

クロスサイトプロセスのモニタリングビューを提供する メタモデルと識別子情報に関する検討

富士通(株) ソフトウェア事業本部 加藤光幾、金井剛

日本電気(株) システム基盤ソフトウェア開発本部 神南吉宏、島村栄、山本拓磨、菊地伸治

抄録: 民間企業、公共団体に限らず、より利用者指向の高付加価値サービスを実現するための情報システム基盤として SOA、SOC が注目を集めている。安定した SOA による運用環境を実現するには、ミクロ的なサービスレベルのみならず、マクロ的なクロスサイトプロセスレベルを可視化することも必要になる。従来、システム運用領域では ITIL や DMTF の CIM 等が標準として検討、策定されているが、これらはクロスサイトプロセスレベルの論点で検討されている訳ではない。そのため、新たなモニタリングビューとなるメタモデルの定義は、一つの重要な研究課題である。本稿では、上記目的のため、新たなプロセスモデルと識別子を検討し、それを用いた運用例を定義したので報告する。

Abstract: SOA and SOC have had a bigger role in an information system platform for realizing higher value added services for common customers, regardless of organizations. By adopting the SOA, it is required to make a process over multi-sites visible, not only individual elemental service, in order to realize a stable service environment. However, recent some standard efforts like ITIL and CIM tend to be insufficient on the point of view of visualizing multi-sites' processes. In this paper, we propose a candidate of the new meta-model and elemental identifiers for monitoring these managed multi-sites' processes.

1. はじめに

利用者にとって便利な高付加価値サービスを提供するため、ネットワーク上で提供される様々なサービス群を柔軟に結合するための技術に対する市場からの要求は、近年、富に増大している。この要求は民間企業のみならず自治体など公共団体でも同様である。例えば、公共領域の場合、2010年までに「いつでも、どこでも、何でも、誰でもネットワークにつながるユビキタス・ネットワーク社会」へと発展させることを旨とする「u-Japan 構想」が進行中である。この中では、国や地方公共団体等の行政機関への各種申請手続き等も、インターネットを介した新たなサービスとして定義され、将来的には、利用者が窓口へ直接出向いたり書類に記入したりすることなく、情報端末からの簡便な入力で容易にサービスを楽しむこと、更に利用者の住居引越しなど、生活環境や状況変化に伴うサービス提供者への個人情報の変更申請（電気・ガス・水道の解約および新規申し込み、地方公共団体への転入・転出届、児童手当の認定請求など）などを官民連携して、一括、ワンストップに実施出来る高付加価値サービス等が提案されている。この様な行政機関の住民サービスの効率化、高付加価値サービスの実現、これに伴う地域の活性化を指向し、様々なサービスを柔軟に結合することを目的として、(財)全国地域情報化推進協会を中心に、地域情報プラットフォームという電子自治体基盤の共通アーキテクチャ、標準仕様の開発が推進されている[1]。

このような動きの背景には、サービスを組み合わせ、新たなサービスを構成するための新たな情報システムパラダイム SOA (Service Oriented Architecture)、SOC (Service Oriented Computing) が存在しており、そのパラ

ダイムを具現化する国際標準である WS-I (Web Services Interoperability Organization) Basic Profile 1.0[2]等の通信基盤、サービスを実行する際の実行手順を定義する WS-BPEL (Web Service Business Process Execution Language)[3]の標準言語が策定、等の具体的な動きがある。

SOA をベースとし、上記の高付加価値サービスを実現するためには、ベンダ依存性のない標準的な通信基盤、アーキテクチャを持ち、異機種でも連動動作を可能とした上で、各行政機関、関連民間企業で運用されている各種アプリケーション群を各要素サービスとして定義し、インターネットを介してこれらを最適に組み合わせ、連動させることが必要になる。しかし複数異種のシステムプラットフォーム、各種アプリケーション群が大規模に連携する程、それぞれの運用上の課題や、障害要因等が混在しやすくなり、組み合わせにより想定もしていない新たな問題も顕在化し易い。それ故にサービス提供者、運用者には、提供しているサービスが期待通りに実行されているかを把握し、問題点や改善点を探るための運用・監視手段が必要となる。従来、サービス提供の運用・監視などシステム運用に関して、ITIL (Information Technology Infrastructure Library)[5] や、DMTF (Distributed Management Task Force) の CIM (Common Information Model)[6]等の標準が検討、策定されているが、これらはモニタリング対象の定義と枠組みに関する事項に留まっている、もしくは基本的に単一サイトのリソース管理に関する議論が中心である。更に通販サイト等(例えば Amazon)で、バックヤードの流通システムと連携し、顧客の注文がどこまで処理されているか、を表示するサービスも既に存在するが、アドホックであり、SOA などが目指す汎用的な疎結合環境への拡張は容易ではない。一方、高付加価値サービスは複数単位のサービスが疎結

合するので、各サービスは複数のサービス提供者から提供される。そのため各サービスを横串に連結し、サービス全体(クロスサイトプロセスと称する)の実行状態の把握するための新たなモニタリング・可視化手段の必要等、運用・監視の論点で新たな方法も必要となる。

本稿では独立行政法人 情報通信研究機構(NICT)からの委託研究開発「異なる運用ポリシーや異なるアーキテクチャのサービスが連携し、高付加価値サービスを提供できるためのサービス連携基盤技術の研究開発」で検討されている、新たなクロスサイトプロセスの実行状態の把握、モニタリング・可視化の方式を報告する。特に、複数の自治体に跨る(クロスサイト)プロセスの実行をモニタリングするため、新たなメタモデルおよびコンポーネントを定義し、それを集中や分散形式の運用監視に利用出来ることを述べる。

以下、まず 2. ではクロスサイトで実行される業務プロセスのモニタリングで収集するデータを表現するため、新たに定義したモニタリング向けメタモデルを説明する。3. では、当該メタモデルを元に、運用の際に重要となる各種識別子の構造について言及する。4. ではメタモデルに基づき生成されるモニタリングの記録・ログ情報(audit データ)の利用ケースを 2 種類述べる。5. では、他の従来方式と比較した場合の評価について言及する。

2. メタモデル定義

2.1 概要

本章では、クロスサイトプロセスレベルのモニタリングを指向して定義されるメタモデルについて論じる。新たなシステム要件、世界観としてのメタモデルを定義する場合、IEEE 標準 1471-2000 等で定義されるアーキテクチャに関する概念モデル、枠組みに準拠する必要がある[4]。しかし今回のメタモデルについては、前述する研究プログラムの一過程で達成された成果であり、多分に更新、強化、拡張の余地も残すことから、骨子部分に力点を置き定義している。特に対象の静的側面に関する世界観を表すクラス構成の定義、並びに当該クラス構成を実現するためのシステムアーキテクチャ、アーキテクチャに対して要件を定義し、クラス構成の動的振る舞い定義への与件ともなるユースケース等を検討しているが、本稿では、特にクラス図、コンポーネント図を中心に解説する。尚、実際のモニタリングシステムはこのメタモデルを元に実装アーキテクチャを定義の上で実現される。その意味で、モデルをメタモデルと称している。

2.2 クラス構成の定義

2.2.1 モニタリング要件、クラス設計上の与件、設計思想

一般に、モニタリングを実現する上での、システム構

造面に関する要求事項は幾つか存在するが、主要なものは、以下 3 点に集約、大別されると考えられる。(1)モニタリング対象を含む周辺世界の静的、動的構造を正確に転写・記述・維持管理する機能、(2)識別・選択されたモニタリング対象が生成するその内部状態に関する記録・ログ情報を過不足なく捕捉、維持管理する機能、(3)当該内部状態に関する記録・ログ情報を、利用者の観点で取捨・再整理・加工し、新たな付加価値情報として生成する機能。

クロスサイトプロセスレベルのモニタリングを実現する上では、これも例外ではなく、上記の 3 点を含むことになるが、モニタリング対象が複数サイトに跨り、自立的に構成されるが故に、特有の要件も存在し得る。第一は、(1)のモニタリング対象記述に於いて、各構成サイト内部のマイクロ視点の構造に加え、マクロ視点としてのサイト間の関係情報を明示的、かつ構造的に含める、という点である。第二は、当該(1)に基づき、(2)に於いて個々のサイトから取得されるモニタリング対象が生成する記録・ログ情報をクロスサイトプロセスの一部として識別し、関連性を明確化した上で、維持管理する必要がある。第三は(3)に於いても、クロスサイトプロセス全体のマクロ的、俯瞰的な挙動から、個々のサイトのミクロ的な挙動まで、さまざまなレベルの観点に応じて、かつシームレスに表示データを生成、提供すること、が要件として挙げられる。これらの特徴的要件はクラス構成上に含まれる形で一部表現され最適配置されるが、更に後述するアーキテクチャ上では、各コンポーネントの責任範囲として割り振られることになる。

ここで提案するクラス構成は、上記要件の(3)に関連する事項以外を、考慮して設計されている。

2.2.2 クラス構成の概要

クラス構成は、要件(1)のモニタリング対象を含む周辺世界の静的・動的構造の記述管理部分(以下、“記述管理要素群”と称する。)、並びに、要件(2)の識別・選択されたモニタリング対象が生成するその内部状態に関する記録・ログ情報を捕捉、維持管理する部分(以下、“記録管理要素群”と称する。)から成り立つ。

記述管理要素群は、図 2.1 上では中心から右側にかけて定義されるクラス群により表現され、クロスサイトプロセスを構成する様々な粒度レベル(全体を俯瞰する粒度レベルから単一プロセスの単一挙動に関する粒度レベル)に対応した階層的な構造を持つことを特徴とする。ここで、全体を俯瞰する粒度レベルを記述するために、「仮想プロセス」という概念を導入し、その具体的な挙動を規定するものとして「仮想プロセス定義」というクラスを導入する。仮想プロセスとは、メッセージの送受信によって関係付けられた複数の要素プロセス群、サービス群に

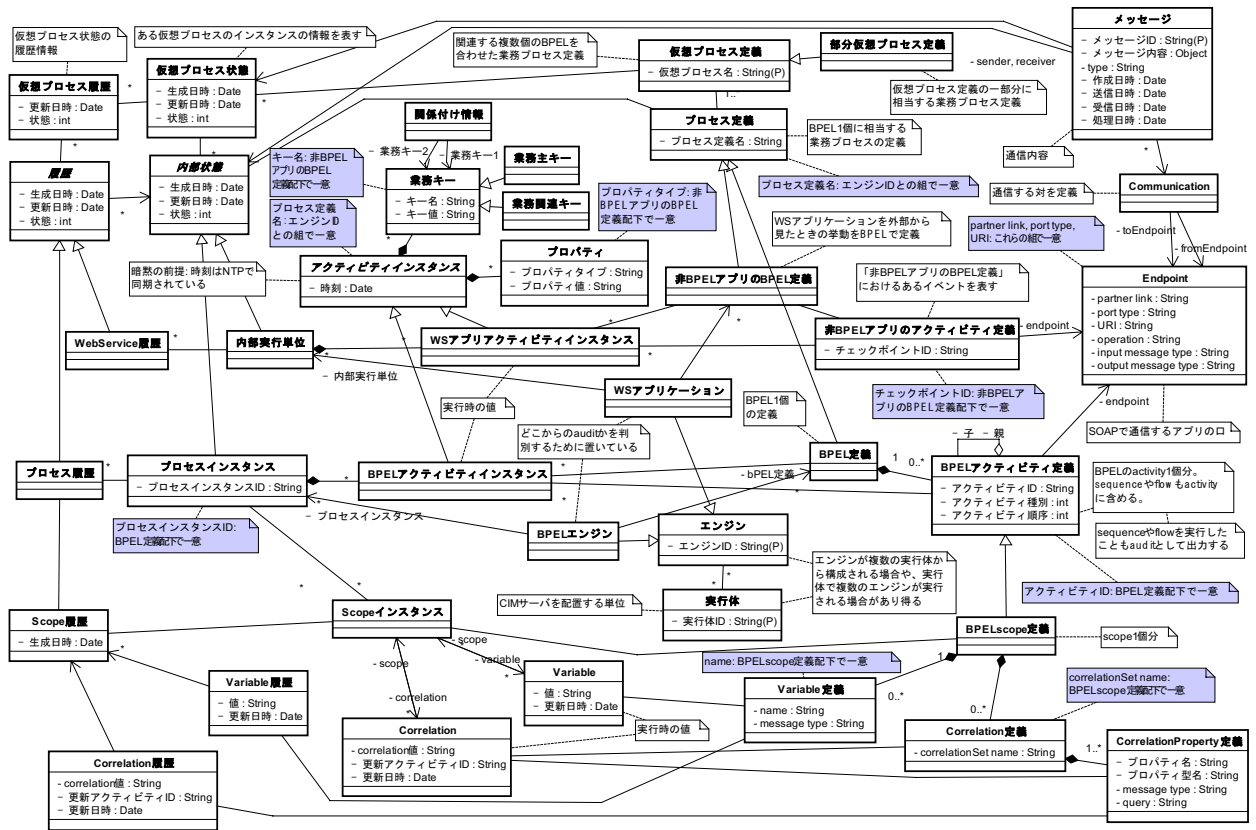


図 2.1 クロスサイトプロセスに関連するクラス構成

よって構成される大局的なプロセス概念のことである。具体的な実システムでは、クロスサイトを通して実施され、完結する一つの手続き、もしくは利用者に対して提供し得る高付加価値サービス単位と等価な粒度となる。また、要素プロセスは、構成上、自己参照出来るものであるが、常識的には一つの WS-BPEL 定義、Web Service で記述し得る粒度のものである。図 2.1 では、「BPEL 定義」や、Web Service 定義を含む「非 BPEL アプリの BPEL 定義」のクラスは、その旨を表現しているものである。これから任意の仮想プロセス定義に属する要素のプロセス定義である BPEL 定義や、非 BPEL アプリの BPEL 定義は、対応する Endpoint クラス間で交換されるメッセージ群を介して、当該仮想プロセス定義に属する 1 つ以上の他 BPEL 定義や、他非 BPEL アプリの BPEL 定義と通信を行う関係にある。

また、要素のプロセス定義である BPEL 定義や、非 BPEL アプリの BPEL 定義は、その内部動作を具体的に記述するため、要素アクティビティが定義されている。具体的には「BPEL アクティビティ定義」や、Web Service 定義を含む「非 BPEL アプリのアクティビティ定義」でありメッセージ種類、送受信等の方向性と内部挙動の対

応関係が、ここで規定される。これらの関係を有向グラフで対応付けることで、一つの仮想プロセス定義の記述単位を生成・識別することが可能となる。更に BPEL 定義、非 BPEL アプリの BPEL 定義の属性として、これらが展開・実装される処理系（ここでは「エンジン」と称する）も明示的に定義される。前者に対しては「BPEL エンジン」、後者に対しては「Web アプリケーション」が対応し、その属性としてエンジンが動作しているリソースの意である「実行体」も合わせて定義される。

それに対して、記録管理要素群は、図 2.1 上では中心から左側にかけて定義されるクラス群により表現される。ここでは「アクティビティ遷移」や「メッセージ送受信」等の個々のモニタリングした結果を前述仮想プロセスの挙動の一事象としてとらえ、仮想プロセスの状態情報として記録するよう構成されている。このため、モニタリング結果を統合する最も大きな単位として、先の仮想プロセス定義に対応する「仮想プロセス状態」が定義される。これは仮想プロセスのインスタンスに相当し、どのような仮想プロセスが動作しているか、記録・管理するためのものである。

仮想プロセス状態に代表される記録管理要素群の構成階層は、仮想プロセス定義に代表される記述管理要素群のそれと、概ね相似の階層構造を持つ。このため、仮想プロセス状態にて記録される仮想プロセスのインスタンスは、実行されている BPEL の「プロセスインスタンス」もしくは、Web Service の「内部実行単位」のインスタンスに対応し、更に、これら間で送受信された「メッセージ」のインスタンスになる。これにより、仮想プロセスのインスタンスに位置付けられる BPEL プロセスとサービスがどのような動作状態にあるかを記録・管理することができる。更に BPEL のプロセスインスタンスもしくは、Web Service の内部実行単位のインスタンスは、アクティビティレベルに分解出来、「BPEL アクティビティインスタンス」もしくは、「WS アプリアクティビティインスタンス」として管理され、各々が内部挙動的にどのような実行状態にあるかが記録される。

以上の説明にて定義した範囲は、モニタリングの計測時間と概ね同時性のある事象を対象としているが、仮想プロセスのインスタンスは、一種の状態遷移機械で表現出来得る。従って履歴状態を持つ。更に終了状態になった後も、中長期的な挙動解析のため、これらの履歴状態を保持する必要がある。そのため、過去を扱う「仮想プロセス履歴」クラスが定義される。更にモニタリング計測時間と同時性のある「仮想プロセス状態」に代表される記録管理要素群の構成階層は、過去を扱う仮想プロセ

ス履歴に代表される構成階層でも引き継がれる。ここでは、仮想プロセスのインスタンスに関する一連の履歴情報を保持し、これにより仮想プロセスのインスタンスがどのような経緯で次状態へと遷移したかを随時、参照することができるため、後の評価等で加工、利用出来る。

2.3 コンポーネントの構成定義

本節ではクロスサイトでのモニタリングで利用されるコンポーネントの役割を説明する。コンポーネント図を図 2.2 に示す。見やすさを考慮し、各コンポーネント間のデータの流れを示す線は省略している。業務システムおよびそれらを監視するコンポーネントは、図 2.2 の縦軸と横軸のように分類定義される。横軸右端から、仮想プロセス等の粒度に関係なく業務プロセスに関する設計ツールおよび定義情報、既存の業務システム(サービスを提供する実体およびそれらを構成するハード、ソフトウェアなどのリソース、実行体)、既存業務システムから挙動状態の記録・ログ情報を抽出し、各種 audit データに変換する audit 提供、当該 audit 提供の生成する audit データを必要に応じて統合、配布する Proxy、audit データを収集し監視する各種監視機能、の分類順でコンポーネントを位置付けている。情報は基本的に右側に分類されたコンポーネントから発生したものを左側のコンポーネントの入力とする様に流れる。ここで、既存システムの欄にある「BPM」とは Web Service のうち BPEL 定義を実行するものを、「WS アプリケーション」は BPEL 定義ではな

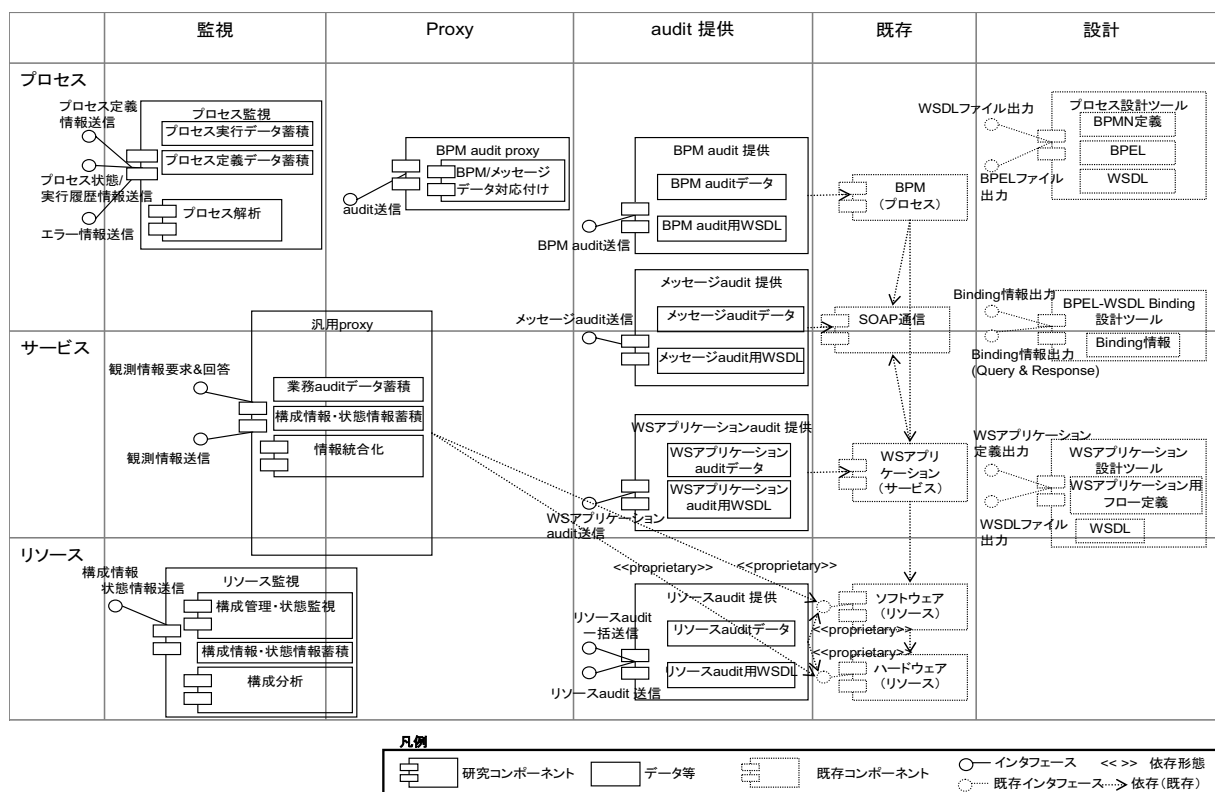


図 2.2 クロスサイトプロセスをモニタリングするコンポーネント構成定義

いものを実行するコンポーネントを表す。

また縦軸は、プロセス監視に関連するコンポーネント群、各種リソース監視に関連するコンポーネント群、およびプロセスから呼び出される Web Service や各種 audit データの再配布を行う汎用 proxy (簡略化した監視機能も持つ) からなるサービス群の 3 つの層から構成される。

図 2.2 においてクロスサイトでのモニタリングを実現するためには、プロセス監視ないし汎用 proxy など監視系のコンポーネントにおいて、複数サイトからの audit データを収集し、関係付ける。その実装形態には分散型と集中型があり、それぞれ 4.2 節と 4.3 節で説明する。これらのコンポーネント群を用いることにより、高付加価値サービスの利用者はプロセス監視機能等にアクセスし、複数サイトに連携・伝搬して実行されているプロセスインスタンスごとの実行状況を把握できる。またプロセス監視が提供する各種統計情報とリソース監視の統計情報を連携させ、高付加価値サービスの改善点をハードウェアリソース、個々のサービス、プロセス定義など複数の観点から検討することも可能となる。尚、リソース監視についての詳細説明は、ここでは割愛する。

3. 各種識別子の定義

本章では、図 2.1、図 2.2 で記されるメタモデルを元に、後述する多様な実装形態に於いても有効に機能する各種識別子構造に関して概略説明する。これらの識別子は、実際の運用局面において重要な役割を持ち、その構造を保障することで、種々の実装形態に対応出来る様になる。

各説明を表 3.1 に示す。

4. 利用ケースに関する定義

4.1 メタモデルを継承する複数の実装形態

本章では、メタモデルの内、定義コンポーネントをもとに、実装し得るソフトウェア形態について説明する。

図 2.2 に記される各コンポーネントは、図 2.1 のクラス構成を分解し、その要素からなる部分集合を写像・反映したものとして定義されることになる。クラス構成の分解の仕方によっては、実装し得るソフトウェアの形態も変化する。これに基づき、本章では図 2.1、図 2.2 によるメタモデルを継承しても、そこから実装し得る形態は複数存在し得ることを説明する。

一つは、図 2.2 に記される audit 提供機能を分散配置し、監視、モニタリングを完全分散化させる形態である。この場合、プロセス監視コンポーネントも完全分散化されることになり、相対的に機能上の役割が減じる。そのため、図 2.1 で定義され、プロセス監視コンポーネントに割り振られるべき仮想プロセス定義クラス、仮想プロセス状態クラス等は、抽象的なクラス概念としてのみ定義され、陽に具体化されず、汎用 Proxy の様なアグリゲーションする機能内に相当するものが暗黙的に定義・管理することになる。暗黙的が故に各種識別子には、それらの関連付けを可能にする定義・処置が必要となる。更に、図 2.2 の一部のコンポーネントでは、仮想プロセス状態クラスを分散的に維持管理する Agent の様な、アーキテクチャ上の機構が必要になって来る。

表 3.1 クラス定義のもとでの識別子の定義と意味

| 種類 | 説明、要件 |
|---------------------------|--|
| プロセス ID、 アクティビティ ID | プロセス・アクティビティには記述管理要素、記録管理要素の 2 面がある。仮想プロセスとの関係付けのためには記述管理要素、記録管理要素のどちらも、且つ、どのレベルにおいてもマルチサイト間でグローバルユニークな識別子が定義されている必要がある。 (1) BPEL プロセス定義：グローバルユニークに識別出来る「処理系」内でのユニークな識別子。 (2) BPEL プロセスインスタンス：同一「処理系」内の同一 BPEL プロセス定義内でのユニークな識別子。 (3) BPEL アクティビティ定義：BPEL プロセス定義相当の XML 記述内の Xpath 表記によりユニーク。 (4) BPEL アクティビティインスタンス：BPEL プロセスインスタンスと BPEL アクティビティ定義によりユニーク。 |
| メッセージ ID | このメッセージ ID とは、流通する業務メッセージに付与されるものである。メッセージの流通は観測される事象の一部を為すため、時刻と伴にマルチサイト間でグローバルユニークな識別子が定義されるべきである。特に時刻に対してはマルチサイト間で同期が保障されている必要がある。グローバルユニークを保障する方式には、(1)時刻、事象の発生場所等を組み合わせる方式、(2)状態遷移機械の特徴を利用し、前状態と遷移先状態、並びに入力データ等を組み合わせる方式、の 2 通りがある。 |
| エンジン ID | マルチサイト間でグローバルユニークな識別子。FQDN と postfix の組み合わせ等が考えられる。 |
| 実行体 ID | マルチサイト間でグローバルユニークな識別子。例えば、MAC アドレス等の利用を想定している。 |
| 業務キー ID | WS-BPEL 以外の Web Service でアプリケーションが一意性を定めるもので、メッセージとの関係では Correlation、もしくはそれ相当のものである。それ故、グローバルユニーク性等の制約は特に存在しない。 |

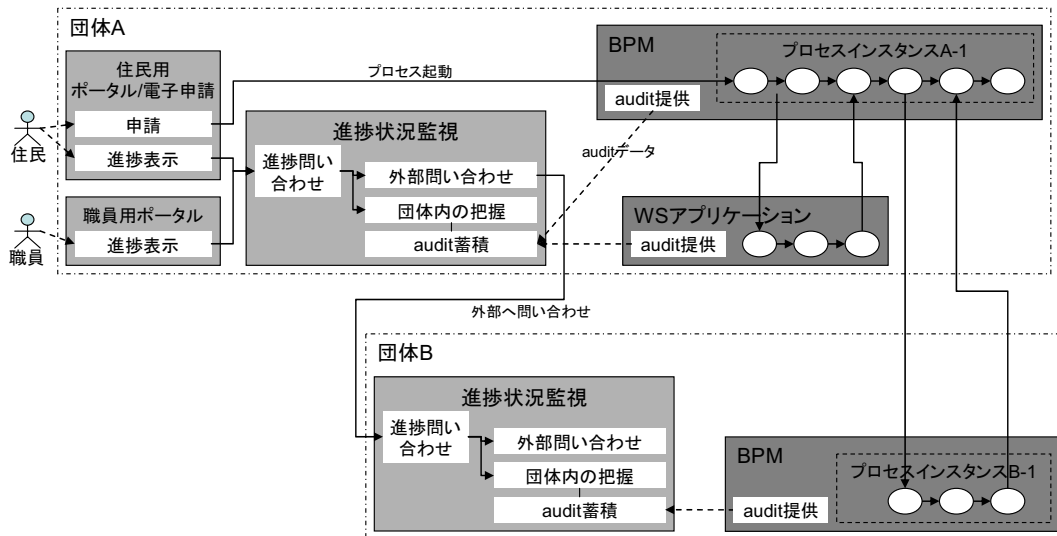


図 4.1 分散方式におけるプロセス情報の抽出

これに対して、図 2.2 に記される audit 提供機能を分散配置し、監視、モニタリングの方式を集中化、もしくは集中を旨とする局所分散化させる形態も存在する。プロセス監視コンポーネントは、前述の仮想プロセス定義クラス、仮想プロセス状態クラス等を、具体的に維持管理するコンポーネントとして具体的に存在することになる。明示的な存在が故に、逆に 2.2.1 節で言及したクロスサイトプロセスレベルのモニタリングを実現する上での特有な要件を陽に扱い、分散されている幾つかのクラス概念を集中維持管理するためのアーキテクチャ上の機構も必要になって来る。ここでは、以下、上記の完全分散化形態、並びに集中形態での実装方式例について説明する。

4.2 モニタリングを完全分散化した方式で実施する場合

図 4.1 は audit 提供機能が出力し、業務進捗を表す audit データを、サイトごとに配置した監視機能が収集・蓄積・関係付け、監視機能間で通信し、クロスサイトでのモニタリングを実現する方式をモデル化したものである。4.3 節で述べる集中化と異なり、2.2 節で述べたクラス構成のうち、仮想プロセス定義クラスを明示的に定義・利用しない。以下では単一サイトの場合とクロスサイトの場合の監視動作について述べる。

4.2.1 単一サイト内で複数 Web Service にまたがる監視

この方式では、ある業務プロセスに関する audit データと業務プロセス定義をそのサイト内の進捗状況監視機能で対応付け、業務の進捗状況を可視化・モニタリングする。業務システムがそのサイト内の Web Service (図 2.2 における BPM ないし WS アプリケーション) を呼び出した場合、Web Service の audit 提供機能が生成した audit データは同一監視機能に蓄積される。

Web Service のうち、通常、概ねの WS アプリケーションには業務の流れを明示的に定義した情報はないが、そのプロセスを WS-BPEL で擬似的に記述した定義を用意することで、BPM と同様の audit データを出力し、解釈することが可能となる。ここで擬似的に記述し audit データとして出力するのは、モニタリングを実施して役立つ業務イベント(たとえば SOAP メッセージ受信、決裁完了、など)が相当する。尚、WS アプリケーションと BPM を関係付ける情報として、WS アプリケーションに渡す業務メッセージ内で定義される適当な業務キー ID を用いることも可能である。

4.3 節の集中化方式と異なり、仮想プロセス定義クラスに相当する概念は直接的には不要であるため、複数業務システムに跨る業務進捗状況を可視化する際は、audit データを手掛かりに、業務実行に用いられた複数 WS-BPEL (疑似 WS-BPEL 定義を含む) から業務プロセスに特化する識別子等で、一連の情報を特定・合成し、更に実行時刻などの管理情報と合わせて利用者に提供、画面表示することになる。

4.2.2 クロスサイトで複数 Web Service にまたがる監視

任意 Web Service から呼び出された Web Service が他サイトに存在する場合、それらも含めてモニタリングするために、呼び出された Web Service を監視している他サイトの監視機能に対してクエリを発行し、進捗状況を収集する。

複数の進捗状況監視機能同士は、Agent の様に定期的に要素進捗状況監視機能が監視する Web Service に関する情報を交換しあう。その際、クエリを発行する監視機能を決定する方法には静的なもの動動的なものが存在する。WS-BPEL と WSDL の組み合わせで Web Service の通信先が静的に決定可能な場合、通信先の Web Service を監

視している他サイト内の進捗状況監視機能にクエリを発行し、WS-BPEL 定義と実行状況を取得する。一方、WS-BPEL 内で通信先が動的に決定される場合、audit データから送信先を特定し、それを監視する他サイト内の進捗状況監視機能に WS-BPEL 定義と実行状況を問い合わせる。収集された結果は、前述 4.2.1 と同様に、合成され、単一サイト、クロスサイトの区別なしにプロセスインスタンスの実行状況を把握する様に加工し、利用者に表示される。また、進捗状況監視機能の連携スコープを広げることで、より広範囲の監視を行うこともできる。

4.3 モニタリングを集中化、局所分散化した方式で実施する場合

図 4.2 は audit 提供機能を分散配置し、監視、モニタリング方式を集中化、もしくは集中を旨とする局所分散化させる方式を採用した場合の得るソフトウェア上の方式をモデル化したものである。図中の矢印は、2つのユースケースを含み、一つは各サイトから、発生する「アクティビティ遷移」や「メッセージ送受信」等の audit データを通知するユースケース、他は、利用者が仮想プロセスの動作状況、進捗把握をするために実施するクエリに関するユースケースである。

図 2.2 で定義された BPM audit 提供機能、メッセージ audit 提供機能、WS アプリケーション audit 提供機能等は、図 4.2 では、各団体が管理するサイト内の BPM の audit 提供機能にマッピングされる。また、必要に応じて、図

2.2 の BPM audit proxy 機能は、各 audit 提供機能間の出力の対応関係をメッセージ ID 等で取り持つ。そして、これも BPM の audit 提供機能の一構成要素としてマップされる。更に、図 2.2 で定義されるプロセス監視機能は、図 4.2 では共通管理機能にマップされる。当該共通管理機能は、監視するドメイン領域を定義し、その配下に複数団体が管理するサイト群が位置付けられることになる。そして、このドメイン全てに対して、階層構造を与えることで、スケーラビリティのある管理方式を実現することになる。当該共通管理機能には、プロセス監視コンポーネントにマップされるが故に、図 2.1 で記した仮想プロセス定義クラス、仮想プロセス状態クラス等は、具体的にこの機能内で維持管理されることになる。仮想プロセス定義クラスを維持管理するため、各団体の持つ BPM 機能から、ビジネスプロセス記述である BPEL 定義を収集し、それらを合成する等の処置を行った後、その結果を、各ドメインを管理する共通管理機能間で共有するための基盤が必要となる。この方式が、図 4.1 の方式と異なる点は、共通管理機能の明示的な定義である。

以上からメタモデルや、識別子等の管理基盤を共有しながら、実装方式によって構成を変更することが可能になる、と言える。

5. 考察

前章迄に、新たに提案するメタモデル、並びにそれらを適用することで、複数の形態を選択出来ることを定

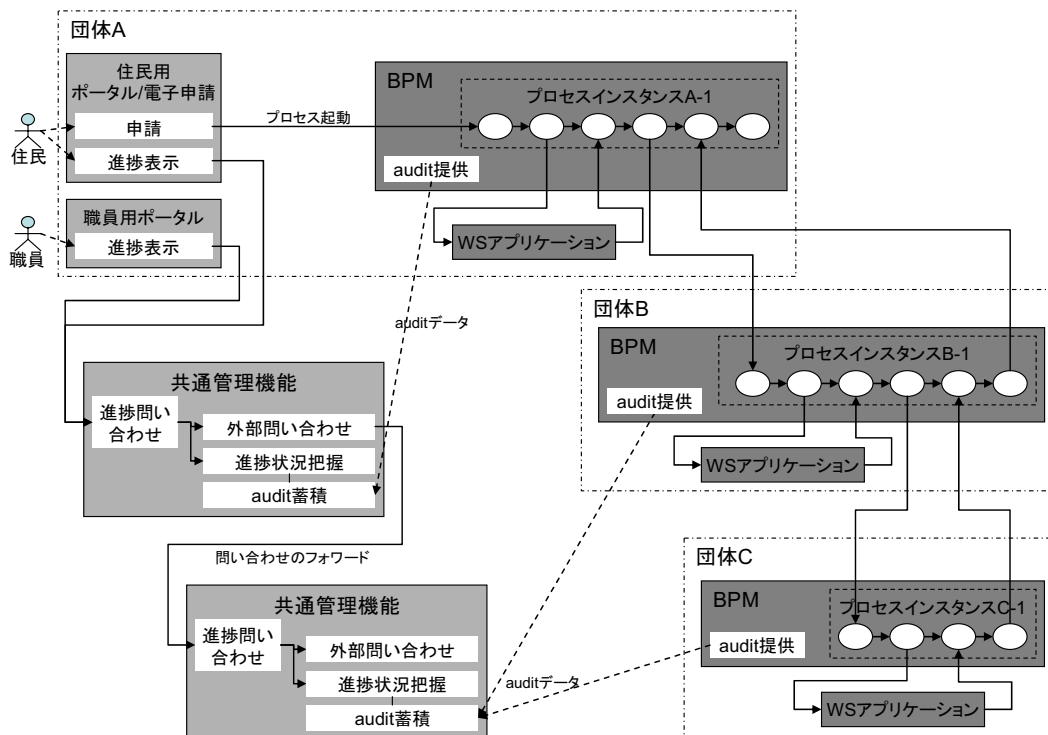


図 4.2 集中方式におけるプロセス情報の抽出

義、説明した。本章では、従来の他モデルと比較した評価、ならびに関連する考察について言及する。

5.1 ITIL との比較評価

サービス提供の運用・監視などシステム運用に関して、ITIL (Information Technology Infrastructure Library)が標準化されている。ITIL はその対象範囲が広いが、モニタリング機能に注目すると、単一サイトの運用管理を中心に定義されており、図 2.2 に記すリソース監視との関連が深い。しかし本稿で説明している様な業務プロセスに関する運用、特に SOA が指向するようなクロスサイトで実行される業務プロセスに関して、必ずしも管理指針が明確に述べられている訳ではない。そのため本稿で定義する事項は、ITIL に対して補完し得る関係にあると言える。

5.2 CIM との比較評価

DMTF (Distributed Management Task Force)の CIM (Common Information Model)は、計算機環境のリソースを広くモデル化しており、ハードウェアリソースのみならずプロセスやアプリケーションに関する情報モデルをも定義している。しかし、これも ITIL と同様に、クロスサイトの業務プロセスの論点からのモデル化は、十二分とは言えず、この点で、図 2.1 に示したクロスサイトプロセスに関するクラス構成は、CIM に対して補完し得る関係にあると言える。

5.3 マルチベンダ対応とその意義

SOA や WS-BPEL を用いて、様々なサービスを連携し、新たなサービスを創造する環境下で、それらを可視化・モニタリングするには、大きく 2つの要求が存在する。第一は、サービス提供体が、完全マルチベンダであるため、これらが提供するサービス群、クロスサイトサービスをモニタリング・可視化する場合、実装に依存しない標準的なマルチベンダ対応の **audit** データを用いることが必須となる。これに対し第二は、モニタリング・可視化を実現する環境自身も、完全にマルチベンダ対応を指向し、複数の要素アプリケーションが様々な観点・ビューで、監視対象のサービスを監視し、関連する情報を提供出来ることである。

第一の立場では、必要に応じて XML Schema による流通形式を定義し、実装環境からフリーにする必要がある。流通形式のセマンティクスを正しく定義把握するためには、図 2.1 に基づくクラス構成が複数アプリケーション間で共有される必要が出て来る。また、第二の立場では、モニタリング・可視化に対する汎用 **Query** 言語、並びに各種 API、バインディング方式等を定義し、ここでも環境依存性を無い様にすることが必要である。この場合、汎用 **Query** 言語のシンタックス、セマンティクの両面で基礎となるものは、前述同様、図 2.1 に基づくクラス構成と

なる。以上の様に、本稿で定義したクラス構成を複数監視アプリケーションで共有し、これに基づく **audit** データ等を、複数監視アプリケーションに提供・利用することで、高付加価値サービスの改善は、より具体的に、且つ容易になる。

6. 結論

本稿では、クロスサイトプロセスの実行状態の把握・捕捉、可視化・モニタリングのため、新たなメタモデルを定義、共有し、それらを用いることで複数の運用監視形態を採り得ることを説明した。加えて当該の多様な形態に於いても、有効に機能する各種識別子に関する要件について説明した。言及したメタモデルは、サービス指向時代の管理フレームワークの基礎をなすものの一つであり、その成果意義は大きいと考えている。今後は更なるブラッシュアップを行うとともに、必要に応じて XML Schema による流通形式等の取り込みを行い、この領域における標準化活動の際の一つの検討候補となることを期待する。

【謝辞】

本研究は、独立行政法人 情報通信研究機構からの委託研究開発「異なる運用ポリシーや異なるアーキテクチャのサービスが連携し、高付加価値サービスを提供できるためのサービス連携基盤技術の研究開発」の成果の一部である。ここに記して謝意を表する。

【参考文献】

- [1]地域情報プラットフォーム基本説明書,
<http://www.jyohoka-suishinky.jp/tech/IDRC/2006/IDRC-0001-2006.pdf>, 2006.
- [2]WS-I(Web Services Interoperability Organization) Basic Profile Version 1.0 Final Material,
<http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>, 2004.
- [3]OASIS Web Services Business Process Execution Language Version 2.0,
<http://www.oasis-open.org/committees/download.php/18714/wsbpel-specification-draft-May17.htm>, (May) 2006.
- [4]IEEE Standard 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems Description.
- [5]サービスサポート, TSO, ISBN 0-11-330950-3, 2003.
- [6]CIM Standard, CIM Schema: Version 2.11,
http://www.dmtf.org/standards/cim/cim_schema_v211, (December) 2005.