

帳票作成自動化によるデジタル・トランスフォーメーションの推進

江谷典子（Peach・Aviation（株））

概要 事業の管理部門において 2017 年度から 2019 年度までデジタル・トランスフォーメーション（DX）を推進している。該当部門では、データ処理がブラックボックス化し、部門内の管理すべき情報資源が不明確となり、自動化による業務改善や新規なシステム化が困難であった。そこで、エクセル帳票からマスタファイル管理やデータ処理を独立させ、帳票生成を行うデザインパターンを導入し、自動化を実現した。その結果、情報共有が進み、業務プロセスやビジネス・モデルを改善できるような新規の情報サービスを提供できるようになり、DX の達成を実現した。本稿では、DX の課題に取り組んだ具体的な情報システム開発を紹介する。

1. はじめに

デジタル・トランスフォーメーション（DX）とは、「IT の浸透が、我々の生活をあらゆる面でより良い方向に変化させる」という概念で、2004 年にスウェーデンのウメオ大学のエリック・ストルターマン教授が提唱したとされる[1]。ストルターマン教授は、この DX の実現に至る段階を次の 3 つのフェーズに区分し、段階的な取り組みから第 3 フェーズになることで、DX が達成されるとしている[2],[3]。

- 第 1 フェーズ：IT 利用による業務プロセスの強化
- 第 2 フェーズ：IT による業務の置き換え
- 第 3 フェーズ：業務プロセスやビジネス・モデルを変革する段階

また、経済産業省 DX レポートでは、多くの経営者がデジタル・トランスフォーメーションの必要性について理解しているとして、次の課題を指摘している[4],[5],[6],[7]。

- 既存システムが事業部門ごとに構築されて全社横断的なデータ活用不可。
- 過剰なカスタマイズがなされているなどによる複雑化・ブラックボックス化。
- データ活用のために上記のような既存システムの問題を解決し、そのためには業務自体の見直しも求められ、現場サイドの抵抗も大きいという条件の中での実現方法。

上記の定義や課題を背景として、私たちは 2017 年度から人工知能やデータ活用による情報処理の自動化による業務改善や新規ソリューションに取り組み、情報資源の管理部門において 2017 年度から 2019 年度まで DX を推進している。該当部門は、一部業務アプリケーションによる帳票用データ作成を行っているが、エクセル帳票による管理を基盤としたデータ処理やデータ管理を行っている前述の第 1 フェーズにあった。そのため、データ処理がブラックボックス化し、部門内の管理すべきデータが不明確となり、自動化による業務改善や新規なシステム化が困難であった。そこで、情報システム開発者 1 名と該当部門担当者 7 名による

タスクフォース[8],[9]により、ブラックボックスとなっている帳票作成の仕様を分析し、サービスを拡張して帳票作成の自動化ができるような情報システムの開発に取り組んだ。その結果、エクセル帳票による管理はそのままとし、マスタファイル管理やデータ処理を Jav コンピュータシステムに移行させることで、第2フェーズへ移行した。第2フェーズの運用に入ると、第2フェーズの情報システムを情報基盤として、出力されたエクセル帳票を共有し再利用した新規の情報提供サービスを実現できるようになり、第3フェーズへ移行した。

本稿では、該当部門における DX の課題解決について、具体的な情報システム開発を紹介し、第1フェーズから第2フェーズの移行、および第2フェーズから第3フェーズの移行について説明する。

2. 第1フェーズの課題

スケジュール管理データベースから抽出するスケジュール管理 CSV ファイルと業務アプリケーションから出力される CSV ファイル（8種類）から手作業でエクセル帳票（6種類）を作成し、スケジュール変更があれば適時エクセル帳票を訂正している。この状況を第1フェーズと捉えている（図1）。第2フェーズへの移行では、手作業を全て自動化し、全帳票を自動作成できるようにする（図2）。また、新規にエクセル帳票を2種類を追加した。改良ポイントは次の通りである。

- 各自の帳票で利用しているデータと手作業で行ってきた帳票の項目データを収集してマスタファイル（13種類）を作成する。
- 帳票作成のためのテンプレート（32種類）を準備する。
- データ処理ではスケジュール管理 CSV ファイルとマスタファイルからエクセル帳票用データファイルを作成する。該当帳票のテンプレートを用いて、エクセル帳票用データを該当新規エクセル帳票（11種類）へ書き込み、印刷設定を行う。

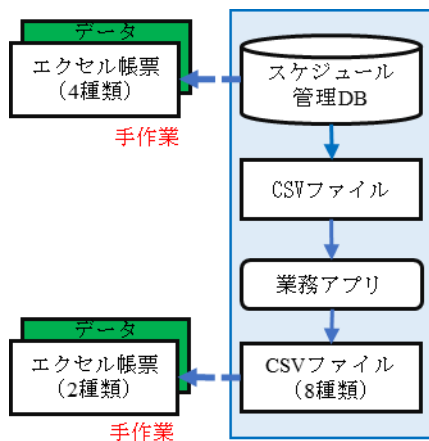


図1 第1フェーズ

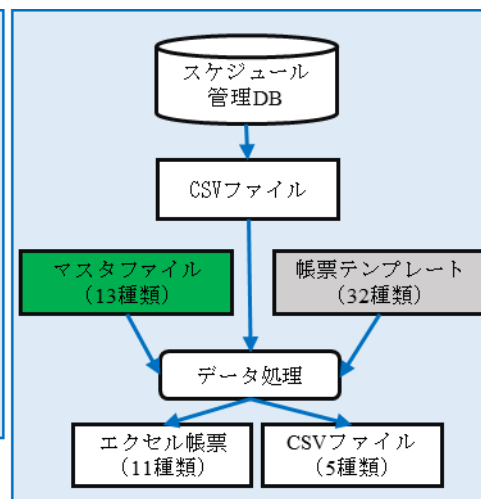


図2 第2フェーズ

3. 第2フェーズの取組み

3.1 クラス図

帳票作成自動化を実現するため、UML ツール[10]を用いてクラス図を作成した。図3はクラス図の鳥瞰図を示す。スケジュール管理 CSV ファイルは更新系データベースとして作成され、マスタファイルはマスタ系データベースとして作成される。データベースは帳票用データセットを作成するためにデータのマージとソートを行うフィルタとして機能する。複数のデータベースからエクセル帳票用データセットは、CSV ファイルあるいはテキストファイルとして作成される。該当データセットを帳票へ書き込んでエクセル帳票を作成している。図3では前述の帳票作成パターンを四角で囲んでおり、6種類のパターンを示している。

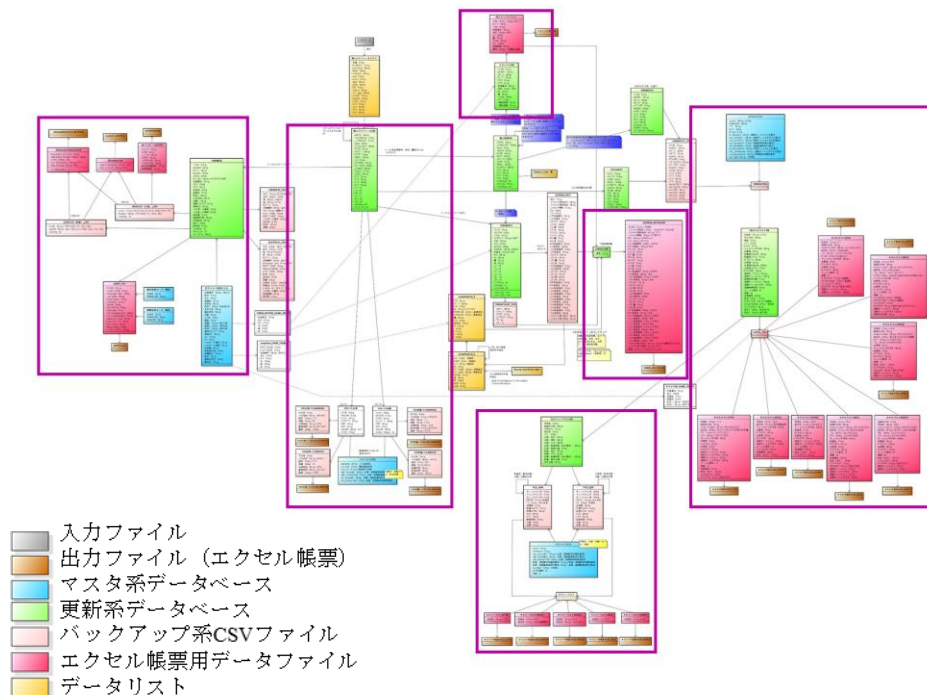


図3 クラス図の鳥瞰図

3.2 デザインパターン

図3で示した6つの帳票作成パターンについて分析を行い、再利用性の高い設計とするためにデザインパターン[11]を導入した。図4ではエクセル帳票作成自動化のデザインパターンを示す。帳票用データセットファイルを作成するために、該当データベースをフィルタ機能として利用する帳票テンプレートフィルタインタフェースクラスを作成した。6種類のインタフェースクラスを実装することで11種類の帳票用データセットファイルクラスは作成される。エクセル帳票クラスでは、エクセル帳票テンプレートクラスが持っている該当のテンプレートファイルを複製して新規帳票を作成し、帳票用データセットクラスが持っている該当のデータセットファイルのデータを新規帳票へ書き込む。このデザインパターンを複数の帳票作成のために実装して自動化を図った。一度に最大300件のエクセル帳票を自動作成することを可能とした。

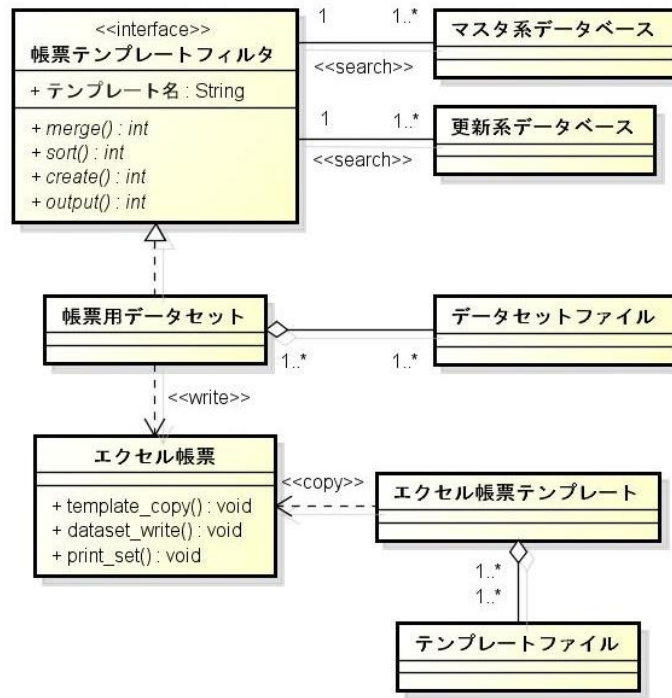


図 4 インタフェースクラスを用いたデザインパターン

3.3 非機能要求

3.3.1 緊急時対応

2018年9月は台風により事業に影響を及ぼした。該当部門の帳票は行数あふれが発生し出力ができないものがあった。帳票の行数について、直接影響の及ぶ4種類のうち、1種類は行数制限なし、3種類は行数設定3種類からの自動選択式とした。この対応でも行数あふれにより帳票を自動作成できない場合、エクセル帳票用データセットクラスの持っているデータセットファイルを用いて手作業で帳票を作成できるようにしている。

3.3.2 エラー処理

マスタファイルやスケジュール管理 CSV ファイルに何らかの異常があり帳票出力ができなかった場合、またデバッグ用として正しい処理フローを実行確認するため、ログを調べて対応ができるようにしている。

Java ではロギングの機能として準備している標準モジュール Logger クラスを使用した[12]。図5にはログファイルへメッセージを書き込む処理を示している。

エラー発生時は、エラー番号から properties ファイルの該当メッセージを抽出しログファイルへ書き込む。properties ファイルは次のように「エラーコード」と「メッセージ」で構成されている。

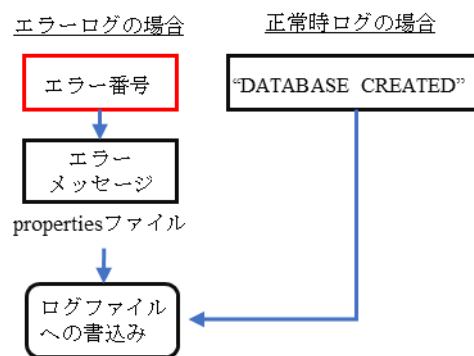


図 5 エラーメッセージの書き込み処理

M41S002=リストデータセットを作成できませんでした。[M41S002]
E6223=リストを作成できませんでした。[E6223]

Java ではログレベルは `java.util.logging.Level` クラスで定義されている[13]。本稿のシステムでは、「情報」「警告」「重大」の設定レベルを履歴としている[14]。ログは、「LEVEL」「年月日」「時間」「メッセージ」「エラー番号」「クラス」「メソッド」「メモ」をログファイルに書き込むのでエラー発生原因を調べることができる。図 6 にはログファイルを自動集計したログ内容の例を示す。

LEVEL	DATE	TIME	Message	Error Code	class	method	memo 1	memo 2	memo3
情報	2019-09-12	15:56:44.788	START-DATASET						
重大	2019-09-12	16:37:14.529	帳票リストを作成できませんでした。	E6223	LIST	createLIST			makeLIST()

図 6 ログ集計例

3.4 実装

下記に開発環境にてシステムを開発した。

- システムの種類 64 ビットオペレーティングシステム
- Windows のエディション Windows 10 Home
- Java version 9
- データベース HSQLDB 2.4.1[15]

図 4 で示した帳票作成自動化を行うデザインパターンでのエクセル帳票クラスの実装について、Java ライブラリである Apache POI [16],[17]を用いて実装している。Apache POI は Java アプリケーションから Excel や Word などの Microsoft 製品のフォーマットファイルを読み書きするための API である。Excel のファイルの読み書きでは Apache POI 中の XSSF を使用する。XSSF は Excel2007 の OOXML(.xlsx)ファイルフォーマットに対応したものを利用している。次に各メソッドの処理について説明する。

- `template_copy()`
該当帳票テンプレートから新規エクセル帳票を作成する。
- `dataset_write()`
エクセル帳票用データセットを新規エクセル帳票への書き込む。

上記のメソッドでは既存のワークブックからワークブックオブジェクトを作成するために `WorkbookFactory` クラスを用いる[18]。 `WorkbookFactory` クラスで用意されている `static` メソッドの `create` メソッドを使ってワークブックを開き、引数に指定した `InputStream` からワークブックを読み込む。読み込んだワークブックに対して、新しいシートを追加や指定したシートの行とカラムにデータを書き込むなどを行った上で、改めて保存することができる。新規エクセル帳票作成では、該当エクセル帳票テンプレートファイル(.xlsx)を既存ワークブックとして、新しいシートを追加し、新規エクセル帳票ファイル名で保存する。エクセル帳票用データセットの書き込みでは、上記で作った新規エクセル帳票ファイルを既存ワークブックとし、指定したシートの行とカラムにデータを書き込み、保存する。

- `print_set()`
新規エクセル帳票の印字設定を行う。印字領域が固定の帳票は、帳票テンプレートに印字設定を行う。また、自由な行数設定の場合は、適時印字領域を計算して設定を行う。

図 9 のように複数の作表ユニットで構成している帳票の場合、全部空白行となっているユニットを削除する。削除は行の移動で用意されている `Sheet` インター

フェースの shiftRows メソッド[19]を利用する。また、印刷設定は PrintSetup クラスを用いる[20]。

	A	B	C	D	E	F	G	H
1		↓開始カラム				↓最終カラム		
2		1	2	3	4	5	ユニット1	
3		A	B	C	D	E		
4								
5								
6		1	2	3	4	5	ユニット2	
7		K	L	M	N	O		
8		F	G	H	I	J		
9								
10		1	2	3	4	5	ユニット3	
11		P	Q	R	S	T		
12		Z	G	A	I	B		
13								
14		1	2	3	4	5	ユニット4	
15		U	V	W	X	Y		
16								
17							←シフト開始行	
18		1	2	3	4	5	ユニット5	
19								
20	空白行						←シフト最終行	
21								

図 9 エクセル帳票例

4. 第3フェーズへの移行

該当部門へ第2フェーズの情報システムを導入し、エクセル帳票作成自動化の運用を開始した。第2フェーズの情報処理を情報基盤と捉えることができるようになり、情報を管理する視点から利用する視点へ移動することができた。その結果、図10が示すように情報共有している「エクセル帳票」「CSVファイル」「マスタファイル」を再利用した事業部内社員への新たな情報サービスを提供できるようになり、第3フェーズへの移行が行われ、業務プロセスやビジネス・モデルの改善が始まった。

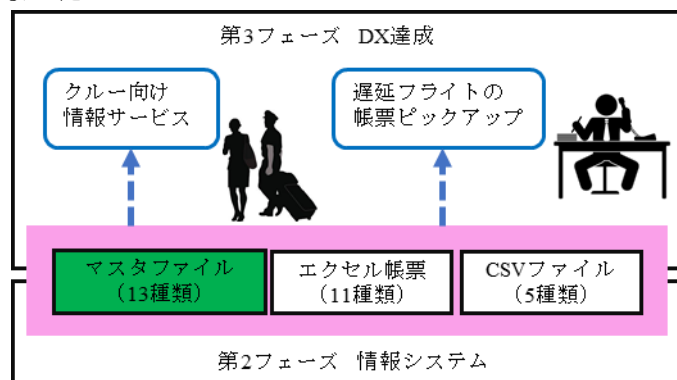


図 10 業務プロセスやビジネス・モデルの変革

5. まとめ

事業の管理部門における DX 推進は、次のような段階を経て DX を達成した。

- 第1フェーズ：手作業を中心にしたエクセル帳票作成と管理
- 第2フェーズ：コンピュータシステムによるエクセル帳票作成自動化。緊急時対応や障害発生時対応が可能。

- 第3フェーズ：第2フェーズの情報システムを情報基盤とした事業部内への新規 Web サービスを提供。

今後は、全社横断的なデータ活用に向けた取り組みに向かう。例えば、顧客満足度を改善していくために航空機の遅延予測[21]による資源準備や遅延への即時対応を実現できるように、エクセル帳票作成自動化との連携を考えていきたい。

このレポートの内容について、デジタル・トランスフォーメーションの確実な実行は所属企業のポリシーであり、本取り組みは技術責任者である筆者と業務管理責任者ならびに担当者との取り決めで行われ、フェーズ遷移の評価は技術責任者である筆者が行ったものである。

参考文献

- 1) Eric Stolterman, Anna Croon Fors: Information Technology and The Good Life, Umeo University (2004), <http://www8.informatik.umu.se/~acroon/Publikationer%20Anna/Stolterman.pdf> (2020年1月7日現在)
- 2) 【図解】 コレ一枚でわかるデジタル・トランスフォーメーション (2018), https://blogs.itmedia.co.jp/itsolutionjuku/2018/03/post_516.html (2020年1月7日現在)
- 3) 【解説】 デジタルトランスフォーメーション (2017), <https://d-marketing.yahoo.co.jp/entry/20171108477965.html> (2020年1月7日現在)
- 4) 通商産業省: DX 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～(サマリ) (2018), <https://www.meti.go.jp/press/2018/09/20180907010/20180907010-1.pdf> (2020年1月7日現在)
- 5) 通商産業省: DX 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～(PowerPoint版) (2018), <https://www.meti.go.jp/press/2018/09/20180907010/20180907010-2.pdf> (2020年1月7日現在)
- 6) 通商産業省: DX 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～(本文) (2018), <https://www.meti.go.jp/press/2018/09/20180907010/20180907010-3.pdf> (2020年1月7日現在)
- 7) 通商産業省: DX 推進ガイドライン Ver. 1.0 (2018), <https://www.meti.go.jp/press/2018/12/20181212004/20181212004.html> (2020年1月7日現在)
- 8) タスクフォースとは？ 迅速な課題解決・編成アプローチ・重要ポイント3点 (2018), <https://boxil.jp/beyond/a4793/> (2020年1月7日現在)
- 9) 経営を学ぶ ～経営学・MBA・起業～ (2019), <http://keiei-manabu.com/strategy/organizational-form.html> (2020年1月7日現在)
- 10) astah* community 6.9.0.0(2018), <http://astah.change-vision.com/ja/product/astah-community.html> (2020年1月7日現在)
- 11) デザインパターンの基本(2001), <https://www.techscore.com/tech/DesignPattern/foundation/foundation1.html/> (2020年1月7日現在)
- 12) クラス Logger – Oracle (2009), <https://docs.oracle.com/javase/jp/6/api/java/util/logging/Logger.html> (2020年1月7日現在)
- 13) java.util.logging の使い方(2017), <https://qiita.com/Qui/items/40077ce9e33738dd3914> (2020年1月7日現在)
- 14) 【Java 入門】 Logger の使い方(ログレベル, 出力先の設定も解説) (2019), <https://www.sejuku.net/blog/61048> (2020年1月7日現在)
- 15) HSQLDB の使い方 (2004), <http://wakhok.ac.jp/~tomoharu/db2004/hsqldb/index.html> (2020年1月7日現在)
- 16) Apache POI - the Java API for Microsoft Documents (2019), <https://poi.apache.org/> (2020年1月7日現在)
- 17) Apache POI で Excel を操作 (2019), <https://www.javadrive.jp/poi/> (2020年1月7日現在)
- 18) 作成済みのワークブックを開く (2019), <https://www.javadrive.jp/poi/workbook/index4.html> (2020年1月7日現在)
- 19) 行の移動 (2019), <https://www.javadrive.jp/poi/sheet/index12.html> (2020年1月7日現在)
- 20) Interface PrintSetup p(2019), <https://poi.apache.org/apidocs/dev/org/apache/poi/ss/usermodel/PrintSetup.html>

(2020年1月7日現在)

- 21) Noriko Etani: Development of a predictive model for on-time arrival flight of airliner by discovering correlation between flight and weather data, *Journal of Big Data* 2019, 6(85) (2019), <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0251-y>
(2020年1月7日現在)

江谷典子 (正会員) kerotan@kcn.ne.jp

2001年3月奈良先端科学技術大学院大学情報科学研究科博士後期課程修了 博士(工学)。現在、Peach・Aviation 株式会社勤務。人工知能、データ活用による業務改善や新規ソリューションの情報システム企画立案開発運用に従事。

投稿受付：2020年2月13日

採録決定：2020年2月14日

編集担当：斎藤彰宏 (日本アイ・ビー・エム)