広島工業大学紀要研究編
第45巻（2011）pp.245-251
論　　文

Ruby/Tk による電力系統図作成支援システムに関する研究

竹本　直史*・永田　武**

（平成22年10月28日受付）

A Power System Drawing Assistance System by Using Ruby/Tk

Naofumi TAKEMOTO and Takesi NAGATA

(Received Oct. 28, 2010)

Abstract

In recent years the term “smart grid” is emerging in the media. The smart grid delivers electricity from suppliers to consumers using two-way digital technology to control appliances at consumers’ homes to save energy, reduce cost and increase reliability and transparency. In order to operate the smart grid properly, the power system analysis is necessary. The analysis of smart grid is required the power system model. However the configuration of the smart grid is frequently changed. To cope with this situation, power system drawing assistance systems are required.

This paper proposes a power system drawing assistance system by using Ruby/Tk. Ruby is a scripting language, in the recent tradition of Perl, Python, and Tcl. Ruby originated in Japan during the mid-1990s and was initially developed and designed by Yukihiro Matsumoto. It allows for a rapid development cycle and the rapid prototyping of applications. In order to the effectiveness of the proposed system, a prototype system is developed. The results show the proposed system is promising the assistance of the power system engineers.

Key Words: power system drawing assistance, Ruby, Tk, low-carbon society, smart grid, power system

1．はじめに

人工のシステムの中で最大規模であるといわれている電力系統は，電力化率の上昇に伴って益々その複雑さが増している。さらに，最近では低炭素社会構築に向けた対策として，“新三種の神器”と呼ばれる“再生可能エネルギー”，“環境対応車”そして“省エネルギー”が注目を浴びている。特に，再生可能エネルギーである太陽光発電の需要地近郊への導入が促進され，わが国の計画では2020年の太陽光発電導入量が，2009年の約10倍である2,800万kWとなっている。このような分散電源の配電系統への大量導入は，既存の電力系統において電圧問題や需給不平衡などの様々な問題を引き起こすことが予想されている。

最近の電力系統では，“スマートグリッド”という新しい概念が出現し，情報通信技術と電力エネルギー技術の融合が進んでおり，今後そのスマートグリッド単位での計画・運用・制御方式の開発が重要になってくると考えられる。したがって，スマートグリッドの運用に携わる様々な技術者が，熟練した技術者と同レベルの解析を実行できる環境が望まれている。そのスマートグリッドを構成する要素（分散電源や需要家）は，上位の系統要素に比較して頻繁に追加削除が発生し，それに対応するために電力系統図の容易

* 広島工業大学大学院工学系研究科情報システム科学専攻
** 広島工業大学情報学部情報工学科

な作成方法が望まれている。

そこで，本研究では，本来検討すべきスマートグリッド内の計画・運用・制御の検討に注力できるように系統モデルを容易に構築できるRuby/Tkを用いた「電力系統図作成支援システム」の開発を行う。

2．電力系統図作成支援システム

これまでにも電力系統図作成支援システムについての研究がなされている。これまでの方法は，全ての要素の自動配置と自動配線を指向した方法[1,2]，既存プログラムの有効利用を図るために解析プログラム毎にAPIを作成する方法[3]，そしてインターネットを利用した方法[4]の３種類に大別できる。文献（1）は，VLSIの配線設計で使用されている混雑度の概念に基づく最短経路探索法を利用することで，ブランチの交差数ができるだけ少ない系統図の自動作成を行っている。文献（2）は，系統図としての描画基準を定めグラフ描画に遺伝的アルゴリズムとスプリングモデルを使用し，基準に適用したノード自動配置を行う自動描画機能を備えた系統図描画支援システムを提案している。文献（3）は，オブジェクト指向データベースと解析目的に適したGUIを用意し解析データの入力，プログラムの実行，解析結果表示までを一貫的に支援するシステムの構築を行っている。また，文献（4）は著者らが提案した方法でJavaアプレットを用いた教育システムを指向したものである。

しかしながら，これらの方法は簡単な単線図モデルを対象とするシステムか，あるいはオブジェクトデータベースを用いた大規模なシステムとなっている。したがって，本研究では，電力系統の一部を構成するスマートグリッドを扱うモデル化に適した実モデルでの系統図作成を行う[5,6]。

3．Rubyと Ruby/Tkについて

本研究では，開発言語としてRuby/Tkを採用する。以下にその概要を述べる。

3.1　Ruby

Rubyは1993年にまつもとゆきひろ氏によって開発されたフリーソフトウェアのオブジェクト指向スクリプト言語である[7]。機能として，クラス定義，ガベージコレクション，プロセス操作，データ構造，強力な正規表言処理，マルチスレッド，例外処理，イテレータ・クロージャ，Mixin，演算子オーバーロード，ネットワーク，GUIなどがある。またRubyは周辺技術も多く，開発環境が充実している。Rubyの特徴として下記のようなものがある。

・シンプルな文法　　Rubyはオブジェクト指向に基づいた一貫した文法を持っている。また，非常にシンプルかつ美しくプログラムを作成できる。シンプルで一貫性のとれたプログラムのため，可読性やメンテナンス性が非常に高くなっている。

・**動的言語**　　動的言語の特徴的な性質として，変数や式に型情報をつけないことである。しかし，これは型情報がないのではなく，データ自体が型情報を持っているためプログラム上で宣言を必要としない。また，型チェックなどはプログラム実行時に行われる。

・スクリプト言語　　コンパイルを必要とせず即実行できるため，開発効率が良い。

・オブジェクト指向　　Rubyは整数などの基本的なデータ型をはじめとしたすべての値がオブジェクトであり，メッセージを送る統一的なインタフェースを持つ。それ故に，値によって特別な扱いをする必要がなく，ユーザが考慮する必要のある事柄がそれだけ少なくて済む。

・GUIプログラミングが容易　　TkやGtkなどの強力なGUIツールキットへのインターファイルが提供されているのでGUIプログラミングの作成が容易である。

・移植性が高い　　Rubyは移植性を考慮してC言語で書かれているため，大抵のUNIX系OSで動作できる。DOS，Windows，Mac，BeOSなどのOSでも動作が可能である。

一方，Rubyの大きな短所として処理時間が遅いことであるが，本研究の系統図作成では問題とはならない。

3.2　Ruby/Tk

Ruby/Tkは，Tcl/Tkに直接リンクし，コマンド文字列の低レベルインタフェースを，Rubyで定義したクラスライブラリで包み込んだものとなっている。この方法によって，Tclインタプリタのインタフェース仕様が変わらない限り，Tcl/Tkの最新版が即座に利用可能となっている。Ruby/Tkには下記のような特徴がある。

・強力で簡潔なGUI機能　　Tkの優れた機能を活用でき，Tcl/Tkの豊富な拡張ライブラリも利用可能である。

・Rubyとの高親和性　　Tk部品をTcl/Tkのスクリプトではなく Rubyの手続きオブジェクトで与えることが可能である。

・マルチスレッド対応　　ユーザがGUI操作をしている間に並列処理を続行可能である。

4．Ruby/Tkを用いた電力系統図作成支援システム

本研究でRubyを取り上げたのは，前章で述べたように非常に簡潔にプログラムの作成が可能であり，プログラム作成の負担が削減できるためである。この削減効果をみるために，１つのボタンからなる簡単なプログラムを

Ruby/Tk と JavaSwing で記述し，そのソースコードの量を比較した。図１にプログラムの比較を示す。同図に示すように Ruby/Tk の方が Java に比べ 1/2 程のステップ数となっていることが分かる。

Javaのプログラム

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class JButtonTest extends JFrame implements
ActionListener {
        JButtonTest() {
                getContentPane().setLayout(new
FlowLayout());

                JButton b1 = new JButton("OK");
                b1.addActionListener(this);
                getContentPane().add(b1);

                setDefaultCloseOperation
(JFrame.EXIT_ON_CLOSE);
                setTitle("JButtonTest");
                setSize(300, 200);
                setVisible(true);
        }
        public void actionPerformed(ActionEvent
e) {
                System.out.println("OK");
        }
        public static void main(String[] args) {
                new JButtonTest();
        }
}
```

Ruby/Tkのプログラム

```
require "tk"
f1 = TkRoot.new{title "Ruby/Tk Button"}
f1.geometry("300x200")
b = TkButton.new(f1,'text'=>'OK','command'=>proc
{puts "OK"},
                "width"=>10,'height'=>2,'font'=>8).pack
('pady'=>20)
b = TkButton.new(f1,'text'=>'EXIT','command'
=>proc{exit},
                "width"=>10,'height'=>2,'font'=>8).pack
('pady'=>20)
Tk.mainloop
```

図１　Ruby/Tk と JavaSwing の比較

4.1　提案システムの概要

提案するシステムの全体構成を図２に示す。同図に示すように，提案システムは系統図作成機能と保存機能の２つの機能を有する。以下にそれらの機能について説明する。

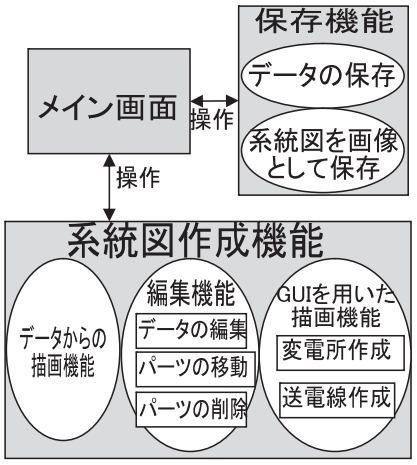


図２　提案システムの構成図

4.2　電力系統の表現方法

提案システムでの電力系統の表現方法を図３に示す。変電所や配電用変電所の母線と調相設備はノード，変圧器と開閉器（しゃ断器，断路器）をブランチとする。一方，送電線は区間をブランチとし，分岐をノードとする。また，ブランチとノードは以下のような設備コードを付けている。

X	X	–	Y	Y	Y	–	Z	Z	Z

ここで，XX：ブランチ / ノード ID，YYY：変電所番号，ZZZ：ブランチ／ノード番号である。ブランチ ID は，変圧器：TR，しゃ断器：CB，断路器：LS，区間：LN とする。ノード ID は，変電所：ND，送電線：NN とする。

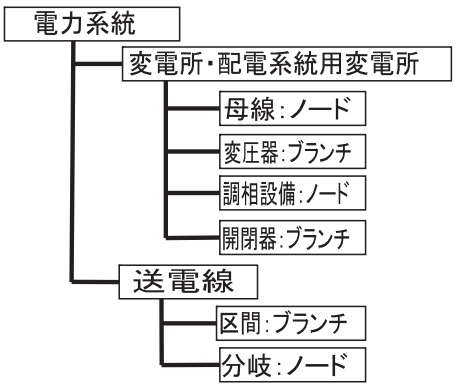


図３　電力系統の表現方法

4.3　系統図作成機能

系統図作成機能は，GUI を用いた描画機能（変電所・送電線・ブランチデータの端子設定作成）と，データからの自動描画機能，そしてデータの編集機能の３つからなる。

(1) GUI を用いた描画機能

変電所の描画は，予め登録した変電所タイプを選択し，その変電所内のデータを入力することにより行う。その後，メイン画面上の任意の箇所に描画する。また，同時に自動で変電所内のブランチデータの端子設定を行う。

ブランチデータの端子設定はシステム上，変電所の設置後に行う。また，図４はブランチ端子設定の処理順路を示す。以下に，端子設定の条件と手順について述べる。

〔条件〕

・母線の１次側から順にノード番号を与える。
・ブランチとブランチの間にノードが存在する。
・ブランチは始端終端側のノード情報を持つ。
・スタート位置を上流側とする。

〔手順〕

・変電所の最初に作成されたブランチから処理する。
・対象ブランチは接続する上流ノードと下流ノードを取得する。また，取得したノードが上流ならば始端，下流ならば終端にノードデータを格納する。
・同様の処理を変電所内の全てのブランチに行う。
・処理順路は原則スタート位置から左向きに行う。

送電線の描画は，始端位置と終端位置を決め，その間に送電線を描画する。クリックする事で始端位置決定し，ダブルクリックする事で終端位置を決定する。また，始端位

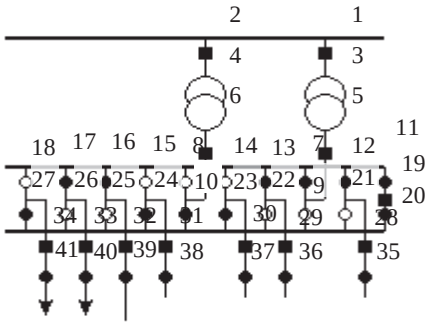


図４　ブランチ端子の探索順路

置決定後に，クリックする事で中間点が作成でき送電線を折れ線で描画できる。

(2) 定義データの編集機能

定義データはブランチデータ，ノードデータ，ステータスデータの三種類からなる。其々の内容は以下の通りである。

・ブランチデータ：ブランチ名，始端，終端，抵抗，リアクタンス，静電容量，タップ
・ノードデータ:ノード名，タイプ，有効電力，無効電力，電圧の大きさ，電圧の位相角
・ステータスデータ：ブランチ / ノード名，状態（開閉器の ON/OFF，充停電状態）

定義データの編集では，ブランチ名・ノード名・始端・終端値を除く値を変更できる。また，作成した変電所及び送電線の移動と削除が可能である。

内部データとして，ブランチデータの抵抗，リアクタンス，静電容量の初期値は 0.0 に，タップは 1.0 に設定されている。ノードデータのタイプの初期値は Null に設定されている。有効電力，無効電力，電圧の大きさ，電圧の位相角は 0.0 に設定されている。ステータスデータのブランチの状態は１次側・２次側によって自動で ON/OFF を設定する。ON/OFF 状態の表示方法は ON を黒，OFF を白とする。ステータスデータの充停電状態は充電を CHARGED，停電を UNCHARGED で記述する。初期値は CHARGED に設定する。

(3) データからの自動描画機能

データからの自動描画機能は以下の手順で行う。

①変電所毎にブランチをまとめ，各変電所内の開閉器数・変圧器数を求める。
②変電所データから変電所座標と SC/ShR 数，端子データから端子位置と端子数を取得する。
③各変電所はブランチ端子より自身に接続する送電線データを取得する。
④各取得データをもとに変電所の配置を行う。

4.4　保存機能

保存機能はデータの保存と画像データの保存の２つからなる。データの保存ではブランチ, ノード, ステータスデータの保存と自動描画機能に必要となる変電所データと端子データを保存される。変電所データには変電所の識別番号と変電所タイプ，端子データには変電所の識別番号と端子数と向きが保存される。画像データの保存は作成した電力系統図を ESP 形式で保存する。

５．Ruby/Tk を用いた電力系統図作成支援システム

開発した電力系統図作成支援システムの系統図作成機能の実行例を本章で述べる。図５にメイン画面を示す。同図左側を設置画面と呼び，ここに電力系統図を作成する。また，同図の右側に拡大した動作ボタンを示す。

図５　メイン画面

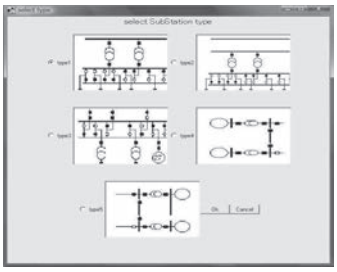
5.1　変電所の作成方法

変電所の実行例を図６に示す。動作ボタン欄から“SS_Create”を選択，その後，同図（a）から，予め登録されている変電所タイプを選択，同図（b）に変電所の識別番号と母線・しゃ断器・断路器・変圧器・SC/ShR 個数を入力後，同図（c）に変電所の接続端子数と位置を入力する。そして同図（d）のクリックした位置に変電所を描画する。また，同図（e）より変電所の選択とパラメーターの変更により，任意の構成が作成可能である。

作成された変電所の詳細データは変更することができる。変更したい変電所内のパーツ（しゃ断器・断路器・変圧器）をクリックすることで図７のウィンドウが表示される。このウィンドウでブランチデータの抵抗・リアクタンス・静電容量・タップを変更できる。また，開閉器の場合は開閉器の ON/OFF を変更できる。開閉器の状態を変更した場合，変電所の表示が更新される。

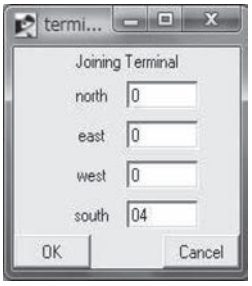
5.2 送電の作成方法

送電線の実行例を図８に示す。動作ボタンから“Line_

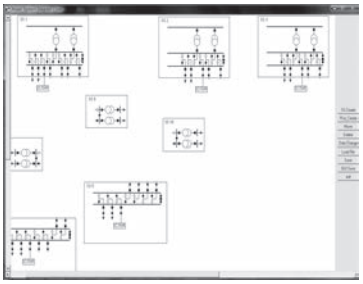


（a）“select type”

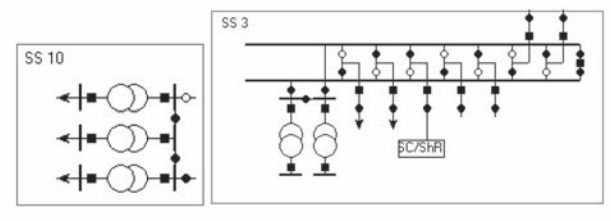
（b）“substation data input”



（c）“terminal input”



（d）“Power System Diagram Creat”



（e）パラメーター変更時の変電所作成例

図６　変電所の実行例

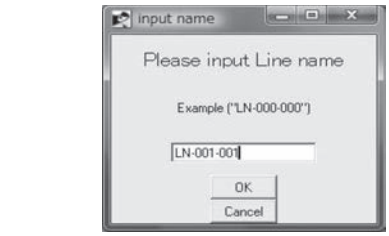
図７　変電所内のデータ変更

Create”を選択すると図８(a)が表示される。同図より“input name”に送電線の識別番号を入力後，設置画面上をクリックして始端位置を決め，ダブルクリックで終端位置を決める。送電線は始端終端間に描画される。また，始端位置の決定後に１クリックすることで中間点を作成し，送電線を折れ線で描画できる。ブランチデータの端子の設定は始端位置と接続するものと終端位置に接続するものを自動で取得する。

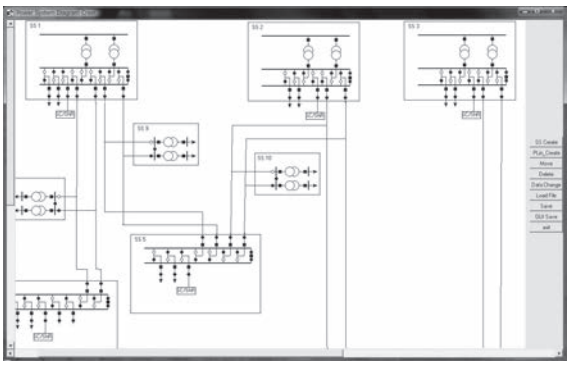
5.3　定義データの編集機能

(1) ブランチ

定義データの編集機能のモデル変電所を図９に示し，編集機能の実行例を図 10 に示す。同図(a)よりブランチデータ，ノードデータ，ステータスデータを選択した後，データを見たい変電所を選択，OK を押下すると同図（b）を表示する。同図（b）は図９のブランチデータを示す。表示されたブランチデータにおいて，変更したい行をクリックすると同図（c）が表示され，抵抗，リアクタンス，静電容量，タップの数値を変更することが出来る。変更値を入力後，OK を押下すると内容を更新する。



（a）“input name”



（b）“Power System Diagram Creat”

図８　送電線の実行例

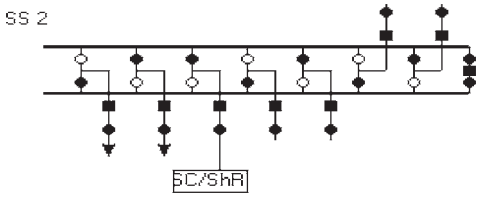


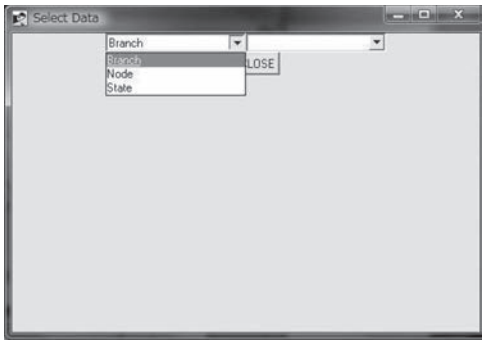
図９　モデル変電所

(2) ノード

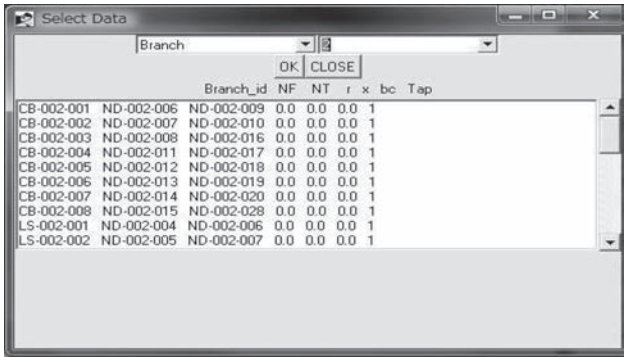
図 11 は図９のノードデータを示す。データ内容を変更する場合は，上記と同様に変更する行をクリックする。すると入力ウィンドウが表示され，ノードタイプ，有効電力，無効電力，電圧の大きさ，電圧の位相角の値を変更することが出来る。

(3) ステータス

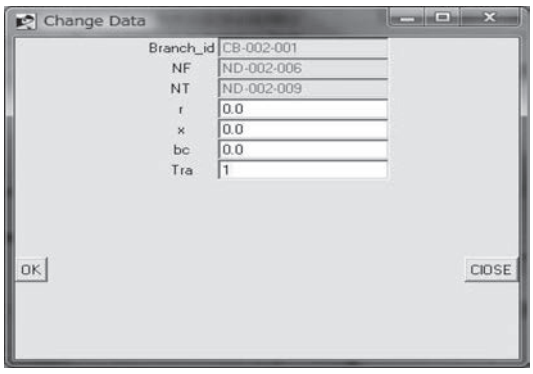
図 12 は図９のステータスデータを示す。データの変更方法は上記と同じであり，状態を変更できる。



（a）データ選択



（b）ブランチデータリスト



（c）ブランチデータ変更画面

図 10　定義データの編集機能の実行例（ブランチ）

6．おわりに

本論文では，Ruby/Tk を用いた電力系統図作成支援システムを開発した。開発システムの特長を以下に示す。

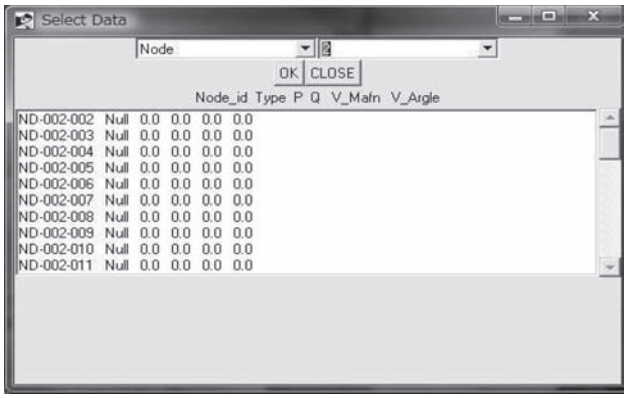
(1) 変電所は標準部品の選択とパラメーターの変更により任意の変電所の作成が可能である。

(2) 送電線は変電所の端子番号と接続させることにより，任意の構成を可能とする。

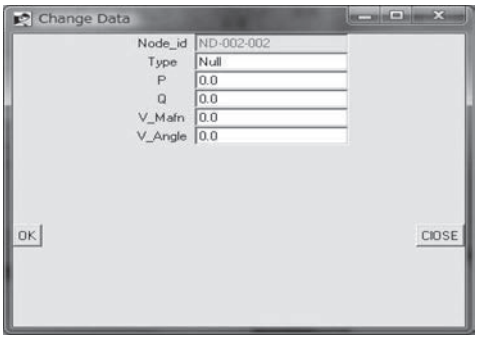
(3) 完成した電力系統は定義ファイルとして保存し，解析等での再利用が可能である。

また，７変電所３送電線の電力系統図（内部データを含む）の作成時間は４分程度であった。この結果から電力系統図作成の時間短縮ができていることが分かる。

今後の課題は，系統解析ソフトウェアとの連携と，Web アプリケーションへの拡張が挙げられる。

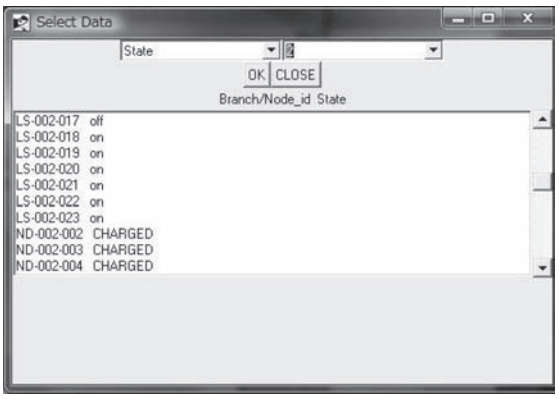


（a）ノードデータ



（b）ノードデータ変更

図 11　定義データの編集機能の実行例（ノード）



（a）ステータスデータ

（b）ステータスデータ変更

図 12　定義データの編集機能の実行例（ステータス）

文　献

1）佐々木，歌谷，竹田，渡辺，三崎：「電力系統図自動作成システムの開発」，電学論Ｂ，116，1，pp.35-41（1996）

2）川原，造賀，佐々木：「GA による系統図自動描画支援システムの構築」，電学論Ｂ，123，8，pp.927-934（2003）

3）佐々木，歌谷，久保川，川原：「統合的電力系統解析支援システムの構築」，電学論Ｂ，115，7，pp.734-740（1995）

4）永田，渡部，大野，佐々木：「WWW 環境を用いた電力系統解析のための教育支援システムの試作」，電学論Ｂ，120，2，pp.128-133（2000）

5）竹本，永田：「Ruby/Tk を用いた電力系統図作成支援システムの開発」，電気学会 C 部門全国大会，pp1704（2010）

6）竹本，永田：「Ruby/Tk による電力系統図作成ツールの開発」，電気･情報関連学会中国支部連合大会（2010）

7）Ruby 公式ホームページ
http://www.ruby-lang.org/ja/about/（2010/10/25）