売	01-060	予約管理帳簿作成	予約記録を検索し、帳簿を作成する。	
A28			Web販売	07-010	旅客総合Webサービス	予約システム・運賃システムと連動して、空席照会・予約・予約確認・取消・運賃照会・チケットレスサービスを取扱う。	
	

図 2-12 「サブシステム一覧」の例

□STEP②サブシステムの機能システム単位へのグルーピング



STEP②の目的：サブシステムを機能的にまとめ、集合体として扱うことができるように機能システム単位にグルーピングする。

関係者：複数のサブシステムを把握している有識者、複数業務を把握している有識者

入力情報：STEP①で作成した「サブシステム一覧」、「サブシステム構成図」、企業全体の組織構成に関する情報

出力情報：

「機能システム一覧」

「階層化された機能システム名」、「機能システム名」を記載。

(図のイメージは、以下の手順 STEP②-1 内に記載)

「全体システム構成図」

「階層化された機能システム」、「機能システム」を記載。

(図のイメージは、以下の手順 STEP②-1, 2 内に記載)

手順：

STEP②-1 サブシステムを機能システム単位へグルーピング

タスク内容：STEP①にて整理した「サブシステム構成図」、「サブシステム一覧」、組織構成の情報を利用し、サブシステムを機能システム単位へグルーピングする。

留意点：

- ・全体システムを表す上で、サブシステム単位では可読性が低くなるため、「サブシステム構成図」、「サブシステム一覧」を用い、5～20 程度のサブシステムの集合を 1 機能システムとして、グルーピングする。
- ・グルーピングに際しては、「サブシステム構成図」の作成単位が 1 つの目安となる。ま

た、主管部門毎に機能的なまとまりを持っているので、組織構成の情報も参考情報となる。総合テストを同時に実施する範囲も機能システム単位に行われており、同じように活用できるので、それらの情報を元に総合的に判断する。以下がグルーピングを行った「サブシステム一覧」の作成例。

No	階層化された機能システム名	機能システム名	主管部門	サブシステム番号	サブシステム名
A01		予約	旅客予約	01-010	空席照会
A02			旅客予約	01-021	予約（一般）
A03			旅客予約	01-022	予約（団体）
A05			旅客予約	01-030	事前座席指定
A06			旅客予約	01-040	予約記録管理
A07			旅客予約	01-050	旅客情報管理
A09			旅客予約	01-060	予約管理帳簿作成
B02		小型貨物運送	貨物販売	09-010	車両・郵便物・貨物運送
B03			貨物販売	09-020	重量管理
B11			貨物販売	09-100	請求書発行
B12			貨物販売	09-110	車両運送状配布
A28		e- コマース	Web販売	07-010	旅客総合Webサービス
A30			Web販売	07-020	携帯・モバイルWebサービス
A31			Web販売	07-030	提携法人Web販売
A32			Web販売	07-040	キャッシュレスチケットサービス
A34			Web販売	07-050	顧客企業支援
B13		貨物営業	貨物営業	10-010	貨物情報公開
B18			貨物営業	10-040	貨物収入管理・精算
B19			貨物営業	10-050	郵便収入管理・精算
A40		販売・収入管理	販売管理	080-010	旅客マーケティング支援
A41			販売管理	080-020	特別割引運賃設定（特定地区）
A44			販売管理	080-060	旅客収入管理
A48			販売管理	080-090	特別販売実績
A50			販売管理	080-110	旅客販売実績統計
A12		発券	旅客管理	02-010	乗船券発券
...	

図 2-13 「サブシステム一覧」の例

- ・グルーピングしてできたまとまりを、「機能システム一覧」として整理する。以下が、作成例。

No	階層化された機能システム名	機能システム名
AA01		予約
BA01		小型貨物運送
AA07		e- コマース
BA02		貨物営業
AA08		販売・収入管理
AA02		発券

図 2-14 「機能システム一覧」の例

- ・機能システムとそこに含まれるサブシステムを「全体システム構成図」として作成し、機能システムの構成を確認していく。以下が、当 STEP②-1 における、仕掛中の「全体システム構成図」の作成例。

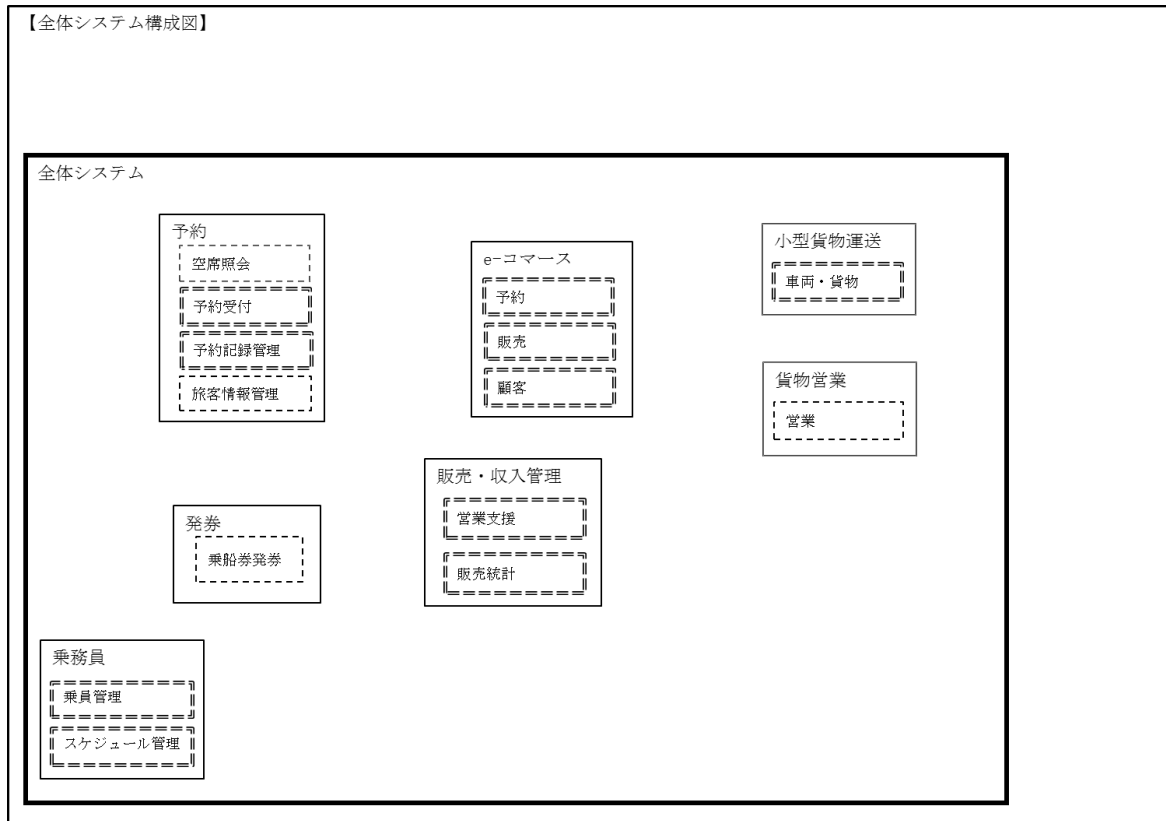


図 2-15 仕掛中の「全体システム構成図」(例)

STEP②-2 機能システムの階層構造の確認

タスク内容:機能システム単位にグルーピングを行った結果、機能システムの数が 2～30 以上となった場合、機能システム自体のグルーピングを行い、階層化する。

留意点：

- ・全体構成の理解のしやすさから、「全体システム構成図」における機能システムを 10 前後の数となるように検討する。主要事業部門は、組織が大きくなっており組織も階層化されているので、該当する機能システムも同じように階層化されている可能性が高い。
- ・STEP②を進めていく中では、STEP②-1 と STEP②-2 は、互いに行き来しながら、検討を進め、機能システム全体の姿を見極める。
- ・階層化した結果は、「機能システム一覧」を更新するとともに、「全体システム構成図」へ反映する。以下が、階層化された機能システムに関して追記した「機能システム一覧」の作成例。

No	階層化された 機能システム名	機能システム名
A01	旅客	予約
A02		
A03		
A05		
A06		
A07		
A09		
A28		e- コマース
A30		
A31		
A32		
A34		
A40		販売・収入管理
A41		
A44		
A48		
A50		
A12		発券
B02		小型貨物運送
B03		
B11		
B12		
B13		貨物営業
B18		
B19		
...

図 2-16 「機能システム一覧」の例

- ・以下が、当 STEP②-2 における、仕掛中の「全体システム構成図」の作成例。階層化された機能システム'旅客'のグルーピングを追記した例(赤の部分)。

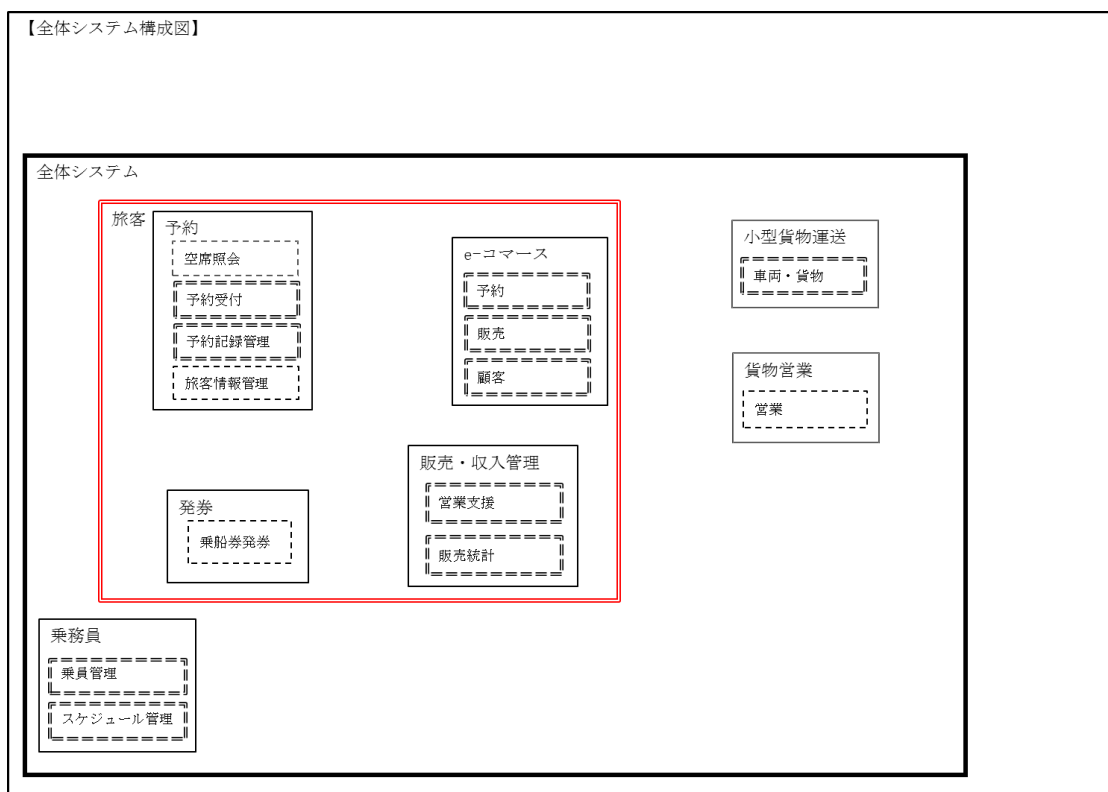
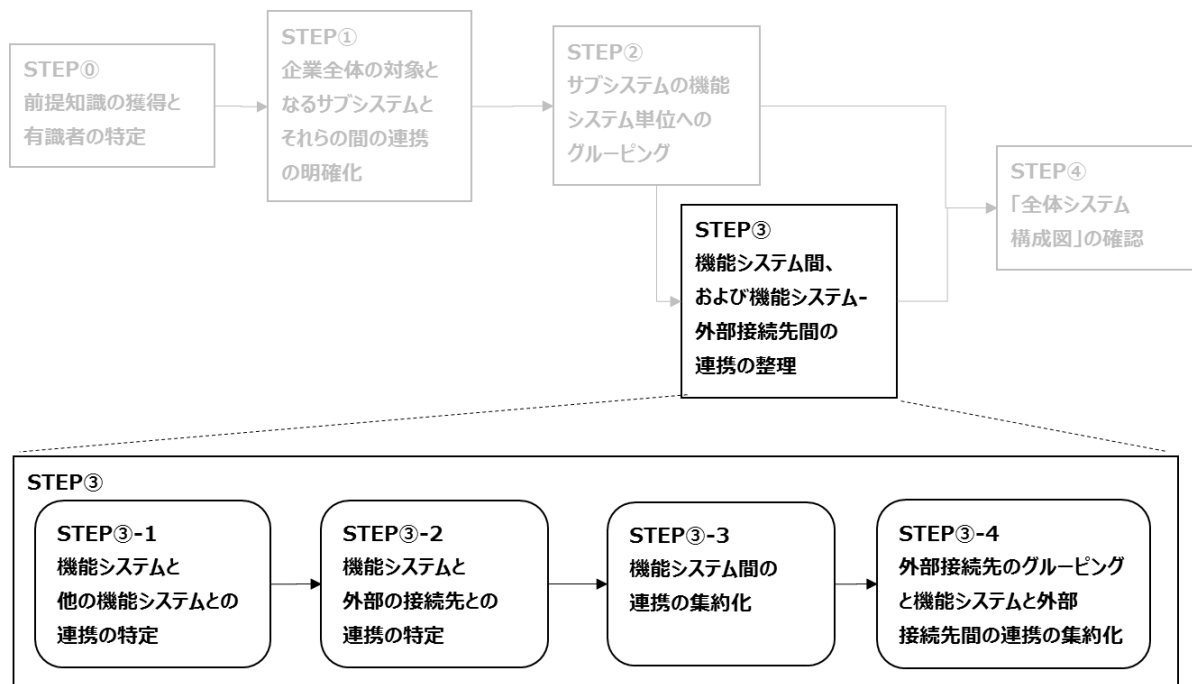


図 2-17 仕掛中の「全体システム構成図」(例)

□STEP③ 機能システム間、および機能システム-外部接続先間の連携の整理



STEP③の目的：機能システム間、および機能システムと外部接続先の間関係を明らかにするため、それぞれの間の連携を整理する。

関係者：各業務の管理者、各サブシステムの管理者

入力情報：STEP②で作成した仕掛中の「全体システム構成図」、「サブシステム構成図」、サブシステム間の連携に関する情報(例えば、インターフェース一覧)

出力情報：

「機能システム間の連携一覧」

’連携元の機能システム名、サブシステム名’、’連携先の機能システム名、サブシステム名’、’キー項目’、’主要項目’を記載。

(表のイメージは、以下の手順 STEP③-1 内に記載)

「機能システム-外部間の連携一覧」

’外部’、’機能システム名’、’サブシステム名’、’主要な連携の内容’を記載。

(表のイメージは、以下の手順 STEP③-2 内に記載)

「外部接続先一覧」

’外部接続先名’、’説明’、’外部’を記載。

(表のイメージは、以下の手順 STEP③-4 内に記載)

「全体システム構成図」

’機能システム’、’外部接続先’、それらの間の連携を記載。

(図のイメージは、以下の手順 STEP③-3,-4 内に記載)

手順：

STEP③-1 機能システムと他の機能システムとの連携の特定

タスク内容：「サブシステム構成図」から、他の機能システムと接続しているサブシステムを確認し、整理する。

留意点：

- ・各機能システムに属するサブシステムから、他の機能システムと接続しているサブシステムを抜き出し、「機能システム間の連携一覧」としてまとめる。以下が参照する「サブシステム構成図」部分と、「機能システム間の連携一覧」の作成例。点線で囲まれた箇所が、対応する部分。

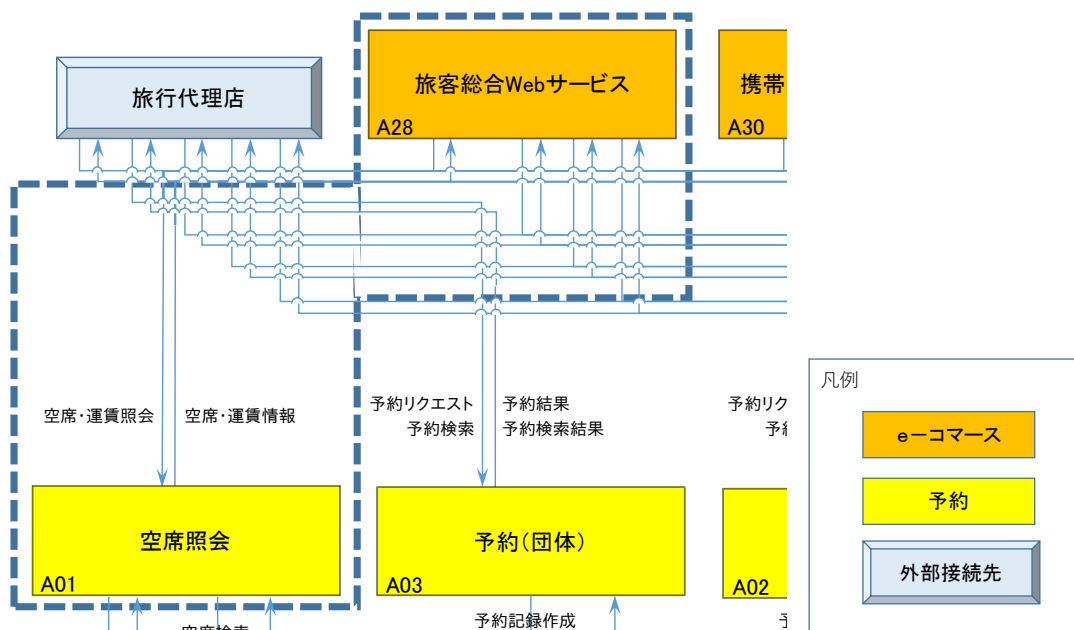


図 2-18 「サブシステム構成図」(部分)の(例)

No	連携元		連携先		連携内容	
	機能システム名	連携元のサブシステム名	機能システム名	連携先のサブシステム名	キー項目	主要項目
01	e-コマース	旅客総合Webサービス	予約	空席照会	照会番号	日時、人数、空席数、運賃
02	予約	空席照会	e-コマース	旅客総合Webサービス	照会番号	日時、空席数、運賃
03	e-コマース	提携法人Web販売	予約	空席照会	照会番号	日時、人数、空席数
04	予約	空席照会	e-コマース	提携法人Web販売	照会番号	日時、空席数
05	小型貨物運送	車両・郵便物・貨物運送	貨物営業	郵便収入管理・精算 旅客情報提供	便コード	郵便種類、数量

図 2-19 「機能システム間の連携一覧」(例)

STEP③-2 機能システムと外部の接続先との連携の特定

タスク内容：「サブシステム構成図」から、外部と接続しているサブシステムを確認し、整理する。

留意点：

- それぞれのサブシステムが属する機能システムから、外部と接続しているサブシステムを抜き出し、「機能システム-外部間の連携一覧」として整理する。以下が、参照する「サブシステム構成図」部分と、「機能システム-外部間の連携一覧」の作成例。点線で囲まれた箇所が、対応する部分。

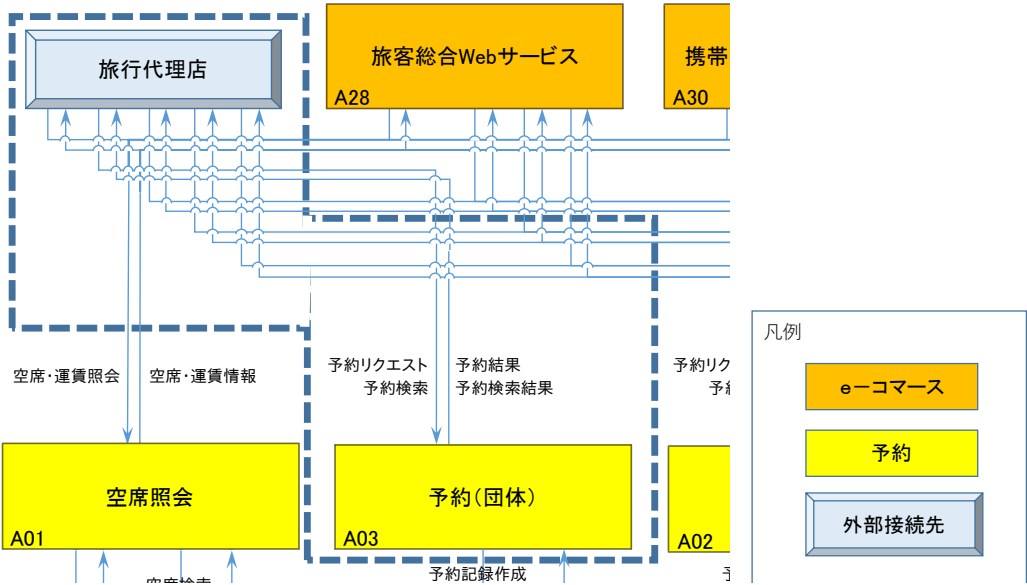


図 2-20 「サブシステム構成図」(部分)の(例)

No	外部→機能システムへの連携			機能システム→外部への連携			連携内容	
	連携元	連携先		連携元	連携先		キー項目	主要項目
	外部	機能システム名	サブシステム名	機能システム名	サブシステム名	外部		
01	旅行代理店	予約	予約(団体)				照会番号	日時、人数、空席数、運賃
02	施設等提供企業	予約	予約(団体)				照会番号	日時、人数、空席数、運賃
03	旅行代理店	予約	事前座席指定				予約番号	便名、座席
04				予約	予約(団体)	旅行代理店	照会番号	日時、空席数、運賃
05				予約	予約(団体)	施設等提供企業	照会番号	日時、空席数、運賃
06				予約	事前座席指定	旅行代理店	予約番号	便名、座席、指定結果
07				販売・収入管理	旅客収入管理	精算機関	請求番号	予約番号、精算金額
08	精算機関	販売・収入管理	旅客収入管理				請求番号	精算日時

図 2-21 「機能システム-外部間の連携一覧」(例)

・STEP③-1 と・2 は、別 STEP としているが、機能システム毎に合わせて進める方法も考えられる。

STEP③-3 機能システム間の連携の集約化

タスク内容：機能システム間の連携について、連携情報を集約する。

留意点：

- ・STEP③-1 で整理した「機能システム間の連携一覧」を使い、サブシステム間の内容を機能システム間で、入力/出力でそれぞれ 1 つ程度にまとめる。まとめる際には、全体の機能の流れが分かるように、機能システム間の連携情報を集約化する。例えば、同じデータの登録/変更/削除といった複数の処理をまとめて表現する。データの種類が複数ある際も、主要なデータの流れに着目する。以下が、「機能システム間の連携一覧」に対して、連携情報の集約化を行った例。

【抽象化前】							【抽象化後】		
No	連携元		連携先		連携内容		連携元	連携先	主要な連携内容
	機能システム名	連携元のサブシステム名	機能システム名	連携先のサブシステム名	キー項目	主要項目	機能システム名	機能システム名	
01	e-コマース	旅客総合Webサービス	予約	空席照会	照会番号	日時、人数、空席数、運賃	e-コマース	予約	予約リクエスト情報
03	e-コマース	提携法人Web販売	予約	空席照会	照会番号	日時、人数、空席数			

図 2-22 「機能システム間の連携一覧」(例)

STEP③-4 外部接続先のグルーピングと機能システムと外部接続先間の連携の集約化

タスク内容: 外部の接続先について、機能システムとやり取りする内容の種類ごとにグルーピングを行う。また、グルーピングした外部接続先と機能システムとの間の連携についても、集約する。

留意点:

- STEP③-2 で、整理した「機能システム-外部間の連携一覧」を用いて、外部接続先のグルーピングと、機能システムとの連携についての集約を行う。
- 外部接続先については、ほぼ同じ内容のやり取りをしている所を 1 つにまとめる。例えば、複数の企業へ商品の販売を委託し、機能システムとデータのやり取りを行っている場合は、それらの外部接続先を「販売企業」とグルーピングする。グルーピングした結果は、「外部接続先一覧」として整理する。以下が、参照する「サブシステム構成図」部分と、「外部接続先一覧」の作成例。点線で囲まれた箇所が、対応する部分。



図 2-24 「サブシステム構成図」(部分)の (例)

No	外部接続先名	説明	外部
01	販売企業	ツアー、乗船券等の予約、販売を委託している先	旅行代理店
02			施設等提供企業
03	精算機関	ツアー、乗船券等の代金の収納代行の依頼先	収納代行会社

図 2-25 「外部接続先一覧」(例)

- ・機能システムと外部接続先間の連携についても、機能システム間の連携と同じように、外部接続先を含めた機能の流れが分かるように、連携情報を集約化する。そのため、1 機能システムと 1 外部接続先間の連携情報を、入力/出力でそれぞれ 1 つ程度に集約する。以下が、「機能システム-外部間の連携一覧」に対して、連携情報の集約化を行った例。

【抽象化前】								
No	外部→機能システムへの連携			機能システム→外部への連携			連携内容	
	連携元	連携先		連携元	連携先		キー項目	主要項目
	外部	機能システム名	サブシステム名	機能システム名	サブシステム名	外部		
01	旅行代理店	予約	予約(団体)				照会番号	日時、人数、空席数、運賃
02	施設等提供企業	予約	予約(団体)				照会番号	日時、人数、空席数、運賃

【抽象化後】				
外部→機能システムへの連携		機能システム→外部への連携		主要な連携の内容
連携元	連携先	連携元	連携先	
外部	機能システム名	機能システム名	外部	
	販売企業	予約		予約リクエスト情報
...

図 2-26 「機能システム-外部間の連携一覧」(例)

- ・グルーピングした外部接続先、および外部接続先と機能システム間の連携について、「全体システム構成図」へ反映する。以下が、当 STEP③-4 における、「全体システム構成図」の作成例。集約化した外部接続先の追記、及び、機能システムと外部接続先の間の連携について、追記をした例(赤の部分)。

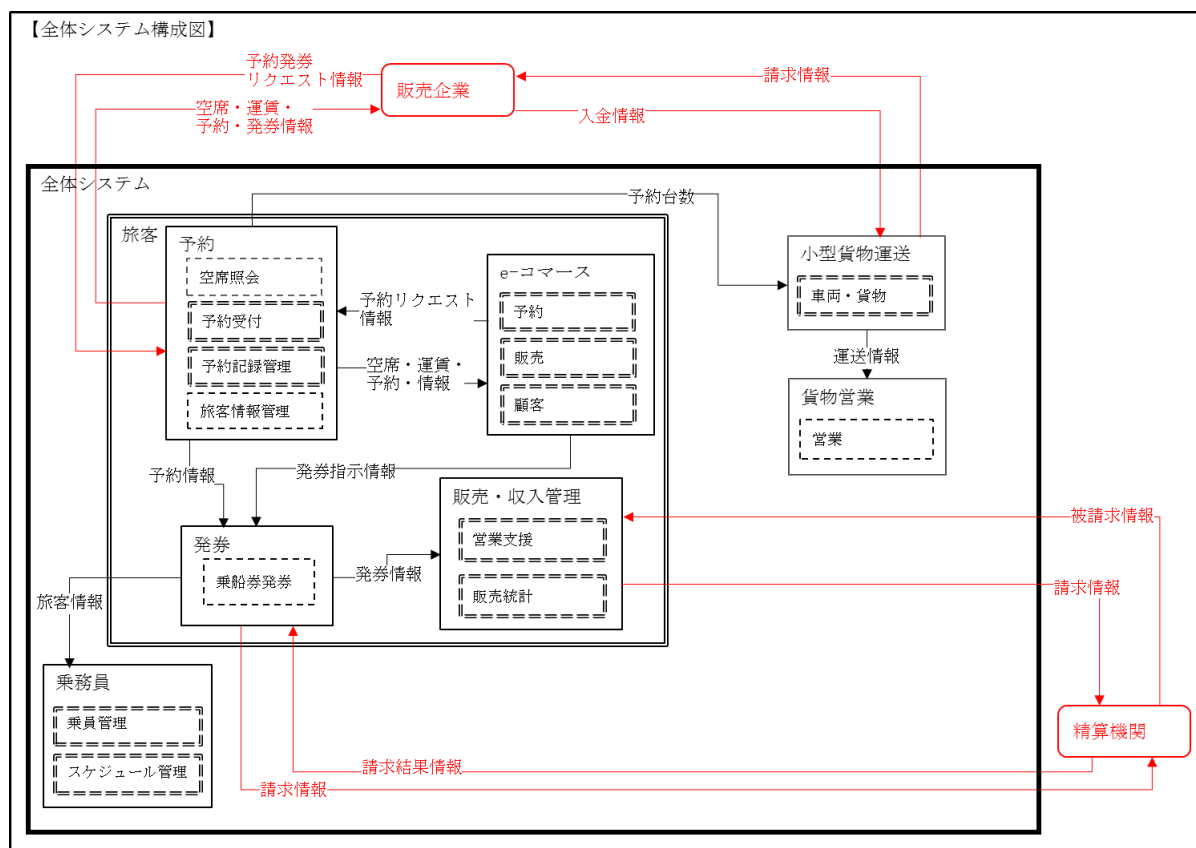
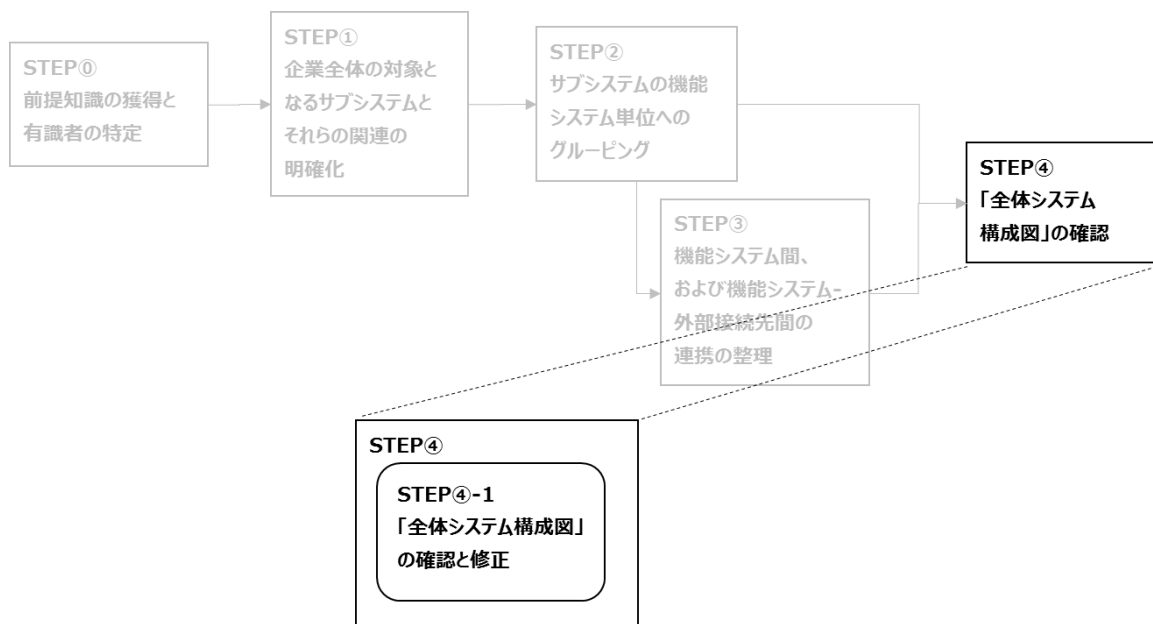


図 2-27 「全体システム構成図」 (例)

□STEP④ 「全体システム構成図」の確認



STEP④の目的：「全体システム構成図」として作成した内容を検証し、全体像を正しく表しているものへ作り上げる。

関係者：複数の業務を把握している有識者、各業務の管理者、複数のサブシステムを把握している有識者、各サブシステムの管理者

入力情報：STEP③までに作成した「全体システム構成図」

出力情報：「全体システム構成図」

手順：

STEP④-1 全体システム構成図の確認と修正

タスク内容：作成した「全体システム構成図」について、これまでの関係者へ確認し、各要素の理解が正しいかを検証する。確認した内容を、「全体システム構成図」へ反映し、完成させる。

留意点：

- ・関係者への確認、その内容の反映というサイクルを繰り返し行い、全体システムに対するお互いの理解を深め、「全体システム構成図」の完成度を上げていく。
- ・全体システム構成図を作成する過程で出てきた、現行 IT システムに関する課題についても、関係者と共有し、認識を合わせておく。

第3章 現行 IT システムの仕様復元

3.1. 現行 IT システムの仕様復元とその必要性

IT システムの仕様復元には、まず現行 IT システムに関する情報の把握と分析が重要である。そのためには、現行 IT システムに関して、全体のシステムがどのような構成となっているかを明らかに把握する必要がある。その手法については第2章「現行 IT システムの全体把握」で示した考え方により、全ての機能システムとそれらを構成する全サブシステムについての把握ができる。

現行 IT システムについてこのように整理を行うと、プラットフォームデジタル化指標を活用することで IT システムの問題点を機能システム単位に明確にすることができる。こうした問題点と、事業上の重要性や優先順位などを併せ考慮したうえで、IT システムの変革計画を作成することが可能になる。その変革計画の中でコストやリスク等の問題で一度に全体の再構築を行うことができずに、現行 IT システムの一部から優先度をつけて再構築せざるを得ないケースがある。こうした場合、現行 IT システムの一部の仕様復元が必要となる。本章では、その仕様復元の考え方を解説する。

なお、仕様復元をするためには地道な情報収集と確認・整理をする作業が必要となるため、対象を十分に絞り込むことが重要であることには留意されたい。

再構築が必要な要因は、以下のような5つのパターンが複合的に組み合わさっている。

1. ソフトウェアおよびハードウェアのサポート切れによるケース

IT システムの維持・メンテナンスを継続していく上で、保守部品や後継ハードウェアの製造中止、そのバージョン・レベルなどメンテナンス対象のソフトウェアサービス（言語など）のサポート終了などが想定される。

2. 部門最適やプログラムコピーによるスパゲティ化や肥大化によるケース

業務の拡張や新規業務の発生時に既存業務の開発に用いたコードをそのままコピーして改変し利用することでプログラムに構造的な錯綜（いわゆるスパゲティ化）が発生したり、使われていないコードが多く内包されていたりする場合、またはリファクタリングを行わず最適なコードの開発を避けてきた場合など、プログラムやコードが複雑化・肥大化する場合が想定される。

3. 当初作っていたときとビジネスモデルが大きく変化して追従できないケース

現行 IT システムを開発した当時と市場やビジネスモデルが大きく変化し、そのままの状態では IT システムが構造的に追従できていない場合が想定される。

4. ランザビジネスコストの増大に耐えられないケース

上記で挙げたような背景から、次第に不要な資産が大きくなりシステムが複雑化し、そのために維持やメンテナンス、運用、もしくは更改に多大なコストが発生している場合が想定される。

5. データ活用や今後のビジネスモデルに対応が難しくなっているケース

新たな技術の進展とその利用により、より正確なマーケティングができるようになってきているが、既存の IT システムを使い続けることで入出力するデータが「タイムリーに取得できない」「精度が低い」「IT システムごとにまちまちな管理がされている」などの原因でデータの粒度や鮮度などの実現できていない場合がある。DX では特にデータ活用が必要だといわれており、データ自体の鮮度や精度も重要となってくる。新たな IT システムにより、リアルタイムで整備されたデータを供給していく仕組みも求められている。

こうした背景から既存 IT システムの多くが抜本的な IT システム変革を求められている。本章で示す仕様復元の考え方は、これらの事態に対応するためのものである。

既存 IT システムは、ウォーターフォールモデルで作られている。そのため、仕様復元するためには、既存のウォーターフォールモデルに沿った遡及方式が必要となり、また、以下の仕様復元の考え方については、その対象がウォーターフォールモデルで作られた IT システムであることを所与の前提とする。

この前提の下で考えると、概要・外部・内部・詳細設計といった各設計工程においてどのような情報が必要かを明確化し、その情報を遡及することが必要となるかを明らかにしなくてはならない。

但し、現在の多くの IT システムでは、残念ながら IT システムの設計構築時に（のちの社員が再構築できるだけの）情報・ドキュメントを残していないことが多い。つまりそもそもの情報が足りないため、再構築・最適化が極めて困難となっている。

本章ではかかる事態に対応し、IT システムを再構築するために必要な概要設計レベルの情報を遡及するための考え方について述べる。

なお、分析の目的はそのまま作り直すことではなく、機能を明らかにすることが目的である。今後新たな IT システムを実装する場合は既存の IT システムとは異なるアーキテクチャで設計・構築することを想定している。新たな設計構築手法については後続の章で記載していく予定であり、本章では業務要件を明らかにすることを中心とした考え方を示すこととする。

それは、すなわち、概要設計を中心とした情報遡及を行うことにより、外部設計以降の分析に関しては、各 IT システムの特異性に合わせて必要な情報を解析することである。今後の新しいビジネスモデル要件と、既存の IT システムの概要設計の遡及情報とを両輪として活用することで、新たなアーキテクチャでの適切な業務要件情報を整理できると考えている。

本章では、上記のような考え方にに基づき、既存の IT システムの概要設計情報の遡及に関する考え方を取り扱うものである。

3. 2. 仕様復元の各工程での段階的情報遡及

既存 IT システムの仕様復元は各工程レベルで実施し、段階的に情報を遡及していくことで結果として、現行の IT システムを再構築するために必要な情報を整備することを基本とする。

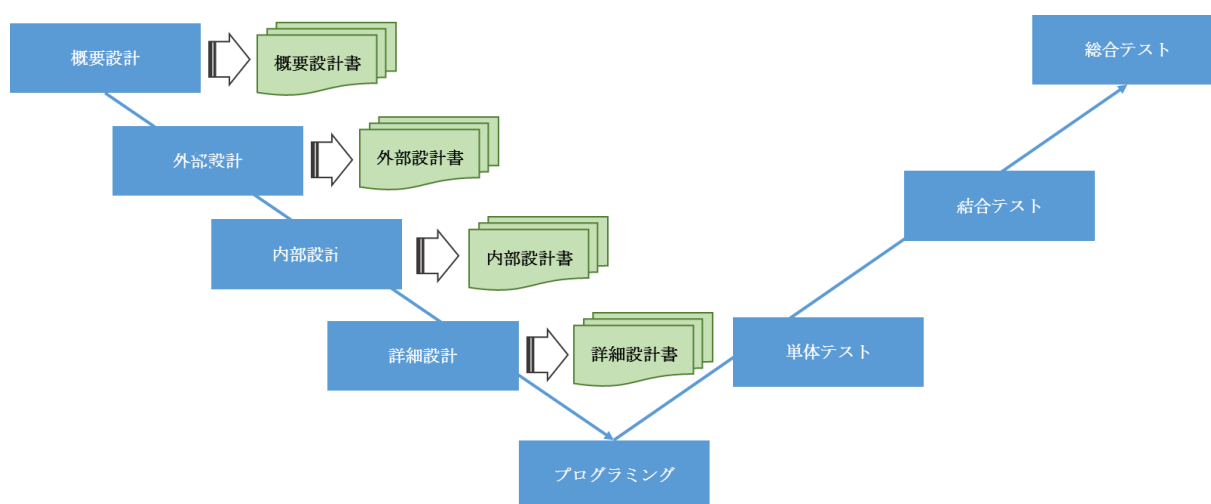


図 3-1 V 字モデル

分析はまず概要設計レベルから、外部設計レベル、内部設計レベル、詳細設計レベルと順に進める。内部設計レベルまで進めると今後新たに開発するシステムとの間でプログラムレベルでの対比が可能となり、この時点で新しい IT システムのプログラムレベルでの担当者も明確になるものと思われる。

またこの段階でプログラムレベルでの担当者が実装すべき範囲と責任も明確となり、結果的に既存 IT システムのプログラムレベルの分析の範囲と目的が明確になる。このように既存のプログラムレベルで必要な機能を遡及することによって、最終的な機能の遡及が網羅的に実施できることになる。

上流工程での分析、たとえば概要設計の機能一覧の分析は過度に詳細に行わず、求められる範囲の概要設計レベルの粒度の調査として実施すべきである。各段階において必要な範囲でプログラム解析をとどめることが合理的である。

設計は、プログラミングに近づくにつれ目的を実現するための手段となっていく。概要設計から外部設計、内部設計、さらには詳細設計と、プログラミングと工程が進む中で目的から手段（設計・実装）という行為が繰り返される。

したがって前行程の目的が何であったかという情報は、次工程では失われていくことになる。すなわちプログラミングや詳細設計から概要設計の情報すべてを復元することはできない。部分的な情報をもとに実際に業務を行っている人、現状 IT システムを維持管理している人、法制度な

どから情報をおぎなっていく必要がある。

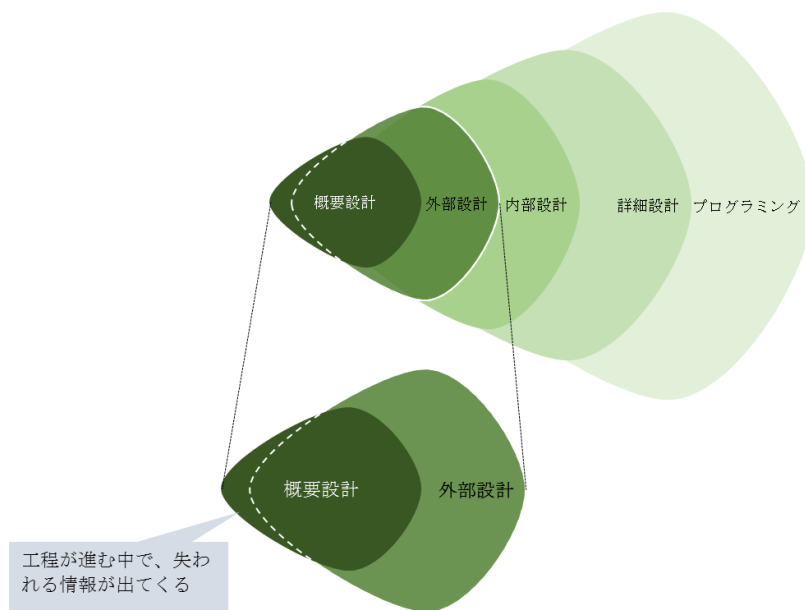


図 3-2 概要設計以降の各工程とそこに含まれる情報の関係

また、IT システムの設計情報などの各種ドキュメント類やログ・稼働実績に加え、業務処理に関する有識者や現場の業務知識のヒアリングによる引き出しなど総合的に情報を収集する必要がある。この情報収集を行うためには、まず、復元すべき情報がなにか定義することが第一である。つまり遡及すべき情報を各工程で明確にしたうえで、既存の設計書やプログラムの分析、および各々の業務の目的をヒアリングする必要がある。ただし広い範囲でこうした手法を用いるとコストや時間もかかり、体制も必要となる。有識者がいない場合、実行モジュールのみでもととのソースコードがない場合など、情報の復元が十分にできないこともある。こうした場合は全ての IT システムの仕様が完全に復元できるとは限らないというリスクがある。不完全な部分を補うためには、その分テストに要する時間なども必要となる。

このように分析を進めながら再構築の範囲を絞り込む必要がある。また、有識者がおらず情報の復元が十分にできない場合には、その IT システムの部分自体が本当に必要な IT システムの部分であるかどうかの判断や、該当 IT システム廃棄も合わせて行う。

3.3. 現行 IT システムの仕様復元の概要

前節までで、仕様復元の重要性について述べたが、さらに重要かつ先に検討すべきこととして、全体システムの中の対応方針の策定および対応する機能システム優先度付けがある。

現在の IT システムには非競争領域に相当するものと競争領域に相当するものがある。非競争領域とは明らかに競争領域であると定義できるものの以外のすべてとしてとらえるべきである。非競争領域に関して積極的に他者とともに共通的に実装するなどコストやリスクを軽減することで、競争領域にリソースを効率的に充てることが可能となる。競争領域は個社として取り組む必要があるが、その場合、PF デジタル化指標の評価の結果や事業のプライオリティなどを鑑み、優先順位をつけたり不要な IT システムを廃棄して絞り込みを行ったりした上で仕様復元の範囲を絞り込み実施していく必要がある。なお、競争領域は時とともに非競争領域に変化していく場合もある。

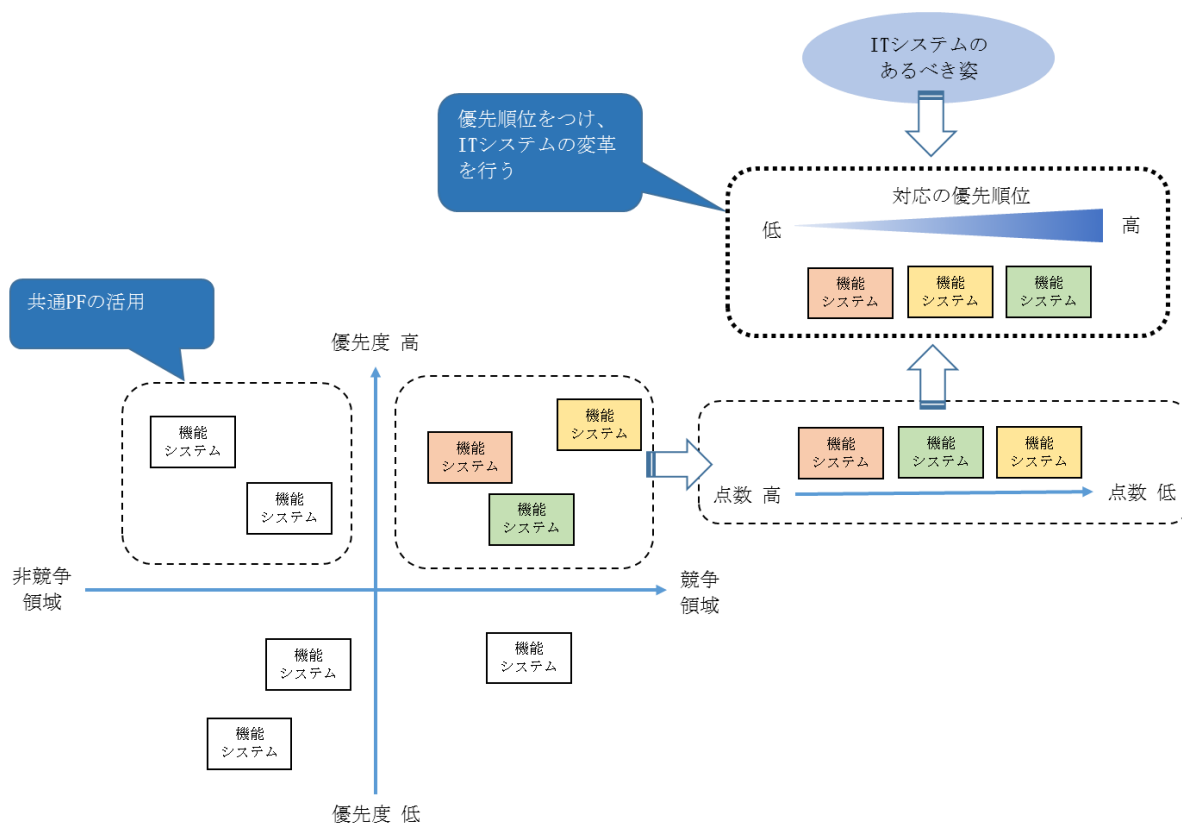


図 3-3 IT システムの対応方針

3. 4. 概要設計の遡及の考え方

概要設計の分析とは、機能システム全体を示す概要設計全体の分析と、そこで明らかになったサブシステムごとの個別の概要設計の分析からなる。全体分析に関しては第 2 章で述べているように、既存 IT システムの AsIs の整理の中で、機能システムに存在する全てのサブシステムが明らかになっており、これらのサブシステムの関係についても整理されているはずである。

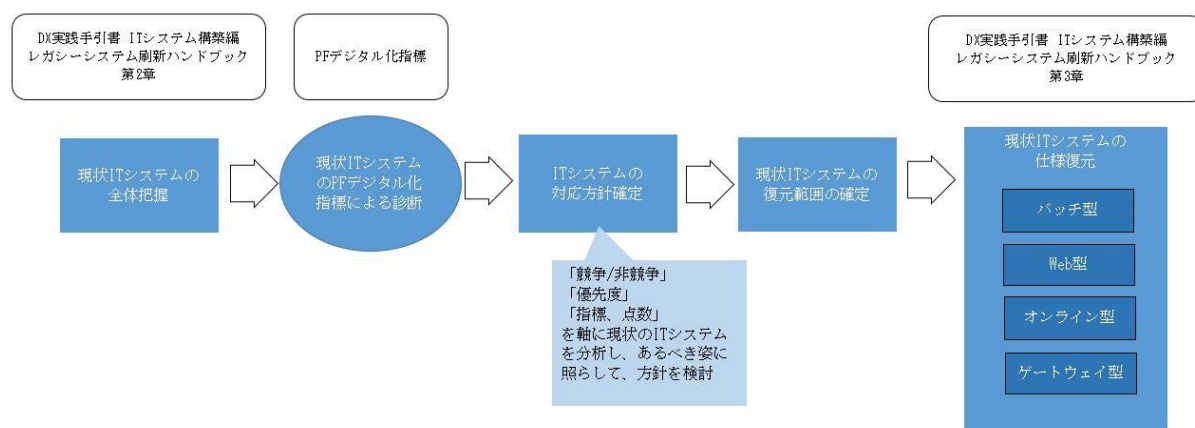


図 3-4 現状 IT システムの全体把握から仕様復元に至る流れ

したがって、本節で説明すべきは個別のサブシステムの概要設計の分析方法である。

各サブシステムの概要設計を分析するにあたっては、それぞれの機能を表すキーとなる設計情報（キー情報）を明確化することが求められる。このキーとなる設計情報は当該サブシステムのシステム形態ごとに異なるというのが、根本的な考え方であり、その種類は現状では 4 つの形態に分けられると考えている。その形態とは、バッチ型・オンライン型・Web 型・ゲートウェイ型である。

バッチ型はシーケンシャルなデータを読み込み順次処理をしていく形態のサブシステムである。通常ジョブフローといわれるものが存在しており、ジョブフローに記述されているプログラムが順次実行されて処理される。本形態の場合はデータ種類ごとに振る舞いを記述する DFD（データフロー図：Data Flow Diagram）が最も業務要件を明確に示すことができる。したがってバッチ型の概要設計のキー情報は、DFD と考える。

オンライン型はクライアントサーバ、Web、メインフレームなどの実現方式はあるが、基本的には業務の流れが明確となっている形態を示し、業務フローを整理することで業務要件を明確にすることができる。したがってオンライン型の概要設計のキー情報は、業務フローと考える。

Web 型は主に Web を活用した実現方式である。ただし Web 型方式でオンライン型の処理を実現している社内（従業員）向けのシステムの場合などはオンライン型に分類する。Web 型はあくまで B2C 向けのシステム形態を表すものである。この場合、業務が一定方向ではなく前の業務に戻ったり、他の業務画面を開いたりするなど、業務の順番が規定できない形態となっている。この場合は業務要件を明確に示すためには画面遷移図が適している、したがって Web 型の概要設計のキー情報は、画面遷移図で示すことができると考える。

ゲートウェイ型は、外部システムとデータを会話型でやり取りするシステム形態を表す。たとえば、証券会社と取引所で株の注文のやり取りをするように他システムと接続するシステム形態を想定している。この場合は業務ごとに、時間とともに双方のやり取りを記述する状態遷移図で整理すると業務要件を明確にできる。したがってゲートウェイ型の概要設計のキー情報は、状態遷移図で示すことができると考える。

実際に新たな IT システムを構築する場合、上記のレベルの粒度の業務要件が明確になると、新たな業務要件等を加えて新しい IT システムの設計をするには、十分な情報となると考えられる。したがって本書では、これらの 4 つのシステム形態ごとのキー情報の復元をするための考え方を整備することとする。ただし詳細な現行機能について保証する場合は、各工程で段階的な機能詳細化をする必要がある。これについては前述した考え方を元に、適用する IT システムごとにアプローチしていくべきと考える。

3.5. 現行 IT システムの仕様復元 (バッチ型)

本節では、バッチ型での仕様復元の分析方法について記載していく。

大量なデータに対して決められた処理を一括して行う IT システムをバッチ型システムと定義する。IT システムの資源を効率的に活用するため、夜間などに一括してデータ処理を実行する仕組みである。バッチ型の IT システムは対象となるデータ群に対して、どのデータをどういう順番で処理するかが決められているのが特徴である。

バッチ型の IT システムは、複数のバッチプログラムから構成されている。一般に業務処理ではこれらのバッチプログラム群が組み合わせて実行されていると考えてよい。

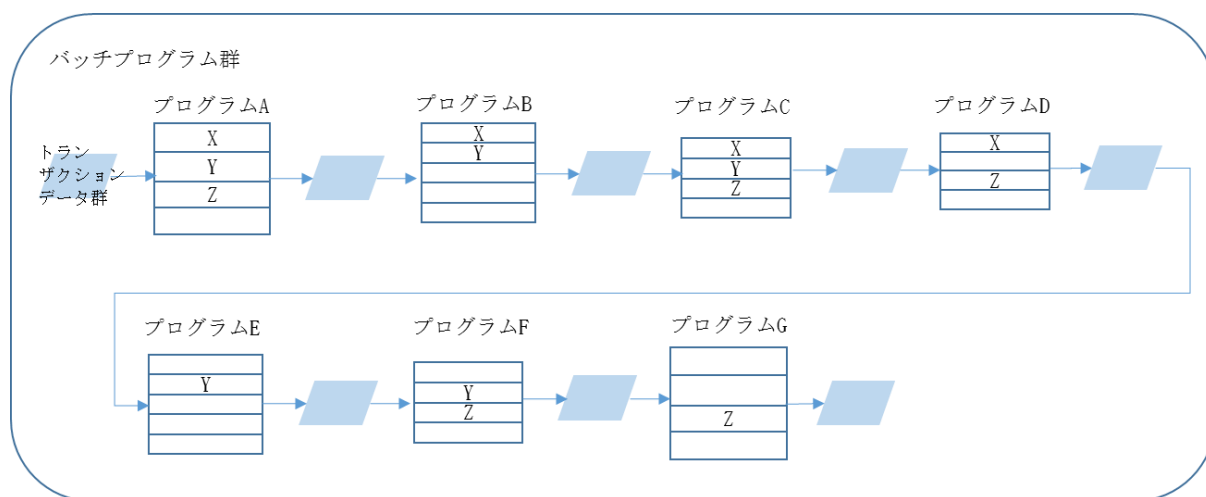


図 3-5 トランザクションデータ群とそれら进行处理するバッチプログラム群(例)

例えば、あるバッチ型の IT システムではバッチプログラム A,B,C,D,E,F,G から構成されている。X トランザクション業務はバッチプログラム A,B,C,D 4 つのプログラムでのみ処理が実行され、同様に Y トランザクション業務はバッチプログラム A,B,C,E,F のみを順に処理することで実行され、また Z トランザクション業務は A,C,D,F,G のみを順に処理することで実行されるといった具合になる。バッチプログラム間の処理結果やデータの引き渡しはファイルやデータベースを介して行われることが多い。

このようにバッチ型の IT システムでは図のように一連のバッチトランザクションプログラム群によって処理されて初めて複数の業務処理が完結するように作られている。

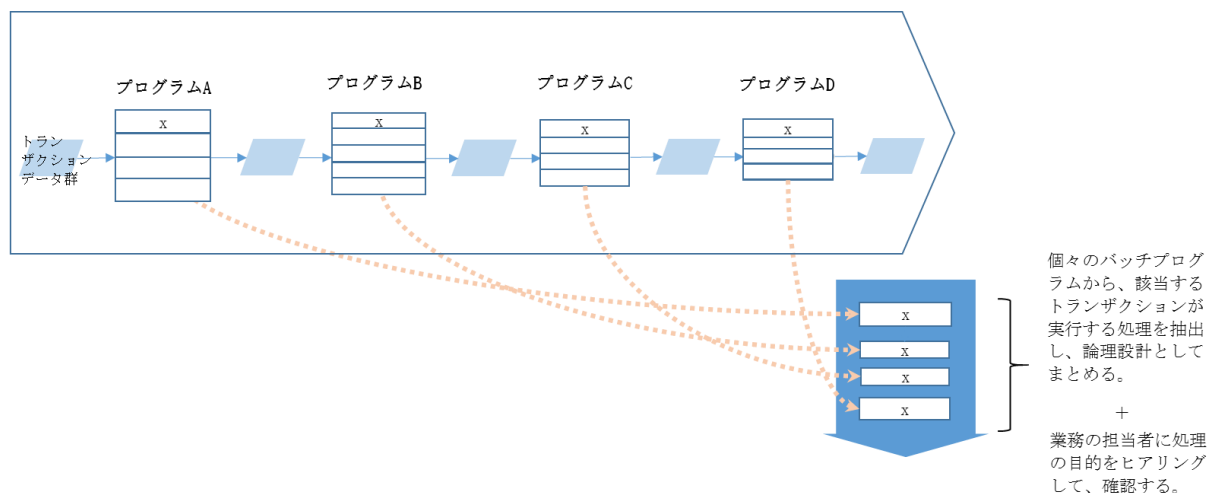


図 3-6 トランザクションデータ毎にバッチプログラム群から論理設計を行う例

この図では、前述した X トランザクションに係るバッチ処理プログラム A,B,C,D の該当部分のみを抽出したものである。

バッチ型の IT システム上に実装されたある業務システムの仕様復元をしようとした場合、単一のバッチプログラム（たとえばプログラム A）だけ解析しても業務全体を把握することができない。すでに述べたように一つの業務は一連のバッチプログラム群が順に処理されて初めて完結する構成となっているため、バッチプログラム群を串刺しで解析し対象の処理の部分だけを抽出して初めて業務の内容が理解できることになる。したがって、バッチ型の IT システムを把握するためには、複数のプログラム処理中で、トランザクションごとに入出力処理・DBの更新処理・演算処理などの処理を洗い出すことが必要となる。したがって、こうした関係を整理するのには DFD での表記が適しているのである。

3.6. 現行 IT システムの仕様復元(オンライン型)

本節では、オンライン型での仕様復元の分析方法について記載していく。

オンライン型の IT システムは企業における業務の中核を担っており、全社の業務フローに対応した機能を実装している。オンライン型の IT システムの仕様復元はまずは業務の順にしたがって、いわゆる業務フローを整理することが第一である。次にこの業務フローに基づいてオペレーション単位に画面を網羅的に洗い出すことで、画面一覧を作成することができる。これにより対象となる業務要件の網羅性を担保できる。このほかオンライン型の IT システムはその IT システム内に閉じず、他の IT システムとの間でインターフェースを持っていることがある。他の IT システムを呼び出したり、他の IT システムから呼び出されたり、データや処理を引き渡したりする動作も同様に洗い出しておく必要がある。さらに、オンライン処理で、参照・更新・変更を行っているDBや、帳票出力なども一覧として整理する必要がある。

重要なのはあくまで対象となる業務フローの正しい把握である。対象となる業務フローの正しい把握ができていないと、上記のような分析が正しくできなくなるからである。

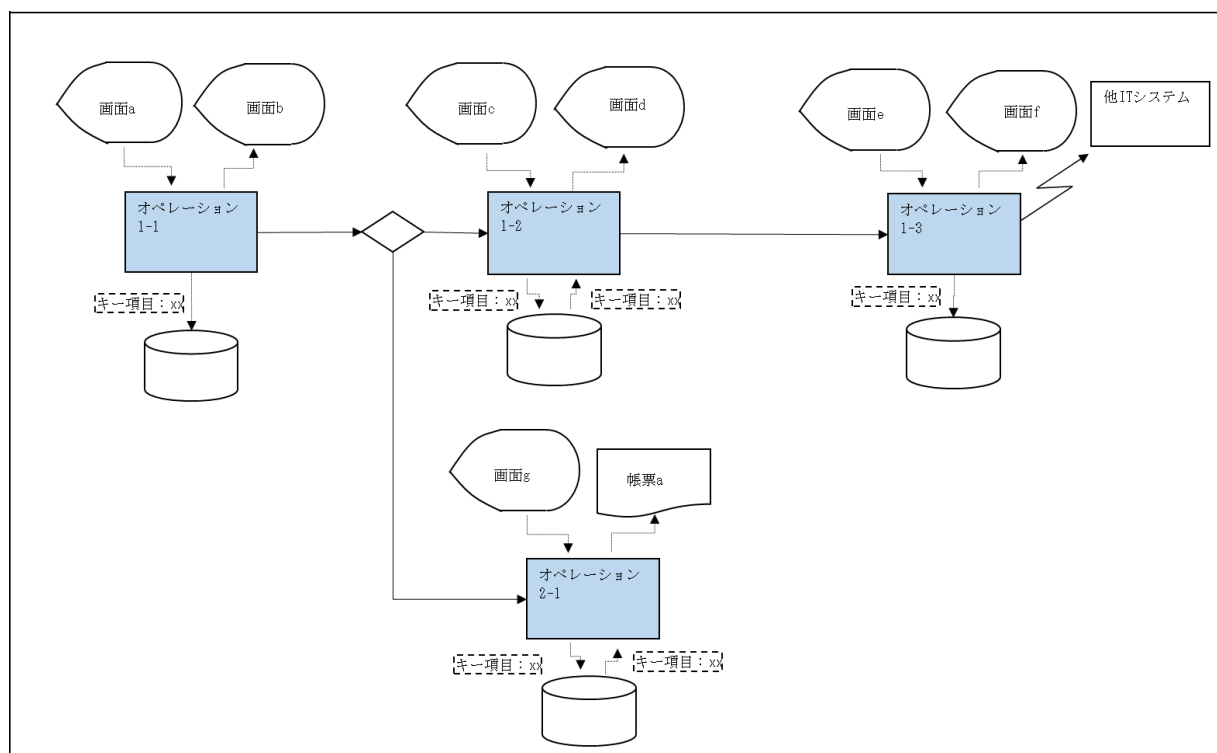


図 3-7 業務フロー(例)

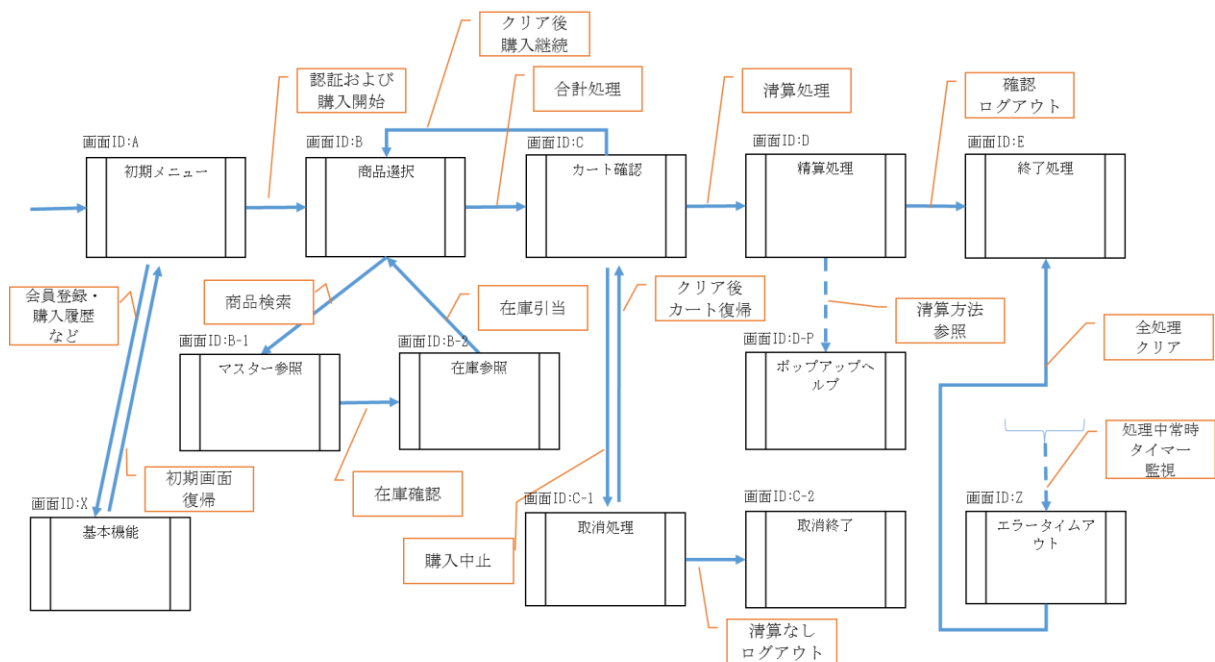
3.7. 現行 IT システムの仕様復元 (Web 型)

本節では、Web 型での仕様復元の分析方法について記載していく。

90 年代の後半からインターネットプロトコルによるネットワークの拡大、HTTP を中心としたプロトコルの普及で Web 型の IT システムが急増した。Web 型はオンライン型と IT システムの形態は似ているように見えるが、オンライン型と Web 型の大きな違いはオンライン型では業務処理の順番が決まっているのに対し Web は業務処理の順番が一定とは限らない点である。このため画面の流れを、業務フロー図として表すことは困難である。これらのことから、Web 型の処理の分析再整理は、画面遷移図を用いると具体的な処理内容を表記しやすい（なお、現方式として Web の処理方式をとる場合でも、社内業務処理に適応している場合は、オンライン型として仕様復元することを考えた方が適切な場合が多いかもしれない）。

手順としては、まず業務そのものや業務マニュアルを元に業務画面を洗い出すことから始める。さらに実際の画面のメニューからすべての業務画面を洗い出し、画面定義からも業務画面を洗い出す。これら 3 つの業務画面の整合性をとることで本来業務に必要な業務画面を確定させ画面一覧を作成する。画面メニューを参考にしながら、画面間の関係を整理することで、最終的にはすべての画面遷移の流れを整理する。

具体的には、オンラインショッピングなどでは、ログインして購入する物品や数量を指定して決済するといった大きな流れは、オンライン型とかわらないように見える。しかしながら Web 型では購入したい物品を画面遷移しながら探したり比較したり、カートに入れたり取りやめたりしながら進め、最後に決済方法や確認をしながら完結させることになる。したがって、画面は一定のルールで進むとは限らず、ときには行ったり来たりしながら一連の作業をすることになる点でオンライン型とは大きく異なる。そのため、業務フローをキー情報としては仕様を明らかにできず、画面遷移の流れを整理することでアプローチせざるを得ないのである。



No	画面ID	画面名	処理概要	関連データベース		遷移先画面
				入力	出力	
1	A	初期メニュー	・初期メニューを表示する ・ユーザの認証を行う ・商品選択へ進む	会員情報		B: 商品選択
			・基本機能へ進む	会員情報 購入履歴		X: 基本機能
・	・	・	・	・	・	・

図 3-8 画面遷移図(例)

3.8. 現行 IT システムの仕様復元(ゲートウェイ型)

本節では、ゲートウェイ型での仕様復元の分析方法について記載していく。

IT システムはネットワーク化が進んでいるため、他の IT システムと情報を交換しながらお互いの IT システムが処理を進めるシステム形態をゲートウェイ型と呼称することにする。ゲートウェイ型の IT システムでは、ある一定のステートから、データあるいは開始時間などのトリガーに応じて演算や処理分岐・応答振り分け・集約などのアクションを通して次のステートに順次変化していく。このため既存のゲートウェイ型の IT システムの設計情報を再整理する場合には、時間とともに状態が遷移するため、状態遷移図を用いて分析・定義するのが適している。

IT システム分析において状態遷移で表現を行う場合には、通信レイヤーの制御系と、業務系処理の制御系の両方の整理が必要となる。制御系の整理としては正常系では主に業務に先立つセッションの確立や切断、認証処理や監視が挙げられる。おなじく制御系の異常系としては自ステート内での障害や呼び出し先もしくは呼び出し経路上での障害の制御方式が挙げられる。業務アプリ系では正常系に加え、業務的なエラー（メッセージの不整合やデータ型式の不一致）への対応、さらには業務機能呼び出しのタイムアウトなどもこの範疇となる。

ゲートウェイ型では独立した IT システム同士で状態が遷移していくため、全体的に見ると非同期処理として扱われることになる。このため同期が必要となる処理に関しては特別な考慮が欠かせない。

整理の手法としては、まずは業務手順や業務マニュアルなどを参照しながら、業務の正常系の流れから分析を開始する。正常系の流れの整理が完了したらこれに対しての異常系の整理をして状態遷移を追記していくことになる。特に異常系ではトランザクションごとにタイムアウト（応答なし）や異常応答（応答はあるも異常値が返るなど業務的なエラー）にも配慮しながら整理をする。これらに対し最終的には異常発生時の処理も含めて網羅的に整理を完結させる。異常系でも更新系と異なり参照系の場合には再送だけでよい場合などもあり、処理内容に応じてステートも分けて整理することが必要である。

No	分類	処理	処理概要	処理タイミング
A1	制御系処理	開局	・接続先との接続を確立する	毎営業日 09:00
A2		閉局	・接続先との接続を閉鎖する	毎営業日 15:00
...	
B1	業務処理	売買処理	・接続先へ売買の依頼を行い、結果を受領する ・受領した結果を、内部へ連携する	開局中随時
...	
X1	エラー処理	タイムアウト	・接続先より、規定時間応答がない場合に、エラーを発生させる	各業務処理内でのエラー発生時
...

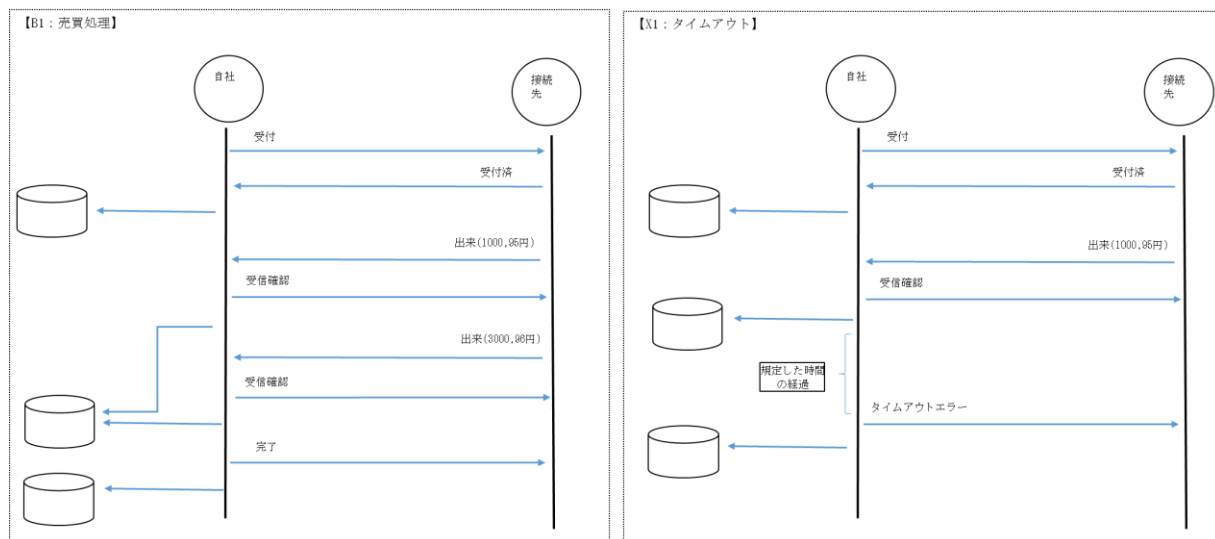


図 3-9 状態遷移

DX 推進事業 DX 実践手引書 IT システム構築編

レガシーシステム刷新ハンドブック

【全体監修】（敬称略）

DX推進部

室脇 慶彦

境 真良

【編著者】（敬称略）

第1章

DX推進部

室脇 慶彦

三部 良太

第2章

DX推進部

鎌田 高輝

DX推進部

河野 太基

第3章

DX推進部

河野 太基

DX推進部

鎌田 高輝

【第2章・第3章素材提供】（敬称略）

株式会社 野村総合研究所

【専門委員】（敬称略）

名古屋大学 名誉教授

山本 修一郎

【改版履歴】

令和 3年 3月31日 「プラットフォーム変革手引書」第1版 発行

令和 3年 11月16日 「プラットフォーム変革手引書」第1版から「DX 実践手引書 IT システム構築編 レガシーシステム刷新ハンドブック」に名称変更・加筆改訂

【監修所】 独立行政法人情報処理推進機構（IPA）

【発行所】 独立行政法人情報処理推進機構（IPA）

社会基盤センター DX 推進部

〒113-6591

東京都文京区本駒込二丁目 28 番 8 号

文京グリーンコートセンターオフィス

URL : <https://www.ipa.go.jp/>

謝辞

本手引書作成にあたり、株式会社 野村総合研究所の「**Method-M**」における現行 IT システムの現状把握の手法、素材等の提供をいただき、多大なるご協力をいただきましたことを、心より感謝申し上げます。