分散システム／インターネット運用技術 20－5
（2000．12．1）

ソフトウェア開発環境と配布管理機能の連携方式

木原 健一† 中島 克雄†† 平田 俊明†
†株式会社日立製作所システム開発研究所
††株式会社日立製作所ソフトウェア開発本部

多数のコンピュータに一括してソフトウェアを導入する際には，ソフト配布管理ツールが用いられる。ただし，ソフト配布の際にはソフト固有の設定作業が必要となるため，システム管理者の作業に煩雑さが残る。それを改善するため，ソフトウェアの導入に関する情報を開発環境側で定義しておき，配布管理ツールがその情報に基づいて配布するシステムを提案する。これにより，従来ソフト登録時に必要だった手作業を一部自動化でき，システム管理者の負担を軽減できる。なお開発環境からの情報の受け渡しは流通ソフトに対応するためファイル経由とし，情報の形式は相互運用性に配慮しDMTFの標準仕様CIM Applicationをベースとする。

Software Delivery Management System Working With Software Development Environment

KENICHI KIHARA,† KATSUO NAKASHIMA,†† TOSHIAKI HIRATA†
†Systems Development Laboratory, Hitachi, Ltd.
††Software Development Division, Hitachi, Ltd.

To install the software on many machines all at once, we usually use software distribution tools. However it is necessary to customize detail settings of the software on the distribution. Such tasks are burdensome to system administrators. To improve it, we propose a new concept of software distribution system. At first, when software is developed, "definition tool" creates software distribution information, and when the software is distributed, "distribution tool" reads the information and distributes the software according to the information. This system can reduce manual procedures of system administrators. The specification of the distribution information is based on DMTF CIM Application for the interoperability.

1．はじめに

1.1 ソフト管理

近年，企業情報システムのTCO(Total Cost of Ownership)低減に対する関心が高まっており，パッケージソフトウェアやユーザアプリケーションについても，開発効率の向上や運用効率・コスト低減に対するニーズが高まってきている。ソフト管理ツールは前記ニーズにこたえるものであり，たとえば表1に示すような機能を提供する。

ソフトウェア管理に関する最近の動向として，Manageabilityの向上がある。これは管理対象であるソフトウェア側も管理されることを意識することによって，ソフトウェアをより管理しやすくすることを狙いとするものである。標準化組織DMTF(Distributed Management Task Force, Inc.)では，その思想に基づいたManagement Readyなアプリケーションを実装するための標準情報モデルCIM(Common Information Model) Applicationが提唱されている[2)3)]。CIM Applicationについては，2.3節で述べる。

表1. ソフトウェア管理の機能

項目	概要
導入・削除	実行マシンへのソフトの導入・削除(ネットワーク経由のソフト配布を含む)。
自動運用	プログラムを決められたスケジュールに基づいて自動実行する。
稼動監視	ソフトの実行・未実行，性能・負荷の状況を監視。
資産管理	各コンピュータへのソフトの導入状況の管理。
セキュリティ	ウィルス対策，ソフトに対するアクセス制限等。

1.2 ライフサイクルと連携機能

ソフトウェアは，ソフトウェア開発環境で作成された後，そのソフトウェアを実行するコンピュータに導入され，使用され，不要になった時に削除される（図1参照)。この開発から削除までの一連の手続きをソフトウェアのライフサイクルと呼ぶ。ソフト管理においてもこのライフサイクルを意識し管理機能間につながりを持たせることによって，よりスムーズな運用が可能になるものと考えられる。

本研究ではこの点に注目し，特にソフトウェアライフサイクルの初期部分である「開発」と「導入」間の連携方式について述べる。

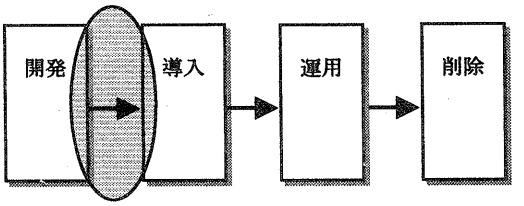

図1. ライフサイクルと今回の検討ポイント

1.3 開発フェーズと導入フェーズの連携

ソフトウェアのライフサイクル上の「開発」フェーズではコンパイラやリンカなどを含む開発環境が使われる。一方，「導入」フェーズでは手動インストールまたはソフト配布ツールが用いられる。

ソフト配布ツールはネットワーク経由でソフトを配布・インストールする機能を持ち，多数のコンピュータに一括してソフトをインストールする際に有効だが，ソフト配布の際に，個々のソフトウェアの特性に合わせた設定処理が必要となる。たとえば，Windows[TM1]アプリケーションの場合，ソフトウェアのセットアップ方法がソフトの種類によって異なるため，ソフト配布ツールはソフト個別の対応が必要になる。このことは，ソフト配布処理の煩雑化を招く原因となる。

本研究では，この点を改善し，配布運用を円滑にすることを目的とする。

2．ソフト配布の概要と解決方式

2.1 ソフト配布処理の概要

図2に従ってソフト配布処理の概要を示す。

(1) システム構成

配布管理システムは配布マネージャと配布ターゲットから成る。配布マネージャはソフト配布を管理し，配布ターゲットに対してファイルを配

[1] Windowsは，米国およびその他の国における米国Microsoft Corp.の登録商標です。

る機能を持つ。一方，配布ターゲットはソフトの導入先であり，配布マネージャからソフトを受け取ってインストールする機能を持つ。

(2) 配布手順

ソフト配布の際には，以下の処理を行う。

STEP1 ソフトの登録：システム管理者が配布するソフトを配布マネージャに登録する。

STEP2 配布処理の指示：システム管理者が配布マネージャに対して登録済みのソフトをどの配布ターゲットに配布するか指定する。

STEP3 配布の実行：配布マネージャから配布ターゲットにソフトが配布される。

STEP4 インストール：配布されたソフトがインストールされる。

STEP5 結果の通知：配布・インストール処理結果（正常・異常）を配布マネージャに通知する。

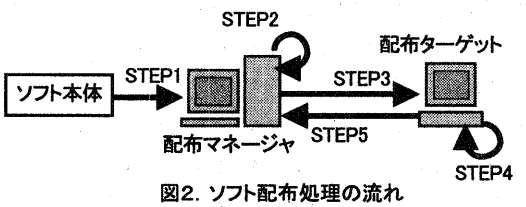

図2. ソフト配布処理の流れ

2.2 課題の解決方式

開発環境とソフト配布ツールを連携させるためには，開発環境をソフト配布時に行われる最初の処理であるSTEP1に接続することが考えられる。すなわち，図3に示すように，ソフト配布時に必要となる情報（ソフト定義情報）を開発環境側で作成し，ソフト本体と共に配布マネージャに渡す。配布マネージャは渡されたソフト定義情報に基づいて一連の配布処理（STEP1～STEP5）を行う。

このような連携を行う際の場合，開発環境から配布管理機能間にソフト本体およびソフト定義情報を渡す必要があるが，その方法として次の二通りの方式が考えられる。

(1) オンライン方式

開発環境と配布管理機能を直接ネットワークで結び，情報を渡す。

(2) 媒体方式

情報をファイルの形で記憶媒体に格納し，その媒体を配布管理機能に読ませる。

両方式の適性を考慮すると，同一場所に開発環境と配布管理機能が存在する場合には(1)が，ソフトをパッケージの形で流通させる場合には(2)がそれぞれ向くと考えられる。本研究では，現在利用されているソフトの多くがCD-ROMなどの媒体によって提供されていることを勘案し，(2)の媒体方式を採用する。

2.3 ソフト定義情報の形式

開発環境から配布管理機能に渡されるソフト定義情報の形式は，情報の再利用性や相互運用性を考慮すると標準仕様を用いるべきである。本連携機能では1.2節で触れたCIM Applicationを利用する。CIM Applicationはソフトウェアの構成管理および配備管理(Deployment Management)を目的としたオブジェクト指向型のデータ形式であり，以下の特徴を持つ。

(1) 三階層構成

ソフトウェアの内部構成を表現するために，CIM Applicationではソフト製品に対応するCIM_Product，ソフト製品が提供する機能に対応するCIM_SoftwareFeature，ソフト製品の機能を実現するために必要となるソフト部品であるCIM_SoftwareElementの三つのオブジェクト（クラス）を用いる。これらのオブジェクトを定義し，オブジェクト間の関連を定義することで，ソフト

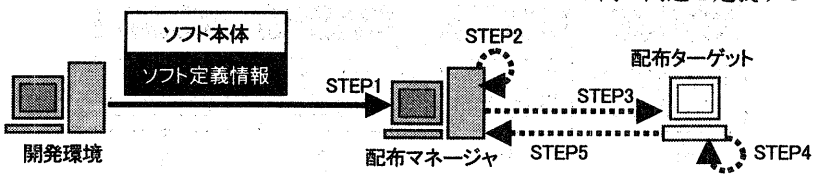

図3. 開発環境と配布ツールの連携

製品が持つ機能，ソフト部品を表現できる。

(2) 四つの状態

また，CIM_SoftwareElement の導入状況・実行状況を示す四つの状態(State)が規定されている。

・Deployable State･･･ソフト配備可能状態

・Installable State･･･インストール可能状態

・Executable State･･･実行可能状態

・Running State･･･実行中

これらの State の関係は，図4に示す通りで，各 State において Next-State または Uninstall の Action (CIM_Action) を行うことで他の状態に遷移する。また，状態遷移に関する前提条件 (CIM_Check) として， State 維持 (In-State) の条件，および次の State に進む (Next-State) 条件を指定できる。

3．提案方式

3.1 連携機能実現のための機能追加項目

開発環境と配布管理機能の連携を実現するためには，以下の開発が必要になる。

(1) 開発環境側

従来，ソフトのインストール・実行に関するファイルを作成する機能を持っていたが，それに加えてソフト定義情報を作成・出力する機能が必要となる。

(2) 配布管理機能に読み込み機能を追加

配布マネージャのソフト登録機能 (2.1節(2) STEP1) に，ソフト定義情報を読み込み，その情報に基づいてソフトを登録する機能が必要となる。

3.2 ソフト定義情報の形式

ソフト定義情報の形式は2.3節で述べた通り，CIM Application の仕様に基づくが，その場合，以下の二項目の未対応部分が発生する。

(1) ソフト配布に関する情報

CIM Application はソフト配布の概念が無く，セットアップに必要なファイルは実行マシン上に存在することを前提としている。

(2) 特定の OS に依存する情報

本研究では配布対象として Windows™ 用アプリケーションを想定しているが，CIM では特定の OS に依存する情報は規定されていない。

これらの不足を補うために，CIM Application の拡張を行った。図5が本連携機能で使用するデータ構造[6]であり，UML(Unified Modeling Language) 形式[7]で記載している。長方形はオブジェクトを，長方形間を結ぶ線はオブジェクト間の関連を示している。関連を示す線の末端が菱形のものは集約（菱形の無い側が菱形のある側に属する）を示し，線の末端が三角のものはクラスの継承（三角のある側がスーパークラス）を示す。

図5に記載されているオブジェクトの内，CIM_Product，CIM_SoftwareFeature，CIM_SoftwareElement，CIM_Check，CIM_Action は CIM Application で規定されているクラスであり，2.3節で述べた。図5の太枠内のオブジェクトが本研究で追加したクラスで，以下その概要を示す。

AIF_PkgProduct はソフト製品に対応するクラスであり，配布時に必要な情報の内，CIM_Product に含まれていない情報を定義する。また，Windows™ アプリケーション特有の概念である「最小」や「標準」などインストールオプション単位のインストールに対応するため，インストールオプションに相当する AIF_InstallSet クラス及びインストーラ定義クラス

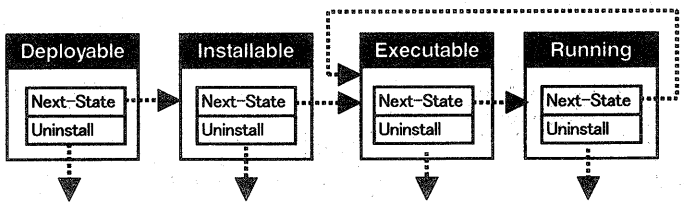

図4. Software Element(ソフトウェアモジュール)の状態遷移

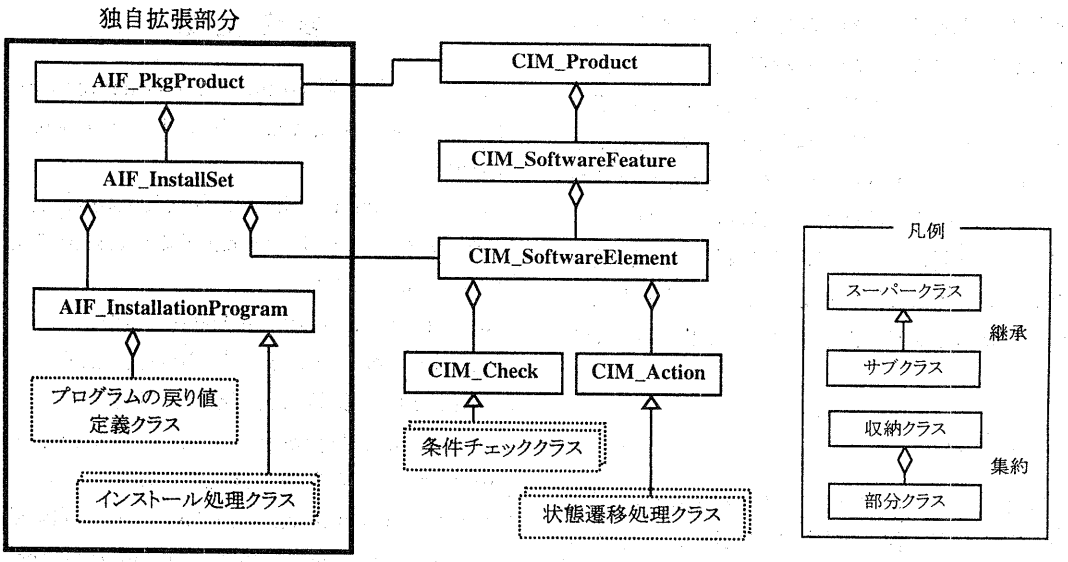

図5. 開発環境と配布管理機能間の連携用クラス

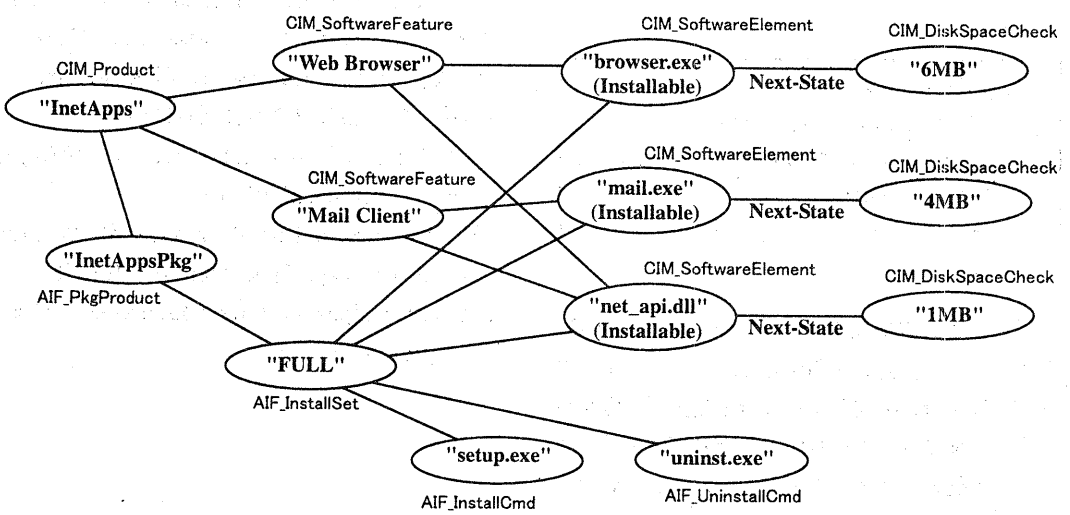

図6. インスタンスの構成例

AIF_InstallationProgramを追加定義した。

以上の追加により,ソフト配布およびWindows™アプリケーションのインストールというCIMでサポートされていない機能への対応が可能になる。

3.3 ソフト定義情報の作成

連携の際に実際に渡される情報は，3.2節で規定したクラス定義に基づいたインスタンスである。図6はインスタンスの例で,以下その概要を示す。

ソフト製品（CIM_Product）の"InternetApps"は，"Web Browser"と"Mail Client"という二つの機能を持っている。さらに，"Web Browser"は"browser.exe"と"net_api.dll"という二つのソフト部品から成っている（"Mail Client"も同様)。各ソフト部品はインストールの条件として，それぞれ空きディスク容量（CIM_DiskSpaceCheck）が設定されている。また，インストールオプション

として，"FULL"が定義されており，すべてのソフト部品をインストールする設定になっている。なおインストール用のコマンドは"setup.exe"，アンインストール用コマンドは"uninst.exe"である。

CIMでは，通常のオブジェクト（名前付きオブジェクト，図6では楕円で示される）に加え，オブジェクト間の関連（図6では直線で表現）もオブジェクトとして扱われる。ソフト定義情報はこの二種類のオブジェクトをXML表記したものである。

4．結果

本連携方式を用いることで従来システム管理者の手作業が必要だった以下の項目入力を自動化できる。このことによってソフトウェアの配布管理を円滑化し，システム管理者の作業負担を軽減できる。

- ソフトウェア名
- ソフトウェアのバージョン番号
- インストール時の前提条件（ハード要件等）
- インストール方法

5．むすび

ソフトウェア開発環境とソフトウェア配布管理ツールの連携方式を開発した。ソフトウェアの配布・運用時に必要となる情報を開発環境側から指定し，配布管理ツール側からその情報を読み取ることで従来システム管理者の手入力が必要だったソフト登録作業を一部自動化した。本システムの特徴は以下の通りである。

(1) 開発環境と配布管理ツール間の情報受け渡し方法は，流通ソフトへの対応に配慮し，ファイル渡しとした。

(2) 開発環境から配布管理ツールへ渡す情報（ソフト定義情報）の形式はDMTFのCIM Applicationをベースとし，OS固有機能および配布機能への対応のため仕様を一部拡張した。

参考文献

1) Rick, S., Winston, B.: Foundations of Application Management, John Wiley & Sons (1998)

2) Understanding the Application Management Model Version 1.0: Distributed Management Task Force, Inc. (1998)

3) Common Information Model (CIM) Specification Version 2.2: Distributed Management Task Force, Inc. (1999)

4) XML As a Representation for Management Information - A White Paper Version 1.0: Distributed Management Task Force, Inc. (1998)

5) Mike, K.: Gain Control of Application Setup and Maintenance with the New Windows Installer, MICROSOFT SYSTEMS JOURNAL VOL.13 NO.9, pp15-27, Miller Freeman Inc. (1988)

6) Application Information File Version 1.0 仕様書: 株式会社日立製作所(2000). http://outside2.soft.hitachi.co.jp/PROD/jp1/v5/news/news2000021701.htm

7) Static Structure Diagrams, OMG Unified Modeling Language Specification (draft) Version 1.3 alpha R5: Object Management Group, Inc., pp243-296 (1999)