

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5524187号
(P5524187)

(45) 発行日 平成26年6月18日 (2014. 6. 18)

(24) 登録日 平成26年4月18日 (2014. 4. 18)

(51) Int. Cl. F I
G06Q 50/10 (2012.01) G06Q 50/10

請求項の数 9 (全 31 頁)

(21) 出願番号	特願2011-508974 (P2011-508974)	(73) 特許権者	509347273
(86) (22) 出願日	平成21年5月14日 (2009. 5. 14)		エアバス オペレーション ソシエテ パ
(65) 公表番号	特表2011-521339 (P2011-521339A)		アクションズ シンプリフィエ
(43) 公表日	平成23年7月21日 (2011. 7. 21)		フランス国, エフ-31060 トゥールズ,
(86) 国際出願番号	PCT/FR2009/000567		ズ, ルート ド ベイヨンヌ 316
(87) 国際公開番号	W02009/147311	(74) 代理人	100099759
(87) 国際公開日	平成21年12月10日 (2009. 12. 10)		弁理士 青木 篤
審査請求日	平成24年4月18日 (2012. 4. 18)	(74) 代理人	100092624
(31) 優先権主張番号	0853122		弁理士 鶴田 準一
(32) 優先日	平成20年5月14日 (2008. 5. 14)	(74) 代理人	100122965
(33) 優先権主張国	フランス (FR)		弁理士 水谷 好男
		(74) 代理人	100141162
			弁理士 森 啓

最終頁に続く

(54) 【発明の名称】 共同サービス指向ワークショップのためのプロセス管理方法

(57) 【特許請求の範囲】

【請求項 1】

情報処理システムにおいて利用される共同サービス指向ワークショップにおいて、処理装置を有する装置がプロセスを管理する方法であって、

前記共同サービス指向ワークショップは、前記処理装置が、実データを表わすデータに関連付けられたオブジェクトと、プロセスを表わすデータに関連付けられたオブジェクトと、を処理するように適合し、

実行エンジンモジュールが、少なくとも一つの第1の関数を決定するステップと、前記実行エンジンモジュールが、前記少なくとも一つの第1の関数の実行を行うことを可能にする、少なくとも一つの情報を決定するステップと、

前記実行エンジンモジュールが、前記少なくとも一つの第1の関数及び前記少なくとも一つの情報に対する少なくとも一つの参照を持つ、前記実データ又は前記プロセスを表すデータに関連付けられた第1のオブジェクトを生成するステップと、

前記実行エンジンモジュールが、前記少なくとも一つの情報に関する前記少なくとも一つの第1の関数を実行するステップと、
を有し、

前記少なくとも一つの第1の関数の実行に応答して、

前記実行エンジンモジュールが、前記前記実データ又は前記プロセスを表すデータに関連付けられた第2のオブジェクトを生成し、

前記実行エンジンモジュールが、前記第2のオブジェクトに含まれる前記第1のオブジ

エクトに対するリンクを生成し、

前記実行エンジンモジュールが、前記第 1 のオブジェクトに含まれる前記第 2 のオブジェクトに対するリンクを生成し、生成した前記第 2 のオブジェクトに対するリンクが、生成した前記第 1 のオブジェクトに対するリンクと異なり、

決定した少なくとも一つの第 1 の関数を、生成した第 1 のオブジェクトの構造及び関連付けられたデータとは独立して決定する方法。

【請求項 2】

前記少なくとも一つの情報は、前記少なくとも一つの第 1 の関数の入力として又は出力として使用されるデータに関連付けられる、少なくとも一つのオブジェクトに対する参照である請求項 1 に記載の方法。

10

【請求項 3】

前記少なくとも一つの第 1 の関数は、前記共同サービス指向ワークショップに対する外部関数である請求項 1 又は 2 に記載の方法。

【請求項 4】

前記実行エンジンモジュールが、前記少なくとも一つの第 1 の関数とは異なる、少なくとも一つの第 2 の関数を決定するステップと、

前記実行エンジンモジュールが、前記少なくとも一つの第 2 の関数を実行する、少なくとも一つの情報を決定するステップと、

前記実行エンジンモジュールが、前記少なくとも一つの第 2 の関数及び前記少なくとも一つの第 2 の関数の実行を可能にする前記少なくとも一つの情報に、少なくとも一つの参照を持つ作成された前記第 1 のオブジェクトを追加するステップと、を更に有する請求項 1 乃至 3 のいずれか一項に記載の方法。

20

【請求項 5】

前記実行エンジンモジュールが、前記第 1 のオブジェクトを、前記共同サービス指向ワークショップに対する外部モジュールにより実行可能な関数に変換するステップを更に有する請求項 1 乃至 4 のいずれか一項に記載の方法。

【請求項 6】

プロセスモデルの実行の環境が、前記共同サービス指向ワークショップにより実行されるプロセスモデル及び外部モジュールにより実行可能な関数に対する実行において同一である請求項 1 乃至 5 のいずれか一項に記載の方法。

30

【請求項 7】

前記実行エンジンモジュールが、外部モジュールにより実行可能な関数及び外部モジュールにより実行可能な前記関数の実行に必要な少なくとも一つの情報に対する少なくとも一つの参照を含むオブジェクト作成するステップを更に有する請求項 5 又は 6 に記載の方法。

【請求項 8】

前記少なくとも一つの第 1 の関数は、タスク実行又はリンクエンジンに合致している請求項 1 乃至 7 のいずれか一項に記載の方法。

【請求項 9】

請求項 1 乃至 8 のいずれか一項に記載の方法の各ステップを行うように適合した命令を有するコンピュータプログラム。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、共同サービス指向ワークショップのための情報処理アーキテクチャに関する。特に、共同サービス指向ワークショップ(atelier)のためのデータ管理方法に関する。

【背景技術】

【0002】

例えば、開発のためのコスト及び時間の削減のために、車両(vehicle)等の対象物の研究段階及び構想段階を最適化するには、一般に、特有な情報処理環境の実現が必要になる

50

。

【0003】

例えば、航空機の航空力学の研究には、多くの専門家の参加を必要とする。同一データを多くの側面から研究するからである。これらの側面は、相互に関係があったり、なかったりするものである。上記専門家は、一般に、種々の情報処理ツール及び種々の装置を用いることによって、利用する各プロセスを実行する。更に、これらの専門家は、地理的に離れた場所にいることがある。

【発明の概要】

【発明が解決しようとする課題】

【0004】

したがって、明らかに、情報処理データ及びリソース（特に計算機リソース）を共有することができるオープンアーキテクチャに基づく環境を利用することが有利である。このような環境は、異種の装置及びオペレーティングシステムのセットに適合していることが好ましい。

【0005】

同様に、別々の種々の既存ツールにアクセスできるインターフェースを使うことが有利である。このような共通インターフェースは、特に、これらのツール間で交換されるデータを管理し、これらのデータに容易にアクセスし選択できるよう適合する必要がある。

【0006】

更に、前記環境は、新規ツールの統合、新規タイプのデータの管理、新規タイプの結果の生成などの将来の需要に簡単に適合できることが好ましい。

【0007】

更に、計算又は他のプロセスにおいて、種々の変更及び使用の過程で生成されたデータの起源（origine）及びライフサイクルを決定するデータトレーサビリティ（tracabilite'）が必要である。

【0008】

これらの課題は、部分的に、相互に独立して解決されている。

【0009】

例えば、Fujitsu（登録商標）により開発されたSynfiniWay（登録商標）というソフトウェアアプリケーションが存在する。このアプリケーションは、異種環境におけるタスクの実行の最適化を可能にするものである。アプリケーションSynfiniWayによって、特に、データのシーケンス及び依存関係により関連するサービスの形式で、グローバル資源を仮想化しアプリケーションを提供することによって、情報処理プロセスを自動化することができる。

【0010】

更に、一定のタスクを自動化し既存のツールを呼び出すことができるPython（登録商標）及びJava（登録商標）等のスクリプト言語が存在する。Pythonは、インタプリタでオブジェクト指向（interprete et oriente objet）の高級レベルのプログラミング言語である。Pythonは、専用ライブラリを使って多数のコンテキストで使用したり多数の使用法に適合したりすることができる。

【0011】

データのトレーサビリティに関しては、一般に、データに至ったプロセスにより決定される。例えば、プロダクトデータ管理（Product Data Management：PDM）タイプのアプリケーションにおいては、ユーザは、予め定められたプロセスに、違反することなく、従わなければならない。したがって、データの起源は、データを取得することができたプロセスにより決定される。

【0012】

しかし、異種のオープン共同環境における、システムの研究及び構想のための集中化及び最適化された環境は、存在しない。

【0013】

10

20

30

40

50

本発明は上記の課題の少なくとも一つを解決するためのものである。

【課題を解決するための手段】

【0014】

本発明は、実データ及びプロセスを表わすデータに関連付けられたオブジェクトを処理するように構成された共同サービス指向ワークショップにおけるデータ管理方法を対象とする。

各オブジェクトは他の対象に対するリンクを記憶する構成を有しており、前記方法は次のステップを含む。

- 少なくとも一つの関数を決定するステップ。
- 前記少なくとも一つの関数の実行を可能にする、少なくとも一つの情報を決定するステップ。
- 前記少なくとも一つの関数及び前記少なくとも一つの情報に対する、少なくとも一つの参照(reference)を持つ一つのオブジェクトを生成するステップ。

10

【0015】

本発明の方法によって、データの性質、格納場所及び大きさ等の固有の特徴とは独立してデータを管理することによって、データのグローバル管理を行うことができる。

【0016】

前記少なくとも一つの情報は、前記少なくとも一つの関数の入力として又は出力として使用されるデータに関連付けられる、少なくとも一つのオブジェクトに対する参照(reference)である。

20

【0017】

本発明による方法によれば、オブジェクトにより表わされるプロセスを実行し、‘実行方法に関連するオブジェクト’と‘前記プロセスにより実行される実データに関連付けられるオブジェクト’(即ち、プロセスにより使用される、又は、生成される)の間のリンクを確立することができる。

【0018】

一つの特定の実施例によれば、前記少なくとも一つの関数は、前記共同サービス指向のワークショップに対する外部関数で、既存の関数を使用し、ソフトウェア開発を制限するためのものである。本発明による方法を使うと、ソフトウェアモジュールの類似した複数のバージョンのメンテナンスを回避することができる。

30

【0019】

また、一つの特定の実施例によれば、本方法は、次のステップを含む。

- 前記関数(第1関数)とは異なる、少なくとも一つの第2関数を決めるステップ。
 - 前記少なくとも一つの第2関数を実行する、少なくとも一つの情報を決めるステップ。
- 前記少なくとも一つの外部第2関数及び少なくとも一つの情報に、少なくとも一つの参照(reference)を持つオブジェクトを追加するステップ。

【0020】

本発明の方法を使うと、複合プロセスに関連するオブジェクトを作成することができる。つまり、関数の組み合わせに関連するオブジェクトである。

【0021】

ある特定の実施例では、この方法は、さらに、前記共同サービス指向ワークショップにおける外部モジュールにより実行可能な関数に、オブジェクトを変換する段階を含んでいる。前記共同サービス指向ワークショップにより、対応するコードを再書き込みせず、共同サービス指向ワークショップの外部関数を使うことができる。

40

【0022】

プロセスモデルのランタイム環境(environnement d'execution)が、関数実行の均一性(homogeneite d'execution des fonctions)を提供するために、前記共同サービス指向ワークショップにより実行されるプロセスモデルに関して、及び、外部モジュールにより実行可能な関数に関して、類似していると有利である。

【0023】

50

1つの特定の実施例では、この方法は、さらに、“その外部モジュールで実行可能な関数に対する”及び“外部モジュールにより実行可能な前記関数の実行に必要な少なくとも一つの情報に対する”少なくとも一つの参照を含むオブジェクト作成するステップを含む。

【0024】

前記プロセスの生成は、本実行方法により作成される及び/又は使用されるデータのトレーサビリティに望ましい粒度 (granularity) のレベルによりコントロールされる。

【0025】

前記外部関数は、1つのランタイム又は連鎖するタスクに合致していると有利である。これにより、前記外部関数の実行と標準ツールのコンパチビリティが可能になる。

10

【0026】

また、本発明は、上記方法の各ステップを実現するよう適合した命令を有するコンピュータプログラムを対象とする。

【0027】

本発明の、他の有利な点、目的及び特徴は、次の図面を参照して、非制限的な例を示す、以下の詳細な説明から明らかである。

【図面の簡単な説明】

【0028】

【図1】本発明に係るデータ及びリソースを共有するためのワークスペースのアーキテクチャの一例を示す図である。

20

【図2a】図1に示されるデータマネージャの簡略化した機能概略図である。

【図2b】データツリーを確立することができるアルゴリズムの一例である。

【図3】Pythonタイプの表現に基づいてプロセスを実行するための、図1に示される実行エンジンの機能要素と、これらの機能要素間で転送される情報とを示す概略図である。

【図4】アプリケーションの視点から見た、外部開発部 (developpements externes) 及び特有開発部 (developpements specifiques) によって制作された実行可能コードを実行することができる、本発明に係る共同サービス指向ワークショップを示す図である。

【図5】プロセスモデルにおける、ModelCenterモデルタイプの外部関数のカプセル化の一例を示す図である。

【図6】XMLタイプの記述形式で記憶されるプロセスモデルの分析のためのXMLタイプの概略図の一例を合成的な手法で示した図である。

30

【図7】インスタンス化されたプロセスモデルの単位プロセスの実行のためのアルゴリズムの一例である。

【図8】本発明の共同サービス指向ワークショップから実行できるデータ処理の一例である。

【図9】本発明の一部を利用するよう適合した装置の一例である。

【発明を実施するための形態】

【0029】

図1は、本発明による、データ及びリソースを共有するためのワークスペースのアーキテクチャ100の一例を概略的に示す。これを共同サービス指向ワークショップとも呼ぶ。図示のように、該アーキテクチャは、次の4つの主なモジュールを中心として実施される。すなわち、データにアクセスするためのモジュール105と、データマネージャであるモジュール110と、実行エンジンであるモジュール115と、プロセスを管理するためのモジュール120と、である。

40

【0030】

データアクセスモジュール105は、データ管理モジュール110及び中央データベース125 (例えば、Oracle (登録商標) タイプのデータベース) に接続している。データ管理モジュール110は、中央データベース125、実行エンジン115、及び分散格納ゾーンを表すセット130に接続している。このセットは、例えば、ファイル形式又はデータベース形式など種々に編成することができる。実行エンジン115自体は、格納ゾーンセット130及びプロセ

50

ス管理モジュール120に接続している。

【0031】

このアーキテクチャは、航空機などの特に複雑な対象物の構想に適合するものであるが、以下では、明瞭かつ簡潔にするため単純な例に基づいて説明する。この例は例示として示すものである。

【0032】

アーキテクチャ100は、広義の意味のデータを表わす情報処理オブジェクトを操作するよう適合している。これらのデータは、例えば、航空機などの実オブジェクト（現実の対象物（objet re'el））又は航空力学の抵抗の計算プロセスなどのプロセスを表わすことができる。簡単に言えば、共同サービス指向ワークショップは、自体が特徴データを表すメタデータに基づく情報処理オブジェクトにより表されるデータ上で動作する。

10

【0033】

フレームワーク データもしくはFDと呼ばれるデータは、例えば、ファイル形式で格納ゾーン130に記憶された物理データである。これらのデータは、例えば、椅子のグリッド（maillage）もしくは構造などの実オブジェクト、又は、例えば、引越しプロセスの各ステップなどのプロセスを特徴付ける。これらのデータの大きさは、制限されず、数百メガバイトでもよい。使用される全てのデータは、別々の物理位置に格納されてもよい。

【0034】

フレームオブジェクトもしくはFOと呼ばれるオブジェクトは、実際に適用される意味において特定の意義を持つオブジェクトを表す。前に示したように、これらのオブジェクトは、共同サービス指向ワークショップにより操作することができる全てを表すのに用いられる。これらのオブジェクトは、基礎的要素又は特徴的要素として含まれているメタデータセット（ensemble de meta-donnees）に基づく。したがって、1つのメタデータセットは、各オブジェクトに関連付けられる。各タイプのメタデータセットは、複数のオブジェクトに関連付けられてもよい。1つのタイプのメタデータセットの性質の特定（qualification）により、1つのオブジェクトを形成することができる。例えば、あるタイプのメタデータセットが椅子に関連付けられている場合、オブジェクトを形成する椅子の性質の特定は、子供椅子でもよい。

20

【0035】

明瞭にするために、ここでは、フレームワークオブジェクトプロキシ（framework object proxies）又はFOプロキシ（FO proxies）と呼ばれるオブジェクトの情報処理での実装を、オブジェクト自体と同等に扱う（assimilie）。

30

【0036】

オブジェクトの中には、インスタンス化プロセステンプレート（instanciated process templates：IPT）と呼ばれるインスタンス化されたプロセスモデルを象徴するものがある。該インスタンス化プロセスモデルは、プロセスモデルのインスタンス、即ち、1つ又は複数のオブジェクトに適用されたプロセスモデルである。プロセステンプレート（process templates：PT）と呼ばれるプロセスモデルは、一般的なものである。これらは、例えば、複数の椅子と1つのテーブル（table）からなるリビングルームの構築プロセス、家のリフォームプロセス、又は、引越しプロセスに関する。インスタンス化プロセスモデルは、オブジェクト形式で表され、共同サービス指向ワークショップでそれ自体として操作することができる。

40

【0037】

フレームワークメタデータ（framework meta-data：FMD）と呼ばれているメタデータは、データの特定情報に対応している。メタデータによってデータの特徴を表すことによって、データの操作を容易にすることができる。1つのデータは複数のメタデータセットに関連付けられてもよい。メタデータは、表（tableau）又は拡張可能マークアップ言語（Extensible Markup Language：XML）ファイルの形式で、中央データベース125に記憶されると有利である。ここでは、1つのタイプは、各メタデータセットに関連付けられる。メタデータは、識別子、作成日時及び作成者名などの一般情報と、例えば、椅子の材

50

料、色及び重量などの関連データに固有の特有情報と、を含んでいる。メタデータは、特に、前記データがどのようにして取得されたか、及び/又は、データがどのように使われたかを決定することのできる、オブジェクトへのリンクを含む。これらのリンクによって、データモデル、即ち、データのツリー構造を定義することができる。

【0038】

オブジェクトの整合性 (consistance) は、オブジェクトに関連付けられているメタデータセットにおいて定義されたリンクを使って検証される。例えば、テーブルのデータモデルは、一つのテーブルが一つの椅子に関連する、などのように決定することができる。実際にテーブルタイプのオブジェクトが椅子タイプのオブジェクトに関連付けられている場合は、テーブルタイプのオブジェクトは、これらのリンクを介して定義されたデータモデルの意味において整合している (consistant)といわれる。

10

【0039】

メタデータは、例えば、本書付属の「extrait de code 1 (コード1の抜粋)」で参照される構造に従って、XMLタイプの言語で記述される。

【0040】

メタデータの一般的な部分は、全てのデータに共通しているので、共同サービス指向ワークショップを実現するときに定義される。たとえ、この部分が先験的に変更する理由がない場合であっても、その定義は必要性に基づいて改訂することができる。しかしながら、この改訂は、共同サービス指向ワークショップで管理されるオブジェクトセットに影響するものである。

20

【0041】

メタデータの特有な部分は、各タイプのデータに固有であり、その結果、各タイプのメタデータセットに固有である。この部分の定義は、共同サービス指向ワークショップのユーザにより実施される。メタデータの特有な部分の情報は、データに到達する前に自動抽出 (extraction) オペレーションによって決定される。

【0042】

一般メタデータ及び特有メタデータは、共同サービス指向ワークショップにおけるデータの検索及び分類に使用されることが有利である。

【0043】

本書付属の「extrait de code 2 (コード2の抜粋)」で参照される例は、椅子を記述するために使用される、メタデータタイプのXMLタイプの記述を示す。

30

【0044】

前述の通り、メタデータは、一般的な部分及び特有な部分を含むことが好ましい。ここでは、特に、一般的な部分は以下の要素を含む。

- データへのアクセスコントロールを管理するキー。
- オブジェクトタイプの記述。
- データを記述するコメント。
- 作成日時。
- データへのアクセスを可能にするキー又はURLでもよいデータの識別子。
- データの一意の識別子。
- データを作成したユーザに関連する情報。
- 地理的位置を有するデータの物理位置。
- 変更が進行中であるかどうか、及び誰により行われているかを知ることができる状態 (son etat)。

40

【0045】

操作されるオブジェクト (ここでは椅子) に特有な、メタデータの第2の部分は、本例において以下の属性を有する。

- 足の数。
- 厚さ。
- 色。

50

- 材料。
- 重量。

【0046】

一般的な部分及び特有な部分は、当然、他のタイプの情報を含むことができる。

【0047】

メタデータセットの識別子又はアドレスによって、それに関連付けられているデータを探し出すことができる。

【0048】

椅子を表すメタデータセットの性質の特定 (qualification) は、特に、「子供」又は「成人」でもよく、異なる2つのオブジェクトに帰着してもよい。

10

【0049】

同様に、テーブルのためのメタデータ及びオブジェクトを定義することができる。テーブルを定義するオブジェクトをインポートするときに、データモデルのレベルで、このオブジェクトは、少なくとも1つの椅子タイプのオブジェクトに関連している場合しか整合 (consistant) しないと定義してもよい。オブジェクト間のこのような関係は、メタデータにおいて定義されたリンクにより定義される。これらのリンクは、XMLフォーマット又は表 (tableau) で表すことができる。テーブルタイプのオブジェクトをインポートするときに、テーブルタイプのオブジェクトと椅子タイプのオブジェクトとの間のデータモデルタイプのリンクを配置することによって、このオブジェクトをこのように定義されるデータモデルと整合 (consistant) させなければならない。

20

【0050】

代替として、メタデータは表形式で表すことができる。例えば、1つの表は、各タイプのデータに関連づけることができる。表の各列は、メタデータを表す一方、各行はオブジェクトを表している。本書の付属資料の表1は、椅子タイプのデータに関連付けることのできる1つのタイプのメタデータセットの表現を簡略化した例を示す。ここでは、リンクはメタデータを表す表に含まれていないことに留意すべきである。この例においては、リンクはリンクテーブル (table de liens) に記憶されている。

【0051】

リンクテーブルは、例えば、本書付属の表2で示す表形式で表すことができる。この表において、各列は、リンクが確立されているオブジェクトの一对の識別子を表している。

30

【0052】

このように、メタデータの表及びリンクテーブルに記憶されている情報によって、現在であろうと将来であろうと、共同サービス指向ワークショップの全てのデータを操作し、データモデルを構築することが可能になる。

【0053】

ここで、メタデータの表形式の表現は、XMLの表現と等価であること、及び、変換ツールを用いて、1つの表現から別の表現に移行できることに留意すべきである。

【0054】

物理オブジェクト又はプロセスを特徴付けるデータの表現は、複数のレベルで実施される。

40

- 生データ (donnees brutes)。
- 生データから抽出された生データのサブセット (部分集合 (sous-ensembles)) を表わし、生データの特徴づけることができるメタデータ。
- メタデータの値に基づき、アプリケーションモジュールにより操作されるオブジェクト。

- 別のオブジェクトにリンクされたオブジェクトであり整合性をもつ (consistant) オブジェクト。

【0055】

この構造によると、共同サービス指向のワークショップにおいて、同期プロセスを利用する必要がないことに留意すべきである。

50

【 0 0 5 6 】

データアクセスモジュール105によって、特に、ユーザは、選択的及び集中的に、中央データベース125に記憶されているメタデータにアクセスすることによって、特に、これらのメタデータを視覚化したり、これらのメタデータに関するビュー（vue）を組み立てたりすることができる。これにより、共同サービス指向ワークショップにおいてアクセス可能なデータセットに対して効率的にソーティングを行うことができ、また、場合により、これらのデータのうちのいくつかを結ぶリンクを追跡することができる。

【 0 0 5 7 】

また、モジュール105は、中央データベースにメタデータを登録することができる。したがって、ユーザは、マンマシンインターフェースを介してメタデータを入力し、格納ゾーン130に記憶されているデータ中に現われない情報を指定することができる。メタデータにアクセスし記録するために、ユーザインターフェースは、特に、構造化問合せ言語（Structured Query Language : S Q L）タイプの要求(requetes)を使うことができる。このメタデータの手動による記録は、オブジェクトをワークショップにインポートする時にデータからの自動抽出の対象にならない特有情報を追加するのに有利であることは明白である。

【 0 0 5 8 】

同様に、データアクセスモジュール105によって、特に、ユーザは、選択的及び集中的に、データ管理モジュール110を介して、格納ゾーン130に記憶されているデータにアクセスし、データを記録することができる。データにアクセスし記録するために、ユーザインターフェースは、特有なアプリケーションプログラミングインターフェース（Application Programming Interface : A P I）及びwebサービスを使うと有利である。

【 0 0 5 9 】

データマネージャ（gestionnaire de donnees）110は、データアクセスモジュール105、中央データベース125、格納ゾーン130、及び、特にデータにプロセスを適用するよう適合した実行エンジン115間のインターフェースとして使用される。また、データマネージャ110は、ユーザのアクセス権をコントロールすることができる。

【 0 0 6 0 】

特定の実施例において、データマネージャ110は、クライアント部とサーバ部とから構成されている。クライアント部は、Pythonタイプの言語で実装することによって、サーバ部においてアクセス可能なwebサービスにアクセスすることができる。インポート、エクスポート、パブリケーション、コラボレーション及び権利管理などのデータ管理の全ての標準関数（fonction standard）を使うことができることが好ましい。このリストは全部を網羅しているものではない。データのライフサイクルの過程において使用される関数セットは、このAPIを介して使用可能である。

【 0 0 6 1 】

サーバ部は、例えば、Java（登録商標）又はPythonタイプの言語で実装することによって、種々のデータベース及び格納ゾーンとのインターフェースを可能にすることができる。このサーバ部は、分散格納ゾーン130に記憶されているデータから、メタデータを有している中央データベース125へ、メタデータを抽出するメカニズムを有することが有利である。

【 0 0 6 2 】

実行エンジン115は、例えば、専用アプリケーションに基づくプロセスモデルを介してプロセスを実行することができる。プロセスの実行時に、実行エンジン115と格納ゾーン130との間のインターフェースによって、データへの直接アクセス又はAPIを介したアクセスが可能になる。実行エンジン115は、例えば、SynfiniWayタイプのコンピューティンググリッド（grille de calcul）に接続されることが好ましい。このコンピューティンググリッドは、1つの分散インフラストラクチャ上の種々のプロセスのサブミット及び実行中のデータの移動を管理する。

【 0 0 6 3 】

10

20

30

40

50

又、実行エンジン115は、ModelCenter（登録商標）又はSynfiniWayアプリケーション（つまり、タスク実行又はリンクエンジン）などの、ワークフロー（workflow）を使うアプリケーションとのインターフェースを有することが好ましい。

【0064】

プロセス管理モジュール120は、複合プロセスと呼ばれ、複数の単位処理（traitements unitaires）をカプセル化した（encapsuler）ハイレベルのワークフローを作成することができる。これらのワークフローは、パラメータ及びパラメータに対するセンシビリティの調査を対象にして動作モデルを作成することができる。このため、これらの機能は、最適化の多分野に関わる調査の枠内で使うことができる。

【0065】

したがって、非単位プロセス（processus non unitaires）処理、つまり、種々の複数の関数を呼び出すプロセス処理は、プロセス管理モジュールにより管理されることによって、例えば、実行エンジンにより実行される単位プロセスに分割される。

【0066】

図2 aは、データマネージャ110の簡略化した機能概略図（schema fonctionnel）である。

【0067】

例えば、ファイルインタフェース又は共通オブジェクトリクエストブローカーアーキテクチャ（Common Object Request Broker Architecture : Corba）（登録商標）タイプのインターフェースを使って、データが種々のデータベース（図示しない）において作成される。

【0068】

次に、抽出、変換及びロードのメカニズム200（ETL（Extract, Transform and Load））を利用することによって、格納ゾーン130に記憶されているデータからメタデータを抽出する。これらは中央データベース125に格納される。こうして、このデータベースによって、種々の分散格納ゾーン130において作成された物理データの情報が収集される。

【0069】

データが作成されたり、変更されたり、又は削除されたりするときに、抽出メカニズムが利用されることに留意すべきである。このメカニズムは、データ及びメタデータの各タイプに固有のものである。

【0070】

性質特定関数（fonction de qualification）205により、抽出されるメタデータの性質を特定することによって、中央データベース125に記憶することができるオブジェクトを形成することができる。性質の特定に続いて、オブジェクトの整合性（consistance）について検証が行われる。この整合性は、メタデータと共に記憶されている又は記憶されていないことがあるリンクを介して保証される。これらのリンクは、バージョン、コンフィギュレーション、データモデル及びユーザなどの種々の性質に関することがある。

【0071】

データモデルを定義するオブジェクト間のリンクセットは、複数のオブジェクト間のリンクを決定するために使用されてもよい。これらのリンクは、メタデータに関連付けられる。上記したように、それらは、後者と共に又は別々に、例えば、中央データベース125に記憶されているリンクテーブル（table de liens）に、記憶される。

【0072】

前記リンクによって、後で、データのトレーサビリティ、即ち、プロセスの実行時に生成又は使用された種々のデータ間のトレーサビリティを、確立することができる。複数のタイプのリンクがあるが、特に、次のようなものである。

【0073】

- ユーザが手動で決定する、2つのデータ間のユーザリンク。
- データモデルを定義することができるモデルリンク。これらのリンクは、データモデルの作成のときに、一般的に決定される。
- 1セットのオブジェクト間でリンクを確立することによって、オブジェクトをコンフィ

10

20

30

40

50

ギューレーションにおいて管理、即ち、エクスポート、インポート及び権利変更の関数等の同一関数を適用することができるコンフィギュレーションリンク。

- 種々のバージョンに従って、同一のオブジェクトを管理することができるバージョンリンク。

- 1セットのオブジェクト間における実行リンクの確立を可能にする生成 (production) リンク。インスタンス化されたプロセス (I P T) の実行中又は終了時のデータパブリケーション (publication) 段階において、このタイプのリンクは自動的に作成される。

【 0 0 7 4 】

対で配置されるこれらのリンクのセットによって、データに伴うトレーサビリティを得ることができる。これらのデータには、特に、プロセス、バージョン、コンフィギュレーション、ユーザ及びモデルに関する全ての履歴が含まれる。

10

【 0 0 7 5 】

こうして、特異なメタデータ及びリンクにより相互に関連付するオブジェクトの完全なツリーを構築することが可能である。椅子とテーブルの例では、テーブルのメタデータ (色及び位置) を知得することができるツリーによって、メタデータとして特異な色を有する椅子にテーブルをリンクするデータモデルを検証することができる。したがって、次から次へと、データ取得のための完全なツリーを再構築することができる。

【 0 0 7 6 】

データモデル、即ち、データの階層構造 (arborescences) は、共同サービス指向ワークショップにおいて操作されるデータに特有であることに留意することが必要である。しかし、仮にデータモデルがオブジェクト間のリンクから導くことができたととしても、データモデルは、共同サービス指向ワークショップ中にそれ自体として存在しない。それらは、オブジェクト、即ち、特定のデータタイプに対応するメタデータセットが定義されるときに、暗黙的に作成される。

20

【 0 0 7 7 】

図 2 a に示されているように、データマネージャ110の関数210によって、リンク及びオブジェクトにアクセスすることによって、整合性のあるオブジェクトのビュー (vue)、即ち、オブジェクト間の関係を表わすビューを作成することができる。このビューは、ユーザにデータモデル215のビュー形式で送信することができる。

【 0 0 7 8 】

同様に、ユーザは、例えばリストなどのオブジェクトビュー220及び関係ビュー225の形式で、オブジェクト及び関係を可視化することができる。

30

【 0 0 7 9 】

したがって、この実施例によれば、ユーザ、即ち、データマネージャ110のクライアント部は、次の3つのタイプのビュー (vue) を配置することができる。

- 種々のデータベースが有するオブジェクトに関するメタデータを介したビュー。
 - アーキテクチャ100において作成されたデータモデルに関するビュー。
 - 例えば、トレーサビリティを可能にするバージョンリンク、ユーザリンク、コンフィギュレーションリンク及び生成リンクなどのメタデータのデータベースで定義されているリンク、又は、それらに関連付けられたリンクを介して、種々のオブジェクト間をナビゲートすることができるビュー。

40

【 0 0 8 0 】

各データに関連付けられた全てのメタデータを含む、プロキシオブジェクト (object proxy) タイプのオブジェクトのXMLタイプの記述が、作成されて、プロセスの実行中に、例えば、実行エンジン115により、格納ゾーン130に記憶されたデータにアクセスするのに使用されると有利である。

【 0 0 8 1 】

図 2 b は、アルゴリズムの例を示す。このアルゴリズムによって、共同サービス指向ワークショップで操作されるオブジェクトからデータツリーを作成することができる。これらのオブジェクトは実データ又はプロセスを区別せずに表わす。

50

【 0 0 8 2 】

例えば、マンマシン インターフェースを介して第 1 オブジェクトを選択した（ステップ230）後、該第 1 オブジェクトに含まれる、第 2 オブジェクトへのリンクにアクセスする（ステップ235）。次に、該第 2 オブジェクトに含まれる第 3 オブジェクトへのリンクにアクセスする（ステップ240）ことによって、第 1 オブジェクトと第 3 オブジェクトとの間のリンク、又は、第 1 オブジェクトに関連付けられたデータと第 3 オブジェクトに関連付けられたデータとの間リンクを確立することができる（ステップ245）。

【 0 0 8 3 】

他の例として、該第 1 オブジェクトに含まれる第 3 オブジェクトへのリンクにアクセスする（ステップ250）ことによって、第 2 オブジェクトと第 3 オブジェクトとの間のリンク、又は、第 2 オブジェクトに関連付けられたデータと第 3 オブジェクトに関連付けられたデータとの間のリンクを確立することができる（ステップ255）。

10

【 0 0 8 4 】

第 2 b 図において、破線の矢印で示されるように、これらのステップを反復して繰り返すことによって、データツリーを決定することができる。該データツリーにおいては、葉はデータ（又はデータに関連付けられたオブジェクト）を表わし、枝はデータ間のリンク（またはデータに関連付けられたオブジェクト間のリンク）を表わす。

【 0 0 8 5 】

図 3 は、図 1 に示した実行エンジン115の機能要素と、これらの要素間で、Pythonタイプの表現に基づいてプロセスを実行するために送られる情報と、を概略的に示す図である。実行エンジン115は、ここではXMLタイプの文書の形式で表わされた、インスタンス化されたプロセスモデル300で構成される（consistent）オブジェクトによりアクティブになる。

20

【 0 0 8 6 】

インスタンス化プロセスモデル300を、XMLパーサ（XML parser）と呼ばれるXMLタイプのシンタックスアナライザ305において解析することによって、プロセスの実行を可能にするための情報を抽出する。特に、プロセスの実行に必要なデータに関するパラメータ、コード及び情報が、XML文書から抽出される。同様に、該プロセスで使われプロセスの入力を形成するデータを、関連オブジェクト上のリンクを使って探し出すことができる。XML文書から抽出したプロセスの実行に必要な全ての情報は、実行エンジンのコア310に送信される。

30

【 0 0 8 7 】

インスタンス化プロセスモデル300のコード、即ち、インスタンス化プロセスモデルを表わすオブジェクトに関連付けられたデータは、実行エンジンにより直接実行可能な命令、例えばPythonタイプのコードなどの特有なインターフェースを介して実行可能な命令、又は、ModelCenter関数等の実行エンジンにより呼び出される外部関数又はモジュールに対応する。

【 0 0 8 8 】

インスタンス化プロセスモデル300に関連付けられたプロセスを実行することによって、プロセスの出力を形成する新規オブジェクト315が生成されることがある。これらのオブジェクトは、前記オブジェクトと同様に扱われる。特に、中央データベースに記憶されているメタデータ及びリンク並びに格納ゾーン130に格納されているデータは、前記オブジェクトに関連づけられている。代替として又は補完的に、プロセスの入力は、これらが実行されている間に、変更されてもよい。変更されプロセスの出力を形成する入力は、入力/出力と呼ばれる。

40

【 0 0 8 9 】

実行エンジンのコア310は、インスタンス化プロセスモデルの実行コンテキストとも呼ばれる実行環境320を呼び出すことが好ましい。特に、この実行環境は、プロセスの実行履歴（325）を保持し、データを実行エンジンのフォーマット（例えば、Pythonタイプのオブジェクトの形式）でアクセス可能にし（330）、プロセスの実行を監視し（335）、外

50

部モジュールの実行に必要なインターフェース340を実行エンジンのコアに提供するためのものである。

【0090】

プロセスが実行される時、特に、新規オブジェクトが作成される時、1つ又は複数のリンクが、プロセス(入力、出力、入力/出力)で使用されるデータに関連付けられたメタデータ、並びに、インスタンス化プロセスモデルに関連付けられたメタデータにおいて、自動的に作成される。

【0091】

プロセスが実行される前、インスタンス化プロセスモデルに関連付けられたメタデータは、使用される前のデータに関連付けられたオブジェクトへの参照(reference)を含むが、リンクは含まない。したがって、リンクは、プロセスが実行されるときに、各オブジェクトにおいて作成される。

10

【0092】

例えば、プロセスがデータを使用することによって結果を生成するとき、プロセスを実行する過程で、次のリンクが作成される。

- プロセスモデルに関連付けられたオブジェクトへのリンクが、入力データに関連付けられたオブジェクトにおいて、作成される。

- 入力データに関連付けられたオブジェクトへのリンクが、プロセスモデルに関連付けられたオブジェクトにおいて、作成される。

- 出力データに関連付けられたオブジェクトへのリンクが、プロセスモデルに関連付けられたオブジェクトにおいて、作成される。

- プロセスモデルに関連付けられたオブジェクトへのリンクが、出力データに関連付けられたオブジェクトにおいて、作成される。

20

【0093】

図4は、本発明による共同サービス指向ワークショップの実施例をアプリケーションの観点から示す。この実施例によって、外部開発部(developpements externes;405)及び特有開発部(410)によって制作された実行可能コード400を実行することができる。図示されているように、インスタンス化プロセスモデルの実行は、再帰可能である。

【0094】

実行可能コードの性質と共同サービス指向ワークショップの実装に従って、実行可能コードは、共同サービス指向ワークショップ又は外部アプリケーションにより実行される。説明として、ここでは、実行可能コードは、Pythonタイプの場合、共同サービス指向ワークショップにより実行され、例えば、ModelCenter(MC)タイプの場合、外部アプリケーションにより実行される。

30

【0095】

プロセスモデルにおける外部関数のカプセル化メカニズムに従って、共同サービス指向ワークショップの外部関数が実行される。図5は、プロセスモデルにおけるModelCenterモデルのカプセル化(encapsulation)の一例を示す。

【0096】

実行可能コードがPythonタイプの場合、特に入力及び出力データへの参照並びに実行環境への参照を有する、コードの実行に必要なデータは、Pythonフォーマット(モジュール415)に従ってフォーマットされる。そして、こうしてフォーマット化されたプロセスは、共同サービス指向ワークショップが標準オブジェクト(モジュール425)として操作できるプロセスモデルにおいてカプセル化される(モジュール420)。モジュール425をモジュール415に連結する矢印で図示されているように、プロセスを実行しなければならないとき、プロセスモデルはPythonインターフェースを使う。

40

【0097】

同様に、実行可能コードがModelCenterモデルの場合、特に入力及び出力データへの参照並びに実行環境への参照を有する、コードの実行に必要なデータは、ModelCenterフォーマット(モジュール430)に従ってカプセル化される。そして、こうしてフォーマット

50

化されたプロセスは、共同サービス指向ワークショップが標準オブジェクト（モジュール425）として操作できるプロセスモデルにおいてカプセル化される（モジュール435）。また、モジュール425をモジュール430に連結する矢で図示されているように、共同サービス指向ワークショップの外部プロセスの実行を可能にするためプロセスを実行しなければならないとき、プロセスモデルは、ここではModelCenterインターフェースなどの適合するインターフェースを使う。

【0098】

参照符号460で示されるように、別のフォーマットが使用されてもよい。

【0099】

プロセスモデルのインスタンス化は、マンマシンインターフェース、即ち、ここでは、モジュール420と425との間及びモジュール435と425との間におけるインターフェースを使って、プロセスのカプセル化の後に実行することができる。

10

【0100】

特に、カプセル化は、予め定められたフォーマットに従ってプロセスモデルを適合させることを目的とする。具体的には、カプセル化により、データの有効性をコントロールし、必要なパラメータを決定し、欠落しているパラメータにデフォルト値を与え、適切なコードをコピー及び/又は創出することができる。また、オプションとして、カプセル化により、実行指令（directives d'execution）の定義、即ち、例えば、プロセス又はプロセスの一部を実行すべきマシンが何であるかを定義することができる。

【0101】

20

プロセスモデルにおける外部関数のカプセル化は、次のXMLセクションを使って、実施されることが好ましい。

【0102】

- 入力及び出力。共同サービス指向ワークショップにおいて管理されるデータに関連付けられたオブジェクトへのリンク。これらのリンクによって、外部関数において上記データを使用することができる。

- パラメータ。外部関数により使用されるパラメータ。

- コード。外部関数の呼び出し。

【0103】

図5は、プロセスモデル505におけるModelCenterモデル500のカプセル化の例を示す。図に示されているように、リンクが、プロセスモデルの各セクションとModelCenterモデルの対応するフィールドとの間で確立される。ここでは、ModelCenterモデルの実行可能コードへの呼び出しが、Pythonスクリプト510を介して実行される。

30

【0104】

したがって、プロセスモデルは、共同サービス指向ワークショップにおいて外部関数を呼び出すことができる。プロセスモデルに関連付けられたオブジェクト及び特にリンクの構造によって、外部関数を呼び出すプロセスモデルの入力又は出力として使用されるデータのトレーサビリティが可能になる。

【0105】

同様に、例えばModelCenterモデル等の外部関数において、プロセスモデルをカプセル化することも可能である。したがって、共同サービス指向ワークショップで開発されたプロセスモデルは、外部アプリケーションが直接使用することができる。このメカニズムを介して、実行時に、サービス指向共同ワークショップにおいて管理されるデータの使用及び関連付けられたトレーサビリティを全て維持しながら、プロセスモデルを使用し他の外部関数へリンクすることが可能である。

40

【0106】

外部アプリケーションから開始するプロセスモデルの実行は、SRUNと呼ばれ、インスタンス化プロセスモデルの実行に使用される、共同サービス指向ワークショップのインテグレーション層のニュートラル実行モジュールにより実施される。

【0107】

50

上記方法で、プロセスモデルは、外部関数をカプセル化することができ、また、外部関数においてプロセスモデルをカプセル化することができる。

【0108】

このカプセル化二重メカニズムにおいては、データのトレーサビリティの程度は、このように、共同サービス指向ワークショップ及び外部アプリケーションによって実行される関数 (fonctions) によって決定される。

【0109】

説明として、共同サービス指向ワークショップのプロセスモデルAが、関数B、C及びDを呼び出す場合、但し、B及びCは、共同サービス指向ワークショップのプロセスモデルであり、Dは、自体が関数E及びFを呼び出すModelCenterモデルであり、Eは外部関数であり、Fは、共同サービス指向ワークショップのカプセル化されたプロセスモデルである場合、B、C及びD間のリンクを追跡することはできるが、共同サービス指向ワークショップの外部で実行されるE及びF間 (簡単にはEの入力とFの出力との間) のリンクを追跡することはできない。したがって、データのトレーサビリティについて望まれる精度レベルを決定することができる。

【0110】

前述したように、オブジェクトに関連付けられているデータは、インスタンス化されたプロセスモデルを特徴づけるものであり、例えば、XMLタイプのファイルなどのファイルとして表現することができる。プロセスモデルの記述は、各タイプの情報に特有な複数のセクションを有することが有利である。

【0111】

プロセスモデルは、自体をサブプロセスモデルで構成することができる。これらのサブプロセスモデルは、プロセスモデルのセクションに記憶することができる。この場合、該サブプロセスモデルは、共同サービス指向ワークショップにおいてアクセス可能なプロセスモデルへの参照である。したがって、複数の単位プロセスモデル又は複合プロセスモデルを参照する複合プロセスモデルを作成することができる。

【0112】

インスタンス化プロセスモデルの属性は、例えば、次のようなものである。

【0113】

- XMLタイプのスキーマ(schema)のバージョン。これにより、プロセスモデルから情報を抽出することができる。
- プロセスモデルの識別子、名前、バージョン及びタイプ。
- インスタンス化プロセスモデルの識別子。
- インスタンス化プロセスモデルがModelCenterタイプの関数を参照している場合の、ModelCenterタイプの実行識別子。
- (サンドボックス(sandboxes)と呼ばれる)セキュリティ保護された方法でプロセスを実行することができるメカニズムへのアクセスパス。
- インスタンス化プロセスモデルが、例えば、サブミット済み、実行中、正常に実行完了又はエラーで実行完了などのどの実行状態にあるかを指定する、プロセスモデルの実行状態。
- 例えば、初期状態(primaire)、部分的にインスタンス化済み、インスタンス化済み、実行中又は実行完了等のプロセスモデルの内部状態を定義するプロセスモデルの状態。
- コメント。

【0114】

インスタンス化プロセスモデルの入力、出力、入力/出力及びパラメータに関するセクションには、例えば、名前、タイプ及びカテゴリ等が含まれている。関連するユーザのタイプ、記述及びヘルプなどの他の情報を、インスタンス化プロセスモデルの入力、出力、入力/出力及びパラメータに関連付けることができる。1つのセクションが、各入力、出力、入力/出力及びパラメータに使用される。

【0115】

10

20

30

40

50

インスタンス化プロセスモデルの実行コンテキストに関連するセクションには、インスタンス化プロセスモデルの実行中に使用されるツールを指定する分類セクションが含まれている。又、インスタンス化プロセスモデルの実行コンテキストに関連するセクションには、インスタンス化プロセスモデルの実行コンテキストを記憶することができるコンテキスト作成セクションと、プロセスを実行するために使用するべきコンフィギュレーションを指定し、グリッドコンピューティングの制御装置により実行される選択を記憶することができる現(courant)実行コンテキストセクションとが含まれる。さらに、インスタンス化プロセスモデルの実行コンテキストに関連するセクションには、プロセスモデルのインスタンス化を行う時に選択されるグリッドコンピューティングのコンフィギュレーションを指定するための実行サーバセクションと、プロセスの実行に関連する、例えばエラーコード又はジョブ(metiers)情報等を含み、プロセスの実行の最後に書き込まれる実行レポートセクションとが含まれてもよい。コンテキスト作成及び現(courant)実行コンテキストセクションの属性には、例えば次のものがある。

10

【0116】

- プロセスを実行するために使用されるマシンの名前及びタイプ並びにこのマシンのオペレーティングシステムのタイプ。
- コンテキストの作成日又はプロセスの実行日。

【0117】

インスタンス化プロセスモデルのデータに関連するセクションは、プロセス実行時に使用可能な、プロセスモデルのプライベートデータを記憶するよう用意される。このセクション自体に、ModelCenterタイプのプロセス等の特定のプロセスに特有なセクション等のセクションを含むことができる。

20

【0118】

インスタンス化プロセスモデルのコードに関連するセクションには、実行するプロセスのコードが含まれている。このセクションの属性は、例えば、Python言語などの使用言語、コードのバージョン、及び、場合により、コメントなどである。

【0119】

更に、インスタンス化プロセスモデルには、プロセスを実行するのに使用可能なサーバ及び/又は好ましいサーバのリストの記述が含まれてもよい。

【0120】

図6は、XMLタイプの記述形式で記憶されているプロセスモデルの分析のための、XMLタイプのスキーマ600の一例を示す合成図である。XMLタイプのスキーマ(schema)600によって、プロセスモデルを表わすファイルに含まれる種々のセクションとデータ管理部及びプロセス管理部のXMLタイプのセクションへのリンクとを記述する。

30

【0121】

太線で示されたリンクが、包含(inclusion)を表し、実線のリンクは、インポートの概念及び点線のリンクは、暗黙的なインポートの概念を表わす。

【0122】

プロセステンプレート(Process Template: PT)タイプの拡張子は、プロセスモデル、即ち、実行エンジンに関連付けられている。データ管理(Data Management: DM)タイプの拡張子は、データ、即ち、データマネージャに関連付けられている。MCタイプの拡張は、ModelCenter、即ち、特定のプロセスマネージャに関連付けられている。

40

【0123】

図示されているように、プロセスモデル605には、入力、出力、パラメータ、サーバ、実行コンテキスト、データ及びコードのセクションにそれぞれ対応する複数のサブセット610乃至640が含まれている。入力、出力及びパラメータにリンクされたセクション自体には、引数(arguments)645が含まれる。

【0124】

プロセスモデルには、例えば、EDMG(Editeur De Modele Generalise), ModelCenter及びExcel(登録商標)等の、特有なモデル650-1乃至650-nに関する情報が暗黙的に含まれ

50

ている。プロセスモデルのModelCenterモデルは、特定モデル655から情報を受け取る。

【0125】

Excelモデルは、例えばURL (Uniform Resource Locator) 等の使用される関数への参照を使う。これに対し、ModelCenterモデルは、カプセル化によってプロセスモデルの中に含まれていることに留意すべきである。

【0126】

セクション610、615及び620において指定されたデータは、自体が他のオブジェクトへのリンク (665,670) を有するオブジェクト660によって制作されたものである。

【0127】

このように、参照符号600乃至650は一般プロセスモデルに関連付けられている。同時に、この参照セット (l'ensemble des references) は、インスタンス化プロセスモデルに対応している。しかしながら、パラメータ620は、また、ユーザにより手動で変更することができることに注意する必要がある。

10

【0128】

図7には、インスタンス化プロセスモデルの単位 (unitaires) プロセスを実行するためのアルゴリズム700の一例が示されている。

【0129】

インスタンス化プロセスモデル705は、最初に、予め定められたXMLスキーマによってその有効性をコントロールするため解析される (ステップ710)。もし有効でない場合、プロセスの実行を中止し、別のインスタンス化プロセスモデルを処理することができる。インスタンス化プロセスモデルが有効な場合、このモデルに含まれている情報が抽出される (ステップ720)。

20

【0130】

この情報抽出により、特に、実行コンテキストの構成725と、特に、入力、出力、入力/出力、パラメータ及び/又はインスタンス化プロセスモデルに固有のデータを含むことがある実行環境730と、を決定することができる。好ましい実施例によれば、実行コンテキストの構成 (configuration) 725及び実行環境はPythonフォーマットで決定される。

【0131】

次に、実行コンテキストの構成は、実行ドメインの有効性に基づいてコントロールされる (ステップ735)。実際に、プロセスの作成者は、例えば、ソフトウェアの特定のバージョン等の、有効性の領域 (domain) を指定することができる。ユーザが別のバージョンの使用を所望した場合、警告されるか、自分の選択を変更するよう強制されることがある。実行ドメインの有効性の条件が満たされないからである。構成 (configuration) が有効でない場合、プロセスの実行を中止し、別のインスタンス化プロセスモデルを処理することができる。

30

【0132】

構成が有効である場合、入力及び入力/出力に関する条件が検証される (ステップ740)。プロセスのインスタンス化するとき、ユーザがプロセスの作成者によって定義された条件を満たさないためこのプロセスで使用することができないオブジェクトを、参照することが起こりえる。テーブル及び椅子に関する上記の例において、引越しプロセスは、例えば、椅子の特有な色を条件とすることができる。ユーザが別の色の椅子を選択した場合、引越しプロセスが行われないことがありえる。入力及び入力/出力に関する条件が有効でない場合、プロセスの実行を中止し、別のインスタンス化プロセスモデルを処理することができる。

40

【0133】

入力及び入力/出力に関する条件が有効な場合、プロセスの実行は、インスタンス化プロセスモデルのコード、即ち、インスタンス化プロセスモデルを表わすオブジェクトに関連付けられたデータに従って実現される (ステップ745)。

【0134】

次に、インスタンス化プロセスモデルのコードが実行される (ステップ750)。一般に

50

、SRUNと呼ばれる実行モジュールのコアは、入力、入力/出力及びコードを実行するのに必要なPythonモジュールをメモリにロードする。必要に応じて、実行モジュールSRUNの能力に基づいて分散実行環境が利用されてもよい。

【0135】

複合インスタンス化プロセスモデル (modeles de processus instances composites) の場合、実行モジュールSRUNは、サブプロセスモデルのインスタンスをメモリに作成し、これらのインスタンスを単位インスタンス化プロセスモデルとして順次実行する。種々の単位プロセス間の一貫性は、ジョブの流れ又はワークフローの実行エンジンにより保証される。

【0136】

プロセス実行中に、コードを含むスクリプトは、グリッドコンピューティングマネージャ (gestionnaire de la grille de calcul) の選択に関して、場合により、データマネージャのマシンとは異なるマシン上で、対応する実行環境において実行される。セキュリティ保護された方法でプロセスを実行できるメカニズムが、物理データを読み書きしたり、プロセス実行の履歴を記憶したり、データを一時的に記憶したりするのに使われることが好ましい。

【0137】

スクリプト、即ち、プロセスを実行した後、実行エンジンは、適切なドメインに出力を送り、実行レポートを記憶する。

【0138】

図8は、本発明による共同サービス指向ワークショップで行うことができるデータ800処理の一例を示している。この例によると、データ805は、傷んだ家に関する情報に対応している。オブジェクトは、メタデータを参照しており、傷んだ状態を詳述するこれらのデータ並びに関連すると判断される他の情報（ここでは、部屋の数及び窓の数）に関連付けられる。データ810は、居住可能な家に関する情報に対応する。ここでも、オブジェクトは、メタデータを参照しており、この場合居住可能な状態である状態を示すこれらのデータ並びに関連すると判断された情報（すなわち、ここでは、部屋の数及び窓の数）に関連付けられる。データ810は、データ805にリンクされたオブジェクトから得ることができ、したがって、データ805から、リフォームのインスタンス化プロセスモデル 8 1 5 を表わすオブジェクトによって得ることができる。

【0139】

同様に、引越し830のインスタンス化プロセスモデルにおいて、データ810対応するオブジェクトを、データ820にリンクされたダイニングルームタイプのオブジェクト及びデータ825にリンクされたベッドタイプのオブジェクトと組み合わせることによって、データ835にリンクされた居住中の家タイプのオブジェクトを作成することができる。データ835にリンクされたメタデータは、ここでは、居住状態並びに部屋及び窓の数を詳述する。

【0140】

付属資料の「extrait de code 3 (コード3の抜粋)」に示されている例は、リフォームプロセスを実行する前の、図8のデータ805に関連付けられたメタデータを簡略化した例である。

【0141】

この例に示されているように、メタデータは、特にオブジェクトのタイプ及び名前並びに対応するデータにアクセスするためのリンクなどの一般情報と、ここでは状態並びに窓及び部屋の数などの特有情報とを有する。メタデータは、INタイプである、即ち、インスタンス化プロセスモデルにより作成されるデータに対応しないことに留意すべきである。

【0142】

傷んだ家を表わす上記の例を処理するのに使用できるリフォームプロセスモデルは、一般に、付属資料の「extrait de code 4 (コード4の抜粋)」に示されるように記述することができる。

10

20

30

40

50

【0143】

このプロセスモデルは、家タイプのオブジェクトを入力として有し、家タイプの別のオブジェクトを出力として作成する。このプロセスモデルは、インスタンス化されることによって、特に、図8のデータ805及び810に関連付けられたオブジェクトへの参照を有することができる。この場合、このインスタンス化されたプロセスモデルは、例えば、付属資料の「extrait de code 5 (コード5の抜粋)」に示されている。

【0144】

示されているように、インスタンス化されたプロセスモデルは、さらに、実行パラメータ、実行マシンの識別子、使用される適用コンフィギュレーション及び実行履歴のリンクを有する。これらのリンクは、データ805に関連付けられたオブジェクトとインスタンス化プロセスモデル815に関連付けられたオブジェクトとの間のリンク、及び、インスタンス化プロセスモデル815に関連付けられたオブジェクトとデータ810に関連付けられたオブジェクトとの間のリンクである。

10

【0145】

このインスタンス化プロセスモデルに対応するプロセスを実行した後、データ805にリンクしているメタデータは、データのトレーサビリティを可能にする履歴のリンクを追加するよう変更される。したがって、データ805に関連付けられたこのメタデータは、プロセスの実行後、付属資料のextrait de code 6 (コード6の抜粋)」に示されているように表現することができる。

【0146】

この例で詳述されているように、データ805に関連付けられたオブジェクトとインスタンス化プロセスモデル815に関連付けられたオブジェクトとの間にリンクが作成される。したがって、インスタンス化プロセスモデル815に関連付けられたオブジェクトを使って、データ805に関連付けられたオブジェクトとデータ810に関連付けられたオブジェクトとの間にリンクを確立することができる。

20

【0147】

同様に、プロセスの実行後に、データ810と、該データに関連付けられリンクを有するメタデータとが作成される。データ810に関連付けられたメタデータは、付属資料のextrait de code 7 (コード7の抜粋)」に示されているように表現することができる。

【0148】

前述のように、データ810に関連付けられたオブジェクトとインスタンス化プロセスモデル815に関連付けられたオブジェクトとの間にリンクが作成された。したがって、インスタンス化プロセスモデル815に関連付けられたオブジェクトを使って、データ805とデータ810との間に存在するリンクを確立することができる。メタデータは、ここでは、OUTタイプである、即ち、サービス指向共同ワークショップにおいて、インスタンス化プロセスモデルにより作成されたデータに対応していることに留意すべきである。

30

【0149】

同様に、第2のプロセスモデルを作成してインスタンス化することによって、データ810、820及び825を処理してデータ835及び関連オブジェクトを作成することができる。引越しプロセスを特徴づける第2のインスタンス化プロセスモデル830が実行されるときに、データ810に関連付けられたメタデータは、インスタンス化プロセスモデル830の実行により得られたリンクを統合するよう変更される。データ810に関連付けられたメタデータは、付属資料の「extrait de code 8 (コード8の抜粋)」に示されているように表現することができる。

40

【0150】

データ810に関連付けられたオブジェクトとインスタンス化プロセスモデル830に関連付けられたオブジェクトとの間に作成されたリンクによって、後者のオブジェクトを使って、データ810及び835に関連付けられたオブジェクト間でリンクを確立することができる。

【0151】

また、第2インスタンス化プロセスモデル830の実行時に、データ835及び関連付けられ

50

たオブジェクトが作成される。関連付けられたメタデータは、付属資料の「extrait de code 9 (コード9の抜粋)」に示されているように表現することができる。

【0152】

データ835に関連付けられたオブジェクトとインスタンス化プロセスモデルに関連付けられたオブジェクト830”との間に作成されたリンクによって、後者のオブジェクトを使って、データ835及び810に関連付けられたオブジェクト間、データ835及び820に関連付けられたオブジェクト間、並びに、データ835及び825に関連付けられたオブジェクト間のリンクを決定することができる。

【0153】

図9には、発明の一部を利用するよう適合した装置が示されている。装置900は、例えば、マイコン、コンピュータまたはワークステーションである。

10

【0154】

ここでは、装置900は、通信バス902を有し、該バスには次のものが接続されている。
 - 中央処理装置 (Central Processing Unit : CPU) 又はマイクロプロセッサ903。
 - 読み取り専用メモリ904 (Read Only Memory : ROM)。 プログラム " prog "、" prog1 " 及び " Prog2 " を有することができる。
 - ランダムアクセスメモリ (Random Access Memory : RAM) 又はキャッシュメモリ906。 上記プログラムの実行中に作成され変更される変数及びパラメータを記憶するよう適合したレジスタを有する。
 - データを送受信するよう適合した通信インターフェース918。

20

【0155】

オプションとして、装置900は次のものを配置してもよい。
 - 表示装置908。 データを視覚化し、及び/又は、グラフィックユーザインターフェースとして動作する表示装置908。 グラフィックユーザインターフェースによって、ユーザは、キーボード及びマウス910もしくはその他のポインティングデバイス、タッチスクリーン、又は、リモコンを使って、本発明に係るプログラムと対話することができる。
 - ハードディスク912。 上記プログラム " prog "、" prog1 " 及び " Prog2 "、並びに、本発明に従って処理された又は処理するデータを有することができる。
 - メモリカードリーダー914。 メモリカード916を受け入れ、本発明に従って処理された又は処理するデータを読み書きするよう適合している。

30

【0156】

通信バスによって、装置900が有する又は装置に接続されている種々のコンポーネント間の通信及び相互運用 (interoperabilite) が可能になる。 バスの表現は制限的ではない。特に、CPUは、装置900の全てのコンポーネントに、直接又は装置900の別のコンポーネントを介して、命令を伝えることができる。

【0157】

プログラム可能な装置に本発明に係るプロセスを利用させることができる各プログラムの実行可能コードは、例えば、ハードディスク912又はROM 904に記憶することができる。

【0158】

一つの変形例によれば、メモリカード916は、データ及び上記プログラム実行可能コードを有することができる。 このコードは、一旦、装置900で読み取られると、HDD 912に格納される。

40

【0159】

別の変形例によれば、プログラム実行可能コードは、インターフェース918を介して、少なくとも部分的に受信され、前記と同じ方法で格納することができる。

【0160】

より一般的には、1つ又は複数のプログラムは、実行される前に、装置900の格納手段の1つに、ロードすることができる。

【0161】

中央装置903は、本発明に係る1つ又は複数のプログラムの命令又はソフトウェアコー

50

ドの一部の実行を命令し指示する。これらの命令は、ハードディスク912もしくはROM 904又は上記の他のストレージ手段に格納される。電圧が与えられると、例えばハードディスク912及びROM 904等の不揮発性メモリに格納されている1つ又は複数のプログラムは、ランダムアクセスメモリ906に転送される。これによって、ランダムアクセスメモリ906は、本発明に係る1つ又は複数のプログラムの少なくとも一部の実行可能コードと、本発明の実現に必要な変数及びパラメータを記憶するためのレジスタと、を有する。

【0162】

当然、特定のニーズを満たすために、本発明の分野における当業者は、上記説明に変更を加えることができる。

【表 1】

付属資料

Extrait de code 1 :

```
<Atelier.DM:FrameworkMetaData>
  <Atelier.DM:Attributes.Generic />
  <Atelier.DM:Attributes.Specific />
</Atelier.DM:FrameworkMetaData>
```

Extrait de code 2 :

```
<Atelier.DM:FrameworkMetaData
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM">
  <Atelier.DM:Attributes.Generic acl_name="Prive" comments="None"
creation_date="2006-04-27 10:29:43" db_type="ELEMENT"
external_id="chaise_numero_0010100432325" id="1411" lock_owner="None" lock_state="0"
mdb_instance="MDB_exploit" modification_date="2006-04-27 10:29:43" name="chaise1"
owner="nom_prenom" publication_level="Prive" shared_state="None" site="Toulouse"
source="LOCAL" status="Prive" version_comments="None" version_number="None"
version_owner="None" version_state="None" version_tag="None"/>
  <Atelier.DM:Attributes.Specific nb_pieds="4" epaisseur="5cm"
couleur="rouge" matiere="bois" poids="2kg"/>
</Atelier.DM:FrameworkMetaData>
```

10

Extrait de code 3 :

```
<Atelier.DM:EObjectProxy xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM"
publish="0" state="0" type="IN">
  <Atelier.DM:OriginalPointer>
    <Atelier.DM:Pointer external_id="description pour accès à la donnée physique"/>
  </Atelier.DM:OriginalPointer>
  <Atelier.DM:ObjectProxy type="MAISON">
    <Atelier.DM:Pointer external_id="description pour accès à la donnée physique"/>
  <Atelier.DM:FrameworkMetaData
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM">
    <Atelier.DM:Attributes.Generic acl_name="Prive" db_type="MAISON"
external_id="description pour accès à la donnée physique" id="identifiant"
mdb_instance="MDB_exploit" modification_date="" name="ma maison bleue" source="CMSDK"
status="Private"/>
    <Atelier.DM:Attributes.Specific etat="en ruine" nb_fenetre="4" nb_pieces="10"/>
  </Atelier.DM:FrameworkMetaData>
</Atelier.DM:ObjectProxy>
</Atelier.DM:EObjectProxy>
```

20

30

Extrait de code 4 :

```
<?xml version="1.0" ?>
  <Atelier.PT:ProcessTemplate comment="mon premier traitement" creator="createur"
id="identifiant" name="Renovation maison" schemaVersion="0.1" state="primary"
execution_state="not_submitted" type="action" version="0.1"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM"
xmlns:Atelier.MC="http://www.societe.com/egat/Atelier/MC"
xmlns:Atelier.PT="http://www.societe.com/egat/Atelier/PT"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

40

【表 2】

```

xsi:schemaLocation="http://www.societe.com/egat/Atelier/PT Atelier.PT.ProcessTemplate.xsd
http://www.societe.com/egat/Atelier/DM Atelier.DM.xsd">
<Atelier.PT:SandboxPointer external_id="" source_name="" source_site=""
source_type="" type="">
<Atelier.PT:Input category="object" desc="maison en ruine" name="" type="MAISON">
<Atelier.PT:Condition>
<![CDATA[nombre de fenetres > 3]]>
</Atelier.PT:Condition>
</Atelier.PT:Input>
<Atelier.PT:Output category="object" desc="maison habitable" name="" type="MAISON">
</Atelier.PT:Output>
<Atelier.PT:Parameter desc="paramètres d exécution" name="para_list" type="list"/>
<Atelier.PT:Server>
<Atelier.PT:ValidServer>
<![CDATA[mot clé et valeurs décrivant les caractéristiques machines]]>
</Atelier.PT:ValidServer>
<Atelier.PT:DefaultServer>
<![CDATA[mot clé et valeurs caractéristiques de la machine par défaut]]>
</Atelier.PT:DefaultServer>
</Atelier.PT:Server>
<Atelier.PT:ExecutionContext>
<Atelier.PT:Classification processType="batch">
<Atelier.PT:Suite name="Suite1"/>
<Atelier.PT:Tool name="Tool1"/>
</Atelier.PT:Classification>
<Atelier.PT:CreationContext class="" date="" hostname="" osname="" type="">
<![CDATA[[configuration] mot clé et valeur décrivant la configuration applicative par
default
[validity] condition de validite de la configuration applicative]]>
</Atelier.PT:CreationContext>
</Atelier.PT:ExecutionContext>
<Atelier.PT:Data/>
<Atelier.PT:Code comment="commentaire sur le code" langage="python" version="">
<![CDATA[Code a executer]]>
</Atelier.PT:Code>
</Atelier.PT:ProcessTemplate>

```

Extrait de code 5 :

```

<?xml version="1.0" ?>
<Atelier.PT:ProcessTemplate comment="mon premier traitement" creator="createur"
id="identifiant" name="Renovation maison" schemaVersion="0.1" state="primary"
execution_state="not_submitted" type="action" version="0.1"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM"
xmlns:Atelier.MC="http://www.societe.com/egat/Atelier/MC"
xmlns:Atelier.PT="http://www.societe.com/egat/Atelier/PT"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.societe.com/egat/Atelier/PT Atelier.PT.ProcessTemplate.xsd
http://www.societe.com/egat/Atelier/DM Atelier.DM.xsd">
<Atelier.PT:SandboxPointer external_id="" source_name="">
<Atelier.PT:Input category="object" desc="maison en ruine" name="Donnée_805"
type="MAISON">
<Atelier.PT:Condition>
<![CDATA[ nombre de fenetres > 3 ]]>
</Atelier.PT:Condition>
<Atelier.DM:EEObjectProxy publish="0" state="0" type="IN">
<Atelier.DM:OriginalPointer>

```

【表 3】

```

    <Atelier.DM:Pointer external_id="description pour accès à la donnée
physique"/>
  </Atelier.DM:OriginalPointer>
  <Atelier.DM:ObjectProxy type="MAISON">
    <Atelier.DM:Pointer external_id="description pour accès à la donnée
physique"/>
    <Atelier.DM:FrameworkMetaData
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM">
      <Atelier.DM:Attributes.Generic acl_name="Prive" db_type="MAISON"
external_id="description pour accès à la donnée physique" id="identifiant"
mdb_instance="MDB_exploit" modification_date="" name="ma maison bleue" source="CMSDK"
status="Private"/>
      <Atelier.DM:Attributes.Specific etat="en ruine" nb_fenetre="4"
nb_pieces="10" />
      <Atelier.DM:Links>
        <Atelier.DM:Link comments="description du lien" date=""
id_FO_1="identifiant donnée 805" id_FO_2="identifiant IPT 815" mdb_instance="MDB_exploit"
owner="to39751" subtype="sous type de lien" type="HISTORY"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM" />
      </Atelier.DM:Links>
    </Atelier.DM:FrameworkMetaData>
  </Atelier.DM:ObjectProxy>
</Atelier.DM:EEOBJECTProxy>
</Atelier.PT:Input>
<Atelier.PT:Output category="object" desc="maison habitable" name="Donnée_810"
type="MAISON">
  <Atelier.DM:EEOBJECTProxy publish="1" state="1" type="OUT">
    <Atelier.DM:SandboxPointer>
      <Atelier.DM:Pointer external_id="description pour accéder au bac a sable"
source_name="" source_site="" source_type="" type="" />
    </Atelier.DM:SandboxPointer>
    <Atelier.DM:FinalPointer>
      <Atelier.DM:Pointer external_id="description pour accès à la donnée
physique"/>
    </Atelier.DM:FinalPointer>
  </Atelier.DM:EEOBJECTProxy>
</Atelier.DM:ObjectProxy type="MAISON">
  <Atelier.DM:Pointer external_id="description pour accès à la donnée
physique"/>
  <Atelier.DM:FrameworkMetaData
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM">
    <Atelier.DM:Attributes.Generic acl_name="Prive" db_type="MAISON"
external_id="description pour accès à la donnée physique" id="identifiant"
mdb_instance="MDB_exploit" modification_date="" name="ma maison bleue" source="CMSDK"
status="Private"/>
    <Atelier.DM:Attributes.Specific etat="habitable" nb_fenetre="6"
nb_pieces="12" />
    <Atelier.DM:Links>
      <Atelier.DM:Link comments="description du lien" date=""
id_FO_1="identifiant IPT 815" id_FO_2="identifiant Donnée 810" mdb_instance="MDB_exploit"
owner="to39751" subtype="sous type de lien" type="HISTORY"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM" />
    </Atelier.DM:Links>
  </Atelier.DM:FrameworkMetaData>
</Atelier.DM:ObjectProxy>
</Atelier.DM:EEOBJECTProxy>
</Atelier.PT:Output>
<Atelier.PT:Parameter desc="parametres d execution" name="para_list" type="list">
  <Atelier.PT:ParameterValue>

```

【 表 4 】

<pre> <![CDATA[parametre d execution]]> </Atelier.PT:ParameterValue> </Atelier.PT:Parameter> <Atelier.PT:Server> <Atelier.PT:ValidServer> <![CDATA[mot clé et valeurs décrivant les caractéristiques machines]]> </Atelier.PT:ValidServer> <Atelier.PT:DefaultServer> <![CDATA[mot clé et valeurs caractéristiques de la machine par défaut]]> </Atelier.PT:DefaultServer> </Atelier.PT:Server> </Atelier.PT:Server> <Atelier.PT:ExecutionContext> <Atelier.PT:Classification processType="batch"> <Atelier.PT:Suite name="Suite1" /> <Atelier.PT:Tool name="Tool1" /> </Atelier.PT:Classification> <Atelier.PT:CreationContext class="" date="" hostname="" osname="" type=""> <![CDATA[[configuration] mot clé et valeur décrivant la configuration applicative par défaut [validity] condition de validité de la configuration applicative]]> </Atelier.PT:CreationContext> <Atelier.PT:RunContext class="" date="" hostname="" osname="" type=""> <![CDATA[[configuration] mot clé et valeur décrivant la configuration applicative utilisée]]> </Atelier.PT:RunContext> <Atelier.PT:RunServer> <![CDATA[mot clé et valeur décrivant la machine d'exécution utilisée]]> </Atelier.PT:RunServer> <Atelier.PT:ExecutionReport return_code="0"> <![CDATA[Rapport d execution]]> </Atelier.PT:ExecutionReport> </Atelier.PT:ExecutionContext> <Atelier.PT:Data /> <Atelier.PT:Code comment="commentaire sur le code" langage="python" version=""> <![CDATA[Code a exécuter]]> </Atelier.PT:Code> </Atelier.PT:ProcessTemplate> </pre>	<p>10</p> <p>20</p>
<p style="text-align: center;"><u>Extrait de code 6 :</u></p> <pre> <Atelier.DM:EEObjectProxy xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM" publish="0" state="0" type="IN"> <Atelier.DM:OriginalPointer> <Atelier.DM:Pointer external_id="description pour accès à la donnée physique" source_name="" source_site="" source_type="" type="" /> </Atelier.DM:OriginalPointer> <Atelier.DM:ObjectProxy type="MAISON"> <Atelier.DM:Pointer external_id="description pour accès à la donnée physique"/> <Atelier.DM:FrameworkMetaData xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM"> <Atelier.DM:Attributes.Generic acl_name="Prive" db_type="MAISON" external_id="description pour accès à la donnée physique" id="identifiant" mdb_instance="MDB_exploit" modification_date="" name="ma maison bleue" source="CMSDK" status="Private"/> <Atelier.DM:Attributes.Specific etat="en ruine" nb_fenetre="4" nb_pieces="10" /> </Atelier.DM:FrameworkMetaData> </Atelier.DM:ObjectProxy> </Atelier.DM:EEObjectProxy> </pre>	<p>30</p> <p>40</p>

【表 5】

```

    <Atelier.DM:Link comments="description du lien" date="" id_FO_1="identifiant
Donnée 805" id_FO_2="identifiant IPT 815" mdb_instance="MDB_exploit" owner="to39751"
subtype="sous" type de lien" type="HISTORY"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM" />
  </Atelier.DM:Links>
</Atelier.DM:FrameworkMetaData>
</Atelier.DM:ObjectProxy>
</Atelier.DM:EEOObjectProxy>

```

Extrait de code 7 :

10

```

<Atelier.DM:EEOObjectProxy xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM"
publish="1" state="1" type="OUT">
  <Atelier.DM:SandboxPointer>
    <Atelier.DM:Pointer external_id="description pour accès au bac a sable"/>
  </Atelier.DM:SandboxPointer>
  <Atelier.DM:FinalPointer>
    <Atelier.DM:Pointer external_id="description pour accès à la donnée physique"/>
  </Atelier.DM:FinalPointer>
  <Atelier.DM:ObjectProxy type="MAISON">
    <Atelier.DM:Pointer external_id="description pour accès à la donnée physique"/>
    <Atelier.DM:FrameworkMetaData
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM">
      <Atelier.DM:Attributes.Generic acl_name="Prive" db_type="MAISON"
external_id="description pour accès à la donnée physique" id="identifiant"
mdb_instance="MDB_exploit" modification_date="" name="ma maison bleue" source="CMSDK"
status="Private"/>
      <Atelier.DM:Attributes.Specific etat="habitable" nb_fenetre="6" nb_pieces="12" />
    </Atelier.DM:Attributes>
    <Atelier.DM:Links>
      <Atelier.DM:Link comments="description du lien" date="" id_FO_1="identifiant IPT
815" id_FO_2="identifiant Donnée 810" mdb_instance="MDB_exploit" owner="to39751"
subtype="sous" type de lien" type="HISTORY"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM" />
    </Atelier.DM:Links>
    </Atelier.DM:FrameworkMetaData>
  </Atelier.DM:ObjectProxy>
</Atelier.DM:EEOObjectProxy>

```

20

Extrait de code 8 :

30

```

<Atelier.DM:EEOObjectProxy xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM"
publish="1" state="1" type="OUT">
  <Atelier.DM:SandboxPointer>
    <Atelier.DM:Pointer external_id="description pour accès au bac à sable" source_name=""
source_site="" source_type="" type="" />
  </Atelier.DM:SandboxPointer>
  <Atelier.DM:FinalPointer>
    <Atelier.DM:Pointer external_id="description pour accès à la donnée physique"/>
  </Atelier.DM:FinalPointer>
  <Atelier.DM:ObjectProxy type="MAISON">
    <Atelier.DM:Pointer external_id="description pour accès à la donnée physique"/>
    <Atelier.DM:FrameworkMetaData
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM">
      <Atelier.DM:Attributes.Generic acl_name="Prive" db_type="MAISON"
external_id="description pour accès à la donnée physique" id="identifiant"

```

40

【表 6】

```

mdb_instance="MDB_exploit" modification_date="" name="ma maison bleue" source="CMSDK"
status="Private"/>
  <Atelier.DM:Attributes.Specific etat="habitable" nb_fenetre="6" nb_pieces="12" />
  <Atelier.DM:Links>
    <Atelier.DM:Link comments="description du lien" date="" id_FO_1="identifiant IPT
815" id_FO_2="identifiant Donnée 810" mdb_instance="MDB_exploit" owner="to39751"
subtype="sous" type de lien" type="HISTORY"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM" />
    <Atelier.DM:Link comments="description du lien" date="" id_FO_1="identifiant
Donnée 810" id_FO_2="identifiant IPT 830" mdb_instance="MDB_exploit" owner="to39751"
subtype="sous" type de lien" type="HISTORY"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM" />
  </Atelier.DM:Links>
</Atelier.DM:FrameworkMetaData>
</Atelier.DM:ObjectProxy>
</Atelier.DM:EEObjectProxy>

```

10

Extrait de code 9 :

```

<Atelier.DM:EEObjectProxy xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM"
publish="1" state="1" type="OUT">
  <Atelier.DM:SandboxPointer>
    <Atelier.DM:Pointer external_id="description pour accès au bac à sable"/>
  </Atelier.DM:SandboxPointer>
  <Atelier.DM:FinalPointer>
    <Atelier.DM:Pointer external_id="description pour accès à la donnée physique"/>
  </Atelier.DM:FinalPointer>
  <Atelier.DM:ObjectProxy type="MAISON">
    <Atelier.DM:Pointer external_id="description pour accès à la donnée physique"/>
    <Atelier.DM:FrameworkMetaData
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM">
      <Atelier.DM:Attributes.Generic acl_name="Prive" db_type="MAISON"
external_id="description pour accès à la donnée physique" id="identifiant"
mdb_instance="MDB_exploit" modification_date="" name="ma maison bleue" source="CMSDK"
status="Private"/>
      <Atelier.DM:Attributes.Specific etat="habite" nb_fenetre="6" nb_pieces="12" />
      <Atelier.DM:Links>
        <Atelier.DM:Link comments="description du lien" date="" id_FO_1="identifiant IPT
830" id_FO_2="identifiant Donnée 835" mdb_instance="MDB_exploit" owner="to39751"
subtype="sous" type de lien" type="HISTORY"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM" />
        <Atelier.DM:Link comments="description du lien" date="" id_FO_1="identifiant IPT
830" id_FO_2="identifiant Donnée 835" mdb_instance="MDB_exploit" owner="to39751"
subtype="sous" type de lien" type="HISTORY"
xmlns:Atelier.DM="http://www.societe.com/egat/Atelier/DM" />
      </Atelier.DM:Links>
    </Atelier.DM:FrameworkMetaData>
  </Atelier.DM:ObjectProxy>
</Atelier.DM:EEObjectProxy>

```

20

30

40

【表 7】

テーブル 1 :

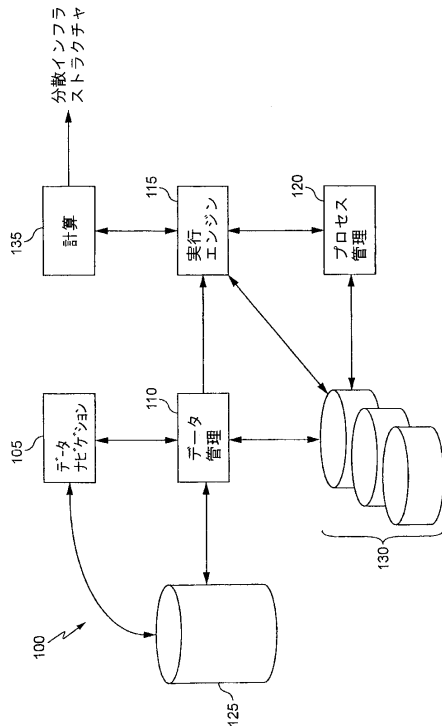
メタデータ							
一般				特別			
ID	日付	URL	タイプ	nb pieds	材料	色	重量
1	12/09/07	http://x.y/z/	コンポーネント	3	木	赤	3500
...
n	29/01/08	http://m.n/o/p/q	コンポーネント	4	金属	緑	1800

テーブル 2 :

ID1	5	8	3	5
ID2	8	5	1	1

10

【図 1】



【図 2 a】

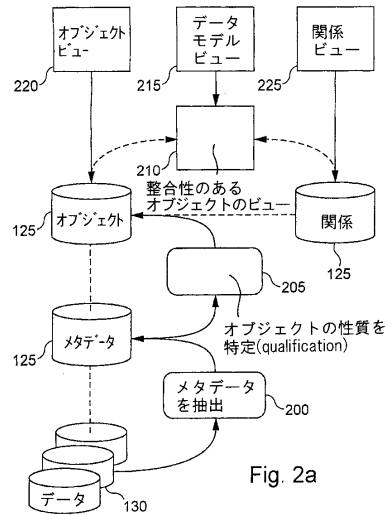


Fig. 1

Fig. 2a

【図2b】

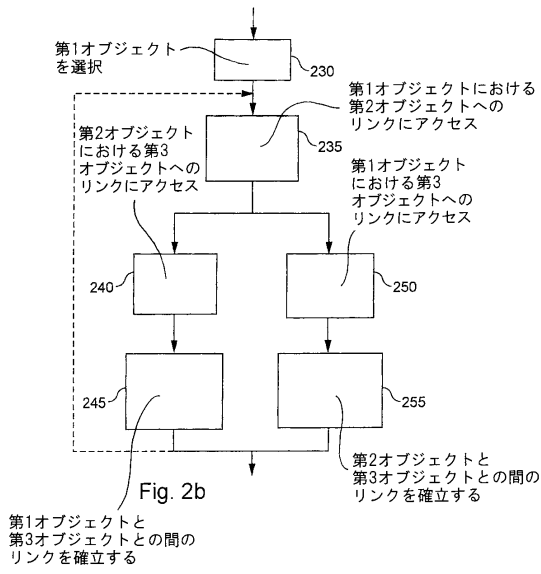


Fig. 2b

【図3】

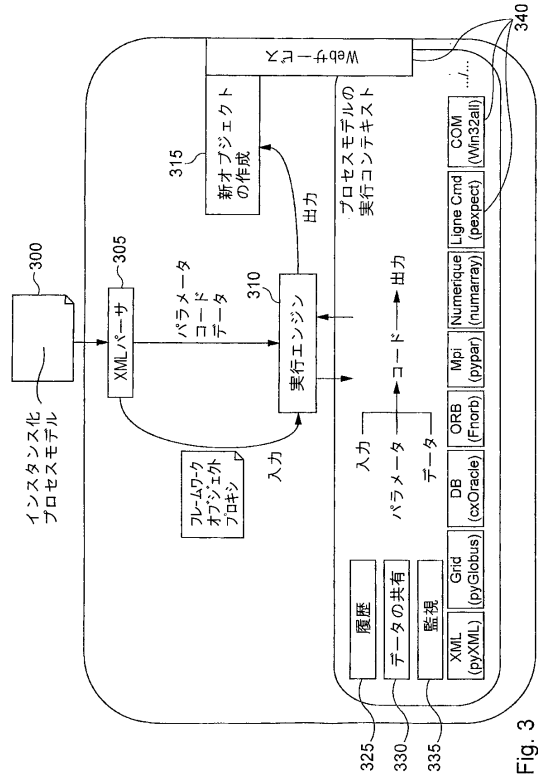


Fig. 3

【図4】

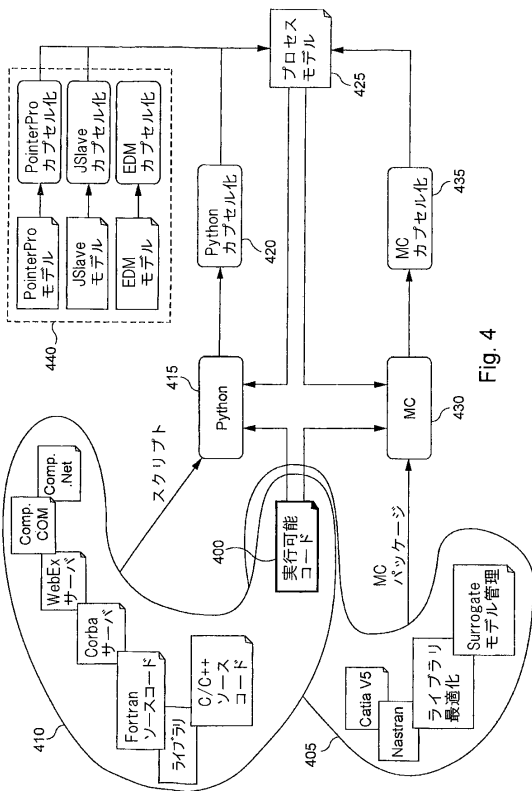


Fig. 4

【図5】

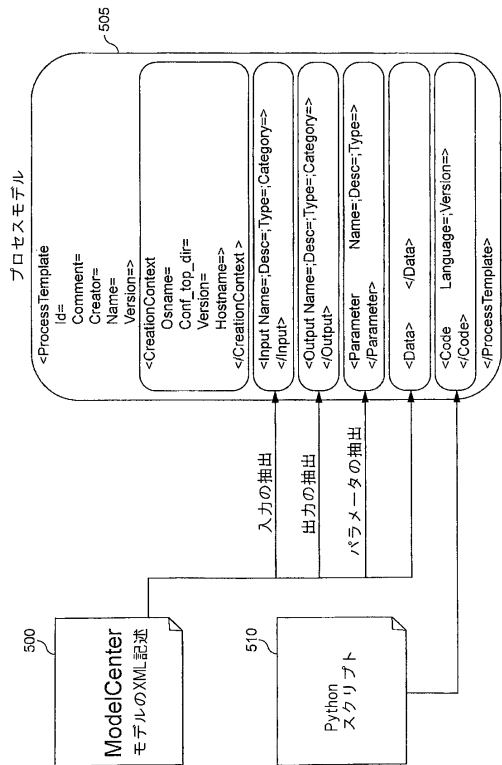


Fig. 5

【 図 6 】

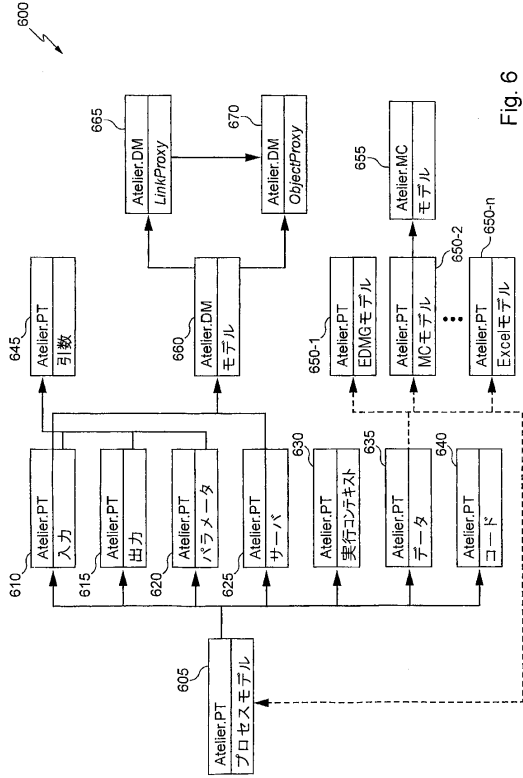


Fig. 6

【 図 7 】

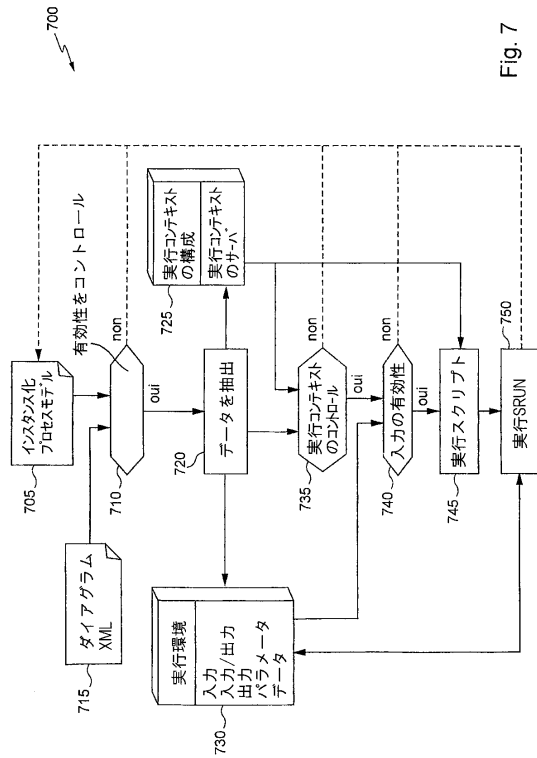


Fig. 7

【 図 8 】

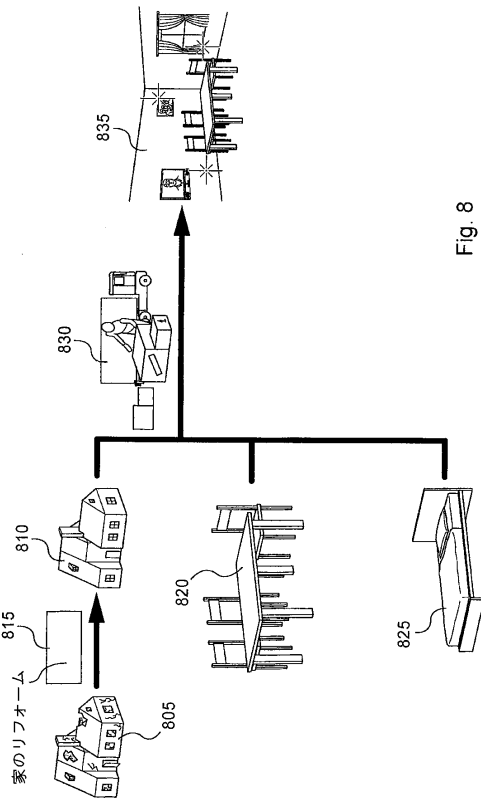


Fig. 8

【 図 9 】

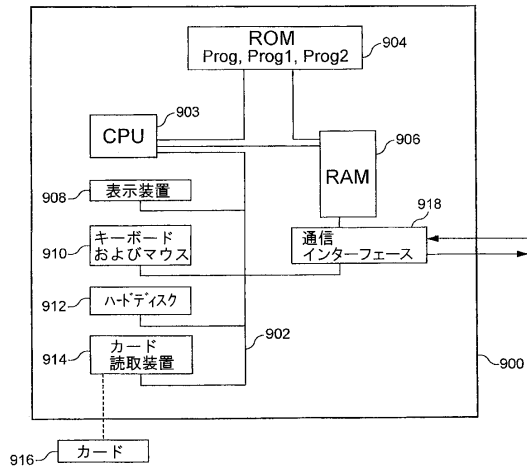


Fig. 9

フロントページの続き

- (72)発明者 マルク, ベルナール
フランス国, エフ - 3 1 5 3 0 ル カステラ, ルート ドゥ レゲバン
- (72)発明者 シュバリエ, ティエリー
フランス国, エフ - 3 1 8 2 0 ピブラック, ルート ドゥ レゲバン, 3 5

審査官 唐橋 拓史

- (56)参考文献 特開平 1 0 - 0 9 1 4 4 9 (J P , A)
特開平 0 8 - 2 8 6 9 6 2 (J P , A)
特開平 1 0 - 1 1 1 8 0 2 (J P , A)
特表 2 0 0 2 - 5 0 2 0 7 2 (J P , A)
特開 2 0 0 1 - 0 6 0 2 1 5 (J P , A)
特開 2 0 0 6 - 2 6 8 4 7 4 (J P , A)
特開 2 0 0 7 - 2 2 6 3 7 5 (J P , A)
Tretola, G. 他, Extending Web Services Semantics to Support Asynchronous Invocations and Continuation, Web Services, 2007. ICWS 2007. IEEE International Conference on, 2 0 0 7 年 7 月 9 日, pp.208-215
万谷 哲, HPC向けR&Dソリューション, インターネット<URL: <http://www.sskn.gr.jp/MAINSITE/download/newsletter/2005/sci/1/doc2/doc2.pdf>>, [online], [平成25年7月11日検索], 2 0 0 5 年 8 月 3 0 日, pp.1-28

- (58)調査した分野(Int.Cl., DB名)
G 0 6 Q 1 0 / 0 0 - 5 0 / 3 4