

疎結合な関係にある企業を またがるトレーサビリティの実現

Introduction of Traceability over Loosely-Coupled Companies

百合山 まどか[◇] 小金山 美賀[‡]
渡邊 裕治[‡] 北山 文彦[‡] 沼尾 雅之[‡]

Madoka YURIYAMA Mika KOGANEYAMA
Yuji WATANABE Fumihiko KITAYAMA
Masayuki NUMAO

グローバルに展開される製造や輸送において、個々の製品および部品の品質を管理するためにトレーサビリティを実現することが求められている。ある製品の情報を追跡するためには、その製品に関する部品・原料等を扱う複数の企業から情報を収集する必要がある。本稿では、企業間の関係は動的に変更される疎結合なものであるため、企業システムからデータを収集する手段として Web サービスを採用し、トレーサビリティのための標準的な 2 つのインターフェースを定義した。また (1) 企業システムのデータを管理するデータソース管理者、(2) 製造や輸送等のビジネスプロセスを定義するプロセス定義者、(3) トレーサビリティシステムを利用して情報を検索するアプリケーションを設計する検索要求設計者、(4) アプリケーションを通して検索を実行する検索要求発行者というシステムのアクターの役割に応じて、各アクターがそれぞれ持ちうる知識も異なるため、本稿で提案するシステムでは、役割に応じて分割した定義を利用する。

Traceability is an essential component of quality management systems in the global manufacturing and distribution industry. Traceability requires harvesting information about parts and materials from multiple distributed data sources across several companies. Our traceability system provides a loosely-couple integration mechanism to multiple systems via Web Service, in order to cope with the dynamics of relationships between companies. We defined the two interfaces for traceability. There are four actors: (1) data source administrator to manage data of each company's system, (2) process definer to define business process, (3) query designer to design application using traceability system, and (4) query requester to get information by the application. We divided definitions for traceability system based on each actor's knowledge.

1. はじめに

[◇] 正会員 日本アイ・ビー・エム株式会社 東京基礎研究所
yuriyama@jp.ibm.com

[‡] 日本アイ・ビー・エム株式会社 東京基礎研究所
{mkogane, miew, ktym, numao}@jp.ibm.com

グローバル化が加速するビジネス環境の中で、世界にある経営資源を効率よく活用し、最適なところで調達・生産し、市場に届けることが重要となりつつある。それに伴い、グローバルに展開される製造や輸送において、個々の製品および部品の品質を管理するためにトレーサビリティを実現することが求められている。特に、不具合品の早期発見や特定に対応するために、製造業は最終製品、いわゆる完成品だけでなく製品を構成する部品についても履歴を辿ることが必要となってきている。トレーサビリティの必要性は製造業や輸送業のみならず、食品、医薬品、個人情報等、多種に渡って求められているが、本稿では主に製造業のトレーサビリティに焦点をあてている。

1.1 トレーサビリティとは

トレーサビリティとは、ISO9000:2000年度版[1]において、「考慮の対象となっているものの履歴、適用又は所在を追跡できること」と定義されている。本稿における「トレーサビリティ」は主に「製品のトレーサビリティ」を指し、「(最終製品だけでなく)製品の材料および部品の源、処理の履歴を追跡できること」を意味する。例えば、ある製品がどの部品を使用して、誰に生産され、どのような流通過程でこの業者に扱われたかなどといった情報がその好例であろう。

EPC-IS[2][3]に代表されるRFID(Radio Frequency Identification)によるトレーサビリティも注目されているが、本稿で扱うトレーサビリティ対象の製品や部品は、その全てにRFIDが付けられていることを仮定しない。そのため、例えば製品から、それを構成する部品の情報を得るためには、製品と部品の組み付け履歴データを取得する必要がある。組み付け履歴データとは、部品Aと部品Bを組みつけて、製品Cを作成した等という情報を含むデータを指す。このように本稿では、RFIDさえ読み取れば、関連する全ての情報が取得できるような状況を仮定せず、データを紐付けながら必要な情報を取得していくことでトレーサビリティを実現する。

1.2 トレーサビリティを実現する上での課題

トレーサビリティを実現するには、ビジネスプロセスの実行に伴い生成されるデータを紐付けることにより、データを統合し統一的なビューで提供することと考えられる。追跡対象のデータは、ライフサイクルに応じて様々なデータソース中に蓄積されるため、分散したデータソースの統合が不可欠である。そのためには以下に挙げる潜在的な課題がある。

- (1) 複数のデータソースにまたがったデータの間の関連を解決する必要がある。例えば、ある製品を構成する部品情報のトレーサビリティにおいては、組み付け履歴データベースにある組み付けデータと部品情報データベースにある部品の詳細データ(部品Aは2005/3/21に工場Xから納入された等)は、異なるデータソースにまたがっている可能性がある。
- (2) データソース毎に異なる表現を統合する必要がある。例えば、修理作業履歴は修理を担当する企業毎に異なる表現を用いている可能性がある。その一方で、修理履歴としてデータを統合するには、各企業の表現の違いを吸収して統合する必要がある。
- (3) データソース間の関連は時間に応じて変化する可能性がある。データソース間の関連は、データソースにデータを登録する主体(企業・アプリケーション・人)がどういふビジネスプロセスに沿って行動しているかに依存する。ところが、ビジネスプロセスは時間に応じて動的に変化するため、データソース間関係もそれに伴い柔軟に変更

できる必要がある。こういった場合、複数データソースから必要なデータを取得するために検索処理を設計すること自体が極めて煩雑な作業となる。

本稿では上記の課題を解決するために、分散したデータソースに保持されるデータに対して、統一的に検索手段を実現する基盤を提供するシステムを提案する。

2. トレーサビリティのためのデータソース

2.1 Web サービス経由のデータ収集

トレーサビリティシステムは追跡対象およびそれに関連するデータを分散したデータソースから収集する。単一企業・単一アプリケーション内でのトレーサビリティの場合は、トレーサビリティシステムとデータソースは密接な関係にあるので、データベースやファイルシステムに直接アクセスしてデータを収集することも可能だろう。しかし本稿がターゲットとするようなグローバルに展開される製造や輸送におけるデータは複数企業・複数アプリケーションで取り扱われているケースが多く、それらの企業間の関係は疎な関係である。取引状況や契約状況により、企業間の関係は動的に変化するため、データソース間を密に結合させるのは難しい。そこで本稿ではトレーサビリティシステムとデータソースの間のコネクションにWebサービスを採用した。

2.2 Web サービスのインターフェース

トレーサビリティシステムがデータソースに対する問い合わせで取得したいデータは、当然問い合わせ条件に合致するデータ一覧であるが、常に可能な限りの全ての情報を返すことはパフォーマンスの悪化につながるため、望ましくない。取得目的に応じて返答の構造を定義する必要がある。トレーサビリティシステムがデータソースに問い合わせをする目的は主に2つに分類できる。

- データソースで管理している詳細情報を取得すること。例えば、修理作業履歴を保持しているデータソースに対して、車のIDを入力して、その修理作業の詳細データを取得するなど。
- 次のデータソースへの問い合わせ条件に使用できるデータを取得すること。例えば、エンジン工場の出荷履歴を保持しているデータソースに対して、エンジンの型番と出荷年月を入力して、該当するエンジンのID一覧を取得するなど。トレーサビリティシステムは取得したエンジンのIDを使って、次のデータソースである組み付け工場へ該当エンジンを組み付けた車のデータを取得する。

前者の場合は、詳細データを返答として返すため、柔軟なデータ構造が望ましいと考えて、本稿ではXML文書を返答の形式として選択した。一方、後者の場合は、必要なデータセットは限定されるため、データ構造の柔軟性よりも処理の高速化を優先して、テーブル構造を返答の形式とした。本稿で提案するトレーサビリティシステムでは、以下の2つのインターフェースをデータソースは提供し、システムは取得目的に応じたインターフェースのWebサービス呼び出す。

- `getRecordObjects` インターフェース
トレーサビリティシステムから入力される検索条件(文字列で指定)に対して、該当するデータをXML文書で返す。返答のXML文書の構造(スキーマ)はあらかじめ定義しておき、システムに登録する必要がある。
- `findRecords` インターフェース
トレーサビリティシステムから入力される検索条件に対して、該当するデータを、テーブル構造で返す。テー

ブル定義はデータソース固有のため、システムに登録する。

3. トレーサビリティシステムのアクター

3.1 検索処理要求の概要

データベース検索において、事前にクエリの形式を入力しておくことでクエリの実行戦略の最適化を行うことは多くの場面で利用されている。とりわけ、トレーサビリティにおいては、分散したデータソースに保持されるレコードに対してクエリを発行する必要が生じるため、実行されるクエリのパターンに応じた最適化の余地が大きい。従って、本稿が提案するシステムにおいてはクエリの実設計情報と、パラメータ割り当てを分離して取り扱う方針を採用しており、検索要求はクエリ設計とクエリ発行の2段階で行われる。まず、こういった形式の検索要求を行うかを設計するアクター(検索要求設計者)がクエリの実設計情報をトレーサビリティシステムに入力する。クエリ設計に基づくクエリ要求を検索要求発行者が実際にトレーサビリティシステムに入力すると、システムは分散したデータソースから必要なWebサービス呼び出してデータを取得し、複数のデータソースから収集したデータを統合し、結果を応答する。

3.2 アクターの分類

(1) データソース管理者

トレーサビリティシステムが接続対象とするデータソースを管理するアクターである。各データソースの管理者は、データソースの構成情報やデータソースがどのような問い合わせを受け付けられるかなどの知識を有する。

(2) プロセス定義者

ビジネスプロセスがどのようなフローで処理されるか知るアクターである。またビジネスプロセスの実行に伴いどんな情報が生成されるのか、各データソースに保持されているデータをビジネスの形態に併せてどう統合するのが妥当か、というビジネスモデルの設計も行える。

(3) 検索要求設計者

トレーサビリティシステムに対する検索要求を設計するアクターである。例えば、トレーサビリティシステムを利用して情報を検索するアプリケーションの開発者などが該当する。クエリ設計は、プロセス定義者が定義したビジネスモデルに基づいて設計される。クエリパラメータとして何を入力し、検索結果としてシステムからどのような情報が出力されるべきかを知っている。

(4) 検索要求発行者

アプリケーションなどを通じて、トレーサビリティシステムに対する検索要求を発行するアクターである。検索要求の発行は、検索要求設計者の定義したクエリ設計に基づき行われる。検索要求発行者は、クエリ設計に対応したクエリパラメータを含むクエリ要求をシステムに入力することで必要な情報を取得する。

なお各アクターは必ずしも物理的に別の主体であることを意味しない。即ち、検索要求の設計者は、ビジネスモデルの知識もあり、プロセス定義者である場合もあるし、検索要求の発行者にもなりうる。またデータソース管理者がプロセス定義者として、システムにプロセスやデータモデルに関する定義を登録する場合もある。

3.3 アクターとデータ定義の関係

疎結合な関係にある企業間のトレーサビリティを実現するためには、異なる役割のアクターが独立に、かつ柔軟に必要な定義を設定し、運用できる必要がある。定義の詳細につ

いては、4章に記述するが、ここではアクターとデータ定義の関係の概略図である図1を参照しながら説明する。

- データソース管理者は自らが管理するデータソース(a1, a2, b, c1, ...)の定義をシステムに登録する。
- プロセス定義者は以下の定義を設定する。
 - *ビジネスプロセスにおける標準データモデル(A, B, ...)
 - *プロセスのフローの定義(AとB, AとC, BとE, ...を結ぶ線)
 - *データソースのデータと標準的なデータモデルとの変換定義(a1とA, a2とA, bとB, ...を結ぶ点線)
- 検索要求設計者はプロセス定義者が定義した標準データモデルやフローの定義から使用したいもの(A, B, E, G, H)を選択し、クエリパラメータ(Aの中の要素X), クエリの出力(Eの中の要素Y, Hの中の要素Z)を定義してクエリ設計を行う。
- 検索要求発行者はクエリ設計に対応したクエリパラメータ(Aの要素X = "xxx")を入力して、出力(Eの要素Y = "yyy", Hの要素Z = "zzz")を得る。

各アクターは自ら持ちうる知識の範囲内で必要な定義をすることができ、責任範囲外で変更が生じて、多大な影響を受けない。例えば、標準データモデルBに対応する新しいデータソースb'がデータソース管理者により登録されてもフローの定義やクエリ設定には影響が及ばない場合が多い。

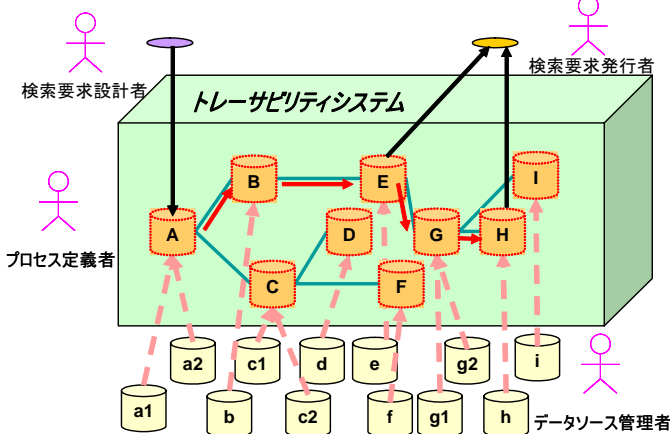


図1 アクターとデータ定義の関係

Fig.1 Relationship among Actors and Data Definitions

4. データモデル

本章では、トレーサビリティシステムがクエリ要求に応じてクエリ結果を返すために必要な定義を誰(アクター)がどのような定義を設定するかについて記述する。

4.1 データソースの定義

企業システムのデータソースに関する知識があるデータソース管理者は、トレーサビリティシステムに以下のような情報を含むデータソースの定義を登録する。

- データソースID(データソースを識別するための識別子)
- レコードスキーマ(データソース内のデータ構造)
- getRecordObjects インターフェースの Web サービス(入力として受け付ける検索条件/出力の XML 文書の構造 /Web サービスの接続情報)
- findRecords インターフェースの各 Web サービス(入力として受け付ける検索条件/出力のテーブル構造/Web

サービスの接続情報)

4.2 標準データモデルおよび変換の定義

データソースが出力するデータ(レコードオブジェクト)はレコードスキーマをスキーマ定義とする。このレコードスキーマはデータソースが独立に定義するものである。例えば、A社のエンジン組み立て履歴を示すレコードオブジェクトは、A社が独自に定義したレコードスキーマを持つと考えられる。一方で、一般的な「エンジンの組み立て履歴」のデータ形式があるとするれば、A社の定義するレコードスキーマから標準的なデータ形式への変換が定義される必要がある。こういったレコードスキーマでスキーマ定義されるレコードオブジェクトを別のデータ形式で表現する場合に、この変換先のデータ形式のことをレコードビューと呼ぶ。またレコードビューへの変換自体をレコード変換定義と呼ぶ。レコード変換定義はレコードオブジェクトをどのようにして変換してレコードビューのデータ形式に変換するかを定義する。レコード変換定義は単一のレコードオブジェクトだけでなく複数のレコードオブジェクトからの変換や多段のレコード変換定義を記述することもできる。これらの定義はビジネスモデルやビジネスプロセスの知識を持つプロセス定義者が行う。ビジネス定義者が独自の定義を行ってもよいし、業界の標準データモデルなどが存在する場合はそれをレコードビューの定義として用いることも考えられる。

4.3 データ関連の定義

トレーサビリティでは、異なるデータソースに存在する複数のレコードオブジェクトを特定の関連に基づいて統合する処理が頻出する。例えば、組み付け情報のトレーサビリティの場合、最終製品の組み付け情報と、部品の製造情報を統合する必要があるが、これら2つの情報をどうやって統合するかは、各情報がどういった構造しているか、あるいはデータソース間に部品納入→製品組み立てという関係があるか、といった情報に基づいて決定される。こういったデータソースをまたがる関連の定義を「レコードリレーション」と呼ぶ。レコードリレーションは、「2つのレコードオブジェクトが関連する」か否かを判定するためのブーリアン関数と考えることができる。レコードリレーションは、プロセスフローの知識があるプロセス定義者が、レコードビューの定義を参照して定義する。

4.4 クエリ設計とクエリ要求の定義

検索要求設計者が、トレーサビリティシステムに対する入力(クエリパラメータ)とシステムからの出力をクエリ設計として定義する。クエリ設計はクエリ設計IDを識別子として持つ。クエリ設計中のクエリパラメータやシステムからの出力はレコードビューを用いて定義する。一方、システムに対する検索要求をクエリ要求と呼び、アプリケーションを通して検索を実行する検索要求発行者が検索実行時にクエリパラメータを入力することで作成される。クエリ要求はクエリID及びクエリパラメータを含むデータである。

5. トレーサビリティシステムの構成

本章では、提案システムの構成について記述する。

5.1 コンポーネント

提案システムのコンポーネント図を図5に示す。本システムは大別すると以下の4つのコンポーネントから構成されている。

- データ検索インターフェース：トレーサビリティシステムを利用する検索アプリケーションからのクエリ要求を

受け付けて、実行結果を返す Web サービスを提供する。

- データコレクター：データソースが提供する Web サービスを呼び出して、データを取得する。
- 定義管理部：各アクターにより本システムに登録される各種定義を管理する。
- 検索エンジン：各種定義を基に、クエリ設計からのデータソースのどの Web サービスをどのような手順で呼び出せばよいかという検索ストラテジーを組み立てる。データ検索インターフェースを通じて、アプリケーションからクエリ要求を受け取ると、検索ストラテジーに従って、データコレクターを呼び出し、各データソースから必要なデータを取得し、結果を構成して出力する。

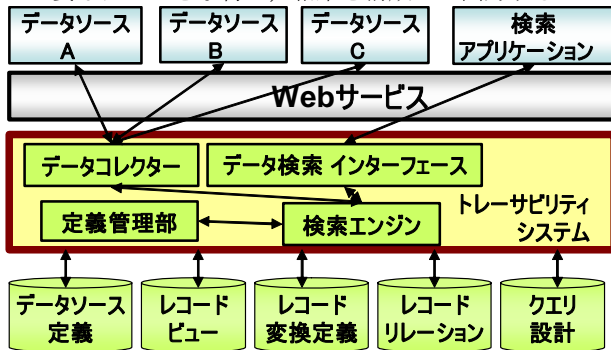


図2 システム構成図

Fig. 2 System Organization

なお実装には、データコレクターに IBM 社の WebSphere® Information Integrator V8.2[4]を適用した。またトレーサビリティシステム全体をエンタープライズアプリケーションとして構築し、WebSphere Application Server V5.2[5]上に展開することで運用した。

5.2 コンポーネントフロー

クエリ設計の登録時とクエリ要求の実行時の流れを示す。あらかじめデータソース定義、レコードビュー、レコード変換定義、レコードリレーションの登録は終了し、定義管理部からアクセスできる状態にしておく。検索要求設計者がクエリ設計をトレーサビリティシステムに登録すると、検索エンジンは関係のある定義類を定義管理部からロードし、どのデータソースのどの Web サービスをどのような手順で呼び出せばよいかという検索ストラテジーを組み立てて、保存する。つづいて、検索要求発行者が、登録済みのクエリ設計に基づくクエリ要求を検索インターフェースの Web サービスを通じて本システムに入力する。検索エンジンはクエリ ID から検索ストラテジーをロードし、データコレクターを通じて分散したデータソースから必要な Web サービスを呼び出してデータを取得した後、複数のデータソースから収集したデータを統合し、結果をアプリケーションに返す。

6. まとめ

本稿では疎結合な関係にある企業間のトレーサビリティを実現するため、以下のような特徴を持つシステムを提案した。

- トレーサビリティのための 2 種類のインターフェースを Web サービスで提供するデータソースが検索対象
- 検索処理、ビジネスモデル、データソースという 3 つの

異なる構成定義を独立したアクターにより実現

- ビジネスプロセスに関連して生成されるデータに対して標準的なビューとデータソース間の関連に基づく分散検索の検索要求設計
- 頻繁なビジネスプロセスの変更に伴う検索手順の煩雑な変更手続きを大幅に軽減

本システムにより、分散したデータソースに保持されるデータに対して、統一的に検索手段を用いてトレーサビリティが実現される。

【謝辞】

本研究について御討議いただいた松澤裕史氏、伊藤貴之先生、Christine Robson さんに感謝致します。

【文献】

- [1] ISO9000:2000 品質マネジメントシステム- 基本と用語.
- [2] EPCglobal, <http://www.epcglobalus.org/>.
- [3] EPC Tag Data Standards Version 1.1 Rev. 1.24, http://www.epcglobalinc.org/standards_technology/EPC_TagDataSpecification11rev124.pdf.
- [4] IBM WebSphere Information Integrator V8.2, <http://www-06.ibm.com/jp/software/websphere/ii/v82/>.
- [5] IBM WebSphere Application Server V5.2, http://www-06.ibm.com/jp/software/websphere/wv5/app_serv_v5.html.

百合山 まどか Madoka YURIYAMA

2001 京都大学大学院情報学研究科社会情報学専攻修士課程修了。同年日本アイ・ビー・エム株式会社入社。東京基礎研究所副主任研究員。プライバシー保護、トレーサビリティ技術に関する研究・開発に従事。情報処理学会正会員。日本データベース学会正会員。

小金山 美賀 Mika KOGANEYAMA

2004 奈良女子大学大学院人間文化研究科複合領域科学専攻博士後期課程修了。同年日本アイ・ビー・エム株式会社入社。東京基礎研究所副主任研究員。トレーサビリティ、ビジネスプロセスモデリングの研究開発に従事。情報処理学会正会員。博士(理学)。

渡邊 裕治 Yuji WATANABE

2001 東京大学大学院工学系研究科電子情報工学専攻博士課程修了。同年日本アイ・ビー・エム株式会社入社。東京基礎研究所副主任研究員。2004 年度情報処理学会論文賞受賞。トラステッドコンピューティング、ネットワークセキュリティ、プライバシー保護、トレーサビリティに関する研究開発に従事。情報処理学会正会員。博士(工学)。

北山 文彦 Fumihiko KITAYAMA

日本アイ・ビー・エム株式会社 東京基礎研究所勤務。1990 東京大学大学院理学系研究科修士課程終了。オブジェクト指向、ビジネスアプリケーション開発技術、トレーサビリティ技術などに関する研究・開発に従事。情報処理学会正会員。

沼尾 雅之 Masayuki NUMAO

1983 東京大学大学院工学系研究科電子工学専攻修士課程修了。同年日本アイ・ビー・エム株式会社入社。現在、同社東京基礎研究所にてビジネス変換技術担当。トレーサビリティ、ネットワークセキュリティ、プライバシー保護方式に関する研究開発に従事。91 年と 95 年に人工知能学会論文賞受賞。2004 年に情報処理学会論文賞受賞。博士(情報理工学)。

* IBM, WebSphere は IBM Corporation の商標である