

Title	データ連携におけるサービス運用要求の把握によるインタオペラビリティの達成
Author(s)	矢島, 健一; 落水, 浩一郎
Citation	情報処理学会論文誌, 52(9): 2799-2811
Issue Date	2011-09-15
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/10877
Rights	社団法人 情報処理学会, 矢島健一, 落水浩一郎, 情報処理学会論文誌, 52(9), 2011, 2799-2811. ここに掲載した著作物の利用に関する注意: 本著作物の著作権は(社)情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。 Notice for the use of this material: The copyright of this material is retained by the Information Processing Society of Japan (IPSJ). This material is published on this web site with the agreement of the author (s) and the IPSJ. Please be complied with Copyright Law of Japan and the Code of Ethics of the IPSJ if any users wish to reproduce, make derivative work, distribute or make available to the public any part or whole thereof. All Rights Reserved, Copyright (C) Information Processing Society of Japan.
Description	



データ連携におけるサービス運用要求の把握による インタオペラビリティの達成

矢 嶋 健 ^{†1,†2} 落 水 浩 一 郎 ^{†1}

サービス利用型のソフトウェア構築では、インタフェース仕様を用いて、結合するサービスを十分に把握したとしても、そのサービスの基になっている要求に対して誤解を生ずる可能性がある。この誤解は、サービス間のデータ連携において、運用上の問題を引き起こすため、サービス利用を検討している段階で解決すべきである。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。本手法は、サービス間のデータのインタオペラビリティをゴールとして取り扱うことにより、サービス提供者とサービス利用者の、相互運用に対する要求の選択肢を明らかにする。これにより、提供されるサービスにおけるデータの複雑さや、サービスの想定外の振舞いを考慮できる。実際のサービス型開発プロジェクトに本手法を適用し、その有効性を示す。

Achieving Interoperability for Data Operations between Services with Understanding Operational Requirements

KENICHI YAJIMA ^{†1,†2} and KOICHIRO OCHIMIZU ^{†1}

In the service-oriented software development, there are possibilities to misunderstand service requirements even if coupling services are understood sufficiently with interface specifications. The misunderstandings cause for occurring operational problems and should be solved when service coupling designs are considered. In this paper, we propose a method for checking inefficient operations between services. The method makes options of service operations for the service provider and the service consumer be explicit with resolving goals of interoperability, and enables to consider complexities of data and unexpected behavior in provided services. We apply the method for actual service development project and show 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。サービスとは、ビジネスのニーズに合わせた規模で開発された、アプリケーションを単位とするソフトウェア部品で、それらを組み合わせることにより、システムが構築される。サービス提供者がサービスを発行し、それをサービス登録者に登録し、サービス利用者がそれを発見し、新たなビジネスにそのまま組み込むという、相互の役割が提唱されている²⁾。サービスに関する情報をサービス登録者に集約し、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⁾。サービスを問題なく利用するためには、サービス利用者が、提供されるサービスをあらかじめ正確に把握する必要がある。なぜならば、サービス自体はカプセル化されて提供されるため、サービスの利用にあたって想定外の現象が発生し、システム全体の動作に問題を生ずる恐れがあるからである。一般に、提供されるサービスを利用する際、サービスどうしがやりとりするのはデータなので、想定外の現象は、あらかじめ想定していたデータとは異なるデータを授受するなどの事象によって顕在化する。

サービスを把握する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り、WSDL⁵⁾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、サービスに包含さ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などを別途明示的に記述する必要がある。サービス内部におけるデータの操作は、サービスに対する運用の要求に基づいたものである。したがって、サービスの結合にあたっては、利用するサービスを誤解しないよう、どのような要求に基づいているのかを明らかにする必要がある。

図 1 は、上段に結合されるサービスを、下段にサービスの基となっている要求をそれぞれ

^{†1} 北陸先端科学技術大学院大学

Japan Advanced Institute of Science and Technology

^{†2} 株式会社 i.JTB

i.JTB Corp.

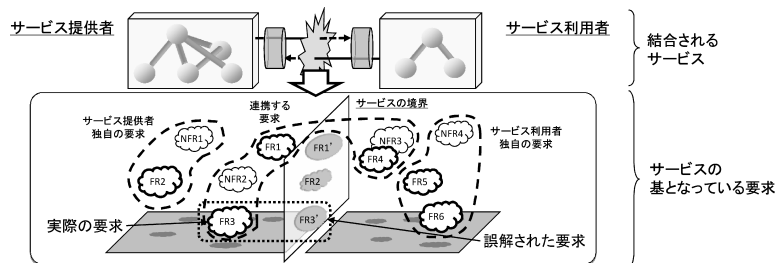


図 1 サービス間の要求の誤解

Fig. 1 Misunderstandings of requirements between services.

れ例示している。構築するサービスは、非機能要求 NFR1 から NFR4 と、機能要求 FR1 から FR6 で構成されている。ここで機能要求とは、サービスの利用者がサービスに対して果たしてほしいと望んだ役割を表したものであり、非機能要求とは、機能要求を選択するための判断に必要な選択基準である⁶⁾。破線で囲まれた部分は、それぞれの要求がサービス提供者によるものか、サービス利用者によるものか、双方が連携する要求なのかを示したものであり、点線で囲まれた部分は、サービス利用者から見た機能要求 FR3' が、実際の機能要求 FR3 と異なって見え、提供されているサービスが誤解されていることを示したものである。たとえば、サービス提供者が想定しているデータの更新タイミング (FR3) と、サービス利用者が想定しているデータの更新タイミング (FR3') が異なると、実際のサービス結合において、想定とは異なるデータをやりとりする可能性があり、問題となる。

従来、このような問題は、システムテストや実際の運用開始後に把握されることが多かった。本来、このような問題は、利用するサービスを検討している段階、すなわち、提供されるサービスを利用してシステム全体をどのように構築するのかを検討するソフトウェア開発の上流工程で解決すべきである。このとき、サービス提供者とサービス利用者がそれぞれ定義したデータ操作に関する運用の要求を比較してその関係を明らかにし、システム全体の要求を達成するよう調整する必要がある。システムを構成する要求の組合せは、機能要求どうしや非機能要求どうしだけとは限らない。一方のサービスの機能要求を選択するために、もう一方のサービスにおける非機能要求が関係するような、サービスをまたいだ機能要求と非機能要求の組合せになる場合も考えられる。したがって、サービス間の要求を関係付け、調整する場合、機能要求と非機能要求の関係も合わせて明らかにすることが必要である。

本稿では、サービス運用に支障をきたす想定外データの問題を対象にして、サービス結合

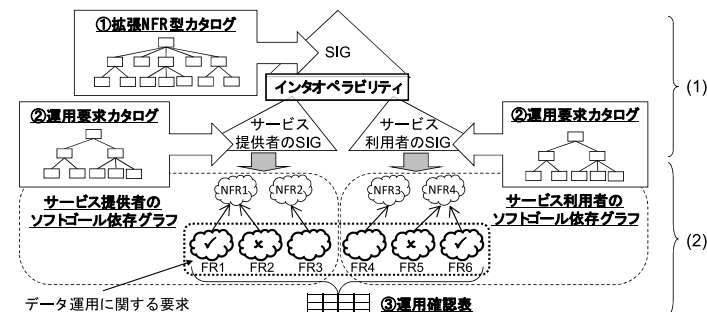


図 2 サービス間の要求の明示と運用定義のための手法

Fig. 2 A method to be explicit for requirements and to define service operations.

における要求間の関係を明らかにし調整することで要求の誤解を解消し、サービス提供者とサービス利用者の間のデータ運用に関する要求の組合せの妥当性を、要求定義の段階で判断できる手法を提案する⁷⁾。図 2 に、本手法の概念を示す。

本手法は、サービスを利用するソフトウェア・システムの設計者が、まず、① 拡張 NFR 型カタログを用いてソフトゴール依存グラフ (SIG) にサービス連携に関する要求を定義する。たとえば、再利用性やインタオペラビリティといった非機能要求や、それらを達成するための機能要求を定義する。次に、② データ連携向け運用要求カタログを用いて、サービス提供者、サービス利用者それぞれに、インタオペラビリティを達成するためのデータ連携に関する要求を SIG に追加する。たとえば、データの更新タイミングに関する機能要求を定義する。最後に、③ データ連携向け運用確認表を用いて、それぞれの SIG に定義されたサービス提供者とサービス利用者のデータ運用に関する要求の組合せの妥当性を判断するというものである。ここでデータ連携とは、サービス利用者が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、サービス全体を運用することである。

本手法の利用対象として、旅行業のサービスを取り上げる。旅行業は、情報を基に付加価値を提供するという特徴を持つ情報産業である。業務が情報に左右されるため、業務の要求仕様が把握しにくく、開発中のサービス仕様の変更が多い。提案する手法は、このようにサービスの要求を把握しにくい分野で有効であると考えられる。そこで、旅行業における情報提供機能の事例を用いてその有効性を検証する。

以下、2 章ではサービス連携のための拡張 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の妥当性を確認するための手順を説明し、4 章でそれを実際の旅行 Web 予約システム開発プロジェクトの中で、サービス間のデータ運用に適用した結果を示す。さらに、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、提案する手法を評価する。5 章で関連する研究を紹介し、6 章でまとめを行う。

2. サービスにおける要求

2.1 ゴール指向と NFR フレームワーク

ソフトウェア・システムの要求を定義する手法は、すでに数多く提案されており、ゴール指向による手法もその 1 つである⁸⁾。ゴールとは、ソフトウェア・システムが達成すべき目標である。ゴール指向による手法は、ゴールを設定し、それを分解することによりソフトウェア・システムが達成すべき要求を定義するものである。一般にゴールは複数個あり、お互いに連携をしているため、ゴール指向による手法を用いることで要求間の関係を把握することが可能である。たとえば、提供されるサービスの要求を表したゴールとそれを利用するサービスの要求を表したゴールが、どのように連携するのかを明らかにすることで、要求の関係を把握することが可能となる。つまり、ゴール指向による手法は、サービスどうしの連携する要求を関係付け、調整することに活用できると考えられる。

代表的なゴール指向による手法として KAOS 法、i* (アイスター) 法、NFR フレームワークなどがある⁸⁾。KAOS 法は、ゴールによる記述内容の正当性や、記述したゴール間の無盾性を検証する点に特徴を持った手法であり、i* 法は、人やシステム、役割を表したアクタを中心に、他のアクタへの依存関係や関心を設定していくことにより、システム全体の要求をゴールとして構成していく手法である。これらに対し、NFR フレームワークは、ソフトウェア・システムのゴールである非機能要求から、ユーザのゴールである機能要求に展開し、展開した要求どうしの関係に対し、理由を付加することでシステム全体のゴールを構成していく手法である⁶⁾。

サービス間の要求を取り扱うためには、個々のサービスにおける機能的な要求どうしや、システムの性能要求や特定の品質、制約といった特性を表す非機能要求⁹⁾ との関係だけではなく、サービスをまたいで、一方のサービスの非機能要求を実現するために、もう一方のサービスにおける機能要求が関係するような組合せも考えられる。NFR フレームワークでは、非機能要求と機能要求をゴールにより関係付けていく手法であり、サービスどうしの非機能要求・機能要求の関係を明らかにするという必要性に対応可能であるという優位点を備

えている。

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非機能要求の型があらかじめ階層構造化され、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、非機能要求とその副特性が矢印を用いて階層構造として表される。一般的な非機能要求は、詳細な非機能要求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 SIG が作成できる。ソフトゴールには、非機能要求を表す NFR ソフトゴール、非機能要求を達成するための機能的な要素を表す操作ソフトゴール、そしてソフトゴールどうしの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

図 1 におけるサービス提供者とサービス利用者はそれぞれ独立しているため、その要求も独立している。しかしながら、サービスは運用の段階で別のサービスと連携することによりその役割を果たすので、独立しているそれぞれのサービスの要求を、運用に支障をきたさないよう連携させる必要がある。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(リユースビリティ)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非機能特性を考慮、調整する必要がある。このことは、拡張 NFR 型カタログを用いた、連携するサービスにわたるゴールを達成するための取り組みの中で提案した^{11),12)}。この拡張 NFR 型カタログを図 3 に示す¹²⁾。カタログ

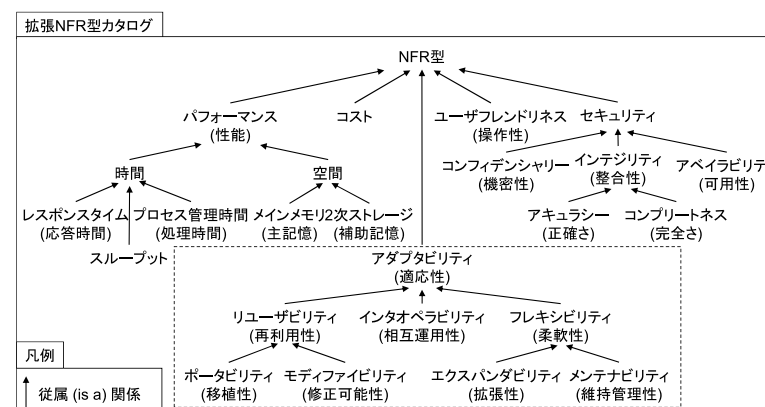


図 3 拡張 NFR 型カタログ

Fig. 3 Extended NFR type catalogue.

の中で、点線で囲まれた非機能特性が、サービスの連携向けに拡張されたものである。

サービス間の要求の関係を明らかにし、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 に定義し、それらの関係を明示的に表現する必要がある。そのためには、まず、双方のサービスにおいて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、操作ソフトゴールによる満足や否定といった貢献関係とその理由を付加した SIG を構築していく。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を比較し、関係を明示化する。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示しており、サービス結合に関する基本的な要求の把握と、誤解の解消を可能にする。

2.3 インタオペラビリティ

サービス間の要求に関する誤解は、データ授受の運用に影響を及ぼす。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舞いが発生する。そして、その振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。

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の作成

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、インタオペラビリティ達成のための手段を表したデータ連携向け運用要求カタログを図 4 に示す。インタオペラビリティ達成のためには、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。この 2 つのゴールをサブゴールとし、上位ゴールを達成するように詳細化し、ゴールを達成するために必要となる運用の要求を明らかにし、カタログ化する。

3.1.1 サービス提供者のゴール

図 4 の左側は、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、ソフトウェア・システムが果たす処理を通じ、データが追加、更新、削除されることのないマスターデータと、ソフトウェア・システムの処理に従って追加、更新、削除といった操作が行われるトランザクションデータのいずれかに分類できる。マスターデータの例とし

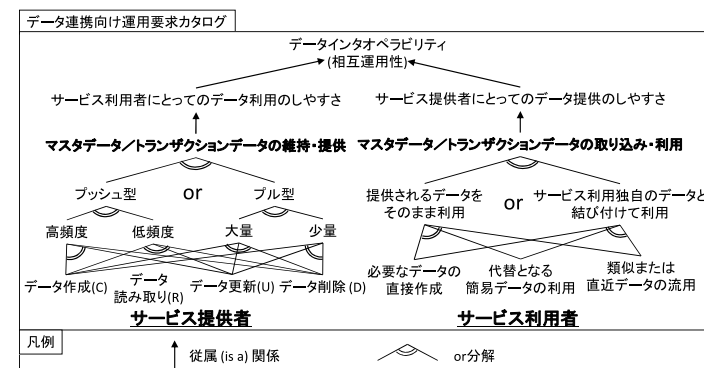


図 4 データ連携向け運用要求カタログ

Fig. 4 Data operational type catalogue.

て、旅行の方面コードや空港コードといったデータがあり、トランザクションデータの例として、旅行の予約記録や、日程データがある。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す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するプル型の処理のいずれかに分類できる。

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のいずれかに分類する。プル型の場合、提供の頻度は利用者が制御できるが、1 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。したがって、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のいずれかに分類する。

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、提供す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。したがって、データの提供は、提供す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類する。

3.1.2 サービス利用者のゴール

図 4 の右側は、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、マスターデータとトランザクションデータに分類できる。これらのデータを利用する際の一連の流れを図 5 に示す。

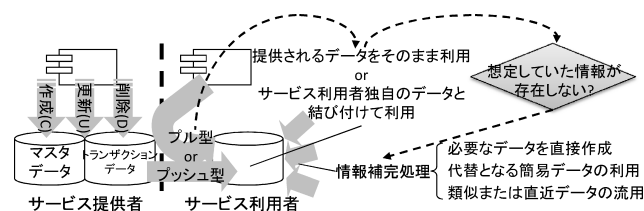


図 5 サービス間のデータ運用の流れ
Fig. 5 Data operations between services.

図 5 において、サービス利用者は、データを利用する際、提供されたデータをそのまま利用するのか、それとも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うえで利用するのかのいずれかに分類できる。これらをサブゴールとして、図 4 のカタログのサービス利用者側のゴールに表す。サービス提供者におけるデータ操作の結果、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かった場合、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処置が必要となる。また、提供されるデータを、サービス利用者独自のデータと結び付けたうえで利用する場合、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なると、同様の処置が必要となる。このようなデータの不整合に対する処置を情報補完処理と呼び、それには 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、その場合負荷を軽減するために代替となる簡易なデータ を利用する、もしくは既存の 類似または直近データを流用 することとなる。そこで、これら下線部の処置を、図 4 のカタログのサービス利用者側のゴールに、3 つのサブサブゴールとして表す。

3.2 データ連携向け運用確認表の検討

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定義され、運用に関する要求がカタログ化された。これにより、サービス提供者、サービス利用者の運用に関する要求の組合せが検討できる。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。したがって、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對し、サービス利用者が必要な情報を 1 つ 1 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、1 回あたりのデータ提供量は少ないと想定され、頻度が低ければ必要な情報を 1 つ 1 つ作成する作業負荷も必然的に小さくなるからである。

(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。このとき、もし提供されるデータ上に想定していた情報が存在していなくても、情報補完処理として、あらかじ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。ただし、情報補完処理に使用する代替データは、本来サービス提供者側に依存しているものであることから、簡便な既存データの再利用などではなく、サービス提供者の状況や意図を十分に考慮したものを用意する必要がある。

● 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、情報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。したがって、直近のデータに対して結び付ける、もしくは、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。この場合、情報補完処理で結び付けられるデータは、元々サービス利用者のものであることから、既存の類似または直近データの流用が可能であり、これにより情報補完処理の作業負荷を軽減することができる。

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。

表 1 データ連携向け運用確認表
Table 1 Check table for data interoperability operations.

データの提供			データの利用	
			提供されたデータの運用	
			そのまま利用	サービス利用者の独自データと連携
提供される	プル型	少量	必要なデータを直接作成	必要なデータを直接作成
	プッシュ型	低頻度	必要なデータを直接作成	必要なデータを直接作成
	プル型	多量	代替となる簡易データの利用	類似または直近データの流用
	プッシュ型	高頻度	代替となる簡易データの利用	類似または直近データの流用

3.3 利用手順

サービス運用に支障をきたす想定外データの問題に対応するためには、サービス提供者、サービス利用者それぞれの運用の要求の関係を明らかにし、要求の組合せが妥当か否かを判断する。この手順は、次のとおりである。この手順は、サービスの利用者におけるソフトウェア要求を定義する際、サービス利用者が利用するサービスを含めサービス全体をどのように構築するかを検討する際に用いる。

(1) サービス連携における要求の定義

サービス提供者とサービス利用者の要求を定義するために、NFR フレームワークの手順に従い、図 3 で示した拡張 NFR 型カタログに記された非機能要求を、NFR ソフトゴールとして SIG に定義する。拡張 NFR 型カタログは、リユーザビリティや、インタオペラビリティ、フレキシビリティといった、サービス結合において重要な非機能要求を含んでいる¹²⁾。次に、定義した NFR ソフトゴールを達成するために、必要となるサブゴールに分解する。ゴール分解により、上位ゴールに貢献する他の非機能要求を NFR ソフトゴールとして、NFR ソフトゴールに貢献する機能的な要求を操作ソフトゴールとして、それぞれ SIG に追加定義する。これにより、サービス提供者とサービス利用者のそれぞれの要求の関係が SIG において明らかになる。これらのソフトゴールを定義した SIG の一例を図 6 に示す。細線の雲型は NFR ソフトゴールを、太線の雲型は操作ソフトゴールを、矢印は操作ソフトゴールによる貢献関係を表している。次に、サービス提供者およびサービス利用者のインタオペラビリティ達成に必要なゴールを分解し、データ連携向け運用要求カタログに記されたデータ維持・提供とデータ取り込み・利用に関する要求を、ソフトゴールとして SIG に追加定義する。その際、SIG にある他のソフトゴールとの相互依存関係を検討し、必要となる要求を抽出、誤った要求を訂正、不要な要求を削除する。その結果、必要となるデータ操作に関する要求が定義され、サービス利用者が利用するデータを列挙できる。

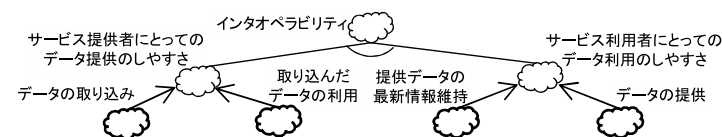


図 6 NFR ソフトゴールと操作ソフトゴールが定義された SIG の例
Fig. 6 Example of SIG defined NFR softgoals and operationalizing softgoals.

(2) データの運用に関するサービスの振舞いの把握

列挙したデータをサービス提供者が提供可能であれば、提供データごとに提供方法と頻度/量を明らかにする。これは、業務内容に依存するが、プッシュ型の提供の場合、サービス提供者が提供頻度を 1 時間に 1 回といった具体的な数値として提示し、サービス利用者がその頻度を高頻度が低頻度かに分類する。もしくはサービス利用者が、自らの運用から目安となるデータ提供の頻度を具体的な数値で提示し、サービス提供者の想定する運用がそれより高頻度が低頻度かに分類する。プル型の場合も同様に、サービス提供者が提供量を 1 回 100 件といった具体的な数値として提示し、サービス利用者がその量を多量が少量かに分類するか、もしくはサービス利用者が、自らの運用から目安となるデータ提供の量を具体的な数値で提示し、サービス提供者の想定する運用がそれより多量が少量かに分類する。サービス利用者は、提供されるデータをそのまま用いるのか、サービス利用者の独自データと連携させて利用するのか、その利用方法を定め、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の手段として、必要なデータを直接作成するのか、代替となる簡易なデータを利用するのか、既存の類似または直近データを流用するのかを定める。これらの一連の手続きを列挙したデータすべてに対し実施する。

(3) 機能要求の組合せの妥当性判断

データ連携向け運用要求カタログによって定義されたサービス提供者の運用に関する要求と、サービス利用者の運用に関する要求が、妥当な組合せかどうかを確かめる。これには、データ連携向け運用要求カタログで決定し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。これらの処理方法をデータ連携向け運用確認表にあてはめ、確認表と同じかどうかを確かめる。同じでない場合は、それらの組合せが妥当ではない可能性があるため、運用開始後にどういった問題が生ずるかを想定し、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。

ここまでの一連の流れを図 7 に示す。

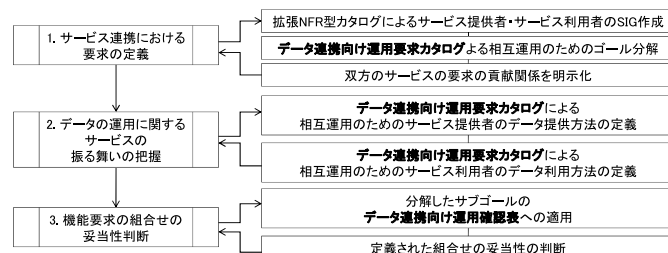


図7 データ連携向け運用要求カタログとデータ連携向け運用確認表の利用手順
Fig. 7 Procedures for the data operational type catalogue and the check table.

4. 事例研究

4.1 システム概要

本章では、実際の旅行 Web 予約システムの開発プロジェクトにおいて発生したデータの運用に関する問題を、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することにより再解決に導いた。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、あわせてその有効性を確認した。

プロジェクトで開発された旅行商品の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、旅行商品の造成卸売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造成卸売システムから取得している。SOA に基づいて構築されたこれらのシステム構成を図8に示す。

旅行販売の場合、たと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、サービスの連携を有効活用している典型的な例だといえる。

4.2 サービス連携における要求の定義

オンライン販売システムと造成卸売システムの相互運用を達成するために、図7で示した利用手順に従い、まず運用に関する要求を詳細検討し、サービス提供者、サービス利用者それぞれのサービス連携における要求を定義した。

拡張 NFR 型カタログを用いて、サービス連携における要求を SIG に定義し、データの運用に関するサービスの振舞いを把握するために、データ連携向け運用要求カタログによる

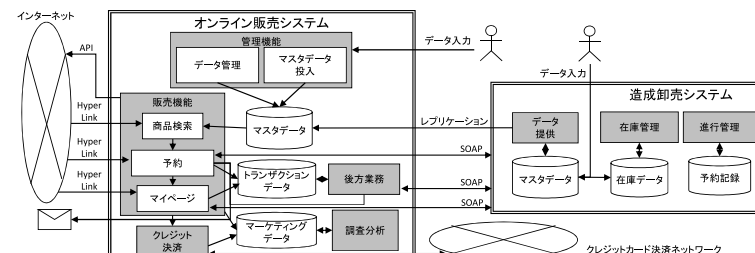


図8 旅行商品のオンライン販売システムと造成卸売システムの構成
Fig. 8 System structure of the online reservation system and the wholesale system.

ゴールの分解を行った。インタオペラビリティを達成するためには、まず、サービス提供者である造成卸売システムが、サービス利用者にとってのデータ利用のしやすさを達成する必要があり、これを NFR ソフトゴールとして、作成した SIG に追記した。この NFR ソフトゴールを達成するためには、マスターデータ/トランザクションデータの維持・提供が必要であり、これらを達成する機能的な要素であるデータの提供と、提供データの最新情報維持という2つの操作ソフトゴールを SIG に追加した。これらの操作ソフトゴールは、NFR ソフトゴールを達成するという貢献関係があるため、その貢献関係を矢印で示す。サービス間でやりと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。これらのデータは、データの提供と、提供データの最新情報維持という2つの操作ソフトゴールの対象となることから、サブゴールとして SIG に追加した。サービス提供者は、この4つのデータすべてを提供する必要があったため、これらのサブゴールの上位のゴールとの相互依存関係を、下位ゴールのすべてが成り立ったときに上位ゴールが成り立つことを意味する And 分解として記した。

同様に、サービス利用者であるオンライン販売システムは、サービス提供者にとってのデータ提供のしやすさを達成する必要があり、これを NFR ソフトゴールとして SIG に追加した。この NFR ソフトゴールを達成するためには、マスターデータ/トランザクションデータの取り込み・利用が必要であり、これらを達成する機能的な要素であるデータの取り込みと、取り込んだデータの利用という2つの操作ソフトゴールを SIG に追加した。これらの操作ソフトゴールは、NFR ソフトゴールを達成するという貢献関係があるため、その貢献関係を矢印で示した。データ取り込みの対象は、サービス間でやりとりする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、ツアー日程表のデータであることから、デー

タの取り込みという操作ソフトゴールのサブゴールとして SIG に追加した。サービス利用者は、すべてのデータを取り込む必要があるため、これらのサブゴールの上位のゴールとの相互依存関係を And 分解として記した。

取り込んだデータのうち、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、造成卸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。これを、取り込んだデータの利用という操作ソフトゴールに貢献するサブゴールとして、SIG に追加した。

以下の 3 つのデータについても、同様に SIG に追加した。

- 旅行代金データについて、いったん決定した代金を変更するケースはきわめて稀ではあるが、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。したがって、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ごと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。

作成した SIG において、インタオペラビリティを達成するためのゴールを図 9 に示す。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、サービスが持つデータに対し影響を及ぼす。

4.3 適用結果

図 7 で示した利用手順に従い、次にデータの運用に関するサービスの振舞いを把握した。データ連携向け運用要求カタログに基づき、図 9 に示したサービス提供者の SIG におけるマスタデータ/トランザクションデータである予約用在庫データ、旅行代金データ、空港マスタデータ、ツアー日程表データの操作ソフトゴールをさらに分解し、これら維持データを

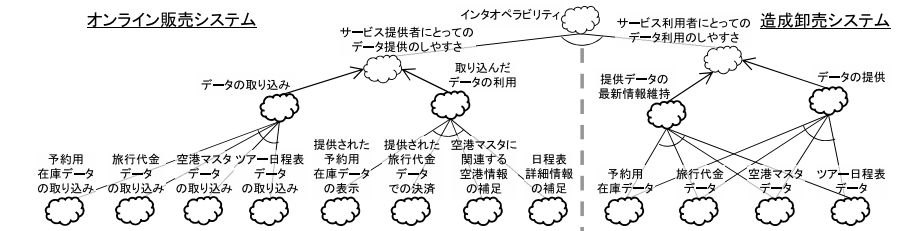


図 9 オンライン販売システムと造成卸売システムのインタオペラビリティ SIG
Fig. 9 SIG of Interoperability for the online reservation system and the wholesale system.

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
Table 2 Results of applying the data operational type catalogue and the check table.

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
予約用在庫	ブッシュ	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	ブッシュ	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	ブル	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	ブル	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

提供するための方法がブッシュ型なのかブル型なのか、そしてその提供の頻度/量を分類し、サービス提供者の要求仕様として決定した。この結果を表 2 のサービス提供者の運用として示す。同様に、サービス利用者の SIG におけるマスタデータ/トランザクションデータの操作ソフトゴールをカタログに基づいて分解し、取り込んだデータを利用するための方法が、提供されるデータをそのまま利用するのか、サービス利用者独自のデータと結び付けて利用するのかを、サービス利用者の要求仕様として決定した。この結果を表 2 のサービス利用者の運用として示す。

最後に、機能要求の組合せの妥当性を判定した。これは、表 2 に示したサービス提供者の運用とサービス利用者の運用を表 1 で示したデータ連携向け運用確認表にあてはめ、インタオペラビリティを達成するための情報補完処理が、必要なデータを直接作成するのか、代替となる簡易なデータを利用するのか、既存の類似または直近データを流用するのかを確認した。この結果を、表 2 の運用確認表の適用結果として示す。

実際のプロジェクトでは、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ため、これをプロジェクトでの定義として、表 2 にあわせて示す。運用確認表の適用結果と、プロジェクトでの定義を比較し、定義された組合せの妥当性を判断すると、予約用在庫データ、旅行代金データ、空港マスタデータに対する運用は、プロジェクトでの定義と運用確認表の適用結果が同じであり、妥当だと判断できるが、ツアー日程表データに対する運用は、プロジェクトの定義とは異なるものを採用すべきであり、妥当な組合せではないという判断結果が得られた。

プロジェクトでは、ツアー日程表の補足のための詳細情報データが、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うえ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。実際に本番運用を開始したところ、サービス利用者が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、問題の発覚から解決までに約 6 カ月間を要した。その間、提供されたデータは、結合すべき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、システムが十分に機能しない事態に陥った。

問題の解決にあたっては、3 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た。これにより、ツアー日程表のデータで、もし想定していた情報が存在しない場合、既存の類似または直近データを流用するようにソフトウェアを変更し、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。

予約用在庫データ、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 旅行業以外での適用事例

旅行業以外のサービスでも、業務に変化が多く連携するサービスの要求が把握しにくい分野において、サービス運用に支障をきたす想定外データの問題であれば、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、出版システムの 1 つである時刻表のシステムとのデータ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。出版のシステムは、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。実際に発生した問題は、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。

相互運用を達成するために、図 4 に示したデータ連携向け運用要求カタログに従い、サービス提供者において維持・提供するデータが「到着時刻」であることを、操作ソフトゴール“到着時刻（の入力）”として SIG に追加した。SIG には、拡張 NFR 型カタログに基づいて定義された NFR ソフトゴールや、NFR ソフトゴールを達成するための操作ソフトゴールもあるので、これらのソフトゴールと「到着時刻」に関する操作ソフトゴールとの間に相互依存関係があるかを検討した。この場合、サービス提供者は時刻表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った。操作ソフトゴール“到着時刻の削除”は、時刻表紙面を有効活用できるので、コストに貢献できる。この関係も SIG で記している。この事例の場合、他に追加で必要なゴールや不要なゴールはなかった。同様にサービス利用者において取り込み・利用するデータが「到着時刻」であることを、操作ソフトゴールとして SIG に追加した。このようにして作成された、インタオペラビリティを達成するための SIG を図 10 に示す。

次に、サービス提供者の SIG における到着時刻データの操作ソフトゴールをさらに分解し、これら維持データを提供するための方法がプッシュ型ではなくプル型であり、その提供量を多量として分類し、サービス提供者の要求仕様として決定した。この結果を表 3 のサービス提供者の運用として示す。同様に、サービス利用者の SIG における「到着時刻」デー

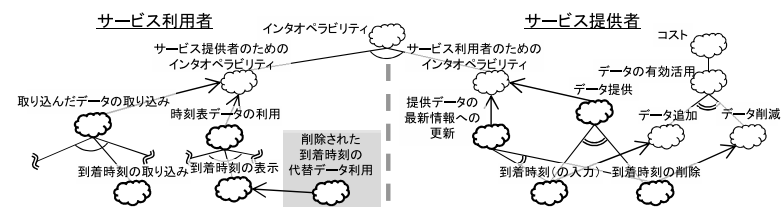


図 10 時刻表データの提供・利用におけるデータインタオペラビリティ SIG

Fig. 10 SIG of interoperability for timetable data.

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

Table 3 Results of applying the Type Catalogue and the Check Table for Timetable System.

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	ブル	多量	そのまま利用	代替となる簡易データの利用	必要なデータを直接作成

タの操作ソフトゴールをカタログに従って分解し、取り込んだデータを利用するための手段は、サービス利用者独自のデータと結合して利用するのではなく、提供されるデータをそのまま利用することを、サービス利用者の要求仕様として決定した。この結果を表 3 のサービス利用者の運用として示す。表 3 に示したサービス提供者の運用とサービス利用者の運用を、表 1 で示したデータ連携向け運用確認表へあてはめた。その結果、インタオペラビリティを達成するための情報補完処理は、代替となる簡易なデータを利用することになるため、これを表 3 の運用確認表の適用結果として示す。そして、実際のプロジェクトで定義していた、到着時刻データに関連するデータの運用法を、表 3 にあわせて示す。運用確認表の適用結果と、プロジェクトでの定義を比較すると、「到着時刻」データに関してプロジェクトで定義された運用は、妥当な組合せではないという判断結果が得られた。

時刻表のシステムとのデータの連携で、サービスが正常に動作しなかった原因は、サービス提供者が紙面の都合で「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、提供されるデータに値があったりなかったりするという事象が、サービス利用者側に伝わっていなかったことにあった。問題は、時刻表システム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。この現象は、単に値が存在しなかったただだが、このことはテスト時まで分からなかった。その理由は、「到着時刻」がインタフェース仕様にあったため、利用者側は必要なとき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。問題の解決には、表 3 に従い、値が存在しないなど想定外の場合、文字データを代替となる簡易データとして表示する対応を行い、無事サービス運用開始となった。このゴールは、図 10 の灰色の領域に示されている。

4.5 評価

このように、旅行と出版という 2 種類のサービス間において生じたデータの授受に関する問題を、3 章で述べた手法を適用することで解決に導いた。これは、ツアー日程表のデータにおいて、サービス利用者が対応しきれない大量のデータ更新という本番運用段階におけ

表 4 提案手法の適用に関するコストの比較

Table 4 Cost comparison for applying proposed method.

問題の 発覚時期	データの授受に関する 問題の影響	問題を発生させないために 生ずるコスト	その他の コスト	事例
本番運用 段階	問題発生から解決までに 約 6 カ月間を要し、その 間の販売機会を損失	問題の調査や応急的な対応のための作業 費用と、問題解決に向けた SIG 作成の ための作業費用 (10 人月超)	販売機会の 損失による 損害	ツアー 日程表
テスト 段階	問題発生時に、一時的な 作業負荷が増大	SIG 作成のための作業費用 (1 人月未満)	なし	時刻表 データ
要求定義 段階	なし	SIG 作成のための作業費用 (1 人月未満)	なし	—

る問題から回復し、また時刻表データの「到着時刻」に値の存在しない場合があり機能しないという問題を、テスト段階で解決したものである。どちらの問題も、その原因は要求にあったので、問題の解決は要求定義工程で行われることが望ましい。これらの問題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る。すなわち、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。したがって、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、提供されるデータに対する運用を、要求定義段階で正確に把握するために有効だといえる。

ツアー日程表において実際に発生した問題の解決にあたり、当初は本稿で述べたデータ連携向け運用確認表の考え方をういておらず、問題が発覚した本番データでの運用開始から解決までに約 6 カ月間を要した。本稿で述べた、拡張 NFR 型カタログを基に異なるサービス間での要求を調整し、運用に関する要求の組合せの妥当性を確認する手法は、要求の誤解によるデータ連携の運用の問題を解決するために必要となる調査や応急的対応、解決に向けた作業などの時間や労力を削減することができる。そのためには、この手法を要求定義の段階で活用する必要がある。このとき、データインタオペラビリティ SIG を作成する時間や労力は生ずるものの、問題解決に要した時間や労力に比べれば小さくて済む。さらに、問題が解決するまでシステムの稼働が遅れることによる販売機会損失を防ぐという効果も、あわせてもたらしてくれる。

以上の効果を基に、サービス間において生じたデータの授受に関する問題解決のために、提案した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を、どの段階で利用するかにより発生するコストの違いを比較した。これを表 4 に示す。コスト面で比較すると、問題の調査や応急的な対応、問題解決に向けた SIG 作成のための作業費用と、その間の販売機会の損失による損害を合算したものに比べ、要求定義段階において本稿で述べた手

法を利用することにより新たに発生する SIG の作成コストは、1 人月に満たない。

5. 関連研究

本稿では、サービス提供者とサービス利用者のデータ運用に関する要求を明らかにし、その組合せに矛盾があればそれを解消する方法を提案した。要求の矛盾については、これまで多くの研究が取り組まれている。

本稿で利用している NFR フレームワークは、ゴール指向に基づく手法の 1 つであるが、ゴール指向に基づく手法ではゴールどうしの連携や関係を明らかにすることができるため、ゴール間の矛盾の検出にも利用されている。代表的なゴール指向に基づく手法である KAOS 法を用いて、要求獲得の際に明らかになった要求間の衝突や矛盾を解決するための、形式的な技術について提案がされている¹⁴⁾。ここでは、単一の視点の中での矛盾だけでなく、ソフトウェア内の異なる利害関係者間での要求の矛盾についても形式化されている。この提案は、ゴール指向に基づく手法が、サービス提供者とサービス利用者の要求における矛盾を解消可能であることを示唆している。

一方、インターネットの普及にともない、ネットワークを介したより広範囲な連携が浸透することにより、ソフトウェアの構築は、SOA をはじめとするサービスを利用する形態へと移ってきている。サービスを利用した連携では、サービスがビジネスに合わせて開発・利用されている。サービスは、ビジネスのレベルでカプセル化されているため、その連携の検討にあたっては、ソフトウェアとしての検討にとどまらず、ビジネスレベルでの検討が必要である。つまり、サービスの連携では、システム視点での要求の矛盾を解決するだけでなく、ビジネス間の矛盾を解決する必要性にも迫られている。ゴール指向技術の観点でいえば、単一のゴール図の中での矛盾の解決だけでなく、複数のゴール図にまたがって矛盾を解決しなければならない。本稿の手法は、サービス提供者とサービス利用者の独立した SIG にまたがって、双方の要求を組み合わせることによる矛盾について解決に取り組んだものである。

IEEE Std 830-1998¹⁵⁾ は、要求仕様の標準的な記述方法として規格化されたものである。この規格では、優れた要求仕様における特性の 1 つに、矛盾が生じないよう要求仕様の中で一貫性が保たれていることがあげられている。あわせて、要求仕様の中で生じそうな矛盾として、そのソフトウェアが対象としている実世界で生じている矛盾、論理上もしくは一過的な矛盾、同じ次元の要求が異なる用語で書かれていることにより生ずる矛盾という、3 種類の具体例について述べられている。この規格は、要求仕様の記述にあたり、推奨される標準的な手法について記したものである。この規格を用いて要求仕様をとりまとめることによ

り、この具体例で示された 3 種類の矛盾について、回避できると考えられる。

文献 15) が規格化された時点では、SOA のようなサービスを利用するソフトウェア構築ははまだ提唱されていなかったが、サービスを利用するソフトウェアについても、前述の 3 種類の矛盾の具体例については、同様に生じる可能性があると考えられる。さらにサービスでは、単一のソフトウェア・システムが対象としているビジネスの領域を越え、連携するサービスが対象とするビジネスの領域との間に生ずる矛盾についても解決する必要がある。しかしながら、この規格の中で述べられている具体例は、あくまで単一のソフトウェア・システム内で生ずる矛盾についてである。本稿では、サービス提供者とサービス利用者のビジネスのレベルでカプセル化された運用の要求の組合せという、それぞれのビジネス領域にまたがった要求の妥当性を判断する手法を提案しており、組合せが妥当ではない場合、その矛盾を解決するものである。

サービスの利用が広がることにより、ソフトウェア・システムを取り巻く環境も複雑化しているが、そういった中で要求を取り扱う新たな手法についても提案されている¹⁶⁾。この手法では、ゴールやタスク、制約事項、ソフトウェア・システムが対象とするドメインにおける仮定などの関係に基づいて要求を明らかにし、要求の矛盾の取扱いも可能となっているモデル化のための言語である。このモデル化言語では、サービス利用における要求のモデル化や矛盾についても取り扱われているものの、あくまでもモデル化のための言語であり、利害関係者の視点やビジネス間の矛盾を解決する点にまでは言及されていない。

サービス提供者とサービス利用者の連携において、インタフェースは重要な役割を担っている。サービス間でやりとりするデータは、インタフェースを介しておりその影響を受けるので、インタフェースに注目した検討は、本稿の取り組みと関連が深い。

サービス提供者とサービス利用者の連携において、両者のインタフェース相違を吸収する研究について、報告がある¹⁷⁾。この手法は、インタフェースどうしの差を埋めるインタフェースのマッピングを動的に行う点に特徴がある。柔軟に様々なサービスと連携するような場合、動的なインタフェースのマッピングは有効であると考えられる。しかしながら、本稿が問題視している、連携するサービスを把握しにくい複雑な状況においては、動的にインタフェースをマッピングすることに対し依然として課題が残る。そもそも、サービス連携が複雑になる原因は、そのサービスが実現する環境やビジネスが複雑だからである。インタフェースのマッピングでは、インタフェースのメタ情報を活用してはいるものの、正確に連携するためには、インタフェースの本質を表している要求定義に立ちかえるべきである。本稿で提案した手法は、要求レベルで連携先を把握するものである。

サービス間のデータのやりとりとサービスの振舞いに関する研究も、これまでに取り組まれている。サービスの要求に基づいて、その振舞いを明確化するために、機能的な要素を考慮することは有用であるが、実際にサービスを運用するためには、非機能的な要素についてもサービスの選定の際に検討可能な手法が必要である¹⁸⁾。

データを中心とした Web サービスにおけるデータのやりとりに関して、再利用可能なモデルが提案されている¹⁹⁾。このモデルは、サービスやデータの相互作用を仕様化するために一般化されたものであり、データを中心としたサービスの振舞いを明確にすることが可能である。しかしながら、サービス運用の観点でいえば、振舞いのような機能的な要素だけではなく品質や制約条件も考慮すべきであり、ソフトウェアとしてこういった非機能的な要素の取扱いについても検討が必要である⁶⁾。Web サービスの仕様や利用方法を記述するためには、WSDL だけでなく OWL-S²⁰⁾ のように、Web サービス用のオントロジを利用する手法も存在する。しかしながら、WSDL も OWL-S も、サービスにおける機能面の記述に特化しており、非機能的な要素の取扱いは考慮されていない。

このように、結合するサービスにおけるデータ運用の問題は、連携におけるデータの操作に対する振舞いを要求に基づいて検討することにより解決可能だが、その検討にあたっては、非機能的な要求の達成も考慮された、統合された手法が必要となっている。本稿で提案した手法では、インタオペラビリティという非機能要求に注目し、サービスの振舞いを総合的に検討できるものである。インタオペラビリティ以外の、サービス間の連携に影響を及ぼす非機能要求についても、拡張 NFR 型カタログによって取扱いが可能である¹²⁾。

6. おわりに

本稿では、サービス間で生ずる運用上の問題のうち、データに起因する問題に注目し、インタオペラビリティをゴール分解することにより得られた、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、それらの利用手順を提案した。この提案手法を、旅行商品のオンライン販売システムと造成卸売システムの連携に適用し、得られた結果から提案手法が有効であることを検証した。さらに、出版システムにおける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。これらの問題解決に必要な情報は、テスト段階と要求定義段階で差がないため、要求定義工程で問題を解決することが可能であると考えられる。

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にくい場合に有効で、一例としてサービス提供者が、サービスの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あげられる。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。

本稿の成果は、業務に変化が多く、連携するサービスの要求が把握しにくい分野におけるシステム開発に対して有効であり、そのシステム開発の具体例として、パンフレットデータのような情報提供がともなう旅行業のシステム、コンテンツを取り扱う出版系のシステム、電子カルテといったものが考えられ、相互に連携するシステムにおいて、利用するデータに対する認識の誤解や、誤った前提条件の設定という問題を見つけることができる。さらに、サービスの運用を正常化させるという効果が期待できる。

なお、データ連携向け運用要求カタログを用いて作成された SIG を、サービス登録者に集約することにより、サービス再利用の促進が図れるものと期待する。

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), available from (<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) World Wide Web Consortium: Web Services Description Language (WSDL), available from (<http://www.w3.org/TR/wsdl>).
- 6) 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).
- 7) 矢嶋健一, 落水浩一郎: 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, 情報処理学会ソフトウェア工学研究会, Vol.2010-SE-170, No.12 (2010).
- 8) van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour, *Proc. 5th IEEE International Symposium on Requirements Engineering (RE'01)*, pp.249–263 (2001).

- 9) Glinz, M.: On Non-Functional Requirements, *Proc. 15th International Conference on Requirements Engineering (RE'07)*, pp.15–19 (2007).
- 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, *Proc. 17th Asia Pacific Software Engineering Conference (APSEC2010)* (2010).
- 13) Kilov, H.: From semantic to object-oriented data modeling, *Proc. 1st International Conference on Systems Integration on Systems Integration '90*, pp.385–393 (1990).
- 14) van Lamsweerde, A. and Darimont, R.: Managing Conflicts in Goal-Driven Requirements Engineering, *IEEE Trans. Softw. Eng.*, Vol.24, No.11, pp.908–926 (1998).
- 15) IEEE Computer Society: IEEE Recommended Practice for Software Requirements Specifications-IEEE Std. 830-1998 (1998).
- 16) Jureta, I., Borgida, A., Ernst, N. and Mylopoulos, J.: Techne: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences, and Inconsistency Handling, *Proc. 18th International Conference on Requirements Engineering (RE'10)*, pp.115–124 (2010).
- 17) 森谷高明, 大西浩行, 吉田 誠, 平野美貴: サービス連携のためのサービス間調停に関する一検討, 電子情報通信学会技術研究報告, Vol.106, No.577, pp.89–94 (2007).
- 18) 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)*, p.1 (2007).

- 19) Saleh, I., Kulczycki, G. and Blake, M.B.: A Reusable Model for Data-Centric Web Services, *Proc. 11th International Conference on Software Reuse: Formal Foundations of Reuse and Domain Engineering (ICSR2009)*, pp.288–297 (2009).
- 20) World Wide Web Consortium.: OWL-S: Semantic Markup for Web Services, available from (<http://www.w3.org/Submission/OWL-S/>).

(平成 22 年 11 月 19 日受付)

(平成 23 年 6 月 3 日採録)



矢嶋 健一

1970 年生。1995 年立教大学大学院理学研究科原子物理学専攻博士課程前期課程修了。株式会社 iJTB において、Web を活用した電子商取引システム開発の、要求定義や開発管理、プロジェクトマネジメントを中心に従事。2011 年北陸先端科学技術大学院大学情報科学研究科博士課程修了。博士 (情報科学)。



落水浩一郎 (正会員)

1946 年生。1969 年大阪大学基礎工学部卒業。1974 年同大学大学院基礎工学研究科博士課程修了。工学博士。静岡大学工学部講師、助教授、教授を経て、1992 年より北陸先端科学技術大学院大学情報科学研究科教授。ソフトウェア工学、特に、オブジェクト指向開発方法論とその支援環境、分散共同開発のプロセスモデルと支援環境、ソフトウェアアカウントビリティ機能の実現に関する研究に従事。著書に『ソフトウェア工学実践の基礎』、『オブジェクトモデリング』等。IEEE、日本ソフトウェア科学会、教育システム情報学会各会員。