

明 細 書

発明の名称：プログラム編集用プログラム

技術分野

[0001] この発明は、制御装置のプログラム編集用プログラムに関するものである。

背景技術

[0002] F A (Factory Automation) システムを構成するプログラマブルコントローラなどの制御装置で実行させるプログラムは、一般的にリレー制御回路の考えを基本とした、リレーシンボリック語による回路図形式（以下、ラダープログラムという）で図形を用いて記述される。また、このほかにも F B (Function Block)、F B D (Function Block Diagram)、S F C (Sequential Function Chart) などの形式で図形を用いて記述されることもある。このような図形を用いて記述されるプログラムの作成は、情報処理端末で、エンジニアリングツールと呼ばれるプログラム編集用プログラムを用いて、グラフィカルに行われている。

[0003] 一般的な制御装置のエンジニアリングツールでは、図形を用いて記述されるプログラムはマウスとキーボードを用いて編集されている。たとえば、ラダープログラムに回路記号あるいは接続線などのプログラム構成要素を追加する場合は、プログラムを編集するための編集画面において、マウスまたはキーボードを用いて、プログラム構成要素を追加する位置を指定した後に、追加したいプログラム構成要素を選択する、という処理を繰り返して、プログラムの編集を可能にしている。また、プログラム構成要素を追加する以外の操作を行う場合も、操作の対象を指定し、実行したい処理（以下、コマンドという）を選択する、という処理を繰り返して、プログラムの編集を可能にしている。

[0004] 一方、近年では、タッチパネル入力機能を有する情報処理装置（以下、タブレット端末という）を使った入力技術が盛んになってきている。このタブ

レット端末にて、エンジニアリングツールを利用する場合には、タッチパネルに指を当てる操作を、マウスカーソルにて操作の対象を指定してマウスボタンを押す操作と同等に扱って、プログラムの編集を可能にする方法が一般的な方法となっている。

先行技術文献

特許文献

[0005] 特許文献1：特開2009-259080号公報

発明の概要

発明が解決しようとする課題

[0006] ところで、タブレット端末にて、エンジニアリングツールを利用する場合には、上記した従来の一般的な方法を用いると、プログラム構成要素の配置先を指定し、プログラム構成要素の種類を指定する操作を、指を用いて繰り返すことになるため、マウスで操作する場合と比べて手の移動距離が大きくなり、操作が煩雑になるという課題があった。また、指がマウスポインタより大きいために、プログラム編集画面上に、プログラム構成要素の種類あるいはコマンドを指定するための選択肢の一覧を大きく表示する必要があり、マウスで操作する場合と比べて、画面に表示できる情報を減らす必要が生じるという課題があった。

[0007] この発明は上記に鑑みてなされたもので、図形を用いて記述されるプログラムを、タブレット端末を用いて容易に編集できるプログラム編集用プログラムを得ることを目的とする。

課題を解決するための手段

[0008] 上記目的を達成するため、この発明にかかるプログラム編集用プログラムは、プログラム編集画面上で図形を用いて記述されるプログラムを編集するプログラム編集用プログラムにおいて、コンピュータをプログラム編集画面上に入力される座標情報、前記座標の軌跡、前記軌跡の向きを含む入力ジェスチャ特性情報を取得する入力ジェスチャ特性情報取得手段と、入力ジェス

チャ特性情報に基づいて、操作対象を特定するとともに、前記操作対象に適用するコマンドも特定して、前記コマンドを実行し、プログラム編集画面に表示する処理実行手段と、して機能する。

発明の効果

[0009] この発明によれば、図形を用いて記述されるプログラムのプログラム編集画面にユーザが指を当てたまま動かす入力操作から得られる入力ジェスチャ特性情報を取得し、入力ジェスチャ特性情報に基づいて、操作の対象を特定するとともに、その操作対象に適用するコマンドを特定して、そのコマンドを実行してプログラム編集画面に表示するようにしたので、タブレット端末にて、指をプログラム編集画面に当てたまま動かす入力操作を用いて、操作の対象（以下、操作対象という）の指定と、その操作対象に対して行うコマンドの指定と、を同時に行うことができ、図形を用いて記述されるプログラムの編集を容易に行うことができるという効果を有する。

図面の簡単な説明

[0010] [図1]図1は、実施の形態1によるプログラム編集用プログラムの構成の一例を示すブロック図である。

[図2]図2は、ジェスチャ定義の一例を示す図である。

[図3]図3は、ジェスチャ定義の一例を示す図である。

[図4]図4は、実施の形態1によるプログラム編集方法の処理手順の一例を示すフローチャートである。

[図5]図5は、実施の形態1によるプログラム編集処理の手順の一例を示す図である。

[図6]図6は、実施の形態1によるプログラム編集処理の手順の一例を示す図である。

[図7]図7は、実施の形態2によるジェスチャ定義の一例を示す図である。

[図8]図8は、実施の形態2によるプログラム編集方法の手順の一例を示すフローチャートである。

[図9]図9は、実施の形態2によるプログラム編集の処理手順の一例を示す図

である。

[図10]図10は、実施の形態2によるプログラム編集の処理手順の一例を示す図である。

[図11]図11は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図12]図12は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図13]図13は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図14]図14は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図15]図15は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図16]図16は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図17]図17は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図18]図18は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図19]図19は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図20]図20は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図21]図21は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図22]図22は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[図23]図23は、実施の形態2による具体的なプログラム編集処理の手順の

一例を示す図である。

[図24]図 2 4 は、実施の形態 2 による具体的なプログラム編集処理の手順の一例を示す図である。

[図25]図 2 5 は、実施の形態 2 による具体的なプログラム編集処理の手順の一例を示す図である。

[図26]図 2 6 は、実施の形態 2 による具体的なプログラム編集処理の手順の一例を示す図である。

[図27]図 2 7 は、実施の形態 2 による具体的なプログラム編集処理の手順の一例を示す図である。

[図28]図 2 8 は、実施の形態 2 による具体的なプログラム編集処理の手順の一例を示す図である。

[図29]図 2 9 は、実施の形態 2 による具体的なプログラム編集処理の手順の一例を示す図である。

[図30]図 3 0 は、一般的なエンジニアリングツールにおけるプログラム編集手順を模式的に示す図である。

[図31]図 3 1 は、実施の形態 3 による領域管理情報の概念を模式的に示す図である。

[図32]図 3 2 は、実施の形態 3 によるラダープログラムにおける文字入力領域の区分の一例を示す図である。

[図33]図 3 3 は、実施の形態 3 によるジェスチャ定義の一例を示す図である。

[図34]図 3 4 は、実施の形態 3 によるプログラム構成要素が何も配置されていないプログラム編集画面でのプログラム編集方法の処理手順の一例を示すフローチャートである。

[図35]図 3 5 は、実施の形態 3 によるプログラム編集処理の手順の一例を示す図である。

[図36]図 3 6 は、実施の形態 3 によるプログラム編集処理の手順の一例を示す図である。

[図37]図37は、実施の形態3によるプログラム構成要素が配置されている場合のプログラム編集画面でのプログラム編集方法の処理手順の一例を示すフローチャートである。

[図38]図38は、実施の形態3によるプログラム編集の処理手順の一例を示す図である。

[図39]図39は、実施の形態3によるプログラム編集の処理手順の一例を示す図である。

[図40]図40は、実施の形態3によるプログラム編集の処理手順の一例を示す図である。

[図41]図41は、実施の形態3によるプログラム編集の処理手順の一例を示す図である。

[図42]図42は、実施の形態3によるプログラム編集処理の手順の一例を示す図である。

[図43]図43は、実施の形態3によるプログラム編集処理の手順の一例を示す図である。

[図44]図44は、実施の形態3による軌跡を描く入力操作による文字入力時のガイド表示の一例を示す図である。

[図45]図45は、プログラム編集画面の一例を示す図である。

[図46]図46は、実施の形態4によるジェスチャ定義の一例を示す図である。

[図47]図47は、実施の形態5によるプログラムの編集処理の手順の一例を示す図である。

[図48]図48は、実施の形態6による具体的なプログラムの編集処理の手順の一例を示す図である。

[図49]図49は、実施の形態7によるプログラム構成要素の移動条件の一例を示す図である。

[図50]図50は、実施の形態7によるプログラム構成要素の移動方法の一例を示す図である。

[図51]図5 1は、実施の形態7によるプログラム構成要素の移動方法の一例を示す図である。

[図52]図5 2は、実施の形態7によるプログラム構成要素の移動方法の一例を示す図である。

[図53]図5 3は、実施の形態7によるプログラムの編集の処理手順の一例を示す図である。

[図54]図5 4は、実施の形態8によるプログラムの編集の処理手順の一例を示す図である。

[図55]図5 5は、実施の形態8によるプログラムの編集の処理手順の一例を示す図である。

[図56]図5 6は、実施の形態8によるプログラムの編集の処理手順の一例を示す図である。

[図57]図5 7は、実施の形態9によるプログラム編集用プログラムが行う処理の手順の一例を示すフローチャートである。

[図58]図5 8は、実施の形態9によるテストモード状態における処理の手順の一例を示す図である。

[図59]図5 9は、実施の形態9によるテストモード状態における処理の手順の一例を示す図である。

[図60]図6 0は、実施の形態10によるヘルプ情報表示の処理手順の一例を示す図である。

[図61]図6 1は、実施の形態11によるプログラム編集用プログラムの機能構成の一例を模式的に示すブロック図である。

[図62]図6 2は、実施の形態11によるデバイス名入力処理の手順の一例を示す図である。

[図63]図6 3は、入力操作から入力ジェスチャ特性情報の座標情報を取得する方法の他の例を示す図である。

発明を実施するための形態

[0011] 以下に添付図面を参照して、この発明にかかるプログラム編集用プログラ

ムの好適な実施の形態を詳細に説明する。なお、以下では、ラダープログラムを編集する場合を例に挙げて説明するが、これらの実施の形態によりこの発明が限定されるものではない。

[0012] 実施の形態 1.

図 1 は、実施の形態 1 によるプログラム編集用プログラムの構成の一例を示すブロック図である。プログラム編集用プログラム 10 は、入力ジェスチャ特性情報取得手段 11 と、処理実行手段 12 と、を備える。

[0013] 入力ジェスチャ特性情報取得手段 11 は、指などの入力指示体を介してプログラム編集画面上に入力される座標情報、前記座標の軌跡、前記軌跡の向きから成る入力ジェスチャ特性情報を取得する。座標情報としては、プログラム編集画面に定義した座標系に基づいて得られる座標を用いることができる。

[0014] 処理実行手段 12 は、入力ジェスチャ特性情報に含まれる座標情報を、プログラム編集画面の構成要素の座標と照合して操作対象を特定するとともに、入力ジェスチャ特性情報に含まれる座標の軌跡とその軌跡の向きを、前記操作対象に関するジェスチャ定義に含まれる軌跡の形状と向きの情報と照合して前記操作対象に適用するコマンドも特定して、そのコマンドを実行し、プログラム編集画面に表示する。

[0015] プログラム編集画面の構成要素は少なくとも、a 接点を表す図形、b 接点を表す図形、コイルを表す図形、図形と図形を接続する接続線などのプログラム構成要素、および、それ以外の領域を構成する下地要素からなる。下地要素としては、左母線の左側に配置される左側下地要素、右母線の右側に配置される右側下地要素、左母線と右母線の間に配置される図形配置可能下地要素がある。

[0016] ジェスチャ定義は、前記構成要素の種類と、軌跡の形状と向きの情報と、実行するコマンドを対応付けたものである。図 2 と図 3 は、ジェスチャ定義の一例を示す図である。ここでは、タイル状の構成要素をプログラム編集画面に敷き詰めて、タイルの種類で軌跡の経過を把握する方法について例示し

ている。図2（a）は、あるタイル状のマス目を中心として、水平方向または垂直方向に2マス用いて定義する場合のタイル状のマス目の配置関係を示す図であり、図2（b）は、図2（a）の場合のジェスチャ定義の一例を示す図である。また、図3（a）は、あるタイル状のマス目を中心として、水平方向、垂直方向または斜め方向に直線状の軌跡を2マスを用いて定義する場合のタイル状のマス目の配置関係を示す図であり、図3（b）は、図3（a）の場合のジェスチャ定義の一例を示す図である。

[0017] この例では、ジェスチャ定義は、入力操作の軌跡の形状とその向きと、構成要素と、その構成要素に適用するコマンドと、を対応付けたものである。ジェスチャの軌跡は、上記したようにプログラム編集画面に定義された座標系に基づいて得られる座標の軌跡となっている。また、この軌跡の描画に対して、ジェスチャ描画領域に描かれるプログラム構成要素が定義されている。軌跡の形状の定義は、軌跡を判別できる範囲で最小限の点で表現してよい。すなわち、たとえば、軌跡の形状として、直線のみを用いる場合は、2つのマス目を用いて定義できる。

[0018] これに直線以外の形状を加える場合は、3つ以上のマス目を用いて定義できる。2つのマス目で定義する場合は、軌跡は始点から終点までの向きで判別され、2つ以上のマス目を用いて定義する場合は、軌跡の形状は、始点から終点に至るまでに経由したマス目の座標で判別され、軌跡の向きは、始点から終点に至るまでに経由したマス目の順序で判別される。

[0019] ジェスチャ定義のマス目は、物理的な寸法を用いて定義してもよいが、物理的な特定の寸法を持たない論理的な位置関係のみを定義してもよい。図2、図3に示したジェスチャ定義は、論理的な位置関係を定義した場合を表している。

[0020] なお、ここでは矩形のマス目で説明したが、たとえば正六角形など、ほかの形状のマス目を用いてもよい。

[0021] つぎに、プログラム編集方法について説明する。図4は、実施の形態1によるプログラム編集方法の処理手順の一例を示すフローチャートである。ま

ず、入力ジェスチャ特性情報取得手段 11 は、ユーザによってプログラム編集画面に入力される入力ジェスチャ特性情報を取得する（ステップ S 11）。入力ジェスチャ特性情報は、入力指示体の接触位置である座標情報、座標の軌跡、および軌跡の向きからなる。

[0022] ついで、処理実行手段 12 は、入力ジェスチャ特性情報に格納されている座標情報に基づいて操作対象を特定し（ステップ S 12）、入力ジェスチャ特性情報に対応する軌跡の形状とその向きがその操作対象のジェスチャ定義に存在するかを判定する（ステップ S 12）。入力ジェスチャ特性情報に対応する軌跡の形状とその向きがジェスチャ定義にある場合（ステップ S 13 で Yes の場合）には、処理実行手段 12 は、ジェスチャ定義に定義されているコマンドを実行し（ステップ S 14）、プログラム編集画面の情報を更新する（ステップ S 15）。以上によって、プログラムの編集処理が終了する。

[0023] また、ステップ S 13 で入力ジェスチャ特性情報に対応する軌跡がジェスチャ定義にない場合（ステップ S 13 で No の場合）には、コマンドの実行は不要と判定して（ステップ S 16）、処理が終了する。

[0024] なお、処理実行手段 12 は、入力指示体を用いた入力操作が完了する前から処理を開始してもよい。また、そのとき、入力操作が完了するまでコマンドの実行を保留するとともに、入力操作が完了したときに実行される予定のコマンドをプログラム編集画面に表示するように構成してもよい。また、このとき、入力操作が完了したときにコマンドが適用される操作対象を特定できる情報をプログラム編集画面に表示するように構成してもよい。

[0025] 以下に、プログラム作成処理の具体例について説明する。図 5 と図 6 は、実施の形態 1 によるプログラム作成処理の手順の一例を示す図である。図 5 は、左母線 101 と、右母線 102 と、これらの母線間を結ぶ接続線 111 と、を配置する入力操作を示している。まず、プログラム構成要素が何も配置されていないプログラム編集画面上に、図 5(a) に示されるように、ユーザがプログラム編集画面に入力指示体の先端を当て（F 1）、先端を画面に

当てたまま画面の右方向に入力指示体を移動させ（F 2）、ある点で先端を画面から離す（F 3）。図では、画面に触れた先端の軌跡が点線の矢印によって示されている。

[0026] 入力ジェスチャ特性情報取得手段 1 1 は入力操作 J から入力ジェスチャ特性情報を取得し、処理実行手段 1 2 はこの入力ジェスチャ特性情報に対応する操作対象およびコマンドをジェスチャ定義から取得する。

[0027] すなわち、はじめに、入力ジェスチャ特性情報取得手段 1 1 は、入力操作 J の F 1, F 2, F 3 の座標情報および座標の軌跡、並びに F 1 に始まって F 3 で終わる軌跡の向きを取得する。

[0028] 次に、処理実行手段 1 2 は、入力ジェスチャ特性情報取得手段 1 1 が取得した入力ジェスチャ特性情報に含まれる座標情報に基づいて、操作対象を特定する。この例では、図 5（a）は母線を表示する前の状態を表しており、プログラム編集画面は全ての領域が図形配置可能下地要素のみで構成されているので、操作対象は図形配置可能下地要素と特定される。

[0029] 次に、処理実行手段 1 2 は、特定された操作対象、すなわちこの例では図形配置可能下地要素に関するジェスチャ定義に含まれる軌跡の形状とその向きと、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きを照合し、図形配置可能下地要素に適用するコマンドを特定する。すなわちこの例では、入力ジェスチャ特性情報に含まれる軌跡が水平方向の直線であって、その軌跡の向きは右向きであるため、水平方向の接続線を追加するコマンドが実行される。このとき、母線はまだ表示される前であるため、プログラム編集画面に、左母線、右母線および左母線と右母線をつなぐ接続線を表示して、処理を終了する。

[0030] その結果、図 5（b）に示されるように、プログラム編集画面の入力操作を行った付近には、左母線 1 0 1 と、右母線 1 0 2 と、これらを結ぶ接続線 1 1 1 と、が描かれる。

[0031] 図 6 は、左母線 1 0 1 と、右母線 1 0 2 と、a 接点記号 1 2 1 およびコイル記号 1 3 1 を有する 2 つの母線 1 0 1, 1 0 2 間を結ぶ接続線 1 1 1 と、

を配置するジェスチャ動作を示している。図6（a）に示されるように、プログラム編集画面に入力指示体の先端を当て（F1）、先端をプログラム編集画面に当てたままプログラム編集画面の下方方向に指を移動させ（F2）、ある点で先端をプログラム編集画面から離す（F3）。入力ジェスチャ特性情報取得手段11は入力操作Jから入力ジェスチャ特性情報を取得し、処理実行手段12はこの入力ジェスチャ特性情報に対応する操作対象およびコマンドをジェスチャ定義から取得する。その結果、図6（b）に示されるように、ジェスチャ描画領域付近には、左母線101と、右母線102と、これらを結ぶ接続線111が描かれるとともに、接続線111上の左側の領域にa接点記号121が配置され、接続線111上の右側の領域にコイル記号131が配置される。これは、ラダープログラムではよく使われる基本的なプログラムであり、このようによく使われるプログラムを生成するコマンドを、ジェスチャ定義を用いて入力操作に関連付けることで、プログラムの編集工程数を減らすことができる。

[0032] これらは一例であり、ジェスチャと、プログラム構成要素との組み合わせは任意に設定することができる。

[0033] たとえば、図5では、入力操作Jに図5（b）に示した構成要素を追加するコマンドを組み合わせたが、図5（b）の代わりに図6（b）に示した構成要素を追加するコマンドを組み合わせてもよい。

[0034] なお、以上では、入力ジェスチャ特性情報に入力される座標情報は、入力指示体がプログラム編集画面に接触している間入力されるという前提で説明したが、タブレット端末のプログラム編集画面が、入力指示体がプログラム編集画面に接触している間に加えて、入力指示体がプログラム編集画面に近接している間も入力指示体を検出するように構成されている場合は、入力ジェスチャ特性情報には、入力指示体とプログラム編集画面が接触しているか、接触せずに近接しているか、またはまったく離れているかといった近接状態情報を加えて、前記近接状態情報に基づいて、入力指示体とプログラム編集画面が接触している場合と、接触していないが近接している場合を区別し

て、実行するコマンドを区別できるようにしてもよい。

[0035] また、図4のステップS13でN oの場合に、近傍の別の操作対象について改めてステップS12およびステップS13を行ってもよい。

[0036] さらに、入力終了からコマンドが実行されるまでの時間を短縮するために、入力ジェスチャ特性情報取得手段11が行う処理と、処理実行手段12が行う処理を並行して行ってもよい。つまり、入力ジェスチャ特性情報取得手段11が入力ジェスチャ特性情報を取得している最中、すなわち軌跡の入力が終了する前に、処理実行手段12が操作対象およびコマンドの特定を開始するように構成してもよい。

[0037] また、ジェスチャ検出中に、図5(c)に示されるように、入力指示体とプログラム編集画面の接触位置の軌跡を表示するようにしてもよい。軌跡の表示の仕方として、たとえば所定の時間ごとに検出した位置を示すマークTを配置することができる。こうすることで、入力がどのように受け付けられているかを確認しやすくなるという効果を奏する。

[0038] 実施の形態1では、何も配置されていないプログラム編集画面にユーザが入力指示体で入力操作を行うことで、その入力操作に対応するプログラム構成要素を、プログラム編集画面に配置するようにした。

[0039] これによって、操作対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なうことができるようになるため、タブレット端末において、図形を用いて記述されるプログラムをより少ない入力操作労力で編集できるという効果をする。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0040] また、ジェスチャ定義に、軌跡として、プログラムに加えたい変化と相似している図形を定義することで、ユーザにとって、入力操作に用いる軌跡の記憶が容易になり、より少ない期間で操作方法を習得できるという効果も有する。

[0041] 実施の形態 2.

実施の形態 1 では、何も配置されていないプログラム編集画面に入力指示体にて入力操作を行ってラダープログラムを編集する場合について説明した。ラダープログラムでは、一般的に左母線と右母線との間に記号を配置し、左母線、記号および右母線の間を接続線で結ぶことによって、プログラミングを行う。そのため、実施の形態 2 では、プログラム編集画面に左母線と右母線とが配置された状態で、入力指示体を用いる入力操作によってラダープログラムを編集する場合について説明する。

[0042] 実施の形態 2 のプログラム編集用プログラムでは、ジェスチャ定義は、軌跡が通過した構成要素に関する情報をさらに含み、処理実行手段 12 は、軌跡が通過した構成要素を判定してコマンドを特定する機能をさらに有する。

[0043] 図 7 は、実施の形態 2 によるジェスチャ定義の一例を示す図である。この図 7 は、図 3 (a) のように、あるタイル状のマス目を中心として、水平方向、垂直方向または斜め方向に 2 マス用いて定義する場合のジェスチャ定義を示している。この図に示されるように、ジェスチャ定義では、プログラム構成要素に対して、入力操作の軌跡と、その軌跡が通過する構成要素と、が対応付けられている。これによって、同一の軌跡を複数のプログラム構成要素に対応付けることが可能になる。

[0044] 図 8 は、実施の形態 2 によるプログラム編集方法の手順の一例を示すフローチャートである。まず、入力ジェスチャ特性情報取得手段 11 は、プログラム編集画面に入力される入力ジェスチャ特性情報を取得する（ステップ S31）。入力ジェスチャ特性情報は、入力指示体の接触位置である座標情報、座標の軌跡、および軌跡の向きからなる。

[0045] ついで、処理実行手段 12 は、入力ジェスチャ特性情報に対応する操作対象を特定する（ステップ S32）。これは、たとえば、入力ジェスチャ特性情報に格納されている座標情報と、処理実行手段 12 が構成要素をプログラム編集画面に表示する際に構成要素の配置先として指定した座標情報を比較することによって取得することができる。

- [0046] なお、ここでは、入力操作の軌跡の中央付近に存在する構成要素を操作対象として特定する例を示すが、ほかの特定方法を用いてもよい。たとえば、入力操作の軌跡の始点に在る構成要素を操作対象として特定するように構成してもよい。あるいは、入力操作の軌跡の終点に在る構成要素を操作対象として特定するように構成してもよい。あるいは、入力操作の軌跡の重心に在る構成要素を操作対象として特定してもよい。
- [0047] 操作対象が下地要素の場合には、処理実行手段 12 は、入力ジェスチャ特性情報に格納されている軌跡の形状とその向きがその下地要素に関するジェスチャ定義に存在するかを判定する（ステップ S 33）。入力ジェスチャ特性情報に格納されている軌跡の形状とその向きに対応するコマンドがジェスチャ定義に存在する場合（ステップ S 33 で Yes の場合）には、処理実行手段 12 は、そのコマンドを実行し（ステップ S 34）、プログラム構成要素をプログラム編集画面に表示する（ステップ S 35）。以上によって、プログラムの編集処理が終了する。
- [0048] すなわち、ジェスチャが左母線と右母線を横切るものであれば、ステップ S 32 にて下地要素が操作対象であると判定し、ジェスチャ描画領域内の左母線と右母線の間に、ジェスチャ定義で定義されるプログラム構成要素が配置される。また、ジェスチャが接続線を跨ぐものである場合には、ステップ S 33 にてプログラム構成要素の一つである接続線が操作対象であると判定し、ジェスチャ描画領域の接続線が、ジェスチャ定義のコマンドに従って別のプログラム構成要素によって置換される。
- [0049] なお、入力操作が描く軌跡が、a 接点などの回路記号のほかに、その記号に接続される接続線の一部を横断する場合もある。この場合には、回路記号と接続線の 2 つが操作対象の候補に含まれることになるが、接続線を除く記号を優先して取得するような設定を設けてもよい。その後、ステップ S 35 のプログラム構成要素をプログラム編集画面に表示する処理を行う。
- [0050] ステップ S 33 で入力ジェスチャ特性情報に対応する軌跡の形状とその向きに対応するコマンドが操作対象に関するジェスチャ定義にない場合（ステ

ップS 3 3でN oの場合) には、該当するコマンドが無いと判断し (ステップS 3 6)、処理が終了する。なお、認識エラーとなったときに、エラーメッセージを表示してもよいし、エラーメッセージを表示しないように構成してもよい。

[0051] 以下に、(1) プログラム構成要素が配置されていないプログラム編集画面に入力操作を行う場合、(2) 既にプログラム構成要素が配置されたプログラム編集画面に入力操作を行う場合、についてプログラム編集処理の具体例を説明する。

[0052] (1) プログラム構成要素が配置されていないプログラム編集画面に入力操作を行う場合

図9～図10は、実施の形態2によるプログラム編集の処理手順の一例を示す図である。図9(a)のように、左母線101と右母線102のみが配置されたプログラム編集画面200Aに、図9(b)のように、ユーザが入力指示体で、左母線101の左側を始点とし、右方向に移動しながら右母線102の右側の領域で終点となるように(領域L→領域M→領域Rと画面をなぞるように)入力操作Jを入力する。

[0053] すなわち、はじめに、入力ジェスチャ特性情報取得手段11は、入力操作JのF1、F2、F3の座標情報および座標の軌跡、並びにF1に始まってF3で終わる軌跡の向きを取得する。

[0054] ついで、処理実行手段12は、入力ジェスチャ特性情報取得手段11が取得した入力ジェスチャ特性情報に含まれる座標情報に基づいて、操作対象を特定する。この例では、操作は左側下地要素(L)を始点として(F1)、図形配置可能下地要素(M)を経由して(F2)、右側下地要素(R)で入力を終了している(F3)ことから、操作対象は図形配置可能下地要素と特定される。

[0055] その後、処理実行手段12は、特定された操作対象、すなわちこの例では図形配置可能下地要素に関するジェスチャ定義に含まれる軌跡の形状とその向きと、入力ジェスチャ特性情報に含まれる入力操作の軌跡の形状とその向

きを照合し、図形配置可能下地要素に適用するコマンドを特定する。すなわちこの例では、入力ジェスチャ特性情報に含まれる軌跡が水平方向の直線であって、その軌跡の向きは右向きであるため、水平方向の接続線を追加するコマンドが実行される。

[0056] これによって、図9（c）のように、プログラム編集画面200には、左母線101と右母線102との間を接続する接続線111が配置される。

[0057] また、図9（b）に示される入力操作Jを入力することによって、図10に示されるように、左母線101と右母線102との間を接続する接続線111と、接続線111の左母線101側にa接点記号121と、接続線111の右母線102側にコイル記号131と、が配置されるようにしてもよい。

[0058] これは、ラダープログラムではよく使われる基本的なプログラムであり、このようによく使われるプログラムを生成するコマンドを、ジェスチャ定義を用いて入力操作に関連付けることで、接続線111のみを配置する場合に比してプログラムの編集工程数をさらに減らすことができる。

[0059] （2）既にプログラム構成要素が配置されたプログラム編集画面に入力操作を行う場合

図11～図29は、実施の形態2による具体的なプログラム編集処理の手順の一例を示す図である。

[0060] （2-1）接続線に対して構成要素を配置する例

図11～図15は、接続線を跨ぐようにジェスチャを描く場合の例を示している。図11では、（a）のように接続線111の上側の領域から下側の領域に向かって入力操作Jを行うと、（b）のようにその位置に接続線の代わりにa接点記号121が配置される場合が示されている。また、図12では、（a）のように接続線111の下側の領域から上側の領域に向かって入力操作Jを行うと、（b）のようにその位置に接続線の代わりにb接点記号122が配置される場合が示されている。このように軌跡の形状が同一の入力操作でしかも操作対象が同じでも、軌跡の向きが異なれば異なるコマンド

を実行し、その結果、異なるプログラム構成要素に置き換える編集操作を行うことが可能になる。

[0061] 図 13 では、(a) のように図形配置可能下地要素中の接続線 111 の上側を始点とし、下側の領域に向かって半円弧を描きながら、接続線 111 の下側が終点となるように入力操作 J を行う。

[0062] すなわち、はじめに、入力ジェスチャ特性情報取得手段 11 は、入力操作 J の F1, F2, F3 の座標情報および座標の軌跡、並びに F1 に始まって F3 で終わる軌跡の向きを取得する。

[0063] ついで、処理実行手段 12 は、入力ジェスチャ特性情報取得手段 11 が取得した入力ジェスチャ特性情報に含まれる座標情報に基づいて、操作対象を特定する。この例では、接続線 111 の上側の領域を始点として (F1)、接続線 111 を経由して (F2)、接続線 111 の下側の領域で入力を終了している (F3) ことから、操作対象はプログラム構成要素である接続線 111 と特定される。

[0064] その後、処理実行手段 12 は、特定された操作対象、すなわちこの例では接続線 111 に関するジェスチャ定義に含まれる軌跡の形状とその向きと、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きを照合し、接続線 111 に適用するコマンドを特定する。すなわちこの例では、入力ジェスチャ特性情報に含まれる軌跡が上下方向の半円弧状の曲線であって、その軌跡の向きは下向きであるため、コイル記号 131 を追加するコマンドが実行される。

[0065] これによって、図 13 (b) のように、入力操作 J が行われた位置にコイル記号 131 が配置される。

[0066] また、図 14 では、(a) のように接続線 111 の右上側の領域で接続線 111 の上側を始点とし、接続線 111 に平行に移動した後、接続線 111 の下側の領域に向かって約 90 度向きを変えて接続線 111 を跨いだ後、接続線 111 の下側が終点となるように入力操作 J を行う。

[0067] すなわち、はじめに、入力ジェスチャ特性情報取得手段 11 は、入力操作

J の F 1 , F 2 , F 3 の座標情報および座標の軌跡、並びに F 1 に始まって F 3 で終わる軌跡の向きを取得する。

[0068] ついで、処理実行手段 1 2 は、入力ジェスチャ特性情報取得手段 1 1 が取得した入力ジェスチャ特性情報に含まれる座標情報に基づいて、操作対象を特定する。この例では、接続線 1 1 1 の上側の領域を始点として (F 1) 、接続線 1 1 1 を経由して (F 2) 、接続線 1 1 1 の下側の領域で入力を終了している (F 3) ことから、操作対象はプログラム構成要素である接続線 1 1 1 と特定される。

[0069] その後、処理実行手段 1 2 は、特定された操作対象、すなわちこの例では接続線 1 1 1 に関するジェスチャ定義に含まれる軌跡の形状とその向きと、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きと座標情報を照合し、接続線 1 1 1 に適用するコマンドを特定する。すなわちこの例では、入力ジェスチャ特性情報に含まれる軌跡が、接続線 1 1 1 に平行方向の直線と、接続線 1 1 1 に垂直方向の直線とを組み合わせたものであるため、命令行 1 3 2 を追加するコマンドが実行される。

[0070] これによって、図 1 4 (b) のように、入力操作 J が行われた位置に命令行 1 3 2 が配置される。配置された時点では、この命令行 1 3 2 には命令が何も入力されていない状態にある。

[0071] 上述した例では、入力指示体を 1 本使用する場合を示したが、2 本以上同時にプログラム編集画面に触れて行う入力操作に対しても、コマンドを割り当ててもよい。図 1 5 では、(a) のように、a 接点記号 1 2 1 a とコイル記号 1 3 1 との間で、2 本の入力指示体を接続線 1 1 1 の上側を始点とし、下側の領域に向かって直線的に移動した後、接続線 1 1 1 の下側が終点となるように入力操作 J を行う。

[0072] すなわち、はじめに、入力ジェスチャ特性情報取得手段 1 1 は、入力操作 J の F 1 , F 2 , F 3 の座標情報および座標の軌跡、並びに F 1 に始まって F 3 で終わる軌跡の向きを取得する。

[0073] ついで、処理実行手段 1 2 は、入力ジェスチャ特性情報取得手段 1 1 が取

得した入力ジェスチャ特性情報に含まれる座標情報に基づいて、操作対象を特定する。この例では、入力操作は、接続線 1 1 1 の上側の領域を始点として（F 1）、接続線 1 1 1 を経由して（F 2）、接続線 1 1 1 の下側の領域で入力を終了している（F 3）ことから、操作対象はプログラム構成要素である接続線 1 1 1 と特定される。

[0074] その後、処理実行手段 1 2 は、特定された操作対象、すなわちこの例では接続線 1 1 1 に関するジェスチャ定義に含まれる軌跡の形状とその向きと、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きを照合し、接続線 1 1 1 に適用するコマンドを特定する。すなわちこの例では、入力ジェスチャ特性情報に含まれる軌跡が 2 本の上下方向の直線であって、その軌跡の向きは下向きであるため、a 接点記号 1 2 1 b を追加するコマンドが実行される。

[0075] これによって、図 1 5（b）のように、入力操作 J が行われた位置に a 接点記号 1 2 1 b が配置される。

[0076] 以上、図 1 1 ～図 1 5 にて説明したように、接続線 1 1 1 を横切る入力操作 J を行なうことで、接続線に関するジェスチャ定義に基づいて入力ジェスチャ特性情報に対応するコマンドが実行されて、別のプログラム構成要素に置き換えられる。

[0077] すなわち、操作対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なうことができるようになるため、タブレット端末において、図形を用いて記述されるプログラムを、より少ない労力で編集できるという効果を奏する。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0078] （2－2）記号を変更する例

図 1 6 では、（a）のように a 接点記号 1 2 1 の右上の領域から左下の領域に向かって入力操作 J を行う。

- [0079] すなわち、はじめに、入力ジェスチャ特性情報取得手段 11 は、入力操作 J の F 1, F 2, F 3 の座標情報および座標の軌跡、並びに F 1 に始まって F 3 で終わる軌跡の向きを取得する。
- [0080] ついで、処理実行手段 12 は、入力ジェスチャ特性情報取得手段 11 が取得した入力ジェスチャ特性情報に含まれる座標情報に基づいて、操作対象を特定する。この例では、入力操作は、接続線 111 の上側の領域を始点として (F 1)、a 接点記号 121 を経由して (F 2)、接続線 111 の下側の領域で入力を終了している (F 3) ことから、操作対象はプログラム構成要素である a 接点記号 121 と特定される。
- [0081] その後、処理実行手段 12 は、特定された操作対象、すなわちこの例では a 接点記号 121 に関するジェスチャ定義に含まれる軌跡の形状とその向きと、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きを照合し、a 接点記号 121 に適用するコマンドを特定する。すなわちこの例では、入力ジェスチャ特性情報に含まれる軌跡が斜め方向の直線であって、その軌跡の向きは左下向きであるため、a 接点記号 121 を b 接点記号 122 に置き換えるコマンドが実行される。
- [0082] これによって、図 16 (b) のように、入力操作 J が行われた位置に b 接点記号 122 が配置される。
- [0083] 図 17 は、図 16 の場合と逆である。すなわち (a) のように b 接点記号 122 の右上の領域から左下の領域に向かって入力操作 J を行くと、(b) のように a 接点記号 121 が配置される。この処理の詳細は、図 16 の場合と略同様であるので、その説明を省略する。
- [0084] また、図 18 では、(a) のように a 接点記号 121 の上側の領域から下側の領域に向かって入力操作 J を行くと、(b) のように立ち下がり a 接点記号 123 が配置される。この処理の詳細は、図 16 の場合と略同様であるので、その説明を省略する。
- [0085] 以上、図 16 ~ 図 18 にて説明したように、プログラム構成要素を横切る入力操作 J を行なうことによって、プログラム構成要素に関するジェスチャ

定義に基づいて入力ジェスチャ特性情報に対応するコマンドが実行されて、別のプログラム構成要素に置き換えられる。

[0086] すなわち、操作対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なうことができるようになるため、タブレット端末において、図形を用いて記述されるプログラムを、より少ない労力で編集できるという効果を奏する。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0087] (2-3) 並列回路を作成する例

図19では、(a)のように、領域Lを始点として(F1)、領域Mに向かって水平方向に所定の距離を移動した後、上に向かって約90度方向転換し、接続線111を経由して(F2)、接続線111の上側の領域まで移動させる(F3)ように入力操作Jを行う。

[0088] すなわち、はじめに、入力ジェスチャ特性情報取得手段11は、入力操作JのF1、F2、F3の座標情報および座標の軌跡、並びにF1に始まってF3で終わる軌跡の向きを取得する。

[0089] ついで、処理実行手段12は、入力ジェスチャ特性情報取得手段11が取得した入力ジェスチャ特性情報に含まれる座標情報に基づいて、操作対象を特定する。この例では、入力操作は、接続線111の下側の領域を始点として(F1)、左母線101と図形配置可能下地要素内の接続線111を経由して(F2)、接続線111の上側の領域で入力を終了している(F3)ことから、操作対象は図形配置可能下地要素と特定される。

[0090] その後、処理実行手段12は、特定された操作対象、すなわちこの例では図形配置可能下地要素に関するジェスチャ定義に含まれる軌跡の形状とその向きと、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きを照合し、図形配置可能下地要素に適用するコマンドを特定する。すなわちこの例では、入力ジェスチャ特性情報に含まれる水平な直線と垂直な直線の組み合わせ

せであって、その軌跡の向きは左から右方向に向かい、途中から上向きに変わるものであり、その軌跡は接続線 1 1 1 を横切る。その結果、(b) のように入力操作を行った位置に a 接点記号 1 2 1 b、左母線 1 0 1 と a 接点記号 1 2 1 b を結ぶ接続線 1 1 2 a、および a 接点記号 1 2 1 b と接続線 1 1 1 を結ぶ接続線 1 1 2 b を有する並列回路が配置される。

[0091] なお、(c) に示されるように、接続線 1 1 1 の上側の領域から下側の領域に向かって所定の距離を移動した後、領域 L に向かって約 90 度方向転換し、左母線 1 0 1 の左側の領域 L まで移動させる入力操作 J に対応して、(b) の並列回路が配置されるようにしてもよい。図 19 は a 接点記号の並列回路であったが、同様にして図 20 に示されるようにコイル記号 1 3 1 a, 1 3 1 b の並列回路を配置することもできる。

[0092] 図 19 と図 20 では、左母線 1 0 1 または右母線 1 0 2 に接続する並列回路を作成する例を示したが、接続線 1 1 1 の途中で分岐するような並列回路とすることも可能である。図 21 では、(a) のように接続線 1 1 1 に左から 2 つの a 接点記号 1 2 1 a, 1 2 1 b と 1 つのコイル記号 1 3 1 が配置されている。このうち、a 接点記号 1 2 1 a、1 2 1 b の間に在る接続線 1 1 1 の上側の領域を始点として、a 接点記号 1 2 1 b を囲むように U 字型の入力操作 J を行う。その結果、(b) のように、入力操作を行った位置に a 接点記号 1 2 1 c が接続線 1 1 1 から分岐されて、a 接点記号 1 2 1 b に並列して配置される。この処理の詳細は、図 19 の場合と略同様であるので、その説明を省略する。

[0093] 以上、図 19 と図 20 にて説明したように、ラダープログラムの追加先をなぞる入力操作 J を行なうことによって、下地要素に関するジェスチャ定義に基づいて入力ジェスチャ特性情報に対応するコマンドが実行されて、下地要素が、プログラム構成要素と接続線からなるラダープログラムに置き換えられる。

[0094] すなわち、操作対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なうことができるようになるため、タブレット端末において、図

形を用いて記述されるプログラムを、より少ない労力で編集できるという効果を奏する。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0095] (2-4) 記号を切り取り・削除する例

図22では、(a)のように入力指示体をa接点記号121の左上側から右下側に向かって移動させ、a接点記号121の右下側で水平方向に左方に向かって移動させた後、a接点記号121の左下側から右上側に向かって移動させるように入力操作Jを行う。

[0096] すなわち、はじめに、入力ジェスチャ特性情報取得手段11は、入力操作JのF1, F2, F3の座標情報および座標の軌跡、並びにF1に始まってF3で終わる軌跡の向きを取得する。

[0097] ついで、処理実行手段12は、入力ジェスチャ特性情報取得手段11が取得した入力ジェスチャ特性情報に含まれる座標情報に基づいて、操作対象を特定する。この例では、操作は、a接点記号121の左上側の領域を始点として(F1)、右下側に向かって移動させ、a接点記号121の右下側で水平方向に左方に向かって移動させた後、a接点記号121の左下側から右上側に向かって移動させ(F2)、a接点記号121の右上側の領域で入力を終了している(F3)ことから、操作対象はプログラム構成要素であるa接点記号121と特定される。

[0098] その後、処理実行手段12は、特定された操作対象、すなわちこの例ではa接点記号121に関するジェスチャ定義に含まれる軌跡の形状とその向きと、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きを照合し、a接点記号121に適用するコマンドを特定する。すなわちこの例では、a接点記号121を切り取って接続線に置き換えるコマンドが実行される。

[0099] これによって、(b)に示されるように、入力操作Jの軌跡が横切るa接点記号121が切り取られ、その領域に接続線111が配置される。

- [0100] なお、切り取られた a 接点記号 1 2 1 は、他の領域に貼り付けることを可能としてもよい。また、この場合のジェスチャ定義は、全てのプログラム構成要素を対象として定義してもよい。
- [0101] 図 2 3 も図 2 2 と同様であるが、図 2 2 では入力操作 J の軌跡が横切るプログラム構成要素が切り取られる処理であったのに対して、図 2 3 では入力操作 J の軌跡が横切るプログラム構成要素（この場合には a 接点記号 1 2 1）が削除されて、接続線に置き換わる場合を示している。この処理の詳細は、図 2 2 の場合と略同様であるので、その説明を省略する。
- [0102] なお、削除された a 接点記号 1 2 1 は、他の領域に貼り付けるとことを不能としてもよい。また、この場合のジェスチャ定義は、全てのプログラム構成要素を対象として定義してもよい。
- [0103] 図 2 4 では、並列回路を削除する場合を示している。（a）に示されるように、並列回路の接続線 1 1 2 a, 1 1 2 b と a 接点記号 1 2 1 b をなぞるように、入力指示体を領域 L 中の接続線 1 1 1 よりも下側の領域から領域 M に向かって水平方向に移動させた後、上側に向かって約 90 度方向転換し、接続線 1 1 1 の上側の領域まで移動させるように入力操作 J を行う。その結果、（b）のように入力操作を行った位置の並列回路が削除される。この処理の詳細は、図 1 9 の場合と略同様であるので、その説明を省略する。
- [0104] 以上、図 2 2 から図 2 4 にて説明したように、削除したいプログラム構成要素をなぞる入力操作 J を行なうことによって、そのプログラム構成要素に関するジェスチャ定義に基づいて入力ジェスチャ特性情報に対応するコマンドが実行されて、プログラム構成要素が、接続線ないし下地要素に置き換えられる。
- [0105] すなわち、操作対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なうことができるようになるため、タブレット端末において、図形を用いて記述されるプログラムを、より少ない労力で編集できるという効果を奏する。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンド

を指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0106] (2-5) その他の例

(2-5-1) 2つ以上の構成要素とジェスチャとを組み合わせた配置処理
上記した例では、プログラム編集画面の構成要素と入力操作の組み合わせで、その構成要素に適用するコマンドを定義していた。しかし、これに限られず、たとえば入力操作の対象の近傍に存在するプログラム構成要素の有無によって実行されるコマンドを区別できるように、ジェスチャ定義を構成してもよい。

[0107] 図25では、プログラム構成要素の一つである命令行の左側の近傍で接続線を跨ぐように入力操作を行って、接続線を別のプログラム構成要素に変更する場合を示している。すなわち、(a)のように、命令行132の左側に接続線111の上側の領域から下側の領域に向かう入力操作Jを行う。

[0108] すなわち、はじめに、入力ジェスチャ特性情報取得手段11は、入力操作JのF1、F2、F3の座標情報および座標の軌跡、並びにF1に始まってF3で終わる軌跡の向きを取得する。

[0109] ついで、処理実行手段12は、入力ジェスチャ特性情報取得手段11が取得した入力ジェスチャ特性情報に含まれる座標情報に基づいて、操作対象を特定する。この例では、入力操作の軌跡が、命令行132の左側の接続線111の上側の領域を始点として(F1)、接続線111を経由して(F2)、接続線111の下側の領域で入力を終了している(F3)ことから、操作対象はプログラム構成要素である命令行132の左に在る接続線111と特定される。

[0110] その後、処理実行手段12は、特定された操作対象、すなわちこの例では接続線111に関するジェスチャ定義に含まれる軌跡の形状とその向きと、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きを照合し、適用するコマンドを特定する。すなわちこの例では、入力ジェスチャ特性情報に含まれる軌跡が上下方向の直線であって、その軌跡の向きは下向きであって、

この接続線 1 1 1 の右に命令行 1 3 2 が隣接しているため、命令行 1 3 2 を立ち下がり命令行 1 3 2 a に変更するコマンドが実行される。

[0111] これによって、(b) に示されるように、命令行 1 3 2 が、命令行の左側に下向きの矢印が付加された立ち下がり命令行 1 3 2 a に変更される。

[0112] 図 2 6 では、(a) のように、一本の接続線 1 1 1 には、2 つの a 接点記号 1 2 1 a, 1 2 1 b とコイル記号 1 3 1 とが配置されている。このラダープログラムで、(b) のように 2 つの a 接点記号 1 2 1 a と a 接点記号 1 2 1 b の間に a 接点記号を追加する入力操作 J が行われたとする。このとき、処理実行手段 1 2 は、操作対象として接続線 1 1 1 を特定し、接続線 1 1 1 に関するジェスチャ定義から入力操作 J に該当するコマンドを実行する。すなわち、接続線 1 1 1 を a 接点記号に置き換える。

[0113] このとき、処理実行手段 1 2 は、単に入力操作 J を行った領域に a 接点記号を配置するだけでもよいし、領域が小さい場合に 2 つの a 接点記号 1 2 1 a, 1 2 1 b 間の距離を広げてそこに a 接点記号を配置するようにしてもよい。後者の場合には、たとえば処理実行手段 1 2 を、論理的にラダープログラムの該当箇所にプログラム構成要素 (a 接点) を挿入した上で、プログラム全体を表示する機能を有するように構成することで実現できる。あるいは、処理実行手段 1 2 に、プログラム構成要素 (a 接点記号) を配置する領域がそのプログラム構成要素を配置するのに十分な面積を有しているかを判定する機能と、十分な面積を有していない場合にその領域に隣接するプログラム構成要素間の距離を広げてからプログラム構成要素を配置する機能と、を設けるようにしてもよい。

[0114] その結果、2 つの構成要素が近接している場合に、(c) のようにその距離を広げるとともに、その広げた領域に入力操作 J に対応する a 接点記号 1 2 1 c を配置することが可能になる。

[0115] すなわち、操作対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なうことができるようになるため、タブレット端末において、図形を用いて記述されるプログラムを、より少ない労力で編集できるという効

果を奏する。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0116] (2-5-2) 2画以上の軌跡と配置要素とを組み合わせた配置処理

上記した例では、ジェスチャ定義が、1画の軌跡で定義される場合を説明したが、2画以上の軌跡を有するジェスチャ定義を定義してもよい。

[0117] 図27は、2つのジェスチャ定義を組合わせた編集の例を示す図である。

(a) のようなプログラムP10, P11を含むラダープログラムがあり、この中のプログラムP11の実行可否を指定できるようにマスタコントロールと呼ばれるプログラムを行ないたい場合の例を説明する。

[0118] このような場合には、(b) に示されるように、まず1画目として、プログラムP11の外縁をコの字形状の入力操作J1でなぞるように描く。この入力操作によって、処理実行手段12はプログラムP11を選択状態にするコマンドを実行する。ついで、(c) に示されるように、2画目として、プログラムP10とプログラムP11との間の左母線101を左から右に向かって横切るような入力操作J2を描く。これによって、処理実行手段12は、(d) に示されるように、プログラムP10とプログラムP11との間に、マスタコントロール回路の開始を指示するプログラムP13が配置され、プログラムP11の下には、マスタコントロールの終了を指示するプログラムP14が配置されたマスタコントロール回路が作成される。なお、マスタコントロール回路は一例であり、このように処理の対象となるプログラムを選択する入力操作と、そうして選択されたプログラムに対してコマンドを実行する入力操作を組み合わせる手順は、他の処理に対して適用できることはもちろんである。

[0119] このように、一度の入力操作にて、その入力操作にて指定した構成要素に対してコマンドを行うだけでなく、操作対象を選択する入力操作と、選択した操作対象に対して実行するコマンドを指定する入力操作を組み合わせられ

るように構成することで、入力操作に対応付けることができる操作の種類をさらに増やすことができる。

[0120] すなわち、プログラムに加えたい変化と相似している同一の軌跡を用いて、操作対象に応じた異なるコマンドを実行できるようになるため、ユーザにとって、入力操作に用いる軌跡の記憶が容易になり、タブレット端末において、図形を用いて記述されるプログラムの編集方法を、より少ない期間で習得できるという効果を有する。

[0121] 加えて、操作対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なうことができるため、タブレット端末において、図形を用いて記述されるプログラムを、より少ない労力で編集できるという効果を奏する。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0122] (2-5-3) 入力操作にプログラム編集画面に対するコマンドを定義する場合

上記した例では、ジェスチャ定義にて、入力操作と構成要素に対するコマンドを定義づける場合を示したが、入力操作に、プログラム編集画面に対するコマンド、すなわち、たとえば、プログラム編集画面に表示するプログラムの範囲を変更する処理（以下、スクロール処理という）、直前に実行したコマンドを実行する前の状態に戻す処理または、直前に実行したコマンドと同じコマンドを実行する処理などを定義してもよい。

[0123] たとえば、構成要素の上で行った入力操作によってスクロール処理が行われるように、2本の入力指示体で行う入力操作を全ての構成要素のジェスチャ定義にてスクロール処理に関する軌跡として定義したり、2本の入力指示体で行う入力操作をプログラム編集画面に対する処理と判別する機能を処理実行手段12に設けたりしてもよい。または、ジェスチャ定義にて、左母線の左側の領域L（左側下地要素L）もしくは右母線の右側の領域R（右側下

地要素R)に関するコマンドとしてスクロール処理を定義してもよい。

[0124] 図28では、2本の入力指示体による入力操作によるスクロール処理の例を示している。(a)のプログラム編集画面180に表示されているラダープログラムよりも下部に在るラダープログラムを見たい場合には、(b)のようにプログラム編集画面180のラダープログラムが描かれている領域を2本の入力指示体でスクロールしたい方向に入力操作Jを行う。その結果、入力指示体の移動とともにプログラム編集画面180に表示されているラダープログラムの範囲が移動する。そして、(c)のように、所望の範囲が現れたところで、入力指示体をプログラム編集画面180から離して入力操作Jを終了することで、スクロール処理が終了する。

[0125] 図29では、左側下地要素に関するジェスチャ定義に直線の軌跡に対応するコマンドとしてスクロール処理を定義した場合の例を示している。(a)のプログラム編集画面180の状態よりも後方(下部)に存在するラダープログラムを見たい場合には、(b)のようにプログラム編集画面180の左側下地要素Lに対して1本の入力指示体にてスクロールしたい方向に入力操作Jを行う。その結果、入力指示体の移動とともにプログラム編集画面180に表示されているラダープログラムの範囲が変わる。そして、(c)のように、所望の範囲が現れたところで、入力指示体をプログラム編集画面180から離して入力操作Jを終了することで、スクロール処理が終了する。なお、ここでは入力操作Jを左側下地要素Lに対して行う場合を説明したが、右側下地要素Rに対して行なう場合についても同様のジェスチャ定義を行なってもよい。また、入力操作の軌跡の時間変化の大小によって、入力操作を終えた後のスクロールの継続時間を変化するようにしてもよい。また、スクロール以外の編集処理についても、ジェスチャ定義に定義することによって、実行することが可能である。例えば、処理を取り消すコマンド、取り消した処理をやり直すコマンド、または処理を繰り返すコマンドをジェスチャ定義に定義することによって実行するようにしてもよい。

[0126] ここで、一般的なエンジニアリングツールでのラダープログラムの作成処

理と比較した効果について説明する。図30は、一般的なエンジニアリングツールにおけるプログラム編集手順を模式的に示す図である。ここで示される手順は、入力部としてマウスなどのポインティングデバイスを有するパーソナルコンピュータ用のエンジニアリングツールをタブレット端末で実行させるものである。一般的な場合には、まず、図30(a)に示されるように、左母線101と右母線102をプログラム編集画面に配置する。その後、図30(b)に示されるように、プログラム構成要素を配置したいプログラム編集画面上の位置P1を入力指示体でタップし、ついで、プログラム構成要素の配置用ボタンが並べて配置されたツールバーからたとえばa接点記号配置用ボタン141を入力指示体でタップする。これによって、図30(c)に示されるように、プログラム編集画面にa接点記号121aが追加される。このa接点記号121aは、左母線101と接続線111aで接続されているが、a接点記号121aの右側に配置される接続線111bは右母線102に接続されていない。

[0127] その後、図30(d)に示されるように、ツールバー中のコイル記号配置用ボタン142を入力指示体でタップすると、a接点記号121aと右母線102との間にコイル記号131が追加されるとともに、コイル記号131とa接点記号121aとの間が接続線111bで結ばれ、コイル記号131と右母線102との間が接続線111cで結ばれる。

[0128] ついで、図30(e)に示されるように、ツールバー中のa接点記号配置用ボタン141をタップすると、図30(f)に示されるように、上記で作成したa接点記号121aの下に、a接点記号121bが配置される。このa接点記号121bも左母線101とは接続線112aで接続されているが、a接点記号121bの右側に配置される接続線112bは右母線102に接続されていない。

[0129] その後、図30(g)に示されるように、接続線111bを置きたい位置P2、すなわちここでは、a接点記号121bに接続される接続線112bの右端に対応する接続線111b上の位置P2をタップし、ツールバー中の

上下方向に延在する接続線配置用ボタン 1 4 3 をタップする。これによって、図 3 0 (h) に示されるように、接続線 1 1 1 b 上の位置 P 2 と接続線 1 1 2 b の右端との間を接続する接続線 1 1 2 c が配置され、 a 接点記号 1 2 1 a , 1 2 1 b が並列に配置された並列回路が作成される。

[0130] このように、一般的なエンジニアリングツールでは、左母線 1 0 1 と右母線 1 0 2 との間にプログラム構成要素を配置する場合に、記号を配置する位置を選択し、ツールバーで配置する記号を選択するという手順が必要であった。

[0131] これに対して、実施の形態 2 では、入力操作を入力ジェスチャ特性情報取得手段 1 1 にて取得し、処理実行手段 1 2 は、入力ジェスチャ特性情報取得手段 1 1 にて取得した入力ジェスチャ特性情報の座標情報に基づいて、操作対象を特定し、前記ジェスチャ特性情報に含まれる軌跡の形状とその向きと、前記構成要素に関するジェスチャ定義に定義されている軌跡の形状とその向きとコマンドに基づいてコマンドを実行するようにした。その結果、操作対象の指定とその操作対象に対して実行するコマンドの指定を一つの入力操作で行うことができる。

[0132] つまり、図 3 0 で行なった操作を、実施の形態 2 で説明した手順で入力する場合、図 3 0 (i) に示したように、2 つの入力操作、すなわち、左母線 1 0 1 の左側の領域から右母線 1 0 2 の右側の領域に至る水平方向の入力操作 J 1 を行なったのち、左母線の左側の領域から母線間の領域に水平方向に移動し、上に向かって約 9 0 度方向転換し、入力操作 J 1 によって引かれた接続線 1 1 1 に交差する入力操作 J 2 を行なうことで、図 3 0 (g) のラダープログラムを入力できる。

[0133] つまり、タブレット端末において、一般的なエンジニアリングツールに比して少ない操作回数および労力でプログラムを編集することができるという効果を奏する。

[0134] さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定する

ための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0135] また、ジェスチャ定義を、構成要素と入力操作の軌跡の形状とその向きとコマンドの組み合わせで定義するようにしたので、同一の軌跡の形状とその向きに対応するコマンドを、構成要素ごとに定義することが可能であるので、軌跡の種類（数）を抑えることが可能となる。つまり、ユーザが覚える軌跡の種類（数）が少なくて済むので、ユーザにとって、入力操作に用いる軌跡の記憶が容易になり、より少ない期間で操作方法を習得しやすいプログラム作成環境を提供することができるという効果も有する。

[0136] なお、以上では、処理実行手段１２は、入力ジェスチャ特性情報に含まれる座標情報に基づいて操作対象を特定し、その操作対象に関するジェスチャ定義に基づいて、実行されるコマンドを特定する例を説明したが、構成要素の上下左右で領域を区別して、その領域に関するジェスチャ定義に基づいて、実行されるコマンドを特定するように構成してもよい。

[0137] すなわち、たとえば、左母線の左側の領域と、左母線と右母線の間の領域と、右母線の右側の領域を座標情報で区別して、処理実行手段１２にて、入力操作Ｊが各領域をどのような順序で横切ったかを判定し、ジェスチャ定義には、座標情報と軌跡の形状とその向きと実行するコマンドのほかに、構成要素の代わりに、領域を横切る順序の組み合わせを定義し、ジェスチャ定義に基づいて対応するコマンドを実行するように構成して、図９、図１０について説明した効果を得てもよい。

[0138] また、たとえば、接続線の上側の領域と下側の領域を区別し、上側の領域と下側の領域を座標情報で区別して、処理実行手段１２にて、入力操作Ｊが各領域をどのような順序で横切ったかを判定し、ジェスチャ定義には、座標情報と軌跡の形状とその向きと実行するコマンドのほかに、構成要素の代わりに、領域を横切る順序の組み合わせを定義し、ジェスチャ定義に基づいて対応するコマンドを実行するように構成して、図１１について説明した効果を得てもよい。

[0139] また、たとえば、a 接点記号の上下左右斜めの領域と a 接点記号の領域を座標情報で区別して、処理実行手段 12 にて、入力操作 J が各領域をどのような順序で横切ったかを判定し、ジェスチャ定義には、座標情報と軌跡の形状とその向きと実行するコマンドのほかに、構成要素の代わりに、領域を横切る順序の組み合わせを定義し、ジェスチャ定義に基づいて対応するコマンドを実行するように構成して、図 16 について説明した効果を得てもよい。

[0140] 実施の形態 3.

ジェスチャ定義に定義する軌跡の形状とその向きは、文字を模して定義してもよい。実施の形態 3 では、文字に模した軌跡をジェスチャ定義に定義する場合について説明する。

[0141] ジェスチャ定義に文字を模した軌跡を定義する場合には、ジェスチャ定義に文字を模した軌跡を定義しない場合に比して、入力ジェスチャ特性情報には高い解像度の位置情報を格納することが必要になる。そこで、実施の形態 3 では、入力ジェスチャ特性情報取得手段 11 は、入力ジェスチャ特性情報から 2 種類の解像度で入力軌跡を生成する。

[0142] プログラム編集画面は、所定の大きさの第 1 分割単位によって複数の領域に区切られている。この領域は、入力指示体から文字が入力された場合を識別するときに用いられるものである。以下では、第 1 分割単位で区切られる領域を詳細領域という。また、プログラム編集画面は、第 1 分割単位よりも大きく、かつ第 1 分割単位の面積の整数倍の大きさの第 2 分割単位によっても領域管理される。この領域は、入力指示体から文字を模していない軌跡が入力された場合を識別するときに用いられるものである。以下では、第 2 分割単位で管理される領域を粗領域という。そして、入力ジェスチャ特性情報取得手段 11 は、詳細領域と粗領域とを対応付けた領域管理情報に基づいて、2 つの入力ジェスチャ特性情報を生成する。

[0143] プログラム編集画面を、詳細領域と粗領域に対応付けるのは、プログラム編集画面に表示する文字情報に含まれる文字の大きさ（面積）が、プログラム構成要素の大きさと比して大幅に小さいため、処理実行手段 12 にて、文

字を模した軌跡と文字を模していない軌跡を区別することを可能し、演算処理の量を減らせるようにするためである。

[0144] プログラム編集画面に表示する文字の大きさがプログラム編集画面に表示するプログラム構成要素と比してさほど小さくない場合は、ひとつの領域で構成してもよい。

[0145] また、粗領域に対応付けた管理情報を設けずに、詳細領域に対応付けた領域管理情報に基づいて、1つの入力ジェスチャ特性情報を生成して、処理実行手段12にて、文字を模した軌跡と文字を模していない軌跡を区別しないように構成してもよい。

[0146] 領域管理情報は、プログラム編集画面上で設定された領域を管理する。ここでは、領域管理情報として、プログラム編集画面上で設定された詳細領域を区分する情報と、詳細領域を粗領域と対応付けた情報と、を例示する。図31は、領域管理情報の概念を模式的に示す図であり、(a)はプログラム編集画面での詳細領域と粗領域との関係を示す図であり、(b)は領域管理情報の一例を示す図である。図31(a)に示されるように、プログラム編集画面は、たとえば画面左上を原点として、左右方向にx軸が設けられ、x軸に垂直な方向にy軸が設けられるものとする。このx-y面上に第1分割単位の詳細領域 $FM \times i \times y \times j$ (i, j は自然数)が複数配置されている。詳細領域の大きさは、入力指示体のプログラム編集画面との接触領域の大きさに応じて任意に設定することができる。たとえば、入力指示体として指のみが用いられる場合には、スタイラスペンが用いられる場合に比して、接触領域の大きさを大きくすることができる。また、1つの粗領域 $CM \times m \times y \times n$ (m, n は自然数)は、この例では5行×2列の詳細領域 $FM \times i \times y \times j$ からなる。さらに、1つの詳細領域 $FM \times i \times y \times j$ は、座標の単位が 10×10 の大きさの領域からなるものとする。このような関係が、図31(b)に示されるようにたとえばテーブル形式で表されている。すなわち、各詳細領域 $FM \times i \times y \times j$ に含まれるプログラム編集画面上での座標範囲が規定され、各粗領域 $CM \times m \times y \times n$ に含まれる詳細領域 $FM \times i \times y \times j$ が規定されている。

- [0147] 入力ジェスチャ特性情報取得手段 11 は、座標単位で得られる入力ジェスチャ特性情報と、領域管理情報と、から、粗領域単位での粗入力軌跡（第 1 入力軌跡）と、詳細領域単位での詳細入力軌跡（第 2 入力軌跡）と、を生成する。
- [0148] 処理実行手段 12 は、たとえば、粗入力軌跡に含まれる座標情報に基づいて操作対象を特定し、その操作対象に関するジェスチャ定義について、最初に粗入力軌跡に対応する軌跡が存在するか判定し、存在する場合には、その軌跡に対応するコマンドを実行する。この場合には、詳細入力軌跡を用いた操作対象の特定処理は行われない。また、処理実行手段 12 は、粗入力軌跡に対応する軌跡がジェスチャ定義に存在しない場合には、詳細入力軌跡に含まれる座標情報に基づいて操作対象を特定し、その操作対象に関するジェスチャ定義について、詳細入力軌跡に対応する軌跡が存在するか判定し、存在する場合には、その軌跡に対応するコマンドを実行する。詳細入力軌跡を用いる場合は、ジェスチャが文字である場合であるので、詳細入力軌跡に含まれる軌跡の形状とその向きを用いて文字認識を行って、入力された文字を特定する。
- [0149] さらに、構成要素配置可能下地要素の一種として、標準下地要素と、プログラム構成要素のデバイス名を入力するためのデバイス名入力下地要素と、構成要素のコメント情報を入力するためのコメント入力下地要素をそれぞれ定義し、操作対象を特定することで、入力された文字がプログラム構成要素に対するデバイス名であるのか、またはコメント情報であるのかを判別できるようにしてもよい。たとえば、処理実行手段 12 は、操作対象がデバイス名入力下地要素の場合は、ジェスチャ定義に定義されているコマンドに基づいて、入力ジェスチャ特性情報に格納されている軌跡の形状、または軌跡の形状およびその向きに基づいて文字認識処理を行って取得した文字を、そのデバイス名入力下地要素に関連付けられているプログラム構成要素のデバイス名として登録し、そのデバイス名をプログラム編集画面の所定の位置に配置する処理を行う。また、操作対象がコメント入力下地要素である場合は、

ジェスチャ定義に定義されているコマンドに基づいて、入力ジェスチャ特性情報に格納されている軌跡の形状、または軌跡の形状およびその向きに基づいて文字認識処理を行って取得した文字をそのデバイス名入力下地要素と関連付けられている構成要素のコメント情報として登録し、そのコメント情報をプログラム編集画面の所定の位置に配置する処理を行う。なお、ここに、デバイス名は、プログラム構成要素のひとつひとつを区別するための名前である。コメントはプログラム構成要素につける説明文である。

[0150] なお、このとき、処理実行手段１２と入力ジェスチャ特性情報取得手段１１は、１画目の入力操作に基づいて操作対象を特定するように構成してもよい。また、入力操作が予め定めた時間無いことをもって入力操作の終了と判定するように構成することで、下地要素の領域の外に入力操作が続いた場合にも文字入力を継続できるようにしてもよい。

[0151] 図３２は、デバイス名入力下地要素とコメント入力下地要素を用いて、ラダープログラムにおける文字入力領域を指定する一例を示す図である。この図に示されるように、ラダープログラムは、左母線１０１と右母線１０２との間を結ぶ接続線１１１、１１２に貫かれてプログラム構成要素記号が配置され、接続線１１１、１１２は、左母線１０１と右母線１０２との間で上下方向に配置される。また、ラダープログラムでは、記号よりも上側にその記号に対するデバイス名を表記し、記号よりも下側にその記号に対するコメントを表記するのが一般的である。そのため、接続線１１１、１１２で挟まれた領域では、入力された文字がデバイス名なのかコメントなのかが不明になる虞がある。そこで、実施の形態３では、接続線１１１、１１２の上側の所定の範囲にデバイス名入力下地要素を配置してデバイス名入力領域Ｒ_dを設け、接続線１１１、１１２の下側の所定の範囲にコメント入力下地要素を配置してコメント入力領域Ｒ_cを設けるようにしている。そして、各構成要素には、デバイス名入力下地要素とコメント入力下地要素が結び付けられている。

[0152] また、接続線１１１のａ接点記号１２１ａの上部には、デバイス名入力領

域 R d 1 2 1 a が設けられ、下部にはコメント入力領域 R c 1 2 1 a が設けられている。接続線 1 1 1 のコイル記号 1 3 1 a の上部には、デバイス名入力領域 R d 1 3 1 a が設けられ、下部にはコメント入力領域 R c 1 3 1 a が設けられている。さらに、接続線 1 1 2 の a 接点記号 1 2 1 b の上部には、デバイス名入力領域 R d 1 2 1 b が設けられ、下部にはコメント入力領域 R c 1 2 1 b が設けられている。接続線 1 1 2 のコイル記号 1 3 1 b の上部には、デバイス名入力領域 R d 1 3 1 b が設けられ、下部にはコメント入力領域 R c 1 3 1 b が設けられている。

[0153] 図 3 3 は、ジェスチャ定義の一例を示す図である。ここでは、図 2 または図 3 と同様に、タイル状の領域をプログラム編集画面に敷き詰めて、領域を 1 単位として軌跡の経過を把握する方法について例示している。図 3 3 (a) は、粗領域の場合のタイル状のマス目の配置関係を示す図であり、図 3 3 (b) は、詳細領域の場合のタイル状のマス目の配置関係を示す図であり、図 3 3 (c) は、図 3 3 (a)、(b) の場合のジェスチャ定義の一例を示す図である。

[0154] この例では、ジェスチャ定義は、構成要素の種類と、軌跡の形状と、その向きと、実行するコマンドと、を対応付けたものである。軌跡の形状とその向きの情報は、粗領域用の情報と詳細領域用の情報を備える。入力ジェスチャ特性情報に格納される座標情報は、上記したようにプログラム編集画面に定義された座標系に基づいて得られる座標となっている。この例では、粗領域用の情報は、図 2 または図 3 の場合と同様である。詳細領域用の情報は、粗領域のマス目をさらに 9 分割し、入力操作の軌跡を粗領域の場合よりも詳細に区別することができる。ここでは、複数のマス目を用いて軌跡の形状を定義している。すなわち、軌跡の形状は、始点から終点に至るまでに経由したマス目の座標で判別され、軌跡の向きは、始点から終点に至るまでに経由したマス目の順序で判別される。

[0155] 粗領域および詳細領域のマス目の大きさは、プログラム編集画面に表示している構成要素の大きさと、入力指示体の太さに基づいてユーザが調整する

ことも可能な物理的な寸法を有するが、ジェスチャ定義のマス目は、物理的な寸法を定義してもよいが、物理的な特定の寸法を持たない論理的な位置関係のみを定義してもよい。図 3 3 に示したジェスチャ定義は、論理的な位置関係を定義した場合を表している。

[0156] なお、ここでは矩形のマス目で説明したが、たとえば正六角形など、ほかの形状のマス目を用いてもよい。

[0157] つぎに、プログラムの編集方法について説明する。図 3 4 は、実施の形態 3 によるプログラム構成要素が何も配置されていないプログラム編集画面でのプログラム編集方法の処理手順の一例を示すフローチャートである。

[0158] まず、入力ジェスチャ特性情報取得手段 1 1 は、プログラム編集画面に入力される入力ジェスチャ特性情報を取得する（ステップ S 5 1）。入力ジェスチャ特性情報は、入力指示体の接触位置である座標情報、座標の軌跡、および軌跡の向きからなる。

[0159] ここで、入力ジェスチャ特性情報取得手段 1 1 は、座標情報に基づいて、軌跡の形状とその向きの情報として、粗領域単位で粗入力軌跡を生成するとともに、詳細領域単位での詳細入力軌跡を作成する（ステップ S 5 2）。

[0160] ついで、処理実行手段 1 2 は、詳細入力軌跡から文字を認識し、入力文字を取得する（ステップ S 5 3）。その後、処理実行手段 1 2 は、入力文字を取得できたかを判定する（ステップ S 5 4）。

[0161] 詳細入力軌跡から入力文字を取得できた場合（ステップ S 5 4 で Y e s の場合）には、詳細入力軌跡の形状とその向きに対応するコマンド、すなわちステップ S 5 7 で取得した文字に対応するコマンドがジェスチャ定義に定義されているかを判定する（ステップ S 5 5）。詳細入力軌跡の形状とその向きに対応するコマンド、すなわちステップ S 5 3 で取得した文字に対応するコマンドがジェスチャ定義に存在する場合（ステップ S 5 5 で Y e s の場合）には、処理実行手段 1 2 は、ジェスチャ定義に格納されているコマンドを実行し（ステップ S 5 6）、プログラム構成要素に登録した文字をプログラム編集画面に表示する（ステップ S 5 7）。以上によって、プログラムの編

集処理が終了する。

[0162] 一方、ステップS 5 4で詳細入力軌跡から入力文字を取得できなかった場合（ステップS 5 4でN oの場合）には、処理実行手段1 2は、粗入力軌跡の形状とその向きに対応するコマンドがジェスチャ定義に存在するかを判定する（ステップS 5 8）。粗入力軌跡の形状とその向きに対応するコマンドがジェスチャ定義に存在する場合（ステップS 5 8でY e sの場合）には、処理実行手段1 2は、ジェスチャ定義に格納されているコマンドを実行し（ステップS 5 9）、プログラム構成要素をプログラム編集画面に表示する（ステップS 6 0）。以上によって、プログラムの編集処理が終了する。

[0163] また、ステップS 5 8で粗入力軌跡の形状とその向きに対応するコマンドが操作対象に関するジェスチャ定義にない場合（ステップS 5 8でN oの場合）またはステップS 5 5で詳細入力軌跡の形状とその向きに対応するコマンドが操作対象に関するジェスチャ定義にない場合（ステップS 5 5でN oの場合）には、該当するコマンドが無いと判定し（ステップS 6 1）、処理が終了する。なお、該当するコマンドが無いと判定したときに、エラーメッセージを表示するように構成してもよいし、エラーメッセージを表示しないように構成してもよい。

[0164] なお、入力ジェスチャ特性情報取得手段1 1は、ステップS 5 1で入力ジェスチャ特性情報として、入力操作の軌跡を取得しながら、ステップS 5 2での粗入力軌跡と詳細入力軌跡の生成とを、実行するようにしてもよい。また、処理実行手段1 2は、粗入力軌跡を用いたステップS 5 8～S 6 0の処理と、詳細入力軌跡を用いたステップS 5 3～S 5 7の処理と、を同時並行的に処理するようにしてもよい。

[0165] ここで、プログラム構成要素が配置されていないプログラム編集画面に文字に模した軌跡の入力操作を行う場合の具体例について説明する。

[0166] 図3 5は、実施の形態3によるプログラム編集処理の手順の一例を示す図である。この図3 5では、左母線1 0 1と、右母線1 0 2と、a接点記号1 2 1およびコイル記号1 3 1を有する2つの母線1 0 1， 1 0 2間を結ぶ接

続線 1 1 1 と、を配置する入力操作を示している。まず、図 3 5 (a) に示されるように、ユーザが入力指示体を用いてプログラム編集画面上に入力操作 J を行う。入力操作 J から入力ジェスチャ特性情報が取得され、粗入力軌跡と詳細入力軌跡とが生成される。

[0167] 処理実行手段 1 2 は、ステップ S 5 7 にて入力ジェスチャ特性情報（詳細入力軌跡）を用いて文字認識を行い「X 1」という入力文字を取得する。次に、入力ジェスチャ特性情報の最初の 1 文字「X」に対応するコマンドを実行する。その結果、図 3 5 (b) に示されるように、プログラム構成要素が配置される。この例では、左母線 1 0 1 と、右母線 1 0 2 と、a 接点記号 1 2 1 およびコイル記号 1 3 1 を有する 2 つの母線 1 0 1, 1 0 2 間を結ぶ接続線 1 1 1 と、を有するプログラム構成要素がプログラム編集画面上に配置される。また、認識された入力文字「X 1」が、プログラム構成要素の a 接点記号 1 2 1 の上の領域にデバイス名 1 5 1 として配置される。

[0168] これは一例であり、ジェスチャと、実行されるコマンドの組み合わせはジェスチャ定義にて他の定義をおこなってもよい。

[0169] また、図 3 5 では、文字に模した軌跡の入力操作を行うことによってプログラム編集画面へのプログラムの追加およびプログラム構成要素のデバイス名への文字入力を行う例を示したが、文字入力を他の方法で実行してもよい。図 3 6 は、実施の形態 3 によるプログラム編集処理の手順の一例を示す図である。(a) のようにプログラム編集画面 2 0 0 内で所定の入力操作 J を行う（たとえば、この例ではタップ動作を行っている）ことで、(b) のようにプログラム編集画面 2 0 0 上に文字入力ウィンドウ 2 1 0 を表示させるとともに、入力操作 J を行った領域に文字入力領域 2 2 0 を配置する。そして、(c) のように文字入力ウィンドウ 2 1 0 から文字を入力することによって、(d) のようにプログラム 1 0 0 を配置するようにしてもよい。なお、ここで例示されている文字入力ウィンドウ 2 1 0 は、「X」、「Y」、「M」、「L」のアルファベットと、「0 (0 0)」～「9」の数字と、「← (戻る)」、「取消」、「決定」の各命令と、からなる。これは、ラダープ

プログラムで使用される文字は決まっているので、すべてのアルファベットを配置せず、よく使用されるもののみを配置するようにしている。これによって、文字を誤り無く入力（選択）し易くなるとともに、文字入力ウィンドウ 210 のプログラム編集画面 200 内で占める割合を小さくしてプログラム編集画面 200 の視認性を高めることができる。

[0170] また、たとえば、デバイス名と関連付けられて用いられるラベルと呼ばれる変数名の一覧を配置するようにして、ラベルを誤り無く入力（選択）し易くするようにしてもよい。

[0171] つぎに、プログラム構成要素がプログラム編集画面に配置されている場合のプログラム編集方法について説明する。図 37 は、実施の形態 3 によるプログラム構成要素が配置されている場合のプログラム編集画面でのプログラム編集方法の処理手順の一例を示すフローチャートである。まず、入力ジェスチャ特性情報取得手段 11 は、プログラム編集画面上に入力される入力ジェスチャ特性情報を取得する（ステップ S71）。

[0172] ついで、入力ジェスチャ特性情報取得手段 11 は、領域管理情報を用いて、粗領域単位の座標で粗入力軌跡および詳細領域単位の座標で詳細入力軌跡を含む入力ジェスチャ特性情報を生成する（ステップ S72）。処理実行手段 12 は、操作対象が文字入力用の構成要素であるか否かを判定する（ステップ S73）。

[0173] 操作対象が文字入力用の構成要素でない場合（ステップ S73 で No の場合）には、処理実行手段 12 は、粗入力軌跡の形状とその向きに対応するコマンドが操作対象に関するジェスチャ定義に存在するかを判定する（ステップ S74）。ここでは、粗入力軌跡を含む入力ジェスチャ特性情報に格納されている座標情報に基づいて軌跡が通過した構成要素を特定し、その構成要素を操作対象として位置づける。

[0174] 粗入力軌跡の形状とその向きに対応するコマンドが操作対象に関するジェスチャ定義に存在する場合（ステップ S74 で Yes の場合）には、処理実行手段 12 は、ジェスチャ定義に格納されているコマンドを実行し（ステッ

プS 7 5)、コマンドを実行した後のプログラムがプログラム編集画面に表示される(ステップS 7 6)。以上によって、プログラムの編集処理が終了する。

[0175] 一方、ステップS 7 3で操作対象が文字入力用の構成要素である場合(ステップS 7 3でY e sの場合)には、処理実行手段1 2は、詳細入力軌跡から文字を認識し、入力文字を取得する(ステップS 7 7)。

[0176] ついで、処理実行手段1 2は、入力ジェスチャ特性情報に格納されている座標情報に基づいて、操作対象がデバイス名入力領域R dであるかを判定する(ステップS 7 8)。操作対象がデバイス名入力領域R dである場合(ステップS 7 8でY e sの場合)には、認識した文字列を、ジェスチャ定義に定義されているコマンドに基づいてそのデバイス名入力領域R dが関連付けられているプログラム構成要素のデバイス名情報として登録するとともに、そのプログラム構成要素のデバイス名表示領域に、認識した入力文字をデバイス名として配置したプログラム編集画面を表示する(ステップS 7 9)。以上によって、プログラムの編集処理が終了する。

[0177] さらに、ステップS 7 8で操作対象がデバイス名入力領域R dではない場合(ステップS 7 8でN oの場合)には、操作対象がコメント入力領域R cであるかを判定する(ステップS 8 0)。操作対象がコメント入力領域R cである場合(ステップS 8 0でY e sの場合)には、認識した文字列を、ジェスチャ定義に定義されているコマンドに基づいて、そのコメント入力領域R cが関連付けられている構成要素のコメント情報として登録するとともに、プログラム構成要素のコメント表示領域に認識した入力文字を配置したプログラム編集画面を表示する(ステップS 8 1)。以上によって、プログラムの編集処理が終了する。

[0178] また、ステップS 7 4で粗入力軌跡の形状とその向きに対応するコマンドが操作対象に関するジェスチャ定義に存在しない場合(ステップS 7 4でN oの場合)には、該当するコマンドは無いと判定し(ステップS 8 2)、処理が終了する。さらに、ステップS 8 0で操作対象がコメント入力領域R c

でない場合（ステップS80でNの場合）には、処理が終了する。

[0179] 以下に、プログラム構成要素が既に配置されているプログラム編集画面に、（１）接続線をプログラム構成要素に置き換える場合、（２）既に配置されているプログラム構成要素にデバイス名を配置する場合、（３）デバイス名の文字を文字の一覧表を用いて入力する場合についてのプログラムの編集処理の具体例を説明する。

[0180] （１）接続線をプログラム構成要素に置き換える場合

図38～図40は、実施の形態3によるプログラム編集の処理手順の一例を示す図である。図38では、左母線101と、右母線102と、2つの母線101, 102間を結び、コイル記号131を有する接続線111と、が配置されたプログラム編集画面で、デバイス名入力領域にデバイス名を入力してa接点記号を配置する入力操作を示している。

[0181] まず、図38（a）のように、ユーザが入力指示体を用いてプログラム編集画面の接続線111よりも上で接続線111の中央よりも左寄りのデバイス名入力領域で入力操作Jを行う。入力操作Jから入力ジェスチャ特性情報が取得され、入力ジェスチャ特性情報から入力操作Jの軌跡の下にデバイス名入力下地要素が在るため、処理実行手段12が、操作対象はデバイス名入力下地要素であると判別する。

[0182] その後、処理実行手段12は、入力ジェスチャ特性情報（詳細入力軌跡）から文字認識を行い「X1」という入力文字を取得する。次に、ジェスチャ定義に基づいて、入力文字の最初の1文字「X」に対応するコマンドを実行する。その結果、図38（b）に示されるように、プログラム構成要素が配置される。この例では、入力操作Jが行われた領域に対応する位置の接続線111がa接点121に置き換わり、また、認識された入力文字「X1」が、プログラム構成要素のa接点記号121の上の領域にデバイス名151として配置される。

[0183] 図39では、左母線101と、右母線102と、2つの母線101, 102間を結び、a接点記号を有する接続線111と、が配置されたプログラム

編集画面で、デバイス名入力領域にデバイス名を入力してコイル記号または a 接点記号を配置する入力操作を示している。

[0184] まず、図 39 (a) のように、ユーザが入力指示体を用いてプログラム編集画面の接続線 1 1 1 よりも上で接続線 1 1 1 の中央よりも右寄りのデバイス名入力領域で入力操作 J を行う。入力操作 J から入力ジェスチャ特性情報が取得され、入力ジェスチャ特性情報から入力操作 J の軌跡の下にデバイス名下地要素が在るため、処理実行手段 1 2 は、操作対象がデバイス名入力下地要素であると判別する。

[0185] その後、処理実行手段 1 2 は、入力ジェスチャ特性情報（詳細入力軌跡）から文字認識を行い「Y 1」という入力文字列を取得する。次に、入力文字列の最初の 1 文字「Y」に対応するコマンドを実行する。その結果、図 39 (b) に示されるように、プログラム構成要素が配置される。この例では、入力操作 J が行われた領域に対応する位置の接続線 1 1 1 がコイル記号 1 3 1 に置き換わり、また、認識された入力文字「Y 1」が、コイル記号 1 3 1 に重ねられてデバイス名 1 5 1 として配置される。

[0186] なお、ラダープログラムにおいて、ここでは、「Y」は出力デバイスのコイルを指すデバイス名であるとともに、コイルの励磁状態を表す入力デバイスを指すデバイス名でもある。そのため、(c) のように、接続線 1 1 1 上における出力デバイス（コイル記号 1 3 1）となる領域と、入力デバイス（a 接点記号 1 2 1）となる領域とを予め定めておくことで、どの領域に入力操作が行われたかによって、配置する記号の種類を変えてもよい。この例では、左母線 1 0 1 と右母線 1 0 2 との間を a : b に区切り、それぞれの領域を R a, R b とする。領域 R b に「Y」というジェスチャの入力が行われた場合には、(b) のようにコイル記号 1 3 1 がデバイス名 1 5 1 とともに配置され、領域 R a に「Y」というジェスチャの入力が行われた場合には、(c) のように a 接点記号 1 2 1 がデバイス名 1 5 1 とともに配置される。

[0187] また、接続線 1 1 1 の上側の領域に配置したデバイス名流力領域 R d において、(d) のように「Y」という入力操作によって、入力ジェスチャ特性

情報の座標情報に対応する接続線 1 1 1 上にプログラム構成要素を配置するときに、配置する記号は「a 接点」であるのか「コイル」であるのかを選択することができる選択ウィンドウ 1 6 1 を表示させるようにしてもよい。

[0188] なお、図 3 9 では、a 接点記号、コイル記号に関する例を説明したが、ほかのプログラム構成要素を配置する場合も、ジェスチャ定義に定義することによって、同様のことを行なってもよい。たとえば、領域 R a に図 1 4 (b) に示した入力操作 J を行なったときに、命令行の代わりに、条件入力行を配置するようにしてもよい。

[0189] 図 4 0 では、左母線 1 0 1 と、右母線 1 0 2 と、2 つの母線 1 0 1, 1 0 2 間を結び、a 接点記号を有する接続線 1 1 1 と、が配置されたプログラム編集画面で、デバイス名入力領域にデバイス名を入力して命令行を配置し、さらに、命令行に引数を設定する入力操作を示している。

[0190] まず、図 4 0 (a) のように、ユーザが入力指示体を用いてプログラム編集画面の接続線 1 1 1 よりも上で接続線 1 1 1 の中央よりも左寄りのデバイス名入力領域で入力操作 J を行う。入力操作 J から入力ジェスチャ特性情報が生成される。

[0191] その後、処理実行手段 1 2 は、入力ジェスチャ特性情報（詳細入力軌跡）から文字認識を行い「MOV」という入力文字を取得する。次に、入力ジェスチャ特性情報の最初の 1 文字「M」に対応するコマンドを実行する。この例では、2 文字目以降に「OV」という文字が続いており、これが命令後に該当するため、入力操作 J が行われた領域に対応する接続線 1 1 1 上の位置に命令行 1 3 2 が配置され、また、認識された入力文字「MOV」が、プログラム構成要素の命令行 1 3 2 の所定の位置に重ねて配置される。

[0192] なお、図 4 0 (a)、(b) ではデバイス名入力領域 R d に描いた文字によって命令文を含む命令行 1 3 2 を配置する場合を示しているが、命令行 1 3 2 には通常、命令文のほかに引数の入力操作が必要である。そこで、(c) のように、引き続いて命令行 1 3 2 の上のデバイス名入力領域 R d に入力操作 J による文字を入力していくと、(d) のように命令行 1 3 2 の内部に

引数 1 5 2 b を入力するようにしてもよい。引数が複数ある場合には、この工程を引数の個数分繰り返せばよい。

[0193] これらは一例であり、入力操作の軌跡の形状とその向きと、実行されるコマンドの組み合わせはジェスチャ定義にて他の定義をおこなってもよい。

[0194] 以上、図 3 8 から図 4 0 にて説明したように、文字に模した軌跡の入力操作 J を行なうことによって、下地要素に関するジェスチャ定義に基づいて入力ジェスチャ特性情報に対応するコマンドが実行されて、下地要素が、デバイス名が登録されたプログラム構成要素に置き換えられる。

[0195] すなわち、文字に相似している軌跡を用いてプログラム構成要素を追加できるので、ユーザにとって、入力操作に用いる軌跡の記憶が容易になり、タブレット端末において、図形を用いて記述されるプログラムの編集方法を、より少ない期間で習得できるという効果を有する。

[0196] 加えて、対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なって、文字情報の登録も同時に行なうことができるようになるため、タブレット端末において、図形を用いて記述されるプログラムを、より少ない労力で編集できるという効果を奏する。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0197] (2) 既に配置されているプログラム構成要素にデバイス名を配置する場合

図 4 1 は、実施の形態 3 によるプログラム編集の処理手順の一例を示す図である。図 4 1 では、a 接点記号 1 2 1 とコイル記号 1 3 1 とが配置された接続線 1 1 1 が配置され、各記号にはデバイス名もコメントも付加されていない状態のプログラム編集画面で、a 接点記号 1 2 1 のデバイス名入力領域 R d にデバイス名を配置する入力操作を示している。

[0198] まず、図 4 1 (a) のように、ユーザが入力指示体を用いてプログラム編集画面の a 接点記号 1 2 1 の上部にあるデバイス名入力領域 R d で入力操作

Jを行う。入力操作Jから入力ジェスチャ特性情報が生成される。

[0199] その後、処理実行手段12は、入力ジェスチャ特性情報（詳細入力軌跡）から文字認識を行い「X1」という入力文字を取得する。この入力操作Jは接続線111を横切らないので、操作対象はデバイス名入力領域Rdと特定され、デバイス名入力領域Rdに関するジェスチャ定義に基づいてコマンドが実行されて、a接点記号121にデバイス名が登録される。その結果、図41（b）に示されるように、a接点記号121の上部に入力文字「X1」がデバイス名151として配置されることになる。また、図示しないがコメント入力領域に文字の形を模した入力操作が行われた場合も同様の過程を経て、a接点記号にコメント情報が登録される。

[0200] 以上、図41にて説明したように、文字に模した軌跡の入力操作Jを行なうことによって、下地要素に関するジェスチャ定義に基づいて入力ジェスチャ特性情報に対応するコマンドが実行されて、プログラム構成要素にデバイス名が登録される。

[0201] すなわち、文字に相似している軌跡を用いてプログラム構成要素に文字情報を登録できるので、ユーザにとって、入力操作に用いる軌跡の記憶が容易になり、タブレット端末において、図形を用いて記述されるプログラムの編集方法を、より少ない期間で習得できるという効果を有する。

[0202] 加えて、対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なうことができるようになるため、タブレット端末において、図形を用いて記述されるプログラムを、より少ない労力で編集できるという効果を奏する。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。

[0203] （3）デバイス名の文字を文字の一覧表を用いて入力する例

上記した説明では、ジェスチャ定義に文字の形に模した軌跡を用いてデバイス名またはコメントをプログラム構成要素に登録する処理について説明し

た。タブレット端末では、文字入力軌跡を用いる入力だけに限られず、たとえば文字の一覧表（以下、文字入力ウィンドウという）を用いた入力も行われることがある。この場合には、ジェスチャ定義に、文字入力ウィンドウを表示して、そのウィンドウから入力された文字を操作対象の特定の領域に格納するコマンドを定義して、文字入力ウィンドウによる文字入力の後に、ステップS 7 9以降の処理が実行されるようにしてもよい。図4 2と図4 3は、実施の形態3によるプログラム編集処理の手順の一例を示す図である。

[0204] 図4 2では、（a）のようにa接点記号1 2 1の上側のデバイス名入力領域R dで入力操作Jが行われる。この文字入力処理用の入力操作Jが入力されると、（b）のように文字入力ウィンドウ1 6 3が表示されるとともに、入力操作Jが行われた領域に文字入力領域1 6 2が設けられる。その後、（c）に示されるように、ユーザは、文字入力ウィンドウ1 6 3に表示された文字または数字をタップすることでその内容が文字入力領域1 6 2に入力される。ここでは、1 1→1 2→1 3の順にタップしたものとする。最後に決定ボタンをタップする（1 4）ことで文字入力領域1 6 2に入力された文字が確定され、（d）のようにプログラムに反映される。ここでは、a接点記号1 2 1の上側のデバイス名入力領域R dで入力処理が行われたので、入力された文字「X 1 0」はa接点記号1 2 1のデバイス名1 5 1として、ラダープログラムに登録される。

[0205] 図4 3は、文字を修正する場合である。この場合には、（a）のようにa接点記号1 2 1の上に配置された「X 1」というデバイス名1 5 1 aの上部で、（b）のように入力操作Jを行うと、（c）のように文字入力ウィンドウ1 6 3が表示されるとともに、入力操作Jが行われた領域に文字入力領域1 6 2が設けられる。そして、文字入力ウィンドウ1 6 3に表示された文字または数字などをタップすることで、その内容が文字入力領域1 6 2に入力される。ここでは、1 1→1 2→1 3の順にタップし「X 1 0」と入力したものとする。最後に決定ボタンをタップする（1 4）ことで文字入力領域1 6 2に入力された文字が確定され、入力された文字「X 1 0」は（d）のよ

うに a 接点記号 1 2 1 の新たなデバイス名 1 5 1 b として、ラダープログラムに登録される。

[0206] なお、図 4 2 と図 4 3 では、文字入力用の入力操作をデバイス名入力領域 R d で行う場合を示しているが、コメント入力領域で行う場合も同様である。また、ここでは、デバイス名またはコメントについての入力について説明したが、たとえば図 4 0 (b) に示されるような命令行 1 3 2 に対しても同様の処理を適用することができる。

[0207] なお、図 4 2、図 4 3 では、入力操作として、水平方向の直線状の軌跡を用いる場合を示したが、他の入力操作を文字入力処理に割り当ててもよい。たとえば、水平方向の直線状の軌跡の代わりに、タップ操作またはダブルタップ操作を割り当ててもよい。

[0208] また、上記の説明では、文字の形を模した入力操作が行われた場合に、プログラム構成要素の配置、またはプログラム構成要素に対するデバイス名またはコメントの配置に関する処理を行う場合を示したが、これに限定されるものではない。たとえば、実施の形態 2 の図 1 4 の例では、命令行 1 3 2 を配置しているが、この命令行 1 3 2 への文字の入力についても、同様の処理を適用することができる。

[0209] また、図 4 2 と図 4 3 では、デバイス名の配置の例を示したが、プログラム構成要素が配置されていない接続線 1 1 1 の上側のデバイス名入力領域で同様の入力操作を行うことで、入力操作を行った座標に対応する接続線 1 1 1 上の位置にデバイス名が配置されたプログラム構成要素を配置するように構成してもよい。

[0210] このような構成によれば、プログラム構成要素の上下の領域の何も配置されていない領域で文字入力処理が行われると、そのプログラム構成要素にデバイス名またはコメントが登録され、ラダープログラムが更新される。また、プログラム構成要素の上下の領域の文字が配置されている領域で文字入力処理が行われると、そのプログラム構成要素のデバイス名またはコメントが変更され、その変更に基づいてラダープログラムが更新される。

- [0211] すなわち、文字に相似している軌跡を用いてプログラム構成要素を編集できるので、ユーザにとって、入力操作に用いる軌跡の記憶が容易になり、タブレット端末において、図形を用いて記述されるプログラムの編集方法を、より少ない期間で習得できるという効果を有する。
- [0212] 加えて、対象の指定とその操作対象に対して行なうコマンドの指定を同時に行なって、文字情報の編集も同時に行なうことができるようになるため、タブレット端末において、図形を用いて記述されるプログラムを、より少ない労力で編集できるという効果を奏する。さらに、操作対象に対して行なう処理を一覧から選択する過程が不要となるので、プログラム編集画面に操作対象に対して行なうコマンドを指定するための選択肢の一覧を表示する必要がなくなるため、プログラム編集画面を有効に利用できるという効果を奏する。
- [0213] なお、上記した軌跡を描く入力操作による文字入力の場合は、たとえば、デバイス名入力領域を判別できずに、プログラム編集画面のデバイス名入力領域よりも上の領域に入力操作を行って、デバイス名入力領域に入力したかったものがコメント入力領域に入力されてしまう可能性もある。そのような事態を防ぐために、デバイス名の入力、コメントの入力などの文字列を、軌跡を描く入力操作で入力できる領域を区別できる図形をプログラム編集画面に表示するようにしてもよい。
- [0214] 図44は、実施の形態3による文字入力領域のガイド表示の一例を示す図である。図44(a)では、デバイス名またはコメントの入力操作を受け付ける領域がユーザに認識できるように文字入力操作受付領域164をプログラム編集画面上に明示する場合を示している。
- [0215] また、図44(b)では、ユーザが入力操作Jの入力中に、認識処理の領域を示すガイド線165をプログラム編集画面に表示する場合を示している。
- [0216] 実施の形態3では、プログラム編集画面に文字を模した軌跡を描く入力操作を行った場合に、操作対象が文字を入力可能な構成要素である場合には、

詳細入力軌跡を含む入力ジェスチャ特性情報に含まれる入力操作の軌跡または軌跡の形状とその向きから入力文字を識別する。そして、詳細入力軌跡に対応するコマンドをジェスチャ定義から取得し、実行するとともに、操作対象のデバイス名またはコメントに入力された文字を配置するようにした。これによって、文字の形を模した入力操作によって、所望の位置にプログラム構成要素をそのデバイス名またはコメントとともに配置することができるようになるため、文字の形を模した軌跡を描く操作によって図形を用いて記述されるプログラムの編集を少ない操作で容易に行えるという効果を奏する。

[0217] 実施の形態4.

左母線と右母線のみが配置されている初期プログラム編集画面にプログラミングを行う場合には、接続線を最初に配置するのが通常である。そのため、初期プログラム編集画面の場合と、他のプログラム構成要素が配置されたプログラム編集画面の場合とで、取得する入力ジェスチャ特性情報の解像度を変えることができる。以下に、その場合の例について説明する。

[0218] 左母線と右母線のみが配置されているプログラム編集画面では、第2分割単位よりも大きな大領域の単位で、プログラム編集画面を区切ることができる。図45は、母線のみが配置されているプログラム編集画面の一例を示す図である。この図に示されるように、母線のみが配置されているプログラム編集画面200Aでは、予め所定の位置に左母線101と右母線102とが配置された状態にある。また、母線のみが配置されているプログラム編集画面200Aは、左母線101よりも左側の領域L（左側下地要素）と、左母線101と右母線102とに挟まれた領域M（図形配置可能下地要素）と、右母線102よりも右側の領域（右側下地要素）Rと、を有する。なお、これらの領域L，M，Rを、大領域というものとする。なお、ここでは、プログラム編集画面に母線が表示されている例を示したが、大領域を定義する一方で、母線を表示しないように構成してもよい。

[0219] また、領域管理情報として、大領域と粗領域との間を対応付ける情報が含まれる構成となる。具体的には、領域管理情報は、粗領域に対応する座標範

囲と、大領域に含まれる粗領域と、を含む。なお、この実施の形態4では、文字を模した軌跡を入力操作する場合は扱わないので、実施の形態3のように詳細領域を設けていないが、実施の形態3のように入力操作を詳細領域単位の座標で検出する場合には、詳細領域を設けてもよい。

[0220] 入力ジェスチャ特性情報取得手段11は、プログラム編集画面に左母線と右母線以外のプログラム構成要素が配置されていない場合には、大入力軌跡（第3入力軌跡）を作成し、プログラム編集画面に左母線と右母線以外のプログラム構成要素が配置されている場合には、粗入力軌跡を作成する。

[0221] 図46は、ジェスチャ定義の一例を示す図である。ここでは、図2と同様に、タイル状の構成要素をプログラム編集画面に敷き詰めて、タイルの種類で軌跡の経過を把握する方法について例示している。図46(a)は、ジェスチャ定義に用いるタイル状のマスの配置関係を示す図であり、図46(b)は、プログラム編集画面のタイル状のマスの配置関係を示す図であり、図46(c)は、図46(a)、(b)の場合のジェスチャ定義の一例を示す図である。なお、ここで、Lには左側下地要素、Mには図形配置可能下地要素、Rには右側下地要素をそれぞれ敷き詰めている。

[0222] この例では、ジェスチャ定義は、入力操作の軌跡の形状とその向きと、構成要素と、その構成要素に適用するコマンドと、を対応付けたものである。ジェスチャの軌跡は、上記したようにプログラム編集画面に定義された座標系に基づいて得られる座標の軌跡となっている。また、この軌跡の描画に対して、ジェスチャ描画領域に描かれるプログラム構成要素が定義されている。粗領域の場合は、図2または図3の場合と同様である。大領域の場合には、プログラム編集画面を領域L、M、Rに分割し、入力操作の軌跡を粗領域の場合よりも大まかに取得することができる。ここでは、複数の領域L、M、Rを用いて軌跡の形状を定義している。すなわち、始点と終点までの向きに加えて、始点から終点に至るまでに経由した領域L、M、Rの座標で判別され、軌跡の向きは、始点から終点に至るまでに経由した領域L、M、Rの順序で判別される。

- [0223] 処理実行手段 12 は、入力ジェスチャ特性情報取得手段 11 で生成作成された入力ジェスチャ特性情報に含まれる大入力軌跡または粗入力軌跡と座標情報に基づいて、軌跡が通過した領域の組み合わせに対応するコマンドを実行し、コマンドを実行した結果を反映した後のプログラムをプログラム編集画面に表示する処理を行う。
- [0224] つぎに、このようなプログラム編集用プログラムでのプログラム編集方法について説明する。実施の形態 2 の図 8 のフローチャートのステップ S33 で操作対象が下地要素である場合には、入力ジェスチャ特性情報取得手段 11 は、大領域を単位とした大入力軌跡を生成し、処理実行手段 12 は、ステップ S33 で、入力ジェスチャ特性情報に基づいて大入力軌跡に対応する軌跡が存在するかを判定すればよい。
- [0225] また、ステップ S33 で操作対象がプログラム構成要素である場合には、入力ジェスチャ特性情報取得手段 11 は、入力ジェスチャ特性情報から粗領域を単位とした粗入力軌跡を生成し、処理実行手段 12 は、粗入力軌跡と操作対象（構成要素）との組み合わせに対応する軌跡が存在するかを判定すればよい。なお、その他の処理については、たとえば図 8 で説明したものと同様であるので、その説明を省略する。
- [0226] 実施の形態 4 では、プログラム編集画面 200A において、左母線 101 よりも左側の領域 L と、右母線 102 よりも右側の領域 R と、左母線 101 と右母線 102 との間の領域 M とに区切り、領域 L から領域 M を介して領域 R に至るジェスチャ（またはその逆の軌跡のジェスチャ）によって、プログラム編集画面 200A に、ラダープログラムの基本構成要素となる接続線 111、またはラダープログラムで一般的に使用される a 接点記号 121、コイル記号 131 およびこれらの間を結ぶ接続線 111 などが配置されるようにした。これによって、プログラムを作成する初期の段階においては、実施の形態 3 のような詳細領域または粗領域よりも大きな領域を単位としてジェスチャの検出を行って、基本的なプログラム構成要素を配置することができる。その結果、プログラムの作成処理にかかる演算負荷を実施の形態 3 に比

して抑えることができるという効果を有する。

[0227] 実施の形態 5.

実施の形態 5 では、命令行内の命令の変更について説明する。

[0228] 実施の形態 5 による入力ジェスチャ特性情報取得手段 11 は、粗入力軌跡と詳細入力軌跡を生成する。

[0229] また、処理実行手段 12 は、入力操作の軌跡が通過した領域の構成要素に命令行が存在する場合に、命令行に関するジェスチャ定義に基づいてコマンドを実行する。具体的には、命令行上で所定の文字を模した軌跡が入力された場合に、ジェスチャ定義で定義されている内容にしたがって、命令行中の命令の変更を行う。

[0230] 命令行に関するジェスチャ定義には、所定の軌跡が描画された場合の処理がさらに規定されている。具体的には、軌跡の形状とその向きに対応して、すでに格納されている命令文の種類に応じて実行するコマンドを切り替えるための定義を有する。

[0231] 図 47 は、実施の形態 5 によるプログラムの編集処理の手順の一例を示す図である。ここでは、(a) に示されるように、「MOV」命令 152a が入力された命令行 132 を有するラダープログラムが配置されている。(b) に示されるように、命令行 132 の「MOV」命令 152a の上で命令行 132 を跨ぐように「D」という入力操作 J が描かれると、(c) に示されるように 16 ビットで演算を行う「MOV」命令 152a が 32 ビットで演算を行う「DMOV」命令 152b に変更される。

[0232] さらに、(d) に示されるように、命令行 132 の「DMOV」命令 152b の上で、命令行 132 を跨ぐように「D」という文字の形に模した入力操作 J2 が行なわれると、(e) に示されるように、「DMOV」命令が「MOV」命令に変更される。

[0233] 実施の形態 5 では、命令行 132 上で所定の入力操作を行うことによって、命令行 132 中の命令文を変更するようにしたので、構成要素に格納されている命令文の変更を容易に行うことができるという効果を有する。

[0234] 実施の形態 6.

この実施の形態 6 では、上記の実施の形態で配置されたプログラム構成要素を変更する場合について説明する。

[0235] 処理実行手段 12 は、プログラム構成要素上で所定の入力操作を行ったときに、そのプログラム構成要素と類似する 1 以上のプログラム構成要素を変更候補として表示する機能をさらに有する。この変更候補は、たとえば形状が類似した記号を集めた変更候補情報から取得される。この変更候補情報は、たとえばジェスチャ定義に含まれる。また、変更候補は、変更候補ウィンドウ内に表示され、この変更候補ウィンドウが、入力操作が行われた位置付近のプログラム編集画面上に表示される。

[0236] 図 48 は、実施の形態 6 による具体的なプログラムの編集処理の手順の一例を示す図である。図 48 (a) には、左母線 101 と右母線 102 との間に、a 接点記号 121 とコイル記号 131 とが配置され、接続線 111 で結ばれたラダープログラムが記載されている。ここで、(b) のように、入力操作 J として、a 接点記号 121 に所定の時間以上入力指示体を当てることで、変更対象となる記号の選択と、変更候補情報の表示と、を指示することになるようジェスチャ定義に定義しておく。a 接点記号 121 上で入力操作 J が行われると、入力ジェスチャ特性情報取得手段 11 は入力ジェスチャ特性情報を生成し、処理実行手段 12 は、ジェスチャ定義に定義されているコマンドに基づいて、a 接点記号 121 に類似する記号を含む変更候補情報をジェスチャ定義から取得する。そして、処理実行手段 12 は、(c) のように取得した変更候補情報に基づいて変更候補ウィンドウ 166 を表示する。この例では、a 接点記号 121 の変更候補情報として、a 接点記号 166 a、b 接点記号 166 b、立ち上がり a 接点記号 166 c、および立ち下がり a 接点記号 166 d が変更候補ウィンドウ 166 に表示される場合が挙げられている。そして、(d) のようにユーザによって、この変更候補ウィンドウ 166 中からいずれか（たとえば b 接点記号 166 b）が選択されると（T1）、(e) のように、ユーザによって選択された a 接点記号 121 が、

ユーザによって変更候補ウィンドウ 166 から選択された b 接点記号 122 に置き換えられる。

[0237] なお、変更候補情報には、構成要素の変更の中止を指示する選択肢を含んでもよい。また、変更候補ウィンドウの外の領域に入力操作が行なわれた場合に変更を中止するようにしてもよい。また、変更候補ウィンドウを表示した後に、予め定めた期間、入力が無い場合に、変更を中止するようにしてもよい。

[0238] この実施の形態 6 では、プログラム構成要素上で所定の入力操作を行うことで、操作対象の構成要素に対応付けられた変更候補を表示するようにしたので、プログラム構成要素の変更を容易に実行することができるという効果を有する。

[0239] 実施の形態 7.

実施の形態 7 では、上記のようにして配置されたプログラム構成要素の移動に関する実施の形態を説明する。

[0240] 処理実行手段 12 は、入力ジェスチャ特性情報取得手段 11 で取得された入力ジェスチャ特性情報を用いて、描かれた軌跡の始点が構成要素の所定の位置にあるかを判定し、構成要素の所定の位置上に軌跡の始点がある場合には、その構成要素をジェスチャの軌跡に沿って移動させる処理を行う。また、構成要素の所定の位置上にジェスチャの始点がない場合には、上記した実施の形態で説明した場合に相当する処理を行う。

[0241] 図 49 は、実施の形態 7 によるプログラム構成要素の移動条件の一例を示す図である。図 49 (a) に示されるように、軌跡の下に配置されているプログラム構成要素が接続線 111 である場合には、所定の位置は接続線 111 上の位置（移動操作領域 R_m）であるものとする。また、図 49 (b) に示されるように、軌跡の下に配置されているプログラム構成要素が接続線 111 以外の記号 120 である場合には、たとえば記号 120 を含む所定の範囲の領域の右上に設けられる移動操作領域 R_m を所定の位置とする。この移動操作領域 R_m を軌跡の始点として入力操作を行うことで、記号の移動が可

能になる。

[0242] 図50～図52は、実施の形態7によるプログラム構成要素の移動方法の一例を示す図である。図50は、接続線を移動させる場合を示している。(a)には、2つのコイル記号131a, 131bが並列回路構成となっているラダープログラムが示されている。この並列部分の母線101, 102に平行な上下方向の接続線112aの位置を左に移動したい場合には、(b)に示されるように、接続線112a上を入力指示体で押さえ、その後、接続線112aを押さえた状態で、左方向に向かって入力指示体を移動させ、移動を終えたい位置で入力指示体を離す入力操作Jを行う。これによって、(c)に示されるように、接続線112aが移動される。ここでは、接続線112aを画面内の左右方向に移動させる場合を示したが、たとえば接続線111に平行な方向の並列回路部分の接続線112bを上下方向に移動させる場合も同様の処理によって行うことができる。

[0243] 図51は、a接点記号を移動する場合を示している。(a)には、接続線111に1つのa接点記号121aが配置され、このa接点記号121aと並列回路を構成するように接続線111と左母線101との間に2つのa接点記号121b, 121cが配置された接続線112が設けられている。このうちのa接点記号121cを移動する場合には、a接点記号121cの右上の移動操作領域Rmを押さえながら、移動したい位置まで入力指示体を移動させるように入力操作Jを行う。そして、移動後に配置したい位置で入力指示体をプログラム編集画面から離すことによって、(b)のようにa接点記号121cが接続線111上に移動することになる。

[0244] また、実施の形態3の図39で、デバイス名の先頭の文字が「Y」であるプログラム構成要素は、出力にも入力にもなることを述べた。そこで、デバイス名の先頭の文字が「Y」であるプログラム構成要素の移動を行う場合には、図52(a)のように、記号の移動先の位置によって記号の種類を変えるようにしてもよい。たとえば、コイル記号131を(a)の入力操作Jのように領域Rb内から領域Ra内へと移動させる場合には、(b)のように

コイル記号 1 3 1 から a 接点記号 1 2 1 b へと変更されて、a 接点記号 1 2 1 b が配置されるようにすることができる。

[0245] さらに、移動操作領域 R_mを押さえながら入力指示体を移動させている場合に、記号を移動させていることがわかるように、移動対象の記号をゆがめさせたり、浮き上がるような形状としたり、移動対象である記号の形を変形させてもよい。

[0246] また、ここでは、移動の場合を説明したが、移動ではなくプログラム構成要素の複製の追加（以下、コピーという）も同様の方法で行うことができる。図 5 3 は、実施の形態 7 によるプログラムの編集の処理手順の一例を示す図である。この図 5 3（a）に示されるように、記号にコピー操作領域 R_cを設け、コピーしたい場合には、入力指示体でコピー操作領域 R_cを押さえながら移動させることにより、押さえつけたプログラム構成要素が複製されたものが移動する。この例では、並列回路を構成する a 接点記号 1 2 1 a の中央のコピー操作領域 R_cを押さえながら、複製した a 接点記号を配置したい位置まで入力指示体を移動させるように入力操作 J を描く。そして、新たな a 接点記号を配置したい位置で入力指示体をプログラム編集画面から離すことによって、図 5 3（b）のように、指定した位置に a 接点記号 1 2 1 b が配置されることになる。

[0247] なお、ここではコピー操作領域 R_cを設ける例を説明したが、この後で説明する実施の形態 8 で示す編集用のボタンとしてコピー操作を指定するための編集ボタンを設けて、その編集ボタンと移動操作領域 R_mの組み合わせを用いることで、コピー操作領域 R_cを設けずに、その編集ボタンを押している状態でプログラム構成要素を移動する入力操作を行うことでプログラム構成要素をコピーできるように構成してもよい。また、同様に、コピー操作を指定するための編集ボタンを設けて、その編集ボタンを押している状態では、プログラム構成要素に対する全ての入力操作が、そのプログラム構成要素をコピーする入力操作として処理されるように構成してもよい。

[0248] この実施の形態 7 では、構成要素の所定の位置を入力指示体を用いて入力

操作を行うことで、プログラム構成要素の再配置またはコピーを容易に行うことができるという効果を有する。

[0249] 実施の形態 8.

上記した実施の形態では、入力指示体による入力操作で、実行するコマンドを特定するようにしていた。実施の形態 8 では、編集ボタンを設け、その編集ボタンの選択の有無と入力操作との組み合わせによって、実行するコマンドを選択できるようにする場合について説明する。

[0250] 実施の形態 8 によるプログラム編集用プログラムが導入されるエンジニアリングツールには、所定の編集機能の許可／不許可を選択可能な編集ボタンが設けられている。編集ボタンとして、たとえば入力操作によるプログラムの変更の許可／不許可を設定する変更許可ボタン、入力操作によるプログラム構成要素の移動の許可／不許可を設定する移動許可ボタン、入力操作によるプログラム構成要素のコピーの許可／不許可を設定するコピー許可ボタン、またはプログラム構成要素の削除の許可／不許可を設定する削除許可ボタンなどを例示することができる。

[0251] 編集ボタンが許可状態の場合には、その編集ボタンに対応付けられた編集処理（変更、移動、削除など）が可能になる。一方、編集ボタンが不許可状態の場合には、その編集ボタンに対応付けられた編集処理（変更、移動、削除など）が不可能になる。この場合、単にその編集処理を実行することができない状態とするだけでなく、編集処理の不許可と入力指示体を用いる入力操作とを対応付けた処理を実行可能な状態とするようにしてもよい。

[0252] その場合、ジェスチャ定義は、構成要素の種類と、軌跡の形状とその向きと、編集ボタンの許可状態または不許可状態と、実行するコマンドと、の組み合わせに対して処理を規定したレコードを格納する。

[0253] 処理実行手段 12 は、入力ジェスチャ特性情報取得手段 11 から取得した入力ジェスチャ特性情報に含まれる座標情報に基づいて操作対象を特定し、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きと、編集ボタンの許可／不許可の状態と、に基づいて、操作対象に関するジェスチャ定義に格

納されているコマンドを実行し、コマンドを実行した後で得られるプログラムをプログラム編集画面に表示する。なお、編集ボタンの許可状態または不許可状態は、編集ボタンの状態を示す編集許可状態情報としてプログラム編集画面に表示してもよい。

[0254] なお、この実施の形態 8 におけるプログラム作成処理の基本的な流れは、ジェスチャ定義における、コマンドを特定するための選択肢として編集ボタンの状態が加わる点を除いて、上記した実施の形態と同様であるので、その説明を省略する。

[0255] 以下では、（１）編集ボタンが不許可状態の場合にプログラムが変更されるコマンドを実行しない場合、（２）編集ボタンが不許可状態の場合に異なるジェスチャを実行するようにジェスチャ定義を行う場合、の編集ボタンを用いたプログラムの編集処理の具体例について説明する。図 5 4 ～図 5 6 は、実施の形態 8 によるプログラムの編集の処理手順の一例を示す図である。

[0256] （１）不許可状態の場合にプログラムが変更されるコマンドを実行しない場合

ここでは、編集ボタンが許可（オン）状態の場合には、上記した実施の形態で説明したプログラム構成要素の配置の変更などの処理を行うことができるが、編集ボタンが不許可（オフ）状態の場合には、プログラム構成要素の配置の変更などの処理を行うことができない状態となる例について示す。

[0257] 図 5 4 （a）のように編集ボタンとしての変更許可ボタン 1 7 1 を押して変更許可状態とした場合は、上記したようにプログラム構成要素の配置を変更する処理などを行うことができるので、プログラム編集画面上でたとえば a 接点記号を配置するための入力操作 J 1 を行うと、（b）のようにその入力操作に対応する a 接点記号 1 2 1 が配置される。

[0258] その後、変更許可ボタン 1 7 1 を押して変更不許可状態とした場合は、ジェスチャ定義にて、変更不許可状態のときに該当するコマンドを実行しない定義を行うことにより、上記したようにプログラム構成要素の配置を変更する処理などを行うことができない。そのため、（c）のようにプログラム編

集画面上でたとえば a 接点記号を配置するための入力操作 J 2 を行っても、
(d) のようにプログラム編集画面には a 接点記号は配置されない。

[0259] また、編集ボタンとして、変更許可ボタンのほかに移動許可ボタンを設けた場合も動作は同様である。この場合には、移動許可ボタンが許可状態の場合にはプログラム構成要素の移動が可能であるが、不許可状態の場合には、移動が起こらない。

[0260] 図 5 5 では、編集ボタンとして削除許可ボタン 1 7 2 を設けた場合を示している。(a) のように、削除許可ボタン 1 7 2 を押して削除許可状態にし、接続線 1 1 1 に対して設けられる並列回路の接続線 1 1 2 をなぞるように入力操作 J を行うと、(b) のように、入力操作 J の下に配置されていた接続線 1 1 2 が削除される。なお、削除許可ボタン 1 7 2 を再度押して削除不許可状態とした場合には、図 5 4 (c)、(d) の場合と同様に、描画されるジェスチャに対応した処理が行われる。

[0261] (2) 不許可状態の場合に異なるジェスチャを実行するようにジェスチャ定義を行う場合

ここでは、編集ボタンを不許可状態として入力操作を行ったときに、編集ボタンを許可状態として入力操作を行ったときとは別のコマンドを実行する例について示す。

[0262] 図 5 6 (a) のように編集ボタンとしての変更許可ボタン 1 7 1 を押して変更許可状態とした場合は、たとえば、入力操作に対応して実行されるコマンドとして、プログラム構成要素を配置するコマンドを定義しておくことで、プログラム編集画面にてたとえば a 接点記号を配置するための入力操作 J 1 を行うと、(b) のようにその入力操作に対応する a 接点記号 1 2 1 が配置される。

[0263] その後、変更許可ボタン 1 7 1 を押して変更不許可状態とした場合に、(c) のようにプログラム編集画面で (a) と同様の入力操作 J 2 を行う。すると、変更許可が不許可状態での入力操作 J 2 に対して、ジェスチャ定義にて、スクロール処理が定義されている場合には、プログラム編集画面に表示

されるプログラムの範囲が移動する。そして、（d）のように、ユーザが入力指示体をプログラム編集画面200から離して入力操作J2を終了させると、スクロール処理が終了する。

[0264] なお、編集ボタンは、1回押すごとに許可と不許可が切替るものであってもよいし、通常は不許可状態で、押し続けることで許可状態とするものであってもよい。このような編集ボタンはタブレット端末の表示部上に表示させるものであってもよいし、タブレット端末のタッチ入力装置が有するマルチタッチ機能を用いて構成してもよい。あるいは、タブレット端末に接続したキーボードの入力キーを用いて構成してもよい。タブレット端末に物理的な操作ボタンを設け、そのボタンを編集ボタンに割り当ててもよい。

[0265] 実施の形態8では、編集ボタンを設け、編集ボタンのオンまたはオフによって、実行されるコマンドを変更できるようにした。これによって、軌跡の形状とその向きの向きが同一の入力操作に割り当てられる処理を増やすことができるという効果を有する。また、削除または移動などの編集処理について、誤ってプログラムを変更することを防止できる効果が得られるため、ユーザによる操作性を向上させることができるという効果も有する。

[0266] 実施の形態9.

上記した実施の形態は、プログラム編集用プログラムでプログラムの編集を行う場合を示した。この実施の形態9では、実際に動作中の制御装置の内部の変数（以下、デバイス変数という）の値などの動作状態をプログラム編集画面に表示しているときに、実行中のデバイス変数の状態を、入力指示体を用いる入力操作によって制御する場合を説明する。

[0267] プログラム編集用プログラムにおいて、入力ジェスチャ特性情報取得手段11が、入力操作から入力ジェスチャ特性情報を取得する処理は、上述した実施の形態と同じである。

[0268] 処理実行手段12は、入力ジェスチャ特性情報に対応するコマンドをジェスチャ定義から取得し、通信回線を介して接続されるプログラマブルコントローラなどの制御装置に対して制御装置の内部のデバイス変数の値を変更す

るコマンドを実行する処理を行う。たとえば、ラダープログラム中の a 接点記号がオン状態にあるときに、デバイス変数の状態を変更するコマンドに対応する入力操作をユーザが行ったときには、実際にそのラダープログラムが実行されている制御装置において、対応する a 接点記号が指しているデバイス変数の状態をオフに変更し、プログラム編集画面の a 接点記号をオフ状態に変更する機能を有する。

[0269] ここでは、操作対象に関するジェスチャ定義に、デバイス変数の状態を変更するコマンドを定義する場合を説明したが、操作対象に関するジェスチャ定義に、デバイス変数の状態がオンのときに実行するコマンドと、デバイス変数の状態がオフのときに実行するコマンドをそれぞれ定義して、処理実行手段 12 が、デバイス状態の状態に応じて実行するコマンドを切り替えるように構成してもよい。

[0270] 図 57 は、実施の形態 9 によるプログラム編集用プログラムが行う処理の手順の一例を示すフローチャートである。ここでは、制御装置で実行中のラダープログラムがプログラム編集画面に表示されており、ラダープログラムのプログラム構成要素に、そのプログラム構成要素が指すデバイス変数の、制御装置内部での現在の状態が合せて表示されており、制御装置に指令を送ることでデバイス変数の状態を変更できる状態（以下、テストモード状態という）にあるものとする。

[0271] まず、入力ジェスチャ特性情報取得手段 11 は、入力操作に基づいて入力ジェスチャ特性情報を生成する（ステップ S111）。なお、入力ジェスチャ特性情報には、上記したように、入力指示体のプログラム編集画面上における軌跡の形状とその向きのほかに、その軌跡が描画された座標情報も含まれる。

[0272] ついで、処理実行手段 12 は、入力ジェスチャ特性情報に基づいて操作対象を特定する（ステップ S112）。その後、処理実行手段 12 は、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きが操作対象に関するジェスチャ定義に存在するかを判定する（ステップ S113）。

- [0273] 入力ジェスチャ特性情報に含まれる軌跡の形状とその向きが操作対象に関するジェスチャ定義に存在する場合（ステップS 1 1 2でY e sの場合）には、処理実行手段1 2は、入力ジェスチャ特性情報の軌跡の形状とその向きに対応するコマンドを実行する（ステップS 1 1 4）。すなわち、処理実行手段1 2は、通信部を介して制御装置に対して、操作対象として指定されたプログラム構成要素の状態を変更する指令を実行し（ステップS 1 1 5）、ついで、処理実行手段1 2は、操作対象の表示の状態を、制御装置内部での現在の状態に合わせて変更し（ステップS 1 1 6）、処理が終了する。
- [0274] また、ステップS 1 1 3で、入力ジェスチャ特性情報に含まれる軌跡の形状とその向きが操作対象のジェスチャ定義に存在しない場合（ステップS 1 1 3でN oの場合）には、処理実行手段1 2は、コマンドを何も実行せずに処理が終了する。
- [0275] 以下に、具体例を挙げて、テストモード状態にしてあるプログラム編集画面を用いた編集処理について説明する。図5 8と図5 9は、実施の形態9によるテストモード状態における処理の手順の一例を示す図である。
- [0276] 図5 8は、テストモード状態においてa接点の状態をオフからオンにする場合である。（a）に示されるように、プログラム編集画面にオフ状態のa接点記号1 2 1 aと、コイル記号1 3 1を有するラダープログラムが表示されている。この状態で、（b）のようにa接点記号1 2 1 aの左上の領域から右下の領域にかけて入力指示体を移動する入力操作Jが行われたとする。その結果、通信部を介してa接点記号1 2 1 a（1 2 1 b）に対応するデバイス名M 0の接点がオン状態となるように指示がプログラム編集用プログラムを実行しているタブレット端末から制御装置に出される。そして、（c）に示されるように、オフ状態のa接点記号1 2 1 aはオン状態のa接点記号1 2 1 bに切り替わる。
- [0277] 図5 9は、テストモード状態においてa接点の状態をオンからオフにする場合の例を示している。（a）に示されるように、プログラム編集画面にオン状態のa接点記号1 2 1 bと、コイル記号1 3 1を有するラダープログラ

ムが表示されている。この状態で、(b)のようにa接点記号1 2 1 bの左上の領域から右下の領域にかけて入力指示体を移動する入力操作Jが行われたとする。その結果、(c)に示されるように、通信部を介してa接点記号1 2 1 b (1 2 1 a)に対応するデバイス名M0の接点がオフ状態となるように指示が、プログラム編集用プログラムを実行しているタブレット端末から制御装置に出される。そして、オン状態のa接点記号1 2 1 bはオフ状態のa接点記号1 2 1 aに切り替わる。

[0278] この実施の形態9では、プログラム編集画面で制御装置の内部のデバイス変数の状態をラダープログラムに重ね合わせて表示している際に、プログラム構成要素上に入力指示体を用いる入力操作を行うと、そのプログラム構成要素に対応するデバイス変数の状態を変更する指示を出すとともに、動作の表示状態を変更するようにした。これによって、制御装置のテストモード状態中に、制御装置内部のプログラム構成要素の状態を容易に切り替えることができるという効果を有する。

[0279] なお、たとえば、右側下地要素のジェスチャ定義に、制御装置のテストモードなどの動作状態を変更するコマンドを定義して、入力操作によって制御装置の動作状態を変更できるように構成してもよい。

[0280] 実施の形態10.

この実施の形態10では、プログラム編集に関するヘルプ情報を表示することができるプログラム編集用プログラムについて説明する。

[0281] 処理実行手段12は、入力ジェスチャ特性情報取得手段11で取得された入力ジェスチャ特性情報がヘルプ情報を表示するコマンドである場合には、ジェスチャ定義に格納されているコマンドに基づいて、ヘルプ情報をヘルプ情報表示ウィンドウに表示させる処理を行う。ヘルプ情報は、プログラムの編集方法、あるいは記号を配置するための軌跡の形状とその向きの一覧などを含む。また、ヘルプ情報は、記号ごとの入力操作方法に関するアニメーションをファイル化したものを含んでいてもよい。

[0282] なお、ジェスチャ定義には、ヘルプ情報を表示させるコマンドに対応付け

たジェスチャの定義がさらに含まれている。なお、この実施の形態 10 におけるプログラム編集処理の基本的な流れは、上記した実施の形態と同様であるので、その説明を省略する。

[0283] 以下では、ヘルプ情報の表示処理の具体例について説明する。図 60 は、実施の形態 10 によるヘルプ情報表示の処理手順の一例を示す図である。図 60 では、ユーザは、a 接点記号を配置したいがどのようにジェスチャ入力したらよいのかわからない場合を例に挙げる。

[0284] まず、(a) のように、ユーザは、ヘルプ情報を表示させる入力操作 J をプログラム編集画面上で行う。ついで、入力ジェスチャ特性情報取得手段 11 は入力ジェスチャ特性情報を取得し、処理実行手段 12 はジェスチャ定義に基づいて (b) のようにヘルプ情報検索ウィンドウ 185 をジェスチャ描画領域の近傍に表示する。このヘルプ情報検索ウィンドウ 185 には、検索したい内容を入力することができる検索条件入力領域 185 a と検索ボタン 185 b が設けられている。ここでは、ユーザは検索条件入力領域 185 a に「a 接点」と入力し、検索ボタン 185 b を押したものとする。

[0285] すると、(c) のように a 接点記号の配置方法を示すヘルプ情報が検索され、ヘルプ情報表示ウィンドウ 186 に表示される。ユーザは、このヘルプ情報表示ウィンドウ 186 を参照しながら、プログラムの編集を行うことができる。

[0286] この実施の形態 10 によれば、ヘルプ情報の表示を入力指示体を用いる入力操作に対応付けたので、プログラムの編集に慣れないユーザ、またはプログラムの編集集中に操作がわからなくなったユーザなどに対してプログラムの編集方法の説明を提供できるという効果を有する。

[0287] 実施の形態 11.

ラダープログラムの編集では、デバイス名を配置せず、記号のみを配置してラダープログラムを作成した後、各記号に対してデバイス名を設定することがある。このような場合でのデバイス名の設定方法について説明する。

[0288] 図 61 は、実施の形態 11 によるプログラム編集用プログラムの機能構成

の一例を模式的に示すブロック図である。このプログラム編集用プログラム 10Bは、実施の形態1の構成に、デバイス名入力処理手段14と、をさらに備える。

[0289] デバイス名入力処理手段14は、ユーザによって入力されたデバイス名をデバイス名入力順序にしたがって、各記号にデバイス名を入力する。デバイス名入力順序は、プログラム編集画面に表示されたラダープログラム中の各記号について、デバイス名を入力する順序を規定する情報である。デバイス名入力順序として、たとえば、上の接続線から下の接続線に向かって、また各接続線では左母線から右母線に向かう順序とすることができる。なお、このデバイス名入力順序はユーザによって任意に設定することができる。

[0290] なお、上記した実施の形態と同一の構成要素には同一の符号を付してその説明を省略している。また、この実施の形態11におけるプログラム編集処理の基本的な流れは、上記した実施の形態と同様であるので、その説明を省略する。

[0291] 以下では、デバイス名の入力処理の具体例について説明する。図62は、実施の形態11によるデバイス名入力処理の手順の一例を示す図である。（a）のようなデバイス名が入力されていないラダープログラムがユーザによって作成される。ついで、（b）のようにデバイス名が、たとえばタブレット端末に接続されたキーボードを用いてキー入力される。ここではデバイス名とデバイス名との間には、区切りを示す「エンター」キーが入力されるものとする。デバイス名の入力終了後、デバイス名入力処理手段14は、（b）のデバイス名とデバイス名入力順序とから、ラダープログラムの各記号のデバイス名入力領域に（b）で入力したデバイス名を入力する。ここでは、デバイス名入力順序を上で説明した順序であるとする、（c）のように一番上の接続線111の左側のa接点記号121aから一番下の接続線113の右側のコイル記号131cに向かって順に入力されたデバイス名が配置される。

[0292] なお、上記した例では、タブレット端末に接続されたキーボードを用いて

デバイス名を入力する場合を示したが、これに限定されるものではない。たとえば、タブレット端末のタッチパネルにソフトウェアキーボードを表示し、そのソフトウェアキーボードを用いてキー入力するようにしてもよい。あるいは、音声によって、図62(b)のようにデバイス名と区切り記号を入力し、その入力された音声を識別してデバイス名入力順序にしたがってラダープログラムのプログラム構成要素にデバイス名を設定するようにしてもよい。

[0293] この実施の形態11によれば、プログラム構成要素に与える文字列を、キーボードなどの文字入力に向けた装置を用いて、画面へのタッチ操作などのほかの入力操作を途中にはさむことなく、連続して入力することが可能になるため、ユーザがプログラム構成要素に文字列を効率よく指定できるという効果を有する。

[0294] なお、上述した実施の形態では、入力操作から入力軌跡を取得する際の解像度を、プログラム編集画面に一樣に粗入力軌跡、詳細入力軌跡または大入力領域に関する軌跡を複数生成する場合を示した。しかし、この発明がこれに限定されるものではない。図63は、入力操作から入力ジェスチャ特性情報の座標情報を取得する方法の他の例を示す図である。図63(a)に示されるように、プログラム編集画面に左母線101と右母線102とそれらの間を接続する接続線111とが描かれると、接続線111よりも上の所定の領域と下の所定の領域を詳細領域R_fとし、詳細領域R_fに挟まれた領域を粗領域R_cとしている。そして、入力操作の軌跡がどの領域で描かれるかによって、入力軌跡の解像度を変えるようにしてもよい。すなわち、入力ジェスチャ特性情報取得手段11は詳細領域でジェスチャが描かれる場合には、詳細入力軌跡を生成し、粗領域でジェスチャが描かれる場合には、粗入力軌跡を生成する。

[0295] また、図63(a)の状態からプログラム構成要素が配置され、たとえば図63(b)のように接続線111に2つのa接点記号121a, 121bとコイル記号131がプログラム構成要素として配置された場合には、プロ

グラム構成要素の上と下の所定の領域を詳細領域 R_f とし、それ以外の領域を粗領域 R_c としてもよい。これによって、入力ジェスチャ特性情報取得手段 11 は詳細領域に入力操作が行われる場合には、詳細入力軌跡を生成し、粗領域に入力操作が行われる場合には、粗入力軌跡を生成する。

[0296] また、詳細領域と粗領域を重ね合わせて定義し、一方の領域に入力が始まった後に、もう一方の領域に軌跡が移ったと判定する境界が、もう一方の領域に重なるように構成してもよい。そうすることで、例えば、詳細領域に入力する文字に模した軌跡をより大きな軌跡で描くことが可能になり、ユーザの操作性を向上することが可能になる。

[0297] また、上述した実施の形態では、入力ジェスチャ特性情報には、軌跡に関する情報として、軌跡の形状とその向きを格納していたが、これに、軌跡の筆圧の情報を加えたり、形状、向き、筆圧の情報の3種の情報のうち、1つまたは2つを用いるように構成してもよい。すなわち、たとえば、軌跡に関する情報として、軌跡の形状のみを用いるように構成してもよい。

[0298] なお、上述した実施の形態では、プログラム編集用プログラムを実行する情報処理装置としてタブレット端末を例に挙げて説明したが、マウスなどのポインティングデバイスを有するパーソナルコンピュータ、または表示部にタッチパネルが設けられ、タブレット端末に比して表示部のサイズが小さい携帯情報端末もしくは多機能携帯電話などの情報処理装置を用いることもできる。

[0299] また、上述した実施の形態では、ラダープログラムを作成する場合を例に挙げたが、このほかにもFB、FBD、SFCなどの図形を用いたプログラミングに対して上記した実施の形態を適用することができる。

[0300] なお、上述したプログラム編集用プログラムでのプログラム編集方法および制御装置の制御方法は、その処理手順を格納したプログラムとして構成することができる。そして、このプログラムを上記した情報処理装置で実行させることによって実現することができる。また、このプログラムは、ハードディスク、SSD (Solid State Drive)、CD (Compact Disk) - RO

M (Read Only Memory)、MO (Magneto-Optical disk)、DVD (Digital Versatile DiskまたはDigital Video Disk)、メモリカードなどの、情報処理装置で読取可能な記録媒体に記録される。この場合、情報処理装置によって記録媒体からプログラムが読み出されるようになる。また、このプログラムは、インターネットなどのネットワーク（通信回線）を介して配布することもできる。

産業上の利用可能性

[0301] 以上のように、この発明にかかるプログラム編集用プログラムは、タブレット端末において、図形を用いて記述されるプログラムを、より少ない労力で編集できる効果を有するとともに、プログラム編集画面を有効に利用できるという効果、および、プログラムの編集方法をより少ない期間で習得できるという効果を有しているので、プログラマブルコントローラなどの図形を用いたプログラミングに有用である。

符号の説明

[0302] 10、10B プログラム編集用プログラム、11 入力ジェスチャ特性情報取得手段、12 処理実行手段、14 デバイス名入力処理手段、101 左母線、102 右母線、111、111a～111c、112、112a～112c、113 接続線、120 記号、121、121a～121c a接点記号、122 b接点記号、123 立ち下がりa接点記号、131、131a～131c コイル記号、132 命令行、132a 立ち下がり命令行、151、151a、151b デバイス名、161 選択ウィンドウ、162 文字入力領域、163 文字入力ウィンドウ、164 文字入力操作受付領域、165 ガイド線、166 変更候補ウィンドウ、171 変更許可ボタン、172 削除許可ボタン、180、200、200A プログラム編集画面、185 ヘルプ情報検索ウィンドウ、185a 検索条件入力領域、185b 検索ボタン、186 ヘルプ情報表示ウィンドウ、210 文字入力ウィンドウ、220 文字入力領域。

請求の範囲

- [請求項1] プログラム編集画面上で図形を用いて記述されるプログラムを編集するプログラム編集用プログラムにおいて、
- コンピュータを
- プログラム編集画面上に入力される座標情報、前記座標情報の軌跡の形状を含む入力ジェスチャ特性情報を取得する入力ジェスチャ特性情報取得手段と、
- 入力ジェスチャ特性情報に基づいて、操作対象を特定するとともに、前記操作対象に適用するコマンドも特定して、前記コマンドを実行し、プログラム編集画面に表示する処理実行手段と、
- して機能させるためのプログラム編集用プログラム。
- [請求項2] 前記処理実行手段は、前記操作対象の種類と、前記軌跡の形状と、実行するコマンドと、を対応付けたジェスチャ定義に基づいて、前記コマンドを特定することを特徴とする請求項1に記載のプログラム編集用プログラム。
- [請求項3] 前記入力ジェスチャ特性情報は、前記座標情報の軌跡の向きを含むことを特徴とする請求項1に記載のプログラム編集用プログラム。
- [請求項4] 前記処理実行手段は、前記操作対象の種類と、前記軌跡の形状および向きと、実行するコマンドと、を対応付けたジェスチャ定義に基づいて、前記コマンドを特定することを特徴とする請求項3に記載のプログラム編集用プログラム。
- [請求項5] 前記処理実行手段は、前記入力ジェスチャ特性情報に含まれる情報に基づいて入力文字列を取得し、前記入力文字列に対応するコマンドが前記ジェスチャ定義に存在する場合に、前記操作対象に前記入力文字列を登録することを特徴とする請求項2または4に記載のプログラム編集用プログラム。
- [請求項6] 前記入力ジェスチャ特性情報取得手段は、第1解像度での前記座標情報の始点から終点までの通過点の記録である第1入力軌跡と、前記

第1解像度よりも高い第2解像度での前記座標情報の前記始点から前記終点までの前記通過点の記録である第2入力軌跡と、を取得し、

前記処理実行手段は、前記ジェスチャ定義に、前記第1入力軌跡に対応するコマンドが存在するか確認するとともに、前記第2入力軌跡に対応するコマンドが存在するか確認することを特徴とする請求項2または4に記載のプログラム編集用プログラム。

[請求項7] 前記処理実行手段は、前記第2入力軌跡に基づいて入力文字列を取得し、前記入力文字列に対応するコマンドが前記ジェスチャ定義に存在する場合に、前記操作対象に前記入力文字列を登録することを特徴とする請求項6に記載のプログラム編集用プログラム。

[請求項8] 前記処理実行手段は、前記操作対象が文字入力用の構成要素であるか否かを判定し、

前記入力ジェスチャ特性情報取得手段は、前記操作対象が文字入力用の構成要素でないと判断された場合に、第1解像度での前記座標情報の始点から終点までの通過点の記録である第1入力軌跡を取得し、前記操作対象が文字入力用の構成要素であると判断された場合に、前記第1解像度よりも高い第2解像度での前記座標情報の前記始点から前記終点までの前記通過点の記録である第2入力軌跡を取得することを特徴とする請求項1から4のいずれか1つに記載のプログラム編集用プログラム。

[請求項9] 前記処理実行手段は、前記第2入力軌跡が取得された場合に、前記第2入力軌跡に基づいて入力文字列を取得し、前記操作対象に前記入力文字列を登録することを特徴とする請求項8に記載のプログラム編集用プログラム。

[請求項10] 前記入力ジェスチャ特性情報取得手段は、前記プログラム編集画面に左母線と右母線のみが配置されている場合に、前記第1解像度よりも低い第3解像度での前記座標情報の前記始点から前記終点までの前記通過点の記録である第3入力軌跡を取得することを特徴とする請求

項 1 から 9 のいずれか 1 つに記載のプログラム編集用プログラム。

[請求項11] 前記処理実行手段は、前記座標情報に基づいて前記操作対象を特定する際に、前記操作対象の候補が複数存在する場合に、前記複数の操作対象の候補を一覧表示する機能を有することを特徴とする請求項 1 から 10 のいずれか 1 つに記載のプログラム編集用プログラム。

[請求項12] 前記処理実行手段は、前記複数の操作対象の候補から前記操作対象を選択できる機能を有することを特徴とする請求項 11 に記載のプログラム編集用プログラム。

[請求項13] 前記処理実行手段は、前記操作対象の種類の変更の際に、前記種類の選択肢を一覧表示する機能と、前記選択肢から前記種類を選択できる機能と、を有することを特徴とする請求項 1 から 12 のいずれか 1 つに記載のプログラム編集用プログラム。

[請求項14] 前記プログラム編集画面には、入力操作の軌跡が表示されることを特徴とする請求項 1 から 13 のいずれか 1 つに記載のプログラム編集用プログラム。

[請求項15] 前記プログラム編集画面には、入力操作が可能な範囲を示すガイド線が表示されることを特徴とする請求項 1 から 14 のいずれか 1 つに記載のプログラム編集用プログラム。

[請求項16] 前記処理実行手段は、前記軌跡の始点が前記操作対象の所定の位置にある場合に、前記操作対象を入力操作に沿って移動させることを特徴とする請求項 1 から 15 のいずれか 1 つに記載のプログラム編集用プログラム。

[請求項17] 前記処理実行手段は、前記既配置要素が移動状態にある場合には、前記既配置要素の表示状態を変更する処理を行うことを特徴とする請求項 16 に記載のプログラム編集用プログラム。

[請求項18] 前記処理実行手段は、プログラム編集可否の情報に合せて実行するコマンドを変更できることを特徴とする請求項 1 から 17 のいずれか 1 つに記載のプログラム編集用プログラム。

- [請求項19] 前記処理実行手段は、制御装置の入出力の状態を変更できることを特徴とする請求項 1 から 18 のいずれか 1 つに記載のプログラム編集用プログラム。
- [請求項20] 前記処理実行手段は、前記ジェスチャ定義についてのヘルプ情報の表示を指示するコマンドを含むことを特徴とする請求項 1 から 19 のいずれか 1 つに記載のプログラム編集用プログラム。
- [請求項21] 前記ジェスチャ定義には、前記操作対象の種類として命令文を含むプログラム構成要素が定義され、前記コマンドとして前記命令文を含むプログラム構成要素の前記命令文の変更が定義され、
- 前記処理実行手段は、前記操作対象の種類が前記命令文を含むプログラム構成要素である場合に、前記入力ジェスチャ特性情報と前記ジェスチャ定義に基づいて、前記プログラム構成要素に既に格納されている前記命令文を変更することを特徴とする請求項 2、4 から 20 のいずれか 1 つに記載のプログラム編集用プログラム。