

結線ラベル作成ソフトの開発*

外山 真也*¹・樋口 誠*²・黒木 美音子*²

Development of the Label Making System for Codes of Control Box

Masaya TOYAMA, Makoto HIGUCHI and Mineko KUROKI

株式会社興電舎では、受電配電盤などを設計製作しており、電気配線図の設計を市販CADソフトを利用して行っている。現在、この配電盤の設計製作工程において、各計器間を結線するコードのラベル作成に人手と時間を要しており、この工程の省力化を実現すべく、ソフト開発を実施した。このことについて報告する。

キーワード：CAD、DXFデータ

1 はじめに

株式会社興電舎では、受電配電盤などを設計製作しており、電気配線図の設計を市販CADソフトを利用して行っている。現在、この配電盤の設計製作工程において、各計器間を結線するコードのラベル作成に人手と時間を要しており、この工程の省力化について相談を受けた。

結線ラベルは、図1に示すような、作成された配線図(この例で示すものは、非常に簡単なシーケンス配線図である)を見て、各計器の端子部分に接続する位置を明確にするため、コードの名称と接続部の記号をラベルとして、専用のラベル出力機械に入力し作成している。なお、入力データは、表計算ソフト「EXCEL」のCSVデータでも受け取られることが分かっている。

このような状況において、市販CADで作成された配線図をDXFデータで取り込み、そのデータから文字列データのみを抽出することにより、工程の省力化を提案した。

その結果、シーケンス図を見ながらコードラベルを手入力してラベル作成を行っていたために約1時間を要していた従来のラベル作成の作業時間を、5分程度に大きく低減することができた。

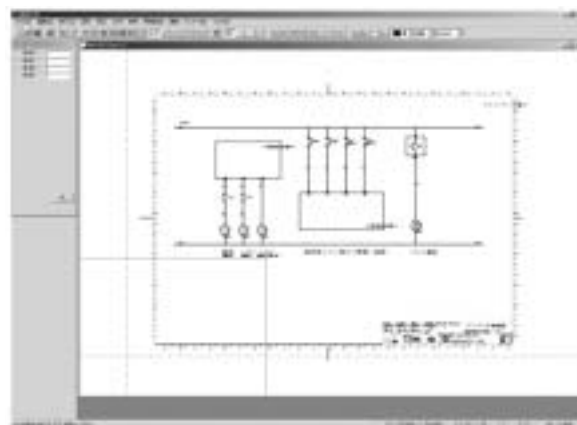


図1 シーケンス図

```
using System;
using System.Collections;

using ClsMtcAMLib;

namespace PickUpText
{
    public class ClsLabelText
    {
        public String strLabel = "";
        public int Layer=0;
        public int Count=0;

        public ClsLabelText()
        {
            Init();
        }

        public void Init() {
            strLabel = "";
            Layer =0;
            Count =0;
        }
    }
}
```

図2 ラベル文字データのクラス化

* 共同研究

* 1 機械電子・デザイン部

* 2 株式会社 興電舎 情報システム課

2 開発方法

最初に、配線図を描いたCADのDXFデータを受け取り、このデータを解析して文字列のデータのみを抽出した。この機能の開発は、既にTOM-CADなどのCAD/CAM開発で蓄積したコマンドがあるので、比較的容易に実現できた。

次に、その文字列データからレイヤに関する情報を取り出し、レイヤ、個数、文字列のデータとして整理した。

そして、それらのデータをリストアップ表示し、不要なデータの削除、あるいは編集や修正ができるように、各種機能のコマンドを開発した。

以下に、その具体的な手法について述べる。

2-1 クラス化

最終的に必要なデータは、ラベル文字とその個数であるが、配線図に描かれているデータには、ラベルとして不要なデータも多く、それら不要データをすべて削除する作業は容易ではない。そこで、ラベル作成に必要なデータか否かでレイヤ毎に分類し、図面を描くよう提案した。

そのため、ラベル作成用文字データの構造を図2に示すとおりクラス化(ClsLabelText)し、ラベル文字、レイヤ、個数の三つのデータを有するものとした。

次に、このラベル文字データの配列クラスを作成し、配列データとして取り扱えるようにした。このプログラムを図3に示す。これを実施することにより、ラベル文字を配列として扱うことが可能となった。



図3 ラベル文字の配列クラス

2-2 プログラム開発

上記の方法によりプログラムを開発した。図4に起動初期状態を示す。データを読み込む場合は、「Ref」(ファイル参照ボタン)をクリックするとファイルオープンダイアログが表示され、希望のフォルダを検索し、DXFファイルを選択することが可能である。図4では、その操作を行い、データファイルを参照している状態となっている。

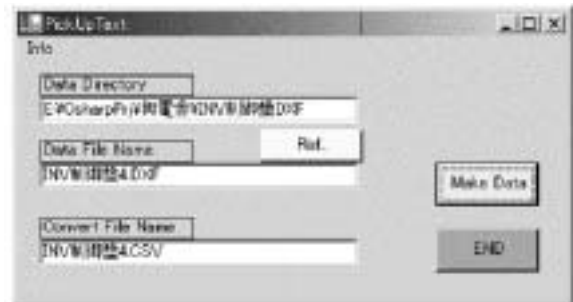


図4 起動初期状態

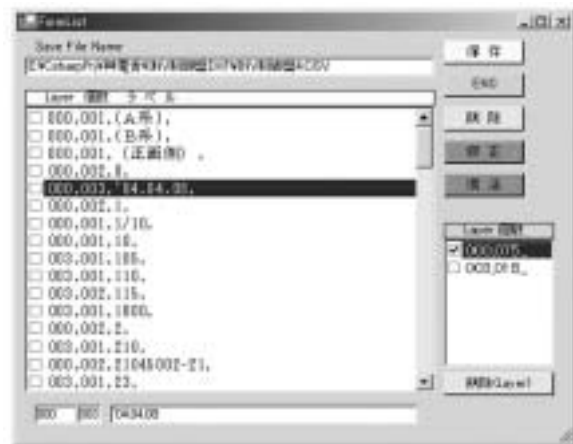


図5 読み込まれた文字データ

ここで、データ作成ボタン「Make Data」をクリックすると、図5に示すように、DXFデータから文字データが読み込まれ、リスト表示される。このとき、データは「レイヤ」、「文字データ」の順に整列させ、右下部にレイヤと、そのレイヤでのデータ数を表示するようにした。この機能によって、文字データのレイヤの分類数、レイヤ毎のデータ数の把握が容易になった。

また、レイヤのリスト表示部分のチェックボックスを「ON」にして、その下にあるレイヤ単位でのデータの一括削除「削除(レイヤ)」ボタンをクリックすると、チェックされたレイヤのデータ

をすべて削除することができる。この機能によって、不要なデータの一つずつ選択して削除するという面倒な処理を省力化できた。

さらに、一度削除されたデータを復活させる機能を図6に示す。これは、先のレイヤ単位、もしくは個別の削除によって処理されたデータを復活させるためのものである。プログラムは、このデータリスト表示のFormとプログラムを再帰的に利用して実現している。



図6 削除されたデータの復活処理

データは、配列データクラスを二つ用意し、一つは「正」のデータ、もう一つは削除されたデータを扱うようにし、「削除」や「復活」のコマンド操作において、指定されたデータを「正」から「削除」へ、もしくは「削除」から「正」の配列データへ移動させることにより実現している。図7に作成されたラベルデータを、図8にそのデータを利用してラベルを作成している様子を示す。

3 結果および考察

従来は、配線図面を見て、人手によってラベル文字を打ち込んでいた。しかし、このソフトを利用することにより、従来約1時間を要していた作業を、5分程度で処理できるようになり、入力ミスや作業者の負担も大きく軽減されたことが報告されている。

今回のDXFデータの読み込みに関する機能の開発においては、これまでに実施してきた二次元CAD/CAMや生産工程の合理化に関する各種



図7 保存されたラベル文字データ



図8 ラベル文字打ち出しの様子

ソフトウェアの開発で蓄積されてきた技術をそのまま利用することができた。そのため、開発期間を1ヶ月程度に大幅に短縮することができた。

4 まとめ

今回の開発において、結線コードラベルの作成工程の大幅な省力化が実現した。

株式会社興電舎では、市販の二次元CADを利用しているが、DXFデータを有効利用しての省力化の可能性については気づいていなかった。

しかし、この共同研究開発により、設計工程の標準化および省力化の可能性について、積極的に検討を実施することとなった。

今後、開発した各種コマンドを整理し、より活用しやすいライブラリを構築したい。