

紋織物データの規格確認システムの開発

企画情報室 コンピュータ応用チーム 村山 大策, 岩崎 健太

要 旨

紋織物の業界では、データの記録メディアとしてフロッピーディスク(以下FD)の供給停止に伴いメモリー系のメディアへの転換期を迎えている。そのため、従来データ変換を行う必要がなかった製織現場でデータを扱う必要が生じたが、データに不具合があった場合、変換処理に支障を来すなど大きな問題として取り上げられるようになってきた。

そこで、紋データが規格に適合したものかどうか、また、専門知識を必要とせず、簡単に効率よく検査することができるシステムを開発し、この混乱について対応できるよう業界に提供し、使用できるよう指導することを目的とした。

パソコンを使って紋データを検査し、簡単な操作で軽微なエラーを修復できるプログラムを作成することができた。

1. はじめに

メディアとしてのFDが国内で供給停止となり、装置としてのFDドライブも製造されなくなった事は、紋織物業界全体にメディアの変更を余儀なくさせる状況となった。業界が、その経過措置として最初に選択したのは、メディアに依存しないフォーマット形式¹⁾(CGSII形式)の採用であったが、最終的には、製織現場では従来のFDを使用するためFDに依存するデータフォーマット²⁾(CGS形式)に変換して製織を行うことであった。この変換作業は、従来不要であった作業を製織工程に強いることとなり、データ処理作業が不慣れなため、正常にデータ変換ができていても大きな不安が残るうえ、エラーなどが生じたときにはお手上げという状態になり仕事が滞る状況を招くこととなった。

この様なエラーがあるデータは、当研究所へ持ち込まれば、調査分析を行い、また、不整合な点を修正するなど対応は可能だが、これを製織現場に求めることはかなり難しいと考えられる。

そこで、専門知識を必要とせず、また、専用の装置を導入することなく、製造現場などでも簡単に効率よく紋データが規格に適合したものかどうかを判定し、軽微な不整合点は修正できるツールを研究開発し、この問題解決を図ることを目的とした。

販品がないためエラーチェックの方法を研究し、それをソフトウェアとして開発することでこの目的を実現させた。

その開発については図1の流れで行った。

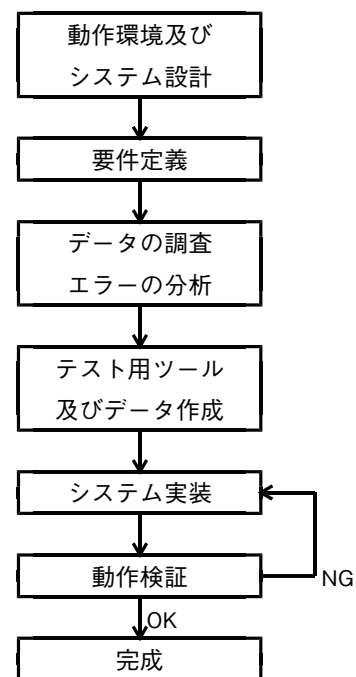


図1 システム開発の手順

2. システム設計と要件定義

紋データを、専門知識を必要とせず簡単に規格と適合しているかを判定するには、その部分をパソコンに行わせることがデータチェックを容易に行う方法と考えられる。必要なものとしてはチェック用ソフトウェアで、市

開発に必要な要件は以下の通り定義した。

①ハードウェア

装置：一般のWindowsパソコン

OSが推奨する標準仕様を満たすもの

OS : Microsoft Windows Vista 以降
※メーカーのサポートが継続されているもの

②ソフトウェア

- 1) チェック対象データは CGSII 形式とする。
- 2) データは一旦読み込み処理を行う。
- 3) 利用者の直接データ編集は極力避ける。
- 4) データは「チェックのみ」「修正を行う」の2つのモードで処理できる。
- 5) 各々のチェック内容は、記述ではなく選択で指定できるようにする。
- 6) 初心者でもボタン1つで処理できるように標準設定を設ける。
- 7) 規格を熟知した作業者が、必要に応じ検査項目を設定できるようにする。
- 8) 出力するデータは、上書き、名前を変えて保存を選択できるようにする。
- 9) データの修正は確実に修正内容を確定できるものに限る。正解を1つに特定できない場合は修正を行わない。
- 10) 使用者が入力する処理は最小限に止める。
- 11) 最大限エラー内容を明示する。

3. データの調査と分析

製織現場やデータの処理中に発見される不備のあるデータとはどのようなものか、その内容と分類を行った。

- 1) タイトルブロックに記載された内容が実際のデータと異なる。
- 2) インデックスに記載された内容が実際のデータと異なる。
- 3) 仕様 (CGSII) に規定された値の範囲でデータが記録されていない。
- 4) 仕様 (CGSII) に必要なデータが記録されずに抜けている。
- 5) 記録されたデータ自体にエラーがある。

分類としては多くないが、実際のデータの不備はこの様な少数範囲には収まらない。例えば、1)にあるタイトルブロックに記載された内容が実際のデータと異なる場合については、

- ・「柄名」は ASCII コードを使用しているか
- ・国内用に限り 2 バイト日本語コードを使用しているか
- ・そのコードは JIS コードもしくはシフト JIS コードか

・余白は 00H で満たしているか

など、1つの項目でも多数のチェックポイントが存在する。

また、「先頭レコードシリアルナンバー」は、実際のデータからその値は一意に取得できるので、整合性を確認するのは容易であるが「最終レコードシリアルナンバー」は最終レコードから取得すればその値は得られるが、最終レコード位置が特定できない場合は不定となり一意に決めることができない。

正しい規格のデータと判断するには、何の(どの)値を信じるかという問題に遭遇する。この事が解決できない限り、不備のあるデータを正しいデータへと修復を行うことは不可能である。

本研究では、この課題に対して詳細な場合分けを行い、簡易な入力というアルゴリズムの工夫と人的な支援を組み入れることでシステムに実装できるよう解決を図った。

4. 開発環境と実装、動作検証

プログラム開発の環境については以下の通り。

装置 : CPU Intel Core i5

Memory 8GB

OS : Windows 7 Professional 64bit SP1

環境 : Microsoft Visual Studio 2010

.NET Framework 4.0

言語 : Visual Basic 2010

「.NET Framework³⁾」は、Windows や Windows サーバなどのアプリを構築するための一般的な開発プラットフォームになっている。本研究用のプログラムについては、この「.NET Framework」対応アプリケーションとして開発する。完成後、このプログラムを利用するには、使用するコンピュータに「.NET Framework ランタイム」が組み込まれていることが必要となる。「.NET Framework ランタイム」は複数のバージョンがあり、OSにより対応しているバージョンが異なるので、バージョンの管理が必要となる(図2)。

「.NET Framework」の各バージョンの関係については、以下の通り。

ア) 以下のグループに分かれ、各グループのものは各々独立してインストールできる。

① .NET Framework Ver 1.0

② .NET Framework Ver 1.1

- ③ .NET Framework Ver.2.0 / 3.0 / 3.5
- ④ .NET Framework Ver.4 / 4.5.x

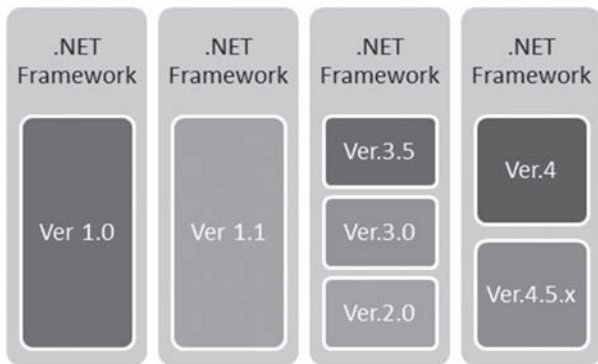


図2 .NET Framework ランタイムの種類

- イ) 各々のバージョンは個別にインストールする必要がある。
- ウ) 下位互換性は完全には確保されていない。(上位のバージョンをインストールしても、下位のバージョンが使えるわけではない)
- エ) インストールには順序があり、手順を誤るとインストールできない。基本的には下位のバージョンから先にインストールする。
- オ) Ver.2.0 / 3.0 / 3.5 のグループについては、上位の Ver 3.5 をインストールすれば、下位の互換は確保されるようになっている。具体的には、Ver3.0 をインストールすれば Ver2.0 が同時にインストールされ、Ver3.5 のインストールでは Ver3.0, Ver2.0 が自動的にインストールされ共存する形になる。
- カ) Ver.4.5.x は Ver.4 の上位互換となっている。これは、Ver.4 のコンポーネントが Ver.4.5.x に置き換えられるので、オ) の様な複数のバージョンが共存する形にはならない。

開発したプログラムは、以下の装置にインストールし動作の検証を行った。また、開発について「.Net Framework4.0」を使用しているため、プログラムを動作させるには、.Net Framework ランタイムの Ver4.0 か 4.5.x が必要になる。

《テスト用装置》

- OS : Windows7 Professional 64bit SP1
- 装置 : CPU Intel Core i5 Note PC
Memory 4GB
- 環境 : .Net Framework 4.5 ランタイム

動作検証を行うために、データに修正を加えることができるツールを作成し、エラーのない正しいデータに故意に規格外の部分を組み込み検証用データを作成した。

データに修正を加える処理については、日常の相談業務でデータエラーについて情報を得ていることや、業界一般が使用しているデータ処理ツールについて熟知していることなどから、エラーの発生状況は概ね把握できており、それら情報を元にデータを加工し、種々のエラーを含ませた検査用のデータを準備した。

5. 結果

検査用のデータは、1つのデータにつき1カ所にエラーがあるものと複数種のエラーがあるものの2つのグループに分けて準備した。前者については、各々のデータに同じ種類のエラーを複数存在させる場合もテストした。

動作については良好で、エラーを発生させたとおりに検出することができた。また、エラーを自動で修正する場合は、正解を一意に特定できるような軽微のエラーについては、想定したとおりに修正可能なことも確認できた。表1にテストを行う項目(抜粋)を示す。各々の内容について全てチェックを行いエラーの確認を行った。

表1 テスト内容(抜粋)

確認項目	処理内容
CGS 表記	CGS 表記を確認する
拡張データ形式認定番号	0000h を確認
1 柄ファイル数	0000h ならば単独ファイルモード
ファイルシリアルナンバー	単独ファイルモードの場合は 0000h を確認
最終ファイルフラグ	単独ファイルモードの場合は 01h を確認
先頭レコードシリアルナンバー	先頭のレコードシリアルナンバーと一致しているかを確認
最終レコードシリアルナンバー	最終のレコードシリアルナンバーと一致しているかを確認
レコードインデックス開始オフセット	レコードインデックスの内容を確認
ファイルインデックス開始オフセット	単独ファイルモードの場合は 00000000h を確認
データエンドマーク (EOC)	最終データ位置について確認

※レコードインデックス開始オフセットの確認については、さらにいくつかのチェック項目がある。修正可能な場合は、エラー部分を自動で修正する。

動作, 入力の画面を図3に示す。入力很简单で, チェックするファイルを選択し, 標準のエラーチェックでよければ, そのまま「実行」ボタンをクリックするだけで検査は完了する。このモード以外に「標準エラーチェック + 修正」「指定条件でエラーチェック・修正」というモードがあり, 前者はエラーチェックだけでなく修正可能なエラーならば修正まで行い, 後者は細かくチェックする条件を設定できるモードになっている。

ファイルを選択する右側に「柄確認」というボタンがあるが, これは紋データの確認用で, クリックするとデータの紋様が画面上に絵として表示されるようにした。

図3の画面は「指定条件でエラーチェック・修正」を選択したときに表示される画面で, その他のモードでは, 簡単に操作できるように動作指定する右側の入力部分を隠し左半分のみ表示するよう配慮した。

検査結果は矢印部分の画面に表示される。この画面はOSに付属するメモ帳のようになっており, 多数エラーがあってもスクロールして確認でき, 結果もそのまま名前を付けて保存できるようにした。エラーを記録できるようにしたことで, 相談を行う場合も状況確認が的確にできるようになった。

6. まとめ

本ツールにより, データに発生していた軽微なエラーは修復が可能となった。完成後, 実用化可能かどうか検証するために, 実際の作業現場(製織工程)にて利用してもらい動作状況を確認した。結果は良好で, 設計したとおり, 専門知識を必要とせず, 簡単に効率よく紋データが規格に適合したのかどうか判定でき, 使用者からも使いやすいとの意見を得た。今後は, 公開に向け画面上の文言や処理方法の修正などを行い, 設定や操作法などのマニュアルを作成して関連業界に提供できるよう準備を整える予定。

参考文献

- 1) ジャカードデータ CGSII フォーマット Ver1.0 西陣織工業組合
- 2) ダイレクトジャカード CGS フォーマット平成元年度版 西陣織工業組合
- 3) Microsoft Web サイト .NET Framework
<http://msdn.microsoft.com/ja-jp/vstudio/aa496123.aspx>

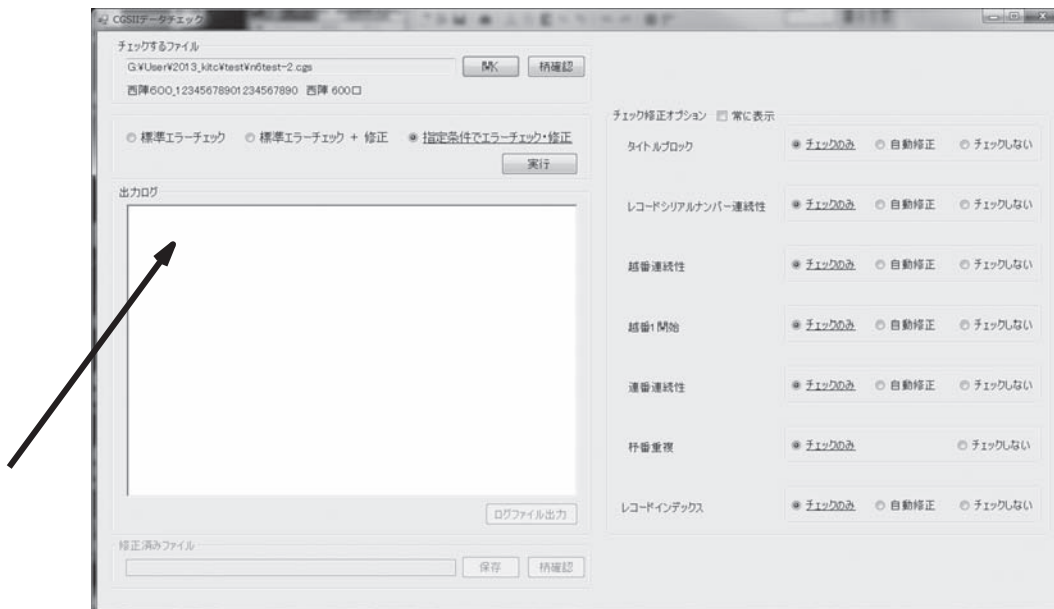


図3 設定画面