

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2015-141489

(P2015-141489A)

(43) 公開日 平成27年8月3日(2015.8.3)

(51) Int.Cl. F I テーマコード (参考)  
**G06F 9/44 (2006.01)** G06F 9/06 620E 5B376

審査請求 未請求 請求項の数 2 O L (全 11 頁)

(21) 出願番号 特願2014-13017 (P2014-13017)  
 (22) 出願日 平成26年1月28日 (2014.1.28)

(71) 出願人 000005496  
 富士ゼロックス株式会社  
 東京都港区赤坂九丁目7番3号  
 (72) 発明者 西澤 光之  
 神奈川県横浜市西区みなとみらい六丁目1番 富士ゼロックスアドバンステクノロジー株式会社内  
 (72) 発明者 野原 有人  
 神奈川県横浜市西区みなとみらい六丁目1番 富士ゼロックスアドバンステクノロジー株式会社内  
 Fターム(参考) 5B376 BC06 BC08 BC33 BC44

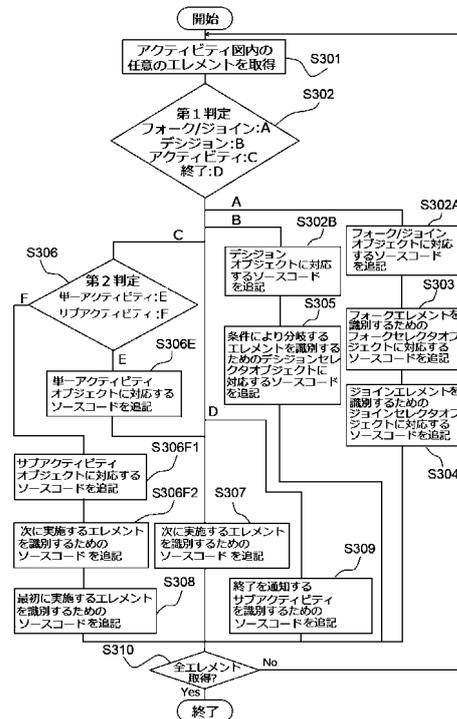
(54) 【発明の名称】 ソースコード作成装置、ソースコード作成プログラム

(57) 【要約】

【課題】 1つのフォークでフォークされた処理は、1つのジョインでジョインしなければならない制約をなくしたソースコード作成装置及びソースコード作成プログラムを提供する。

【解決手段】 アクティビティ図内のエレメントが待ち合わせ/分岐処理、選択処理、実行処理、終了処理のいずれであるかを判定し、待ち合わせ/分岐処理エレメントと選択処理エレメントを識別してソースコード追記する追記手段、分岐するエレメントに対応するソースコードを追記する追記手段、アクティビティエレメントが、単一アクティビティかサブアクティビティかを判定して次に実施するエレメントに対応するソースコードを追記する追記手段、最初に実施するアクティビティに対応するソースコードを追記する追記手段、終了と判定されたエレメントが、終了を通知するサブアクティビティに対応するソースコードを追記する追記手段とを有する。

【選択図】 図4



**【特許請求の範囲】****【請求項 1】**

待ち合わせ / 分岐処理、選択処理、実行処理、及び終了処理を含んで構成されるアクティビティ図内の任意のエレメントを取得する取得手段と、

前記取得手段により取得された前記エレメントが待ち合わせ / 分岐処理、選択処理、実行処理、終了処理のいずれであるかを判定 ( R 1 ) する第 1 の判定手段と、

前記第 1 の判定手段により判定された結果に対応するソースコードを予め定められたファイルに追記する第 1 の追記手段と、

前記第 1 の判定手段により待ち合わせ / 分岐処理と判定された場合、前記エレメントが、待ち合わせ完了後に分岐するエレメントを識別 ( R 2 ) するためのフォークセクタに対応するソースコードを予め定められたファイルに追記する第 2 の追記手段と、

前記第 1 の判定手段により待ち合わせ / 分岐処理と判定された場合、前記エレメントが、待ち合わせするエレメントを識別 ( R 3 ) するためのジョインセクタに対応するソースコードを予め定められたファイルに追記する第 3 の追記手段と、

前記第 1 の判定手段により選択処理と判定された前記エレメントが、条件により分岐するエレメントを識別 ( R 4 ) するためのデシジョンセクタに対応するソースコードを予め定められたファイルに追記する第 4 の追記手段と、

前記第 1 の判定手段によりアクティビティと判定された場合、単一アクティビティかサブアクティビティかを判定 ( R 5 ) する第 2 の判定手段と、

前記第 2 の判定手段によりサブアクティビティと判定された場合、前記エレメントが、最初を実施するアクティビティを識別 ( R 7 ) するためのソースコードを予め定められたファイルに追記する第 6 の追記手段と、

前記第 1 の判定手段によりアクティビティと判定された場合、前記エレメントが、次に実施するエレメントを識別 ( R 6 ) するためのソースコードを予め定められたファイルに追記する第 5 の追記手段と、

前記第 1 の判定手段により終了と判定された場合、前記エレメントが終了を通知するサブアクティビティを識別 ( R 8 ) し、前記識別されたサブアクティビティがある場合は通知先のサブアクティビティを識別するソースコードを追記し、ない場合は追記しない第 8 の追記手段と、を有する、

ことを特徴とするソースコード作成装置。

**【請求項 2】**

コンピュータを、

待ち合わせ / 分岐処理、選択処理、実行処理、及び終了処理を含んで構成されるアクティビティ図内の任意のエレメントを取得する取得手段と、

前記取得手段により取得された前記エレメントが待ち合わせ / 分岐処理、選択処理、実行処理、終了処理のいずれであるかを判定 ( R 1 ) する第 1 の判定手段と、

前記第 1 の判定手段により判定された結果に対応するソースコードを予め定められたファイルに追記する第 1 の追記手段と、

前記第 1 の判定手段により待ち合わせ / 分岐処理と判定された場合、前記エレメントが、待ち合わせ完了後に分岐するエレメントを識別 ( R 2 ) するためのフォークセクタに対応するソースコードを予め定められたファイルに追記する第 2 の追記手段と、

前記第 1 の判定手段により待ち合わせ / 分岐処理と判定された場合、前記エレメントが、待ち合わせするエレメントを識別 ( R 3 ) するためのジョインセクタに対応するソースコードを予め定められたファイルに追記する第 3 の追記手段と、

前記第 1 の判定手段により選択処理と判定された前記エレメントが、条件により分岐するエレメントを識別 ( R 4 ) するためのデシジョンセクタに対応するソースコードを予め定められたファイルに追記する第 4 の追記手段と、

前記第 1 の判定手段によりアクティビティと判定された場合、単一アクティビティかサブアクティビティかを判定 ( R 5 ) する第 2 の判定手段と、

前記第 2 の判定手段によりサブアクティビティと判定された場合、前記エレメントが、

10

20

30

40

50

最初に実施するアクティビティを識別 ( R 7 ) するためのソースコードを予め定められたファイルに追記する第 6 の追記手段と、

前記第 1 の判定手段によりアクティビティと判定された場合、前記エレメントが、次に実施するエレメントを識別 ( R 6 ) するためのソースコードを予め定められたファイルに追記する第 5 の追記手段と、

前記第 1 の判定手段により終了と判定された場合、前記エレメントが終了を通知するサブアクティビティを識別 ( R 8 ) し、前記識別されたサブアクティビティがある場合は通知先のサブアクティビティを識別するソースコードを追記し、ない場合は追記しない第 8 の追記手段として機能させる、

ことを特徴とするソースコード作成プログラム。

10

【発明の詳細な説明】

【技術分野】

【 0 0 0 1 】

本発明は、ソースコード作成装置、ソースコード作成プログラムに関する。

【背景技術】

【 0 0 0 2 】

単一処理かそれ以外の処理 ( 複合処理 ) かを判定する第 1 判定処理、終了したことを通知するための複合処理が存在するか否かを判定する第 3 判定処理、複合処理と判定した場合に、フォークジョイン処理かサブ処理かを判定する第 2 判定処理、サブ処理と判定した場合に、最初に行う処理を認識する第 1 認識処理、サブ処理と判定した場合に、いずれの処理が実行中であることを認識する第 2 認識処理、実行中の処理の次に実行する処理が存在するか否かを判定する第 4 判定処理、サブ処理を実行した際に実行中の処理の次に実行する複数の処理が存在する場合に、いずれの処理を次に実行するかを判定する第 5 判定処理、フォークジョイン処理においてフォークする複数の処理を認識する第 3 認識処理、及びフォークした各処理の終了をジョインさせるためのジョイン処理をソースコードに追記するソースコード作成装置が知られている ( 特許文献 1 ) 。

20

【先行技術文献】

【特許文献】

【 0 0 0 3 】

【特許文献 1】特開 2 0 1 0 - 1 9 8 4 5 7 号公報

30

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 4 】

先行技術では、アクティビティ図のフォークとジョインを実現するために、フォークにより分岐した処理を、ジョインにより終了を待ち合わせするという手段 ( 同期処理 ) を用いた。これにより、1つのフォークでフォークされた処理は、1つのジョインでジョインしなければならない制約があった。

本発明は、この制約をなくしたソースコード作成装置及びソースコード作成プログラムを提供することを目的とする。

【課題を解決するための手段】

40

【 0 0 0 5 】

前記課題を解決するために、請求項 1 に記載のソースコード作成装置は、

待ち合わせ / 分岐処理、選択処理、実行処理、及び終了処理を含んで構成されるアクティビティ図内の任意のエレメントを取得する取得手段と、

前記取得手段により取得された前記エレメントが待ち合わせ / 分岐処理、選択処理、実行処理、終了処理のいずれであるかを判定 ( R 1 ) する第 1 の判定手段と、

前記第 1 の判定手段により判定された結果に対応するソースコードを予め定められたファイルに追記する第 1 の追記手段と、

前記第 1 の判定手段により待ち合わせ / 分岐処理と判定された場合、前記エレメントが、待ち合わせ完了後に分岐するエレメントを識別 ( R 2 ) するためのフォークセクタに

50

対応するソースコードを予め定められたファイルに追記する第2の追記手段と、

前記第1の判定手段により待ち合わせ/分岐処理と判定された場合、前記エレメントが、待ち合わせするエレメントを識別(R3)するためのジョインセクタに対応するソースコードを予め定められたファイルに追記する第3の追記手段と、

前記第1の判定手段により選択処理と判定された前記エレメントが、条件により分岐するエレメントを識別(R4)するためのデシジョンセクタに対応するソースコードを予め定められたファイルに追記する第4の追記手段と、

前記第1の判定手段によりアクティビティと判定された場合、単一アクティビティかサブアクティビティかを判定(R5)する第2の判定手段と、

前記第2の判定手段によりサブアクティビティと判定された場合、前記エレメントが、最初に実施するアクティビティを識別(R7)するためのソースコードを予め定められたファイルに追記する第6の追記手段と、

前記第1の判定手段によりアクティビティと判定された場合、前記エレメントが、次に実施するエレメントを識別(R6)するためのソースコードを予め定められたファイルに追記する第5の追記手段と、

前記第1の判定手段により終了と判定された場合、前記エレメントが終了を通知するサブアクティビティを識別(R8)し、前記識別されたサブアクティビティがある場合は通知先のサブアクティビティを識別するソースコードを追記し、ない場合は追記しない第8の追記手段と、を有する、

ことを特徴とする。

#### 【0006】

前記課題を解決するために、請求項2に記載のソースコード作成プログラムは、コンピュータを、

待ち合わせ/分岐処理、選択処理、実行処理、及び終了処理を含んで構成されるアクティビティ図内の任意のエレメントを取得する取得手段と、

前記取得手段により取得された前記エレメントが待ち合わせ/分岐処理、選択処理、実行処理、終了処理のいずれであるかを判定(R1)する第1の判定手段と、

前記第1の判定手段により判定された結果に対応するソースコードを予め定められたファイルに追記する第1の追記手段と、

前記第1の判定手段により待ち合わせ/分岐処理と判定された場合、前記エレメントが、待ち合わせ完了後に分岐するエレメントを識別(R2)するためのフォークセクタに対応するソースコードを予め定められたファイルに追記する第2の追記手段と、

前記第1の判定手段により待ち合わせ/分岐処理と判定された場合、前記エレメントが、待ち合わせするエレメントを識別(R3)するためのジョインセクタに対応するソースコードを予め定められたファイルに追記する第3の追記手段と、

前記第1の判定手段により選択処理と判定された前記エレメントが、条件により分岐するエレメントを識別(R4)するためのデシジョンセクタに対応するソースコードを予め定められたファイルに追記する第4の追記手段と、

前記第1の判定手段によりアクティビティと判定された場合、単一アクティビティかサブアクティビティかを判定(R5)する第2の判定手段と、

前記第2の判定手段によりサブアクティビティと判定された場合、前記エレメントが、最初に実施するアクティビティを識別(R7)するためのソースコードを予め定められたファイルに追記する第6の追記手段と、

前記第1の判定手段によりアクティビティと判定された場合、前記エレメントが、次に実施するエレメントを識別(R6)するためのソースコードを予め定められたファイルに追記する第5の追記手段と、

前記第1の判定手段により終了と判定された場合、前記エレメントが終了を通知するサブアクティビティを識別(R8)し、前記識別されたサブアクティビティがある場合は通知先のサブアクティビティを識別するソースコードを追記し、ない場合は追記しない第8の追記手段として機能させる、

10

20

30

40

50

ことを特徴とする。

【発明の効果】

【0007】

請求項1に記載の発明によれば、待ち合わせ/分岐処理、選択処理、及び実行処理を含むエレメントで構成されるアクティビティ図を示すアクティビティ情報、及び前記エレメントに対応するソースコードからアクティビティ図が示す処理を行なうためのソースコードを制約なく作成するソースコード作成装置を提供することができる。

請求項2に記載の発明によれば、待ち合わせ/分岐処理、選択処理、及び実行処理を含むエレメントで構成されるアクティビティ図を示すアクティビティ情報、及び前記エレメントに対応するソースコードからアクティビティ図が示す処理を行なうためのソースコードを制約なく作成するソースコード作成プログラムを提供することができる。

10

【図面の簡単な説明】

【0008】

【図1】ソースコード作成装置1の機能構成の一例を示すブロック図である。

【図2】(a)は待ち合わせ/分岐処理の一例、(b)は選択処理の一例を示す図である。

【図3】実施形態に係るモデルを示すクラス図である。

【図4】ソースコード作成装置1によるソースコード作成の流れを示すフローチャートである。

【図5】アクティビティ図で記述された動作からソースコードを作成できない処理のアクティビティ図の一例を示す図である。

20

【図6】先行技術のモデルを示すクラス図である。

【発明を実施するための形態】

【0009】

次に図面を参照しながら、以下に実施形態及び具体例を挙げ、本発明を更に詳細に説明するが、本発明はこれらの実施形態及び具体例に限定されるものではない。

また、以下の図面を使用した説明において、図面は模式的なものであり、理解の容易のために説明に必要な要素以外の図示は適宜省略されている。

【0010】

(1)ソースコード作成装置の全体構成

30

図1は本実施形態に係るソースコード作成装置1の機能構成の一例を示すブロック図である。

ソースコード作成装置1は、CPU(Central Processing Unit)10、ROM(Read Only Memory)11、RAM(Random Access Memory)12、HDD(Hard Disk Drive)13、ネットワークI/F部20、操作情報部30、バス40を含んで構成されている。

【0011】

CPU10は、ソースコード作成装置1の全体の動作を司るものであり、後述するソースコード作成装置1の処理を示すフローチャートは、CPU10により実行される。

ROM11は、ソースコード作成装置1の起動時に動作するブートプログラムなどが記憶されている不揮発性の記憶装置である。

40

RAM12は、OS(Operating System)、プログラム、各種データが展開される揮発性の記憶装置である。

HDD13は、各データベース、ソース作成プログラムなどのプログラム、OSなどが記録される不揮発性の記憶装置である。

【0012】

ネットワークI/F部20は、ネットワークに接続するためのものであり、NIC(Network Interface Card)及びドライバで構成される。操作情報部30は、液晶表示パネル、各種操作ボタン、タッチパネル等を組み合わせて構成され、使用者(オペレータ)がソースコード作成装置1の操作や情報を入力する際に用いられる。また、ソースコード作成装置1に関する情報を使用者(オペレータ)に表示する。バス40は、情報のやりとりが

50

行われる際に使用される。

【0013】

(2) アクティビティ図による動作の記述

図2(a)は待ち合わせ/分岐処理の一例、(b)は選択処理の一例を示す図、図7は、先行技術では、アクティビティ図で記述された動作からソースコードを作成できない処理のアクティビティ図の一例を示す図である。

以下、図面を参照しながらアクティビティ図による動作の記述とソースコード作成について説明する。

【0014】

(2.1) アクティビティ図

システムの記述には、統一モデリング言語(UML: Unified Modeling Language)を使用することができ、UMLはシステムを記述するための一連の方法を提供する。このような方法の1つが、アクティビティ図の使用である。

アクティビティ図は、アクティビティ(処理)の観点やアクティビティ間の制御フローの観点からシステムを記述する。

【0015】

図2(a)は、待ち合わせ/分岐(ジョイン/フォーク)ノードと呼ばれるアクティビティ図の要素を示している。この例では、ジョイン/フォークノード110は、ステップS101のActivity11、ステップS104のActivity14、ステップS102のActivity12、及びステップS103のActivity13の間の

関係を記述している。すなわち、フォークノード110は、Activity11とActivity14の完了を待ち合わせた後にActivity12及びActivity13が同時に開始される分岐処理を示している。

【0016】

図2(b)は、判断(デシジョン)ノードと呼ばれるアクティビティ図の要素を示している。ここで、デシジョンノード130は、ステップS131のActivity31、ステップS132のActivity32、及びステップS133のActivity33の間の関係を記述している。

すなわち、デシジョンノード130は、Activity31の完了時にActivity32及び32のいずれか1つのみが開始されることを示している。Activity32及びActivity33のいずれが開始されるかは、デシジョンノード130に関連付けられたロジック条件によって判断される。例えば、システムの特定のパラメータが何らかの所定値を上回るか又は下回るかである。

【0017】

UMLアクティビティ図では、分岐(フォーク)されたアクティビティ要素と、待ち合わせ(ジョイン)を無関係に記述できるにも拘わらず、先行技術では一つの並列実行(フォーク/ジョイン)から分岐(フォーク)されたアクティビティ要素は、他の一つの並列実行(フォーク/ジョイン)により合流(ジョイン)される場合に制約があった。

そのために、図5に示すアクティビティ図で記述された動作から、ソースコードを作成できない部分があった。

【0018】

具体的には、図5のアクティビティ図において、フォークノード210からフォークされたActivityAとActivityBは、同一のフォーク/ジョインノード220にジョインしなければならないが、ActivityBはフォーク/ジョインノード250にジョインされることができなかった。

同様に、フォーク/ジョインノード220からフォークされたActivityC、ActivityD、ActivityEは、同一のフォーク/ジョインノード240にジョインしなければならないが、ActivityEはフォーク/ジョインノード230にジョインされることができなかった。

10

20

30

40

50

## 【0019】

## (2.2) クラス図

図3は本実施形態に係るモデルを示すクラス図、図4は本実施形態に係るソースコード作成の流れを示すフローチャート、図6は先行技術のモデルを示すクラス図である。

本実施形態に係るデザインパターンを説明する前に、図6を参照しながら比較例として、先行技術に記載されたソースコード作成装置について説明する。

## 【0020】

図6に示すR9は、Activityオブジェクト50には、単一Activityオブジェクト52とそれ以外である複合Activityオブジェクト54の2種類あることを示している。

また、図5に示すR10は、複合Activityオブジェクト54には、フォークジョインActivityオブジェクト56とサブActivityオブジェクト58の2種類あることを示している。

## 【0021】

比較例のソースコード作成装置は、まず、アクティビティ図の記述にかかわらず、仮のサブActivityに対応するソースコードを所定のファイルに追記する。

## 【0022】

次に、最初に取得したActivity要素を仮のサブActivityの最初に実行するActivity(R2)とするための順序オブジェクト68のソースコードを追記し、仮のサブActivityを終了通知先(R7)とするソースコードを追記する。

## 【0023】

そして、次に取得したActivity要素が存在する場合、それを次に実行するActivity(R8)とするための順序オブジェクト68のソースコードを追記し、仮のサブActivityを終了通知先(R7)とするソースコードを追記する。

存在しない場合は、次に実行するActivity(R8)がないことを示す順序オブジェクト68のソースコードを追記し、終了となる。

## 【0024】

次に取得したActivity要素が単一Activityか複合Activityかを判別し、単一Activityである場合は、単一Activityに対応するソースコードを所定のファイルに追記する。

## 【0025】

複合Activityの場合、それがフォークジョインActivityかサブActivityかを判別し、サブActivityである場合はサブActivityに対応するソースコードを追記し、フォークジョインActivityである場合はフォークジョインActivityに対応するソースコードを追記する。

## 【0026】

取得したActivity要素がサブActivityの場合、サブActivity内の最初に実行するActivity要素に対応するソースコードを追記する。そして最初のサブActivity同様の処理を繰り返す。

## 【0027】

取得したActivity要素がフォークジョインActivityの場合、フォーク後ジョインするActivity要素に対応したソースコードを追記し、フォークジョインActivityを終了通知先(R7)とするソースコードを追記する。

そして、取得したActivity要素をフォークジョインするActivity(R6)とするためのフォークジョインオブジェクト66のソースコードを追記する。

## 【0028】

取得したActivity要素がデシジョンの場合、選択する判定を実施するデシジョンオブジェクト62に対応するソースコードを追記する。

## 【0029】

これにより、図6で示したクラス図で、アクティビティ図で示される処理を実行可能な

10

20

30

40

50

ソースコードを作成することができるが、フォークジョイン `Activity` にフォークされた `Activity` 要素は、同一のフォークジョイン `Activity` でジョインしなければならないという制約があった。

#### 【0030】

##### (2.3) ソースコード作成装置1の動作

本実施形態に係るソースコード作成装置1は、フォークジョインする `Activity` とジョインする `Activity` 毎に別のソースコードを追記する。

以下本実施形態に係るモデルを示すクラス図(図3)及び本実施形態に係るソースコード作成の流れを示すフローチャート(図4)を参照しながらソースコード作成装置1の動作について説明する。

10

#### 【0031】

まずソースコード作成装置1は、待ち合わせ/分岐処理(フォーク/ジョイン)、選択処理(デシジョン)、実行処理(アクティビティ)、及び終了処理(終了)を含んで構成されるアクティビティ図内の任意の要素300を取得する(S301)。

#### 【0032】

次に、取得された要素300がフォーク/ジョインオブジェクト310、デシジョンオブジェクト320、アクティビティオブジェクト330、終了オブジェクト340のいずれであるかを判定する第1判定処理(R1)を行う(S302)。

#### 【0033】

そして、取得された要素300がフォーク/ジョインオブジェクト310と判定された場合(S302:A)には、フォーク/ジョインオブジェクト310に対応するソースコード追記し(S302A)、フォーク/ジョインオブジェクト310と判定された要素300が待ち合わせ完了後に分岐(フォーク)する要素を識別(R2)するためのフォークセクタオブジェクト370に対応するソースコードを予め定められたファイルに追記する(S303)。

20

#### 【0034】

そして、待ち合わせ(ジョイン)する要素300を識別(R3)するためのジョインセクタオブジェクト380に対応するソースコードを予め定められたファイルに追記する(S304)。

#### 【0035】

取得された要素300がデシジョンオブジェクト320と判定された場合(S302:B)には、デシジョンオブジェクト320に対応するソースコードを追記し(S302B)、デシジョンオブジェクト320と判定された要素300が、条件により分岐する要素を識別(R4)するためのデシジョンセクタオブジェクト390に対応するソースコードを予め定められたファイルに追記する(S305)。

30

#### 【0036】

取得された要素300がアクティビティオブジェクト330と判定された場合には、アクティビティオブジェクト330と判定された要素300が、単一アクティビティオブジェクト350かサブアクティビティオブジェクト360かを判定する第2判定処理(R5)を行う(S306)。

40

第2判定処理(S306)により要素300が単一アクティビティオブジェクト350と判定された場合(S306:E)には、単一アクティビティオブジェクト350に対応するソースコードを追記し(S306E)、次に実施する要素を識別(R6)するためのソースコードを予め定められたファイルに追記する(S307)。

#### 【0037】

第2判定処理(S306)により要素300がサブアクティビティオブジェクト360と判定された場合(S306:F)には、サブアクティビティオブジェクト360に対応するソースコードを追記する(S306F1)。

そして、次に実施する要素を識別(R6)するためのソースコードを予め定められたファイルに追記し(S307F2)、要素300が最初に実施するアクティビ

50

ティオブジェクト330を識別するための(R7)ソースコードを予め定められたファイルに追記する(S308)。

【0038】

第1判定処理(S302)によりエレメント300が終了(S302:D)と判定された場合には、エレメント300が終了を通知するサブアクティビティオブジェクト360を識別(R8)するためのソースコードを予め定められたファイルに追記する(S309)。

【0039】

最後に、全エレメントの取得が完了したかを判定する(S310)。

【0040】

(3)作用・効果

本実施形態に係るソースコード作成装置1によれば、フォークジョインActivityにフォークされたActivity要素は、同一のフォークジョインActivityでジョインしなければならないという制約を受けることがなく、比較例のソースコード作成装置ではソースコードを作成できなかった一例として、図6に図示した「不可」部分であってもソースコード作成が可能になる。

【符号の説明】

【0041】

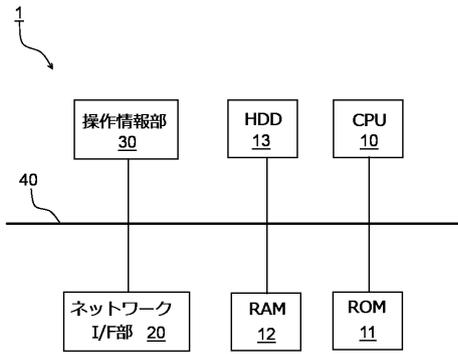
- 1・・・ソースコード作成装置
- 10・・・CPU
- 11・・・ROM
- 12・・・RAM
- 13・・・HDD
- 20・・・ネットワークI/F部
- 30・・・操作情報部
- 40・・・バス
- 300・・・エレメント
- 310・・・フォークジョインオブジェクト
- 320・・・デシジョンオブジェクト
- 330・・・アクティビティオブジェクト
- 340・・・終了オブジェクト
- 350・・・単一アクティビティオブジェクト
- 360・・・サブアクティビティオブジェクト
- 370・・・フォークセクタオブジェクト
- 380・・・ジョインセクタオブジェクト
- 390・・・デシジョンセクタオブジェクト

10

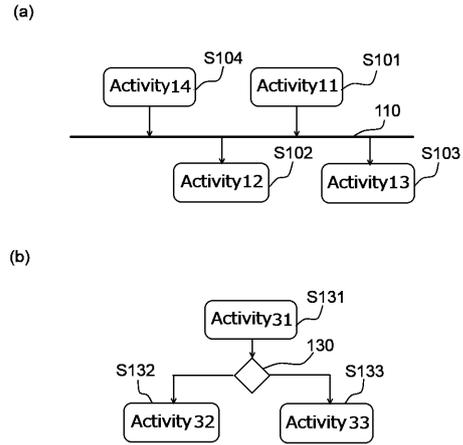
20

30

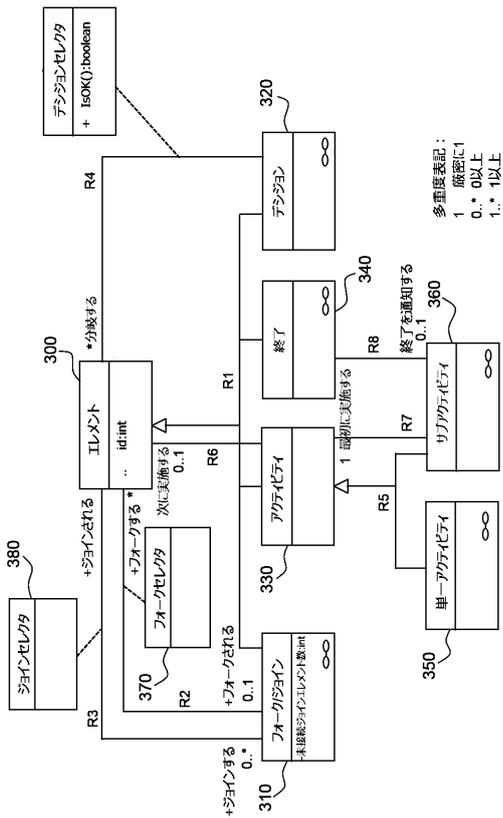
【 図 1 】



【 図 2 】



【 図 3 】



【 図 4 】

