

(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2019-159752  
(P2019-159752A)

(43) 公開日 令和1年9月19日(2019.9.19)

(51) Int.Cl.		F I			テーマコード (参考)
<b>G06F 21/12</b>	<b>(2013.01)</b>	G06F 21/12	310		5B376
<b>G06F 8/61</b>	<b>(2018.01)</b>	G06F 8/61			

審査請求 未請求 請求項の数 8 O L (全 28 頁)

(21) 出願番号	特願2018-45097 (P2018-45097)	(71) 出願人	000002945 オムロン株式会社 京都府京都市下京区塩小路通堀川東入南不動堂町801番地
(22) 出願日	平成30年3月13日 (2018.3.13)	(74) 代理人	110001195 特許業務法人深見特許事務所
		(72) 発明者	西山 佳秀 京都府京都市下京区塩小路通堀川東入南不動堂町801番地 オムロン株式会社内
		(72) 発明者	小島 達矢 京都府京都市下京区塩小路通堀川東入南不動堂町801番地 オムロン株式会社内
		(72) 発明者	川ノ上 真輔 京都府京都市下京区塩小路通堀川東入南不動堂町801番地 オムロン株式会社内 最終頁に続く

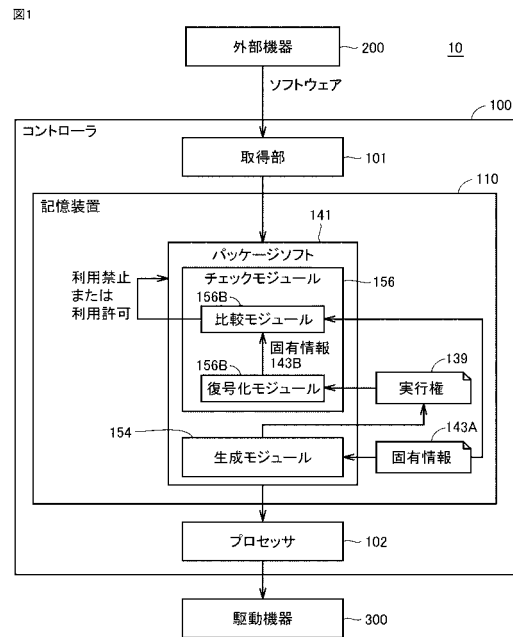
(54) 【発明の名称】 コントローラ、ライセンスの管理方法、およびライセンスの管理プログラム

(57) 【要約】

【課題】コントローラにダウンロードされるソフトウェアの不正利用を防止できる技術を提供する。

【解決手段】コントローラは、ソフトウェアと、当該ソフトウェアについての第1ライセンス情報とを外部機器から取得するための取得部と、ソフトウェアを実行するためのプロセッサとを備える。ソフトウェアは、コントローラまたは外部機器を一意に識別するための第1固有情報からソフトウェアの実行権を生成するための生成モジュールと、実行権の生成後の所定のタイミングにおいて実行権から第2固有情報を生成し、第2固有情報が第1固有情報と一致しないときに、ソフトウェアの利用を禁止するためのチェックモジュールとを含む。

【選択図】 図1



**【特許請求の範囲】****【請求項 1】**

制御対象を制御するためのコントローラであって、

前記コントローラで利用されるソフトウェアを外部機器から取得するための取得部と、  
前記ソフトウェアを実行するためのプロセッサとを備え、

前記ソフトウェアは、

前記コントローラまたは前記外部機器を一意に識別するための第 1 固有情報から前記ソフトウェアの実行権を生成するための生成モジュールと、

前記実行権の生成後の所定のタイミングにおいて前記実行権から第 2 固有情報を生成し、当該第 2 固有情報が前記第 1 固有情報と一致しないときに、前記ソフトウェアの利用を禁止するためのチェックモジュールとを含む、コントローラ。

10

**【請求項 2】**

前記外部機器は、前記コントローラと通信可能に構成されている通信機器を含み、

前記取得部は、前記コントローラと通信接続された前記通信機器から前記ソフトウェアをダウンロードする、請求項 1 に記載のコントローラ。

**【請求項 3】**

前記外部機器は、前記コントローラに電氣的に接続され得るメモリカードを含み、

前記取得部は、前記コントローラに電氣的に接続されたメモリカードから前記ソフトウェアを取得する、請求項 1 または 2 に記載のコントローラ。

20

**【請求項 4】**

前記プロセッサは、前記取得したソフトウェアを前記コントローラから前記外部機器に移すときに、前記実行権を削除する、請求項 1 ~ 3 のいずれか 1 項に記載のコントローラ。

**【請求項 5】**

前記プロセッサは、前記取得したソフトウェアを前記コントローラから前記外部機器に移すときに、前記生成モジュールおよび前記チェックモジュールを削除する、請求項 1 ~ 4 のいずれか 1 項に記載のコントローラ。

**【請求項 6】**

前記外部機器から取得され得るソフトウェアは、前記コントローラのファームウェア上で動作するプログラムと、当該ファームウェアに追加されるプログラムと、前記コントローラで利用され得るデータとの少なくとも 1 つを含む、請求項 1 ~ 5 のいずれか 1 項に記載のコントローラ。

30

**【請求項 7】**

制御対象を制御するコントローラで利用されるソフトウェアのライセンスの管理方法であって、

前記コントローラで利用されるソフトウェアと、当該ソフトウェアについての第 1 ライセンス情報とを外部機器から取得するステップと、

前記ソフトウェアを実行するステップとを備え、

前記実行するステップは、

前記コントローラまたは前記外部機器を一意に識別するための第 1 固有情報から前記ソフトウェアの実行権を生成するステップと、

前記実行権の生成後の所定のタイミングにおいて前記実行権から第 2 固有情報を生成し、当該第 2 固有情報が前記第 1 固有情報と一致しないときに、前記ソフトウェアの利用を禁止するステップとを含む、管理方法。

40

**【請求項 8】**

制御対象を制御するコントローラで利用されるソフトウェアのライセンスの管理プログラムであって、

前記管理プログラムは、前記コントローラに、

制御対象を制御するコントローラで利用されるソフトウェアを外部機器から取得するステップと、

50

前記ソフトウェアを実行するステップとを実行させ、  
前記実行するステップは、

前記コントローラまたは前記外部機器を一意に識別するための第1固有情報から前記ソフトウェアの実行権を生成するステップと、

前記実行権の生成後の所定のタイミングにおいて前記実行権から第2固有情報を生成し、当該第2固有情報が前記第1固有情報と一致しないときに、前記ソフトウェアの利用を禁止するステップとを含む、管理プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

10

本開示は、コントローラにダウンロードされたソフトウェアの利用を管理するための技術に関する。

【背景技術】

【0002】

様々な生産現場において、生産工程を自動化するFA (Factory Automation) システムが普及している。FAシステムは、種々の産業用の駆動機器によって構成される。産業用の駆動機器は、たとえば、ワークを移動するための移動テーブルや、ワークを搬送するためのコンベアや、予め定められた目的の場所までワーク移動するためのアームロボットなどを含む。これらの駆動機器は、PLC (Programmable Logic Controller) やロボットコントローラなどの産業用のコントローラによって制御される。

20

【0003】

近年、外部機器からソフトウェアをインストールすることができるコントローラが普及している。このようなコントローラが、国際公開第2015/136965号(特許文献1)、国際公開第2015/136972号(特許文献2)、国際公開第2015/136966号(特許文献3)、および、国際公開第2015/136964号(特許文献4)に開示されている。

【先行技術文献】

【特許文献】

【0004】

30

【特許文献1】国際公開第2015/136965号

【特許文献2】国際公開第2015/136972号

【特許文献3】国際公開第2015/136966号

【特許文献4】国際公開第2015/136964号

【発明の概要】

【発明が解決しようとする課題】

【0005】

近年、種々のソフトウェアが開発されており、ソフトウェアのライセンスの管理が複雑化となっている。ライセンスが適切に管理されていない場合、購入されたソフトウェアが不正にコピーされて、ソフトウェアがライセンスを有さない他のコントローラで不正に利用される可能性がある。したがって、このような不正利用を防止する仕組みが望まれている。

40

【0006】

本開示は上述のような問題点を解決するためになされたものであって、ある局面における目的は、コントローラにダウンロードされるソフトウェアの不正利用を防止できる仕組みを提供することである。

【課題を解決するための手段】

【0007】

本開示の一例では、上記コントローラで利用されるソフトウェアを外部機器から取得するための取得部と、上記ソフトウェアを実行するためのプロセッサとを備える。上記ソフトウェアは、上記コントローラまたは上記外部機器を一意に識別するための第1固有情報

50

から上記ソフトウェアの実行権を生成するための生成モジュールと、上記実行権の生成後の所定のタイミングにおいて上記実行権から第2固有情報を生成し、当該第2固有情報が上記第1固有情報と一致しないときに、上記ソフトウェアの利用を禁止するためのチェックモジュールとを含む。

【0008】

この開示によれば、ライセンス情報に基づいて、取得したソフトウェアの利用が禁止されることで、当該ソフトウェアの不正利用が防止される。

【0009】

本開示の一例では、上記コントローラと通信可能に構成されている通信機器を含む。上記取得部は、上記コントローラと通信接続された上記通信機器から上記ソフトウェアをダウンロードする。

10

【0010】

この開示によれば、通信機器からダウンロードしたソフトウェアの不正利用が防止される。

【0011】

本開示の一例では、上記外部機器は、上記コントローラに電氣的に接続され得るメモリカードを含む。上記取得部は、上記コントローラに電氣的に接続されたメモリカードから上記ソフトウェアを取得する。

【0012】

この開示によれば、メモリカードから取得したソフトウェアの不正利用が防止される。

20

本開示の一例では、上記プロセッサは、上記取得したソフトウェアを上記コントローラから上記外部機器に移すときに、上記実行権を削除する。

【0013】

この開示によれば、ソフトウェアのライセンス情報が複数存在することがなくなり、ソフトウェアの不正利用がより確実に防止される。

【0014】

本開示の一例では、上記プロセッサは、上記取得したソフトウェアを上記コントローラから上記外部機器に移すときに、上記生成モジュールおよび上記チェックモジュールを削除する。

【0015】

この開示によれば、生成モジュールおよびチェックモジュールが削除されることで、コントローラの資源が有効に利用され得る。

30

【0016】

本開示の一例では、上記外部機器から取得され得るソフトウェアは、上記コントローラのファームウェア上で動作するプログラムと、当該ファームウェアに追加されるプログラムと、上記コントローラで利用され得るデータとの少なくとも1つを含む。

【0017】

この開示によれば、不正利用の防止がファームウェア上で動作するプログラムと、当該ファームウェアに追加されるプログラムとの不正利用が防止される。

【0018】

本開示の他の例では、制御対象を制御するコントローラで利用されるソフトウェアのライセンスの管理方法は、上記コントローラで利用されるソフトウェアと、当該ソフトウェアについての第1ライセンス情報とを外部機器から取得するステップと、上記ソフトウェアを実行するステップとを備える。上記実行するステップは、上記コントローラまたは上記外部機器を一意に識別するための第1固有情報から上記ソフトウェアの実行権を生成するステップと、上記実行権の生成後の所定のタイミングにおいて上記実行権から第2固有情報を生成し、当該第2固有情報が上記第1固有情報と一致しないときに、上記ソフトウェアの利用を禁止するステップとを含む。

40

【0019】

この開示によれば、ライセンス情報に基づいて、取得したソフトウェアの利用が禁止さ

50

れることで、当該ソフトウェアの不正利用が防止される。

【0020】

本開示の他の例では、制御対象を制御するコントローラで利用されるソフトウェアのライセンスの管理プログラムは、上記管理プログラムは、上記コントローラに、制御対象を制御するコントローラで利用されるソフトウェアを外部機器から取得するステップと、上記ソフトウェアを実行するステップとを実行させる。上記実行するステップは、上記コントローラまたは上記外部機器を一意に識別するための第1固有情報から上記ソフトウェアの実行権を生成するステップと、上記実行権の生成後の所定のタイミングにおいて上記実行権から第2固有情報を生成し、当該第2固有情報が上記第1固有情報と一致しないときに、上記ソフトウェアの利用を禁止するステップとを含む。

10

【0021】

この開示によれば、ライセンス情報に基づいて、取得したソフトウェアの利用が禁止されることで、当該ソフトウェアの不正利用が防止される。

【発明の効果】

【0022】

ある局面において、コントローラにダウンロードされるソフトウェアの不正利用を防止できる。

【図面の簡単な説明】

【0023】

【図1】実施の形態に従うFAシステムの概略を示す図である。

20

【図2】実施の形態に従うFAシステムの装置構成を概略的に示す模式図である。

【図3】パッケージソフトをダウンロード販売する仕組みを示す図である。

【図4】コントローラと開発支援装置とサーバとの間のデータの流れを示すシーケンス図である。

【図5】ソフトウェア管理テーブルのデータ構造を示す図である。

【図6】実施の形態に従うコントローラのハードウェア構成の一例を示す模式図である。

【図7】実施の形態に従う外部機器のハードウェア構成を示す模式図である。

【図8】パッケージソフトを開発支援装置からダウンロードする処理を表わすフローチャートである。

【図9】パッケージソフトの利用を有効化する処理を表わすフローチャートである。

30

【図10】実行権のチェック処理を表わすフローチャートである。

【図11】ダウンロードしたパッケージソフトの実行処理を表わすフローチャートである。

【図12】パッケージソフトをメモリカードで販売する仕組みを示す図である。

【図13】パッケージソフトをメモリカードからダウンロードする処理を表わすフローチャートである。

【図14】パッケージソフトのディプロイ処理の流れを概略的に示す図である。

【図15】図14に示されるディプロイ処理の続きを示す図である。

【図16】コントローラとメモリカードとの間でパッケージソフトを移している様子を示す図である。

40

【図17】パッケージソフトをBTO(Build To Order)販売する仕組みを示す図である。

【発明を実施するための形態】

【0024】

以下、図面を参照しつつ、本発明に従う各実施の形態について説明する。以下の説明では、同一の部品および構成要素には同一の符号を付してある。それらの名称および機能も同じである。したがって、これらについての詳細な説明は繰り返さない。

【0025】

< A . 適用例 >

図1を参照して、本発明の適用例について説明する。図1は、FAシステム10の概略

50

を示す図である。

【0026】

FAシステム10は、設備および装置などの制御対象を制御し、生産工程を自動化するためのシステムである。FAシステム10は、1つ以上のコントローラ100と、1つ以上の外部機器200と、1つ以上の駆動機器300とで構成されている。

【0027】

コントローラ100は、ハードウェア構成として、取得部101と、プロセッサ102と、記憶装置110とを含む。記憶装置110は、外部機器200からダウンロードしたパッケージソフト141と、コントローラ100の固有情報143Aとを格納している。

【0028】

外部機器200は、コントローラ100で実行可能な種々のソフトウェアを提供する。このような外部機器200としては、後述の開発支援装置200A(図2参照)、後述のサーバ200B(図2参照)、後述のメモリカード200C(図2参照)などが挙げられる。

【0029】

取得部101は、外部機器200からソフトウェアを取得する機能を有する任意のハードウェアである。一例として、取得部101は、外部機器200と通信するための通信インターフェイスや、SDカードなどのメモリカードが接続されるメモリカードインターフェイスなどを含む。

【0030】

図1の例では、外部機器200からダウンロードされたソフトウェアがパッケージソフト141として示されている。パッケージソフト141は、種々の機能がパッケージ化された販売対象のプログラムの単位を示す。パッケージソフト141の一例として、コントローラ100への追加機能やデータなどが挙げられる。

【0031】

プロセッサ102は、コントローラ100の各種プログラムを実行することで、制御対象である駆動機器300を制御する。一例として、プロセッサ102によって実行されるプログラムは、ダウンロードされたパッケージソフト141などの各種プログラムを含む。

【0032】

ダウンロードされたパッケージソフト141は、生成モジュール154と、チェックモジュール156と、その他の機能モジュールとを含む。当該その他の機能モジュールがユーザによって利用され得る機能モジュールであり、生成モジュール154およびチェックモジュール156は、当該機能モジュールに付随するものである。

【0033】

生成モジュール154は、コントローラ100を一意に識別するための固有情報143A(第1固有情報)からパッケージソフト141の実行権139を生成する。実行権139は、パッケージソフト141の実行を行うための権限に相当する。すなわち、実行権139を保持しているコントローラ100がパッケージソフト141を実行することができる。固有情報143Aは、たとえば、コントローラ100またはプロセッサ102の、形式、生産情報(製造番号など)、名前、および、コントローラ100を一意に識別するためのその他の情報を含む。

【0034】

実行権139の生成方法は、任意である。一例として、生成モジュール154は、予め定められた暗号化方式で固有情報143Aを暗号化することで実行権139を生成する。この場合、生成モジュール154は、所定の暗号化キーを用いて実行権139を生成する。あるいは、生成モジュール154は、可逆計算が可能な予め定められた算出式に固有情報143Aを代入して得られた結果を実行権139として生成してもよい。生成された実行権139は、記憶装置110に格納される。

【0035】

10

20

30

40

50

チェックモジュール 156 は、コントローラ 100 の起動時やパッケージソフト 141 の実行時など実行権 139 の生成後の所定のタイミングにおいて、実行権 139 から固有情報（第 2 固有情報）を生成する。以下では、説明の便宜のために、コントローラ 100 に予め保持されている固有情報を「固有情報 143 A」と称し、実行権 139 の生成後に実行権 139 から生成された固有情報を「固有情報 143 B」と称する。チェックモジュール 156 は、固有情報 143 B が本来の固有情報 143 A と一致しないときに、パッケージソフト 141 の利用を禁止する。

【0036】

より具体的には、チェックモジュール 156 は、復号化モジュール 156 A と、比較モジュール 156 B とを含む。復号化モジュール 156 A は、所定の復号化キーを利用する  
10

【0037】

比較モジュール 156 B は、コントローラ 100 の起動時またはパッケージソフト 141 の利用時に生成された固有情報 143 B を本来の固有情報 143 A と比較する。固有情報 143 B が固有情報 143 A と一致する場合、比較モジュール 156 B は、ダウンロードしたパッケージソフト 141 の利用を許可する。一方で、固有情報 143 B が固有情報 143 A と一致しない場合、比較モジュール 156 B は、ダウンロードしたパッケージソフト 141 の利用を禁止する。

【0038】

ここでいう「一致する」とは、固有情報 143 B の全てが固有情報 143 A の全てと一致する場合だけでなく、固有情報 143 B の所定箇所が固有情報 143 A の所定箇所と一致する場合も含み得る。「一致しない」とは、「一致する」との条件が満たされない場合を指す。  
20

【0039】

以上のように、本来の固有情報 143 A と、パッケージソフト 141 の実行時などに生成される固有情報 143 B との比較結果に基づいて、パッケージソフト 141 の利用の許可/禁止が切り替えられる。これにより、たとえば、パッケージソフト 141 がダウンロード時とは異なるコントローラに不正に移動された場合に、パッケージソフト 141 の利用を防止することができる。また、ライセンスをチェックする仕組みがパッケージソフト 141 に組み込まれていることで、ライセンスのチェック機能をコントローラ 100 に予め搭載する必要がなくなる。  
30

【0040】

< B . F A システム 10 の装置構成 >

図 2 を参照して、F A システム 10 の装置構成の一例について説明する。図 2 は、F A システム 10 の装置構成を概略的に示す模式図である。

【0041】

F A システム 10 は、1 つ以上のコントローラ 100 と、1 つ以上の外部機器 200 とを含む。図 2 の例では、F A システム 10 は、2 つのコントローラ 100 A , 100 B で構成されている。コントローラ 100 に接続され得る外部機器 200 の例として、開発支援装置 200 A と、サーバ 200 B と、メモリカード 200 C とが示されている。  
40

【0042】

以下では、コントローラ 100 A , 100 B を総称してコントローラ 100 ともいう。コントローラ 100 は、複数の物理的な通信ポートを有する。それぞれの通信ポートには異なるネットワークが接続され得る。図 2 の例では、コントローラ 100 は、2 つの通信ポート P 1 , P 2 を有する。通信ポート P 1 には、ネットワーク NW 1 が接続されている。通信ポート P 2 には、ネットワーク NW 2 が接続されている。

【0043】

ネットワーク NW 1 には、たとえば、E t h e r N e t / I P（登録商標）や E t h e r N e t（登録商標）などが採用される。一例として、ネットワーク NW 1 には、開発支援装置 200 A やサーバ 200 B など、コントローラ 100 との通信機能を有する各種通  
50

信機器が接続され得る。

【0044】

開発支援装置200Aは、たとえば、PC(Personal Computer)、タブレット端末、または、スマートフォンなどである。開発支援装置200Aには、プログラム開発ツールがインストールされ得る。プログラム開発ツールは、コントローラ100用の制御プログラムの開発を支援するためのアプリケーションである。一例として、プログラム開発ツールは、オムロン社製の「Sysmac Studio」である。ユーザは、プログラム開発ツール上でコントローラ100用の制御プログラムを設計し、設計した制御プログラムをコントローラ100にインストールすることができる。作成された制御プログラムは、開発支援装置200Aによってコンパイルされた実行形式のファイルとしてコントローラ100に送られる。

10

【0045】

サーバ200Bは、コントローラ100で実行可能な各種のパッケージソフト141を提供する装置である。提供されるパッケージソフト141は、たとえば、外部データベースに接続する機能、NC(Numerical Control)制御に関する各種機能、ロボット制御に関する各種機能、AI(Artificial Intelligence)に関する各種機能、コントローラ100で利用され得る各種データ(たとえば、AIの学習処理によって得られた各種データ)、および、特定の駆動機器300を制御するための各種機能(たとえば、振動制御機能)などを含む。なお、パッケージソフト141の提供元は、サーバ200Bに限られず、任意の外部機器200から提供される。一例として、パッケージソフト141は、メモリカード200Cから提供されてもよい。

20

【0046】

ネットワークNW2には、データの到達時間が保証される、定周期通信を行うフィールドネットワークを採用することが好ましい。このような定周期通信を行うフィールドネットワークとしては、EtherCAT(登録商標)、Component(登録商標)などが知られている。コントローラ100は、開発支援装置200A上で作成された制御プログラムに従って制御対象の駆動機器300を制御する。

【0047】

駆動機器300は、直接的または間接的にワークに対して所定の作業を行うための機器の集合である。図2の例では、駆動機器300は、ロボットコントローラ300Aや、サーボドライバ300Bや、ロボットコントローラ300Aに制御されるアームロボット301Aや、サーボドライバ300Bによって制御されるサーボモータ301Bなどを含む。また、駆動機器300は、ワークを撮影するための視覚センサや、生産工程で利用されるその他の機器などを含んでもよい。

30

【0048】

<C. パッケージソフト141の販売形態>

図3を参照して、パッケージソフト141の販売形態の一例について説明する。図3は、パッケージソフト141をダウンロード販売する仕組みを示す図である。

【0049】

図3の例では、パッケージソフト141の提供元のサーバ200Bと、パッケージソフト141の提供先の開発支援装置200Aとが示されている。一例として、サーバ200Bは、販売元の会社「A」に設置されている。開発支援装置200Aは、パッケージソフト141の販売先の会社「B」に設置されている。

40

【0050】

ステップ(1)において、開発支援装置200Aのユーザは、サーバ200Bのウェブサイトなどにアクセスし、販売されているパッケージソフトの中からダウンロード対象のパッケージソフトを選択する。販売されているパッケージソフトは、コントローラ100のファームウェア上で動作するプログラムや、ファームウェアに追加されるファームウェアプログラムなどを含む。ファームウェア上で動作するパッケージソフトは、ファームウェアから提供される基本機能に依存して動作する。

50

## 【 0 0 5 1 】

開発支援装置 2 0 0 A のユーザは、購入対象のパッケージソフト 1 4 1 の選択後、購入手続きを行う。購入手続きが完了すると、サーバ 2 0 0 B は、購入したパッケージソフト 1 4 1 を開発支援装置 2 0 0 A にインストールする。このとき、ステップ ( 2 ) において、サーバ 2 0 0 B は、パッケージソフト 1 4 1 の利用を許可するためのライセンス権 1 4 2 を開発支援装置 2 0 0 A に送信する。

## 【 0 0 5 2 】

ステップ ( 3 ) において、開発支援装置 2 0 0 A は、サーバ 2 0 0 B からインストールしたパッケージソフト 1 4 1 とライセンス権 1 4 2 とをコントローラ 1 0 0 にインストールする。

10

## 【 0 0 5 3 】

ステップ ( 4 ) において、開発支援装置 2 0 0 A は、パッケージソフト 1 4 1 のライセンス権を有効にするための有効化操作を受け付けたとする。このことに基づいて、開発支援装置 2 0 0 A は、有効化命令をコントローラ 1 0 0 に送信する。コントローラ 1 0 0 は、有効化命令を受信したことに基づいて、コントローラ 1 0 0 の固有情報 1 4 3 A から実行権 1 3 9 を生成する。

## 【 0 0 5 4 】

ステップ ( 5 ) において、開発支援装置 2 0 0 A は、パッケージソフト 1 4 1 の有効化処理が正常に完了したことに基づいて、パッケージソフト 1 4 1 の利用権を行使したことをサーバ 2 0 0 B に送信する。このとき、利用権を行使した利用者の情報がサーバ 2 0 0 B に送信され、当該情報がサーバ 2 0 0 B に登録される。

20

## 【 0 0 5 5 】

< D . データフロー >

図 4 および図 5 を参照して、上述の図 3 に示されるダウンロード販売を実現するための制御フローについて説明する。図 4 は、コントローラ 1 0 0 と開発支援装置 2 0 0 A とサーバ 2 0 0 B との間のデータの流れを示すシーケンス図である。

## 【 0 0 5 6 】

ステップ S 1 0 において、開発支援装置 2 0 0 A のユーザは、サーバ 2 0 0 B のウェブサイトにアクセスし、パッケージソフト 1 4 1 の販売操作を行ったとする。このとき、ユーザは、パッケージソフト 1 4 1 を購入するために各種の情報 ( 以下、「購入情報」ともいう。 ) を入力する。当該購入情報は、購入対象のパッケージソフト 1 4 1 を特定するための情報、購入者に関する各種情報 ( たとえば、名前や住所など ) や、ダウンロード対象のコントローラ 1 0 0 に関する各種情報を含む。入力された購入情報は、サーバ 2 0 0 B に送信される。

30

## 【 0 0 5 7 】

ステップ S 1 4 において、サーバ 2 0 0 B は、開発支援装置 2 0 0 A から受信した購入情報に基づいて、パッケージソフト 1 4 1 のライセンス権 1 4 2 を発行する。発行されたライセンス権 1 4 2 は、購入対象のパッケージソフト 1 4 1 とともに、開発支援装置 2 0 0 A に送信される。

## 【 0 0 5 8 】

ステップ S 2 0 において、開発支援装置 2 0 0 A は、サーバ 2 0 0 B からダウンロードしたパッケージソフト 1 4 1 のダウンロード操作を受け付けたとする。このことに基づいて、開発支援装置 2 0 0 A は、パッケージソフト 1 4 1 とライセンス権 1 4 2 とを指定されたコントローラ 1 0 0 に送信する。

40

## 【 0 0 5 9 】

ステップ S 2 6 において、コントローラ 1 0 0 は、パッケージソフト 1 4 1 を内部の記憶装置 1 1 0 ( 図 1 参照 ) に配置するディプロイ処理を実行する。このとき、コントローラ 1 0 0 は、パッケージソフトを内部の記憶装置に配置する管理するためのソフトウェア管理テーブルを更新する。図 5 は、ソフトウェア管理テーブル 1 5 5 のデータ構造を示す図である。

50

## 【 0 0 6 0 】

ソフトウェア管理テーブル 1 5 5 は、コントローラ 1 0 0 にダウンロードされたパッケージソフトを管理するためのデータベースである。ソフトウェア管理テーブル 1 5 5 は、パッケージソフトの一覧テーブル 1 6 0 と、パッケージソフトの詳細情報 1 6 1 ~ 1 6 3 とを含む。

## 【 0 0 6 1 】

パッケージソフトの一覧テーブル 1 6 0 は、パッケージソフトの略称を規定するカラム 1 6 0 A と、パッケージソフトのバージョンを規定するカラム 1 6 0 B と、パッケージソフトの分類を規定するカラム 1 6 0 C と、パッケージソフトのアドオンタイプを規定するカラム 1 6 0 D と、パッケージソフトの提供形態を規定するカラム 1 6 0 E と、パッケージソフトのフルネームを規定するカラム 1 6 0 F と、パッケージソフトの提供元を規定するカラム 1 6 0 F とを含む。

10

## 【 0 0 6 2 】

パッケージソフトの一覧テーブル 1 6 0 の各レコードは、パッケージソフトの詳細情報に関連付けられている。一例として、一覧テーブル 1 6 0 のカラム 1 6 0 A の情報をキーとして、各パッケージソフト 1 4 1 の詳細情報が関連付けられる。図 5 の例では、「E C A T」として示されるパッケージ名には、詳細情報 1 6 1 が関連付けられている。「A I」として示されるパッケージ名には、詳細情報 1 6 2 , 1 6 3 とが関連付けられている。

## 【 0 0 6 3 】

詳細情報 1 6 1 は、パッケージソフトの略称を規定するカラム 1 6 1 A と、対応言語の種類を示すカラム 1 6 1 B と、パッケージソフトのフルネームを規定するカラム 1 6 1 C と、パッケージソフトの提供元を規定するカラム 1 6 1 D とを含む。

20

## 【 0 0 6 4 】

詳細情報 1 6 2 は、パッケージソフトの略称を規定するカラム 1 6 2 A と、対応言語の種類を示すカラム 1 6 2 B と、パッケージソフトのフルネームを規定するカラム 1 6 2 C と、パッケージソフトの提供元を規定するカラム 1 6 2 D とを含む。

## 【 0 0 6 5 】

詳細情報 1 6 3 は、パッケージソフトに含まれる各種メソッド（処理）が規定されている。詳細情報 1 6 3 は、メソッドの名前を示すカラム 1 6 3 A と、パッケージソフトの呼出しのタイプを示すカラム 1 6 3 B と、パッケージソフトのディプロイメソッドを規定しているカラム 1 6 3 C と、パッケージソフトのロールバックメソッドを規定しているカラム 1 6 3 D とを含む。

30

## 【 0 0 6 6 】

再び図 4 を参照して、ステップ S 2 8 において、コントローラ 1 0 0 は、上述の生成モジュール 1 5 4（図 1 参照）として、自身の固有情報 1 4 3 A から実行権 1 3 9 を生成する。生成モジュール 1 5 4 は、予め定められた暗号化方式で固有情報 1 4 3 A から実行権 1 3 9 を生成する。一例として、生成モジュール 1 5 4 は、所定の暗号化キーを用いて実行権 1 3 9 を生成する。生成された実行権 1 3 9 は、コントローラ 1 0 0 の記憶装置 1 1 0 に格納される。

## 【 0 0 6 7 】

ステップ S 3 0 において、コントローラ 1 0 0 は、所定のタイミングが到来したか否かを判断する。当該所定のタイミングは、たとえば、コントローラ 1 0 0 の起動時や、パッケージソフト 1 4 1 の実行時や、実行権 1 3 9 の生成後の予め定められた任意のタイミングを含む。コントローラ 1 0 0 は、当該所定のタイミングが到来したと判断した場合（ステップ S 3 0 において Y E S）、制御をステップ S 3 2 に切り替える。そうでない場合には（ステップ S 3 0 において N O）、コントローラ 1 0 0 は、ステップ S 3 0 の処理を再び実行する。

40

## 【 0 0 6 8 】

ステップ S 3 2 において、コントローラ 1 0 0 は、上述のチェックモジュール 1 5 6（図 1 参照）として、ステップ S 2 8 で生成した実行権 1 3 9 から固有情報 1 4 3 B を生成

50

する。一例として、コントローラ 100 は、ステップ S 28 で固有情報 143 A から実行権 139 を生成するために用いられた暗号化方式に対応する復号化方式で実行権 139 を復号化し、固有情報 143 B を生成する。

【0069】

ステップ S 40 において、コントローラ 100 は、上述のチェックモジュール 156 (図 1 参照) として、ステップ S 32 で生成された固有情報 143 B が本来の固有情報 143 A と一致するか否かを判断する。コントローラ 100 は、生成された固有情報 143 B が本来の固有情報 143 A と一致すると判断した場合 (ステップ S 40 において YES)、制御をステップ S 42 に切り替える。そうでない場合には (ステップ S 40 において NO)、コントローラ 100 は、制御をステップ S 44 に切り替える。

10

【0070】

ステップ S 42 において、コントローラ 100 は、ダウンロードしたパッケージソフト 141 の利用を許可する。パッケージソフト 141 の利用を許可する手段は、任意である。一例として、利用の禁止 / 許可を示すフラグがコントローラ 100 で管理されており、コントローラ 100 は、許可を示すように当該フラグを書き換える。

【0071】

ステップ S 44 において、コントローラ 100 は、ダウンロードしたパッケージソフト 141 の利用を禁止する。パッケージソフト 141 の利用を禁止する手段は、任意である。一例として、利用の禁止 / 許可を示すフラグがコントローラ 100 で管理されており、コントローラ 100 は、禁止を示すように当該フラグを書き換える。

20

【0072】

< E . ハードウェア構成 >

図 6 および図 7 を参照して、コントローラ 100 および外部機器 200 のハードウェア構成について順に説明する。

【0073】

( E 1 . コントローラ 100 のハードウェア構成 )

まず、図 6 を参照して、コントローラ 100 のハードウェア構成について説明する。図 6 は、コントローラ 100 のハードウェア構成の一例を示す模式図である。

【0074】

コントローラ 100 は、通信インターフェイス 101 A と、メモ리카ードインターフェイス 101 B と、CPU (Central Processing Unit) や MPU (Micro-Processing Unit) などのプロセッサ 102 と、チップセット 104 と、主メモリ 106 と、不揮発性の記憶装置 110 と、内部バスコントローラ 122 と、フィールドバスコントローラ 124 とを含む。

30

【0075】

プロセッサ 102 は、記憶装置 110 に格納された制御プログラム 130 を読み出して、主メモリ 106 に展開して実行することで、制御対象の駆動機器 300 などに対する任意の制御を実現する。制御プログラム 130 は、コントローラ 100 を制御するための各種プログラムを含む。一例として、制御プログラム 130 は、ファームウェアとしてのシステムプログラム 140、パッケージソフト 141、ユーザプログラム 145 などを含む。システムプログラム 140 は、データの入出力処理や実行タイミング制御などの、コントローラ 100 の基本機能を提供するための命令コードを含む。ユーザプログラム 145 は、外部機器 200 からダウンロードされたものである。ユーザプログラム 145 は、制御対象に応じて外部機器 200 上で任意に設計され、シーケンス制御を実行するためのシーケンスプログラム 145 A およびモーション制御を実行するためのモーションプログラム 145 B とを含む。

40

【0076】

チップセット 104 は、各コンポーネントを制御することで、コントローラ 100 全体としての処理を実現する。

【0077】

50

記憶装置 110 は、制御プログラム 130 の他にも種々のデータを格納する。一例として、記憶装置 110 は、パッケージソフト 141 のライセンス権 142 と、コントローラ 100 を一意に識別するための固有情報 143 A と、上述のソフトウェア管理テーブル 155 (図 5 参照)などを格納する。ライセンス権 142 は、パッケージソフト 141 の実行権 139 を含む。実行権 139 は、たとえば、たとえば、パッケージソフト 141 のダウンロード時(配置時)など予め定められたタイミングで生成される。

【0078】

内部バスコントローラ 122 は、コントローラ 100 と内部バスを通じて連結される各種デバイスとデータを遣り取りするインターフェイスである。このようなデバイスの一例として、I/O ユニット 126 が接続されている。

10

【0079】

フィールドバスコントローラ 124 は、コントローラ 100 とフィールドバスを通じて連結される各種の駆動機器 300 とデータを遣り取りするインターフェイスである。このようなデバイスの一例として、ロボットコントローラ 300 A やサーボドライバ 300 B が接続されている。他にも、視覚センサなどの駆動機器が接続されてもよい。

【0080】

内部バスコントローラ 122 およびフィールドバスコントローラ 124 は、接続されているデバイスに対して任意の指令を与えることができるとともに、デバイスが管理している任意のデータを取得することができる。また、内部バスコントローラ 122 および/またはフィールドバスコントローラ 124 は、ロボットコントローラ 300 A やサーボドライバ 300 B との間でデータを遣り取りするためのインターフェイスとしても機能する。

20

【0081】

通信インターフェイス 101 A は、上述の取得部 101 (図 1 参照)の一例である。通信インターフェイス 101 A は、各種の有線/無線ネットワークを通じたデータの遣り取りを制御する。コントローラ 100 は、通信インターフェイス 101 A を介して、開発支援装置 200 A やサーバ 200 B などの外部機器 200 と通信を行う。コントローラ 100 は、通信インターフェイス 101 A を介して外部機器 200 からパッケージソフト 141などをダウンロードすることができる。

【0082】

メモリカードインターフェイス 101 B は、上述の取得部 101 (図 1 参照)の一例である。メモリカードインターフェイス 101 B は、外部記憶媒体の一例であるメモリカード 200 C (たとえば、SD カード)を電氣的に接続するためのインターフェイスである。メモリカードインターフェイス 101 B は、メモリカード 200 C を着脱可能に構成されており、メモリカード 200 C に対してデータを書き込み、メモリカード 200 C からデータを読み出すことが可能になっている。コントローラ 100 は、メモリカードインターフェイス 101 B を介してメモリカード 200 C からパッケージソフト 141などをダウンロードすることができる。また、メモリカード 200 C には、パッケージソフト 141 のライセンス権 142 や、メモリカード 200 C を一意に識別するための固有情報 243 A などが格納されている。

30

【0083】

(E2. 外部機器 200 のハードウェア構成)

次に、図 7 を参照して、外部機器 200 のハードウェア構成について説明する。図 7 は、外部機器 200 のハードウェア構成を示す模式図である。

40

【0084】

外部機器 200 は、一例として、汎用的なコンピュータアーキテクチャに準じて構成されるコンピュータからなる。外部機器 200 は、たとえば、上述の開発支援装置 200 A または上述のサーバ 200 B である。外部機器 200 は、CPU や MPU などの制御装置 202 と、主メモリ 204 と、不揮発性の記憶装置 208 と、通信インターフェイス 211 と、I/O (Input/Output) インターフェイス 214 と、表示インターフェイス 220 とを含む。これらのコンポーネントは、内部バス 225 を介して互いに通信可能に接続さ

50

れている。

【0085】

制御装置202は、記憶装置208に格納されている開発支援プログラム208Aを主メモリ204に展開して実行することで、プログラム開発ツールにおける各種処理を実現する。開発支援プログラム208Aは、ユーザプログラム145の開発環境を提供するためのプログラムである。記憶装置208は、開発支援プログラム208Aの他にも、配布対象のパッケージソフト141など各種データを格納する。

【0086】

通信インターフェイス211は、他の通信機器との間でネットワークを介してデータを遣り取りする。当該他の通信機器は、たとえば、コントローラ100、サーバなどの外部機器を含む。外部機器200は、通信インターフェイス211を介して、当該他の通信機器から、開発支援プログラム208Aなどの各種プログラムをダウンロード可能なように構成されてもよい。

10

【0087】

I/Oインターフェイス214は、操作部215に接続され、操作部215からのユーザ操作を示す信号を取り込む。操作部215は、典型的には、キーボード、マウス、タッチパネル、タッチパッドなどからなり、ユーザからの操作を受け付ける。

【0088】

表示インターフェイス220は、表示部221と接続され、制御装置202などからの指令に従って、表示部221に対して、画像を表示するための画像信号を送出する。表示部221は、LCD(Liquid Crystal Display)や有機EL(Electro Luminescence)ディスプレイなどからなり、ユーザに対して各種情報を提示する。表示部221には、プログラム開発ツールによって提供される各種画面が表示され得る。なお、図7の例では、外部機器200および表示部221が別体として示されているが、外部機器200および表示部221は、一体的に構成されてもよい。

20

【0089】

< F . 制御構造 >

図8～図12を参照して、コントローラ100の制御構造について説明する。図8は、パッケージソフト141を開発支援装置200Aからダウンロードする処理を表わすフローチャートである。図9は、実行権139を生成することでパッケージソフト141の利用を有効化する処理を表わすフローチャートである。図10は、実行権139のチェック処理を表わすフローチャートである。図11は、ダウンロードしたパッケージソフト141の実行処理を表わすフローチャートである。

30

【0090】

図8～図11に示される処理は、コントローラ100のプロセッサ102がプログラムを実行することにより実現される。他の局面において、処理の一部または全部が、回路素子またはその他のハードウェアによって実行されてもよい。

【0091】

以下では、図8～図11に示されるフローチャートについて順に説明する。

( F 1 . パッケージソフト141のダウンロード処理 )

40

まず、図8を参照して、開発支援装置200Aからのパッケージソフト141のダウンロード処理について説明する。

【0092】

ステップS110において、プロセッサ102は、パッケージソフト141のダウンロード命令を開発支援装置200Aから受信したか否かを判断する。当該ダウンロード命令は、たとえば、ユーザが開発支援装置200Aに対してダウンロード操作を行うことにより発せられる。プロセッサ102は、パッケージソフト141のダウンロード命令を開発支援装置200Aから受信したと判断した場合(ステップS110においてYES)、制御をステップS120に切り替える。そうでない場合には(ステップS110においてNO)、図8に示されるダウンロード処理を終了する。

50

## 【0093】

ステップS120において、プロセッサ102は、ダウンロード対象のパッケージソフト141がダウンロード先のコントローラ100に対応しているか否かを判断する。より具体的には、プロセッサ102は、ダウンロード対象のパッケージソフト141に関するソフト情報を外部機器200から受信する。当該ソフト情報は、たとえば、対応しているコントローラ100の形式情報などを含む。コントローラ100は、パッケージ情報に規定されている形式情報を、コントローラ100の形式情報と比較し、これらの形式情報が対応しているか否かを判断する。これらの形式情報が一致する場合、プロセッサ102は、ダウンロード対象のパッケージソフト141がコントローラ100に対応していると判断する。

10

## 【0094】

プロセッサ102は、ダウンロード対象のパッケージソフト141がコントローラ100に対応していると判断した場合（ステップS120においてYES）、制御をステップS122に切り替える。そうでない場合には（ステップS120においてNO）、プロセッサ102は、制御をステップS126に切り替える。

## 【0095】

ステップS122において、プロセッサ102は、パッケージソフト141と、当該パッケージソフト141のライセンス権142とを開発支援装置200Aから受信する。

## 【0096】

ステップS124において、プロセッサ102は、受信したパッケージソフト141とライセンス権142とをコントローラ100の記憶装置110に保存する。

20

## 【0097】

ステップS126において、プロセッサ102は、ダウンロード対象のパッケージソフト141がコントローラ100に対応していないことを示すエラーを出力する。当該エラーの出力態様は、任意である。一例として、当該エラーは、エラーログとして出力されてもよいし、エラー音やメッセージなどで出力されてもよい。

## 【0098】

（F2．有効化処理）

次に、図9を参照して、パッケージソフト141の利用を有効化するための有効化処理について説明する。当該有効化処理の実行は、任意のタイミングで開始される。一例として、当該有効化処理の実行は、開発支援装置200Aが有効化命令を受信しことに基づいて開始される。当該有効化命令は、たとえば、ユーザが開発支援装置200Aに対して有効化操作を行ったことに基づいて指定されたコントローラ100に発せられ、有効化処理の実行が開始されると、図9に示される各処理が順次実行される。

30

## 【0099】

ステップS180において、コントローラ100のプロセッサ102は、パッケージソフト141のライセンス権142があるか否かを判断する。典型的には、プロセッサ102は、ライセンス権142がコントローラ100の記憶装置110に格納されているか否かに基づいて、ライセンス権142の有無を判断する。プロセッサ102は、パッケージソフト141のライセンス権142があると判断した場合（ステップS180においてYES）、制御をステップS182に切り替える。そうでない場合には（ステップS180においてNO）、プロセッサ102は、制御をステップS184に切り替える。

40

## 【0100】

ステップS182において、プロセッサ102は、上述の生成モジュール154（図1参照）として、コントローラ100の固有情報143Aからパッケージソフト141の実行権139を生成する。

## 【0101】

ステップS184において、プロセッサ102は、パッケージソフト141のライセンス権がないことを示すエラーを出力する。当該エラーの出力態様は、任意である。一例として、当該エラーは、エラーログとして出力されてもよいし、エラー音やメッセージなど

50

で出力されてもよい。

【0102】

(F3. 実行権のチェック処理)

次に、図10を参照して、図9で生成された実行権のチェック処理について説明する。当該チェック処理は、パッケージソフト141の利用を有効化または無効化する処理である。当該チェック処理の実行は、任意のタイミングで開始される。一例として、当該チェック処理の実行は、コントローラ100の起動時や、パッケージソフト141の実行時などに開始される。ライセンスのチェック処理の実行が開始されると、図10に示される各処理が順次実行される。

【0103】

より具体的には、ステップS210において、プロセッサ102は、パッケージソフト141の実行権139が存在するか否かを判断する。プロセッサ102は、パッケージソフト141の実行権139が存在すると判断した場合(ステップS210においてYES)、制御をステップS212に切り替える。そうでない場合には(ステップS210においてNO)、プロセッサ102は、制御をステップS224に切り替える。

【0104】

ステップS212において、プロセッサ102は、上述の復号化モジュール156A(図1参照)として、実行権139からコントローラ100の固有情報143Bを生成する。固有情報143Bは、たとえば、コントローラ100またはプロセッサ102の製造番号、コントローラ名、MACアドレス、または、コントローラ100を一意に識別するためのその他の情報を含む。

【0105】

ステップS220において、プロセッサ102は、上述の比較モジュール156B(図1参照)として、ステップS212で生成した固有情報143Bがコントローラ100に格納されている本来の固有情報143Aに一致しているか否かを判断する。プロセッサ102は、固有情報143Bがコントローラ100の本来の固有情報143Aに一致していると判断した場合(ステップS220においてYES)、制御をステップS222に切り替える。そうでない場合には(ステップS220においてNO)、プロセッサ102は、制御をステップS224に切り替える。

【0106】

ステップS222において、プロセッサ102は、パッケージソフト141の利用を許可する。一例として、利用の禁止/許可を示すフラグがコントローラ100の記憶装置110で管理されており、プロセッサ102は、許可を示すように当該フラグを書き換える。

【0107】

ステップS224において、プロセッサ102は、パッケージソフト141の利用を禁止する。一例として、利用の禁止/許可を示すフラグがコントローラ100の記憶装置110で管理されており、プロセッサ102は、禁止を示すように当該フラグを書き換える。

【0108】

(F4. パッケージソフト141の実行処理)

次に、図11を参照して、ダウンロードしたパッケージソフト141の実行処理について説明する。

【0109】

ステップS310において、プロセッサ102は、パッケージソフト141の実行命令を受け付けたか否かを判断する。プロセッサ102は、パッケージソフト141の実行命令を受け付けたと判断した場合(ステップS310においてYES)、制御をステップS320に切り替える。そうでない場合には(ステップS310においてNO)、プロセッサ102は、ステップS310の処理を再び実行する。

【0110】

10

20

30

40

50

ステップ S 3 2 0 において、プロセッサ 1 0 2 は、実行対象のパッケージソフト 1 4 1 の利用が許可されているか否かを判断する。一例として、プロセッサ 1 0 2 は、パッケージソフト 1 4 1 の利用の禁止 / 許可を示すフラグが許可を示す場合に、プロセッサ 1 0 2 は、パッケージソフト 1 4 1 の利用が許可されていると判断する。プロセッサ 1 0 2 は、実行対象のパッケージソフト 1 4 1 の利用が許可されていると判断した場合 (ステップ S 3 2 0 において YES)、制御をステップ S 3 2 2 に切り替える。そうでない場合には (ステップ S 3 2 0 において NO)、プロセッサ 1 0 2 は、制御をステップ S 3 2 4 に切り替える。

【 0 1 1 1 】

ステップ S 3 2 2 において、プロセッサ 1 0 2 は、パッケージソフト 1 4 1 を実行する

10

【 0 1 1 2 】

ステップ S 3 2 4 において、プロセッサ 1 0 2 は、パッケージソフト 1 4 1 を実行せずに、パッケージソフト 1 4 1 の実行が禁止されていることを示すエラーを出力する。当該エラーの出力態様は、任意である。一例として、当該エラーは、エラーログとして出力されてもよいし、エラー音やメッセージなどで出力されてもよい。

【 0 1 1 3 】

< G . 変形例 1 >

( G 1 . パッケージソフト 1 4 1 の他の販売形態 )

上述では、パッケージソフト 1 4 1 がダウンロード販売される前提で説明を行っていたが、パッケージソフト 1 4 1 の販売形態はこれに限定されない。一例として、パッケージソフト 1 4 1 は、メモリカード 2 0 0 C に格納された上で販売されてもよい。

20

【 0 1 1 4 】

以下では、図 1 2 を参照して、パッケージソフト 1 4 1 の販売形態の他の例について説明する。図 1 2 は、パッケージソフト 1 4 1 をメモリカード 2 0 0 C で販売する仕組みを示す図である。

【 0 1 1 5 】

図 1 2 の例では、会社「 A 」 ~ 会社「 C 」が示されている。会社「 A 」は、パッケージソフト 1 4 1 の管理を行うサーバ 2 0 0 B を有する。会社「 B 」は、パッケージソフト 1 4 1 を格納したメモリカード 2 0 0 C を販売する。会社「 C 」は、パッケージソフト 1 4 1 の販売先である。

30

【 0 1 1 6 】

より具体的な販売形態として、ステップ ( 1 ) において、会社「 B 」において、販売対象のパッケージソフト 1 4 1 を格納したメモリカード 2 0 0 C が製造される。

【 0 1 1 7 】

ステップ ( 2 ) において、会社「 B 」は、パッケージソフト 1 4 1 の登録を会社「 A 」に対して行う。このとき、販売対象のパッケージソフト 1 4 1 に関するパッケージ情報が会社「 A 」のサーバ 2 0 0 B に送信される。サーバ 2 0 0 B は、受信したパッケージ情報に基づいて、パッケージソフト 1 4 1 の登録処理を行う。

【 0 1 1 8 】

ステップ ( 3 ) において、会社「 C 」の購入者は、会社「 B 」のウェブサイトなどにアクセスし、購入対象のパッケージソフト 1 4 1 を選択し、購入手続きを行う。購入手続きが完了すると、会社「 B 」は、購入されたパッケージソフト 1 4 1 を格納したメモリカード 2 0 0 C を準備する。このとき、パッケージソフト 1 4 1 のライセンス権 1 4 2 が発行され、ライセンス権 1 4 2 は、メモリカード 2 0 0 C に保存される。会社「 B 」は、パッケージソフト 1 4 1 およびライセンス権 1 4 2 を格納したメモリカード 2 0 0 C を会社「 C 」に送る。

40

【 0 1 1 9 】

ステップ ( 4 ) において、購入者は、購入したメモリカード 2 0 0 C をコントローラ 1 0 0 に接続し、パッケージソフト 1 4 1 とライセンス権 1 4 2 とをメモリカード 2 0 0 C

50

からインストールする。

【0120】

(G2. パッケージソフト141のダウンロード処理)

図13を参照して、本変形例におけるパッケージソフト141のダウンロード処理について説明する。図13は、パッケージソフト141をメモリカード200Cからダウンロードする処理を表わすフローチャートである。

【0121】

図13に示される処理は、コントローラ100のプロセッサ102がプログラムを実行することにより実現される。他の局面において、処理の一部または全部が、回路素子またはその他のハードウェアによって実行されてもよい。

10

【0122】

ステップS150において、プロセッサ102は、メモリカード200Cがコントローラ100に接続されたか否かを判断する。プロセッサ102は、メモリカード200Cがコントローラ100に接続されたと判断した場合(ステップS150においてYES)、制御をステップS160に切り替える。そうでない場合には(ステップS150においてNO)、プロセッサ102は、図13に示されるダウンロード処理を終了する。

【0123】

ステップS160において、プロセッサ102は、パッケージソフト141のダウンロード命令を開発支援装置200Aから受信したか否かを判断する。当該ダウンロード命令は、たとえば、ユーザが開発支援装置200Aに対してダウンロード操作を行うことにより発せられる。プロセッサ102は、パッケージソフト141のダウンロード命令を開発支援装置200Aから受信したと判断した場合(ステップS160においてYES)、制御をステップS170に切り替える。そうでない場合には(ステップS160においてNO)、図13に示されるダウンロード処理を終了する。

20

【0124】

ステップS170において、プロセッサ102は、ダウンロード対象のパッケージソフト141がダウンロード先のコントローラ100に対応しているか否かを判断する。より具体的には、プロセッサ102は、ダウンロード対象のパッケージソフト141に関するソフト情報をメモリカード200Cから取得する。当該ソフト情報は、たとえば、対応しているコントローラ100の形式情報などを含む。コントローラ100は、パッケージ情報に規定されている形式情報を、コントローラ100の形式情報と比較し、これらの形式情報が対応しているか否かを判断する。これらの形式情報が一致する場合、プロセッサ102は、ダウンロード対象のパッケージソフト141がダウンロード先のコントローラ100に対応していると判断する。

30

【0125】

プロセッサ102は、ダウンロード対象のパッケージソフト141がコントローラ100に対応していると判断した場合(ステップS170においてYES)、制御をステップS172に切り替える。そうでない場合には(ステップS170においてNO)、プロセッサ102は、制御をステップS176に切り替える。

【0126】

ステップS172において、プロセッサ102は、パッケージソフト141と、当該パッケージソフト141のライセンス権142とをメモリカード200Cから取得する。

40

【0127】

ステップS174において、プロセッサ102は、取得したパッケージソフト141とライセンス権142とをコントローラ100の記憶装置110に保存する。

【0128】

ステップS176において、プロセッサ102は、ダウンロード対象のパッケージソフト141がコントローラ100に対応していないことを示すエラーを出力する。当該エラーの出力態様は、任意である。一例として、当該エラーは、エラーログとして出力されてもよいし、エラー音やメッセージなどで出力されてもよい。

50

## 【 0 1 2 9 】

< H . パッケージソフト 1 4 1 のディプロイ処理 >

図 1 4 および図 1 5 を参照して、パッケージソフト 1 4 1 をコントローラ 1 0 0 に配置するディプロイ処理について説明する。図 1 4 は、パッケージソフト 1 4 1 のディプロイ処理の流れを概略的に示す図である。図 1 5 は、図 1 4 に示されるディプロイ処理の続きを示す図である。

## 【 0 1 3 0 】

コントローラ 1 0 0 のプロセッサ 1 0 2 は、ダウンロードしたパッケージソフトを管理するパッケージ管理モジュール 1 5 0 を含む。メモリカード 2 0 0 C には、ダウンロード対象のパッケージソフト 1 4 1 がパッケージソフト " X " として示されている。

10

## 【 0 1 3 1 】

ステップ ( 1 ) において、ある呼出し元が、パッケージソフト " X " のディプロイ命令をパッケージ管理モジュール 1 5 0 に出力する。当該呼出し元としては、たとえば、ファームウェアの U p d a t e r 、外部機器 2 0 0 (たとえば、開発支援装置 2 0 0 A )、コントローラ 1 0 0 内のプロセス、コントローラ 1 0 0 内のライブラリなどが挙げられる。

## 【 0 1 3 2 】

ステップ ( 2 ) において、パッケージ管理モジュール 1 5 0 は、メモリカード 2 0 0 C 上のパッケージソフト " X " のディプロイメソッドを実行する。ディプロイメソッドは、「 Package\_X.db 」に予め定義されており、パッケージ管理モジュール 1 5 0 は、このメソッドを「 Package\_X.db 」から探し実行する。

20

## 【 0 1 3 3 】

ステップ ( 3 ) , ( 4 ) において、ディプロイメソッド 1 4 1 A は、コントローラ 1 0 0 のプロセッサ 1 0 2 の形式情報がパッケージソフト " X " に対応しているか否かのチェックを開始する。

## 【 0 1 3 4 】

より具体的には、ディプロイメソッド 1 4 1 A は、プロセッサ 1 0 2 の固有情報 1 4 3 A を参照して、プロセッサ 1 0 2 の形式情報を取得する。ディプロイメソッド 1 4 1 A は、プロセッサ 1 0 2 の形式情報と、パッケージソフト " X " の対応形式情報とを比較し、パッケージソフト " X " をコントローラ 1 0 0 に配置できるか否かをチェックする。ディプロイメソッド 1 4 1 A は、取得したプロセッサ 1 0 2 の形式がパッケージソフト " X " の対応形式に合致する場合、パッケージソフト " X " をコントローラ 1 0 0 に配置できると判断する。そうでない場合には、ディプロイメソッドは、パッケージソフト " X " をコントローラ 1 0 0 に配置できないと判断する。

30

## 【 0 1 3 5 】

ステップ ( 5 ) , ( 6 ) において、ディプロイメソッド 1 4 1 A は、ダウンロード対象のパッケージソフト 1 4 1 と、コントローラ 1 0 0 に既にダウンロードされているパッケージソフト (以下、「既存のパッケージソフト」ともいう。)との間の依存関係をチェックする。

## 【 0 1 3 6 】

より具体的には、ディプロイメソッド 1 4 1 A は、パッケージソフトの一覧テーブル 1 6 0 を参照して、既存のパッケージソフトの一覧を取得する。その後、ディプロイメソッド 1 4 1 A は、ダウンロード対象のパッケージソフト 1 4 1 と、既存のパッケージソフトとの間の依存関係をチェックする。ダウンロード対象のパッケージソフト 1 4 1 が、既存のパッケージソフトと共存できない関係にある場合、パッケージソフト " X " をコントローラ 1 0 0 に配置できないと判断する。

40

## 【 0 1 3 7 】

ステップ ( 7 ) において、ディプロイメソッド 1 4 1 A は、各種ファイル (たとえば、soファイル) をプロセッサ 1 0 2 に配置する。

## 【 0 1 3 8 】

ステップ ( 8 ) において、ディプロイメソッド 1 4 1 A は、自身の情報を記述した D B

50

ファイル「Package\_X.db」をプロセッサ 1 0 2 に配置する。当該 D B ファイルの一例が上述の詳細情報 1 6 1 ~ 1 6 3 である。

【 0 1 3 9 】

ステップ ( 9 ) において、ディプロイメソッド 1 4 1 A は、各種ファイルの配置が終了したことに基づいて、パッケージ管理モジュール 1 5 0 にパッケージソフト " X " を登録する。

【 0 1 4 0 】

ステップ ( 1 0 ) において、パッケージ管理モジュール 1 5 0 は、新たに配置したパッケージソフト " X " の情報を、パッケージソフトの一覧テーブル 1 6 0 に追加する。

【 0 1 4 1 】

ステップ ( 1 1 ) において、パッケージ管理モジュール 1 5 0 は、パッケージ変更 / 追加の記録を変更履歴 1 6 5 に保存する。

【 0 1 4 2 】

なお、図 1 4 および図 1 5 では、パッケージソフト 1 4 1 をメモリカード 2 0 0 C からダウンロードする場合のディプロイ処理について説明を行ったが、パッケージソフト 1 4 1 をメモリカード 2 0 0 C 以外の外部機器 2 0 0 からパッケージソフト 1 4 1 をダウンロードする場合であっても、図 1 4 および図 1 5 に示されるディプロイ処理が適用され得る。

【 0 1 4 3 】

< I . パッケージソフト 1 4 1 の移動処理 >

コントローラ 1 0 0 にダウンロードされたパッケージソフト 1 4 1 は、外部機器 2 0 0 に移され得る。このとき、パッケージソフト 1 4 1 の実行権 1 3 9 を含むライセンス権 1 4 2 や、ライセンスのチェックを行うための各種機能モジュールがコントローラ 1 0 0 から削除される。これにより、実行権 1 3 9 やライセンス権 1 4 2 が複数存在することがなくなり、ライセンス権 1 4 2 が一意に管理され得る。

【 0 1 4 4 】

以下では、図 1 6 を参照して、パッケージソフト 1 4 1 の移動処理について説明する。図 1 6 は、コントローラ 1 0 0 とメモリカード 2 0 0 C との間でパッケージソフト 1 4 1 を移している様子を示す図である。

【 0 1 4 5 】

図 1 6 ( A ) には、パッケージソフト 1 4 1 をコントローラ 1 0 0 にダウンロードする前の様子が示されている。コントローラ 1 0 0 は、自身の固有情報 1 4 3 A を格納している。メモリカード 2 0 0 C は、ダウンロード対象のパッケージソフト 1 4 1 と、その実行権 1 3 9 を含むライセンス権 1 4 2 と、上述の生成モジュール 1 5 4 と、上述のチェックモジュール 1 5 6 と、メモリカード 2 0 0 C の固有情報 2 4 3 A とを格納している。

【 0 1 4 6 】

図 1 6 ( B ) には、パッケージソフト 1 4 1 をメモリカード 2 0 0 C からコントローラ 1 0 0 にダウンロードする場合の様子が示されている。パッケージソフト 1 4 1 が移されたことに基づいて、パッケージソフト 1 4 1 と、その実行権 1 3 9 を含むライセンス権 1 4 2 と、生成モジュール 1 5 4 と、チェックモジュール 1 5 6 とがメモリカード 2 0 0 C から削除される。メモリカード 2 0 0 C からコントローラ 1 0 0 へのパッケージソフト 1 4 1 の移動機能は、たとえば、パッケージソフト 1 4 1 自体に組み込まれおり、コントローラ 1 0 0 のプロセッサ 1 0 2 によって実行される。

【 0 1 4 7 】

図 1 6 ( C ) には、パッケージソフト 1 4 1 をコントローラ 1 0 0 からメモリカード 2 0 0 C に移す場合の様子が示されている。パッケージソフト 1 4 1 が移されたことに基づいて、パッケージソフト 1 4 1 と、その実行権 1 3 9 を含むライセンス権 1 4 2 と、生成モジュール 1 5 4 と、チェックモジュール 1 5 6 とがコントローラ 1 0 0 から削除される。コントローラ 1 0 0 からメモリカード 2 0 0 C へのパッケージソフト 1 4 1 の移動機能は、たとえば、パッケージソフト 1 4 1 自体に組み込まれおり、コントローラ 1 0 0 のプ

10

20

30

40

50

ロセッサ 102 によって実行される。

【0148】

なお、図 16 では、コントローラ 100 とメモリカード 200C との間におけるパッケージソフト 141 の移動処理について説明を行ったが、メモリカード 200C 以外の外部機器 200 とコントローラ 100 との間におけるパッケージソフト 141 の移動時においても移動機能は応用され得る。

【0149】

< J . 変形例 2 >

上述では、パッケージソフト 141 がダウンロード販売される例、および、パッケージソフト 141 がメモリカード 200C に格納された上で販売される例について説明を行ったが、パッケージソフト 141 の販売形態はこれらに限定されない。一例として、パッケージソフト 141 は、BTO 販売されてもよい。

【0150】

以下では、図 17 を参照して、パッケージソフト 141 の販売形態のさらに他の例について説明する。図 17 は、パッケージソフト 141 を BTO 販売する仕組みを示す図である。

【0151】

図 17 の例では、会社「A」～会社「C」が示されている。会社「A」は、パッケージソフト 141 の管理を行うサーバ 200B を有する。会社「B」は、パッケージソフト 141 を販売する会社である。会社「C」は、会社「B」や他の会社からパッケージソフト 141 やコントローラ 100 などを購入し、これらを組み合わせたシステムを販売する会社である。

【0152】

より具体的な販売形態として、ステップ(1)において、会社「B」の設計者は、開発支援装置 200A を用いて、販売対象のパッケージソフト 141 を設計する。

【0153】

ステップ(2)において、パッケージソフト 141 の販売元である会社「B」は、販売対象のパッケージソフト 141 の登録を会社「A」に対して行う。このとき、販売対象のパッケージソフト 141 に関するパッケージ情報が会社「A」のサーバ 200B に送信される。サーバ 200B は、受信したパッケージ情報に基づいて、パッケージソフト 141 の登録処理を行う。

【0154】

ステップ(3)において、会社「C」の購入者は、会社「B」のウェブサイトなどにアクセスし、ダウンロード対象のパッケージソフト 141 を選択し、購入手続きを行なう。このとき、ダウンロード先のコントローラ 100 の固有情報が入力される。購入手続きが完了すると、ダウンロード先のコントローラ 100 を一意に識別するための固有情報からライセンス権 142 が生成され、購入されたパッケージソフト 141 とともに会社「C」に送信される。

【0155】

ステップ(4)において、購入対象のパッケージソフト 141 とライセンス権 142 とがコントローラ 100 にインストールされる。

【0156】

ステップ(5)において、会社「C」は、パッケージソフト 141 とライセンス権 142 とがインストールされたコントローラ 100 を他の会社に販売する。

【0157】

< K . まとめ >

以上のように、コントローラ 100 は、パッケージソフト 141 のダウンロード時などにコントローラ 100 の固有情報 143A からパッケージソフト 141 の実行権 139 を生成しておく。その後、コントローラ 100 は、パッケージソフト 141 の実行時やコントローラ 100 の起動時などにおいて実行権 139 から固有情報 143B を生成する。生

10

20

30

40

50

成された固有情報 143B が本来の固有情報 143A と一致しないときに、コントローラ 100 は、パッケージソフト 141 の利用を禁止する。

【0158】

これにより、たとえば、パッケージソフト 141 がダウンロード時とは異なるコントローラに不正に移動された場合に、パッケージソフト 141 の利用することができる。また、ライセンスをチェックする仕組みがパッケージソフト 141 に組み込まれていることで、ライセンスのチェック機能をコントローラ 100 に予め搭載する必要がなくなる。

【0159】

< L . 付記 >

以上のように、本実施形態は以下のような開示を含む。

10

【0160】

[ 構成 1 ]

制御対象を制御するためのコントローラ ( 100 ) であって、

前記コントローラ ( 100 ) で利用されるソフトウェア ( 141 ) を外部機器 ( 200 ) から取得するための取得部 ( 101 ) と、

前記ソフトウェア ( 141 ) を実行するためのプロセッサ ( 102 ) とを備え、

前記ソフトウェア ( 141 ) は、

前記コントローラ ( 100 ) または前記外部機器 ( 200 ) を一意に識別するための第 1 固有情報から前記ソフトウェア ( 141 ) の実行権 ( 139 ) を生成するための生成モジュール ( 154 ) と、

20

前記実行権 ( 139 ) の生成後の所定のタイミングにおいて前記実行権 ( 139 ) から第 2 固有情報を生成し、当該第 2 固有情報が前記第 1 固有情報と一致しないときに、前記ソフトウェア ( 141 ) の利用を禁止するためのチェックモジュール ( 156 ) とを含む、コントローラ。

【0161】

[ 構成 2 ]

前記外部機器 ( 200 ) は、前記コントローラ ( 100 ) と通信可能に構成されている通信機器を含み、

前記取得部 ( 101 ) は、前記コントローラ ( 100 ) と通信接続された前記通信機器から前記ソフトウェア ( 141 ) をダウンロードする、構成 1 に記載のコントローラ。

30

【0162】

[ 構成 3 ]

前記外部機器 ( 200 ) は、前記コントローラ ( 100 ) に電氣的に接続され得るメモリカードを含み、

前記取得部 ( 101 ) は、前記コントローラ ( 100 ) に電氣的に接続されたメモリカードから前記ソフトウェア ( 141 ) を取得する、構成 1 または 2 に記載のコントローラ。

【0163】

[ 構成 4 ]

前記プロセッサ ( 102 ) は、前記取得したソフトウェア ( 141 ) を前記コントローラ ( 100 ) から前記外部機器 ( 200 ) に移すときに、前記実行権 ( 139 ) を削除する、構成 1 ~ 3 のいずれか 1 項に記載のコントローラ。

40

【0164】

[ 構成 5 ]

前記プロセッサ ( 102 ) は、前記取得したソフトウェア ( 141 ) を前記コントローラ ( 100 ) から前記外部機器 ( 200 ) に移すときに、前記生成モジュール ( 154 ) および前記チェックモジュール ( 156 ) を削除する、構成 1 ~ 4 のいずれか 1 項に記載のコントローラ。

【0165】

[ 構成 6 ]

50

前記外部機器（２００）から取得され得るソフトウェア（１４１）は、前記コントローラ（１００）のファームウェア上で動作するプログラムと、当該ファームウェアに追加されるプログラムと、前記コントローラ（１００）で利用され得るデータとの少なくとも一つを含む、構成１～５のいずれか１項に記載のコントローラ。

【０１６６】

[構成７]

制御対象を制御するコントローラ（１００）で利用されるソフトウェア（１４１）のライセンスの管理方法であって、

前記コントローラ（１００）で利用されるソフトウェア（１４１）と、当該ソフトウェア（１４１）についての第１ライセンス情報とを外部機器（２００）から取得するステップと、

10

前記ソフトウェア（１４１）を実行するステップとを備え、

前記実行するステップは、

前記コントローラ（１００）または前記外部機器（２００）を一意に識別するための第１固有情報から前記ソフトウェア（１４１）の実行権（１３９）を生成するステップと、

前記実行権（１３９）の生成後の所定のタイミングにおいて前記実行権（１３９）から第２固有情報を生成し、当該第２固有情報が前記第１固有情報と一致しないときに、前記ソフトウェア（１４１）の利用を禁止するステップとを含む、管理方法。

【０１６７】

20

[構成８]

制御対象を制御するコントローラ（１００）で利用されるソフトウェア（１４１）のライセンスの管理プログラムであって、

前記管理プログラムは、前記コントローラ（１００）に、

制御対象を制御するコントローラ（１００）で利用されるソフトウェア（１４１）を外部機器（２００）から取得するステップと、

前記ソフトウェア（１４１）を実行するステップとを実行させ、

前記実行するステップは、

前記コントローラ（１００）または前記外部機器（２００）を一意に識別するための第１固有情報から前記ソフトウェア（１４１）の実行権（１３９）を生成するステップと、

30

前記実行権（１３９）の生成後の所定のタイミングにおいて前記実行権（１３９）から第２固有情報を生成し、当該第２固有情報が前記第１固有情報と一致しないときに、前記ソフトウェア（１４１）の利用を禁止するステップとを含む、管理プログラム。

【０１６８】

今回開示された実施の形態は全ての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内での全ての変更が含まれることが意図される。

【符号の説明】

【０１６９】

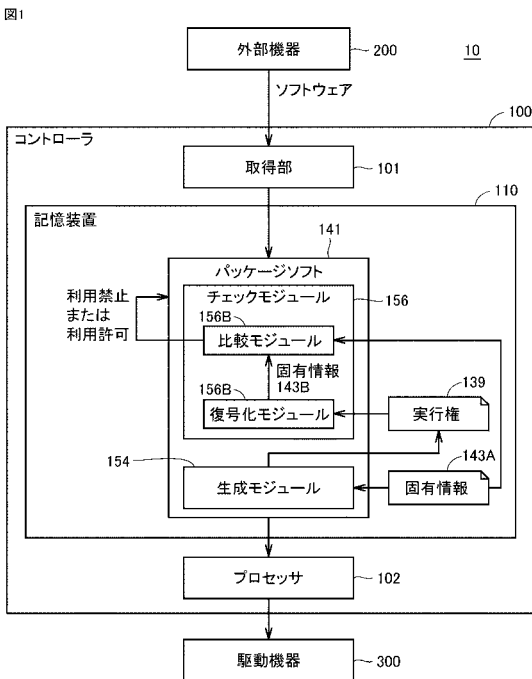
40

１０　ＦＡシステム、１００、１００Ａ、１００Ｂ　コントローラ、１０１　取得部、  
 １０１Ａ、２１１　通信インターフェイス、１０１Ｂ　メモリカードインターフェイス、  
 １０２　プロセッサ、１０４　チップセット、１０６、２０４　主メモリ、１１０、２０  
 ８　記憶装置、１２２　内部バスコントローラ、１２４　フィールドバスコントローラ、  
 １２６　Ｉ／Ｏユニット、１３０　制御プログラム、１３９　実行権、１４０　システム  
 プログラム、１４１　パッケージソフト、１４１Ａ　ディプロイメソッド、１４２　ライ  
 センス権、１４３Ａ、１４３Ｂ、２４３Ａ　固有情報、１４５　ユーザプログラム、１４  
 ５Ａ　シーケンスプログラム、１４５Ｂ　モーションプログラム、１５０　パッケージ管  
 理モジュール、１５４　生成モジュール、１５５　ソフトウェア管理テーブル、１５６  
 チェックモジュール、１５６Ａ　復号化モジュール、１５６Ｂ　比較モジュール、１６０

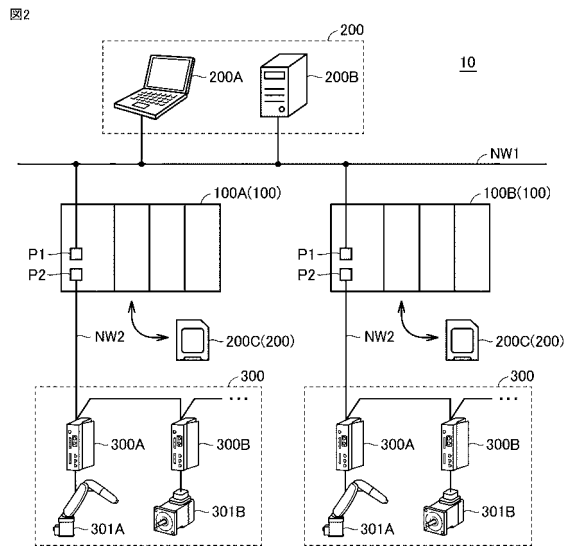
50

一覧テーブル、160A, 160B, 160C, 160D, 160E, 160F, 161A, 161B, 161C, 161D, 162A, 162B, 162C, 162D, 163A, 163B, 163C, 163D カラム、161, 162, 163 詳細情報、165 変更履歴、200 外部機器、200A 開発支援装置、200B サーバ、200C メモリカード、202 制御装置、208A 開発支援プログラム、214 インターフェイス、215 操作部、220 表示インターフェイス、221 表示部、225 内部バス、300 駆動機器、300A ロボットコントローラ、300B サーボドライバ、301A アームロボット、301B サーボモータ。

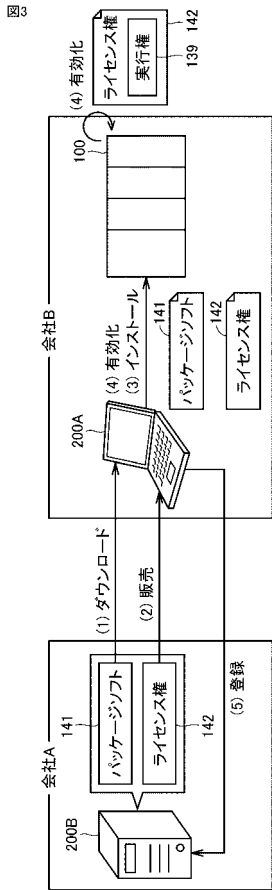
【図1】



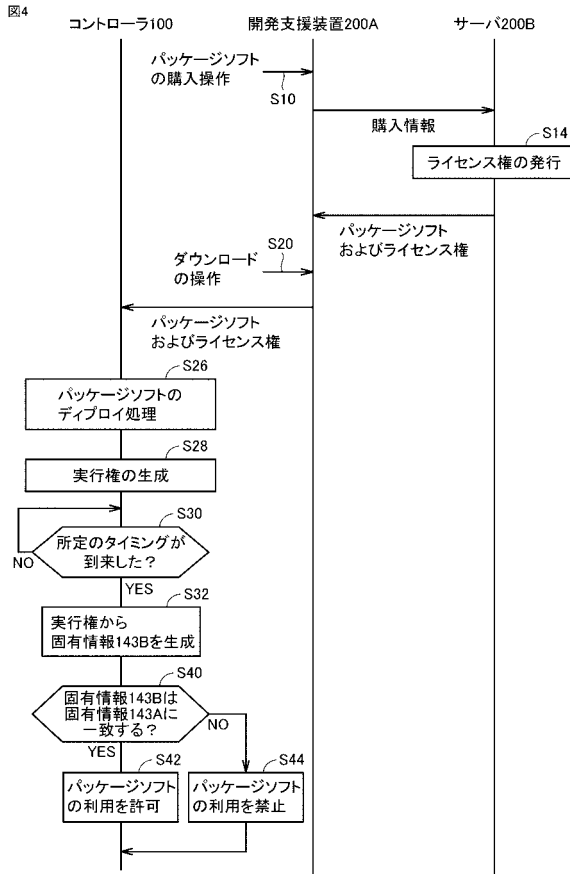
【図2】



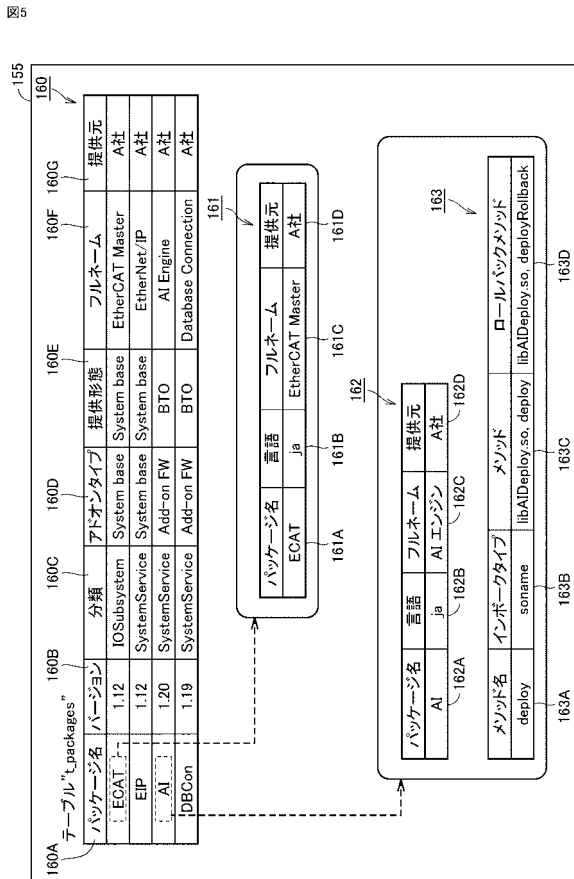
【図3】



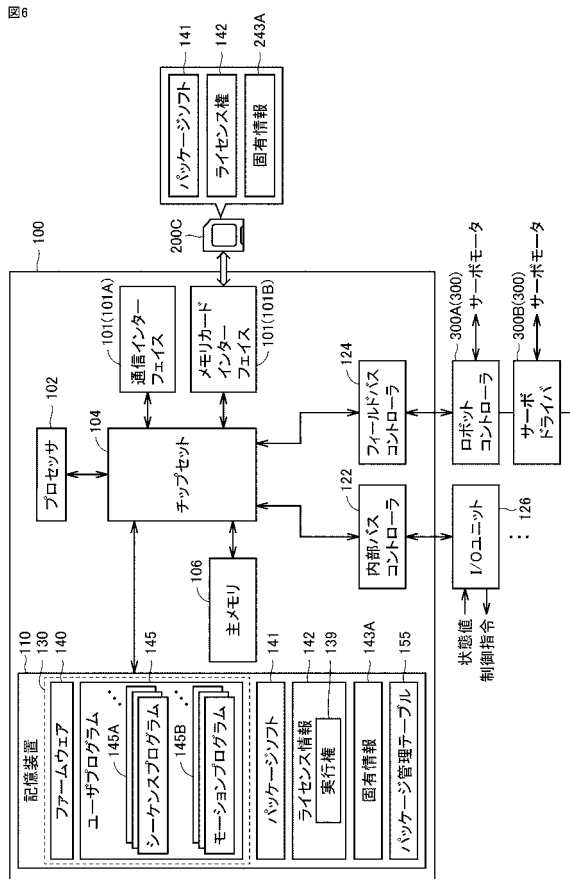
【図4】



【図5】

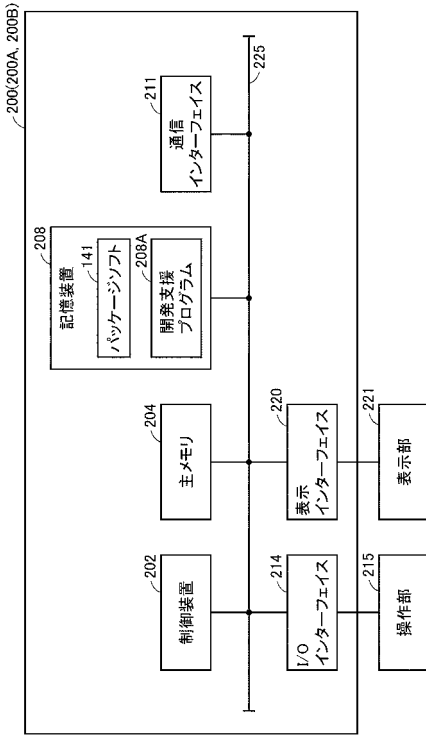


【図6】



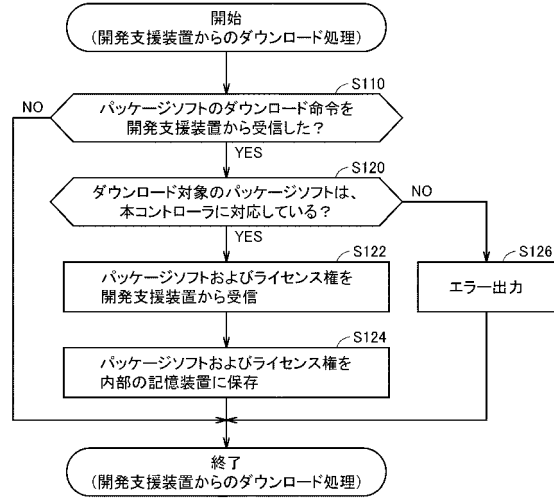
【 図 7 】

図7



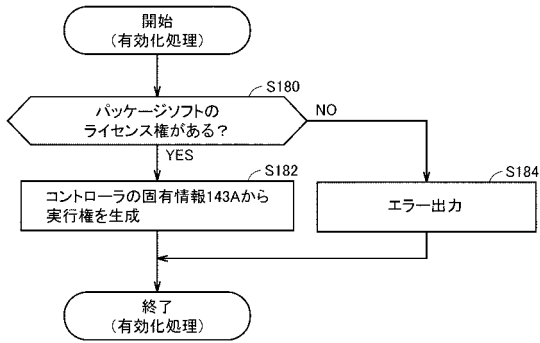
【 図 8 】

図8



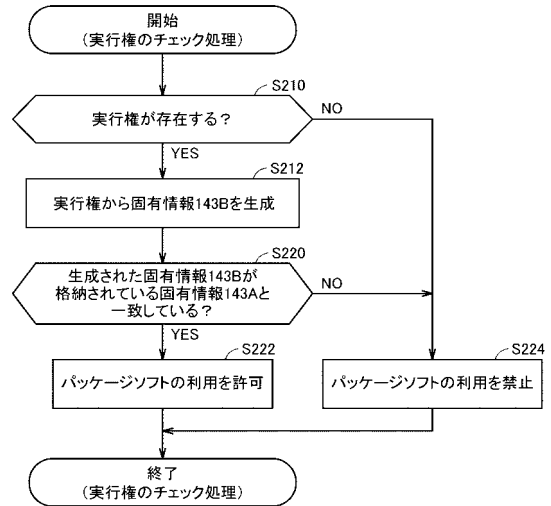
【 図 9 】

図9



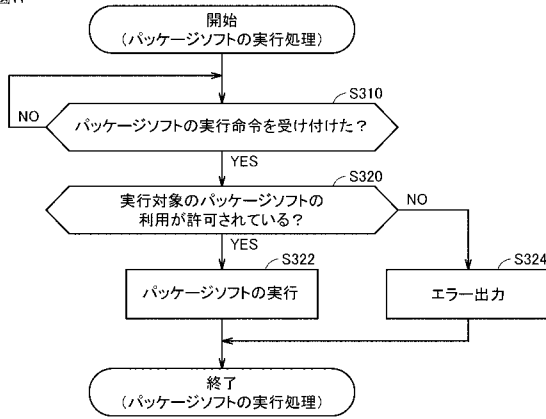
【 図 10 】

図10



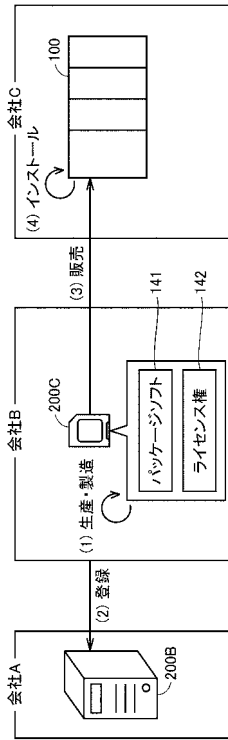
【 図 1 1 】

図11



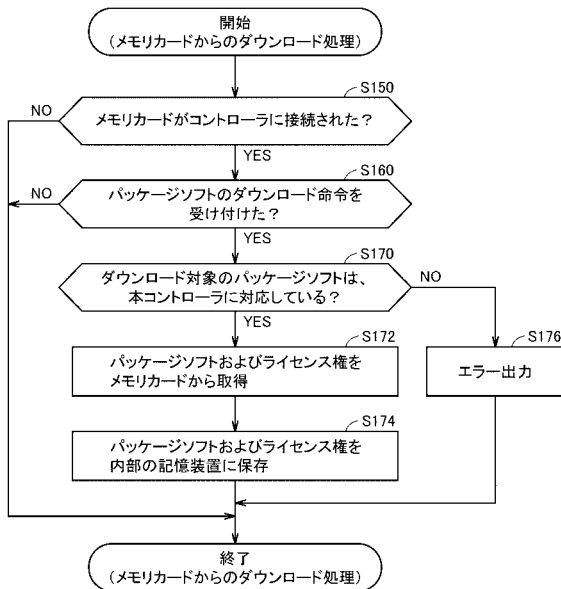
【 図 1 2 】

図12



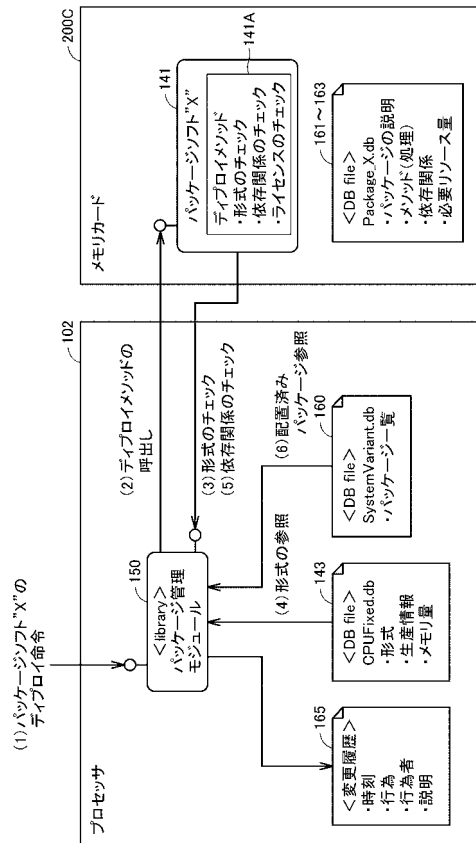
【 図 1 3 】

図13

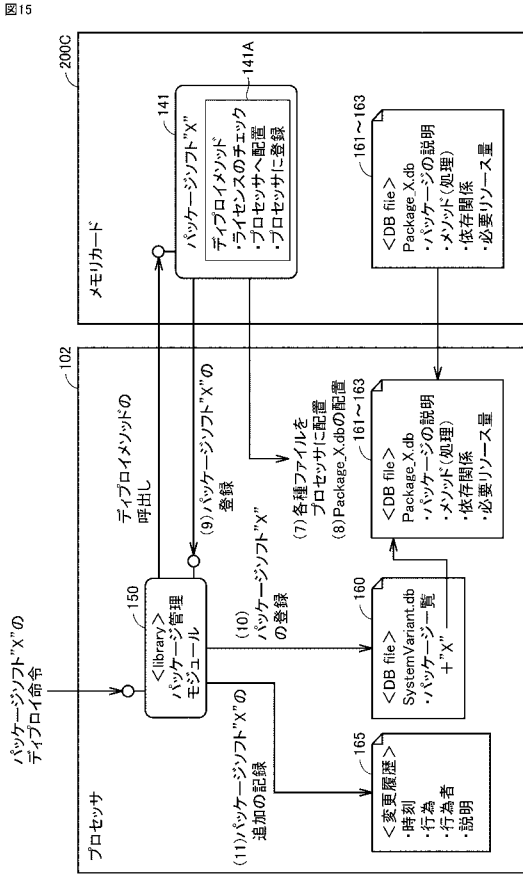


【 図 1 4 】

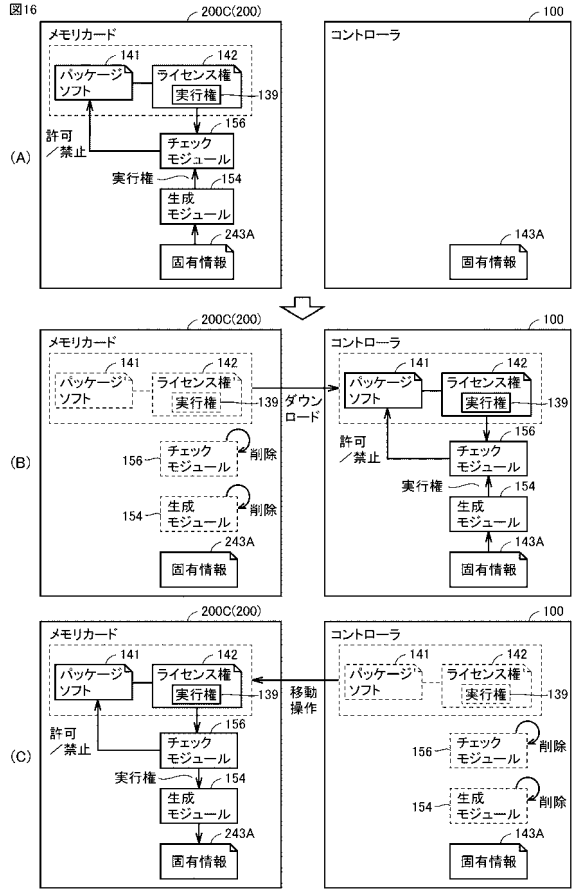
図14



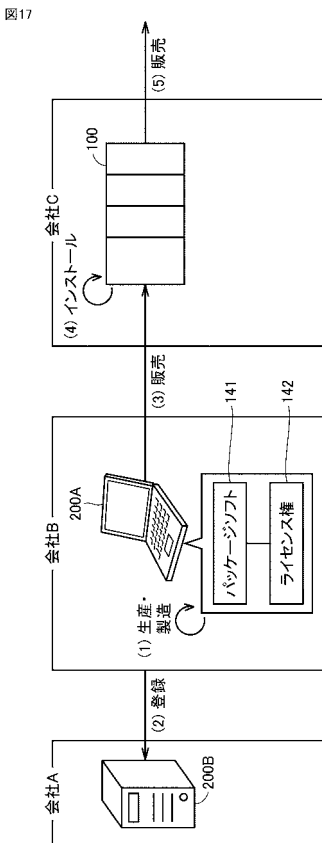
【 図 15 】



【 図 16 】



【 図 17 】



フロントページの続き

Fターム(参考) 5B376 AB01 AB27 AB49 AD05 AD23 GA13