

データ連携における インタオペラビリティ達成のための 運用組合せ確認法の提案

矢嶋健一^{†, ††} 落水浩一郎[†]

サービス利用型のソフトウェア構築では、サービスの把握が十分でないことにより、要求の誤解が生じている。この誤解は、データ連携において運用上の問題を引き起こす可能性があるため、実際にサービスを結合する前に解決すべきである。本稿では、データ連携において、サービス間で誤った運用にならないよう確認するための手法を提案する。この手法は、結合するサービス間の機能要求や非機能要求を調整することで要求の誤解を解消し、データ連携の運用手法を決めるインタオペラビリティをゴール分解したカタログを用いて、運用に対する要求の選択肢を明らかにし、その組合せの妥当性を確かめることができる。実際のサービス型開発プロジェクトにこの手法を適用し、その有効性を確認する。

Proposal of Checking Method for Data Operations between Services to Achieve Interoperability

Kenichi Yajima^{†, ††} and Koichiro Ochimizu^{††}

In the service-based software development, misunderstandings of requirements are caused because the grasp of service is not enough. The misunderstandings have possibilities for occurring operationalizing problems and should be solved before actual service coupling. In this paper, we propose a method for checking irrelevant operations between services. The method enables to solve misunderstandings of requirements with reconciling functional requirements and/or non-functional requirements of services, to be explicit options for service operations catalogued with resolving goals of interoperability, and to check validities for combination of service operationalizing goals. We apply the method for actual industrial development project and evaluate effectiveness of the method.

1. はじめに

サービス指向アーキテクチャ (SOA) [1] とインターネットの普及により、近年のソフトウェア構築は、Web サービスのような、ネットワークを介してサービスを利用する形態が主流になってきている。サービスの利用にあたっては、サービスを発行し、それを発見し、組み込むという、サービス提供者とサービス登録者、サービス利用者の相互の役割が提唱されている[2]。サービスに関する情報はサービス登録者に集約されており、利用するサービスに関する情報を収集できる UDDI 技術[3] も存在する。しかしながら、公開型の UDDI は十分に普及しておらず、サービスの利用者と提供者とが直接結合する形態がいまだに一般的である[4]。従って、サービス利用者は提供されるサービスを予め十分に把握する必要がある。この時、機能要求だけでなく、セキュリティや品質属性などの非機能要求 (NFR) についても把握すべきである。特に、サービス結合における NFR のモデル化と取り扱いは重要な研究課題である[5]。

提供されるサービスの把握が不十分だった場合、次の問題が発生する。図 1 は、データ連携において要求に対する誤解が、サービス提供者とサービス利用者との間に、潜在的に存在していることを示している。サービスが実際に運用されると、要求の誤解が顕在化し、問題となる。ここで言うデータ連携とは、サービス提供者からデータを取得し、取得したデータや関連するサービス利用者独自のデータを利用してサービス全体を成立させることである。

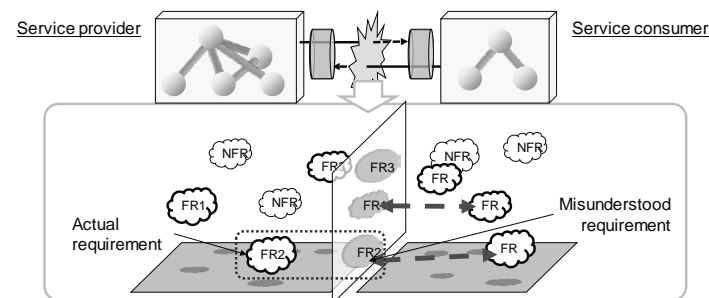


図 1 サービス間の要求の誤解

[†] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
^{††} 株式会社 i.JTB
i.JTB Corp.

サービス把握のための具体的な検討は、インタフェース仕様を用いて実施するのが一般的である。WSDL[6]のように、Web サービスの内容を記述するための、事実上の標準となっている手法も存在する。しかしながら、WSDL ではサービスの運用をデータの入出力の文法でしか表現することができないため、サービスに含まれているデータモデルや、サービス内部でのデータのパターンに対応した操作、そしてその操作がデータに対しどのような影響を及ぼすのかを明らかにすることができないという課題がある。データの操作はサービスの要求に基づいているので、インタフェース仕様からは、その背後にある複数の要求を想定される可能性がある。これは、要求の誤解を招き、データ連携における運用上の問題を引き起こすため、実際のサービス結合前に発見し解決すべきであるが、既存サービスの要求は明示化されておらず、実際の運用を開始して初めて把握することが多い。サービス間でデータをやり取りする以上、このようなデータ連携における運用の問題を、実際のサービス結合の前に解決する手法は不可欠である。これには、サービス提供者とサービス利用者それぞれで定義されたデータ操作に関する要求を比較し、関係づけることにより取り扱われる必要がある。

本稿では、データ連携に関するサービス間の要求を把握し、その組合せから誤った運用にならないように確認するための手法を提案する。サービス間の相互運用を検討する際、サービス間の機能要求と NFR を調整する必要があるため、複数システムを取り扱うために拡張された NFR 型カタログとその適用法を実際のサービス結合に用いる。以下、2 章では拡張 NFR 型カタログと、カタログの基になっている NFR フレームワークを紹介する。3 章では、サービス間での運用組合せを確認するための手順について説明し、4 章でそれを実際の旅行業における Web 予約システム開発プロジェクトの中で、サービス間でやりとりするデータに適用する。更に、旅行業のサービスとは別の事例として、時刻表データとの連携で生じた問題を用いて、本稿で提案する手法を評価する。そして 5 章でまとめを行う。

2. サービスにおける要求

2.1 NFRフレームワーク

サービス間の要求を取り扱うためには、機能的な要求に加えてシステムの性能要求や特定の品質、制約といった特性も考慮し、サービス提供者とサービス利用者それぞれで定義された要求を調整する必要がある。ソフトウェアの要求を定義するための技術は数多くあるが[7]、こういった特性は NFR として分類され[8]、NFR をソフトウェア開発の過程において利用するための、包括的なフレームワークが提案されている[9]。

NFR フレームワークでは、パフォーマンスやセキュリティといった NFR の型が予め階層構造化され、NFR 型カタログとして定義されている[10]。NFR 型カタログでは、NFR とその副特性が矢印を用いて階層構造として表される。一般的な NFR は、詳細

な NFR よりも上位に描かれている。そして、NFR 型カタログを用いて、構造化されたソフトゴール依存グラフ (SIG) が作成できる。ソフトゴールには NFR ソフトゴール、NFR を達成するための要素を表す操作ソフトゴール、そして相互依存関係を説明できる理由ソフトゴールの 3 種類がある。

2.2 拡張NFR型カタログとその利用

サービス結合によってシステムを設計する場合、再利用性 (リユーザビリティ) や、柔軟性 (フレキシビリティ)、インタオペラビリティなど、NFR 型カタログでは取り扱われていないいくつかの NFR 特性を考慮する必要がある。これは、拡張 NFR 型カタログを用いた、連携するシステムに渡るゴールを達成するための取り組みの中で提案されている[11]。

サービス間の要求の誤解を検出し調整するためには、サービス提供者の要求とサービス利用者の要求を、拡張 NFR 型カタログを基にしたそれぞれの SIG として表わす。

まず、双方のサービスでインタオペラビリティや再利用性、柔軟性に関する NFR ソフトゴールを詳細に分解する。そして、操作ソフトゴールの拒絶や満足といった貢献関係とその理由を付加した SIG として表していく。次に、サービス間のインタオペラビリティやその上位、下位のソフトゴールの関係を比較し、関連付けをする。その結果、サービス提供者とサービス利用者の SIG は、サービス間の要求の相互依存関係を表しており、サービス結合に関する基本的な要求の把握を可能にする。このようにしてサービス間の要求の誤解を検出し、誤解を調整するために用いられた拡張 NFR 型カタログを図 2 に示す[12]。このうち、点線で囲まれた非機能特性が、システム連携向けに拡張されたものである。

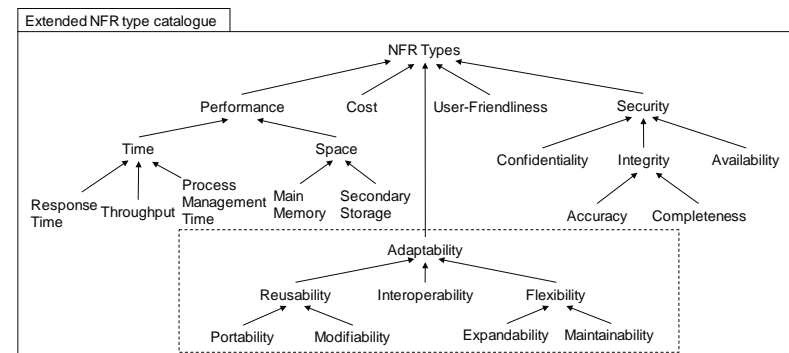


図 2 拡張 NFR 型カタログ

2.3 インタオペラビリティ

要求を把握できないことによる問題は、やり取りするデータの連携に対する運用に影響を及ぼしている。実際に運用を開始すると、サービス提供者が持つ様々なデータやそのデータに対する操作タイミングの影響を受け、様々なサービスの振る舞いが発生する。そして、その振る舞いが考慮されていないと、問題を起こすこととなる。

サービスの振る舞いを考慮するためには、サービス間でやりとりするデータに対し、どのように相互運用するのか、つまりデータのインタオペラビリティを達成する必要がある。インタオペラビリティをゴールとして取り扱おうと、その達成のためには、サービス利用者が要求するデータを提供しやすく、サービス提供者が提供するデータを要求しやすくすることが求められることとなり、これらのサブゴールを更に詳細化することになる。ゴール分解を通じた詳細化により、ゴールを達成するための運用の選択肢が明らかになり、サービスの振る舞いを把握することができる。

サービスにおけるデータのやりとりに関する研究は、いまだ十分に組み立てられていない。拡張 NFR 型カタログにおいても、インタオペラビリティについては詳細な検討が必要である。データのインタオペラビリティを達成するための運用の選択肢が、それぞれお互いのサービスにどのように影響を及ぼすのかを明らかにし、選択された運用が、適切な組合せかどうかを確かめられる手法が必要である。

3. データインタオペラビリティの達成

3.1 データ連携向け運用要求カタログ

サービス提供者とサービス利用者のインタオペラビリティを分解し、達成のための方法を表したデータ連携向け運用要求カタログを図3に示す。

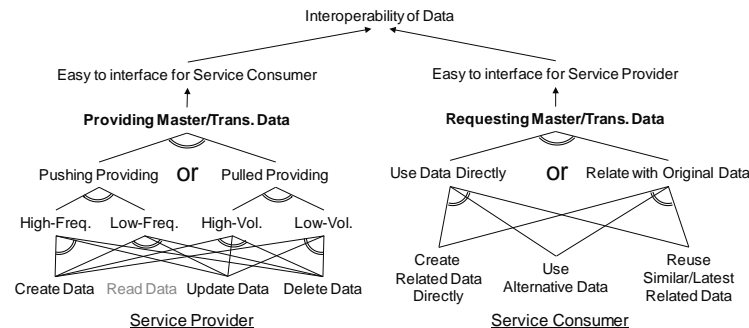


図3 データ連携向け運用要求カタログ

3.1.1 サービス提供者のゴール指向木

図3の左側の部分木は、サービス提供者のデータ運用方法を階層化したものである。提供するデータは、マスタデータとトランザクションデータに分類できるが、どちらのデータも、サービス提供者のタイミングでデータを提供できるプッシュ型の処理と、サービス利用者のタイミングでデータを提供できるプル型の処理の二通りに分解される。

プッシュ型のデータ提供の場合、どのくらいの頻度でデータ提供されるのかによりサービス利用者の運用が変化するため、高頻度のデータ操作によるものと、低頻度の操作によるものとに二分する。プル型の場合、提供の頻度は利用者が制御できるが、一回に提供されるデータの量は制御できないため、データの量がサービス利用者の運用に変化を及ぼす。従って、多量のデータ提供によるものと、少量のデータ提供によるものとに二分する。

提供されるデータは、データに対する操作の影響を受ける。その操作には、CRUD(作成、読み取り、更新、削除)があるが[13]、このうち読み取りでは、提供されるデータには何も変化がないため、データ連携に影響を及ぼさない。従って、データの提供は、提供されるデータに影響を及ぼす操作となる作成、更新および削除に分解される。

3.1.2 サービス利用者のゴール指向木

図3の右側の部分木は、サービス利用者のデータ運用方法を階層化したものである。提供されるデータは、マスタデータとトランザクションデータに分類できる。サービス提供者のゴール分解により、その振る舞いを把握した結果、もし提供されるデータ上に想定していた情報が存在しないと、インタオペラビリティ達成のために、サービス利用者は何らかの情報補完処理という運用が必要となる。また、提供されるデータを関連するサービス利用者独自のデータと結び付けた上で利用する場合、もし提供されるデータや、サービス利用者独自のデータに必要な情報が存在しないと、同様の補完処理が必要となる。このような不整合に対する補完処理は、必要なデータを直接作成できればよいが、その作業が負荷となる可能性もあり、負荷を軽減させるために代替となる簡易なデータの利用、そして既存の類似または直近データの流用も合わせて考えられる。

3.2 データ連携向け運用確認表

データ連携向け運用要求カタログでは、サービス利用者とサービス提供者のデータに対するインタオペラビリティの達成に必要なゴールが分解された。これにより、サービス間の運用方法において、いくつかのサブゴールの組合せが出来上がる。組合せの中には運用上問題のないものや、逆に効果が十分でなく、問題に陥るものもある。従って、ここではその組合せについて検討する。

(1) 提供されるデータの量が少ない場合

データの量が少なければ、提供されるデータ上に想定していた情報が存在しないことに対し、サービス利用者が必要な情報を一つ一つ作成するとしても、作業負荷は

大きくなく対応可能である。これは、プッシュ型のデータ提供における低頻度の際も同様である。プッシュ型では、サービス提供者のデータに対する操作に応じてデータが提供されるので、一回あたりのデータ提供量は少ないと想定され、頻度が低ければ必然的に提供されるデータ量も少なくなるからである。

(2) 提供されるデータの量が多い場合

提供されるデータの量が多いとなると、必要な補完データを直接作成することは作業負荷が大きく、別の手段を検討する必要がある。プッシュ型による高頻度のデータ提供も、結果的に提供されるデータの量は多くなるため、同様である。提供されるデータの量が多い場合、そのデータをサービス利用者がそのまま用いるのか、それともサービス利用者が持つ独自データと、何らかの連携をするのかによって、データの補完方法が変わってくる。

- 提供されるデータをサービス利用者がそのまま用いる場合
 これは、提供されるデータを画面や帳票にそのまま用いたり、データの値をサービス全体の動作を制御するために用いたりする場合を指している。この場合の補完処理として、予め代替データを利用するように施しておけば、サービス全体の運用を維持することが可能となる。
- 提供されるデータを、サービス利用者の独自データと連携させる場合
 これは、提供されるデータをサービス利用者が保持するデータと結び付け、双方のデータを利用してサービス全体の動作を制御する場合を指している。もし、提供されるデータ上に想定していた情報が存在しない、もしくは提供されるデータと結び付けるサービス利用者の独自データとの間に不整合があり、結び付けの見直しが必要となると、補完処理としてどのデータと結び付けるべきなのかを決定しなければならない。従って、直近のデータに対して結び付ける、もしくは、提供されるデータの他の情報を用いて、結び付けできる類似データを発見し、それと結び付けることによりサービス全体の運用を維持する。

表 1 データ連携向け運用確認表

Master/Trans.データ取得		提供されるデータの運用	
		そのまま利用	サービス利用者の独自データと連携
データ提供 する	プル型	少量	必要なデータを
	プッシュ型	低頻度	直接作成
	プル型	多量	代替となる
	プッシュ型	高頻度	簡易データの利用

以上の検討結果を、データ連携向け運用確認表として表 1 に示す。

3.3 利用手順

データ連携向け運用要求カタログと運用確認表を利用するための流れを図 4 に示す。

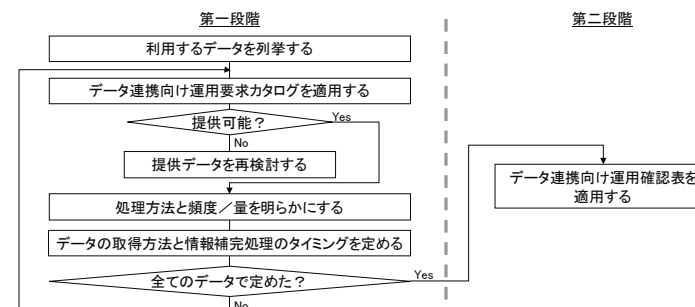


図 4 データ連携向け運用要求カタログとデータ連携向け運用確認表の利用手順

第一段階では、データ連携向け運用要求カタログに基づいて、サービス提供者の運用方法を明示化し、サービス利用者の運用方法を定義する。まず、サービス利用者が利用するデータを列挙する。次に列挙したデータについてデータ連携向け運用要求カタログを適用し、データ提供に関する運用がどのように達成されるのかを明らかにする。そして、サービス提供者がそのデータを提供可能であれば、提供データの処理方法と頻度/量を明らかにする。更にサービス利用者は、提供されるデータの取得方法と、取得したデータや関連する利用者独自のデータに対する情報補完をどのように処理するかを定める。これらの一連の手続きを列挙したデータ全てに対し実施する。

第二段階では、明示化されたサービス提供者の運用と、定義されたサービス利用者の運用が最適な組合せかどうかを確かめる。これは、第一段階の手続きで決まったデータ提供のための処理方法と頻度/量、データの取得方法や取得したデータに関連する連携データの処理方法を用いる。運用組合せに関する確認表から、それらの組合せの評価が得られるが、もし組合せが適切でない場合、どういった問題が運用開始後に生ずるかを想定し、サービス提供者や利用者の運用を変更するなど、対策を講じる。

4. 事例研究

4.1 システム概要

本章では、JTB の旅行オンライン販売 Web サイト[14]の開発プロジェクトにおいて発生した問題を、事例研究として取り上げる。この問題は、従来、アドホックに処理されていたが、本稿で述べたモデルを利用して再解決に導いた。すなわち、サービス

運用に関する要求の問題を、データ連携向け運用要求カタログと運用確認表を適用して解決し、合わせてその有効性を確認した。

この旅行オンライン販売システムは、販売のための在庫を自らが保持しておらず、旅行商品を造成卸売するシステムと直接結合し、在庫を取得している。また、在庫以外にもいくつかのデータをオンライン販売システムから取得している。このシステム構成を図5に示す。

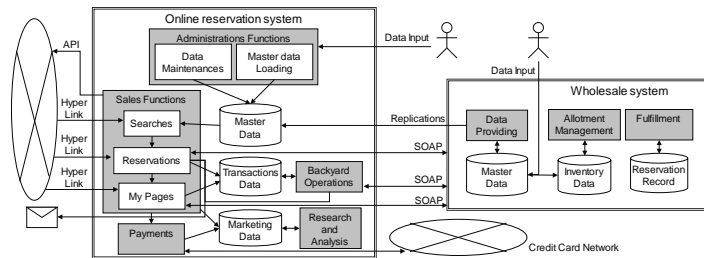


図5 旅行販売のオンラインシステムと造成卸売システムの構成

旅行販売の場合、例えばホテル予約のように、販売可能な期間や在庫数に限りがあるため、集中管理のためにシステムやサービスの連携が広く採用されている。旅行業のシステムは、サービス連携を有効活用している典型的な例だと言える。

4.2 システム要求

オンライン販売システムと造成卸売システムの、運用に関する要求を詳細検討する前に、システム全体の要求を示す。インタフェース仕様に基づいた要求定義では、269の要求が定められ、その中には、機能要求とNFRとが含まれていた。これらの要求から、拡張NFR型カタログに基づいてSIGを作成し、運用上の問題を生じさせないように、提供するデータに対するサービスの振る舞いを明らかにする。サービス間でやり取りする主要なデータは、予約用の在庫データ、旅行代金データ、空港マスターデータ、そしてツアー日程表のデータであった。図6は、データ連携向け運用要求カタログに従い作成した、インタオペラビリティを達成するためのSIGである。細線の雲型はNFRソフトゴールを表し、太線の雲型は操作ソフトゴールを表している。定義された操作ソフトゴールは、サービスの持つ様々なデータに対する振る舞いに影響を及ぼす。

予約用在庫データは、旅行の予約や取消操作に従いその数が増減する。これは、造成卸売システムのユーザや、他のサービスからも頻繁に操作される。オンライン販売システムは、造成販売システムから取得した在庫数に応じてその表示内容を制御する。

旅行代金データについて、一旦決定した代金を変更するケースは極めて稀ではあるが、代金を修正した際にサービス間で差が生じてしまうと、業務への影響は深刻であ

る。従って、オンライン販売システムは、取得した最新の旅行代金データをそのまま表示し、予約時のクレジット決済に利用する。

空港マスターデータは、旅行商品を造成卸売する際に、空港を誤って登録しないように制御するためのもので、商品登録よりもかなり前の段階で登録される。オンライン販売の際には、このデータを空港に関する案内のために用いるが、他の情報の補足が必要なので、それをオンライン販売システムで作成、更新、削除し、造成卸売システムのデータと結び付けて利用する。

ツアー日程表データは、在庫や旅行代金に比べ、柔軟なデータ運用をしており、同じツアーでも出発日毎に様々な日程表へと更新される。ツアー日程表データは造成卸売の専門用語を使っており、オンライン販売システムで使用するためには、関連する詳細情報の補足が必要である。補足のための詳細情報はオンライン販売システムで作成、更新、削除し、造成卸売システムから提供されたデータと結び付けて利用する。

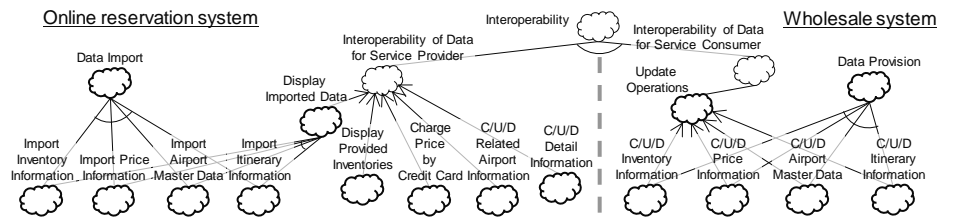


図6 オンラインシステムと造成卸売システムのデータのインタオペラビリティ SIG

4.3 適用結果

図4で示した利用手順に従い、データ連携向け運用要求カタログと運用確認表を適用した。この結果を表2に示す。実際のプロジェクトでは、要求定義に基づき、提供されるデータに関連するデータの運用法が定義されていた。これも合わせて示す。

本稿で提案した手法を適用したところ、ツアー日程表データに対するプロジェクトでの運用定義は、データ連携向け運用確認表の適用結果とは別の運用手法を検討すべきものであるという結果が得られた。

プロジェクトでは、ツアー日程表の補足情報データが、当初直接作成され連携していた。サービス利用者は、サービス提供者との間でデータ運用について検討したものの、その検討時点ではサービス提供者の運用が十分に定まっていなかったことから、サービス利用者はサービス提供者の運用を想定した上で自らの運用を定義し、もし問題があれば、運用開始後にプログラム修正するという回避計画を用意した。サービス利用者が定義した運用は、サービス利用者が必要なデータを直接作成して対応しようというものであった。実際には、サービス利用者がデータを直接対応しきれない、大

量のデータ更新が発生し、問題となった。問題発生時の回避計画として用意していた運用開始後のプログラム修正も、運用しながらの対応ということで難航し、問題の解消までに長い時間を要することとなった。その結果、提供されたデータは、結び付けるデータがサービス利用者に存在しないことから表示できなくなり、表示可能なデータが減少を続け、システムが機能しない事態に陥った。

表 2 データ連携向け運用要求カタログとデータ連携向け運用確認表の適用結果

データ種類	サービス提供者の運用		サービス利用者の運用	運用確認表の適用結果	プロジェクトでの定義
予約在庫	プッシュ	高頻度	そのまま利用	代替となる簡易データの利用	代替となる簡易データの利用
旅行代金	プッシュ	低頻度	そのまま利用	必要なデータを直接作成	必要なデータを直接作成
空港マスタ	プル	少量	独自データと連携	必要なデータを直接作成	必要なデータを直接作成
ツアー日程表	プル	多量	独自データと連携	類似または直近データの流用	必要なデータを直接作成

問題の解決にあたっては、3.2 章で述べたデータ連携向け運用確認表の考え方を利用している。その結果、運用の見直しを達成し、問題の状況から回復することができた。つまり、本稿で提案した手法は有効であった。

予約在庫データ、旅行代金データ、そして空港マスタデータの連携に対して、実際のプロジェクトの中で定義された運用方法は、データ連携向け運用確認表を適用した結果と同じであった。実際の運用でも問題はなく、サービスは順調に稼働している。

4.4 評価

ここまでは、旅行業における情報提供サービスのデータ連携に対し、データ連携向け運用要求カタログと運用確認表を適用した。旅行業以外のサービスでも、情報開示が可能なサービス同士で、業務に変化が多く連携するサービスの要求が把握しづらい分野では、カタログと確認表は有効であると考えられる。そこで、旅行業のサービスとは別の事例として、時刻表データとの連携で生じた問題を用いて、データ連携向け運用要求カタログと運用確認表を評価する。

この問題は、出版系のシステムのひとつである時刻表のシステムとのデータ連携において発生したものである。出版業のシステムは、情報を紙媒体に変換する役割を担っているが、その際、業務が変換対象の情報に左右されることから、業務が情報に大きく影響を受けるという特徴がある。問題は、時刻表のシステムとデータ連携する際、

インタフェース仕様にある「到着時刻」という項目の値を利用しようとシステムを設計したところ、値が存在しない場合があり、サービス利用者側の機能が正常に動作しなかったことである。原因は、提供者がこのデータを本来出版向けに利用しており、紙面の都合で「到着時刻」を使ったり使わなかったりするという、状況に応じ柔軟に対応するという要求があり、その要求に従うと、値があつたりなかったりするという事象が、利用者側に伝わっていなかったことにあった。原因究明にあたっては、問題をデータに対する要求に立ち戻って検討するため、SIG を用いた。これを図 7 に示す。

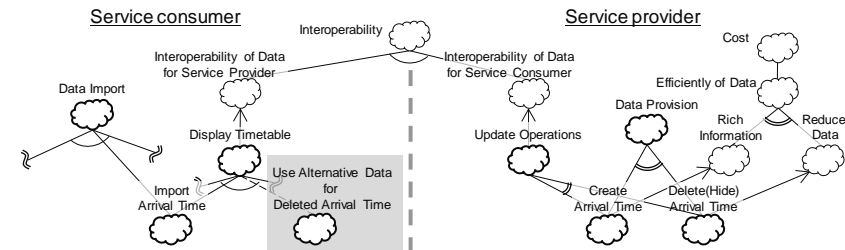


図 7 時刻表データの提供・利用におけるデータインタオペラビリティ SIG

Figure 7 SIG of Interoperability for Timetable Data.

問題は、時刻表のいくつかの本番データを実際に使ったテストにおいて発覚した。これは、単に値が存在しなかったただだが、そのことがテストするまで分からなかった理由を分析すると、「到着時刻」がインタフェース仕様にあったため、利用者側は必要な時にいつでも値があると要求定義の段階で思い込んでしまい、データ作成の裏にある業務の確認まで及ばなかったこと、一方、提供者側はインタフェース仕様で利用者が業務を理解したと思い込んだことにあった。

実際の問題解決にデータ連携向け運用要求カタログと運用確認表を使用したところ、プル型多量のデータ提供に対し、データ利用者は取得したデータをそのまま表示に利用することから、想定外のデータに対し代替となる簡易データを使うことが推奨された。この結果を表 3 に示す。この結果に従い、値が存在しないなど想定外の場合、文言データを代替データとして表示する対応を行うことで問題を解決し、無事サービス運用開始となった。このゴールは、図 7 の灰色の領域に示されている。

表 3 時刻表データにおける運用要求カタログと運用確認表の適用結果

データ種類	サービス提供者の運用		サービス利用者の運用	運用確認表の適用結果	プロジェクトでの定義
	到着時刻	プル	多量	そのまま利用	代替となる簡易データの利用

もし問題の原因が要求定義工程にあれば、その問題は要求定義工程で解決できることが望ましい。提案した手法は、時刻表データの「到着時刻」に関する思い込みを、上述の通りテスト段階で解決した。解決に利用した情報は、要求定義段階で決まっていたものであり、この問題解決に必要な条件は、テスト段階と要求定義段階で差がなかった。従って、同じ手法を要求定義段階で実施すれば、要求定義工程でも問題を解決することが可能である。

以上のことから、提案したデータ連携向け運用要求カタログと運用確認表による手法は、提供されるデータを要求定義の段階で正確に把握するために有効だと言える。

5. おわりに

本稿では、サービス間で生ずる運用上の問題がデータに起因していることに着目し、サービス間の運用に対する要求であるインタオペラビリティをゴール分解することにより作成したデータ連携向け運用要求カタログ、運用組合せを検証できる確認表、およびそれらの利用手順を提案した。データ連携向け運用要求カタログでは、サービス提供者の運用方法を明示化してサービス利用者側の運用方法を定義し、その組合せが最適かどうか、運用組合せに関する確認表を用いて検証できる。

この提案手法を実際の事例に適用し、旅行販売システムと造成卸売システムの連携における要求を分析し詳細化した際、提供されるツアー日程表や詳細情報に対するデータの追加更新で、定義した運用手法が最適かどうかを検証することができた。更に、旅行業のサービスとは別の事例として、時刻表データとの連携で生じた問題を用いて、提案手法を評価することができた。

本稿で提案した手法は、サービス利用者からサービス提供者の運用を把握しづらい場合に有効で、一例としてサービス提供者が、利用者と並行して開発されている場合や、異種のサービス提供者とサービス利用者が、広い範囲で業務連携する場合が挙げられる。サービス連携を有効活用している典型的な例である旅行業のシステムだけでなく、幅広い分野でサイトが連携する Web システムでのサービス連携においても有効であると考えられる。今後はこういった有効活用できる分野での適用に関する研究に取り組むことにより、この手法をより広範囲で利用できるよう進めていく予定である。

参考文献

- 1) Erl, T.: *SOA: Principles of Service Design*, Prentice Hall, 2007.
- 2) Papazoglou, M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions, *Proc. 4th International Conference on Web Information Systems Engineering (WISE2003)*, pp.3-12, 2003.
- 3) Organization for the Advancement of Structured Information Standards, Universal Description, Discovery, and Integration (UDDI), <http://uddi.xml.org/>
- 4) Michlmayr, A., Rosenberg, F., Platzer, C., Treiber, M. and Dustdar, S.: Towards Recovering the Broken SOA Triangle - A Software Engineering Perspective, *Proc. 2nd International Workshop on Service Oriented Software Engineering (IW-SOSWE'07)*, pp.22-28, 2007.
- 5) Kontogiannis, K., Lewis, G.A., Smith, D.B., Litoiu, M., Muller, H., Schuster, S. and Stroulia, E.: The Landscape of Service-Oriented Systems: A Research Perspective, *Proc. International Workshop on Systems Development in SOA Environments (SDSOA'07)*, pp.1, 2007.
- 6) World Wide Web Consortium, Web Services Description Language (WSDL), <http://www.w3.org/TR/wsd1>
- 7) van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour, *Proc. 5th IEEE International Symposium on Requirements Engineering (RE'01)*, pp.249-263, 2001.
- 8) Chung, L. and Sampaio de Prado Leite, J.C.: On Non-Functional Requirements in Software Engineering, *Lecture Notes in Computer Science*, pp.363-379, 2009.
- 9) Mylopoulos, J., Chung, L. and Nixon, B.: Representing and Using Non-Functional Requirements: A Process-Oriented Approach, *IEEE Trans. Softw. Eng.*, Vol.18, No.6, pp.483-497, 1992.
- 10) Chung, L., Nixon, B., Yu, E. and Mylopoulos, J.: *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, 2000.
- 11) 矢嶋健一, 落水浩一郎: NFR フレームワークにおけるシステム連携向け拡張 NFR 型カタログの提案, 日本ソフトウェア科学会 ソフトウェア工学の基礎ワークショップ (FOSE2009), pp.289-296, 2009.
- 12) Yajima, K. and Ochimizu, K.: Reconciling Misunderstandings of Requirements: An Experience of NFR Framework in Service Coupling, *17th Asia Pacific Software Engineering Conference (APSEC2010)* - Accepted, 2010.
- 13) Kilov, H.: From semantic to object-oriented data modeling, *Proc. the First International Conference on Systems Integration on Systems Integration '90*, pp.385-393, 1990.
- 14) JTB Corp., JTB Official Website, <http://www.jtb.co.jp/>