

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5540160号  
(P5540160)

(45) 発行日 平成26年7月2日(2014.7.2)

(24) 登録日 平成26年5月9日(2014.5.9)

(51) Int.Cl. F I  
G O 6 F 21/56 (2013.01) G O 6 F 21/00 1 5 6 A

請求項の数 12 (全 32 頁)

(21) 出願番号	特願2013-544261 (P2013-544261)	(73) 特許権者	503360115
(86) (22) 出願日	平成24年11月12日(2012.11.12)		独立行政法人科学技術振興機構
(86) 国際出願番号	PCT/JP2012/079304		埼玉県川口市本町四丁目1番8号
(87) 国際公開番号	W02013/073504	(74) 代理人	110000338
(87) 国際公開日	平成25年5月23日(2013.5.23)		特許業務法人HARAKENZO WORLD PATENT & TRADEMARK
審査請求日	平成26年3月5日(2014.3.5)	(72) 発明者	前田 俊行
(31) 優先権主張番号	特願2011-249978 (P2011-249978)		日本国兵庫県神戸市東灘区森南町2-5-10-201
(32) 優先日	平成23年11月15日(2011.11.15)		
(33) 優先権主張国	日本国(JP)		
早期審査対象出願		審査官	岸野 徹

最終頁に続く

(54) 【発明の名称】 プログラム解析・検証サービス提供システム、その制御方法、制御プログラム、コンピュータを機能させるための制御プログラム、プログラム解析・検証装置、プログラム解析・検証ツール管

(57) 【特許請求の範囲】

【請求項1】

対象プログラムをプログラム解析・検証ツールを用いて解析、検証するプログラム解析・検証サービス提供システムであって、

対象プログラムを取得する対象プログラム取得手段と、

複数のプログラム解析・検証ツールを格納したツール記憶部と、

上記対象プログラム取得手段が取得した対象プログラムの解析、検証に用いるプログラム解析・検証ツールを、上記ツール記憶部に格納された複数のプログラム解析・検証ツールから抽出するツール抽出手段と、

上記ツール抽出手段が抽出したプログラム解析・検証ツールを上記ツール記憶部から取得するツール取得手段と、

上記ツール取得手段が取得した各プログラム解析・検証ツールを用いて、上記対象プログラム取得手段が取得した対象プログラムを解析、検証するツール実行手段と、を備えることを特徴とするプログラム解析・検証サービス提供システム。

【請求項2】

上記対象プログラム取得手段が取得した上記対象プログラムの解析、検証に用いるプログラム解析・検証ツールを指定するツール指定情報を取得するツール指定情報取得手段をさらに備え、

上記ツール記憶部は、複数のプログラム解析・検証ツールを各プログラム解析・検証ツールのツール関連情報とそれぞれ対応付けて格納しており、

10

20

上記ツール抽出手段は、上記ツール指定情報取得手段が取得した上記ツール指定情報に基づいて、上記ツール関連情報を検索し、適合したツール関連情報と対応付けられたプログラム解析・検証ツールを抽出するものであることを特徴とする請求項1に記載のプログラム解析・検証サービス提供システム。

【請求項3】

上記ツール記憶部は、上記プログラム解析・検証ツールとして、当該プログラム解析・検証ツールを導入、設定した仮想マシンを格納しており、

上記ツール実行手段は、上記ツール取得手段が取得した仮想マシン上で、上記対象プログラム取得手段が取得した対象プログラムを解析、検証するものであることを特徴とする請求項1または2に記載のプログラム解析・検証サービス提供システム。

10

【請求項4】

プログラム解析・検証ツールを導入、設定した仮想マシンを、プログラム解析・検証ツール毎に生成する仮想マシン生成手段をさらに備え、

上記ツール記憶部は、上記プログラム解析・検証ツールとして、上記仮想マシン生成手段が生成した仮想マシンを格納していることを特徴とする請求項3に記載のプログラム解析・検証サービス提供システム。

【請求項5】

上記プログラム解析・検証サービス提供システムは、前処理前プログラムを前処理して上記対象プログラムを得る処理と、該処理によって得た上記対象プログラムを上記プログラム解析・検証ツールを用いて解析、検証する処理とを連続して行うものであって、

20

上記前処理前プログラムを前処理して得た上記対象プログラムを、該対象プログラムの特徴記述情報と対応付けて格納する前処理後プログラム記憶部をさらに備え、

上記対象プログラム取得手段は、上記前処理後プログラム記憶部から、上記対象プログラムを取得するものであり、

上記ツール指定情報取得手段は、上記前処理後プログラム記憶部から、上記対象プログラム取得手段が取得した上記対象プログラムに対応付けられた上記特徴記述情報を、上記ツール指定情報として取得するものであることを特徴とする請求項2に記載のプログラム解析・検証サービス提供システム。

【請求項6】

上記プログラム解析・検証サービス提供システムは、上記対象プログラムを上記プログラム解析・検証ツールを用いて解析、検証する処理と、該処理によって得た結果データを後処理する処理とを連続して行うものであって、

30

上記ツール実行手段が上記プログラム解析・検証ツールを用いて上記対象プログラムを解析、検証して得た結果データを、該結果データの特徴記述情報と対応付けて格納する結果データ記憶部をさらに備えたとともに、

上記ツール記憶部は、上記結果データを後処理する後処理ツールを該後処理ツールのツール関連情報と対応付けて格納しており、

上記ツール指定情報取得手段は、上記結果データ記憶部から、上記結果データに対応付けられた上記特徴記述情報を、上記ツール指定情報として取得するものであり、

上記ツール抽出手段は、上記ツール指定情報取得手段が取得した上記ツール指定情報に基づいて、上記ツール関連情報を検索し、適合したツール関連情報と対応付けられた後処理ツールを、上記ツール記憶部に格納された複数の後処理ツールから抽出するものであることを特徴とする請求項2に記載のプログラム解析・検証サービス提供システム。

40

【請求項7】

対象プログラムをプログラム解析・検証ツールを用いて解析、検証するプログラム解析・検証サービス提供システムの制御方法であって、

上記プログラム解析・検証サービス提供システムは、複数のプログラム解析・検証ツールを格納したツール記憶部を備え、

対象プログラムを取得する対象プログラム取得ステップと、

上記対象プログラム取得ステップにて取得した対象プログラムの解析、検証に用いるプ

50

ログラム解析・検証ツールを、上記ツール記憶部に格納された複数のプログラム解析・検証ツールから抽出するツール抽出ステップと、

上記ツール抽出ステップにて抽出したプログラム解析・検証ツールを上記ツール記憶部から取得するツール取得ステップと、

上記ツール取得ステップにて取得した各プログラム解析・検証ツールを用いて、上記対象プログラム取得ステップにて取得した対象プログラムを解析、検証するツール実行ステップと、を含むことを特徴とするプログラム解析・検証サービス提供システムの制御方法。

【請求項 8】

請求項 1 または 2 に記載のプログラム解析・検証サービス提供システムの上記各手段としてコンピュータを機能させるための制御プログラム。

10

【請求項 9】

請求項 8 に記載の制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 10】

請求項 1 に記載のプログラム解析・検証サービス提供システムを構成するプログラム解析・検証装置であって、

上記対象プログラム取得手段と、上記ツール取得手段と、上記ツール実行手段と、を備えることを特徴とするプログラム解析・検証装置。

【請求項 11】

請求項 1 に記載のプログラム解析・検証サービス提供システムを構成するプログラム解析・検証ツール管理装置であって、

20

上記ツール記憶部と、上記ツール抽出手段と、を備えることを特徴とするプログラム解析・検証ツール管理装置。

【請求項 12】

複数のデータ処理ツールを順次用いることにより、対象データを複数段階にデータ処理するデータ処理サービス提供システムであって、

複数のデータ処理ツールを各データ処理ツールのツール関連情報とそれぞれ対応付けて格納したツール記憶部と、

前段階のデータ処理ツールを用いてデータ処理して得られた結果データを、該結果データの特徴記述情報と対応付けて格納する結果データ記憶部と、

30

上記結果データ記憶部から上記結果データを取得する結果データ取得手段と、

上記結果データ取得手段が取得した上記結果データのデータ処理に用いるデータ処理ツールを指定するツール指定情報として、当該結果データに対応付けられた上記特徴記述情報を上記結果データ記憶部から取得するツール指定情報取得手段と、

上記ツール指定情報取得手段が取得した上記ツール指定情報に基づいて、上記ツール関連情報を検索し、適合したツール関連情報と対応付けられたデータ処理ツールを、上記ツール記憶部に格納された複数のデータ処理ツールから抽出するツール抽出手段と、

上記ツール抽出手段が抽出したデータ処理ツールを上記ツール記憶部から取得するツール取得手段と、

上記ツール取得手段が取得したデータ処理ツールを用いて、上記結果データ取得手段が取得した結果データをデータ処理するツール実行手段と、を備えることを特徴とするデータ処理サービス提供システム。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、対象プログラムをプログラム解析・検証ツールを用いて解析、検証するプログラム解析・検証サービス提供システム等に関する。

【背景技術】

【0002】

プログラム解析・検証とは、計算機のプログラムが何らかの性質を満たすことを解析・

50

検証する技術である。プログラム解析・検証は、プログラムの安全性や信頼性を保証・検証するための非常に有効な手段であると考えられている。

【 0 0 0 3 】

近年、このプログラム解析・検証技術は飛躍的に進歩し、様々な実用的プログラムに対して解析・検証が行えるようになりつつある。実際、プログラム解析・検証を行うプログラム（以降、「プログラム解析・検証ツール」と呼ぶ）が研究用途・商業用途問わず多く開発されている。

【 0 0 0 4 】

ここで、プログラム解析・検証ツールを共有する従来技術について、ウイルス解析・検査サイトを例に説明する。

【 0 0 0 5 】

図 1 1 は、ウイルス解析・検査サイトの典型的な構成例を表したものである。図 1 1 に示すように、ウイルス解析・検査サイト 9 0 0 の利用者は、ウイルス解析・検査したい対象プログラムをウイルス解析・検査サイト 9 0 0 に送付する（P 1）。例えば、下記の非特許文献 1 にリストアップされているサイトであれば、Web インターフェースを經由して送付する。そして、対象プログラムが送付されると、ウイルス解析・検査サイト 9 0 0 は、ウイルス解析・検査ソフトウェア 9 0 1 を利用して、送付された対象プログラムの解析・検査を行う（P 2）。解析・検査が終了したら、ウイルス解析・検査サイト 9 0 0 はその結果を利用者へ送付する（P 3）。

【 0 0 0 6 】

このウイルス解析・検査サイト 9 0 0 の利点は、サイトの利用者が解析・検査のための計算資源（CPU 時間やメモリ容量など）を用意したり、解析・検査ソフトウェアを導入したりする手間がかからないことにある。

【 先行技術文献 】

【 非特許文献 】

【 0 0 0 7 】

【 非特許文献 1 】 “ サンドボックスによるウイルス・マルウェア解析サイト ”、[online]、2011年6月、[2011年10月20日検索]、インターネット<URL: <http://lhsp.s206.xrea.com/misc/virussandbox.html>>

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 0 8 】

しかし、従来のプログラム解析・検証ツールには、以下に述べるような実用上の問題があった。

【 0 0 0 9 】

（ 1 ）プログラム解析・検証ツールが要する計算資源の肥大化

プログラム解析・検証技術の種類によっては、プログラム解析・検証ツールが非常に多くの計算資源（CPU 時間やメモリ容量など）を必要とすることがある。それは、複雑な性質を解析・検証する場合や、解析・検証対象のプログラムの規模が大きい場合に、特に顕著となる。このため、個々のプログラム解析・検証ツールの利用者が現実的に実行可能な解析・検証の種類や規模には限界があった。

【 0 0 1 0 】

（ 2 ）プログラム解析・検証ツールの種類の多様化

解析・検証技術の進歩に伴い、非常に多くの種類のプログラム解析・検証ツールが存在しており、また新たに考案・開発され続けている。このため、プログラム解析・検証ツールの利用者にとって、どのプログラム解析・検証ツールを選択すれば良いかを判断することが困難である。

【 0 0 1 1 】

（ 3 ）プログラム解析・検証ツールの導入・運用の手間・コスト

上述のとおり、多くの種類のプログラム解析・検証ツールが存在しているが、プログラ

10

20

30

40

50

ム解析・検証ツールの利用者の目的が一種類のプログラム解析・検証ツールで満たされることは稀である。このため、利用者は複数のプログラム解析・検証ツールを導入し、運用・維持管理しなければならず、手間やコストが大きくなる。

【0012】

(4) プログラム解析・検証ツール自体のテストの困難さ

プログラム解析・検証ツールの有用性は広く認識されつつある。しかし、その一方で、プログラム解析・検証ツール自体が期待通りに動作するかどうかをテストすることは困難である。これは、プログラム解析・検証ツールの提供者にとっては、プログラム解析・検証ツールの利用者と同様に、プログラムの解析・検証の規模には限界があること、また、プログラム解析・検証ツールの提供者が利用・閲覧可能なプログラムおよびそのソースコードには限界があることにより、様々な対象プログラムに対してプログラム解析・検証ツールのテストができないためである。

10

【0013】

本発明は、上記の問題点に鑑みてなされたものであり、その目的は、複数のプログラム解析・検証ツールから用いるべきものを適切に選択して、プログラムの解析・検証を容易に実行できるプログラム解析・検証サービス提供システム等を実現することにある。

【課題を解決するための手段】

【0014】

上記課題を解決するために、本発明に係るプログラム解析・検証サービス提供システムは、対象プログラムをプログラム解析・検証ツールを用いて解析、検証するプログラム解析・検証サービス提供システムであって、対象プログラムを取得する対象プログラム取得手段と、複数のプログラム解析・検証ツールを格納したツール記憶部と、上記対象プログラム取得手段が取得した対象プログラムの解析、検証に用いるプログラム解析・検証ツールを、上記ツール記憶部に格納された複数のプログラム解析・検証ツールから抽出するツール抽出手段と、上記ツール抽出手段が抽出したプログラム解析・検証ツールを上記ツール記憶部から取得するツール取得手段と、上記ツール取得手段が取得した各プログラム解析・検証ツールを用いて、上記対象プログラム取得手段が取得した対象プログラムを解析、検証するツール実行手段と、を備えることを特徴としている。

20

【0015】

また、本発明に係るプログラム解析・検証サービス提供システムの制御方法は、対象プログラムをプログラム解析・検証ツールを用いて解析、検証するプログラム解析・検証サービス提供システムの制御方法であって、上記プログラム解析・検証サービス提供システムは、複数のプログラム解析・検証ツールを格納したツール記憶部を備え、対象プログラムを取得する対象プログラム取得ステップと、上記対象プログラム取得ステップにて取得した対象プログラムの解析、検証に用いるプログラム解析・検証ツールを、上記ツール記憶部に格納された複数のプログラム解析・検証ツールから抽出するツール抽出ステップと、上記ツール抽出ステップにて抽出したプログラム解析・検証ツールを上記ツール記憶部から取得するツール取得ステップと、上記ツール取得ステップにて取得した各プログラム解析・検証ツールを用いて、上記対象プログラム取得ステップにて取得した対象プログラムを解析、検証するツール実行ステップと、を含むことを特徴としている。

30

40

【0016】

さらに、本発明に係るデータ処理サービス提供システムは、複数のデータ処理ツールを順次用いることにより、対象データを複数段階にデータ処理するデータ処理サービス提供システムであって、複数のデータ処理ツールを各データ処理ツールのツール関連情報とそれぞれ対応付けて格納したツール記憶部と、前段階のデータ処理ツールを用いてデータ処理して得られた結果データを、該結果データの特徴記述情報と対応付けて格納する結果データ記憶部と、上記結果データ記憶部から上記結果データを取得する結果データ取得手段と、上記結果データ取得手段が取得した上記結果データのデータ処理に用いるデータ処理ツールを指定するツール指定情報として、当該結果データに対応付けられた上記特徴記述情報を上記結果データ記憶部から取得するツール指定情報取得手段と、上記ツール指定情

50

報取得手段が取得した上記ツール指定情報に基づいて、上記ツール関連情報を検索し、適合したツール関連情報と対応付けられたデータ処理ツールを、上記ツール記憶部に格納された複数のデータ処理ツールから抽出するツール抽出手段と、上記ツール抽出手段が抽出したデータ処理ツールを上記ツール記憶部から取得するツール取得手段と、上記ツール取得手段が取得したデータ処理ツールを用いて、上記結果データ取得手段が取得した結果データをデータ処理するツール実行手段と、を備えることを特徴としている。

【発明の効果】

【0017】

本発明の一態様によれば、複数のプログラム解析・検証ツールから用いるべきものを適切に選択して、プログラムの解析・検証を容易に実行できるという効果を奏する。そして、プログラム解析・検証を共有システム上で行うことで、従来のプログラム解析・検証技術の実用化上の種々の問題点を解決できるという効果を奏する。また、これらの効果は、プログラムの解析・検証を行う処理に限定されない。

10

【図面の簡単な説明】

【0018】

【図1】本発明の実施形態を示すものであり、プログラム解析・検証サービス提供システムの構成の詳細を示す機能ブロック図である。

【図2】本発明の実施形態を示すものであり、プログラム解析・検証サービス提供システムに含まれる解析・検証実行サブシステムの構成および解析・検証の実行処理の流れを表したブロック図である。

20

【図3】本発明の実施形態を示すものであり、プログラム解析・検証サービス提供システムに含まれる解析・検証実行サブシステムの構成および解析・検証結果の表示処理の流れを表したブロック図である。

【図4】本発明の実施形態を示すものであり、プログラム解析・検証サービス提供システムに含まれる対象プログラム管理サブシステムの構成および対象プログラムの登録処理の流れを表したブロック図である。

【図5】本発明の実施形態を示すものであり、プログラム解析・検証サービス提供システムに含まれる対象プログラム管理サブシステムの構成および解析・検証実行サブシステムからの対象プログラムの取得要求処理の流れを表したブロック図である。

【図6】本発明の実施形態を示すものであり、プログラム解析・検証サービス提供システムに含まれるツール管理サブシステムの構成およびプログラム解析・検証ツールの登録処理の流れを表したブロック図である。

30

【図7】本発明の実施形態を示すものであり、プログラム解析・検証サービス提供システムに含まれるツール管理サブシステムの構成および解析・検証実行サブシステムからのプログラム解析・検証ツールのイメージを含む仮想マシン取得要求処理の流れを表したブロック図である。

【図8】図1に示したプログラム解析・検証サービス提供システムにおいて用いられるデータの説明図であり、依頼管理データベースのデータ構造を示す。

【図9】図1に示したプログラム解析・検証サービス提供システムにおいて用いられるデータの説明図であり、対象プログラム管理データベースのデータ構造を示す。

40

【図10】図1に示したプログラム解析・検証サービス提供システムにおいて用いられるデータの説明図であり、ツール管理データベースのデータ構造を示す。

【図11】従来技術を示すものであり、ウイルス解析・検査サイトの典型的な構成例を表したブロック図である。

【発明を実施するための形態】

【0019】

以下、本発明の一実施形態について、詳細に説明する。図1～図10に基づいて、本実施形態に係るプログラム解析・検証サービス提供システム1について説明すれば以下のとおりである。

【0020】

50

( 1 . プログラム解析・検証ツールの共有のためのシステムの構成手法 )

最初に、前述した従来のプログラム解析・検証ツールの実用上の 4 つの問題を解決するための手法を説明する。なお、具体的な手法の実施例については後述する。

【 0 0 2 1 】

鍵となるアイデアは、近年流行のいわゆる「クラウド」的な考え方に基づき、複数のプログラム解析・検証ツールの利用者と複数のプログラム解析・検証ツールの提供者（ツール提供者）が、1つのシステムを共有して、プログラム解析・検証を行うことにある。

【 0 0 2 2 】

具体的には、プログラム解析・検証サービス提供システム 1 は、解析・検証実行サブシステム 1 0、対象プログラム管理サブシステム 2 0、ツール管理サブシステム 3 0 の 3 つのサブシステムからなる。以降、それぞれのサブシステムについて説明する。なお、これらのサブシステムは、1つのコンピュータ上に3つのサブシステムのすべてを配置してもよいし、ネットワークで接続された複数のコンピュータ群に分散して配置してもよい。

【 0 0 2 3 】

( 1 . 1 . 解析・検証実行サブシステム )

解析・検証実行サブシステム 1 0（図 2、図 3）は、利用者の要求に基づいて、まず、対象プログラム管理サブシステム 2 0 からプログラム解析・検証の対象となる対象プログラム P を取得し、ツール管理サブシステム 3 0 から解析・検証ツール（仮想マシン T）を取得する。次に、取得した対象プログラム P および解析・検証ツール（仮想マシン T）を用いてプログラム解析・検証を実行し、その結果（解析・検証結果データ R）をリポジトリ 1 3 に登録する。また、その結果を利用者に示す。

【 0 0 2 4 】

また、解析・検証実行サブシステム 1 0 は、その稼働状況に応じて、利用者からの要求とは独立に対象プログラム管理サブシステム 2 0 からプログラム解析・検証の対象となる対象プログラム P を取得する。解析・検証実行サブシステム 1 0 が対象プログラム P を取得するときの稼働状況とは、具体的には、（ 1 ）システムの CPU 利用率、（ 2 ）メモリ利用率、（ 3 ）外部記憶利用率などが、あらかじめ設定した閾値を下回った場合などである。解析・検証実行サブシステム 1 0 は、対象プログラム P を取得するとともに、ツール管理サブシステム 3 0 からプログラム解析・検証ツール（仮想マシン T）を取得して、解析・検証を行い、その結果（解析・検証結果データ R）をリポジトリ 1 3 に登録しておく。これにより、後に利用者から要求があった場合に、リポジトリ 1 3 に問合せをすることで、迅速に解析・検証結果を返答することが可能である。

【 0 0 2 5 】

このように、複数の利用者で解析・検証実行サブシステム 1 0 を共有することで、余剰の計算資源を融通し合うことが可能となり、全体としてプログラム解析・検証のコストを抑えることができる。また、利用者の要求とは独立に、プログラム解析・検証をあらかじめ実行できるため、利用者の要求から解析・検証結果の提供までに要する計算資源のコストを隠蔽することもできる。さらに、複数のプログラム解析・検証ツールの中から適切なプログラム解析・検証ツールを選択することが容易になる。

【 0 0 2 6 】

( 1 . 2 . 対象プログラム管理サブシステム )

対象プログラム管理サブシステム 2 0（図 4、図 5）は、利用者の要求にもとづいて、利用者が指定・提供する対象プログラム P の維持管理（取得・保存・更新・削除）を行う。また解析・検証実行サブシステム 1 0 からの要求に対して、適切な対象プログラム P を提供する。

【 0 0 2 7 】

このように、複数の利用者によって登録された対象プログラム P に対してプログラム解析・検証ツールを実行できるため、個々のプログラム解析・検証ツールの提供者は様々な対象プログラム P に対してプログラム解析・検証ツールのテストを行うことができる。

【 0 0 2 8 】

10

20

30

40

50

### ( 1 . 3 . ツール管理サブシステム )

ツール管理サブシステム 3 0 ( 図 6、図 7 ) は、利用者 ( プログラム解析・検証ツールの提供者 ) の要求・指示に基づいて、利用者の指定するプログラム解析・検証ツールの維持管理 ( 導入・設定・削除 ) を行う。また、解析・検証実行サブシステム 1 0 からの取得要求に対して、適切なプログラム解析・検証ツールを提供する。

#### 【 0 0 2 9 】

このように、プログラム解析・検証ツールの提供者が導入・維持管理を行い、複数の利用者でプログラム解析・検証ツールを共有することで、利用者自身がプログラム解析・検証ツールの導入・維持管理を行う必要がなくなる。

#### 【 0 0 3 0 】

### ( 2 . プログラム解析・検証ツールの共有のためのシステムの実装例 )

つづいて、前述したそれぞれのサブシステムの一実装例について説明する。

#### 【 0 0 3 1 】

### ( 2 . 1 . 解析・検証実行サブシステムの一実装例 )

図 2 は、解析・検証実行サブシステム 1 0 の構成および解析・検証の実行処理の流れを表したブロック図である。

#### 【 0 0 3 2 】

解析・検証実行サブシステム 1 0 は、コントローラ 1 1、仮想マシン実行環境 1 2、リポジトリ 1 3 からなる。コントローラ 1 1 は、利用者からの実行要求の処理や、対象プログラム管理サブシステム 2 0 やツール管理サブシステム 3 0 とのやり取り、解析・検証の実行や、解析・検証結果のリポジトリ 1 3 への保存を行う。仮想マシン実行環境 1 2 は、解析・検証を行うための実行環境である。リポジトリ 1 3 は解析・検証の結果を保存する。

#### 【 0 0 3 3 】

なお、仮想マシン実行環境 1 2 を用いて解析・検証を行う理由は、二つある。まず、第 1 に、仮想マシンを用いて解析・検証を実行することで、複数の利用者および複数のプログラム解析・検証ツールの提供者の間で、システムの公平な利用を容易に実現することができる。ここで、システムの公平な利用とは、例えば、CPU やメモリ を大量に消費しているプログラム解析・検証ツールは実行を後回しにして、他のプログラム解析・検証ツールを先に実行するなどという。

#### 【 0 0 3 4 】

また、第 2 に、仮想マシン上にプログラム解析・検証ツールを導入・設定することによって ( 詳細は ( 2 . 3 . ツール管理サブシステムの一実装例 ) を参照 )、プログラム解析・検証ツールの提供者は、普段自らが開発・テストしている実行環境とほぼ同様にプログラム解析・検証ツールを導入・設定できるため、特殊な実行環境に対応させるためにプログラム解析・検証ツールを修正したり、また外部とのデータのやりとりをプログラム解析・検証ツールに行わせるための仕組みを別途用意したりといった、余計な負担をプログラム解析・検証ツールの提供者に掛けないようにすることができる。

#### 【 0 0 3 5 】

具体的な解析・検証の実行の流れは次のとおりである。まず、利用者から解析・検証の実行要求がコントローラ 1 1 に送られる ( A 1 )。この実行要求は、どの対象プログラム P に対して解析・検証を行うかを示すタグと、どのプログラム解析・検証ツール ( 仮想マシン T ) を用いるかを示すタグからなる。実行要求を受け付けたコントローラ 1 1 は、実行要求に指定されたタグおよび利用者の権限に応じて与えられるタグに基づいて、対象プログラム管理サブシステム 2 0 から対象プログラム P を取得し ( A 2 - a、A 3 - a ; 対象プログラム取得ステップ )、ツール管理サブシステム 3 0 からプログラム解析・検証ツールのイメージを含む仮想マシン T を取得する ( A 2 - b、A 3 - b ; ツール取得ステップ )。

#### 【 0 0 3 6 】

なお、上記「利用者の権限に応じて与えられるタグ」とは、個々の利用者に与えられた

10

20

30

40

50



、対象プログラム P およびプログラム解析・検証ツール（仮想マシン T）へのアクセス権限を表すタグである。例えば、最も基礎的なものとしては、利用者 A による実行要求には「USER\_\_A」のようなタグが与えられる。これにより、「USER\_\_A」のタグが付けられた対象プログラム P やプログラム解析・検証ツール（仮想マシン T）を、利用者 A が取得することが可能となる。一方、別の利用者 B に対しては「USER\_\_B」のようなタグが与えられるため、利用者 B は「USER\_\_A」のタグだけが与えられている対象プログラム P やプログラム解析・検証ツール（仮想マシン T）を取得することはできない。つまり、利用者 A の対象プログラム P やプログラム解析・検証ツール（仮想マシン T）を利用者 B から隠蔽することができる。また、より複雑な例として、利用者 A と利用者 B が共同の利用者として各自の対象プログラム P やプログラム解析・検証ツール（仮想マシン T）を共有する場合には、利用者 A・B の実行要求に対して「GROUP\_\_AB」のようなタグが与えられる。これにより、「GROUP\_\_AB」のタグが付けられた対象プログラム P やプログラム解析・検証ツール（仮想マシン T）を利用者 A・B が取得することが可能となる。

10

#### 【0037】

次に、取得した対象プログラム P およびツールイメージ（仮想マシン T）を用いて、仮想マシン実行環境 12 で実際に解析・検証を実行する（A4；ツール実行ステップ）。解析・検証の終了後、コントローラ 11 はその結果を受け取り（A5）、その結果をタグと紐付けしてリポジトリ 13 に保存する（A6）。なお、図 2 には明示的には示されていないが、前述のとおり、解析・検証実行サブシステム 10 の稼働状況に応じて、利用者から 20 の実行要求がなくても、図 2 の（A2）以降の手順をあらかじめ行っておくことも可能である。この場合、解析・検証の結果をリポジトリ 13 に保存する際に紐付けされるタグとしては、対象プログラム P 自身およびプログラム解析・検証ツールのツールイメージ（仮想マシン T）自身に与えられたタグを用いることができる。

20

#### 【0038】

図 3 は、解析・検証実行サブシステム 10 の構成および解析・検証結果の表示処理の流れを表したブロック図である。

#### 【0039】

リポジトリ 13 に保存した解析・検証結果は、以下のように、利用者へ提示される。まず、利用者から解析・検証結果の取得要求（結果取得要求）がコントローラ 11 に送られる（B1）。この結果取得要求は、前述の実行要求と同様のタグからなる。結果取得要求を受け付けたコントローラ 11 は、結果取得要求に指定されたタグおよび利用者の権限に応じて与えられるタグに基づいて、リポジトリ 13 から解析・検証結果を取得する（B2、B3）。次に、コントローラ 11 は取得した結果を利用者に表示する（B4）。なお、リポジトリ 13 に結果が保存されていない場合には、図 2 の（A2）以降の手順を実行することが可能である。

30

#### 【0040】

（2.2.対象プログラム管理サブシステムの一実装例）

図 4 は、対象プログラム管理サブシステム 20 の構成および対象プログラム P の登録処理の流れを表したブロック図である。

40

#### 【0041】

対象プログラム管理サブシステム 20 は、コントローラ 21 とリポジトリ 22 からなる。コントローラ 21 は、利用者からの対象プログラム登録要求の処理や対象プログラム P の取得、リポジトリ 22 への登録を行う。リポジトリ 22 は対象プログラム P を保存する。

#### 【0042】

具体的な対象プログラムの登録の流れは次のとおりである。まず、利用者から対象プログラム P の登録要求がコントローラ 21 へ送られる（C1）。この登録要求には、利用者が指定したタグと対象プログラム P の送付方法の指定からなる。登録要求に利用者が指定したタグは、利用者が任意の内容で任意の個数指定することができる。想定される利用方

50

法は、対象プログラム P で用いられているプログラミング言語の種類やプログラムの種類（ユーザプログラム、システムプログラムの区別など）を指定するなどである。また、対象プログラム P の送付方法としては、ここでは利用者からの直接送付（C 2）と外部システム（バージョン管理システムなど）からのインポートを想定している（C 2 - 1、C 2 - 2）。登録要求を受け付けたコントローラ 2 1 は、登録要求に指定された方法で対象プログラム P を取得し（C 2、C 2 - 1、C 2 - 2）、登録要求に指定されているタグおよび利用者の権限に応じて与えられるタグと紐付けて、リポジトリ 2 2 へ対象プログラム P を保存する（C 3）。

【0043】

図 5 は、対象プログラム管理サブシステム 2 0 の構成および解析・検証実行サブシステム 1 0 からの対象プログラム P の取得要求処理の流れを表したブロック図である。

10

【0044】

解析・検証実行サブシステム 1 0 から対象プログラム P の取得要求を受け付けると（D 1）、コントローラ 2 1 は取得要求に指定されたタグに基づいてリポジトリ 2 2 を検索する（D 2、D 3）。次いで、コントローラ 2 1 は、リポジトリ 2 2 から得た対象プログラム P を解析・検証実行サブシステムに送付する（D 4）。

【0045】

（2.3. ツール管理サブシステムの一実装例）

図 6 は、ツール管理サブシステム 3 0 の構成およびプログラム解析・検証ツールの登録処理の流れを表したブロック図である。

20

【0046】

ツール管理サブシステム 3 0 は、コントローラ 3 1、リポジトリ 3 2、仮想マシン実行環境 3 3 からなる。コントローラ 3 1 は、ツール提供者からの登録要求の処理や、プログラム解析・検証ツールの導入・設定用の仮想マシン（初期仮想マシン）の生成、ツールイメージを含む仮想マシン T のリポジトリ 3 2 への保存を行う。仮想マシン実行環境 3 3 は、ツール提供者がプログラム解析・検証ツールの導入・設定を行うための環境である。リポジトリ 3 2 は、ツール提供者がプログラム解析・検証ツールを導入・設定した仮想マシン T のシステムイメージを保存する。

【0047】

プログラム解析・検証ツールの登録処理の具体的な流れは次のとおりである。まず、ツール提供者からコントローラ 3 1 へプログラム解析・検証ツールの登録要求が送られる（E 1）。この登録要求はツール提供者が指定したタグからなる。登録要求のツール提供者が指定したタグは、ツール提供者が任意の内容で任意の個数指定することができる。登録要求のツール提供者が指定したタグは、（1）プログラム解析・検証ツールが扱うことができるプログラミング言語の種類およびプログラムの種類（ユーザプログラム、システムプログラムの区別など）、（2）プログラム解析・検証ツールが稼働する実行環境（オペレーティングシステムやライブラリの種類・バージョン、必要となる計算資源、すなわち CPU の速度や種類、メモリやディスクのサイズなど）、などを指定するとき利用することが想定される。

30

【0048】

登録要求を受け取ったコントローラ 3 1 は、新たに仮想マシン（初期仮想マシン）を生成する（E 2）。このとき、コントローラ 3 1 は、登録要求に指定されたタグに応じて、（オペレーティングシステムの種類やバージョン、メモリやディスクのサイズなどを調整した、適切な仮想マシン（初期仮想マシン）を作成する。なお、本実施の形態では、プログラム解析・検証ツールが導入、設定される前の状態の仮想マシンを「初期仮想マシン」と記すものとする。

40

【0049】

ここで、頻出するタグ指定の組合せに対しては、事前に適切なオペレーティングシステムやライブラリが導入された半完成状態の仮想マシン（初期仮想マシン）を生成して保存しておき、仮想マシンの生成時にはそれを単純にコピーしてもよい。これにより、仮想マ

50

シンの生成のコストを減らすことができる。

【 0 0 5 0 】

次に、ツール提供者は、コントローラ 3 1 を経由して生成された仮想マシン（初期仮想マシン）を、仮想マシン実行環境 3 3 上で実行し、その仮想マシン（初期仮想マシン）にプログラム解析・検証ツールを導入・設定する（E 3）。このように、登録要求に指定されたタグに基づいて適切なオペレーティングシステムやライブラリ等が仮想マシン（初期仮想マシン）にあらかじめ用意される。これにより、プログラム解析・検証ツールの導入・設定は、ツール提供者が普段行っている導入・設定手順とほぼ同様に行え、実行環境に対応させるためにプログラム解析・検証ツールを修正したり、外部システムとデータをやりとりするために特別な仕組みを用意したりといった余計な手間を、ツール提供者に掛け

10

【 0 0 5 1 】

最後に、コントローラ 3 1 は登録要求に指定されたタグと紐付けて、ツール提供者によってプログラム解析・検証ツールが導入・設定されたツールイメージを含む仮想マシン T をリポジトリ 3 2 に保存する（E 4）

図 7 は、ツール管理サブシステム 3 0 の構成および解析・検証実行サブシステム 1 0 からのプログラム解析・検証ツールのイメージを含む仮想マシン T の取得要求処理の流れを表したブロック図である。

【 0 0 5 2 】

解析・検証実行サブシステム 1 0 から仮想マシン T の取得要求を受け付けると（F 1 ; ツール取得ステップ）、コントローラ 3 1 は、取得要求に指定されたタグに基づいてリポジトリ 3 2 を検索し（F 2、F 3 ; ツール抽出ステップ）、得られた仮想マシン T を解析・検証実行サブシステム 1 0 へ送付する（F 4 ; ツール取得ステップ）。

20

【 0 0 5 3 】

（ 3 . 実施例の詳細 ）

ここでは、プログラム解析・検証サービス提供システム 1 の一実施例について詳細に説明する。

【 0 0 5 4 】

（ 3 . 1 . 構成 ）

図 1 および図 8 ~ 図 1 0 を参照して、プログラム解析・検証サービス提供システム 1 の構成について説明する。図 1 は、プログラム解析・検証サービス提供システム 1 の構成を示す機能ブロック図である。図 8 ~ 図 1 0 は、プログラム解析・検証サービス提供システム 1 において用いられるデータの説明図である。なお、以下の説明中、図 2 ~ 図 7 も適宜参照する。

30

【 0 0 5 5 】

プログラム解析・検証サービス提供システム 1 は、プログラム解析・検証ツールの利用者が、対象プログラム P を、ツール提供者によって提供されるプログラム解析・検証ツールを用いて解析、検証するサービスを提供するものである。

【 0 0 5 6 】

図 1 に示すように、プログラム解析・検証サービス提供システム 1 は、解析・検証部（プログラム解析・検証装置）1 0 0、対象プログラム管理部（プログラム解析・検証対象プログラム管理装置）2 0 0、ツール管理部（プログラム解析・検証ツール管理装置）3 0 0 を備えて構成されている。なお、プログラム解析・検証サービス提供システム 1 は、1 台の装置によって構成されていても良い。すなわち、解析・検証部 1 0 0、対象プログラム管理部 2 0 0、ツール管理部 3 0 0 が、1 台の装置に実装されていてもよい。また、プログラム解析・検証サービス提供システム 1 は、複数台の装置によって構成されていても良い。すなわち、解析・検証部 1 0 0、対象プログラム管理部 2 0 0、ツール管理部 3 0 0 が、2 台以上の装置に分散して実装されていてもよい。

40

【 0 0 5 7 】

また、解析・検証部 1 0 0、対象プログラム管理部 2 0 0、ツール管理部 3 0 0 は、図

50

2～図7に示した、解析・検証実行サブシステム10、対象プログラム管理サブシステム20、ツール管理サブシステム30にそれぞれ対応する。

【0058】

なお、本実施の形態では、プログラム解析・検証サービス提供システム1が外部と入出力する情報および内部で処理する情報は、タグの形式で記述されている場合を例として説明するが、タグを用いた記述形式に限定されるものではない。

【0059】

(3.1.1.解析・検証部)

まず、解析・検証部100は、利用者から対象プログラムPの解析、検証の実行要求を受け付け、その実行要求に従って、対象プログラムPおよびプログラム解析・検証ツール(仮想マシンT)を取得して、解析、検証を実行し、解析・検証結果データRを保存する機能を有する(図2)。また、解析・検証部100は、利用者から結果取得要求を受け付け、保存しておいた解析・検証結果データRを利用者に提示する機能を有する(図3)。

【0060】

そのため、解析・検証部100は、図1に示すように、解析・検証制御部110、仮想マシン実行環境部(ツール実行手段)120、データ記憶部130を備えている。なお、解析・検証制御部110、仮想マシン実行環境部120、データ記憶部130は、図2～図3に示した、解析・検証実行サブシステム10のコントローラ11、仮想マシン実行環境12、リポジトリ13にそれぞれ対応する。

【0061】

上記解析・検証制御部110は、解析・検証部100の全体を制御する。詳細には、解析・検証制御部110は、利用者インターフェース部(ツール指定情報取得手段)111と、ツール実行制御部112と、データ管理部113と、対象プログラム取得部(対象プログラム取得手段)114と、ツール取得部(ツール取得手段)115とを備える。

【0062】

利用者インターフェース部111は、利用者に対して操作画面を提示し、入力を受け付けるユーザインターフェースを提供する。利用者インターフェース部111は、ユーザの操作をキーボードやマウス等の入力装置から直接受け付けるものであってもよいし、インターネット等の通信を介して間接的に受け付けるものであってもよい。

【0063】

具体的には、利用者インターフェース部111は、利用者から操作を受け付ける際、利用者の識別情報のタグ(利用者識別情報)、利用者の権限を示すタグ(利用者権限情報)を取得する。

【0064】

また、利用者インターフェース部111は、利用者から実行要求を受け付ける(図2のA1)。実行要求には、どの対象プログラムPに対して解析・検証を行うかを示すタグ(対象プログラム指定情報)、どのプログラム解析・検証ツール(仮想マシンT)を用いるかを指定するタグ(ツール指定情報)が含まれる。このツール指定情報は、対象プログラム取得部114が取得した対象プログラムPの解析、検証に用いるプログラム解析・検証ツール(仮想マシンT)を指定するものである。

【0065】

また、利用者インターフェース部111は、利用者から結果取得要求を受け付ける(図3のB1)。結果取得要求には、実行要求と同様のタグが含まれる。なお、結果取得要求では、ツール指定情報は省略可能である。

【0066】

ツール実行制御部112は、利用者インターフェース部111が利用者から受け付けた実行要求に基づき、対象プログラムPの解析・検証を行う。具体的には、ツール実行制御部112は、対象プログラム取得部114を制御して、対象プログラム指定情報に基づき対象プログラムPを取得する(図2のA2-a、A3-a;対象プログラム取得ステップ)。また、ツール実行制御部112は、ツール取得部115を制御して、ツール指定情報

10

20

30

40

50

に基づき、プログラム解析・検証ツールのイメージを含む仮想マシンTを取得する（図2のA2-b、A3-b；ツール取得ステップ）。また、ツール実行制御部112は、仮想マシン実行環境部120に、取得した仮想マシンT上で、取得した対象プログラムPを実行させて、プログラムの解析・検証を行わせる（図2のA4；ツール実行ステップ）。なお、仮想マシンT上で対象プログラムPの解析・検証は、仮想マシンT上で対象プログラムPを実行させて行うことが可能であるが、実行させずに行うことも可能である。さらに、ツール実行制御部112は、仮想マシン実行環境部120での解析・検証の終了後、その結果を受け取り（図2のA5）、データ管理部113を制御して、その結果（解析・検証結果データR）を実行要求のタグと対応付けて、データ記憶部130に保存させる（図2のA6）。

10

**【0067】**

なお、ツール実行制御部112は、仮想マシン実行環境部120等の稼働状況（CPU利用率やメモリ利用率、外部記憶利用率など）に応じて、利用者からの実行要求がなくても、図2の（A2）以降の手順をあらかじめ行っておくことが可能である。この場合、データ記憶部130に結果（解析・検証結果データR）を保存する際に対応付けるタグは、対象プログラムP自身にタグ付けられたタグおよび仮想マシンT自身に付けられたタグとする。

**【0068】**

ツール実行制御部112は、利用者インターフェース部111が利用者から受け付けた結果取得要求に基づき、データ管理部113を制御して、データ記憶部130に保存させている解析・検証結果データRを取得し、利用者インターフェース部111に利用者に対して提示させる（図3のB1～B4）。具体的には、解析・検証結果データRの上記取得は、結果取得要求に含まれるタグにより依頼管理データベースDBiを検索し、該当した解析・検証結果データR（すなわち、当該タグに紐付けされている解析・検証結果データR）をデータ記憶部130から取得することによって行う。

20

**【0069】**

データ管理部113は、利用者から実行要求を受け付けたとき、含まれるタグの内容を、データ記憶部130の依頼管理データベースDBiに格納する。また、データ管理部113は、解析・検証結果データRをデータ記憶部130に格納する。このとき、依頼管理データベースDBiに、解析・検証結果データRの識別情報（結果データ識別情報）、実行要求のタグの情報に対応付けて格納する。また、データ管理部113は、利用者から結果取得要求を受け付けたとき、含まれるタグの内容に対応する解析・検証結果データRをデータ記憶部130から読み出す。

30

**【0070】**

対象プログラム取得部114は、対象プログラム管理部200へ対象プログラム指定情報を含む対象プログラム取得要求を送信し（図2のA2-a）、その応答として、対象プログラムPを取得する（図2のA3-a）。

**【0071】**

ツール取得部115は、ツール管理部300へツール指定情報を含む仮想マシン（ツールイメージ）取得要求を送信し（図2のA2-b）、その応答として、プログラム解析・検証ツール（仮想マシンT）を取得する（図2のA3-b）。

40

**【0072】**

さらに、上記仮想マシン実行環境部120は、ツール取得部115が取得した仮想マシンT上で、対象プログラム取得部114が取得した対象プログラムPを動作させることによって、対象プログラムPを解析、検証する（図2のA4）。

**【0073】**

データ記憶部130は、依頼管理データベースDBi、解析・検証結果データRを格納する。図8に示すように、依頼管理データベースDBiには、1レコードに、利用者識別情報、利用者権限情報、対象プログラム指定情報、ツール指定情報、結果データ識別情報が格納されている。解析・検証結果データRは、結果データ識別情報によって、他の情報

50

と対応付けられている。

【 0 0 7 4 】

ここで、図 8 に基づいて、依頼管理データベース D B i に登録される各情報について、具体的に説明する。

【 0 0 7 5 】

・利用者識別情報は、利用者の識別情報である。記述例の“ USER\_A, USER\_B ”は、利用者A、利用者Bに割当てられるタグを表す。

【 0 0 7 6 】

・利用者権限情報は、利用者のアクセス権限である。記述例の“ GROUP\_AB ”は、利用者A・Bで対象プログラムPおよびプログラム解析・検証ツール（仮想マシンT）を共有するためのグループ権限を示すタグを表す。また、“ PUBLIC ”は、任意の利用者がアクセス可能なことを示すタグを表す。また、“ ADMIN ”は、管理者のみがアクセス可能なことを示すタグを表す。

10

【 0 0 7 7 】

・対象プログラム指定情報は、解析・検証を行う対象である対象プログラムPを指定する情報である。対象プログラム指定情報は、対象プログラム管理データベース D B p の対象プログラム識別情報と対象プログラム関連情報との和に相当する。記述例の“ OBJP\_ID\_1, C, SYSTEM\_PROG ”は、「識別情報OBJP\_ID\_1を持ち、C言語プログラムであり、プログラムの種類がシステムプログラムである」対象プログラムを表す。

【 0 0 7 8 】

・ツール指定情報は、解析・検証に用いるプログラム解析・検証ツール（仮想マシンT）を指定する情報である。ツール指定情報は、ツール管理データベース D B t のツール識別情報とツール関連情報との和に相当する。記述例の“ TOOL\_ID\_0, C, USER\_PROG, SYSTEM\_PROG, CPU\_SPEED\_2G, MEM\_MAX\_1G ”は、「識別情報TOOL\_ID\_0を持ち、ユーザプログラムとシステムプログラムを対象にした検証が可能であり、CPUの周波数が2GHzで、メモリを最大で1G使用する」プログラム解析・検証ツール（仮想マシンT）を表す。

20

【 0 0 7 9 】

・結果データ識別情報は、解析・検証の結果データ（解析・検証結果データR）の識別情報である。記述例の“ RESULT\_0, RESULT\_1 ”は、解析・検証の結果データを参照するための識別情報を表す。

30

【 0 0 8 0 】

なお、依頼管理データベース D B i に、さらに、実行要求を直接特定する情報（例えば、連番や要求の発行日時）を設けてもよい。ただし、依頼管理データベース D B i に対して、対象プログラム指定情報とツール指定情報、すなわち、対象プログラムPとプログラム解析・検証ツール（仮想マシンT）が一意に指定されていれば、実行要求を特定することができるので、上記の情報は必須ではない。

【 0 0 8 1 】

（ 3 . 1 . 2 . 対象プログラム管理部 ）

対象プログラム管理部 2 0 0 は、利用者から対象プログラムPの登録要求を受け付け、その登録要求に従って、対象プログラムPを取得して、保存する機能を有する（図 4 ）。また、対象プログラム管理部 2 0 0 は、解析・検証部 1 0 0 から対象プログラム取得要求を受け付け、保存しておいた対象プログラムPを解析・検証部 1 0 0 へ送付する機能を有する（図 5 ）。

40

【 0 0 8 2 】

そのため、対象プログラム管理部 2 0 0 は、図 1 に示すように、対象プログラム管理制御部 2 1 0、対象プログラム記憶部 2 2 0 を備えている。なお、対象プログラム管理制御部 2 1 0、対象プログラム記憶部 2 2 0 は、図 4 ~ 図 5 に示した、対象プログラム管理サブシステム 2 0 のコントローラ 2 1、リポジトリ 2 2 にそれぞれ対応する。

【 0 0 8 3 】

上記対象プログラム管理制御部 2 1 0 は、対象プログラム管理部 2 0 0 の全体を制御す

50

る。詳細には、対象プログラム管理制御部 2 1 0 は、利用者入力部 2 1 1 と、対象プログラム取得部 2 1 2 と、対象プログラム登録・検索部 2 1 3 と、対象プログラム送付部 2 1 4 とを備える。

【 0 0 8 4 】

利用者入力部 2 1 1 は、利用者に対して操作画面を提示し、入力を受け付けるユーザインターフェースを提供する。利用者入力部 2 1 1 は、ユーザの操作をキーボードやマウス等の入力装置から直接受け付けるものであってもよいし、インターネット等の通信を介して間接的に受け付けるものであってもよい。

【 0 0 8 5 】

具体的には、利用者入力部 2 1 1 は、利用者から操作を受け付ける際、利用者の識別情報のタグ（利用者識別情報）、利用者の権限を示すタグ（利用者権限情報）を取得する。

10

【 0 0 8 6 】

また、利用者入力部 2 1 1 は、利用者から対象プログラム P の登録要求を受け付ける（図 4 の C 1 ）。登録要求には、利用者が指定したタグ（対象プログラム関連情報）と、対象プログラム P の送付方法の指定（送付方法指定情報）のタグと、解析・検証を行う対象である対象プログラム P の識別情報（対象プログラム識別情報）が含まれる。

【 0 0 8 7 】

対象プログラム取得部 2 1 2 は、利用者入力部 2 1 1 が利用者から受け付けた登録要求に基づき、対象プログラム P を取得する。具体的には、登録要求に含まれる送付方法指定情報が「直接送付」である場合には、利用者が送信する対象プログラム P を取得する（図 4 の C 2 ）。また、登録要求に含まれる送付方法指定情報が「インポート」である場合には、外部システム（バージョン管理システムなど）から対象プログラム P を取得する（図 4 の C 2 - 1、C 2 - 2 ）。

20

【 0 0 8 8 】

対象プログラム登録・検索部 2 1 3 は、利用者入力部 2 1 1 が利用者から登録要求を受け付けたとき、含まれるタグの内容を、対象プログラム記憶部 2 2 0 の対象プログラム管理データベース DB p に格納する（図 4 の C 3 ）。また、対象プログラム登録・検索部 2 1 3 は、対象プログラム取得部 2 1 2 が取得した対象プログラム P を対象プログラム記憶部 2 2 0 に格納する（図 4 の C 3 ）。このとき、対象プログラム管理データベース DB p に、対象プログラム P の識別情報（対象プログラム識別情報）、登録要求のタグの情報に対応付けて格納する。

30

【 0 0 8 9 】

また、対象プログラム登録・検索部 2 1 3 は、解析・検証部 1 0 0 から対象プログラム取得要求を受け付けたとき、対象プログラム管理データベース DB p を参照して、対象プログラム取得要求に含まれる対象プログラム指定情報（対象プログラム識別情報と対象プログラム関連情報の和）に対応する対象プログラム P を対象プログラム記憶部 2 2 0 から読み出す（図 5 の D 1 ~ D 4 ）。

【 0 0 9 0 】

対象プログラム送付部 2 1 4 は、解析・検証部 1 0 0 から対象プログラム識別情報を含む対象プログラム取得要求を受信し（図 5 の D 1 ）、その応答として、対象プログラム P を解析・検証部 1 0 0 へ送信する（図 5 の D 4 ）。

40

【 0 0 9 1 】

対象プログラム記憶部 2 2 0 は、対象プログラム管理データベース DB p、対象プログラム P を格納する。図 9 に示すように、対象プログラム管理データベース DB p には、1 レコードに、利用者識別情報、利用者権限情報、対象プログラム関連情報、送付方法指定情報、対象プログラム識別情報が格納されている。対象プログラム P は、対象プログラム識別情報によって、他の情報と対応付けられている。

【 0 0 9 2 】

ここで、図 9 に基づいて、対象プログラム管理データベース DB p に登録される各情報について、具体的に説明する。

50

## 【 0 0 9 3 】

・利用者識別情報は、利用者の識別情報である。記述例の“USER\_A, USER\_B”は、利用者A、利用者Bに割当てられるタグを表す。

## 【 0 0 9 4 】

・利用者権限情報は、利用者のアクセス権限である。記述例の“GROUP\_AB”は、利用者A・Bで対象プログラムPおよびプログラム解析・検証ツール（仮想マシンT）を共有するためのグループ権限を示すタグを表す。また、“PUBLIC”は、任意の利用者がアクセス可能なことを示すタグを表す。また、“ADMIN”は、管理者のみがアクセス可能なことを示すタグを表す。

## 【 0 0 9 5 】

・対象プログラム関連情報は、利用者が任意の内容で任意の個数指定できる情報である。対象プログラム関連情報としては、例えば、対象プログラムPで用いられているプログラミング言語の種類、プログラムの種類（ユーザプログラム、システムプログラムの区別など）などが記述できる。記述例には、「プログラミング言語の種類」および「プログラムの種類」が例として挙げられている。プログラミング言語の種類の例である“Java, C, C++, OCaml”は、対象プログラムがどんなプログラミング言語で記述されているかを表すタグの種類を例示している。また、プログラムの種類の例である“USER\_PROG, SYSTEM\_PROG, LIBRARY”は、対象プログラムの種類（ユーザプログラム、システムプログラム、ライブラリなど）を例示している。

## 【 0 0 9 6 】

・送付方法指定情報は、対象プログラムPの送付方法の指定、例えば、利用者から直接送付されるのか、外部システムからインポートされるのかが記述される。記述例の“DIRECT\_UPLOAD”は、対象プログラムPが直接利用者からアップロードされていることを示すタグを表す。また、“IMPORT\_git://XX...”は、対象プログラムPを外部のレポジトリ（ここでは、git://XXX...）から取得する、もしくは、取得したことを示すタグを表す。

## 【 0 0 9 7 】

・対象プログラム識別情報は、解析・検証を行う対象である対象プログラムPの識別情報を示す。記述例の“OBJP\_ID\_0, OBJP\_ID\_1”は、保存してある対象プログラムPを参照するための識別情報を示すタグを表す。

## 【 0 0 9 8 】

## （ 3 . 1 . 3 . ツール管理部 ）

ツール管理部300は、ツール提供者からプログラム解析・検証ツールの登録要求を受け付け、その登録要求に従って、プログラム解析・検証ツールを導入、設定した仮想マシンTを生成して、保存する機能を有する（図6）。また、ツール管理部300は、解析・検証部100から仮想マシン取得要求を受け付け、このとき取得したツール指定情報に基づいて検索して抽出された仮想マシンTを解析・検証部100へ送付する機能を有する（図7）。

## 【 0 0 9 9 】

そのため、ツール管理部300は、図1に示すように、ツール管理制御部310、ツール記憶部320を備えている。なお、ツール管理制御部310、ツール記憶部320は、図6～図7に示した、ツール管理サブシステム30のコントローラ31、リポジトリ32にそれぞれ対応する。そして、ツール管理サブシステム30の仮想マシン実行環境33については、図1のツール管理部300では、ツール管理制御部310の仮想マシン生成部312が対応する機能を有するものとする。

## 【 0 1 0 0 】

上記ツール管理制御部310は、ツール管理部300の全体を制御する。詳細には、ツール提供者入力部311と、仮想マシン生成部（仮想マシン生成手段）312と、ツール登録・検索部（ツール抽出手段）313と、ツール送付部314とを備える。

## 【 0 1 0 1 】

ツール提供者入力部311は、ツール提供者に対して操作画面を提示し、入力を受け付

10

20

30

40

50



けるユーザインターフェースを提供する。ツール提供者入力部 3 1 1 は、ユーザの操作をキーボードやマウス等の入力装置から直接受け付けるものであってもよいし、インターネット等の通信を介して間接的に受け付けるものであってもよい。

【 0 1 0 2 】

具体的には、ツール提供者入力部 3 1 1 は、ツール提供者から操作を受け付ける際、ツール提供者の識別情報のタグ（ツール提供者識別情報）を取得する。

【 0 1 0 3 】

また、ツール提供者入力部 3 1 1 は、ツール提供者からプログラム解析・検証ツールの登録要求を受け付ける（図 6 の E 1）。登録要求には、ツール提供者が指定したタグ（ツール関連情報）と、プログラム解析・検証ツールの識別情報（ツール識別情報）のタグが含まれる。なお、ツール識別情報は、解析・検証に用いるプログラム解析・検証ツールが導入・設定されたツールイメージを含む仮想マシン T の識別情報として使われる。また、ツール提供者入力部 3 1 1 は、登録要求に伴い、プログラム解析・検証ツールを受け付ける（図 6 の E 3）。

10

【 0 1 0 4 】

仮想マシン生成部 3 1 2 は、プログラム解析・検証ツールを導入、設定した仮想マシン T を、プログラム解析・検証ツール毎に生成する（図 6 の E 2）。仮想マシン T の生成には、登録要求に含まれるツール関連情報が、プログラム解析・検証ツールを仮想マシンに導入、設定するためのデータとして使われる。仮想マシン生成部 3 1 2 は、初期仮想マシン生成部 3 1 2 1 と、ツール導入部 3 1 2 2 とを備える。

20

【 0 1 0 5 】

初期仮想マシン生成部 3 1 2 1 は、ツール関連情報に基づいて、プログラム解析・検証ツールが導入、設定される前の状態の仮想マシン（初期仮想マシン）を生成する。これにより、例えば、プログラム解析・検証ツールが扱うことができるプログラミング言語の種類やプログラムの種類（ユーザプログラム、システムプログラムの区別など）、プログラム解析・検証ツールが稼働する実行環境（オペレーティングシステムやライブラリの種類・バージョン、必要となる計算資源、すなわち CPU の速度や種類、メモリやディスクのサイズなど）に適合した初期仮想マシンが得られる。

【 0 1 0 6 】

ツール導入部 3 1 2 2 は、初期仮想マシン生成部 3 1 2 1 が生成した仮想マシン（初期仮想マシン）を実行し、ツール提供者の指示等に基づいて、プログラム解析・検証ツールを導入、設定する（図 6 の E 3）。

30

【 0 1 0 7 】

ツール登録・検索部 3 1 3 は、ツール提供者入力部 3 1 1 がツール提供者から登録要求を受け付けたとき、含まれるタグの内容を、ツール記憶部 3 2 0 のツール管理データベース DB t に格納する（図 6 の E 4）。また、ツール登録・検索部 3 1 3 は、仮想マシン生成部 3 1 2 が生成した仮想マシン T をツール記憶部 3 2 0 に格納する（図 4 の C 3）。このとき、ツール登録・検索部 3 1 3 は、仮想マシン T をツールイメージのデータとしてツール記憶部 3 2 0 に格納する。

【 0 1 0 8 】

また、ツール登録・検索部 3 1 3 は、ツール管理データベース DB t に、仮想マシン T の識別情報（ツール識別情報）等に対応付けて格納する。また、ツール登録・検索部 3 1 3 は、解析・検証部 1 0 0 から仮想マシン（ツールイメージ）取得要求を受け付けたとき、ツール管理データベース DB t を参照して、仮想マシン（ツールイメージ）取得要求に含まれるツール指定情報に適合したツール関連情報を検索する。そして、ツール指定情報に適合したツール関連情報に対応する仮想マシン（ツールイメージ） T をツール記憶部 3 2 0 から読み出す（図 7 の F 1 ~ F 4）。

40

【 0 1 0 9 】

ツール送付部 3 1 4 は、解析・検証部 1 0 0 からツール指定情報を含む仮想マシン（ツールイメージ）取得要求を受信し（図 7 の F 1）、その応答として、ツール登録・検索部

50

3 1 3 によって抽出された 1 または複数の仮想マシン T を解析・検証部 1 0 0 へ送信する ( 図 7 の F 4 ) 。

【 0 1 1 0 】

ツール記憶部 3 2 0 は、ツール管理データベース D B t 、仮想マシン ( イメージデータ ) T を格納する。図 1 0 に示すように、ツール管理データベース D B t には、1 レコードに、ツール提供者識別情報、ツール関連情報、ツール識別情報が格納されている。仮想マシン ( イメージデータ ) T は、ツール識別情報によって、他の情報と対応付けられている。

【 0 1 1 1 】

ここで、図 1 0 に基づいて、ツール管理データベース D B t に登録される各情報について、具体的に説明する。

【 0 1 1 2 】

・ツール提供者識別情報は、ツール提供者の識別情報である。記述例の “ USER\_A, USER\_B ” は、利用者 A、利用者 B に割当てられるタグを表す。

【 0 1 1 3 】

・ツール関連情報は、ツール提供者が任意の内容で任意の個数指定できる情報である。ツール関連情報としては、例えば、プログラム解析・検証ツール ( 仮想マシン T ) が扱うことができるプログラミング言語の種類やプログラムの種類 ( ユーザプログラム、システムプログラムの区別など )、プログラム解析・検証ツール ( 仮想マシン T ) が稼働する実行環境 ( オペレーティングシステムやライブラリの種類・バージョン、必要となる計算資源 ( C P U の速度や種類、メモリやディスクのサイズなど ) など) が記述できる。記述例には、「プログラミング言語の種類」、「プログラムの種類」、「実行環境・計算資源」が例として挙げられている。プログラミング言語の種類の例である “ Java, C, C++, OCaml ” は、プログラム解析・検証ツール ( 仮想マシン T ) がどんなプログラミング言語で記述されたプログラムを対象としているかを示すタグの種類を例示している。また、プログラムの種類の例である “ USER\_PROG, SYSTEM\_PROG, LIBRARY ” は、プログラム解析・検証ツール ( 仮想マシン T ) が解析・検証の対象とできるプログラムの種類 ( ユーザプログラム、システムプログラム、ライブラリなど ) を例示している。また、実行環境・計算資源の例である “ OS\_Linux 3.X, LIB\_libgplk 1.X, CPU\_KIND\_X86, CPU\_SPEED\_2G, CPU\_NUM\_8, MEM\_MAX\_1G, DISK\_MAX\_4G ” は、プログラム解析・検証ツール ( 仮想マシン T ) が実行可能

【 0 1 1 4 】

・ツール識別情報は、解析・検証に用いるプログラム解析・検証ツール ( 仮想マシン T ) が導入・設定されたツールイメージを含む仮想マシン T の識別情報である。記述例の “ TOOL\_ID\_0, TOOL\_ID\_1 ” は、保存してあるプログラム解析・検証ツール、より具体的には、プログラム解析・検証ツールが導入・設定されたツールイメージを含む仮想マシン T を参照するための識別情報である。

【 0 1 1 5 】

( 4 . 従来技術との比較 )

ここでは、本実施の形態に係るプログラム解析・検証サービス提供システム 1 と、従来技術、特にウイルス解析・検査サイト ( 非特許文献 1 ) との違いについて、〔発明が解決しようとする課題〕として挙げた問題ごとに説明する。

【 0 1 1 6 】

( 1 ) プログラム解析・検証ツールが要する計算資源の肥大化

この問題については、従来のウイルス解析・検査サイトは、利用者が計算資源を用意する必要はないが、解析・検証ツールの提供者が独自に計算資源を用意したり、また新たに W e b インターフェースを作成したりする必要がある。

10

20

30

40

50

## 【 0 1 1 7 】

これに対し、上記プログラム解析・検証サービス提供システム1では、複数のプログラム解析・検証ツールの提供者間で計算資源を共有することができるため、ツール提供者の負担が大きく軽減される。また、ツール提供者は、(2.3. ツール管理サブシステムの一実装例)で述べたように、あらかじめ用意された仮想マシンに対してプログラム解析・検証ツールの導入・設定を行うだけでよいため、新たにWebインターフェースなどを作成するなどの余計な手間も掛ける必要がない。

## 【 0 1 1 8 】

## (2) プログラム解析・検証ツールの種類の多様化

この問題については、非特許文献1に記載されているとおり、多数のウイルス解析・検査サイトが存在するが、個々のサイトは異なるツール提供者によって独自に運営されており、統一的なインターフェースは存在しない。そのため、プログラム解析・検証ツールの利用者にとって、どのプログラム解析・検証ツールを選択すれば良いか、すなわち、どのウイルス解析・検査サイトを選択すれば良いかを判断することが困難である。

10

## 【 0 1 1 9 】

## (3) プログラム解析・検証ツールの導入・運用の手間・コスト

この問題についても同様に、多数のウイルス解析・検査サイトが異なるツール提供者によって独自に運営されており、統一的なインターフェースも存在しないため、複数のサイトを効率よく利用して、解析・検証を行ったり、その結果を収集・比較したりすることは困難である。

20

## 【 0 1 2 0 】

この問題の簡単な回避策として、複数のプログラム解析・検証サイトをバックエンドとして利用し、フロントエンドに統一的なインターフェースを提供するサイトを設けることが考えられる。すなわち、利用者から対象プログラムを受け取り、それを複数のプログラム解析・検証サイトに送付し、結果を収集して利用者へ送付する方法である。

## 【 0 1 2 1 】

しかし、以下に述べるとおり、この方法でも依然として問題が残る。まず、単純に複数のプログラム解析・検証サイトに対象プログラムを送るだけでは、利用者が必要としているプログラム解析・検証ツールを提供するプログラム解析・検証サイト以外のサイトにも送られてしまう。その結果、解析・検証の効率が悪く、また解析・検証の結果も膨大となり、比較検討が容易ではなくなる。これに対して、上記プログラム解析・検証サービス提供システム1では、(2. プログラム解析・検証ツールの共有のためのシステムの実装例)で述べたように、対象プログラムやプログラム解析・検証ツールをタグ付けして管理して、必要なプログラム解析・検証ツールを効率的に選択できるようになっている。

30

## 【 0 1 2 2 】

また、別の問題として、単純に複数のプログラム解析・検証サイトに対象プログラムを送るだけでは、解析・検証の結果は利用者が保存・管理しなければならず、利用者にとって計算資源の負担が大きくなるか、または同じ解析・検証を不要に繰り返すことになる可能性がある。これに対し、上記プログラム解析・検証サービス提供システム1では、(2. プログラム解析・検証ツールの共有のためのシステムの実装例)で述べたように、解析・検証の結果をリポジトリに統一的に保存し、タグ付けを利用した結果の検索などを行えるようになっている。

40

## 【 0 1 2 3 】

さらに、複数のサイトをバックエンドとして用いる手法では、(1)(4)の問題にも対処できない。

## 【 0 1 2 4 】

## (4) プログラム解析・検証ツール自体のテストの困難さ

この問題に対しては、まず、個々の解析・検証ツールの提供者が準備できる計算資源には限界があり、実際に解析・検証を行える対象のサイズや複雑さには限界がある。そのため、解析・検証ツール自体のテストが困難であるという問題は、従来のウイルス解析・検

50

査サイトにもある。

【 0 1 2 5 】

これに対し、上記プログラム解析・検証サービス提供システム1では、複数のプログラム解析・検証ツールの提供者の間で計算資源を共有する。その結果、よりサイズが大きく複雑な対象を解析・検証できるため、プログラム解析・検証ツール自体のテストをより厳密に行うことができる。また、解析・検証対象のプログラムを複数のプログラム解析・検証ツールで共有することができるため、より多くの対象についてプログラム解析・検証ツールのテストを行える。

【 0 1 2 6 】

以上のように、従来のウイルス解析・検査サイトは、利用者が解析・検査のための計算資源を用意したり、解析・検査ソフトウェアを導入したりする手間がかからないという利点があるが、本発明が解決しようとする課題（（1）～（4））をそのまま問題点として有している。

【 0 1 2 7 】

（5．多段階の解析・検証への応用など）

上記の記載では、プログラム解析・検証サービス提供システム1がプログラム解析・検証ツールを用いて、対象プログラムを1段階で解析・検証する場合について説明した。しかしながら、プログラム解析・検証サービス提供システム1は、多段階の解析・検証によって対象プログラムの解析・検証を行う場合にも、好適に利用することができる。

【 0 1 2 8 】

そして、例えば、プログラム解析・検証ツールを用いて対象プログラムを解析・検証した後、当該解析・検証で得た結果データに対してさらに解析・検証を行う場合には、各段階の解析・検証を行うツール指定情報、およびツール関連情報を対象プログラム、あるいは結果データに対して適切にタグ付けすることが好ましい。すなわち、プログラム解析・検証サービス提供システム1では、各段階の解析・検証等においてもタグを用いることで、多段階の解析・検証の工程を経て対象プログラムの解析・検証を行う場合も効率的に処理することができる。

【 0 1 2 9 】

まず、プログラム解析・検証サービス提供システム1において、タグを各段階の解析・検証に用いて、多段階の解析・検証によって対象プログラムを解析・検証する例を以下に示す。

【 0 1 3 0 】

ここでは、プログラム解析・検証サービス提供システム1が、対象プログラムをプログラム解析・検証ツールを用いて解析・検証する処理と、当該処理によって得た結果データに対する解析・検証（後処理）を連続して行う場合について説明する。

【 0 1 3 1 】

まず、仮想マシン実行環境部（ツール実行手段）120がプログラム解析・検証ツールを用いて対象プログラムを解析・検証して得た結果データを、結果データの特徴記述情報（タグ）と対応付けて、データ記憶部（結果データ記憶部）130に格納する。一方、ツール記憶部320は結果データを解析・検証するプログラム解析・検証ツール（後処理ツール）を当該解析・検証ツール関連情報と対応付けてツール記憶部320に格納している。

【 0 1 3 2 】

次に、利用者インターフェース部（ツール指定情報取得手段）111は、データ記憶部130から、結果データに対応付けられた特徴記述情報を、当該ツール指定情報として取得する。

【 0 1 3 3 】

ツール登録・検索部（ツール抽出手段）313は、利用者インターフェース部111が取得したツール指定情報に基づいて、ツール関連情報を検索し、適合したツール関連情報と対応付けられた後処理ツールを、ツール記憶部320に格納された複数の後処理ツール

10

20

30

40

50

の中から抽出する。

【0134】

そして、プログラム解析・検証サービス提供システム1は、上記のように抽出された後処理ツールを用いて、対象プログラムを解析・検証する処理の結果データに対する解析・検証（後処理）を連続して行う。

【0135】

上記の例で、解析・検証をする処理の結果データに好適に付けられる特徴記述情報（タグ）としては、例えば、データの種類を表すタグ（例えば、"execution\_path"（実行パスを表す）、"execution\_trace"（実行トレースを表す）、"program\_point"（プログラム中の実行ポイントを表す））などが考えられる。また、データ解析ツールに対しても、プログラム解析・検証と同様に、ツール指定情報（タグ）を付けることができる。

10

【0136】

なお、対象プログラムを解析・検証する処理段階の数は2段階に限定されることはなく、任意の数の処理段階を用いた処理であってもよい。

【0137】

また、ここではプログラム解析・検証サービス提供システム1がプログラム解析・検証ツールを用いて対象プログラムに対して解析・検証する処理を行った結果データを、続けて解析・検証する場合について説明したが、他のシステムにおいて解析・検証を行った結果のデータに対して、プログラム解析・検証サービス提供システム1が解析・検証を行う場合にも、本発明を適用することができる。つまり、多段階の解析・検証を行う処理において、プログラム解析・検証サービス提供システム1がその解析・検証の処理をすべて行ってもよいし、各段階の処理を他のシステムが分担して行うようにすることも可能である。

20

【0138】

具体的には、他のシステムが解析の結果として出力した実行パスをもとに、プログラム解析・検証サービス提供システム1がより精度の高い解析・検証を行う場合などが挙げられる。

【0139】

プログラム解析・検証サービス提供システム1のプログラム解析・検証ツールではない、他のシステムのプログラム解析・検証ツールが対象プログラムを解析・検証した場合も、その解析・検証の結果として出力された結果データにタグ（特徴記述情報）付けを行うことで、データ解析ツールによる当該結果のデータを解析・検証することができる。

30

【0140】

すなわち、他のシステムのプログラム解析・検証ツールが対象プログラムを解析・検証した場合においても、出力される解析・検証結果のデータにタグ付けを行い、保存しておくことで、本発明の枠組みを利用することが可能である。

【0141】

さらに、本発明は、上記のような、対象プログラムの解析・検証、および出力される解析・検証結果のデータを対象にした解析・検証の処理（後処理）に限定的に適用されるものではない。例えば、対象プログラムの解析・検証を行うために前処理が必要である場合においても、タグ付けを好適に応用することが可能である。

40

【0142】

次に、プログラム解析・検証サービス提供システム1が解析・検証する対象プログラムを、前処理前プログラムに対して前処理を行う例を以下に示す。

【0143】

プログラム解析・検証ツールの種類によっては、プログラム（前処理前プログラム）に対して前処理を行った結果として対象プログラム（前処理後プログラム）とする必要がある場合が想定される。

【0144】

すなわち、まず第1段階として、前処理ツール（ツール）を用いて前処理前プログラム

50

に対して前処理を行った結果である対象プログラムを出力し、対象プログラム記憶部（前処理後プログラム記憶部）220に当該対象プログラムを記憶する。次に第2段階として、当該前処理によって得た対象プログラムに対してプログラム解析・検証ツールを用いて解析・検証する処理を連続して行う。

【0145】

この場合には、前処理前のプログラムを前処理することで得た対象プログラムは、対象プログラムの特徴記述情報（タグ）と対応付けて対象プログラム記憶部（前処理後プログラム記憶部）220に格納される。

【0146】

次に、対象プログラム取得部114は、対象プログラム記憶部220から対象プログラムを取得する。一方、利用者インターフェース部（ツール指定情報取得手段）111は、対象プログラム記憶部220から対象プログラムが取得した対象プログラムに対応付けられた特徴記述情報を、ツール指定情報として取得する。

10

【0147】

このように、あるプログラムの解析・検証を行うために、プログラムの前処理を必要とする処理としては、例えば、データフロー解析、制御フロー解析、プログラムスライシング、ポインタ解析などがある。

【0148】

そして、前処理ツール（ツール）および対象プログラム（前処理後プログラム）にタグ付けを行い、保存しておくことで、本発明の枠組みを利用することができる。

20

【0149】

実際、前処理ツールは、解析・検証ツールと同様に様々な種類があり、計算資源を多く消費するものもあるため、解析・検証ツールの場合と同様、本発明は有効である。

【0150】

例えば、前処理ツール（ツール）に付けられるタグ（ツール関連情報）としては、前処理可能な対象を表すタグ（検証ツールに付けるタグと同様のもの）、および、どのような前処理を行うのかを表すタグ（例えば、"data\_flow\_analysis"、"control\_flow\_analysis"、"program\_slicing"、"pointer\_analysis"など前処理の種類を示したタグ）が考えられる。また、前処理結果のデータに対応付けられる特徴記述情報（タグ）としては、"data\_flow\_analysis"、"control\_flow\_analysis"、"program\_slicing"、"pointer\_analysis"などが有効と考えられる。

30

【0151】

また、上述の二つの例を組み合わせ例として、プログラム解析・検証サービス提供システム（データ処理サービス提供システム）1において、対象プログラムを、前処理ツール（前処理データ処理ツール）を用いて前処理を行い、次いで前処理後の対象プログラムに対してプログラム解析・検証ツールを用いて解析・検証を行い、更にその解析・検証の結果データに対して別の解析・検証ツール（後処理データ処理ツール）を用いて解析・検証を行う例を示す。

【0152】

まず第1段階として、前処理ツール（前処理データ処理ツール）を用いて前処理前プログラムに対して前処理を行った結果である対象プログラムを出力し、対象プログラム記憶部（前処理後プログラム記憶部）220に当該対象プログラムを記憶する。前処理前のプログラムを前処理することで得た対象プログラムは、前述の通り、対象プログラムの特徴記述情報（タグ）と対応付けて対象プログラム記憶部（前処理後プログラム記憶部）220に格納される。

40

【0153】

このとき、前処理後の対象プログラムに好適に付けられる特徴記述情報（タグ）としては、上述の例のように、例えば前処理の種類を表すタグ（"data\_flow\_analysis"、"control\_flow\_analysis"、"program\_slicing"、"pointer\_analysis"など）などが考えられる。

【0154】

50

次いで第2段階として、前処理によって得た対象プログラムに対してプログラム解析・検証ツールを用いて解析・検証を行う。具体的にはまず、対象プログラム取得部114は、対象プログラム記憶部220から対象プログラムを取得する。一方、利用者インターフェース部（ツール指定情報取得手段）111は、対象プログラム記憶部220から対象プログラムが取得した対象プログラムに対応付けられた特徴記述情報を、ツール指定情報として取得する。ついで仮想マシン実行環境部（ツール実行手段）120がプログラム解析・検証ツールを用いて対象プログラムを解析・検証し、その結果得られた結果データを、結果データの特徴記述情報（タグ）と対応付けて、データ記憶部（結果データ記憶部）130に格納する。

【0155】

このとき、結果データに好適に付けられる特徴記述情報（タグ）としては、上述の例のように、例えばデータの種類を表すタグ（"execution\_path"（実行パスを表す）、"execution\_trace"（実行トレースを表す）、"program\_point"（プログラム中の実行ポイントを表す）など）などが考えられる。

【0156】

最後に第3段階として、利用者インターフェース部（ツール指定情報取得手段）111は、データ記憶部130から、結果データに対応付けられた特徴記述情報を、当該ツール指定情報として取得する。ここで、ツール記憶部320は、結果データを解析・検証するプログラム解析・検証ツール（後処理データ処理ツール）を当該解析・検証ツール関連情報と対応付けてツール記憶部320に格納している。そして、ツール登録・検索部（ツール抽出手段）313は、利用者インターフェース部111が取得したツール指定情報に基づいて、ツール関連情報を検索し、適合したツール関連情報と対応付けられた後処理ツール（後処理データ処理ツール）を、ツール記憶部320に格納された複数の後処理ツールの中から抽出する。その後、プログラム解析・検証サービス提供システム1は、抽出したデータ処理ツールをツール記憶部320から取得し、取得された後処理ツールを用いて、対象プログラムを解析・検証する処理の結果データに対する解析・検証（後処理）を連続して行う。

【0157】

なお、上述の組合せの例では前処理と後処理の数が一つずつとなっているが、任意の数の前処理、後処理、解析・検証処理を組み合わせることも可能である。また、前処理、後処理、解析・検証処理のいずれかを省略することも可能である。

【0158】

このように、多段階のデータ処理を行う場合、プログラム解析・検証サービス提供システム1は、各段階の処理の結果得られるデータ（プログラムまたはプログラム以外のデータ）に特徴記述情報（タグ）を対応付けて保存（格納）する。各段階の処理結果のデータに対して特徴記述情報（タグ）との対応付けが行われるのは、データを保存すると同時、あるいは、データの保存後で次の段階の処理で用いるツールを抽出するまでの間である。この際、対応付けられる特徴記述情報（タグ）は、各段階の処理で使用したツールに対応した、所定のルールに従って決定されることが好ましい。なお、プログラム解析・検証サービス提供システム1のユーザが手作業で、各段階の処理で得られるデータに特徴記述情報（タグ）を対応付けするようにしてもよい。

【0159】

上述の通り、本発明は、プログラムに前処理を行うことで対象プログラムを得る処理（前処理）と、得られた対象プログラムを解析・検証する処理とを連続して行う、多段階の解析・検証に適用した場合、以下のように構成してもよい。

【0160】

プログラム解析・検証サービス提供システム（プログラム解析・検証サービス提供システム1）は、前処理前プログラムを前処理して対象プログラムを得る処理と、該処理によって得た対象プログラムをプログラム解析・検証ツールを用いて解析、検証する処理とを連続して行うものであって、前処理前プログラムを前処理して得た対象プログラムを、該

10

20

30

40

50

対象プログラムの特徴記述情報（ツール指定情報）と対応付けて格納する前処理後プログラム記憶部（対象プログラム記憶部 220）をさらに備え、対象プログラム取得手段（対象プログラム取得部 114）は、前処理後プログラム記憶部から、対象プログラムを取得するものであり、ツール指定情報取得手段（利用者インターフェース部 111）は、前処理後プログラム記憶部から、対象プログラム取得手段が取得した対象プログラムに対応付けられた特徴記述情報を、ツール指定情報として取得するものであることを特徴としている。

【0161】

また、本発明は、対象プログラムに解析・検証する処理に続けて、得られた結果データを解析・検証する処理を行う多段階の解析・検証に適用した場合、以下のように構成してもよい。

10

【0162】

プログラム解析・検証サービス提供システム（プログラム解析・検証サービス提供システム 1）は、対象プログラムをプログラム解析・検証ツールを用いて解析、検証する処理と、該処理によって得た結果データを後処理する処理とを連続して行うものであって、ツール実行手段（仮想マシン実行環境部 120）がプログラム解析・検証ツールを用いて対象プログラムを解析、検証して得た結果データを、該結果データの特徴記述情報（ツール指定情報）と対応付けて格納する結果データ記憶部（データ記憶部 130）をさらに備えるとともに、ツール記憶部（ツール記憶部 320）は、結果データを後処理する後処理ツールを該後処理ツールのツール関連情報と対応付けて格納しており、ツール指定情報取得手段（利用者インターフェース部 111）は、結果データ記憶部から、結果データに対応付けられた特徴記述情報を、ツール指定情報として取得するものであり、ツール抽出手段（ツール登録・検索部 313）は、ツール指定情報取得手段が取得したツール指定情報に基づいて、ツール関連情報を検索し、適合したツール関連情報と対応付けられた後処理ツールを、ツール記憶部に格納された複数の後処理ツールから抽出するものであることを特徴としている。

20

【0163】

さらに、本発明は、プログラムの解析・検証を行う処理を必須としないサービス提供システムとして構成する場合、以下のように構成することも可能である。

【0164】

30

本発明に係るデータ処理サービス提供システム（プログラム解析・検証サービス提供システム 1）は、複数のデータ処理ツール（プログラム解析・検証ツール、前処理ツール、後処理ツール）を順次用いることにより、対象データ（対象プログラム、結果データ）を複数段階にデータ処理するデータ処理サービス提供システムであって、複数のデータ処理ツールを各データ処理ツールのツール関連情報とそれぞれ対応付けて格納したツール記憶部（ツール記憶部 320）と、前段階のデータ処理ツール（前処理ツール、プログラム解析・検証ツール）を用いてデータ処理して得られた結果データを、該結果データの特徴記述情報（ツール指定情報）と対応付けて格納する結果データ記憶部（データ記憶部 130）と、結果データ記憶部から結果データを取得する結果データ取得手段（対象プログラム取得部 114）と、結果データ取得手段が取得した結果データのデータ処理に用いるデータ処理ツール（プログラム解析・検証ツール、後処理ツール）を指定するツール指定情報として、当該結果データに対応付けられた特徴記述情報を結果データ記憶部から取得するツール指定情報取得手段（利用者インターフェース部 111）と、ツール指定情報取得手段が取得したツール指定情報に基づいて、ツール関連情報を検索し、適合したツール関連情報と対応付けられたデータ処理ツールを、ツール記憶部に格納された複数のデータ処理ツールから抽出するツール抽出手段（ツール登録・検索部 313）と、上記ツール抽出手段が抽出したデータ処理ツールを上記ツール記憶部から取得するツール取得手段（ツール取得部 115）と、ツール取得手段が取得したデータ処理ツールを用いて、結果データ取得手段が取得した結果データをデータ処理するツール実行手段（仮想マシン実行環境部 120）と、を備えることを特徴としている。

40

50



## 【 0 1 6 5 】

## ( 6 . ま と め )

本発明に係るプログラム解析・検証サービス提供システムは、対象プログラムをプログラム解析・検証ツールを用いて解析、検証するプログラム解析・検証サービス提供システムであって、対象プログラムを取得する対象プログラム取得手段と、複数のプログラム解析・検証ツールを格納したツール記憶部と、上記対象プログラム取得手段が取得した対象プログラムの解析、検証に用いるプログラム解析・検証ツールを、上記ツール記憶部に格納された複数のプログラム解析・検証ツールから抽出するツール抽出手段と、上記ツール抽出手段が抽出したプログラム解析・検証ツールを上記ツール記憶部から取得するツール取得手段と、上記ツール取得手段が取得した各プログラム解析・検証ツールを用いて、上記対象プログラム取得手段が取得した対象プログラムを解析、検証するツール実行手段と、を備えることを特徴としている。

10

## 【 0 1 6 6 】

また、本発明に係るプログラム解析・検証サービス提供システムの制御方法は、対象プログラムをプログラム解析・検証ツールを用いて解析、検証するプログラム解析・検証サービス提供システムの制御方法であって、上記プログラム解析・検証サービス提供システムは、複数のプログラム解析・検証ツールを格納したツール記憶部を備え、対象プログラムを取得する対象プログラム取得ステップと、上記対象プログラム取得ステップにて取得した対象プログラムの解析、検証に用いるプログラム解析・検証ツールを、上記ツール記憶部に格納された複数のプログラム解析・検証ツールから抽出するツール抽出ステップと、上記ツール抽出ステップにて抽出したプログラム解析・検証ツールを上記ツール記憶部から取得するツール取得ステップと、上記ツール取得ステップにて取得した各プログラム解析・検証ツールを用いて、上記対象プログラム取得ステップにて取得した対象プログラムを解析、検証するツール実行ステップと、を含むことを特徴としている。

20

## 【 0 1 6 7 】

上記の構成によれば、対象プログラムを取得するとともに、当該対象プログラムの解析、検証に用いるプログラム解析・検証ツールを、ツール記憶部に格納された複数のプログラム解析・検証ツールから抽出し、取得する。そして、取得した各プログラム解析・検証ツールを用いて、対象プログラムを解析、検証する。

## 【 0 1 6 8 】

このように、対象プログラムを取得し、当該対象プログラムに合わせて解析、検証に用いるプログラム解析・検証ツールを、あらかじめ用意された複数のプログラム解析・検証ツールから選択できる。それゆえ、対象プログラムの解析、検証を希望する複数の利用者に対して、あらかじめ用意された複数のプログラム解析・検証ツールから、それぞれの対象プログラムに合わせて選択されたプログラム解析・検証ツールを用いて、解析、検証を実行するサービスを提供することができる。また、プログラム解析・検証ツールの提供を希望する複数のツール提供者に対しても、複数の利用者から対象プログラムの解析、検証の依頼を受け付け、解析、検証を実行するサービスを提供することができる。すなわち、上記の構成によれば、プログラム解析・検証ツールの利用のための共有システムを実現できる。

30

40

## 【 0 1 6 9 】

これにより、以下の効果を奏することができる。(1)複数の利用者でツール実行システムを共有するため、計算資源を個々の利用者が用意する必要がない。それゆえ、余剰の計算資源を融通し合うことができ、解析・検証のコストを抑えられるとともに、実行可能な解析・検証の種類や規模の制限が緩和される。(2)また、プログラム解析・検証ツールの導入・維持管理を、一括して行うことができるので、個々の利用者が導入・維持管理を行う必要がない。(3)また、統一的なユーザインターフェースを提供することが可能であるため、利用者が複数のプログラム解析・検証ツールの中から適切なものを選択することが容易になる。(4)一方、プログラム解析・検証ツールの提供者にとっては、利用者と同様に、計算資源を個々のツール提供者が用意する必要がない。それゆえ、余剰の計

50

算資源を融通し合うことができ、コストを抑えられるとともに、実行可能な解析・検証の種類や規模の制限が緩和される。また、様々な対象プログラムに対してもプログラム解析・検証ツールのテストができる。特に、これまでは計算資源の制限により困難であった対象プログラムに対してもプログラム解析・検証ツールのテストが可能となる。

【0170】

それゆえ、複数のプログラム解析・検証ツールから用いるべきものを適切に選択して、プログラムの解析・検証を容易に実行できるという効果を奏する。そして、プログラム解析・検証を共有システム上で行うことで、従来のプログラム解析・検証技術の実用化上の種々の問題点を解決できるという効果を奏する。

【0171】

さらに、本発明に係るプログラム解析・検証サービス提供システムは、上記対象プログラム取得手段が取得した上記対象プログラムの解析、検証に用いるプログラム解析・検証ツールを指定するツール指定情報を取得するツール指定情報取得手段をさらに備え、上記ツール記憶部は、複数のプログラム解析・検証ツールを各プログラム解析・検証ツールのツール関連情報とそれぞれ対応付けて格納しており、上記ツール抽出手段は、上記ツール指定情報取得手段が取得した上記ツール指定情報に基づいて、上記ツール関連情報を検索し、適合したツール関連情報と対応付けられたプログラム解析・検証ツールを抽出するものであることを特徴としている。

【0172】

上記の構成によれば、さらに、ツール記憶部は、複数のプログラム解析・検証ツールを各プログラム解析・検証ツールのツール関連情報とそれぞれ対応付けて格納している。そして、対象プログラムの解析、検証に用いるプログラム解析・検証ツールを指定するツール指定情報を取得し、このツール指定情報に基づいて、上記ツール関連情報を検索し、適合したツール関連情報と対応付けられたプログラム解析・検証ツールを抽出する。

【0173】

これにより、プログラム解析・検証ツールにはツール関連情報を対応付けておき、対象プログラムとともにツール指定情報を設定することにより、対象プログラムに合わせたプログラム解析・検証ツールを選択することが可能となる。ここで、ツール関連情報には、例えば、プログラム解析・検証ツールが扱うことができるプログラミング言語の種類やプログラムの種類（ユーザプログラム、システムプログラムの区別など）、プログラム解析・検証ツールが稼働する実行環境（オペレーティングシステムやライブラリの種類・バージョン、必要となる計算資源（CPUの速度や種類、メモリやディスクのサイズ等）など）が設定できる。一方、ツール指定情報は、解析・検証に用いるプログラム解析・検証ツールを指定する情報であり、ツール関連情報として設定されている内容の中から設定できる。

【0174】

よって、統一的なユーザインターフェースを提供することが可能となるため、利用者が複数のプログラム解析・検証ツールの中から適切なものを選択することが容易になるという効果を奏する。

【0175】

さらに、本発明に係るプログラム解析・検証サービス提供システムは、上記ツール記憶部は、上記プログラム解析・検証ツールとして、当該プログラム解析・検証ツールを導入、設定した仮想マシンを格納しており、上記ツール実行手段は、上記ツール取得手段が取得した仮想マシン上で、上記対象プログラム取得手段が取得した対象プログラムを解析、検証するものであることを特徴としている。

【0176】

上記の構成によれば、さらに、プログラム解析・検証ツールとして、当該プログラム解析・検証ツールを導入、設定した仮想マシンを用意し、この仮想マシン上で対象プログラムを解析、検証することができる。

【0177】

10

20

30

40

50

よって、以下の効果を奏することができる。まず、第1に、仮想マシンを用いて解析・検証を実行することで、複数の利用者および複数のプログラム解析・検証ツールの提供者の間で、システムの公平な利用を容易に実現することができる。また、第2に、仮想マシン上にプログラム解析・検証ツールを導入・設定することによって、プログラム解析・検証ツールの提供者は、普段自らが開発・テストしている実行環境とほぼ同様にプログラム解析・検証ツールを導入・設定できる。それゆえ、特殊な実行環境に対応させるためにプログラム解析・検証ツールを修正したり、また外部とのデータのやりとりをプログラム解析・検証ツールに行わせるための仕組みを別途用意したりといった、余計な負担をプログラム解析・検証ツールの提供者に掛けないようにすることができる。なお、仮想マシン上での対象プログラムの解析・検証は、仮想マシン上で対象プログラムを実行させて行ってもよいし、実行させずに行ってもよい。

10

【0178】

さらに、本発明に係るプログラム解析・検証サービス提供システムは、プログラム解析・検証ツールを導入、設定した仮想マシンを、プログラム解析・検証ツール毎に生成する仮想マシン生成手段をさらに備え、上記ツール記憶部は、上記プログラム解析・検証ツールとして、上記仮想マシン生成手段が生成した仮想マシンを格納していることを特徴としている。

【0179】

上記の構成によれば、さらに、新たなプログラム解析・検証ツールを受け付ける際、当該プログラム解析・検証ツールを導入、設定した仮想マシンを生成して、ツール記憶部に格納することができる。

20

【0180】

よって、プログラム解析・検証ツールを導入、設定した仮想マシンの生成を、プログラム解析・検証サービス提供システムで行うため、ツール提供者が仮想マシンの生成を行う必要がない。また、それぞれ異なるプログラム解析・検証ツールが導入、設定された複数の仮想マシンを、統一した様式で、特に仮想マシンの実行環境に配慮して作成することができるため、一様にて扱うことが可能となるという効果を奏する。

【0181】

また、本発明に係るプログラム解析・検証装置は、上記プログラム解析・検証サービス提供システムを構成するプログラム解析・検証装置であって、上記対象プログラム取得手段と、上記ツール取得手段と、上記ツール実行手段と、を備えて構成されていてもよい。

30

【0182】

また、本発明に係るプログラム解析・検証ツール管理装置は、上記プログラム解析・検証サービス提供システムを構成するプログラム解析・検証ツール管理装置であって、上記ツール記憶部と、上記ツール抽出手段と、を備えて構成されていてもよい。

【0183】

なお、上記のプログラム解析・検証サービス提供システム、プログラム解析・検証装置、プログラム解析・検証ツール管理装置は、コンピュータによって実現してもよく、この場合には、コンピュータを上記各手段として動作させることにより上記のプログラム解析・検証サービス提供システム、プログラム解析・検証装置、プログラム解析・検証ツール管理装置をコンピュータにて実現させる制御プログラム、およびそれを記録したコンピュータ読み取り可能な記録媒体も、本発明の範疇に入る。

40

【0184】

(7.補足)

最後に、プログラム解析・検証サービス提供システム1、解析・検証部100、対象プログラム管理部200、ツール管理部300の各ブロック、特に解析・検証制御部110、対象プログラム管理制御部210ツール管理制御部310は、ハードウェアロジックによって構成してもよいし、次のようにCPUを用いてソフトウェアによって実現してもよい。

【0185】

50

後者の場合、プログラム解析・検証サービス提供システム1（を構成する各装置）は、各機能を実現するプログラムの命令を実行するCPU（central processing unit）、上記プログラムを格納したROM（read only memory）、上記プログラムを展開するRAM（random access memory）、上記プログラムおよび各種データを格納するメモリ等の記憶装置（記録媒体）などを備えている。そして、本発明の目的は、上述した機能を実現するソフトウェアであるプログラム解析・検証サービス提供システム1（を構成する各装置）の制御プログラムのプログラムコード（実行形式プログラム、中間コードプログラム、ソースプログラム）をコンピュータで読み取り可能に記録した記録媒体を、上記プログラム解析・検証サービス提供システム1（を構成する各装置）に供給し、そのコンピュータ（またはCPUやMPU）が記録媒体に記録されているプログラムコードを読み出し実行することによっても、達成可能である。

10

【0186】

上記記録媒体としては、例えば、磁気テープやカセットテープ等のテープ系、フロッピー（登録商標）ディスク/ハードディスク等の磁気ディスクやCD-ROM/MO/MD/DVD/CD-R等の光ディスクを含むディスク系、ICカード（メモリカードを含む）/光カード等のカード系、あるいはマスクROM/EPROM/EEPROM（登録商標）/フラッシュROM等の半導体メモリ系などを用いることができる。

【0187】

また、プログラム解析・検証サービス提供システム1（を構成する各装置）を通信ネットワークと接続可能に構成し、上記プログラムコードを通信ネットワークを介して供給してもよい。この通信ネットワークとしては、特に限定されず、例えば、インターネット、イントラネット、エキストラネット、LAN、ISDN、VAN、CATV通信網、仮想専用網（virtual private network）、電話回線網、移動体通信網、衛星通信網等が利用可能である。また、通信ネットワークを構成する伝送媒体としては、特に限定されず、例えば、IEEE1394、USB、電力線搬送、ケーブルTV回線、電話線、ADSL回線等の有線でも、IrDAやリモコンのような赤外線、Bluetooth（登録商標）、802.11無線、HDR、携帯電話網、衛星回線、地上波デジタル網等の無線でも利用可能である。なお、本発明は、上記プログラムコードが電子的な伝送で具現化された、搬送波に埋め込まれたコンピュータデータ信号の形態でも実現され得る。

20

【0188】

本発明は上述した実施形態に限定されるものではなく、請求項に示した範囲で種々の変更が可能であり、実施形態に開示された技術的手段を適宜組み合わせ得られる実施形態についても本発明の技術的範囲に含まれる。

30

【産業上の利用可能性】

【0189】

本発明は、対象プログラムをプログラム解析・検証ツールを用いて解析、検証する装置（システム）に利用することができる。

【符号の説明】

【0190】

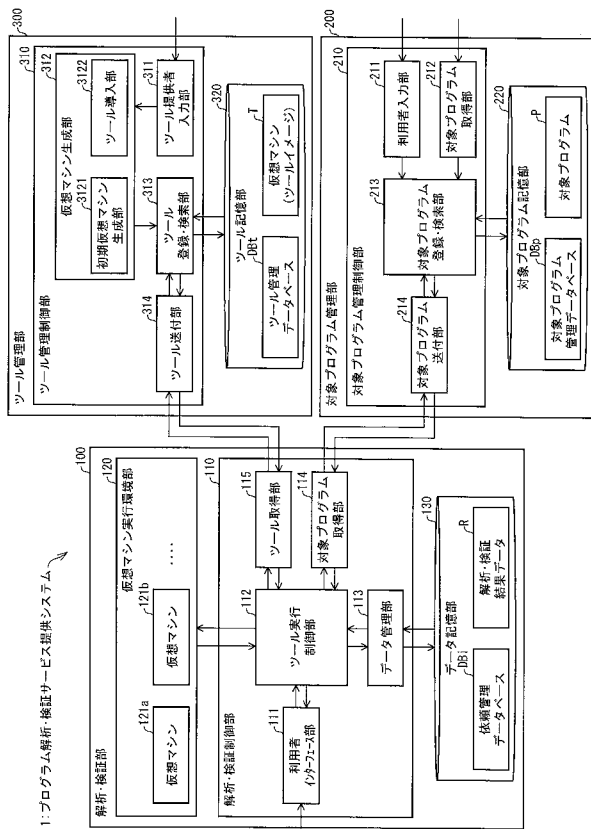
- 1 プログラム解析・検証サービス提供システム（データ処理サービス提供システム）
- 100 解析・検証部（プログラム解析・検証装置）
- 111 利用者インターフェース部（ツール指定情報取得手段）
- 114 対象プログラム取得部（対象プログラム取得手段）
- 115 ツール取得部（ツール取得手段）
- 120 仮想マシン実行環境部（ツール実行手段）
- 300 ツール管理部（プログラム解析・検証ツール管理装置）
- 312 仮想マシン生成部（仮想マシン生成手段）
- 313 ツール登録・検索部（ツール抽出手段）
- 320 ツール記憶部

40

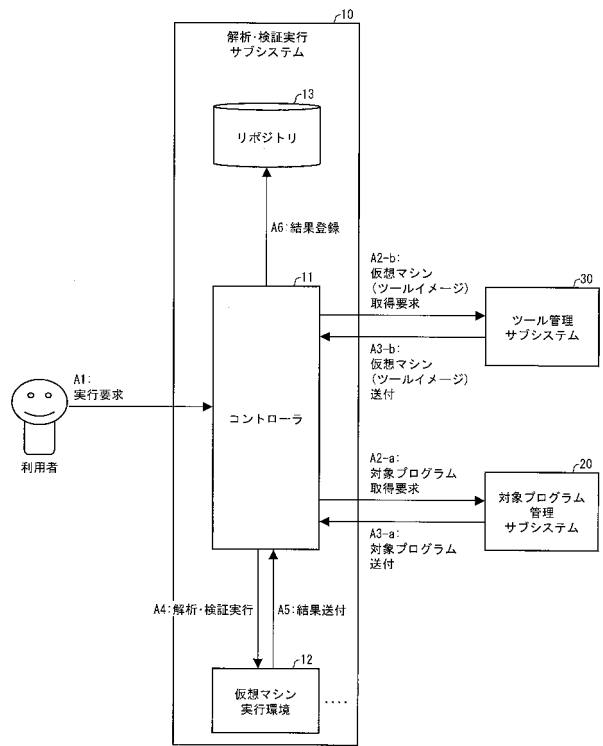
50

P 対象プログラム  
T 仮想マシン

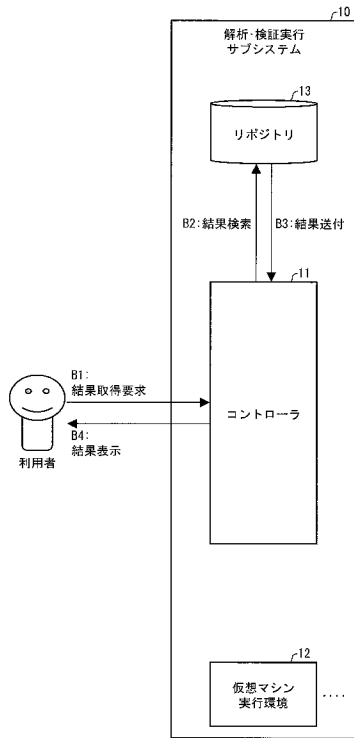
【図1】



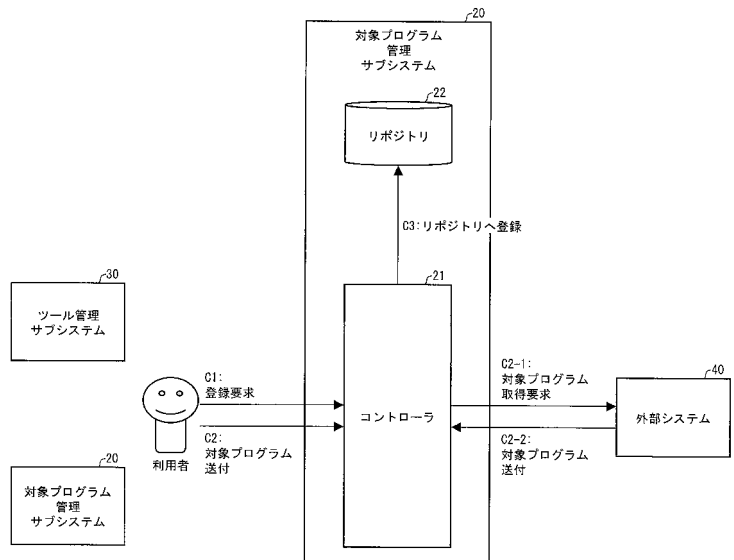
【図2】



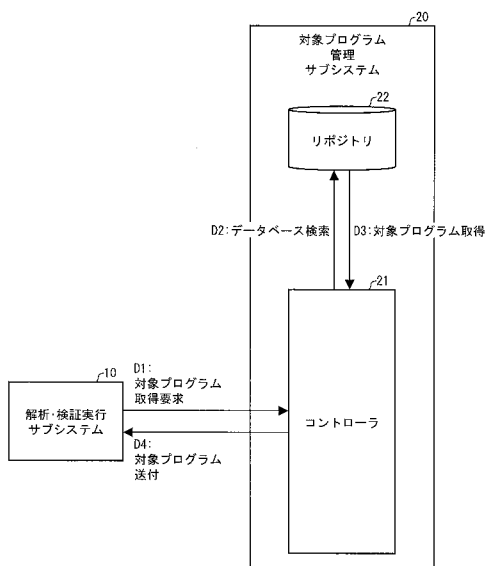
【図3】



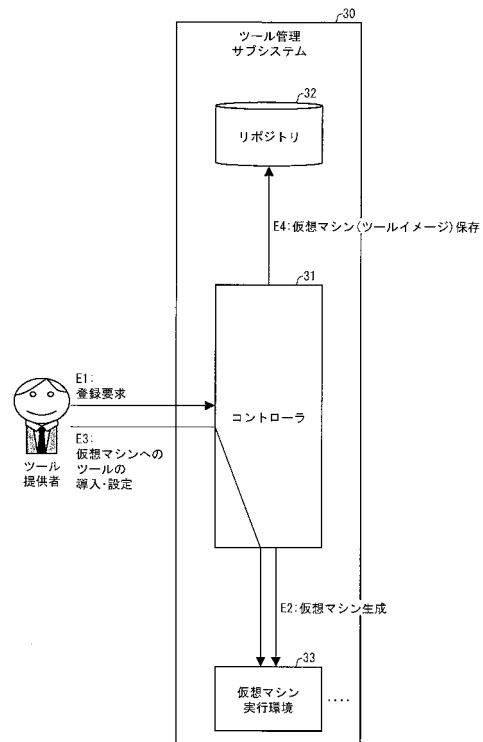
【図4】



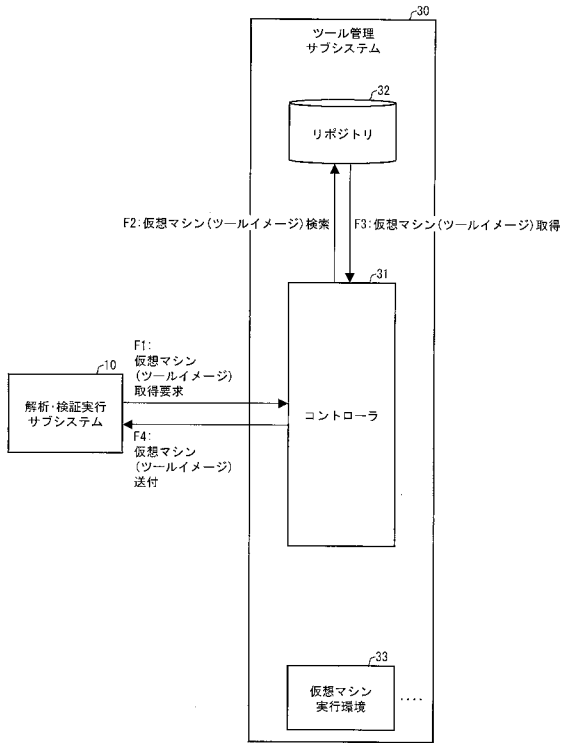
【図5】



【図6】



【図7】



【図8】

依頼管理データベースDBi

情報名	定義	記述例
利用者識別情報	利用者の識別情報	USER_A, USER_B, ...
利用者権限情報	利用者のアクセス権限	GROUP_AB, PUBLIC, ADMIN, ...
対象プログラム指定情報	解析・検証を行う対象である対象プログラムを指定する情報(対象プログラム識別情報と対象プログラム関連情報の和)	OBJP_ID_1, C, SYSTEM_PROG, ...
ツール指定情報	解析・検証に用いるプログラム解析・検証ツールを指定する情報(ツール識別情報とツール関連情報の和)	TOOL_ID_0, C, USER_PROG, SYSTEM_PROG, CPU_SPEED_2G, MEM_MAX_1G, ...
結果データ識別情報	解析・検証の結果データの識別情報	RESULT_0, RESULT_1, ...

【図9】

対象プログラム管理データベースDBp

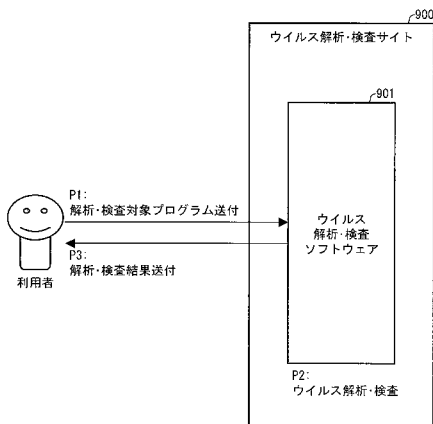
情報名	定義	記述例
利用者識別情報	利用者の識別情報	USER_A, USER_B, ...
利用者権限情報	利用者のアクセス権限	GROUP_AB, PUBLIC, ADMIN, ...
対象プログラム関連情報	利用者が任意の内容で任意の個数指定。例えば、対象プログラムで用いられているプログラミング言語の種類、プログラムの種類(ユーザプログラム、システムプログラム、システムプログラムの区別など)。	(プログラミング言語の種類) Java, C, C++, OCaml, ... (プログラムの種類の例) USER_PROG, SYSTEM_PROG, LIBRARY, ...
送付方法指定情報	対象プログラムの送付方法の指定。例えば、利用者からの直接送付、外部システムからのインポート。	DIRECT_UPLOAD, IMPORT_git://XXX, ...
対象プログラム識別情報	解析・検証を行う対象である対象プログラムの識別情報	OBJP_ID_0, OBJP_ID_1, ...

【図10】

ツール管理データベースDBt

情報名	定義	記述例
ツール提供者識別情報	ツール提供者の識別情報	USER_A, USER_B, ...
ツール関連情報	ツール提供者が任意の内容で任意の個数指定。例えば、プログラム解析・検証ツールが扱うことができるプログラミング言語の種類やプログラムの種類(ユーザプログラム、システムプログラムの区別など)、プログラム解析・検証ツールが稼働する実行環境(オペレーティングシステムやライブラリの種類・バージョン、必要となる計算資源(CPUの速度や種類、メモリやディスクのサイズなど)など。	(プログラミング言語の種類) Java, C, C++, OCaml, ... (プログラムの種類の例) USER_PROG, SYSTEM_PROG, LIBRARY, ... (実行環境・計算資源の例) OS_Linux_3.X, LIB_libgmpk_1.X, CPU_KIND_X86, CPU_SPEED_2G, CPU_NUM_8, MEM_MAX_1G, DISK_MAX_4G, ...
ツール識別情報	解析・検証に用いるプログラム解析・検証ツールが導入・設定されたツールイメージを含む仮想マシンの識別情報	TOOL_ID_0, TOOL_ID_1, ...

【図11】



---

フロントページの続き

- (56)参考文献 特開2007-025812(JP,A)  
特開2002-108778(JP,A)  
特開2010-211456(JP,A)  
特開平09-288592(JP,A)  
特開2010-198054(JP,A)

- (58)調査した分野(Int.Cl., DB名)  
G06F 21/56

- (54)【発明の名称】プログラム解析・検証サービス提供システム、その制御方法、制御プログラム、コンピュータを機能させるための制御プログラム、プログラム解析・検証装置、プログラム解析・検証ツール管理装置