



EPCIS及びCBV導入ガイドライン

EPCIS、CBV標準でビジネス・プロセスを可視化

日本語参考訳

リリース1.2、承認済み、2017年2月

一般財団法人流通システム開発センター 翻訳第一版:2017年3月21日

流通システム開発センターより

1. 本ドキュメント“EPCIS及びCBV導入ガイドライン EPCIS、CBV標準でビジネス・プロセスを可視化 日本語参考訳 リリース1.2”は“EPCIS and CBV Implementation Guideline Using EPCIS and CBV standards to gain visibility of business processes 1.2”(英文)の参考日本語訳であり、内容については、常に原文(英文)を優先するものとする。
2. 本参考日本語訳の無断転載を禁ず。

書誌事項

項目	内容
表題	EPCIS及びCBV導入ガイドライン
日付	2017年2月
版	1.2
刷	
ステータス	承認済み
文書内容	EPCIS標準とCBV標準でビジネス・プロセスを可視化

協賛者

氏名	所属
Andrew Kennedy、ワーキンググループ共同議長	FoodLogiQ
Ralph Troeger、ワーキンググループ共同議長	GS1 Germany
Gena Morgan、ワーキンググループファシリテーター	GS1 本部
Ken Traub, Editor	Ken Traub Consulting LLC
Philip Allgaier	bpcompass GmbH
Paul Arguin	r-pac international
Karla Biggs-Gregory	Oracle
Zsolt Bocsi	GS1 Hungary
Jonas Buskenfried	GS1 Sweden
Jaewook Byun	Auto-ID Labs, KAIST
Karolin Catela	GS1 Sweden
Mario Chavez	GS1 Guatemala
Luiz Costa	GS1 Brasil
Deniss Dobrovolskis	GS1 Sweden
Michael Dols	MET Laboratories
Hussam El-Leithy	GS1 US
Jürgen Engelhardt	Robert Bosch GmbH
Heinz Graf	GS1 Switzerland
Danny Haak	Nedap
Tany Hui	GS1 Hong Kong, China
Jianhua Jia	GS1 China

氏名	所属
Peter Jonsson	GS1 Sweden
Art Kaufmann	Frequentz LLC
Janice Kite	GS1 本部
Jens Kungl	METRO Group
Roar Lorvik	GS1 Norway
Paul Lothian	Tyson
Fargeas Ludovic	Courbon
Noriyuki Mama	GS1 Japan
Kevan McKenzie	McKesson
Reiko Moritani	GS1 Japan
Alice Mukaru	GS1 Sweden
Mauricio Munoz	Axway
Falk Nieder	EECC
Juan Ochoa	GS1 Columbia
Ted Osinski	MET Laboratories
Ben Östman	GS1 Finland
James Perng	GS1 Chinese Taipei
Craig Alan Repec	GS1 本部
Chris Roberts	GlaxoSmithKline
Thomas Rumbach	SAP AG
Chuck Sailer	Frequentz
Michael Sarachman	GS1 本部
Hans Peter Scheidt	GS1 Germany
Michael Smith	Merck & Co., Inc.
Michele Southall	GS1 US
Peter Spellman	TraceLink
Peter Sturtevant	GS1 US
Hristo Todorov	Axway
Geir Vevle	HRAFN AS
Elizabeth Waldorf	TraceLink
Ruoyun Yan	GS1 China
Tony Zhang	FSE, Inc.
Mike Zupec	Abbvie

改訂履歴

リリース	日付	担当者	改定内容

免責事項

GS1では、知的所有権の請求に関する不安を避けるために、GS1 IPポリシーに基づいて、このEPCIS及びCBV実装ガイドラインの策定ワーキンググループの参加者に、GS1メンバーにGS1 IPポリシーの用語定義による“必須クレーム”に対するロイヤルティ・フリー・ライセンス、あるいはRANDライセ

ンスを供与することを求めている。ただし、この仕様の内容のいずれかの部分を実装するにあたっては、必須クレームに関係していない特許その他の知的所有権に、抵触する可能性が皆無なわけではない。そのような特許その他の知的所有権は、GS1が求める実施権の付与の対象から外れる。さらに、GS1 IPポリシーによる実施権付与の合意には、ワーキンググループに参加していない、第三者の知的所有権や特許請求の範囲は含まれていない。

したがって、この仕様に準拠した実装に取り組むにあたっては、関係する特許がないかどうか、そして特許その他の知的所有権の実施権が必要でないかどうかを、各々の組織の側で判断することが望まれる。実施権の要否の判断は、組織が設計するシステムの具体的な詳細に照らしながら、自社の特許顧問などを交えて行う必要がある。

本書は“現状有姿”で提供するものであり、商品性、権利の不侵害、特定目的への適合性などの保証、あるいはこの仕様から派生するその他のいっさいの保証はない。GS1は、この規格の使用、あるいはその誤った使用から生じた損害について、間接的損害、結果的損害、懲罰的賠償、または特別損害賠償を含めて、いっさいの責任を負わないものとする。これは、本書の内容の使用や本書の利用に関係した、いっさいの知的所有権の侵害の賠償責任を含む。

GS1は、この文書にいつでも予告なく、変更を加える権利を保持している。GS1では、本書の使用に関して何らの保証もせず、本書に現れるいっさいの誤りについて責任を負わないほか、本書の記載内容を更新していくことについてもいっさいの確約はできない。

目次

1	はじめに	9
1.1	本書の対象	9
1.2	本書の範囲	9
2	EPCISの概要	9
2.1	EPCIS標準とCBV標準の内容	10
2.2	EPCIS可視化データの例	10
2.3	ビジネス・アプリケーションにおけるEPCIS	11
2.4	メリットとビジネスチャンス	13
2.5	EPCISデータとほかのタイプのデータの関係	13
2.6	典型的なIT環境の中のEPCIS	15
2.7	EPCISとGS1標準	16
3	EPCISイベントの構造	17
3.1	Whatディメンション	17
3.2	Whenディメンション	17
3.3	Whereディメンション	18
3.4	Whyディメンション	18
3.5	EPCISイベントのタイプとアクション	19
3.6	EPCISとコア・ビジネス・ボキャブラリ(CBV)	19
3.7	まとめ	20
4	EPCISを使った可視化システムの設計	21
4.1	ステップ1:可視化の目標と要件の取りまとめ	22
4.2	ステップ2:ビジネス・プロセスのフローの書出し	22
4.3	ステップ3:プロセス・フローのビジネス・ステップへの分解	22
4.4	ステップ4:可視化イベントが必要なビジネス・ステップの決定	24
4.5	ステップ5:各ステップ遂行を可視化イベントとしてモデル化	25
4.6	ステップ6:可視化イベントに入れるデータ・フィールドの決定	27
4.6.1	Whatディメンションの設計	27
4.6.2	Whenディメンションの設計	28
4.6.3	Whereディメンションの設計	29
4.6.4	Whyディメンションの設計	30
4.6.5	例	34
4.7	ステップ7:各データ・フィールドを埋めるボキャブラリの決定	34
4.7.1	Whatディメンションのボキャブラリ	35
4.7.2	Whereディメンションのボキャブラリ	35
4.7.3	Whyディメンションのボキャブラリ	36
4.7.4	例	37
4.8	ステップ8:可視化データ・マトリクスへの可視化イベントの整理	38
5	高度なEPCISのモデリング	39
5.1	アグリゲーション/ディスアグリゲーション	39
5.1.1	アグリゲーションとディスアグリゲーション	39
5.1.2	多階層のアグリゲーション	40
5.2	メーカー直送(Drop Shipment)	40
5.3	クラス・レベルのトレーシング	41

5.3.1	クラス・レベル識別を用いたトレーサビリティにおける固有の制約	42
5.3.2	クラス・レベル識別で始まりを示すイベント	42
5.3.3	アグリゲーションにおけるクラス・レベル識別	43
5.3.4	インスタンス・レベルとクラス・レベルの識別を同時に持つイベント	44
5.4	インスタンス/ロット・マスタ・データ(ILMD)	45
5.5	トランスフォーメーション	47
5.5.1	トランスフォーメーション・イベントの例	48
5.5.2	長期にわたるトランスフォーメーション	48
5.6	クーポンと引換券	49
5.6.1	簡単なクーポン・プロセス	50
5.6.2	クーポン仲介業者が介在するクーポンの例	51
5.7	繰り返し使用する資産のGRAIによる管理	51
5.8	ユーザ/ベンダー拡張要素	53
5.9	誤りのあるイベント	54
5.9.1	例 1: 通常イベントを使用した訂正 – シンプルな追加	55
5.9.2	例 2: 通常イベントを使用した訂正 – 訂正ビジネス・ステップ	56
5.9.3	例 3: 訂正イベントなしで、既存のイベントのエラーを宣言	57
5.9.4	例 4: 訂正イベントとともに、既存のイベントのエラーを宣言	58
5.9.5	エラー宣言イベントと訂正イベントを取得するタイミング	58
5.9.6	エラーと訂正が存在している状態でイベントにクエリを出す	59
6	EPCISデータの共有	59
6.1	1つの組織の中のEPCISデータの共有	59
6.2	EPCISクエリ	61
6.3	クエリ・モード: プル vs プッシュ	63
6.4	EPCISクエリ制御インターフェース	63
6.5	コレオグラフィ・モデル: サプライチェーン全体のデータ共有	64
6.5.1	中央集中型コレオグラフィ	65
6.5.2	分散型プッシュ・コレオグラフィ	66
6.5.3	分散型クエリ・コレオグラフィ	67
6.6	マスタ・データの同期	68
6.7	EPCISイベント・データのリダクション	69
7	データの検証とシステムの相互運用性	69
7.1	EPCISイベントの検証	69
7.2	認証プログラム	70
7.3	プログラムの認証要件	70
7.4	データ検証ポータル	70
7.5	ソフトウェアの認証全般	70
8	参考文献	70
A	付録:XMLの例	71
A.1	表3-1のEPCISイベントのXML	71
A.2	表4-6の例のXML	71
A.3	表5-3の例のXML	72
A.4	表5-4の例のXML	74
A.5	表5-6の例のXML	75
A.6	表5-7の例のXML	77

A.7	表5-8の例のXML	78
A.8	表5-10の例のXML	79
A.9	表5-14の例のXML	80
A.10	表5-15の例のXML	81
A.11	表5-16の例のXML	83
A.12	表5-21の例のXML	84
A.13	表5-22の例のXML	85
A.14	表5-23の例のXML	86
A.15	表5-24の例のXML	87

9 過去バージョンの協賛者89

図一覧

図2-1	ある簡単なビジネス・プロセスにおけるEPCISデータの発生	11
図2-2	トランザクション・データと可視化データの重なり	14
図4-1	ビジネス・プロセス・フローの例	22
図4-2	前向き物流プロセスのフロー(1/2)	23
図4-3	前向き物流プロセスのフロー(2/2)	24
図4-4	前向き物流プロセスのフローと可視化イベントの取得ステップ	25
図6-1	集中型コレオグラフィ	65
図6-2	分散型プッシュ・コレオグラフィ	66
図6-3	分散型クエリ・コレオグラフィ	67

表一覧

表2-1	ビジネス・アプリケーションとEPCISデータの利用の例	12
表2-2	GS1標準の“共有”レイヤーに含まれるデータのタイプ	14
表3-1	ビジネス・プロセス例のステップV3におけるEPCISイベントの情報内容	20
表4-1	ビジネス・プロセス例の各ステップへのイベント・タイプの割振り	26
表4-2	クラス・レベルとインスタンス・レベルのオブジェクト識別	28
表4-3	コア・ビジネス・ボキャブラリで定義されているソース/デスティネーション・タイプ	33
表4-4	プロセス例(4.4節)のステップV4に対するEPCISイベントの情報内容	34
表4-5	コア・ビジネス・ボキャブラリで定義されている標準ボキャブラリ識別子の例	36
表4-6	EPCISイベント(節4.6)の識別子の割振り例	37
表4-7	可視化データ・マトリクスの例	38
表5-1	よく見られるアグリゲーションの例	39
表5-2	アグリゲーション・イベントのアクション値	39
表5-3	EPCISアグリゲーション・イベントの情報内容の例	40
表5-4	階層になったEPCISアグリゲーション・イベントの情報内容の例	40
表5-5	コア・ビジネス・ボキャブラリで定義されているソース/デスティネーション・タイプ	41
表5-6	“メーカ直送”(Drop Shipment)シナリオ例のEPCISイベントの情報内容	41
表5-7	クラス・レベル識別を使ったEPCISイベントの情報内容の例	42
表5-8	クラス・レベルで識別される子オブジェクトのアグリゲーションに関するEPCISイベントの情報内容	43
表5-9	クラス・レベルの親を持った仮想的なEPCISアグリゲーション・イベントの情報内容(規則違反)	44
表5-10	インスタンス・レベルとクラス・レベルで識別した子を同時に持つEPCISアグリゲーション・イベントの情報内容	44
表5-11	インスタンス・レベルとクラス・レベルの識別子が同じオブジェクトを指している誤ったEPCISイベント	45
表5-12	インスタンス/ロット・マスター・データ(ILMO)の入ったEPCISイベントの情報内容	46
表5-13	トランスフォーメーションのビジネス・プロセスの例	47
表5-14	トランスフォーメーション・イベントの情報内容の例	48
表5-15	トランスフォーメーション識別子でリンクしたトランスフォーメーション・イベントの情報内容の例	49
表5-16	簡単なデジタル・クーポン・ビジネス・プロセスのEPCISイベントの情報内容の例	50
表5-17	繰り返し使用する資産の管理側ビジネス・プロセス例のプロセス・フローチャート	51
表5-18	繰り返し使用する資産の管理側ビジネス・プロセスに関するEPCISイベントの情報内容(V1 – V4)	52
表5-19	繰り返し使用する資産の管理側ビジネス・プロセスに関するEPCISイベントの情報内容(V5 – V9)	52

表5-20 ユーザ/ベンダー拡張を使ったEPCISイベントの情報内容の例	53
表5-21 訂正ビジネス・ステップとともに通常イベントを追加することによるエラー訂正の例	56
表5-22 通常イベントの追加によるエラー訂正の例	56
表5-23 エラー宣言イベントの追加によるエラー訂正の例	57
表5-24 エラー宣言イベントの追加によるエラー訂正の例	58
表6-1 EPCISのクエリ条件の例.....	61
表6-2 ビジネス情報の要求と、そのためのEPCISクエリ条件の例	62
表6-3 ビジネス・シナリオと、それに対応するEPCISクエリ・モード.....	63
表6-4 EPCISクエリ制御インターフェースの処理	64
表6-5 中央集中型コレオグラフィの特徴	65
表6-6 分散型プッシュ・コレオグラフィの特徴.....	66
表6-7 分散型クエリ・コレオグラフィの特徴.....	67

1 はじめに

高品質で、安全な、多種多様な商品とサービスを、妥当な価格でありながら、環境に配慮した社会的に責任あるやり方で生み出していくうえで、消費者にとっても企業にとっても鍵になるのは、グローバルなサプライチェーンである。

サプライチェーンには、今日の消費者要件に応えるために、これまで普通にみられたよりも、はるかに精密な水準の可視化が求められる。現状、そのようなきめ細かい可視化は、物流業界には見られるが、そのほかの分野ではまだ進んでいない。商品を注文して配送状況を調べてみると、どこから配達が始まり、途中どこの中間地点を經由して、到着の予定がいつで、もう実際に届いたかどうかまで手に取るように分かる。企業や組織、官公庁、そしてもちろん消費者も、次第にそのような可視性を、購入するあらゆる製品や、食卓に並ぶ食品、あるいは関心を注ぐ事柄に関する電子記録にさえ求めるようになりつつある。

可視化データが記述するのは、仮想/物理オブジェクトの産地や出処、あるいはサプライチェーンその他のプロセスの全体を流れる中で、そのオブジェクトがビジネス・プロセスの対象として処理された場所、それらの処理が行われた日時、そしてそのときオブジェクトに何が起こったかなどである。つまり可視化データは、あるオブジェクトについてのWHAT、WHERE、WHEN、WHYを指している。1つの会社内でも、取引相手間でも、可視化データを取得して共有することで、製造、出荷、入荷、販売のプロセスの履歴に踏み込むことができる。それが効率的で、コストが低く、安全性の高いサプライチェーンの実現につながる。

EPCIS(Electronic Product Code Information Services)は、GS1標準の1つであり、企業内またはオープンなサプライチェーン全体における可視化データの共通データ・モデルと、その取得と共有のためのインタフェースを定義している。EPCISの目的は、それぞれ個別の機能を持ったアプリケーションが、1つの組織内部及び組織間にわたって、可視化イベント・データを生成し、共有できるようにすることである。最終的には、この共有によってユーザがあるビジネス・コンテキストに置かれた物理オブジェクトまたはデジタル・オブジェクトに対して共通の認識を持てるようにすることを目標とする。

1.1 本書の対象

このガイドは、サプライチェーンに関係する製造業者、卸売業者、小売業者、流通業者、ソリューション・プロバイダ、ビジネス・プロセス設計者、IT部門(開発業者)、ITソリューション・プロバイダなどを対象に想定している。それらの関係者に向けて、EPCISとともに、特にコア・ビジネス・ポキャブラリとGS1標準を使いながら、可視化システムを導入する際の初歩的な事項を記載する。

1.2 本書の範囲

本書は、EPCISによる可視化システムの検討に着手するにあたっての、概要と取組みの指針を提供するためにまとめたものである。したがって本書は、各業界の事情を踏まえたテクニカルで詳細な“ハウツー”ガイドではない。医薬品、電子機器、流通、食品及び農業などの分野には、業界別にEPCISの実装ガイドが用意されている。本書は、EPCISの基本的な利用の指針を提供し、各分野に固有の条件を取り入れた業界別のガイドラインを利用するための、土台を築くことを意図している。

2 EPCISの概要

EPCISの目的は、それぞれ個別の機能を持ったアプリケーションが、1つの組織内部及び組織間にわたって、可視化イベント・データを生成し、共有できるようにすることである。最終的には、この共有によってユーザがあるビジネス・コンテキストに置かれた物理オブジェクトまたはデジタル・オブジェクトに対して共通の認識を持てるようにすることを目標とする。

EPCISの中に出てくる“オブジェクト”とは、多くの場合、クラス・レベルまたはインスタンス・レベルで識別され、複数の企業や組織にもわたるビジネス・プロセスの中の各ステップで、物理的な取扱いを受ける、物理オブジェクトを指す。例えば、商品(製品)、物流単位、繰り返し使用する容器、固定資産、紙媒体の文書などがある。“オブジェクト”には、そのほかにデジタル・オブジェクトがあり、同じくクラス・レベルまたはインスタンス・レベルで識別され、ビジネス・プロセスのうち、それに該当する各ステップの中に現れる。例えば、デジタル商品(ダウンロード販売の楽曲、電子ブックその他)、デジタル・ドキュメント(電子クーポンその他)などが、デジタル・オブジェクトにあたる。本書の全体を通して“オブジェクト”という言葉は、クラス・レベルまたはインスタンス・レベルで識別され、ビジネス・プロセスの各ステップで取扱いの対象になる、物理オブジェクトと、デジタル・オブジェクトを指す。EPCISにおけるデータは、“可視化イベント”で構成される。これは、ビジネス・プロセスの中で、オブジェクトを対象にした特定のステップの遂行を記録するものである。

EPCIS標準は、もともと物理/デジタル・オブジェクトの詳しい情報を共有することで取引会社間の協業を拡大する、大きな取組みの中で考え出されたものである。EPCISの名称は、この取組みが、EPC(Electronic Product Code)の開発を起源にすることを反映している。しかしEPCISは、かならずしもEPCや、RFID(Radio-Frequency Identification)データキャリアを使うことを要求せず、EPCIS 1.1にいたっては、本来EPCが目的としていた、インスタンス・レベルの識別でなくても構わない。EPCIS標準は、可視化イベント・データの取得と共有が行われるどんな状況にも適用でき、名称の“EPC”は、歴史的な名残

りにすぎない。

2.1 EPCIS標準とCBV標準の内容

EPCIS標準仕様は、次のことを規定している：

- XML(eXtensible Markup Language)を使った可視化データの厳密な構文を伴う可視化イベント・データの**データ・モデル**
- 社内や会社間の環境において、きちんと定義されたサービスを滑らかに組み込むために用意された、標準化されたオープンな**インタフェース**。EPCIS標準仕様では、2つのインタフェースを定義している：
 - **キャプチャ・インタフェース**。EPCISデータ・モデルに準拠する可視化イベント・データを、キャプチャ・アプリケーションから、受取り側(通常はEPCISデータの永続的なリポジトリ)に送るためのもの
 - **クエリ・インタフェース**。EPCISイベント・データを、ビジネス・アプリケーションまたは取引相手から要求して、受け取るためのもの

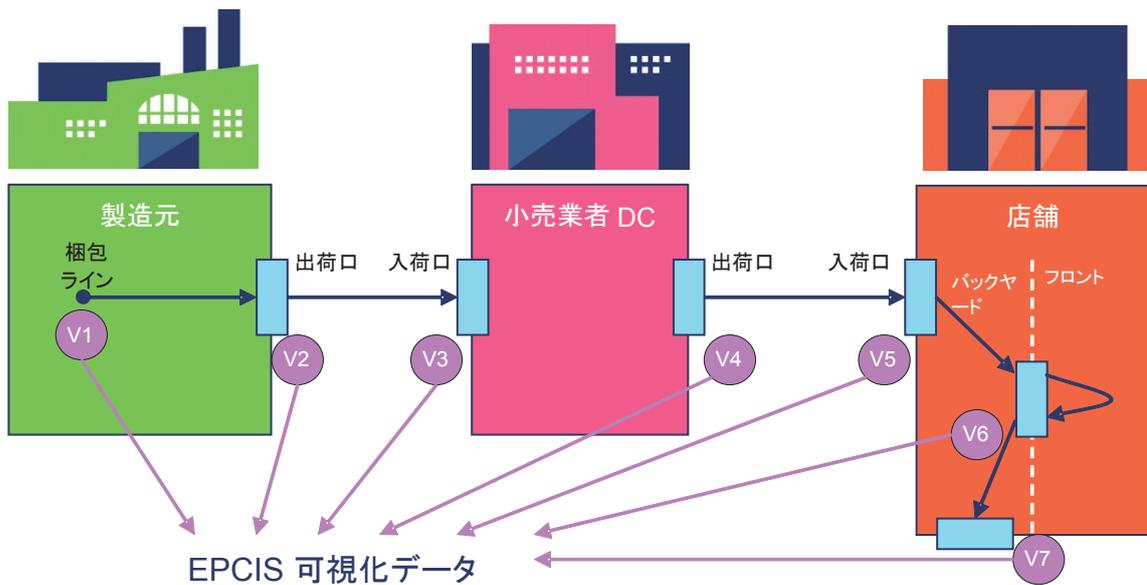
EPCIS標準の中で標準のインタフェースを規定することで、明確なサービス処理とデータ標準を利用した可視化イベント・データの取得と問い合わせを規定している。その際、ユーザ企業のニーズを満たすための適切なセキュリティの仕組みを組み入れることも可能である。キャプチャ・インタフェースとクエリ・インタフェースを直結してデータベースを使わずにアプリケーション間を直接つなぐデータ共有もできるが、ほとんどの場合は、可視化イベント・データを格納したデータベースを使うことになる。

EPCISは、GS1コア・ビジネス・ボキャブラリ(CBV)標準仕様[CBV1.2]と、組み合わせて用いることを想定している。CBV標準は、EPCIS標準仕様で定義されているデータ構造の中で利用するデータ値を定義している。CBV標準が用意している標準のボキャブラリを利用することは、相互運用性を高めるために重要であり、データのクエリの際に、さまざまな会社が、共通の意図をばらつきなく表現するうえでも、きわめて重要になる。キャプチャ・アプリケーションは、EPCISデータを構成する際にはCBV標準を可能な限り利用することが望ましい。

2.2 EPCIS可視化データの例

EPCISデータは、ビジネス・プロセスの中に置かれたモノが、現在または過去にどこにあって、どんな取扱いを受けたかを表すことができる。下の図は、ある簡単なビジネス・プロセスの中で、EPCISデータが生成される場所を示している。

図2-1 ある簡単なビジネス・プロセスにおけるEPCISデータの発生



図の簡単なビジネス・プロセスでは、まず商品が製造されて、物流センター(DC)に向けて出荷される。その荷が物流センターで受け取られると、その後さらに店舗に配送されて、店舗に届いた後、やがて売り場に並べられる。ビジネス・プロセス全体は、製品の包装作業、出荷コンテナへの積み込み作業、出荷作業、入荷作業等々と、1つ1つのビジネス・ステップの流れとして見る事ができる。EPCISデータは、これらのステップの各々の詳細を記録していく。1つのビジネス・ステップの遂行を記述するデータを、EPCISデータの1単位とし、それをEPCISイベントと呼ぶ。EPCISイベントが集まって、時間と空間に広がるビジネス・プロセスの詳しい全体像が描き出される。

例として、物流センターで、出荷品の入荷を記録するEPCISイベントを、取り上げて考えてみる。このイベントが収める内容は、4つのディメンションに整理できる：

- **What** どの商品/荷物を入荷したかの情報
- **When** 受け取った日時と、現地のタイムゾーン
- **Where** 受け取った場所と、このイベント以降に商品が所在すると想定される場所
- **Why** 次のような、ビジネス・コンテキストについての情報：
 - ビジネス・ステップが、出荷その他のビジネス・ステップではなく、入荷処理であること
 - 出荷は、サプライチェーン上を上流方向ではなく通常の下流方向に進むものであること
 - 出荷側と入荷側の情報、及び元の所有者と新しい所有者の情報(出荷元と荷受人とは異なる場合)
 - 発注書、請求書、発送通知書(事前出荷明細書)など、関連するビジネス・トランザクション・ドキュメントへのリンク

図で示したプロセスの中のすべてのビジネス・ステップから、EPCISイベントを生成することができる。それぞれのイベントの詳しい内容は、ビジネス・ステップによって異なるが、すべて4つのディメンションの構造を持つ。

2.3 ビジネス・アプリケーションにおけるEPCIS

EPCISは、ある一定の期間ビジネス・プロセスやサプライチェーンの全体から記録したイベントを収集することで威力を発揮する。例えば、次のようなパラダイムが実現する：

- あるオブジェクトの最新のEPCISイベントを見つけて、現在の場所と状態を知る(“トラック”)。
- あるオブジェクトの過去のイベントを集めて、ビジネス・プロセスやサプライチェーンの中を流れた経路をたどる(“トレース”)。
- 特定の場所や特定のビジネス・プロセスで、ある時間内に起こったイベントを収集して分析する(“分析”)。
- 最新のEPCISイベントから知られるオブジェクトの実際の状態を、前のビジネス・トランザクションやEPCISイベントから期待される状態と比較する(“チェック”)。
- 新しく取得されたEPCISイベントによりあるビジネス・ステップの完了が分かった段階で、ほかのビジネス・プロセスをリアルタイムで開始する(“自動化”)。

このような方法を使って、EPCISデータのメリットを引き出すビジネス・アプリケーションの例を、下の表に挙げている。もちろん、このパラダイムは広く一般化したものであり、実際のビジネス・アプリケーションは、それらを組み合わせたり、パラダイムの外にも踏み出したりしながら、EPCISデータを応用することになる。

表2-1 ビジネス・アプリケーションとEPCISデータの利用の例

ビジネス・アプリケーション	EPCIS データの利用方法	基本パラダイム
偽商品対策、産地の判定	製品の出処や流通経路の検証	トレース、チェック
流通過程(預かり、所有者)の把握	流通過程の製品の属性や、製品を物理的に所有していた全当事者の再現と文書化	トレース
クーポン処理	顧客行動分析、リアルタイムのクーポン検証	分析、チェック
通関処理	電子封印による通関の能率化、不正防止	トレース
リコール	対象製品の精密なトレースによる、リコールの迅速化	トラック(リコール製品の所在確認)、トレース(リコール進捗の監視)
販売促進	消費者に向けた販促品の適時適所の送達	トラック
トレーサビリティ	拡張されたサプライチェーンの各ステージにおいて、特定の段階を前向き/後ろ向きに流れる製品の動きの追跡	トレース
ビジネス・プロセス最適化	リード時間の短縮、設備能力の利用率向上、納入の品質と精度の改善	自動化、分析
例外管理	所定の製品、期限、数量、品質、場所、状態などからの逸脱の監視と、プロセス責任者への通知	チェック、自動化
食品生鮮度管理	賞味期限などの監視	トラック、自動化
資産管理	固定資産の管理と、ビジネス・プロセスへの必要数の確保	トラック、分析
在庫管理	入出庫、棚卸しのデータ取得	トラック、分析
プロセスの文書化	デジタルドキュメントの自動生成とワークフローの自動化、GS1 識別コードで識別されるドキュメント、製品、場所へのリンク	自動化

これらのアプリケーションの導入には、次の3つのかたちが考えられる:

- 内部完結型:ビジネス・プロセスが、1つの会社や組織の管理する施設の範囲に収まる。
- 外部とのクローズなサプライチェーン:ビジネス・プロセスが複数の会社や組織に広がるが、関連する会社や組織が、前もって分かっている。
- 外部とのオープンなサプライチェーン:ビジネス・プロセスが複数の会社や組織に広がり、関連する会社や組織の全体が前もって分からず、時間とともに変化する。このかたちは、相互取引のある大きなサプライチェーンの典型である。

3つのどの場合も、ソリューション設計で鍵になるのは、ビジネス・アプリケーションの要求が満たされるようにEPCISイベントのデータ内容を決めることである。そのほかに外部型では、EPCISイベントを、会社や組織の間でどのようにやり取りするか設計にも考慮が求められる(これは前者のデータの“コンテンツ”に対して、“コレオグラフィ”とも呼ばれる)。

外部オープン型チェーンの場合、EPCISとCBVがオープンな標準であることが、明らかに大きな価値を持つ。すべての当事者が標準に従っていれば、ソリューションの設計段階に全員が参加していなくても、データの相互の運用と理解に問題は起こらない。また、これはクローズ型チェーンや、内部に限られたアプリケーションでもまったく同じように大切である。内部アプリケーションは、時間とともに外部アプリケーションに変わっていき、クローズ型のアプリケーションはオープン型のアプリケーションに変わっていく傾向がある。そこでクローズ型や、純粋に内部向けのアプリケーションの場合でも、外部オープン型アプリケーションのベストプラクティスを取り入れることが大切になる。

2.4 メリットとビジネスチャンス

可視性を高めることが、産業分野を問わず、サプライチェーンのあらゆる段階でさまざまなメリットをもたらす。産地や製造元に始まって、経由する各々の物流拠点、そして最終的に消費者に販売する店舗にわたる、プロセスの記録から発生するメリットとして、次のようなものが考えられる：

- 入荷業務の能率化
- 在庫管理の向上
- ピッキング作業の迅速化
- ミスピックや数量不足などの作業ミスの削減
- 発注精度の向上、請求ミスの削減
- トラックとトレースの際の、製品と場所の特定能力の向上
- 各種のビジネス・プロセスをまたいだ業務の能率化
- より迅速で精密なリコール対応体制の整備
- 消費者保護の拡大
- 製品、アレルギー物質、栄養価についての消費者向け情報の充実

EPCISとコア・ビジネス・ポキャブラリは、可視化データの取得と共有のための技術基盤を用意し、ある物が“今どこにあるか？”、あるいは“過去にどこを通過してきたか？”の問いに答えるうえで役立つ。可視化データを標準化された方法で共有することが、各社独自の方法を取る場合にはない大きな効果をもたらす。EPCISを使うことにより、1つの企業内部でも、取引会社間にわたっても、さまざまなビジネス・アプリケーションの間でデータ共有ができる。EPCISは、ストリーミング処理(イベント・データの取込みと送出し)においても、複雑なイベント処理(イベント・データの検索絞込みなど)においても、リアルタイムに処理を行い、イベント・ベースのデータを返す。近年、自分の購入しようとする商品がどのような経路を流れて手元に届いたかを含めて、ますます詳しい情報に関心を注ぐ消費者の飽くことのない要求に、EPCISを利用したシステムで応えることができる。

EPCISは、データ取得とデータ問い合わせの2つのインタフェース標準を定めている。そして、コア・ビジネス・ポキャブラリが、EPCISのデータ・モデルにビジネス・コンテキストを与える。トレーサビリティや、その他のビジネス・プロセスに目標を置いて、1つの組織内部やいくつもの組織にわたって得られる可視化データを利用しようとするソフトウェア・アプリケーションには、EPCISを基盤として実装しているものが多い。企業や組織において、可視化システムのプランを検討するにあたっては、この標準を使ったソリューションに目を向けるべきである。

2.5 EPCISデータとほかのタイプのデータの関係

GS1標準の“共有”レイヤーには、エンドユーザ間で共有される3つの種類のデータがある：

表2-2 GS1標準の“共有”レイヤーに含まれるデータのタイプ

データ	説明	GS1 標準
マスタ・データ	GS1 識別コードによって識別される現実世界のモノ(例えば商品、当事者、場所など)を説明する属性を提供し、多くの取引相手との間で共有させるデータ	GDSN
トランザクション・データ	明示的な契約(供給契約など)や暗黙の了解(通関処理の場合など)によって定まる、あるビジネス・プロセスの開始から完了まで(例えば、製品の発注から最終の決済まで)にわたる中で、ある機能のトリガーとなる、またはその実行確認を行う売買取引。同じく GS1 識別コードを用いる。	GS1 eCOM XML
可視化データ	サプライチェーンの中で、コードによって識別される製品やその他の資産が関わる、物理/デジタル形態の活動についての詳細であり、それらがいつ、どこで、どのように行われるかを、1つの組織内だけでなく、ほかの組織にもわたって詳しく表すもの	EPCIS

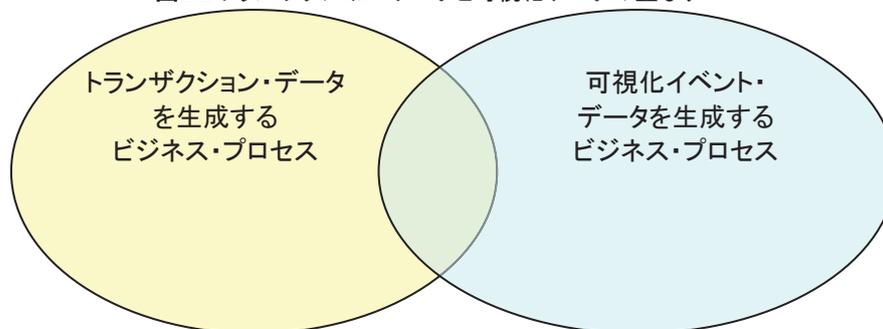
表から分かるように、可視化データ(EPCISイベント・データ)は、マスタ・データともトランザクション・データとも違った特徴をもつ、**新しいタイプのデータ**である。

EPCISデータの際立った特徴は、マスタ・データやトランザクション・データに比べて、はるかに大きな量が発生することである。トランザクション・データと同じように(そしてマスタ・データとは異なり)、組織のビジネス活動に伴って、新しいデータが増えていく。しかし次のような理由から、可視化データの方が発生量は大きい:

- 可視化データは、例えば、GTIN(グローバル・トレードアイテム・ナンバー)+シリアル番号で識別される商品など、オブジェクトの個々のインスタンスを指すことが多い。
- クラス・レベルのオブジェクトを指す場合でも、可視化データは、ビジネス・プロセス全体の中のいくつかのステップで生成される。例えば、商品が製造元から小売店に流れていく場合、関係するビジネス・トランザクションは1つだけ(製造業者から小売業者への販売)でも、発生する可視化イベントは、商品が各々の施設を流れていく中でいくつにもなる。
- 可視化データは、トレーサビリティの面で過去の履歴に価値のあることが多いため、ビジネス・トランザクション・データよりも保持する期間が長くなる傾向がある。

可視化データとトランザクション・データは、相補的な関係にある。ビジネス・トランザクションなしに発生する可視化イベントがあれば、オブジェクトの取扱いなしに発生するビジネス・トランザクションもある。1つのビジネス・プロセスが、可視化データとトランザクション・データを同時に発生するときは、相補的データにあたる。

図2-2 トランザクション・データと可視化データの重なり



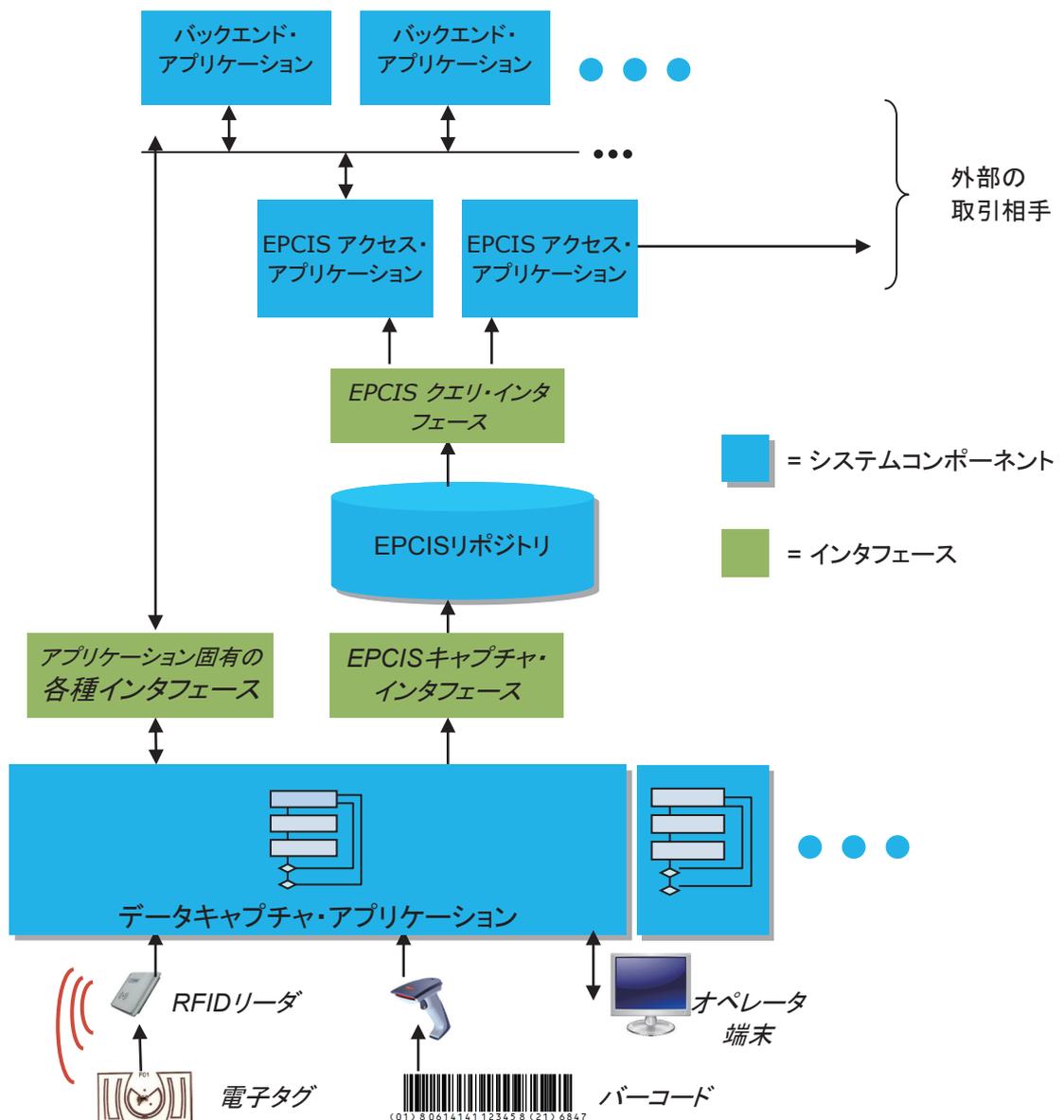
3つの場合は、それぞれ次のようになる:

- 可視化イベントとビジネス・トランザクションが同時に発生する場合は、トランザクション・データと可視化イベント・データとが、同じ出来事を違った側面から記述する。例えば、商品が積込みドックから出荷される場合、発送通知書(送り手が受け手に向けて、特定の商品を送る旨を確認するトランザクション・データ)があり、さらにビジネス・ステップ“shipping”のEPCISイベント(商品の積込みドック出発を検知したことを確認する可視化データ)がある。この場合も、トランザクション・データと可視化イベント・データが、1:1で対応するとは限らない。例えば、荷物をいくつかに分けて出荷する場合、1つの発送通知書に、複数の可視化イベントが結び付く。

- 可視化イベントが、対応するビジネス・トランザクションなしに起こることがある。例えば、小売店で“バックルーム”に在庫している商品を、消費者に陳列する売り場に移す場合が当てはまる。これは、消費者への品ぞろえを評価するうえで、きわめて重要度の高いイベントだが、対応するビジネス・トランザクションはない。
- ビジネス・トランザクションが、対応する可視化イベントなしに起こることもある。例えば、買い手が、“order”メッセージをサプライヤに送る場合、法的にも正式のやり取りであっても、注文製品が所在する物理世界では、何も起こっていない(発注時点では、注文製品がまだ存在しないこともある)。

2.6 典型的なIT環境の中のEPCIS

下の簡略化した図は、企業の典型的なIT基盤の中で、EPCISがどのように組み込まれるかを示している。



便宜上この図では、前述のマスター・データとトランザクション・データを処理する従来からのITコンポーネントを、すべて“バックエンド・アプリケーション”としてひとまとめにしている。具体的にどのようなコンポーネントがそれに当たるかは、会社ごとに違う。一般には、ERP(エンタープライズ・リソース・プランニング)システム、WMS(倉庫管理システム)、MDM(マスター・データ管理)システムなどが見られる。

可視化データは新しいタイプのデータであり、前節で触れたように発生量がずっと大きいことも多いため、専用のITコンポーネントを新たに用意して、可視化データを処理させるのが普通である。コンポーネントには、次のものがある：

- **EPCISリポジトリ**：可視化データのデータ格納場所であって、会社や組織の内部で生成したすべてのEPCISイベントのほか、取引相手から受け取ったEPCISイベントをすべて収める。
- **EPCISキャプチャ・アプリケーション**：工場、倉庫、店舗その他の、会社や組織の“端”に置くソフトウェア・アプリケーションであり、ビジネス・プロセスの各ステップが遂行されるごとに、EPCISイベントを生成する。
- **EPCISアクセス・アプリケーション**：企業や組織で定めた目的(2.3 節で記載した各種の用途)を達成するためEPCISイベントの処理をする、組織レベルのソフトウェア・アプリケーション。バックエンド・アプリケーションにつながるだけが目的の簡単なものから、EPCISデータを使って新しいビジネス処理を実行する、複雑なアプリケーションまで考えられる。

EPCIS標準は、次の2つのインタフェースを定義している：

- **EPCISキャプチャ・インタフェース**：EPCISキャプチャ・アプリケーションが、EPCISイベントをEPCISリポジトリに送るためのインタフェース(リアルタイム処理の場合は、EPCISアクセス・アプリケーションに直接送る)
- **EPCISクエリ・インタフェース**：格納されているEPCISイベント・データを、EPCISアクセス・アプリケーションが取り出すためのインタフェース

そのほかにも、ITコンポーネントとの間で次のようなやり取りが行われる：

- EPCISキャプチャ・アプリケーションは、バーコード・スキャナやRFIDリーダなどのAIDC(自動認識及びデータ取得)機器から、頻繁に入力を受け取る(ALE フィルタリング & コレクション・エンジンソフトウェアなども経由)。特に、バーコードやRFIDの読み取りが、ビジネス・プロセスの該当ステップの発生を認識するトリガーになる場合など。
- EPCISキャプチャ・アプリケーションは、バックエンド・アプリケーションとやり取りして、製品のマスタ・データや、入荷した出荷品の発注書情報など、関連するビジネス・コンテキストの情報を取得する。
- EPCISアクセス・アプリケーションは、バックエンド・アプリケーションとやり取りして、関連するビジネス・コンテキストの情報を取得するか、EPCISイベント・データから導き出した新しい情報を送る(あるいはその両方を行う。)
- EPCISアクセス・アプリケーションは、取引会社との間でEPCISデータの交換を取り次ぐ。

2.7 EPCISとGS1標準

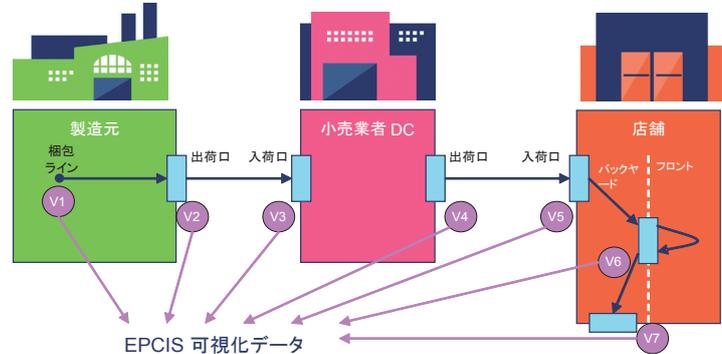
GS1標準は、サプライチェーン内のオブジェクトの情報を、識別、取得、共有するものに分かれる。EPCISは、“共有”グループの標準にあたり、2.5 節に記載したように、マスタ・データとトランザクション・データについての、ほかのGS1データを共有するための標準の仲間に加わる。“識別”グループの標準は、現実世界のオブジェクトに識別子を与え、EPCISイベントを参照できるようにする。“取得”グループの標準は、物理的な世界を情報の世界にリンクする役目をし、2.6 節で述べたように多くはEPCISキャプチャ・アプリケーションに情報を入力する。



3 EPCISイベントの構造

EPCISイベントの情報は、物理/デジタル・オブジェクトを処理するビジネス・プロセスの各ステップで何が起こったかの要点を、*What, Where, When, Why* の4つのディメンションで表現して記録する。この章では、ビジネス・プロセスの特定のステップを表したEPCISイベントを1つ取り上げて、4つのディメンションにどのようなデータが入るかを詳しく見てみる。4章では、さらにビジネス・プロセスのさまざまなステップに対して、EPCISをどのように設計するかを説明する。

例にするビジネス・プロセス・ステップは、2.2節に記載したプロセス・フローのうちの、ステップV3である。全体としては、商品が製造されて、小売業者の物流センターに出荷され、さらに店舗に配送される流れを示している。そのうちステップV3は、製造元から送られた商品が、小売業者の物流センターに入荷する段階である。ここではさらに、商品を自転車やテレビなどの、大型の消費者製品と仮定し、個々の商品を梱包箱に詰め、その梱包箱をパレットに積み付けるなどの作業は考える必要がないものとする。出荷商品は、GTIN+シリアル番号で識別される商品1個の構成になる。



ステップV3のEPCISイベントは、次のデータを含む：

- *What* ディメンションは、入荷した製品を表す。ここでは製品のGTIN+シリアル番号を使う。
- *When* ディメンションは、入荷が起こった時点を示す。
- *Where* ディメンションは、製品を入荷した場所を伝える。ここでは小売業者の物流センター。
- *Why* ディメンションは、ビジネス・コンテキストを与える。ここでは、ビジネス・プロセスのステップが“receiving”であること、そして製品の状態が、サプライチェーンを前向きに進む通常の進捗中であることを表し、さらに、関係する発注書と請求書などの、ビジネス・トランザクション・ドキュメントへのリンク、そして入荷前後で変わる所有者(ここでは製造業者と小売業者)などを示す。

以下の節で、それぞれのデータ・ディメンションの情報内容を見ていく。

3.1 *What* ディメンション

EPCISイベントの*What* ディメンションは、イベントに関連する物理/デジタル・オブジェクトを示す。GS1総合仕様書とGS1タグ・データ標準で説明されているように、商品の識別にはGTIN、GTIN+バッチ/ロット番号、またはGTIN+シリアル番号を使う。

パレットと物流単位にはSSCCを使う。そのほかのGS1標準が対象とするオブジェクトとしては、文書を識別するGDTI、個別資産を識別するGIAI、繰り返し使用する資産を識別するGRAI、サービスを識別するGSRN、クーポンを識別するGCN、部品やコンポーネントを識別するCPIDなどがある。

ステップV3の例では、GTIN+シリアル番号で商品を識別している(これはSGTIN(シリアルライズドGTIN)とも呼ばれる)。したがって、ステップV3のEPCISイベントの*What* ディメンションには、入荷した商品のSGTINが入る。

3.2 *When* ディメンション

EPCISイベント・データの*When* ディメンションは、いつイベントが起こったかを伝える。このディメンションには、3つのデータ要素がある：

- **Event Time:** イベントが起こった日時。
- **Event Time Zone Offset:** イベントが起こった場所と時点のタイムゾーン。アプリケーションが、現地時間でイベント時刻を表示するときこれをを使う。例えば、商品がカリフォルニアからブリュッセルに出荷される場合、Event Time Zone Offsetを使って、出荷日時を米国太平洋標準時で表示し、入荷日時を中央ヨーロッパ時間で表示することができる。

- Record Time**: EPCISイベントが、EPCISリポジトリに記録された日時。EPCISイベントのほかのフィールドとは性質が異なり、イベントが発生した時点には情報が入らず、イベントの間に起こったビジネス・ステップについては何も記述しない。Record Timeは、EPCISリポジトリのデータ・クエリの際の、記録管理の仕組みとして働き、クエリから返されたイベントが前回のクエリより後に発生したイベントかどうかを判断する際に利用する。

ステップV3の例では、製品を入荷した日時がEvent Timeになり、Event Time Zone Offsetで、その場所のタイムゾーンを記録する。

3.3 Whereディメンション

EPCISイベントのWhere ディメンションには、イベントが物理的にどこで発生し、イベントの後に商品がどこに所在するかを記録する。

EPCISイベントには、readPointとbusinessLocationの2つのタイプの場所を収録できる。readPointは、イベントの発生場所である。businessLocationは、次のイベントが発生するまでの間、その後オブジェクトが所在すると考えられる場所である。場所は、GS1 GLN(グローバル・ロケーション・ナンバー)、GLN+拡張フィールド、GLN以外に業界で定めた識別子、あるいは経緯度を使用して識別することができる。

例えば、入口を通過するときに商品がスキャンされる場合、通過した入口が、イベントの取得地点になる。扉口にオペレータが立っている場合もあれば、あるいは、入口に備え付けたりーダがイベントを取得する場合もあるが、これがreadPointである。そして、入口を通過した商品は、特定の場所に置かれる。この置かれる場所が、businessLocationである。場所は、GLN+拡張フィールドを使うと、例えば倉庫の中の特定の位置にある、特定の収納箱などのように、非常に細かい水準で識別することもできる。もっと大まかな水準で、例えば特定の建屋のどこかでよければ、GLNのみで十分である。可視化データを取得するうえで大切な点は、場所をどのように識別するかを理解しておくことである。

補注: 場所についてのマスタ・データを、内部システムと、取引相手との間で同期させておくことが、きわめて重要である。EPCISでGLNやSGLNを使って場所を指すときに、関係者全員が必ず同じ場所として理解できなければならない。

ステップV3の例では、Read Pointは製品を入荷した場所である。ここでは小売業者の物流センターの特定の入荷ドックとし、GLN+拡張フィールドで表す。Business Locationは、入荷後の製品の所在であり、ここでは特定の場所を明示せずに小売業者の物流センターとする。この場合、拡張フィールドのないGLNで表す。

3.4 Whyディメンション

EPCISイベントのWhy ディメンションは、イベントが発生したときのビジネス・コンテキストを記述する。次のデータ要素を、任意に組み合わせた内容にできる:

- Business Step**: そのイベントの時点で、ビジネスの観点から見て何が行われたかを示し、ビジネス・プロセスのうち、どのステップが起こっていたかを表す。例えば“commissioning”、“creating_class_instance”、“inspecting”、“packing”、“picking”、“shipping”、“retail_selling”がある。GS1コア・ビジネス・ボキャブラリ(CBV)標準では、標準のビジネス・ステップ値の一覧を記載している。これについては[3.6](#)節で取り上げる。
- Disposition**: Whatディメンションで示した物理/デジタル・オブジェクトが、そのイベントの後にどのようなビジネス状態になるかを示す。例えば“active”、“in_progress”、“in_transit”、“expired”、“recalled”、“retail_sold”、“stolen”などがある。GS1 CBVが、標準のディスポジション値の一覧を記載している。
- Business Transaction List**: そのイベントが関係するビジネス・トランザクションを示す(複数でもよい)。各ビジネス・トランザクションを2つの識別子で表し、1つの識別子で、どのタイプのビジネス・トランザクションかを示し、もう1つの識別子で、そのタイプの特定のビジネス・トランザクションを挙げる。ビジネス・トランザクションのタイプとしては、発注書(“po”)、船荷証券(“bol”)、発送通知書(“desadv”)などがある。GS1 CBVが、標準のビジネス・トランザクションのタイプ値の一覧を記載している。
- Source List, Destination List**: EPCISイベントが、所有権や、責任、管理の移転を伴うビジネス移転の一部である場合に、ビジネス・コンテキストに情報を付け加える。ビジネス・トランザクションと同じように2つの識別子で表し、ソース/デスティネーションのタイプと、それぞれのタイプのソース/デスティネーションの識別子を示す。GS1 CBV(7.4.2節)では、標準のソース/デスティネーション・タイプとして、“owning_party”、“possessing_party”、“location”の3種類を区別している。

ステップV3の例では、EPCISイベントのWhyディメンションの内容は、次のようになる：

- **Business Step**：GS1 コア・ビジネス・ボキャブラリで定義されているビジネス・ステップReceiving (入荷)。
- **Disposition**：GS1 コア・ビジネス・ボキャブラリで定義されているディスポジションIn Progress (進行中)。これは、製品がサプライチェーンの中を、通常の前向きに流れていることを表す。
- **Business Transaction List**：小売業者の発注書と、製造業者の請求書の2つが考えられる。
- **Source、Destination**：ソース側の所有者が製造業者で、デスティネーション側の所有者が小売業者。

3.5 EPCISイベントのタイプとアクション

物理/仮想世界のオブジェクトに何が起きているかを記述する4つのディメンションは、4つのタイプの“EPCISイベント”からどれか1つを使って記録する。下記は、それぞれのタイプのEPCISイベントを、高い水準から見てまとめたものである。詳しくはEPCIS 1.1標準7.4節を参照のこと。

- **EPCISEvent**：全イベント・タイプの一般基本クラス
 - **ObjectEvent**：物理/デジタル・オブジェクト(複数でもよい)に発生したイベントを表現する。例えば、SSCCで表されたパレットの出荷、または入荷など。イベントのタイプとして最も簡単なものであり、もっともよく使われる。
 - **AggregationEvent**：1つ以上のオブジェクトが、物理的に紐づけされる、または紐づけを解除されるイベントを表現する。例えば、パレットへの梱包箱の積付け、またはパレットからの梱包箱の取出しなど。ObjectEventに続いてよく使われるイベント・タイプであり、この2つのイベント・タイプで、標準的なビジネス・プロセスに起こるイベントの大半に対応可能である。
 - **TransformationEvent**：インプット・オブジェクト(材料オブジェクト)の全体または一部が消費されて、アウトプット・オブジェクト(生成オブジェクト)が生成するイベントを表現する。その中では、どのインプット・オブジェクトが、どのアウトプット・オブジェクトの生成に使われても構わない。例としては、バターとチョコレート・チップをクッキー生地混ぜ込んで、クッキーを焼き上げる工程がこれにあたる。原材料が“トランスフォーメーション”されて、製品ができ上がり、それを包装して、EANやUPCを使って“チョコレートチップ・クッキーの個装”を表すラベル表示をすると、やがて小売店で読み取られる。
 - **TransactionEvent**：1つ以上のオブジェクトが、1つ以上の特定のビジネス・トランザクションと、関連付けられる/関連付けを解除されるイベントを表現する。例えば、チョコレートチップ・クッキーを梱包したパレットや梱包箱を、請求書に結び付ける場合など。

各イベント・タイプは、TransformationEventを除いて、さらに“アクション”によって意味付けされる。詳しくは、本ガイドラインの4.5節を参照のこと。

3.6 EPCISとコア・ビジネス・ボキャブラリ(CBV)

EPCIS標準[EPCIS1.2]が、組織内だけでなく、組織を越えた情報交換の仕組みを定義するのに対して、コア・ビジネス・ボキャブラリ(CBV)は、そこに組み合わせて使用する各種のボキャブラリ要素と、その値を規定している。ボキャブラリの識別子と定義を定めることにより、EPCISデータの交換にコア・ビジネス・ボキャブラリを使う当事者のすべてが、データの意味内容を、食い違いなく理解できるようになる。

このような目標を達成するための基本的な機能を提供するのがCBVである。特にEPCIS抽象データ・モデルのコア(中核)になり、データ共有をしたいと考えている、あるいはする必要がある多くの業界に共通する幅広いビジネス場面に適用可能なボキャブラリを定義するために作られている。サプライチェーンの中の各当事者が理解できる、実用的な値と定義を用意するのが意図である。

また、特定のエンドユーザ要件に対しては、その業界や、ユーザ・グループ、あるいは単独ユーザによって定義された追加のボキャブラリ要素を補うことで対応できる。

CBVIは、次の標準ボキャブラリに対して、識別子の構文(URI構造)と、具体的なボキャブラリ要素の値及びその定義を規定している:

- ビジネス・ステップ識別子
- ディスポジション識別子
- ビジネス・トランザクション・タイプ
- ソース/デスティネーション・タイプ
- エラー理由識別子

CBVIは、次のユーザ・ボキャブラリに対して、識別子構文のオプションを用意している:

- オブジェクト
- ロケーション
- ビジネス・トランザクション
- ソース/デスティネーション識別子
- トランスフォーメーション識別子
- イベント識別子

CBVIは物理的なロケーション、当事者、商品を示すマスタ・データの属性と値を提供するもので、これにはGTINレベル、ロット・レベル、インスタンス・レベルの商品マスタ・データ属性がある。

3.7 まとめ

4つのディメンション*What*、*Where*、*When*、*Why* を合わせると、1つのEPCISイベントの情報ができ上がる。前述のステップV3のEPCISイベントの内容をまとめると、次の表のようになる:

表3-1 ビジネス・プロセス例のステップV3におけるEPCISイベントの情報内容

ディメンション	データ要素	内容	備考
	Event Type	Object Event	
	Action	OBSERVE	
What	EPC List	1 要素のリスト: GTIN 10614141123459 シリアル番号 12345	入荷した製品を示す。
When	Event Time	Sep 23, 2012, at 10:12am UTC	製品を入荷した時点
	Event Time Zone Offset	-05:00	現地時間は、UTC より 5 時間遅れ
Where	Read Point	GLN 5012345678900 拡張 D123	製品を入荷した場所、ここでは物流センターの特定の入荷ドック
	Business Location	GLN 5012345678900	このイベントの後、製品があると予想される場所、ここでは物流センター全体のどこか
Why	Business Step	Receiving (CBV より)	入荷のビジネス・ステップであることを示す、CBV 1.1 で定義されている標準の識別子
	Disposition	In Progress (CBV より)	製品が、サプライチェーンを前向きに進む通常の進捗中であることを示す、CBV 1.1 で定義されている標準の識別子
	Business Transaction List	2 つのビジネス・トランザクションの 参照番号を指すリスト: Purchase Order: GLN 5012345000015 PO# ABC123 Invoice: GLN 0614141000012 Inv# XYZ987	ビジネス・トランザクションの参照番号は、それぞれ全世界で唯一の GLN を用いることで、番号発行元のシステムや当事者を識別する。 “Purchase Order”と“Invoice”は、ビジネス・トランザクションのタイプを示す、CBV 1.1 で定義されている標準の識別子

ディメンション	データ要素	内容	備考
	Source List	1つのソースから成るリスト: Owning Party: GLN 0614141000012	入荷は、所有権をソースからデスティネーションに移転する過程の1ステップであり、ここではソース側の所有者(出荷元)を、GLNで示す。 “Owning Party”(所有者)は、ソースのタイプを表す、CBV 1.1で定義されている標準の識別子
	Destination List	1つのデスティネーションから成るリスト: Owning Party: GLN 5012345000015	入荷は、所有権をソースからデスティネーションに移転する過程の1ステップであり、ここではデスティネーション側の所有者(受け手)を、GLNで示す。 “Owning Party”(所有者)は、デスティネーションのタイプを表す、CBV 1.1で定義されている標準の識別子

4章では、システムを設計する過程を詳しく述べながら、最終的に、標準に準拠したEPCISデータが完成するまでを説明する。

4 EPCISを使った可視化システムの設計

可視化システムの構築には、EPCIS標準の技術的な理解とともに、構造化された方法論が必要である。以下に挙げる方法では、イベント取得に用いる技術は考えずに、ビジネスの視点で可視化プロセスを分析する。まずプロセスを十分に描き出してから、可視化イベントを拾い出して、その内容を明確化する。このガイドでは主にEPCISデータのビジネスへの応用を扱っているため、機器レベルの技術事項は省いている。

可視化システムのモデリング手法は、次のステップを取る：

1. 可視化の目標と要件の取りまとめ
2. ビジネス・プロセスのフローの書出し
3. プロセス・フローのビジネス・ステップへの分解
4. 可視化イベントが必要なビジネス・ステップの決定
5. 各ステップの遂行を可視化イベントとしてモデル化— ビジネスへの応用の観点から見てどの情報が必要かを把握する
6. 可視化イベントに入れるデータ・フィールドの決定
 - a. 標準のEPCISデータ・フィールドから出発
 - b. 必要に応じて拡張フィールドを定義
7. 各データ・フィールドを埋めるボキャブラリの決定(CBV標準7章、8章を使用)
8. 可視化データ・マトリクスへの可視化イベントの記録

これらのステップを、通常の簡単な物流プロセスを例にして説明していく。そのほかの場面からむ事柄については、後の章で述べる。

4.1 ステップ1: 可視化の目標と要件の取りまとめ

会社や組織において、サプライチェーンを流れる物の動きを知り、追跡する要求がますます高まっているが、可視化システムを導入するにあたっては、全体の目標と目的にしっかりと重点を置くことが重要である。つまり“解決したい問題は何か？”である。

目標としては、政府規制への対応や、出荷業務の効率化、あるいは顧客の求める商品がどこにあって、いつ配達できるかを把握することにより可能となる高度な顧客サービスなど、さまざまである。

EPCIS導入の検討に取り掛かる第一歩は、目標を定めて、その達成に必要な条件を明確に書き表すことである。例えば、製品のトラックとトレースについての規制対応が目的ならば、どんなデータが、プロセスのどの時点で必要であって、どこにデータを置いて、外部のどこに向けてどのようにデータを送るか、などを理解しておかなければならない。

全体の要件が理解できた後は、詳細なプロセス・フローと具体的なデータの条件を、EPCISとコア・ビジネス・ボキャブラリに基づいて決めていくことができる。

4.2 ステップ2: ビジネス・プロセスのフローの書出し

通常の下流方向に進む簡単な物流プロセスを例に取り上げて、フローを実際に考えてみる。このビジネス・フローは、以降の節においても、設計プロセスの中のほかのステップを説明する際に用いる。

このビジネス・プロセスでは、まず製造元の工場で商品が製造される。商品は、工場から小売業者の物流センターに向けて出荷され、そこで受け取られて倉庫に収められる。その後、物流センターから店舗に配送され、そこで受け取られた後、消費者に販売される。

図4-1 ビジネス・プロセス・フローの例



このビジネス・プロセス全体のフローは、次のようになる：

1. 品物が製造され、でき上がった製品が梱包箱に詰められ、さらにパレットに積まれる。
2. 製品が、トラックで製造元の工場から小売業者の物流センターに出荷される。
3. 製品が、小売業者の物流センターに到着し、入荷されて倉庫に収められる。
4. 製品が、小売業者の物流センターから、トラックで店舗に配送される。
5. 製品が、店舗に到着し、入荷されて店舗の倉庫に収められる。
6. 製品が、店舗の倉庫から、売り場に出される。
7. 店舗で、製品が消費者に販売される。

4.3 ステップ3: プロセス・フローのビジネス・ステップへの分解

下流方向に進む簡単な物流プロセスの例におけるプロセス・フローは下の図のようになる。水色の矢印が流れを示し、白地の枠が、それぞれプロセスの中の1つのステップである。水平軸は、左から右方向に流れる時間とともに、製品が場所を移す際の位置も表している。

この例には、個々の製品が梱包箱に詰め込まれ、梱包箱がパレットに積まれて、パレットがトラックに積み込まれるという、アグリゲーションの階層関係がある。このような場合、それぞれの階層関係が生じるステップを、縦軸を使って表すとわかりやすい。アグリゲーションの階層のないプロセス・フローならば、図は水平方向にだけ延びるか、縦軸は、フローの別の側面を表すために用いる。この段階では、フローの各ステップを、できるだけ明瞭に表すことを考える。

フロー図に表れるステップの全部がEPCISイベントを生成するわけではない。この点は次の節で取り上げる。

図4-2 前向き物流プロセスのフロー(1/2)

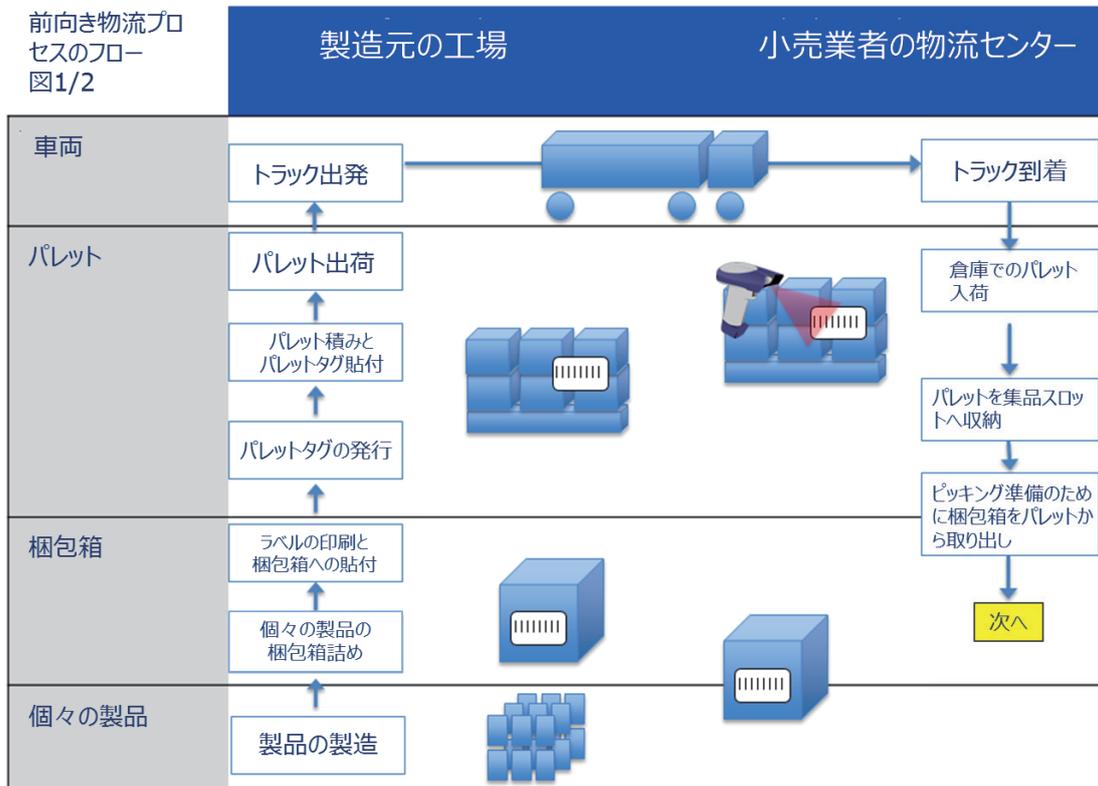


図4-3 前向き物流プロセスのフロー(2/2)



4.4 ステップ4: 可視化イベントが必要なビジネス・ステップの決定

ビジネス・プロセスの中の全部のステップに、可視化イベントが必要なわけではない。それぞれのビジネス・ステップに、イベント・データが必要かどうかは、そのデータが持つ価値と、データを収集する実現性のトレードオフで判断される。

どのデータに価値があるかは、次のような視点から考えることができる：

- このステップについて詳しい可視化イベント情報を持つことが、ビジネス・アプリケーションに役に立つ入力をもたらすか？
- このステップの情報は、アプリケーションが、ほかのステップの情報を理解するうえで必要になるか？例えば、“shipping”ステップのイベントがパレットIDを持つ場合、出荷したパレットの中身を知るためにはその前の“packing”ステップのイベントを取得しておく必要がある。
- このステップの情報は、取引相手が必要としないか、または政府規制によって求められないか？

データ収集の実現性は、次のような点から考える：

- このステップに関連する物理/デジタル・オブジェクトには、適切な識別子があるか？もし無ければ、新たに識別子を与えることは可能か？
- 物理オブジェクトの場合、電子タグやバーコードなどのデータキャリアを使って、識別子を表示することは可能か？それが難しい場合、ほかに識別子を取得する方法はないか？
- 可視化イベントのデータを取得するために、作業手順を変更することは可能か？ここで考えなければならないのは、必要な設備の費用(バーコード・スキャナ、電子タグ・リーダ、ソフトウェアその他)や、プロセス自体への影響(工数の増加、所要時間の加算その他)などである。

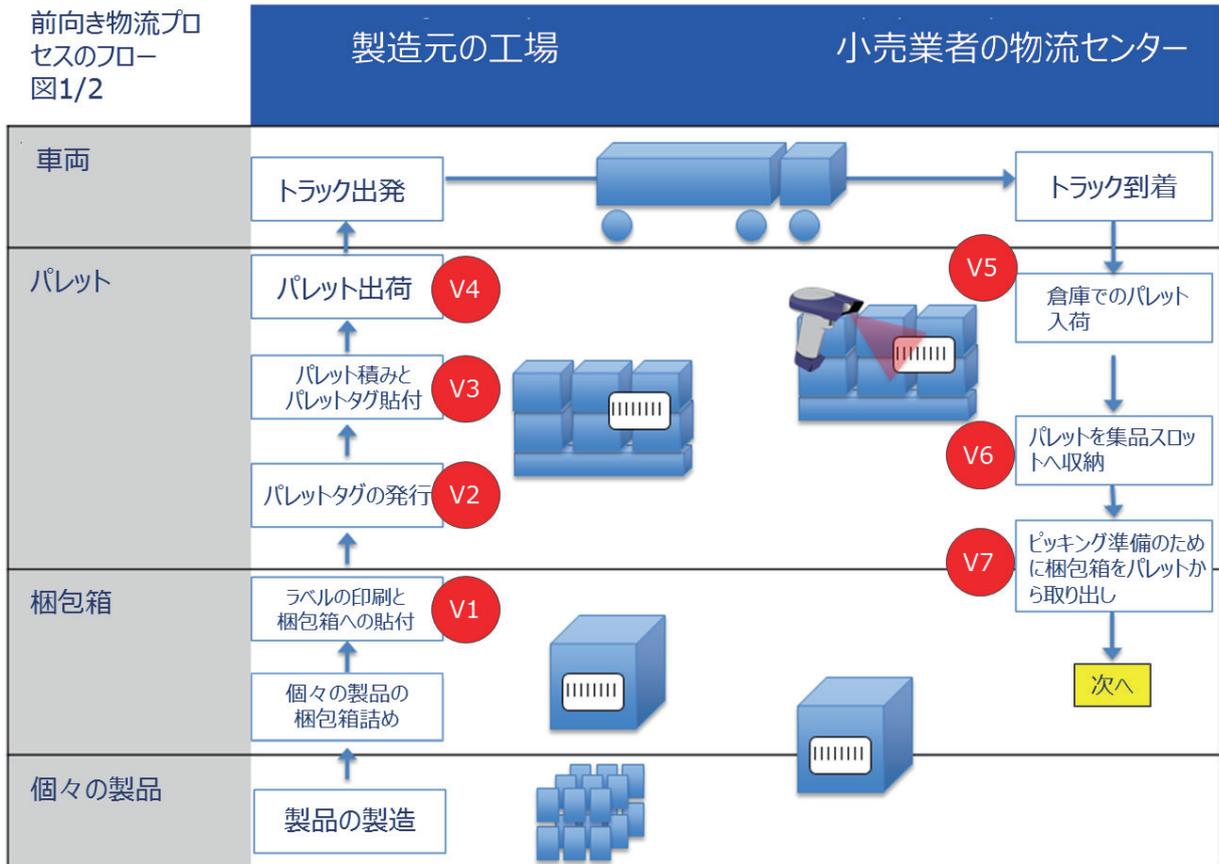
この例では、それぞれの拠点で何が出荷されて、何が入荷されたかの情報が、ビジネス上の観点から必要であると仮定する。多くの場合、何が“コミッション”されたかの記録、つまり新しい識別子を生成した際のイベントを取得することも必要になる。また、実際上の観点から、個品ではなく、梱包箱とパレットのレベルのデータのみ取得可能であることとする。さらに、パレットの輸送トラックに識別子はなく、どのトラックを使ったかは追跡不可能であることとする。

これらを考え合わせると、プロセス例の製造元側では、次のステップで可視化イベントを取得することになる：

- **V1**：梱包箱ラベルの印刷と貼付(コミッション – 後続のステップを理解するために必要)
- **V2**：パレット・ラベルの印刷と貼付(コミッション – 後続のステップを理解するために必要)
- **V3**：梱包箱のパレット積み(パレット識別子の読み取りから、出荷の内容を推定するために必要)
- **V4**：パレットの出荷

これらを、下図の朱色の丸番号V1、V2...で示している。番号のないステップでは、可視化イベントを取得しない。

図4-4 前向き物流プロセスのフローと可視化イベントの取得ステップ



4.5 ステップ5:各ステップ遂行を可視化イベントとしてモデル化

ここからは、ビジネス・プロセスの中から選び出した各ステップにおいて何が起きているかを記録するための、EPCISデータを設計する段階に入る。まず、各ステップの状況にもっともよく合ったイベント・タイプを、3.5節に挙げた中から選ぶ。選定したイベント・タイプによって、イベントのWhatディメンションの情報が持つ構造が決まる。

どんな物理/デジタル・オブジェクトがイベントに関連していて、互いにどんな関係にあるかに基づいて、イベント・タイプを選ぶ。ほとんどの場合は、次の3つのタイプのどれかから選ぶことになる：

- **ObjectEvent**：1つ以上のオブジェクトがイベントに関連していて、そのすべてが同じようにイベントに参加している場合に、これを使用する。群を抜いて一番よく使われるイベント・タイプである。

- **AggregationEvent**: 1つの“親”オブジェクトと、1つ以上の“子”オブジェクトの間の、物理的なアグリゲーションが関係するイベントに、これを使用する。アグリゲーションとは、例えば、製品(“子”)を12個詰めのカートン(“親”)に梱包する場合が当てはまる。そのほかには、梱包箱のパレット積み、商品の運搬箱への詰込み、カートンのトラックへの積込み、コンテナの船積み、コンポーネントのアセンブリ作業なども当てはまる。どの場合においても、子は親にアグリゲーションされている間も元の識別を保っている。つまり、アグリゲーションは可逆的である(“ディスアグリゲーション”が可能である)。
- **TransformationEvent**: 1つ以上の“インプット(材料)”オブジェクトを消費して、1つ以上の“アウトプット(生成)”オブジェクトを作り出すようなプロセスのイベントに、これを使用する。後で子を親から取り出せるアグリゲーションと違って、トランスフォーメーションの場合、インプット・オブジェクトは、イベント以降は存在しなくなる。トランスフォーメーションの例としては、原材料を製法に従って混合して最終製品を作る作業や、商品を再包装して元のパッケージがなくなり、新しいGTINラベルを表示した新しいパッケージにする場合、あるいはスモークサーモンなどのように、生の魚を燻製製品にする場合などが挙げられる。

もう1つのイベント・タイプ**TransactionEvent**は、1つ以上のオブジェクトが、1つ以上のビジネス・トランザクションと、確定的に関係付けられる(関係付けを解除される)場合に使用できる。しかし、どのイベント・タイプでも、ビジネス・トランザクションは*Why* ディメンションに含むことができるため、TransactionEventタイプが必要になることはまれである。

ObjectEventタイプとAggregationEventタイプは、もう1つという付加的な修飾子を持つ。アクションによって、オブジェクトとアグリゲーションそれぞれのライフサイクルにイベントがどのように関係しているかを表現できるようになる。具体的には、次のようになる:

- **ObjectEvent**のアクション値:
 - ADD: そのオブジェクトの、ライフサイクルの始まりにあたるイベント。これより前に、そのオブジェクトのイベントは存在できない。ビジネス・ステップが“commissioning”のときに、使われることがもつとも多い。
 - DELETE: そのオブジェクトの、ライフサイクルの終わりにあたるイベント。これより後に、そのオブジェクトのイベントは存在できない。“decommissioning”や、“destroying”、あるいは消費者への販売を表すビジネス・ステップ(販売後の追跡の可能性がない場合)など、ライフサイクルの終わりのビジネス・ステップで使われることがもつとも多い。
 - OBSERVE: そのほかの場合すべて。
- **AggregationEvent**のアクション値:
 - ADD: イベントの中で、子がアグリゲーションに加わるもの。例えば、商品を梱包箱に詰める場合など
 - DELETE: イベントの中で、子がアグリゲーションから取り除かれるもの。例えば、商品を梱包箱から取り出す場合など
 - OBSERVE: イベントの間、親と子がアグリゲーションの状態にあるが、子の追加や除外は起こらないもの

TransactionEventにも、アクション修飾子がある。内容はEPCIS標準を参照のこと。

TransformationEventには、アクション修飾子はない。

前節の例のイベントV1からV4には、次のようにイベント・タイプを割り振ることになる:

表4-1 ビジネス・プロセス例の各ステップへのイベント・タイプの割り振り

イベント	説明	イベント・タイプ	備考
V1	梱包箱ラベルの印刷と貼付	ObjectEventADD	梱包箱を指す SGTIN のライフサイクルの始まり
V2	パレット・ラベルの印刷と貼付	ObjectEventADD	パレットを指す SSCC のライフサイクルの始まり
V3	梱包箱のパレット積み	AggregationEventADD	子(梱包箱)がアグリゲーションに加わる。
V4	パレットの出荷	ObjectEvent OBSERVE または AggregationEvent OBSERVE	下記を参照

V4イベントで、パレット出荷の処理をEPCISイベントとして記録するにあたって2つの選択肢がある。1つは、**ObjectEvent**(アクションはOBSERVE)を使って、パレットのSSCCだけを*What* デイメンションに入れる方法である。この場合、データ取得が簡単になり、イベントがコンパクトになる反面、出荷されたパレットにどの梱包箱が載っているかは、データを受け取るアプリケーションが、V3イベントも見て推定する必要がある。もう1つは、**AggregationEvent**(アクションはOBSERVE)を使って、パレット(親)のSSCCと、全部の梱包箱(子)のSGTINを、*What* デイメンションに入れるやり方である。パレットの出荷時に梱包箱のSGTINを知ることが可能で、製造元が、パレットにどの梱包箱を載せたかを明示したいならば、この方法に意味がある。V4を受け取ったアプリケーションは、V3から推定しなくても、どの梱包箱がパレットに載っているかが分かる。

このV4は、EPCISを使ってビジネス・プロセスをモデリングする際に、所々で起こる微妙な選択の一例である。このような判断をする場合、各業界に見られる共通のビジネス・プロセスに対して、標準のEPCISモデルを記載している、業界別のガイドラインが役に立つ。

4.6 ステップ6:可視化イベントに入れるデータ・フィールドの決定

基本のイベント・タイプが決まれば、次は、各イベントの*What*、*When*、*Where*、*Why* のデイメンションに、どんなデータを入れるかを定める作業である。その際、電子タグ・リーダやバーコード・スキャナなど、キャプチャ・アプリケーションでどんなデータが取得できるかという観点から取り掛かりがちだが、逆の方向から、つまりデータを利用するビジネス・アプリケーションの視点から見た方が、はるかに役に立つEPCISデータになる。自問してみる点は、“そのイベントの間に起こった事を知るうえで、ビジネス・アプリケーションにどんな情報が必要か？”である。ビジネス・アプリケーションは、*どうやってデータが得られたかを知る必要はなく、それよりもビジネスの観点から、何が起こったかを知らなければならない。*

この先は、4つのデータ・デイメンションを、1つ1つ順番に考えていくのが良いであろう。

4.6.1 *What* デイメンションの設計

What デイメンションは、イベントに関連する物理/デジタル・オブジェクトを示す。イベント・タイプに応じて、*What* デイメンションに入る情報の構造が、次のように変わる：

- **ObjectEvent**の*What* デイメンションには、オブジェクトのリストが入る。すべてのオブジェクトが、どれも同じようにイベントに参加する。
- **AggregationEvent**の*What* デイメンションには、特定のオブジェクト1つを“親”として指定し、さらにほかのオブジェクトのリストを“子”として収める。(例外が2つあり、アクションがOBSERVEの場合は親を省略できる。その場合、子がアグリゲーションの状態にあることが観察されたが、親が未知であることを示す。また、アクションがDELETEの場合は子を省略できる。この場合には、すべての子が親から切り離されたことを表す。)
- **TransformationEvent**の*What* デイメンションには、トランスフォーメーションのインプット(材料)オブジェクトのリストが1つと、トランスフォーメーションのアウトプット(生成)オブジェクト(前者とは別のオブジェクト)のリストを1つ収める。(あるTransformationEventが、TransformationIDによって別のTransformationEventとつながっている場合は、インプットまたはアウトプットを省略してよい。[5.5.2](#)節を参照。)

イベントが表すビジネス・プロセスのステップに関与するどのオブジェクトがイベントに関連しているかだけでなく、それらのオブジェクトを、イベントの中で指し示す方法も決めなければならない。EPCISでオブジェクトを示す方法は2つある：

- **インスタンス・レベル識別**：あるオブジェクトが、そのオブジェクトだけの識別子を持つとき、これをインスタンス・レベル識別と呼ぶ。インスタンス・レベル識別には、例えばGTIN(グローバル・トレードアイテム・ナンバー)+シリアル番号(合わせてSGTIN(シリアライズドGTIN)と呼ばれる)、SSCC(シリアル・ SHIPPING・コンテナ・コード)、シリアル番号の入ったGRAI(繰り返し使用する資産識別番号)、その他を使う。
- **クラス・レベル識別**：あるオブジェクトが、ほかの同種のオブジェクトにも使われている識別子を持つとき、これをクラス・レベル識別と呼ぶ。クラス・レベル識別には、例えばGTIN+バッチ/ロット番号(そのバッチ/ロットの全商品で共通)、GTIN単独、シリアル番号のないGRAI、その他を使う。

インスタンス・レベル識別は、あるイベントに現れたオブジェクトが、その前後のイベントに現れたオブジェクトと、まったく同じものであると認識できるため、アプリケーションがEPCISデータを利用するうえでより強力である。その反面、オブジェクトへのインスタンス・レベル識別の割振りは、通常クラス・レベル識別の割振りよりも複雑な作業になる。

クラス・レベル識別を使う場合は、同じクラスから複数のオブジェクトがイベントに参加することがあるため、クラス・レベル識別子には、通常数量を表す情報が伴う。以上のことから、EPCISイベントの*What* デイメンションでは、インスタンス・レベル識別も含めて、4通りの方法でオブジェクトを指し示すことができる：

表4-2 クラス・レベルとインスタンス・レベルのオブジェクト識別

インスタンス / クラス・レベル	What デिमensionの内容	意味
インスタンス	インスタンス・レベル識別子(SGTIN、SSCC、シリアル番号の入った GRAI、その他)	そのオブジェクトがイベントに参加した。
クラス	クラス・レベル識別(GTIN、GTIN+ロット、シリアル番号のない GRAI、その他)+整数値の数量	そのクラスに属するオブジェクトが、その個数だけイベントに参加した。この場合のクラスは、個数を数えることのできるオブジェクトを指す。
	クラス・レベル識別(GTIN、GTIN+ロット、シリアル番号のない GRAI、その他)+実数値の数量と単位	そのクラスのオブジェクトが、その計量値(大きさ+単位)の量だけイベントに参加した。この場合のクラスは、個数でなく量ではかる性質のオブジェクトを指す。例えば、任意の量で取り扱える液状物、任意の目方で取り扱える固形物など。
	クラス・レベル識別(GTIN、GTIN+ロット、シリアル番号のない GRAI、その他)、数量情報なし	そのクラスのオブジェクトが、不特定の個数または量だけイベントに参加した。

最後に挙げた、数量情報のないクラス・レベル識別子は、数量を決めることができない場合、またはプライバシー保護のために明記を避ける場合など、まれなケースでのみ用いる。

1つのEPCISイベントに、インスタンス・レベル識別を使うオブジェクトと、クラス・レベル識別を使うオブジェクトが混じっていてもよい。例としては、GTIN+ロットで識別する梱包箱(クラス・レベル)を、SSCCで識別するパレット(インスタンス・レベル)にアグリゲーションする場合、あるいは、トランスフォーメーション・イベントで、インプットの原材料に、クラスと量で識別するものと、GTIN+シリアル番号で識別するもの(インスタンス・レベル)があって、さらにアウトプットをGTIN+シリアル番号で識別する場合などが挙げられる。ただし、イベント内のどのオブジェクトも、一定の方法で識別しなければならない。例えば、あるオブジェクト・イベントの中に、GTINが同じでシリアル番号が違う5つのSGTINがある場合、同時にそのGTINを、クラス・レベルの識別子としてこのイベントで使うことはできない。

4.6.2 When デिमensionの設計

When デिमensionは、4つのデिमensionのうちで、最も分かりやすい。全部のイベントに必要で、常に次の2つの情報を収めている:

- **EventTime**: イベントが起こった日時。必ずタイムゾーン識別子を含めたフォーマットで表すため、一定の時点を確実に示すことができる。
- **EventTimeZoneOffset**: イベントの起こった場所の標準時からのオフセット(UTCが基準)。これによって、イベントがあった場所の現地時間でEventTimeを表示できる。

この2つのデータ要素に充当する値は自明であるため、必要な設計作業はほとんどない。

長い時間にわたって起こるビジネス・ステップについては、EventTime をステップの開始時刻にするか、終了時刻にするか、あるいは中間のどこかの時刻にするかに迷うことがある。通常は、ステップの終了時点が最も適切である。しかし、EPCISデータの設計全般について言えるように、データを利用するビジネス・アプリケーションの観点から考える必要がある。アプリケーションが、ステップの開始時刻と終了時刻の両方を知っておく必要があるならば、プロセスの開始時と終了時に、2つのEPCISイベントを使ってプロセスをモデリングすることも考えるべきである。

反対に、ビジネスの観点から見ると、別々のイベントが同時に起こる場合、すなわちそれぞれに違った*EventTime* を割り振ることが難しい場合がある。例えば、自動製造機械で、12個の製品にSGTINを割り振って(ビジネス・ステップ“commissioning”)、梱包箱に別のSGTINを割り振り(同じく“commissioning”)、その製品を梱包箱に詰める(ビジネス・ステップ“packing”)までの工程が、同時に起こる場合が当てはまる。実際には時間差があっても、機械に組み込まれたEPCISのキャプチャ・アプリケーションには区別できないということもある。このようなときは、発生するEPCISイベントに、全部同じイベント時刻を割り振ってもよい。しかし、イベントに論理的な前後関係があるならば、時系列が論理的になるようにイベント時刻をわずかにずらしておく、利用側のアプリケーションにとってずっと助かることが多い。製品を梱包箱に詰める機械ならば、“packing”のEPCISイベント(アグリゲーション・イベント)を、例えば1ミリ秒の差を人為的に加えるだけでよい。コミッションのイベントよりも後の時刻になるようにする。すると利用側のアプリケーションでは、*EventTime* で並べ替えるだけで、論理的なイベントの系列になる。

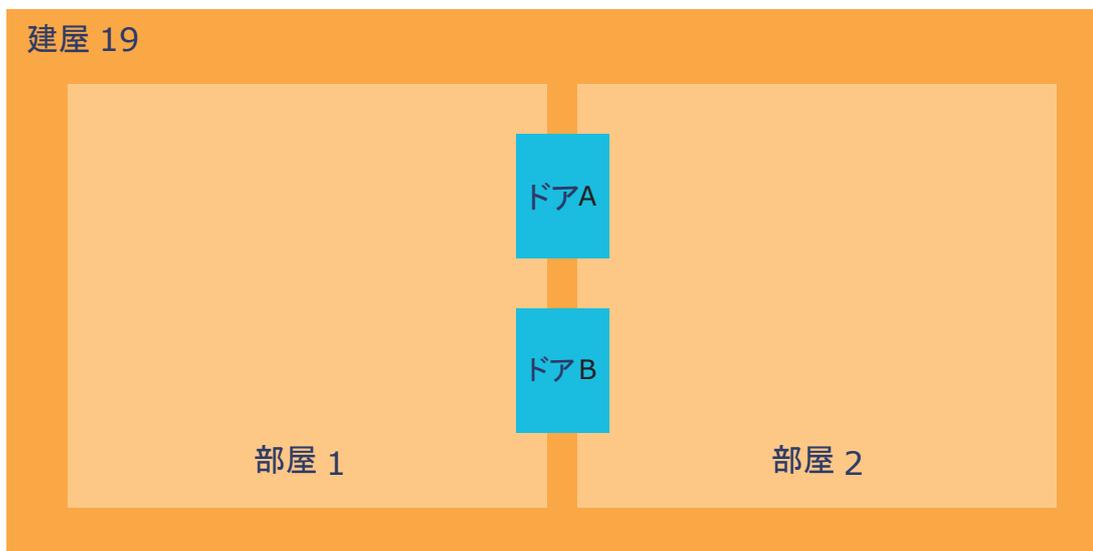
4.6.3 Where ディメンションの設計

Where ディメンションは、イベントに入ったオブジェクトの物理的な場所を表す。*Where* ディメンションの2つのデータ要素は、どちらもオプションだが、ほとんどのEPCISイベントは両方を持つ。2つのデータ要素は次の通りである：

- **ReadPoint**: *ReadPoint* は、*What* ディメンションの指すオブジェクトが、イベントの時点にどこにあったか、つまりイベントの起こった場所を表す。
- **BusinessLocation**: *BusinessLocation* は、*What* ディメンションの指すオブジェクトが、イベントの後、別のイベントが別の場所を知らせるまでの間、どこにあると予想されるかを表す。

ReadPoint と *BusinessLocation* という名前は、少し紛らわしい。例えば、ビジネスの視点から見て、まるで *ReadPoint* は *BusinessLocation* に劣らず大切であり、状況によっては、もっと重要でありそうな印象を受ける。“リード・ポイント”と“ビジネス・ロケーション”という名前の意味をくみ取ろうとするよりも、定義を覚えてしまう方が早い。*ReadPoint* は、イベントが起こった時点のオブジェクトの場所であり、*BusinessLocation* は、イベントの後のオブジェクトの場所である。

ReadPoint と *BusinessLocation* の違いは、下のように、ドアでつながった複数の部屋を持つ建物を想像すると分かりやすい：



どちらかのドアを商品が通過するごとに、例えば各ドアに取り付けた電子タグ・リーダーによって、EPCISイベントが取得されるとする。今、商品がドアAを通過して部屋1から部屋2に移動したとすると、*ReadPoint* (イベントの発生時点の場所)はドアAで、*BusinessLocation* (その後の場所)は部屋2になる。商品が部屋2の中でどのように移動しても、新しいEPCISイベントは発生しないため、部屋2の中に入った後、商品の所在について分かるのは、部屋2のどこかということだけである。商品が、ドアAの代わりにBを通過して部屋1から部屋2に移動した場合にも*BusinessLocation* は同じく部屋2のままとなるが、*ReadPoint* はドアBになる。その後、商品がドアAを逆方向に通って戻った場合、*ReadPoint* は同じくドアAで、*BusinessLocation* が部屋1になる。

BusinessLocation を持たせておくことが役に立つ理由は、“オブジェクトが今の時点でどこにあるか？”の問いに答えられることである。たまたまイベントの発生時点で問いを発すれば、*ReadPoint* が答えになるが、そのほかの時点では、最後に起こったイベントの *BusinessLocation* が、オブジェクトの現在ある場所のもっとも有効な推測になる。一方で、*ReadPoint* も有用であり、“Xが起こったのは、オブジェクトがどこにあるときか？”(該当イベントの *Why* デイメンションによってXが表された場所)といった、過去のことも教えてくれる。

Where デイメンションの設計でポイントになるのは、どの程度の細かさで場所を表すかの判断である。例えば、入荷処理で、入荷ドックを通して商品が入ってくる場合、イベントが起こった場所(*ReadPoint*)の表し方はいくつもある。具体的なもの(細かい粒度)から大まかなもの(粗い粒度)に向かう順に挙げると、例えば下のようになる：

- “XYZ社シカゴ施設、2号建屋入荷ドック#5”
- “XYZ社シカゴ施設、2号建屋入荷場”(どのドックかは不特定)
- “XYZ社シカゴ施設、2号建屋”(2号建屋のどこかは不特定)
- “XYZ社シカゴ施設”(どの建屋かは不特定)
- “XYZ社”(具体的な場所は不特定)

EPCISイベントに、どの程度の「細かさ」を使うかが判断点である。EPCISのほとんどの設計判断に共通するように、データを利用するビジネス・アプリケーションのニーズと、EPCISイベント取得時のデータ収集 実行可能性のトレードオフである。例えば、建屋の入荷ドックを区別しようとする、商品が建屋に入ったことだけを記録する場合に比べて、設備費が大きくなる。一方で、ビジネス・アプリケーションにとって、どのドックを使ったかは重要でないこともある。また、EPCISイベントの設計における「細かさ」は、組織内でのデータの取得を考えた場合と、取引相手との間で共有することを考えた場合とで変わってくる。例えば、*ReadPoint* を、内部では入荷ドックごとに取得しても、そのイベントを取引会社と共有するときには、“建屋”レベルの細かさにも集約することもある(6.7 節も参照)。

単純に、イベント以降に商品がどこにあるかは、イベントが発生した時点にどこにあるかに比べてEPCISキャプチャ・アプリケーションにとって確実性が低くなるため、*BusinessLocation* は *ReadPoint* よりも粗い水準で表すことが多い。また、*ReadPoint* を *BusinessLocation* よりも精密に表すビジネス上の必要性がなければ、*ReadPoint* を *BusinessLocation* と同じ水準の粒度にすることもよく行われる。

出荷とそれに続く入荷のように、オブジェクトが場所を移す場合は、*BusinessLocation* に特別な状況が起こる。商品が出荷されると、イベント以降の商品の所在地は、もちろん出荷元ではない。しかし、入荷のイベントが取得されるまで、商品は受け手の所でも確認されていないため、商品の所在地は受け手側でもない。したがって、出荷時に取得するEPCISイベントの正しい *BusinessLocation* は、“未知”となる。入荷処理が起こるまで、出荷段階では商品がどこにあるかは分からない。これをEPCISでは、出荷イベントから完全に *BusinessLocation* データ要素を省くことで表現される。

4.6.4 Why デイメンションの設計

Why デイメンションは、イベントのビジネス・コンテキストを説明するものであり、ビジネス・アプリケーションがEPCISデータの意味を理解するうえで、重要な役割を果たす。*Why* デイメンションのデータ要素はすべてオプションだが、ほとんどのすべてのEPCISイベントが、少なくとも *BusinessStep* と *Disposition* のデータ要素を含むことになる。ほかの *Why* デイメンションのデータ要素は、実行されているビジネス・ステップに関係する場合にのみ加えられる。

Why デイメンションのデータ要素の定義は、3.4 節で示されている。ここでは、各データ要素を加えるかどうかの判断にあたって考慮すべき点と、値の適切な選び方について述べる。

4.6.4.1 ビジネス・ステップの設計

BusinessStep データ要素は、ビジネス・アプリケーションがEPCISデータの意味を理解するうえで、もっとも重要なものである。*BusinessStep* は、イベントの発生時に、ビジネス・プロセスのどのステップが起こっていたかを示す識別子を値に取る。ビジネス・ステップの情報がなければ、アプリケーションに分かるのは、あるオブジェクトが、ある場所に、ある時点で存在していたことだけである。ビジネス・ステップによって、そのオブジェクトが全体のビジネスに、どのように関係しているかが把握できるようになる。實際上、EPCISイベントは、すべて*BusinessStep* の値をもつ。*BusinessStep* の値は、普通はshipping、receiving、packingなど、何らかの動詞に対応する語で表される。

ビジネス・ステップの値が有用であるためには、それを処理するアプリケーションに、前もってその意味を知らせておかなければならない。そのため、*BusinessStep* の値を定義する何らかの標準が必要である。それぞれの*BusinessStep* 値を、その値の持つ意味と、その値を持ったEPCISイベントをどのように解釈するか、対応付ける文書である。EPCコア・ビジネス・ボキャブラリ(CBV)は、そのような標準の1つである。全世界で使われる標準として、多くの産業分野におけるサプライチェーンのビジネス・プロセスに共通に見られる各種のビジネス・ステップを対象に、数十個のビジネス・ステップの値を定義している。分野をまたいだグローバルな標準であるため、CBVのビジネス・ステップ値を使うと、幅広いアプリケーションに理解可能なEPCISイベントになる。CBVのビジネス・ステップ値が使用できる場合には、できるだけそれを用いるべきである。

しかし、EPCISを利用するビジネス・プロセスには、CBVに定義されたビジネス・ステップ値の範囲でうまく表せないステップが出てくることもある。その場合は、特定の用途向けにほかの識別子を作り、それを用いるしかない。この場合も、識別子とその意味を定義する文書が必要である。もちろん全世界標準ではなくて、内部的な設計文書である。このような固有のビジネス・ステップ値は、少数の取引会社間や、特定業界の標準として定義されることもある。ただし、そのような値を取り入れたEPCISイベントは、それを定義している設計文書や業界の標準を知っている、狭い範囲の企業や組織の間でしか理解できないものになる。CBV以外の定義を使うかどうかを判断する際には、このトレードオフを考慮することが必要となる。

4.7節で、定義の衝突を避けながら、CBVにない識別子を作るための方法を記載している。

4.6.4.2 ディスポジションの設計

Disposition 値は、イベントの後の、オブジェクトのビジネス状態を表す識別子である。*Disposition* 値は、普通はin_progress、recalled、damagedなどのように、ビジネス・プロセス全体に関わるオブジェクトの状態を説明する形容語で表される。

Disposition の大切な用途の1つは、正常なフローと例外フローを区別することである。例えば、コア・ビジネス・ボキャブラリ(CBV)のディスポジション値“in_progress”は、サプライチェーンの中を、正常に進んでいるオブジェクトを表し、“recalled”は、製造元にリコールされているオブジェクトを表す。*BusinessStep* とは別に*Disposition* を持たせることで、次の2つのモデリングが可能になる。1つ目は、結果として例外が起こる可能性のあるイベントにおいて、どちらの結果になったかを、*Disposition* で表すことができる。例としては、*BusinessStep* が“inspecting”(CBVより)のEPCISイベントにおいて、その*Disposition* の値に、検査の結果に応じて、オブジェクトが正常であった場合には“in_progress”を、リコール対象であった場合には“recalled”を持たせる場合が挙げられる。もう1つは、例外状態になったオブジェクトが、さらにイベントを経っていく間に、*Disposition* によって、例外状態であることを示し続けることができる。例えば、“inspecting”ステップの後、リコール対象になったオブジェクトには、製造元に向かって上流側に進みながらも、*BusinessStep* 値が“shipping”や“receiving”のEPCISイベントが起こることが考えられる。*Disposition* が無ければ、これらのEPCISイベントを、正常な出荷と入荷のステップから区別するのは難しい。しかし、*Disposition* 値に“in_progress”でなく、“recalled”を持たせることで、これらのイベントが、通常とは逆向きの物流プロセスの一部であることが明示される。

BusinessStep の場合と同じように、*Disposition* の値が有用であるためには、それを処理するアプリケーションに、前もってその意味が知られていなければならない。したがって、4.6.4.1節に記載した諸点は、*Disposition* 値にも、すべて同じように当てはまる。

4.6.4.3 ビジネス・トランザクション・リストの設計

BusinessTransactionList は、ビジネス・トランザクションを参照するためのリストである。ビジネス・トランザクションのデータ自体は、EPCISとは別のシステムにある。ビジネス・トランザクションの例としては、特定の発注書への参照、特定の請求書への参照などがある。この情報は、EPCISイベントにビジネス・コンテキストを与えるとともに、EPCISデータをほかの情報システムにリンクする役割をする。

BusinessTransactionList の各ビジネス・トランザクションは、2つの識別子で構成される。1つは、ビジネス・トランザクション・タイプ識別子で、どの種類のビジネス・トランザクションを表しているかを示す(発注書、請求書、その他)。もう1つは、そのタイプの特定のトランザクションを指すビジネス・トランザクション識別子である。

BusinessStep と *Disposition* の値と同じように、ビジネス・トランザクション・タイプ識別子が有用であるためには、それを処理するアプリケーションに、前もってその意味が知られていなければならない。したがって、4.6.4.1 節に記載した諸点は、ビジネス・トランザクション・タイプの値にも、すべて同じように当てはまる。コア・ビジネス・ボキャブラリは、発注書、請求書その他の、一般的なビジネス・トランザクションのタイプについて、標準のビジネス・トランザクション・タイプ識別子を定義している。

ビジネス・トランザクションの参照の2つ目の部分、ビジネス・トランザクション識別子は、ある特定のビジネス・トランザクションを示す役目をする。ビジネス・ステップや、ディスポジション、ビジネス・トランザクション・タイプなどの値と違って、ビジネス・トランザクション識別子に固定されたリストはなく、新しいビジネス・トランザクションが発生するとともに、常に新しい識別子が作られていく。通常ビジネス・トランザクション識別子は、例えば、請求書番号がERP(エンタープライズ・リソース・プランニング)システムで作られるように、EPCIS以外の何らかの情報システムで生成される。

EPCISイベントで使うビジネス・トランザクション識別子は、世界中でユニークでなければならない。EPCISデータを処理するアプリケーションは、サプライチェーン全体からEPCISイベントを集める可能性がある。そのような状況において、サプライチェーンの中の、違った当事者の発行した請求書を混同することは許されない。

EPCISのビジネス・トランザクション・リストで使うことのできる、全世界で唯一のビジネス・トランザクション識別子の作成には、2つのやり方がある。1つは、ビジネス・トランザクションを生成するシステム側で、必ず世界でユニークな識別子を使ってトランザクションを表すことである。例えば、ERPシステム自体で、GS1 GDTI(文書識別番号)などのユニークな識別子を使うことが考えられる。正しく割り振られていれば、あるシステムの発行するGDTIが、ほかの当事者のシステムの発行するGDTIと重複することはない。しかし従来からのシステムの多くは、そのような設計がなされていない。通常のERPシステムは、12345のような番号を、各トランザクションに与えているだけである。この番号は、そのERPシステムの中でのみユニークであり、ほかのERPシステムが生成した番号と合わせたときに、重複しない保証はない。

全世界で唯一のビジネス・トランザクション識別子を作成するもう1つの方法は、従来システムの生成する識別子に、全世界でユニークにするためのプリフィックスを付け加えることである。EPCのコア・ビジネス・ボキャブラリは、発行元のGLN(グローバル・ロケーション・ナンバー)をこの目的に使うテンプレートを規定している。例として、X社が0614141123452という番号の当事者GLNを持ち、そのERPシステムが12345という発注書番号を発行した場合、CBVのテンプレートを使った全世界で唯一の識別子は、下のようになる：

```
urn:epcglobal:cbv:bt:0614141123452:12345
```

この識別子の前半urn:epcglobal:cbv:bt:は、CBVのビジネス・トランザクション識別子テンプレートを使っていることを示すプリフィックスである。後の2つの部分がGLNと、そしてERPシステムから割り振られた発注書番号である。ほかのERPシステムが発注書番号12345を割り振っても、異なるプリフィックスが与えられるため、この文字列全体は、1つの識別子として見たときに全世界でユニークになる。(1社内にERPシステムが複数あって、トランザクション番号の割り振りが重なる可能性がある場合は、それぞれのシステムのプリフィックスに異なるGLNを使用する必要がある。)

EPCISデータを処理する際は、プリフィックスを含めたビジネス・トランザクション識別子の全体を取り扱う。例えば、2つのEPCISイベントが同じビジネス・トランザクションを参照しているかを調べる場合、識別子の文字列をまるごと比較する(併せて、ビジネス・トランザクション・タイプ識別子も比較する)。しかし、EPCISデータを従来からのシステムのデータに対応付ける場合は、CBVプリフィックスの部分を見分け、識別子を解析し、どの従来システムのものか、またそのシステムではどのような形式で表されているのかを特定することが必要となるだろう。

4.6.4.4 ソース・リストとデスティネーション・リストの設計

ビジネス・プロセスのステップの中には、所有や物理的な預かりの当事者が移転する、*ビジネス移転* 処理がある。よく見られる例として出荷と入荷の2つがあるが、ほかに委託、受託、返送のほか、輸送の中間段階なども当てはまる。このようなとき、移転の両端を示す情報を入れておくことと役立つことが多い。例えば、出荷イベントは、“出荷元”の場所とともに、“出荷先”の場所を示しておくことが有用である。また、所有者や、物理的に商品を預かる当事者についても、移転の両端の当事者を示すことに意味がある。2つは、同じこともあり、異なることもある。EPCISイベントでは、ソース・リストとデスティネーション・リストを使って、これらの情報を持たせることができる。ソースとデスティネーションの情報は、EPCISイベントの*Why* ディメンションであり、ビジネス・コンテキストを与える役目をする。

ソース・リストは、それぞれソース・タイプとソース識別子が1組になった、ソースのリストである。同じように、デスティネーション・リストは、それぞれデスティネーション・タイプとデスティネーション識別子が1組になった、デスティネーションのリストである。コア・ビジネス・ボキャブラリは、3種類のソース/デスティネーション・タイプを定義しており、それぞれのソース/デスティネーション識別子を、どのように解釈するかを規定している：

表4-3 コア・ビジネス・ボキャブラリで定義されているソース/デスティネーション・タイプ

ソース/デスティネーション・タイプ	意味
urn:epcglobal:cbv:sdt:owning_party	ソース/デスティネーション識別子は、この EPCIS イベントが加わっているビジネス移転の、それぞれ始端と終端で、オブジェクトを所有する(または所有が予定される)当事者を示す。
urn:epcglobal:cbv:sdt:possessing_party	ソース/デスティネーション識別子は、この EPCIS イベントが加わっているビジネス移転の、それぞれ始端と終端で、オブジェクトを物理的に預かる(または預かる予定の)当事者を示す。
urn:epcglobal:cbv:sdt:location	ソース/デスティネーション識別子は、この EPCIS イベントが加わっているビジネス移転の、それぞれ始端と終端の物理的な場所を示す。

ソース/デスティネーション識別子は、それ自身が全世界でユニークな識別子であり、ソース/デスティネーション・タイプに応じて、当事者または物理的な場所を表す。EPC URI形式のGLN(拡張フィールドあり/なし)が使われることが多いが、CBVは、ほかにも使用できる識別子を規定している。

3つのソース/デスティネーション・タイプは、イベントのビジネス・コンテキストに応じて、ソース・リストとデスティネーション・リストの各々に、自由な組み合わせで使用してよい。通常は、どのタイプも、ソースとデスティネーションの両方を含む。

ビジネス移転全体は、複数のEPCISイベントに広がるのが普通で、イベントを生成する当事者も複数になることが多い。例えば、ごく簡単な例でも、出荷ステップのEPCISイベントが1つと、入荷ステップのEPCISイベントを1つ伴う。複雑なものになると、到着と検収のステップが分かれたり、あるいは経路の途中で貨車輸送や船舶輸送の中間ステップがあったりする。このように同じ移転に属する全ステップは、ソースとデスティネーションの情報を持ち得る。そしてソースやデスティネーションの情報を持たせる場合には、全部のイベントで共通の内容になるのが普通である。

例えば、荷物を預かる当事者がAからBに変わる場合、出荷と入荷のEPCISイベントは、どちらもタイプ“possessing party”のソースとして当事者Aを、タイプ“possessing party”のデスティネーションとして当事者Bを持つ。ソースとデスティネーションの情報の解釈については、2つのイベントで微妙に異なる。出荷イベントの場合、ソースは、移転の始点側の、*既知の* 預かり当事者であるが、デスティネーションは移転の終点側の *予定される* 預かり当事者である。入荷イベントの場合、デスティネーションは移転の終点側の *既知の* 預かり当事者であるが、ソースは、移転の始点側の、*確信される* 預かり当事者である。

タイプ“location”のソース/デスティネーションは、イベントによっては、Where デイメンションのリード・ポイントの情報と重なることがある。特に出荷などのステップのリード・ポイントは、タイプ“location”のソースと同じになり、入荷などのステップのリード・ポイントは、タイプ“location”のデスティネーションと同じになる。このようなとき、ソース/デスティネーションの情報は、リード・ポイントと一貫させるようにする。(ただし、リード・ポイントをソース/デスティネーションよりも細かい粒度の識別子で表す場合などには、同一にはならない。)

ビジネス移転の一部でないEPCISイベントには、ソース/デスティネーションの情報は入れない。

ソース/デスティネーション・リストを使用する状況の例については[5.2](#)節を参照のこと。

4.6.5 例

この節で述べた事柄を合わせて、4.4 節の例のステップV4についてEPCISイベントを設計すると、次のようになる。このイベントでは、パレット積みした梱包箱を、製造元から小売業者の物流センターに向けて出荷している。

4.5 節で述べたように、このイベントは、ObjectEventでパレットだけを示すやり方と、AggregationEventでパレットと梱包箱をともに示すやり方の、2通りに表すことができる。ここでは、ObjectEventを使う方法を取る。

表4-4 プロセス例(4.4 節)のステップV4に対するEPCISイベントの情報内容

ディメンション	データ要素	設計選択	備考
	Event Type	Object Event	上記を参照
	Action	OBSERVE	パレットのライフサイクルの、始まりでも終わりでもないため、アクションはOBSERVE (4.5 節を参照)
What	EPC List	1 要素のリスト: パレットの SSCC (インスタンス・レベル識別)	
When	Event Time	パレット出荷の日時	
	Event Time Zone Offset	パレット出荷場所のタイムゾーン	現地時間は、UTC より 5 時間遅れ
Where	Read Point	10 号建屋、出荷ドック#2	ここでは、細かい粒度でリード・ポイントを取得することを選択
	Business Location	(なし)	4.5 節で述べたように、出荷イベントのビジネス・ロケーションは省く。次の入荷のイベントが起こるまで、パレットがどこにあるかは分からない。
Why	Business Step	Shipping (CBV より)	CBV 1.1 で定義されている標準の識別子を使って、どの処理側アプリケーションも、このイベントを理解できるようにする。
	Disposition	In Transit (CBV より)	CBV 1.1 で定義されている標準の識別子を使って、どの処理側アプリケーションも、このイベントを理解できるようにする。“In Transit” は、通常の前向きの流れで、出荷元から受け手に向けて移動中であることを表す。
	Business Transaction List	2 つのビジネス・トランザクションへの参照が入ったリスト: 小売業者の発注書、製造元の請求書	ビジネス・トランザクション・タイプを表す、CBV 1.1 で定義されている標準の識別子“Purchase Order”と“Invoice”を使う。
	Source List	タイプ“owning party”のソースが 1 つ入ったリストで、ソース側の所有者として製造元を示す。	出荷は、ソースからデスティネーションに所有が移転する過程に属するステップ。ここでは、ソース側の所有者(出荷元)を表す。“Owning Party”は、ソースのタイプを表す CBV 1.1 で定義されている標準の識別子
	Destination List	タイプ“owning party”のデスティネーションが 1 つ入ったリストで、デスティネーション側に予定する所有者として小売業者を示す。	出荷は、ソースからデスティネーションに所有が移転する過程に属するステップ。ここでは、デスティネーション側の予定所有者(受け手)を表す。“Owning Party”は、デスティネーションのタイプを表す CBV 1.1 で定義されている標準の識別子

4.7 ステップ7:各データ・フィールドを埋めるボキャブラリの決定

前のステップでは、各EPCISイベントのデータ要素に、何を語らせるかを決定した。次のステップでは、言葉で表した各データ要素の内容を、コンピュータが理解できる具体的な識別子に置き換える。コア・ビジネス・ボキャブラリの、7章と8章が出发点になる。

4.7.1 What ディメンションのボキャブラリ

What ディメンションには、1つまたは複数の物理/デジタル・オブジェクトの参照が入る。ほとんどの場合、GS1識別コードを使って各オブジェクトを指し示す。例えば、商品はGTIN(例:00614141123452)とシリアル番号(例:400)によって示すことができる。EPCISでは、GTIN+シリアル番号を、EPCタグ・データ標準に従ったURI(Uniform Resource Identifier)で表す。この場合、下記のようになる:

```
urn:epc:id:sgtin:0614141.012345.400
```

この識別子の最初の部分(urn:epc:id:)は、この識別子がEPCタグ・データ標準に準拠した識別子であることを表し、次の部分(sgtin:)は、GTIN+シリアル番号(SGTIN)であることを表す。後の3つの部分が、それぞれGS1事業者コード(0614141)、インジケータ値と商品アイテムコード(012345)、そしてシリアル番号(400)である。ここで留意すべきは、EPC URIではGS1事業者コードをGTINのほかの部分から分けて書くということと、チェックデジットが無いということである。

EPCタグ・データ標準の6章と7章が、EPCISイベントのWhat ディメンションに使われるすべてのGS1識別コードについて、EPC URIの構文を規定している。しかし時には、EPCタグ・データ標準が対応していない別の識別方式を使っている物理/デジタル・オブジェクトを、What ディメンションで表すことが必要になる。コア・ビジネス・ボキャブラリ(8.2節、8.3節)は、そのような場合に使う、ほかの非EPCのURI表記方法も用意している。ただし、可能な限りGS1識別コードを使うことが望ましい。

4.7.2 Where ディメンションのボキャブラリ

ReadPoint とBusinessLocation が、Where ディメンションで物理的な場所の識別子を収めるデータ要素である。使用する識別子を選ぶには、まず場所をどのように識別するかを決める必要がある。

場所を識別するためにもっともよく使われる方法は、GLN(グローバル・ロケーション・ナンバー)などで、ユニークな識別子を与えることである。GLNIは、場所の所有者が、任意の番号を割り振って特定の位置を示すものである。どんな詳細さで割り振ってもよく(4.6.3 節を参照)、ある建物の特定の部屋など、細かくGLNを割り振ることもでき、あるいは建物全体など、大まかにGLNを割り振ることもできる。このような場合、GLNIは階層性を持つ。

例えば、各々の出荷ドックや、大きな倉庫の中の各々の収納箱など、特に細かく識別子を割り振るときには、GLNだけでは容量が足りなくなる。その場合、GLNIにGLN拡張フィールドを組み合わせて場所に割り振ることができる。GLN+拡張フィールドで細かく割り振る場合、GLN部分は、建物そのものなど、全体を覆う大きな粒度のGLNIにするのが普通である。

What ディメンションと同じように、Where ディメンションの識別子にもURI表記を使う。例えば、ある場所をGLN 0614141111114と、拡張フィールド987で表す場合には、EPCISは、GLN+拡張フィールドを、EPCタグ・データ標準に従ったURI(Uniform Resource Identifier)で表す。このタイプのURIはSGLNと呼ばれる。SGLNは、GLN+拡張フィールドと、拡張フィールドのないGLNのどちらも表すことができる。GLNが0614141111114で、拡張フィールドが987のSGLNは、以下のようになる:

```
urn:epc:id:sgln:0614141.11111.978
```

拡張フィールドのないGLNIは、拡張フィールドの代わりに0を1つ入れて、次のようにする:

```
urn:epc:id:sgln:0614141.11111.0
```

この識別子の最初の部分(urn:epc:id:)は、この識別子がEPCタグ・データ標準に準拠した識別子であることを表し、次の部分(sgln:)は、GLNか、GLN+拡張フィールドであることを表す(拡張フィールドの部分が0かどうかによる)。後の3つの部分が、それぞれGS1事業者コード(0614141)、ロケーションコード(11111)、そして拡張フィールド(GLN単独のときは0)である。EPC URI表記では、GS1事業者コードを、GLNのほかの部分から分けて書くほか、チェックデジットが無い。

時には、EPCタグ・データ標準が対応していない別の識別方式が使われている場所を、Where ディメンションで示す必要がでてくることもある。そのような場合のために、コア・ビジネス・ボキャブラリ(8.4節)は、ほかの非EPCのURI表記も用意している。ただし、可能な限りGS1識別コードを使うことが望ましい。

一方で、ユニークな識別子ではなく地理座標(経緯度)でしか場所を表せない場合もある。そのケースとして多いのは、*ReadPoint* に輸送手段の場所を記録する場合などで、例えば、航行中の船舶が外洋に出てGLNで表せる定められた場所がなく、GPS(Global Positioning System)レシーバならば使えるといった状況である。この場合、経緯度を用いて場所をURI表記することができ、この場合、下記ようになる：

```
geo:22.300,-118.44
```

この例は、北緯22.300度、西経118.44度の地理位置を表している。さらに詳しくは、コア・ビジネス・ボキャブラリの8.4.4節を参照。

4.7.3 Why ディメンションのボキャブラリ

EPCISイベントの*Why* ディメンションには、たくさんのデータ要素が入り、さまざまな種類の識別子を使う。これらはデータ要素に応じて、2つの種類に分かれる。

4.7.3.1 Why ディメンションの標準ボキャブラリ要素

Why ディメンションのいくつかのデータ要素には、サプライチェーン上の全当事者が事前知っておかなければならない概念の名前が入る。例えば、*BusinessStep* データ要素は、“shipping(出荷)”、“receiving(入荷)”などの概念を表す識別子を収める。これらの識別子は、常に何らかの標準で定義される。その目的においてもっともよく使われるのが、コア・ビジネス・ボキャブラリである。

コア・ビジネス・ボキャブラリの7.1節に、30種余りのビジネス・ステップ値が定義されている。EPCISイベントではURIで値を表し、次のようなかたちになる：

```
urn:epcglobal:cbv:bizstep:shipping
```

この識別子の最初の部分(urn:epcglobal:cbv:)は、コア・ビジネス・ボキャブラリで定義されたものであることを表し、次の部分(bizstep:)が、ビジネス・ステップ値のリストから取った識別子であることを表す。この2つの部分は、CBVで定義されたビジネス・ステップ識別子のすべてに共通である。残りの部分が、具体的なビジネス・ステップを表す。

どのビジネス・ステップ値を選ぶかは、CBVに与えられている定義に基づく。例えば、CBVの定義するurn:epcglobal:cbv:bizstep:packingの意味は、“ビジネス・プロセスの中の特定の処理であり、オブジェクトを大きな容器に収めることを伴うものであり、普通は出荷を目的にする。ある単位を別の単位にアグリゲーションする処理は、典型的にはこの時点で起こる。”となっている。

場合によっては、CBVの識別子に適当なものがないこともある。そのときは、プライベートの識別子を作成できるが、やはりURIの構文に則る必要があり、自身の管理するプリフィックスを使う。ほとんどの場合、これはインターネット・ドメイン名を利用した、HTTP URIを使う。例えば、example.comというドメイン名を持ったExample社が、“fiddling”という新しいビジネス・ステップを必要とする場合、次のようなURIを使う：

```
http://epcis.example.com/bizstep/fiddling
```

先頭にhttp://epcis.example.com/を置いているため、CBVの識別子とも、ほかの組織が作成するプライベートの識別子とも、重複する心配がない。業界団体が、標準を定めてプライベートの識別子を作成するときには、その団体のインターネット・ドメイン名をルート部に使うことができる。4.6.4.1節で述べたように、このような独自のビジネス・ステップを作成すると、取引相手にその意味を知らせておくことが必要であり、CBVで定義されたものを使うのに比べて、相互運用性の劣るものになる。

なお、上記の形式の識別子は、Webブラウザから入力するような見かけをしているが、EPCISに関する限り、ビジネス・ステップの識別子でしかなく、そのURIからアクセスできるWebページが存在するわけではない。一方で、そのURIのWebページがあれば、作成したビジネス・ステップの意味を説明する文書を置いておくのに、うってつけの場所になる。

以上のことは、*Why* ディメンションのいくつかのほかのデータ要素でも同じである。まとめると下のようになる。

表4-5 コア・ビジネス・ボキャブラリで定義されている標準ボキャブラリ識別子の例

EPCIS データ要素	CBV 該当節	例
<i>BusinessStep</i>	7.1	urn:epcglobal:cbv:bizstep:shipping
<i>Disposition</i>	7.2	urn:epcglobal:cbv:disp:in_transit
<i>BizTransaction</i> (タイプ のサブフィールド)	7.3	urn:epcglobal:cbv:btt:po
<i>Source/Destination</i> (タイプ のサブフィールド)	7.4	urn:epcglobal:cbv:sdt:owning_party

どのデータ要素も、CBVで定義の識別子を使うのが最良の選択であるが、それに対応できない場合は、前記のやり方で、プライベートの識別子を作成することができる。

4.7.3.2 Why ディメンションのユーザ・ボキャブラリ要素

Why ディメンションのデータ要素は、ビジネス・トランザクション、ソース、デスティネーション、トランスフォーメーション識別子などのビジネス・オブジェクトを指し示すものもある。これらのデータ要素には、適切な識別子を構成するためのテンプレートが、コア・ビジネス・ボキャブラリに用意されている。

ここで主眼になるのは、EPCISイベントのほかのディメンションも含めて、あいまいさのない識別子にすることである。EPCISイベントが、サプライチェーン全体から集まる場合には特に大切である。例えば、出荷ステップのEPCISイベントの *BusinessTransaction* データ要素に、発注書の参照を入れる場合、“PO# 1234”とするだけでは不十分である。サプライチェーンの中には、同じ番号の発注書を発行してもおかしくない会社があつてもある。EPCISイベントの中の発注書の参照は、全世界でユニークでなければならない。

コア・ビジネス・ボキャブラリは、その解決策として、全世界でユニークな識別子を表記する規則を用意している。この場合、下記のようになる：

```
urn:epcglobal:cbv:bt:06141411111114:1234
```

最初の部分(urn:epcglobal:cbv:)が、CBVに準拠して構成された識別子であることを表し、次の部分(bt:)は、ビジネス・トランザクション(BT)の表記規則を使っていることを表す。その次が、発注書番号を割り振った当事者のGLN(06141411111114)であり、最後がその当事者の割り振った発注書番号(1234)である。このようにすると、サプライチェーンの中で、ほかの会社が番号1234の発注書を発行していても、異なるGLNが1234の前についているため、EPCIS識別子は重複しない。

大きな企業では、発注書を発行するシステムが1つとは限らず、部門によって別のシステムが使われていることがある。すると番号1234の発注書が、社内に2つできる可能性がある。しかし、これは部門レベルのGLNを使うなどで、2つのシステムの発注書番号に違ったGLNを付加すれば、簡単に解決できる。

以上は、世界中でユニークなビジネス・トランザクション識別子を作り出すために、CBVで規定しているいくつかの方法の1つである(8.5節)。別のやり方として、GDTI(シリアル番号を持つもの)などのGS1識別コードを使うこともできる。これは、ビジネス・トランザクションを生成するシステムが、もとより番号割振りにGS1識別コードを使っている場合である。ほかにCBVには、プライベートなプリフィックスを使って、ビジネス・トランザクション識別子を作成する方法も示してあるが、それらはあまり使われない。

トランスフォーメーション・イベントは、高度な使い方として、“Transformation ID”によって複数のイベントを結び付けることがある。CBVの8.7節に、トランスフォーメーションIDを構成するいくつかの方法が記載してあり、上記のようなGLNを使ったやり方もある。

ソース/デスティネーション識別子は、CBVの8.6節に記載してある。通常はGLNを使い、場所の識別子とまったく同じかたちで、SGLNをEPC URI形式で用いる(4.7.2節)。

4.7.4 例

この節で取り上げた事柄をまとめると、4.6節で行った設計選択は、EPCISイベントの識別子として、最終的に次のようなかたちに実現される。

表4-6 EPCISイベント(節4.6)の識別子の割振り例

ディメンション	データ要素	設計選択(4.6節)	EPCIS イベントの実際の内容
	Event Type	Object Event	
	Action	OBSERVE	OBSERVE
What	EPC List	1要素のリスト: パレットのSSCC (インスタンス・レベル識別)	urn:epc:id:sscc:06141411.0123456789
When	Event Time	パレット出荷の日時	2014-03-15T10:11:12Z
	Event Time Zone Offset	パレット出荷場所の標準時からのオフセット	-05:00
Where	Read Point	10号建屋、出荷ドック#2	urn:epc:id:sgln:06141411.11111.2
	Business Location	(なし)	(なし)

ディメンション	データ要素	設計選択(4.6 節)	EPCIS イベントの実際の内容
Why	Business Step	Shipping (CBV より)	urn:epcglobal:cbv:bizstep:shipping
	Disposition	In Transit (CBV より)	urn:epcglobal:cbv:disp:in_transit
	Business Transaction List	2 つのビジネス・トランザクションへの参照が入ったリスト: 小売業者の発注書、製造元の請求書	タイプ urn:epcglobal:cbv:btt:po urn:epcglobal:cbv:bt:5012345678900:1234 タイプ urn:epcglobal:cbv:btt:inv urn:epcglobal:cbv:bt:06141411111114:9876
	Source List	タイプ“owning party”のソースが 1 つ入ったリストで、ソース側の所有者として製造元を示す。	タイプ urn:epcglobal:cbv:sdt:owning_party urn:epc:id:sgln:0614141.11111.0
	Destination List	タイプ“owning party”のデスティネーションが 1 つ入ったリストで、デスティネーション側に予定する所有者として小売業者を示す。	タイプ urn:epcglobal:cbv:sdt:owning_party urn:epc:id:sgln:5012345.67890.0

4.8 ステップ8:可視化データ・マトリクスへの可視化イベントの整理

ステップ5からステップ7までの検討を、ステップ4で拾い出したビジネス・ステップの1つ1つに遂行する。大変な作業に聞こえるが、普通は繰返しが多いため、最初の3つか4つのイベントができると、後は簡単に進むようになる。

それが完了したら、結果のデータ・モデルを、各列に可視化イベントを並べて、各行にデータ要素を列挙したマトリクスの形式にまとめる。これは、前節の表の欄方向に、各イベントを付け加えていったかたちになる。マトリクスの作成には、スプレッドシートがツールとして役に立つ。

これまでの例の、イベントV1からV4までをマトリクスにすると下のようになる:

表4-7 可視化データ・マトリクスの例

ディメンション	データ要素	V1	V2	V3	V4
	説明	梱装箱ラベルの印刷と貼付	パレット・ラベルの印刷と貼付	梱装箱のパレット積み	パレットの出荷
	Event Type	Object Event	Object Event	Aggregation Event	Object Event
	Action	ADD	ADD	ADD	OBSERVE
What	EPC List	梱装箱の SGTIN	パレットの SSCC	親: パレットの SSCC 子: 梱装箱の SGTIN	パレットの SSCC
When	Event Time	現在日時	現在日時	現在日時	現在日時
	Event Time Zone Offset	現地のタイムゾーン	現地のタイムゾーン	現地のタイムゾーン	現地のタイムゾーン
Where	Read Point	梱包ラインの SGLN	梱包ラインの SGLN	梱包ラインの SGLN	出荷ドックの SGLN
	Business Location	工場の GLN	工場の GLN	工場の GLN	(なし)
Why	Business Step	Commissioning (CBV)	Commissioning (CBV)	Packing (CBV)	Shipping (CBV)
	Disposition	Active (CBV)	Active (CBV)	In Progress (CBV)	In Transit (CBV)
	Business Transaction List	(なし)	(なし)	(なし)	小売業者の GLN+発注書# 製造元の GLN+請求書#
	Source List	(なし)	(なし)	(なし)	Owning Party (CBV): 製造元の GLN
	Destination List	(なし)	(なし)	(なし)	Owning Party (CBV): 小売業者の GLN

このマトリクス例では、イベントの内容を、ステップ6のように言葉で表したが、ステップ7で選んだ、識別子の具体例を入れてもよい(ここではスペースの関係で省略する)。

次の章では、特定の状況におけるEPCISイベントの設計例を、さらに挙げていく。

5 高度なEPCISのモデリング

この章では、さらに高度なビジネス・プロセスを取り上げて、EPCISイベントを使ってモデリングする方法を示す。

5.1 アグリゲーション/ディスアグリゲーション

多くのビジネス・プロセスは、子オブジェクトを親オブジェクトの中に梱包したり、親のオブジェクトに載せたりする、物理的なアグリゲーションの作成を伴う。アグリゲーションには、次のような特徴がある：

- アグリゲーションの状態にある間、親のオブジェクトと子のオブジェクトは、同じ時点に同じ場所にあると想定できる。
- 親のオブジェクトと子のオブジェクトは、アグリゲーションの状態にある間も、元の識別を保っている。アグリゲーションは、紐付けを解除して(ディスアグリゲーションして)、親と子のオブジェクトを元のように分離することができる。このことは、**トランスフォーメーション**と対照的で、トランスフォーメーションでは、インプット(材料オブジェクト)が、違った特性を持ったアウトプット(生成オブジェクト)に、不可逆的に変化する(5.5節を参照)。

よく見られるアグリゲーションの例として、次のようなものがある：

表5-1 よく見られるアグリゲーションの例

内 容	親のオブジェクトと識別子	子のオブジェクトと識別子
単品種の商品の梱装箱詰め	梱装箱(SGTIN)	商品(SGTIN)
多種の商品の梱装箱詰め(混載)	梱装箱(SSCC)	商品(SGTIN)
梱装箱のパレット積み	パレット(SSCC)	梱装箱(SGTIN または SSCC)
パレットの輸送コンテナ積み	コンテナ(GRAI)	パレット(SSCC)
コンテナの船舶や貨車その他への積み込み	船舶(GIAI)	コンテナ(GRAI)
コンポーネントのシャーシへの取付け	シャーシ(GIAI)	コンポーネント(GIAI または CPID)

上の例は、子のオブジェクトにすべてインスタンス・レベル識別を使っているが、子にクラス・レベル識別を使うこともできる。しかし、親は常にインスタンス・レベル識別子で表す必要がある。

一般にアグリゲーションを記録するのは、ビジネス・アプリケーションが、1個のオブジェクトの観察からアグリゲーション全体のオブジェクトの存在を知る、*推定* ができるようにするのが目的である。例えば4章の例では、出荷ステップのEPCISイベントにはパレットのSSCCしか入っていないが、受け手は、出荷された梱装箱に入っているケースすべてを推定することができる。ここで受け手の推定の土台になるのは、(a)アグリゲーションを作成する、梱包ステップのEPCISイベントが起こっていること、そして(b)梱包ステップと出荷ステップのイベントの間に、ディスアグリゲーションのイベントが起こっていないと分かっていることである。

5.1.1 アグリゲーションとディスアグリゲーション

EPCISのアグリゲーション・イベントのAction データ要素は、そのイベントの間に、アグリゲーションに何が起こったかを伝える：

表5-2 アグリゲーション・イベントのアクション値

アクション	意 味
ADD	子オブジェクトが親オブジェクトにアグリゲーションされた。このイベントが終わった後には、子は親と(親は子と)、物理的にアグリゲーションされると想定できる。
OBSERVE	親と子が、アグリゲーションの状態を観察されたが、イベントの中で子の追加や切離しは起こっていない。 Action 値 OBSERVE においては、親を省略してよい。子がアグリゲーションの状態にあることが観察されたが、親の識別子は、イベントの中では確認できていないことを表す。
DELETE	子が親から、ディスアグリゲーションされた。イベントの後、子は親と(親は子と)、物理的に分かれていると想定できる。 Action 値 DELETE においては、子を省略してよい。全部の子が、親からディスアグリゲーションされたことを表す。

例として、次の5つのステップからなるビジネス・プロセスを説明する：

1. 出荷元が、単品種の商品を詰めた梱包箱5個(各々をSGTINで識別)を、1個のパレット(SSCCで識別)に積み付ける。
2. 出荷元が、パレット識別子だけを記録してパレットを出荷する。
3. 受け手が、パレットを入荷するとともに、全部の梱包箱の識別子を確認する。
4. 受け手が、パレットから梱包箱2個を取り出す。
5. 受け手が、パレットから残りの梱包箱をすべて取り出す。

これらのステップに対応する5つのEPCISイベントの内容は、次の表ようになる(WhenとWhereのディメンションは省いて簡略化している)：

表5-3 EPCISアグリゲーション・イベントの情報内容の例

ディメンション	データ要素	V1	V2	V3	V4	V5
	説明	梱包箱のパレット積み	パレットの出荷	パレットの入荷	梱包箱2個の取出し	残りの梱包箱の取出し
	Event Type	Aggregation Event	Object Event	Aggregation Event	Aggregation Event	Aggregation Event
	Action	ADD	OBSERVE	OBSERVE	DELETE	DELETE
What	EPC List	親: パレットのSSCC 子: 5個の梱包箱のSGTIN	パレットのSSCC	親: パレットのSSCC 子: 5個の梱包箱のSGTIN	親: パレットのSSCC 子: 2個の梱包箱のSGTIN	親: パレットのSSCC 子: (省略)
Why	Business Step	Packing (CBV)	Shipping (CBV)	Receiving (CBV)	Unpacking (CBV)	Unpacking (CBV)
	Disposition	In Progress (CBV)	In Transit (CBV)	In Progress (CBV)	In Progress (CBV)	In Progress (CBV)

5.1.2 多階層のアグリゲーション

ビジネス・プロセスには、商品を梱包箱に詰めて、その梱包箱をパレットに積むといった、多階層のアグリゲーションを伴うものがある。このとき内側のアグリゲーションの親オブジェクトは、外側のアグリゲーションの子オブジェクトになる。

これは、各階層の親ごとにアグリゲーション・イベント1つを使って、複数のアグリゲーション・イベントにすれば、EPCISで難なくモデリングできる。例えば、5個の商品を梱包箱に詰めて、そのような梱包箱3つを1つのパレットに積む場合(合計15個)、全部で4つのアグリゲーション・イベントになる: 商品を梱包箱に詰め込むイベントが3つと、梱包箱をパレット積みするイベントが1つ。商品を詰めた混載のない梱包箱をSGTINで表し、パレットをSSCCで表すとすると、それぞれ下の表ようになる(WhenとWhereのディメンションは省いて簡略化している)：

表5-4 2階層になったEPCISアグリゲーション・イベントの情報内容の例

ディメンション	データ要素	V1	V2	V3	V4
	説明	商品1-5を梱包箱101に詰め込む	商品6-10を梱包箱102に詰め込む	商品11-15を梱包箱103に詰め込む	梱包箱101、102、103をパレット1001に積む
	Event Type	Aggregation Event	Aggregation Event	Aggregation Event	Aggregation Event
	Action	ADD	ADD	ADD	ADD
What	EPC List	親: 梱包箱101のSGTIN 子: 商品1-5のSGTIN	親: 梱包箱102のSGTIN 子: 商品6-10のSGTIN	親: 梱包箱103のSGTIN 子: 商品11-15のSGTIN	親: パレット1001のSSCC 子: 梱包箱101-103のSGTIN
Why	Business Step	Packing (CBV)	Packing (CBV)	Packing (CBV)	Packing (CBV)
	Disposition	In Progress (CBV)	In Progress (CBV)	In Progress (CBV)	In Progress (CBV)

5.2 メーカー直送 (Drop Shipment)

EPCISイベントのSourceとDestinationのデータ要素は、ビジネス移転(オブジェクトの所有者や預かり当事者の移転)の一部であるプロセス・ステップに関する詳しい情報を与える。EPCISイベントのSourceとDestinationは、それぞれがタイプを持ち、CBVで、これに用いる次の3つのタイプが定義されている：

表5-5 コア・ビジネス・ボキャブラリで定義されているソース/デスティネーション・タイプ

CBVのソース/ デスティネーション・タイプ	意味
owning_party	ソース/デスティネーションは、ビジネス移転の結果として、オブジェクトの所有権の引き渡し側(ソース)/受け取り側(デスティネーション)となる当事者の識別子
possessing_party	ソース/デスティネーションは、ビジネス移転の結果として、オブジェクトの物理的な所持の引き渡し側(ソース)/受け取り側(デスティネーション)となる当事者の識別子
location	ソース/デスティネーションは、オブジェクトが移動する際に始点(ソース)/終点(デスティネーション)となる物理的な場所の識別子 出荷のビジネス・ステップの、タイプ location のソースは、そのイベントのリード・ポイントと一貫するはずであり、同じように、入荷のビジネス・ステップの、タイプ location のデスティネーションは、そのイベントのリード・ポイントと一貫するはずである。

簡単なビジネス移転では、所有者と預かり当事者が、ソースでもデスティネーションでも一致し、ロケーションも、それぞれの当事者の施設になる。しかし、もう少し複雑な処理が表現されることもある。

ここでは、“メーカー直送”(drop shipment)の処理を考えてみる。例えば、医薬品の製造業者Mが、製品を卸売業者Wに販売し、卸売業者Wがさらに病院Hに製品を販売すると仮定する。メーカー直送の場合、卸売業者は、製品を実際には倉庫に納めずに、Mから直接Hに出荷させるように手配する。それでも製品の所有権は、Hとの間で販売トランザクションが起こるまでは、卸売業者が保有する。

EPCISで、これがどう表されるかを、次の2つのイベントで示す(When ディメンションは省いて簡略化している):

表5-6 “メーカー直送”(Drop Shipment)シナリオ例のEPCISイベントの情報内容

ディメンション	データ要素	V1	V2
	説明	製造業者 M が病院 H に製品を直送	直送品が病院 H に到着
	Event Type	Object Event	Object Event
	Action	OBSERVE	OBSERVE
What	EPC List	物流単位の SSCC	物流単位の SSCC
Where	Read Point	M の物流センターの GLN	H の入荷場所の GLN
	Business Location	(なし)	H の施設の GLN
Why	Business Step	Shipping (CBV)	Arriving (CBV)
	Disposition	In Transit (CBV)	In Progress (CBV)
	Source	タイプ owning_party (CBV) M の GLN	タイプ owning_party (CBV) M の GLN
	Source	タイプ possessing_party (CBV) M の GLN	タイプ possessing_party (CBV) M の GLN
	Source	タイプ location (CBV) M の物流センターの GLN	タイプ location (CBV) M の物流センターの GLN
	Destination	タイプ owning_party (CBV) W の GLN	タイプ owning_party (CBV) W の GLN
	Destination	タイプ possessing_party (CBV) H の GLN	タイプ possessing_party (CBV) H の GLN
	Destination	タイプ location (CBV) H の入荷場所の GLN	タイプ location (CBV) H の入荷場所の GLN

5.3 クラス・レベルのトレース

4.6.1 節で述べたように、EPCISのオブジェクトは、インスタンス・レベルとクラス・レベルで識別できる。このガイドラインの中の例の大半は、前節までに用いた例を含めて、インスタンス・レベル識別しか使っていない。この節では、クラス・レベル識別を使うときの留意事項を説明する。

5.3.1 クラス・レベル識別を用いたトレーサビリティにおける固有の制約

クラス・レベル識別は、インスタンス・レベル識別とは異なり固有の制約がある。インスタンス・レベル識別では、特定のオブジェクトを参照しているEPCISイベントを正確に調べることができる。したがって、時刻の前後する2つのEPCISイベントが、同じオブジェクトを指しているかどうかを正確に判定できる。それに対してクラス・レベル識別は、オブジェクトのクラスを指すため、クラスの中のオブジェクト1つ1つは区別できない。クラス・レベル識別でトレースを行うシステムは、このあいまいさを考慮したうえで設計しなければならない。

例として、クラス・レベル識別を使っている、次のような一連のイベントを考える：

- V1: 製造元が、20個の新しい製品を製造(どれもGTIN+ロット番号で識別するだけ)
- V2: 製造元が、10個の製品を、受け手に向けて出荷
- V3: 製造元が、さらに10個の製品を、同じ受け手に向けて出荷
- V4: 受け手が、10個の製品を入荷

これらのEPCISイベントの内容は、下の表のようになる：

表5-7 クラス・レベル識別を使ったEPCISイベントの情報内容の例

ディメンション	データ要素	V1	V2	V3	V4
	説明	20個の新しい製品の製造	10個の製品の出荷	さらに10個の製品の出荷	10個の製品の入荷
	Event Type	Object Event	Object Event	Object Event	Object Event
	Action	ADD	OBSERVE	OBSERVE	OBSERVE
When	Event Time	15 July, 10am	16 July, 10am	17 July, 10am	25 July, 10am
What	EPC Quantity List	GTIN X, ロット 12, 20 個	GTIN X, ロット 12, 10 個	GTIN X, ロット 12, 10 個	GTIN X, ロット 12, 10 個
Where	Read Point	製造ラインの SGLN	製造元の出荷ドックの SGLN	製造元の出荷ドックの SGLN	受け手の入荷ドックの SGLN
	Business Location	製造元の GLN	(なし)	(なし)	受け手の GLN
Why	Business Step	Creating Class Instance (CBV)	Shipping (CBV)	Shipping (CBV)	Receiving (CBV)
	Disposition	Active (CBV)	In Transit (CBV)	In Transit (CBV)	In Progress (CBV)

この例では、イベントV4で7月25日に入荷されたGTIN X、ロット12の10個が、7月16日に出荷された10個(イベントV2)か、7月17日に出荷された10個(イベントV3)のどちらかを知ることは不可能である。これはEPCISの制約ではなく、クラス・レベル識別を使うことで生じた基本的な制約である。

このように、クラス・レベルのデータを使う場合には、ごく普通のトラックとトレースの処理が複雑になる。例えば、製品リコールが発生した際には、特定のロットの製品が現在ある場所をすべて調べ上げて、サプライチェーンから取り除くことがアプリケーションの目標になる。インスタンス・レベル識別を使っていれば、ロットの1つ1つの製品が、コミッションのビジネス・ステップから分かるユニークなシリアル番号を持つため、リコールを処理するアプリケーションは、各インスタンスの識別子の最新のEPCISイベントさえ見つければ、そのイベントのビジネス・ロケーションによって、少なくともEPCISデータ要素から推定できる範囲で現在の場所が分かる。それぞれの製品識別子は、複数のEPCISに表れることがあっても、2カ所に同時に存在することはないため、最新のイベントが現在の位置になる。

ところが同じことを、同じロットの製品がサプライチェーンの中で異なる道のりを経て流れていく可能性がある状況において、ロット・レベルの識別で行おうとすると問題が生じる。ロットの最新のEPCISイベントを見つけただけでは、製品全部の場所は分からない。上の例では、ロット12の最新のEPCISイベントはイベントV4になるが、全20個のうち10個しか把握できていない。残りの10個は輸送途上にあり、イベントV2またはV3が対応している。ロットが現在存在する場所を全部拾い出すためには、各施設を出入りする数量を数え上げる、遠回りな分析処理が必要になる。

クラス・レベル識別を用いたアプリケーションを使う際は、データをどのように使うか、及び性質上どのような制約が生じるかを、よく考えておかなければならない。

5.3.2 クラス・レベル識別で始まりを示すイベント

インスタンス・レベルで識別を行うときは、どの製品も、その製品の識別子が生み出される「始まりを示すイベント」は、た

だ1つしか存在しない。始まりを示すイベントは、アクションADDのオブジェクト・イベントか、トランスフォーメーション・イベント(この場合インスタンス識別子はアウトプット)のどちらかであり、ビジネス・ステップは、CBVで定義されているCommissioningか、あるいは別のポキャブラリで定義された特定分野のビジネス・ステップで、Commissioningと同様の意味を持つものになる。

クラス・レベル識別の場合は、そのクラスの製品の量が増えるごとに生成を表すイベントが発生するため、始まりを示すイベントが複数になることがある。例えば、製造工程で、毎時1パレット分の製品が製造され、でき上がったパレットごとにEPCISイベントが生成され、1日に製造されたパレット全体が1ロットになる場合などが考えられる。毎時のEPCISイベントは、その1時間に製造された製品の始まりを示すものになる。

CBVでは、クラス・レベル識別子のCommissioningは、*それまで使われたことのない* 識別子を、そのクラスの1つ以上のオブジェクトに関係付けるプロセスを表すために定義されている。言い換えると、Commissioningは、オブジェクトの始まりだけでなく、識別子の始まりも表すものである。ビジネス・ステップCommissioningのEPCISイベントは、1つの識別子に1つしか存在できない。

同じクラスの中で複数の始まりを示すイベントを扱う場合に対しては、CBVでは、別にCreating Class Instanceというビジネス・ステップの種類を定義している。Creating Class Instanceは、Commissioningとは異なり、オブジェクトの生成だけを意味し、識別子の生成については示さない。

それぞれのクラス・レベル識別子が初めて使われる時点(例えば、新しい製品ロットの製造開始など)が、はっきり分かるビジネス・プロセスならば、新しいロットの製品を生成する最初のEPCISイベントにビジネス・ステップCommissioningを、以降同じロットの製品を追加生成するイベントに、Creating Class Instanceを使うことができる。それぞれのクラス・レベル識別子が最初に使われるEPCISイベントが分からなかったり、調べることが現実的でなかったりする場合もあるが、その場合は、同じクラスの製品を生成する全部のイベントにCreating Class Instanceを使ってもよい。

5.3.3 アグリゲーションにおけるクラス・レベル識別

アグリゲーション・イベントは、子オブジェクトにクラス・レベル識別を使ったものがあってもよい。しかし親オブジェクトには、常にインスタンス・レベル識別を使う。

例として、それぞれ単品種の製品を詰めた梱包箱が、注文に応じてピッキングされ、出荷され、そして入荷されるプロセスを考えてみる。梱包箱は、GTIN+ロットだけを用いて識別する。イベントは、下のようになる(When とWhere のディメンションは省いて簡略化している):

表5-8 クラス・レベルで識別される子オブジェクトのアグリゲーションに関するEPCISイベントの情報内容

ディメンション	データ要素	V1	V2	V3
	説明	梱包箱のパレット積み	パレットの出荷	パレットの入荷
	Event Type	Aggregation Event	Object Event	Object Event
	Action	ADD	OBSERVE	OBSERVE
What	EPC List	親: パレットの SSCC 子: GTIN X、ロット 12、10 個 GTIN Y、ロット 52、20 個	パレットの SSCC	パレットの SSCC
Why	Business Step	Packing (CBV)	Shipping (CBV)	Receiving (CBV)
	Disposition	In Progress (CBV)	In Transit (CBV)	In Progress (CBV)

この例の場合、受け手は、前のアグリゲーション・イベントを使って、受け取ったパレットの内容物が、GTIN X(ロット12)が10個とGTIN Y(ロット52)が20個であると推定できる。その後、パレットから製品を取り出す際のイベントでも、ディスプレイアグリゲーションする個数を、該当のクラスの数量で示す。

アグリゲーションの親オブジェクトを、クラス・レベル識別子を使って表すことはできない。各アグリゲーションを、親の識別子で代表するなど個別に識別できなければ、内容の推定ができず、推定ができなければ、アグリゲーションを記録する価値がないためである。

説明のために、次のような例を考える。同じGTIN+ロット識別子を持った2つの梱包箱があって、それぞれに違った内容物が詰め込まれた後、その1つが出荷されて、入荷されたとする:

表5-9 クラス・レベルの親を持った仮想的なEPCISアグリゲーション・イベントの情報内容 (規則違反)

ディメンション	データ要素	V1	V2	V3	V4
	説明	1つ目の梱包箱の詰込み	2つ目の梱包箱の詰込み	1つ目の梱包箱の出荷	梱包箱の入荷
	Event Type	Aggregation Event	Aggregation Event	Object Event	Object Event
	Action	ADD	ADD	OBSERVE	OBSERVE
What	EPC List	親(誤り!): GTIN P、ロット 111、1 個 子: GTIN X、ロット 12、10 個 GTIN Y、ロット 52、20 個	親(誤り!): GTIN P、ロット 111、1 個 子: GTIN X、ロット 12、20 個 GTIN Z、ロット 34、30 個	GTIN P、ロット 111、1 個	GTIN P、ロット 111、1 個
Why	Business Step	Packing (CBV)	Packing (CBV)	Shipping (CBV)	Receiving (CBV)
	Disposition	In Progress (CBV)	In Progress (CBV)	In Transit (CBV)	In Progress (CBV)

この場合の受け手は、GTIN P、ロット111で識別される梱包箱1個を受け取ったことが分かるが、梱包箱の内容物が、GTIN Xが10個とGTIN Yが20個か、GTIN Xが20個とGTIN Zが30個のどちらかを推定する方法はない。

子オブジェクトと親オブジェクトをともに、クラス・レベルだけで識別する状況として考えられるのは、GTINまたはGTIN+ロットで識別する製品を、同じくGTINまたはGTIN+ロットで識別する、内容構成が固定された梱包箱に詰める場合である。例えば、必ず12個の同じ製品(それぞれ製品レベルのGTIN Yを持つ)を、標準の箱(梱包箱レベルのGTIN Xを持つ)に詰めるときなどである。この状況では、GTIN Xを、特にGTIN Y用12個詰め箱を表すものとして割り振る。製品の個数を含めた2つのGTINの間の関係は、マスタ・データで表す。

このようにする場合は、GTIN Xのどの梱包箱も、定義によって12個のGTIN Yを取めるため、アグリゲーション・イベントを加えて、12個のGTIN Yを1個のGTIN Xに詰め込んだことを示す必要はない。したがってEPCISでは、GTIN XとGTIN Yのコミッションのビジネス・ステップがあっても、アグリゲーション・イベントはない。GTIN Xで表す梱包箱を10個出荷する際、出荷のEPCISイベントは、10 個のGTIN Xと表しても(この場合、GTIN Yが120個であることはマスタ・データから推定)、120個のGTIN Yと表してもよい(この場合EPCISデータは、製品がバラで出荷されたのか、12個詰め箱に入れて出荷されたのかは示さない)。

この例は、製品と梱包箱をGTINだけで識別することを考えた。製品も梱包箱もGTIN+ロットで識別するならば、各梱包箱に入った製品は、全部ロット番号が同じでなければならない、そして同じロット番号を梱包箱にも使う。これは実際にも、製品と梱包箱をともにGTIN+ロットで表す場合に、一般的に取られているやり方である。

(補足すると、最初の“誤った”例は、いずれにしてもGTINの管理基準のうえで許されない。内容構成が固定されていない梱包箱は、商品ではなく物流単位と見なされるため、SSCC(インスタンス・レベル識別子)で識別し、GTINを使えない。)

5.3.4 インスタンス・レベルとクラス・レベルの識別を同時に持つイベント

1つのEPCISイベントに、インスタンス・レベルとクラス・レベルの識別を、混ぜ合わせて入れることもできる。例えば、伝票に従ってパレット積みした荷に、SGTINで識別した製品が1個と、GTIN+ロットで識別した別の製品が1個、そしてGTINだけで識別したもう1個の製品が入っていてもよい。下に例を示す(When とWhere のディメンションは省いて簡略化している):

表5-10 インスタンス・レベルとクラス・レベルで識別した子を同時に持つEPCISアグリゲーション・イベントの情報内容

ディメンション	データ要素	V1	V2	V3
	説明	梱包箱のパレット積み	パレットの出荷	パレットの入荷
	Event Type	Aggregation Event	Object Event	Object Event
	Action	ADD	OBSERVE	OBSERVE
What	EPC List	親: パレットの SSCC 子: GTIN X、シリアル番号 101 GTIN X、シリアル番号 102 GTIN X、シリアル番号 103 GTIN Y、ロット 12、10 個 GTIN Z、20 個	パレットの SSCC	パレットの SSCC

ディメンション	データ要素	V1	V2	V3
Why	Business Step	Packing (CBV)	Shipping (CBV)	Receiving (CBV)
	Disposition	In Progress (CBV)	In Transit (CBV)	In Progress (CBV)

前に挙げた例と同じように、受け手は、アグリゲーション・イベントによって、パレットの内容を推定することができる。この場合には、インスタンス・レベルとクラス・レベルの識別両方を用いる。

インスタンス・レベルとクラス・レベルの識別を一緒にした使い方は、特に製造現場のトランスフォーメーション・イベントによく見られる。製造工程の原材料に、インスタンス・レベルで識別される主原料と、クラス・レベルだけで識別される副原料を使う場合などがそうである。下に例を挙げる：

- インプット(材料オブジェクト):
 - ツナ・ロイン(ロインごとにシリアル番号を付けて、SGTINで識別 – インスタンス・レベル)
 - オリーブ油(GTIN+ロットで識別 – クラス・レベル)
 - 缶(GTINで識別、缶サプライヤの区別を可能にする目的)
- アウトプット(生成オブジェクト):
 - ツナ缶詰(各缶をインスタンス・レベル(SGTIN)で識別するか、クラス・レベル(GTIN+ロット)で識別するかは、業務上の要件による)

注意する点として、同じEPCISイベントにインスタンス・レベル識別子とクラス・レベル識別子が混じるときは、各識別子が違うオブジェクトを指すと理解される。したがって、下の例は正しくない：

表5-11 インスタンス・レベルとクラス・レベルの識別子が同じオブジェクトを指している誤ったEPCISイベント

ディメンション	データ要素	V1
	説明	梱包箱のパレット積み
	Event Type	Aggregation Event
	Action	ADD
What	EPC List	親: パレットの SSCC 子(誤り!): GTIN X、シリアル番号 101 GTIN X、シリアル番号 102 GTIN X、シリアル番号 103 GTIN X、ロット 12、3 個
Why	Business Step	Packing (CBV)
	Disposition	In Progress (CBV)

前記のように、このイベントは、シリアル番号を持ったもの3個と、ロット番号しか持たないもの3個を合わせた、6個のGTIN Xがパレットに積まれているという意味になる。これは、本来意図されていた表示ではないだろう。

GTIN+シリアル番号で識別する商品に対して、ロット番号を付け加えたい場合は、イベントにはSGTINだけを入れて、ロット番号は、そのシリアル番号のコミッションのイベントの際に、インスタンス/ロット・マスター・データとして表示する(5.4 節を参照)。

5.4 インスタンス/ロット・マスター・データ(ILMD)

EPCIS 1.1標準仕様7.3.6節で説明されているように、インスタンス/ロット・マスター・データ(ILMD)は、物理/デジタル・オブジェクトの特定のインスタンス、またはバッチ/ロット生産されるオブジェクトの特定のバッチ/ロットを説明するデータである。普通のマスター・データと同じように、オブジェクトについての情報を与える、記述的な属性情報の集まりで構成される。しかし、マスター・データの属性が、大きな範囲のオブジェクト(例えば、あるGTINを持つオブジェクト全体)で同じ値を持つものに対して、ILMDの属性の値は、ずっと小さなオブジェクトのグループ(例えば、1つのバッチまたはロット)ごとに変わってよく、オブジェクトごとに違ってよい(インスタンスごとに異なる)。

GTINレベルのマスター・データは、GDSN(グローバルデータ同期化ネットワーク)を使って、EPCISの範囲の外側で取引相手とやり取りできる。インスタンス/ロット・レベルのマスター・データも、EPCISの外側でやり取りすることは不可能ではないが、現状、十分に確立されたインスタンス/ロット・マスター・データのやり取りの仕組みはない。そのためEPCISでは、新しいイン

スタンスの始まりを示すEPCISイベントに、インスタンス/ロット・マスタ・データを添付する手段を用意している。

インスタンス・レベルで識別するオブジェクトのマスタ・データは、そのインスタンスのコミッションのイベントに持たせ、あるいはトランスフォーメーションのアウトプットとして生成されるインスタンスならば、そのトランスフォーメーションのイベントに持たせる。例として、下の表に3つのEPCISイベントの内容を示し、そのうちコミッションのステップにインスタンス・レベルのマスタ・データを入れている(“When”ディメンションは省いて簡略化している)：

表5-12 インスタンス/ロット・マスタ・データ(ILMO)の入ったEPCISイベントの情報内容

ディメンション	データ要素	V1	V2	V3
	説明	新しい製品の製造	製品の出荷	製品の入荷
	Event Type	Object Event	Object Event	Object Event
	Action	ADD	OBSERVE	OBSERVE
What	EPC List	製品の SGTIN	製品の SGTIN	製品の SGTIN
Where	Read Point	製造ラインの SGLN	製造元の出荷ドックの SGLN	受け手の入荷ドックの SGLN
	Business Location	製造元の GLN	(なし)	受け手の GLN
Why	Business Step	Commissioning (CBV)	Shipping (CBV)	Receiving (CBV)
	Disposition	Active (CBV)	In Transit (CBV)	In Progress (CBV)
	ILMD: Expiry	製品の有効期限		
	ILMD: Lot	製品のロット番号		

これに見られるように、オブジェクトをインスタンス・レベルで識別する場合には、ロット番号(存在する場合)が、マスタ・データの属性になる。

受け手は、入荷した製品のマスタ・データの取得を希望する場合、What ディメンションにそのSGTINを持ち、ビジネス・ステップがCommissioning(CBVより)のイベントを、製造元に問い合わせる。

EPCISイベントのXML表現では、ILMDは、EPCISの名前空間とは別に、XML名前空間で定義した要素を使って表す。CBV標準は、XML名前空間(urn:epcglobal:cbv:mda)を使って、もっともよく使用されるマスタ・データの属性を定義する。そうしたマスタ・データの属性は、GDSNやGS1 EDIといったその他のGS1標準で使用される定義と一致する定義を持つ。その他のマスタ・データの属性は、取引相手間の事前の合意など、他の標準で定義しても構わないが、その場合、属性はEPCISやCBVの名前空間以外のXML名前空間を持つ必要がある。

上記のイベントV1は、XMLでは下のようになる(簡略化のために一部省略):

```
<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:
  1"
  xmlns:cbvmda="urn:epcglobal:cbv:mda">
  <EPCISBody>
  <EventList>
    <ObjectEvent>
      ...
      <epcList>
        <epc>urn:epc:id:sgtin:0614141.012345.400</epc>
      </epcList>
      <action>ADD</action>
      <bizStep>urn:epcglobal:cbv:bizstep:commissioning</bizStep>
      <disposition>urn:epcglobal:cbv:disp:active</disposition>
      ...
      <extension>
        <ilmd>
          <cbvmda:itemExpirationDate>2015-03-15</example:expiry>
          <cbvmda:lot>A123</example:lot>
        </ilmd>
      </extension>
    </ObjectEvent>
  </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

ロット・レベルのマスター・データも、インスタンス・レベルのマスター・データと同じように使う。インスタンス・レベルの場合と違う点として、ロット・レベル識別の場合、同じロットに対して始まりを示すイベントがいくつも発生することがある(オブジェクト・イベントまたはトランスフォーメーション・イベントで、ビジネス・ステップがcommissioningまたはcreating-class-instanceのもの)。ロットのILMDは、そのロットに関係する全部のイベントで内容が同じである限り、始まりを示すイベントの全部に入れることができる。

また、commissioningとcreating-class-instanceの両方のビジネス・ステップがあるときには、commissioningのステップにだけILMDを入れることも認められる。

5.5 トランスフォーメーション

EPCISのトランスフォーメーション・イベントは、1つ以上のオブジェクトの全体または一部が、材料オブジェクトとして消費されて、1つ以上のオブジェクトが、生成オブジェクトとして作られる場合の、ビジネス・プロセスのステップを表現するために用いられる。トランスフォーメーション・イベントは、材料オブジェクトと生成オブジェクトの間の関係を記録し、各アウトプットに、いくつかのまたは全部のインプットが何らかのかたちで関与したことを表す。

アグリゲーションと違って、トランスフォーメーションは不可逆的である。トランスフォーメーションの後、消費されたインプットは存在しなくなる。そしてアウトプットは、トランスフォーメーションの前には存在しなかった新しいオブジェクトである。このように、トランスフォーメーションは、アウトプットにとってはライフサイクルの始まりのイベントとして、インプットにとってはライフサイクルの終わりのイベントとして働く(インプットの一部だけが消費された場合は除く)。

一般的に見られるトランスフォーメーションとして、例えば以下のようなものがある:

表5-13 トランスフォーメーションのビジネス・プロセスの例

説明	インプット・オブジェクトと識別子	アウトプット・オブジェクトと識別子
原料を混合して製品を製造	原材料(原料ごとに違った SGTIN、GTIN+ロット、または GTIN)	混合製品(包装パッケージ別の SGTIN、GTIN+ロット、または GTIN)
食肉のプライムカットを組み合わせ、小分けし、小売パックに包装	食肉プライムカット(SGTIN)、調味料その他の副材料(GTIN+ロット)、無菌包装材(GTIN)	小売精肉パック(SGTIN または GTIN+ロット)
医薬品のバルク製品を、小さな販売パッケージに再包装	バルク医薬品(SGTIN または GTIN+ロット)	医薬品小箱(SGTIN または GTIN+ロット)

通常の場合、トランスフォーメーションを追跡する理由は、ビジネス・プロセスの中でどの材料オブジェクトがどの生成オブジェクトに作用しているのかを分かるようにすることである。例えば、ある牧場が産地の食肉のプライムカットに、細菌汚染のあることが発見された場合、トランスフォーメーション・イベントによって前方向にトレースして、汚染されたプライムカットに関係している最終製品を、すべて見つけ出すことができる。逆に、最終製品に汚染が見つかった場合にも、トランスフォーメーション・イベントから逆方向にたどって、原材料をすべて洗い出し、そこから前方向にトレースして、ほかに汚染されているおそれのある最終製品を見つけ出すことができる。

5.5.1 トランスフォーメーション・イベントの例

次のような製造工程を考える：

- 材料オブジェクト：
 - ツナ・ロイン(ロインごとにシリアル番号を付けて、GTIN X+シリアル番号で識別 – インスタンス・レベル)
 - オリーブ油(GTIN Y+ロットで識別 – クラス・レベル)
 - 缶(GTIN Zで識別、缶サプライヤの区別を可能にする目的)
- 生成オブジェクト：
 - ツナ缶詰(GTIN Q+ロットで識別 – クラス・レベル)

この工程を、1回実施したときのトランスフォーメーション・イベントは、下のようになる(When とWhere のディメンションは省いて簡略化)：

表5-14 トランスフォーメーション・イベントの情報内容の例

ディメンション	データ要素	V1
	説明	原材料からツナ缶詰を製造
	Event Type	Transformation Event
What	EPC List	材料オブジェクト： GTIN X、シリアル番号 10 GTIN X、シリアル番号 45 GTIN X、シリアル番号 97 GTIN Y、ロット 12、10 リットル GTIN Z、100 個 生成オブジェクト： GTIN Q、ロット 999、100 個
Why	Business Step	Creating Class Instance (CBV)
	Disposition	Active (CBV)

トランスフォーメーションは、生成オブジェクトの始まりを示すため、始まりを示すビジネス・ステップとディスポジションを使う。ここでは、トランスフォーメーションからロット999の新しいインスタンスができるため、ビジネス・ステップにCreating Class Instanceを使っている。もし、このイベントによって、ロット999が初めて生成されることが分かっている場合は、代わりにCommissioningを使うこともできる。

5.5.2 長期にわたるトランスフォーメーション

トランスフォーメーションが長い時間にわたって継続し、その間定期的に材料オブジェクトが追加され、生成オブジェクトが取り出されていくような場合もある。例えば、混合工程で、材料オブジェクトが工程中に数回のバッチに分けて投入され、その一方で、並行して生成オブジェクトが取り出されていくなどの状況である。

このような長期にわたるトランスフォーメーションも、期間全体に広がって関連する材料オブジェクトと生成オブジェクトをすべて列挙すれば、1つのEPCISイベントとしてモデリングできる。イベント時刻をどう決めるかという問題が生じるが、ほとんどの場合、工程の終了時点イベント時刻にすることが望ましい。

しかし、長時間のトランスフォーメーションを1つのEPCISイベントでモデリングするのが、望ましくない場合もある。特に生成オブジェクトの一部が、全トランスフォーメーションが完了する前に次のビジネス・ステップに進む場合である。例として、原料を混合して、缶入りペンキを製造する設備を考える。混合工程は、同じ混合容器を使いながら、1週間にわたって継続するものとする。この設備は、月曜日に缶入りペンキを製造して、それを火曜日に出荷するが、水曜日と木曜日にも、同じ混合容器から缶入りペンキをさらに製造して、トランスフォーメーションが完了するのは金曜日となる。この場合、火曜日に発生する出荷イベントのために缶入りペンキの新しい識別子を用意するうえで、月曜日の製造を表すEPCISイベントが必要になる。

このような状況では、1つのトランスフォーメーションを、複数のEPCISイベントに分割してモデリングする。そして、全部の材料オブジェクトと生成オブジェクトの間に関係を維持するために、トランスフォーメーション識別子を使って、それらのトランスフォーメーション・イベントをリンクする。トランスフォーメーション識別子は、単純に同じトランスフォーメーションに属するEPCISイベントのすべて(つまり、材料オブジェクトと生成オブジェクトの間に関係のあるもの)で同じ値を取り、関係のないほかのイベントのものとは一致しない、ユニークな識別子である。

下の表は、前の節と同じ内容のトランスフォーメーションに、最初に製造されたいくつかのツナ缶を出荷する処理を付け加えた場合のイベントである。

表5-15 トランスフォーメーション識別子でリンクしたトランスフォーメーション・イベントの情報内容の例

ディメンション	データ要素	V1	V2	V3	V4
	Description	新しいバッチへの初回の原材料投入	初回の缶の完成	初回の缶の出荷	残りの原材料の投入と、製造の完了
	Event Type	Transformation Event	Transformation Event	Object Event	Transformation Event
What	Transformation ID	Xform 123	Xform 123		Xform 123
	EPC List	材料オブジェクト: GTIN X、シリアル番号 10 GTIN X、シリアル番号 45 GTIN Y、ロット 12、5リットル GTIN Z、40個 生成オブジェクト: (なし)	材料オブジェクト: (なし) 生成オブジェクト: GTIN Q、ロット 999、30個	GTIN Q、ロット 999、30個	材料オブジェクト: GTIN X、シリアル番号 97 GTIN Y、ロット 12、5リットル GTIN Z、60個 生成オブジェクト: GTIN Q、ロット 999、70個
Why	Business Step	Creating Class Instance (CBV)	Creating Class Instance (CBV)	Shipping (CBV)	Creating Class Instance (CBV)
	Disposition	Active (CBV)	Active (CBV)	In Transit (CBV)	Active (CBV)

コア・ビジネス・ボキャブラリが、トランスフォーメーション識別子の構成に用いるテンプレートを用意している。

5.6 クーポンと引換券

EPCISの用途は、物理オブジェクトの記録に限らず、例えばデジタル商品(ダウンロード販売の楽曲、電子ブックその他)や、デジタル・ドキュメント(電子クーポンその他)などの、デジタル・オブジェクトの追跡にも利用できる。デジタル・オブジェクトのビジネス・プロセスも、ほとんどの場合物理オブジェクトのプロセスと変わりなく、同じコア・ビジネス・ボキャブラリのビジネス・ステップとディスポジションを使用できる。

この節では、デジタル・クーポンのライフサイクルを記録する2つのプロセスを示して、デジタル・オブジェクトを対象にしたEPCISの利用について説明する。デジタル・クーポンとは、特定の商品を購入する消費者に向けて、製造業者または小売業者から、何かの特典(現金払戻、割引、景品)を提供するサービスである。製造業者や小売業者からの各クーポン・サービスの識別には、GCN(グローバルクーポン番号)が使える。クーポン・サービスのクーポンが消費者に発行されて、それを消費者が特典と交換することになるが、このクーポンの識別にはシリアル番号の入ったGCN(SGCN)が使える。条件となる購入商品のGTINや、払戻金額その他のサービスの詳細は、GCNと関連付けたマスタ・データを参照することになる。瓶の自動回収機から消費者に発行されて、消費者がPOS(Point-of-Sales)で換金するデジタル引換券なども、同じような方法で運用される。

ここで説明する2つのプロセスは、製造業者がクーポンを発行して、それを小売店がPOSで交換処理する簡単なプロセスと、クーポン仲介業者が介在する、もう少し複雑なものである。

どちらの例も、デジタル・オブジェクトにEPCISを使用する際の、全体的な考え方を説明することを目的としている。クーポン・プロセスの具体的なモデリング方法については、GS1のアプリケーション標準か、該当する地域の推奨事項を参照のこと。

5.6.1 簡単なクーポン・プロセス

ごく簡単なクーポン・プロセスでは、EPCISイベントが必要なステップは2つだけである：

- **V1:** 消費者が、クーポン発行元からデジタル・クーポンの発行を受ける。クーポンの発行元になるのは、小売業者が普通だが、製造業者やその他の第三者が発行することもある。消費者が携帯機器の上で利用する、アプリケーションを通してクーポンが発行されるかたちが多い。次のステップでの使用に備えて、クーポンのSGCNがそのアプリケーションに保存される。その際にEPCISイベントが発生して、クーポンがアクティブ状態になったことを示す。
- **V2:** 消費者が、小売店(店舗、オンラインストアとも)で支払いの際に、POS端末で引換えを受ける。POSアプリケーションは、正しいクーポンであることと、オファーの条件が満たされていることを検証する。問題がなければクーポンが引き換えられ、同時にEPCISイベントが生成されてクーポンがアクティブでなくなったことを示す。

この2つのイベントは、commissioningとdecommissioningのビジネス・ステップを使ってEPCISで表す。

表5-16 簡単なデジタル・クーポン・ビジネス・プロセスのEPCISイベントの情報内容の例

ディメンション	データ要素	V1	V2
	説明	デジタル・クーポンの発行	デジタル・クーポンの引換え
	Event Type	Object Event	Object Event
	Action	ADD	DELETE
When	Event Time	15 July, 10am	16 July, 10am
What	EPC	SGCN X	SGCN X
Where	Read Point	クーポン発行元の SGLN (普通は、該当する物理的な場所がなければ、当事者 GLN。あるいはキオスク端末などからのクーポン発行ならば、物理的な場所の SGLN)	小売店の POS 端末の SGLN (オンライン販売などで、物理的な場所がなければ、当事者 GLN)
	Business Location	(なし)	(なし)
Why	Business Step	Commissioning (CBV)	Decommissioning (CBV)
	Disposition	Active (CBV)	Inactive (CBV)
	ILMD	(下記を参照)	(なし)

コミッションのステップ(V1)では、ILMDを使って該当商品のGTIN、クーポン引換え期間、顧客ポイントカード番号その他の、クーポンの各種の属性を収録できる。

EPCISイベントが集まると、イベントのクエリによって、一定期間内に発行されたクーポンの総数、特定のGCN(クラス・レベル)のクーポンの総数、発行後に未引換のSGCNの一覧、引換えの数、クーポンの発行から引換えまでの時間など、その他いろいろなことを調べることができる。

5.6.2 クーポン仲介業者が介在するクーポンの例

前節の例は、消費者へのクーポン発行と、POSでのクーポン引換えの、2つのイベントだけでEPCISの追跡を行うものだった。しかし、ほかのビジネス・プロセスと同じように、クーポン引換えのプロセスも、実際にはもっと複雑な処理になることがあり、EPCISのモデリングに、もう少しステップを加えた方がよいこともある。下の例は、4つのステップをEPCISで記録する、より複雑なものである：

- **V1:** クーポン発行元(小売業者または製造業者)が、クーポン配布業者(電子クーポン処理を取扱うインターネット・アプリケーション・プロバイダなど)に、クーポンを一括発行する。
- **V2:** 消費者が、クーポン配布業者からデジタル・クーポンの発行を受ける。
- **V3:** 消費者が、小売店(店舗、オンラインストアとも)での支払いの際に、POS端末で引換えを受ける。
- **V4:** クーポン配布業者と発行元の間で最終の決済をする。

最初の例と同じように、上記のどのビジネス・ステップも、EPCISイベントでモデリングできる。今回の例の場合、commissioningは、V1でクーポンを配布業者に発行するときに起こり、配布業者が消費者にクーポンを発行するときではない(後者は、むしろ消費者の受取り段階にあたる)。そしてdecommissioningも、ステップV4で最終決済するときに起こり、消費者がクーポンを引き換えるときではない。

4つのイベントで使うビジネス・ステップとディスポジションには、コア・ビジネス・ボキャブラリにないものがあり、ここでは説明は省く。クーポン処理のための詳しい標準やガイドラインは、今後GS1で策定されていくかもしれない。

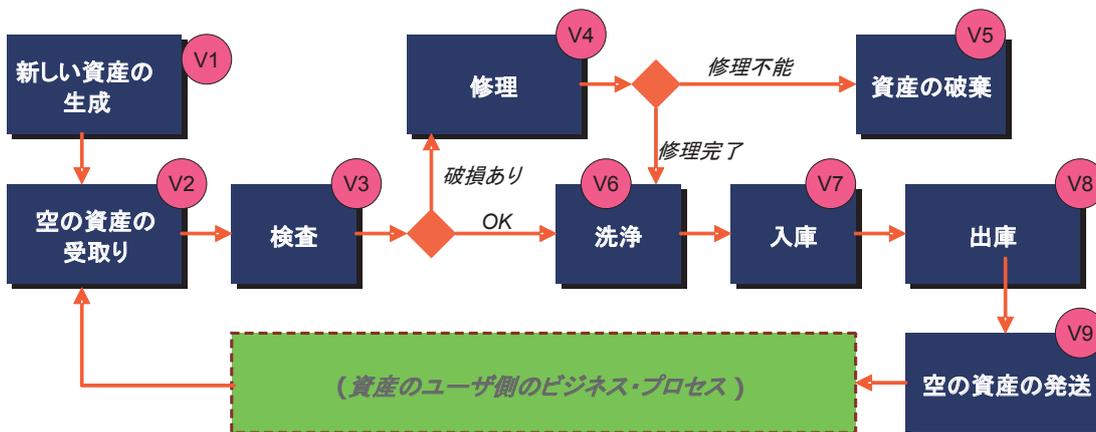
ここでのポイントは、“What”ディメンションが物理オブジェクトであろうとデジタル・オブジェクトであろうと、同じ分析と設計の手続きで、ビジネス・プロセスをEPCISイベントにモデリングできることである。

5.7 繰り返し使用する資産のGRAIによる管理

この節では、共同利用するパレットや運搬箱のプーリングなど、繰り返し使用する資産の追跡イベントの取得に、EPCISがどのように使われるかを説明する。繰り返し使用する資産は、ユニークなシリアル番号を持ったGRAIを用いて識別する。

この場合、繰り返し使用する資産の管理側の当事者が行うプロセスと、そのユーザ側が行うプロセスの2つが組み合わさったビジネス・プロセスになる。管理側のプロセスは、次のようになる。

表5-17 繰り返し使用する資産の管理側ビジネス・プロセス例のプロセス・フローチャート



このプロセスは通常V2からV9までを経た後、資産のユーザ側のビジネス・プロセスを流れて、再びV2に戻る循環経路をたどる。ユーザから返却された資産は、V3で検査を行う。破損していれば、V4で修理をし、修理できなければV5で破棄する。修理の完了したものと、破損の無かったものは、V6で洗浄して、V7でほかの同じ資産と一緒に倉庫に収める。ユーザから空の資産の発注があると、V8でピッキングして、請求書を発行するとともに、V9でユーザに向けて発送する。その後、資産はユーザのビジネス・プロセスに入っていく。

繰り返し使用する資産を管理する当事者が、上記の9つのイベントのすべてを記録する場合には、EPCISによるモデリングは下記ようになる。どのイベントも*What* デイメンションには、そのステップに関連する資産のGRAIが入り、*Where* デイメンションには、該当する管理側の施設の場所が入る。以下では、*What*、*When*、*Where* のデイメンションは省いて簡略化している。

表5-18 繰り返し使用する資産の管理側ビジネス・プロセスに関するEPCISイベントの情報内容(V1 – V4)

ディメンション	データ要素	V1	V2	V3	V4
	説明	新しい資産の生成	空の資産の受取り	資産の検査	修理
	Event Type	Object Event	Object Event	Object Event	Object Event
	Action	ADD	OBSERVE	OBSERVE	OBSERVE
Why	Business Step	Commissioning (CBV)	Receiving (CBV)	Inspecting (CBV)	Repairing (CBV)
	Disposition	Active (CBV)	In Progress (CBV)	(下記を参照)	(下記を参照)

表5-19 繰り返し使用する資産の管理側ビジネス・プロセスに関するEPCISイベントの情報内容(V5 – V9)

ディメンション	データ要素	V5	V6	V7	V8	V9
	説明	修理不能の資産の破棄	資産の洗浄	入庫	出庫	空の資産の発送
	Event Type	Object Event	Object Event	Object Event	Object Event	Object Event
	Action	DELETE	OBSERVE	OBSERVE	OBSERVE	OBSERVE
Why	Business Step	Destroying (CBV)	Cleaning (下記を参照)	Stocking (CBV)	Picking (CBV)	Shipping (CBV)
	Disposition	Destroyed (CBV)	In Progress (CBV)	In Progress (CBV)	In Progress (CBV)	In Transit (CBV)

イベントの補足説明:

- V3のディスポジションと、次に何が起こるかは、検査の結果による:
 - 資産の状態に問題がなければ、ディスポジションはIn Progress(CBVより)で、次のステップはV6(洗浄)
 - 資産の状態に問題があれば、ディスポジションはDamaged(CBVより)で、次のステップはV4(修理)
- V4のディスポジションと、次に何が起こるかは、修理の結果による:
 - 資産の修理が完了すれば、ディスポジションはIn Progress(CBVより)で、次のステップはV6(洗浄)
 - 資産が修理不能ならば、ディスポジションはDamaged(CBVより)で、次のステップはV5(破棄)
- V6のビジネス・プロセス“cleaning(洗浄)”に対応するものは、CBVにないため、ここでプライベートのポキャプラリー要素を用いる状況が起こる。

V9の後は、ユーザのもとで資産が使用されて、そのサプライチェーンの中で商品を運搬する。この繰り返し使用する資産がどのように使われるかは、大きく2つの場合に分かれる:

- ユーザは資産の持つGRAIに関知せず、まったく利用しない。この場合、資産のパレットや運搬箱は何の情報も発さず、GRAIはユーザ側のEPCISイベントにいったい参加しない。ユーザは、資産にほかのGTINを使ったり、あるいは資産で運ぶ荷全体にSSCCを割り振ったりして、資産に積み込んだ製品を追跡するかもしれないが、どちらにしても、資産の管理側が使うGRAIとは、まったく関係ないものになる。
- ユーザが、資産のGRAIのほか、そのデータを持ったバーコードやRFIDを利用する。

5.8 ユーザ/ベンダー拡張要素

EPCISのデータ・モデルは、ビジネス・アプリケーションが、ビジネス・プロセスの各ステップに何が起こったかを理解するうえで必要とする、すべての*What*、*When*、*Where*、*Why* 情報を収めることができるように考えられている。しかし、EPCIS標準で定義されているデータ要素を超えた情報を、ビジネス・アプリケーションが必要とすることも起こる。そのような場合の対処方法として、EPCISイベントには、ユーザ/ベンダーの拡張データ要素を持たせることができる。

ユーザ/ベンダー拡張データ要素とは、単にEPCISイベントに追加するすべてのデータ要素を指す。普通には、ビジネス・コンテキストを加えるため、イベントの*Why* ディメンションの拡張とも見なせるが、追加する内容に制約はなく、ほかのディメンションに関係するものでもよい。

EPCISイベントのXML表現では、拡張データ要素は、EPCIS名前空間とは別の、何らかのXML名前空間のXML要素として表す。EPCIS標準でもCBV標準でも、特定の拡張データ要素をいっさい定義していないため、ほかの標準で定義するか(例えば、業界の標準や、企業グループの間の標準)、あるいは取引相手との間で事前に取り決めておく必要がある。

具体例として、冷蔵コンテナで輸送中に商品の状態を観察することを想定して、オブジェクトの温度と湿度を記録する2つのデータ要素を追加した、EPCISイベントを挙げる：

表5-20 ユーザ/ベンダー拡張を使ったEPCISイベントの情報内容の例

ディメンション	データ要素	V1
	説明	輸送中のコンテナの観察
	Event Type	Object Event
	Action	OBSERVE
When	Event Time	15 July 2015, 10am
What	EPC Quantity List	GTIN X、シリアル番号 101 GTIN X、シリアル番号 102 GTIN X、シリアル番号 103
Where	Read Point	地理座標: (41°40'21"N 86°15'19"W)
	Business Location	(なし)
Why	Business Step	Transporting (CBV)
	Disposition	In Transit (CBV)
	Extension: Celsius Temperature (Example Corp)	15
	Extension: Relative Humidity (Example Corp)	80

このイベントのXML表現は、次のようになる：

```
<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  xmlns:example="http://epcis.example.com/ns"
">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2015-07-15T10:00:00.000-05:00</eventTime>
```

```

<eventTimeZoneOffset>-05:00</eventTimeZoneOffset>
<epcList>
  <epc>urn:epc:id:sgtin:0614141.012345.101</epc>
  <epc>urn:epc:id:sgtin:0614141.012345.102</epc>
  <epc>urn:epc:id:sgtin:0614141.012345.103</epc>
</epcList>
<action>OBSERVE</action>
<bizStep>urn:epcglobal:cbv:bizstep:transporting</bizStep>
<disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
<readPoint>
  <id>geo:41.6725,-86.255278</id>
</readPoint>
<example:TemperatureC>15</example:TemperatureC>
<example:RelativeHumidity>80</example:RelativeHumidity>
</ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>
  
```

この例の2つの拡張要素は、Example社の定義したXML名前空間の中にある。このXML名前空間を使うことで、標準のEPCISデータ要素から拡張を区別できるとともに、ほかの会社が拡張要素に同じ名前を使っても、Example社の拡張と混乱する心配がない。

拡張要素を使う際のガイドラインとして、次のような点に注意する：

- 拡張要素は、あらかじめ取引相手との間で合意しておかなければ、正しく解釈してもらえない。
- 拡張データ要素よりも、常にEPCISの標準のデータ要素を優先して採用して、相互運用性が高くなるようにする。
- 拡張に使うXML名前空間のURIには、拡張を定義する会社や組織の管理下にあるURIを使う。会社や組織のインターネット・ドメイン名に基づいた、HTTP URIを使う方法が推奨される。
- 拡張要素は、そのイベントに情報を加えるデータを与えるために使用し、イベント自体に関係しないデータを伝えるときには使わない。特に識別子についての、インスタンス・レベルやロット・レベルでのマスター・データと見なすべきデータ要素は、拡張要素ではなく、イベントのILMDの部分に入れる。[5.4](#) 節を参照。
- 拡張データ要素は、正しい形式のXMLデータならば、サブ要素や属性など、すべて入れることができる。ただしEPCISのSimpleEventQueryは、値が数値か文字列の拡張要素の問い合わせしかできない。
- EPCISのXMLスキーマが定義しているXML要素<extension>は、ユーザ/ベンダー拡張には使用できない。<extension>要素は、EPCIS標準の将来のバージョンに新しいデータ要素を導入するときに、EPCIS標準自体で使うために確保されたものである。
- EPCISデータを受け取ったアプリケーションは、自分の認識できない拡張があるからと、EPCISイベントを拒否することはできない。そのような場合は、拡張を無視するか、メッセージを出す、あるいは拡張を解釈せずに丸ごと保存するなどして対処する。一方で、取引相手との間で事前に検証規準を確立していて、違反する内容の拡張があった場合は、それを根拠に拒否してよい。

5.9 誤りのあるイベント

本ガイドライン全体を通して説明している通り、EPCISでは、1つのビジネス・プロセスを一連のステップに分解し、それぞれをEPCISイベントとしてモデリングすることによって、そのビジネス・プロセスをモデリングする。これにより、特定のオブジェクトに関連するすべてのイベントの集合(“トレース”)が、EPCIS標準とCBV標準、ならびに関連するその他のボキャブラリ標準で規定される意味に基づいてイベントを解釈し、そのオブジェクトの履歴と現状を正確に示す。

時には、記録済みのイベントに事実が正しく反映されていないことが発見されることもある。しかし、アプリケーションにてEPCISイベントを削除または修正する手段は、EPCISキャプチャ・インタフェースにも、EPCISクエリ・インタフェースにも用意されていない。EPCISイベントを“取り消し”または“修正”する唯一の方法は、記録済みのイベントの効果を無効化または修正するようなビジネス上の意味を持つイベントを新たに生成することである。これにより、完全なトレース(新しいイベント及び不正確なイベントを含むすべての過去イベント)が、上述の原則に示す通りに、履歴と現状を正確に反映する。

イベントを正しく追加する好ましい方法は、誤りのあるイベントの発見とその修正自体が、適切なEPCISイベントの作成によりモデリングできるビジネス・プロセスであると認識しておくことである。ほとんどの場合、これは4章で提示した方法で実行できる。

例 :X社には、シリアル番号101、102、103の製品をY社に出荷したEPCISイベントが記録されている。Y社はそれらの製品を受領したが、シリアル番号101、102、103に加えシリアル番号104も到着していることを発見した。X社に問い合わせた結果、シリアル番号104は確かに出荷されており、その出荷イベントに間違いがあったことが確認された。修正 :X社は、元のイベントと同様の情報にてシリアル番号104が出荷されたという新しいEPCISイベントを記録する。

例 2:X社には、シリアル番号101、102、103の製品をY社に出荷したEPCISイベントが記録されている。Y社はそれらの製品を受領したが、シリアル番号101と102しかなかった。X社に問い合わせた結果、シリアル番号103は出荷されておらず、X社の在庫に残っていることが確認された。両社はシリアル番号103に対する請求を差し戻すことで合意した。修正 :X社は、シリアル番号103の出荷が無効となる新しいEPCISイベントを記録する。

例1の場合、新しいイベントには元のイベントと同じビジネス・ポキャプラリーを用いる。例2の場合は、出荷が無効とするプロセスに関連するポキャプラリーを用いるが、きちんと定義されたビジネス・プロセス・ステップの完了をモデリングするという点で、これは“通常の”EPCISを意味する。これは、そのアクション修正自体がビジネス・プロセスであり、1つのEPCISイベントとしてモデリングできるという事実を反映している。

場合によっては、通常の意味を持つ新しいEPCISイベントを作成し、それによりオブジェクトの履歴を修正することが、不可能な(または極めて望ましくない)こともある。

例 3:X社には、シリアル番号101の製品Xが破棄されたとするEPCISイベントが記録されている。このイベントは、アクションがDELETEのオブジェクト・イベントである。後日、シリアル番号101がまだ倉庫に残っており、破棄されていないことが発見した。この場合、オブジェクト・イベントにとってアクションDELETEは「そのオブジェクトが...その後のイベントに現れることがない」ことを意味しているため、通常イベントで履歴を修正することはできない。

例 4:X社は、複数の製品が出荷されたとするEPCISイベントを記録しており、“why”ディメンションにはビジネス・トランザクションとして発注書番号123が記録されている。Y社はそれらの製品を受領し、入荷イベントを記録した。その後、その出荷イベントの発注参照に間違いがあり、発注番号が123ではなく456と記されていることが発見した。この場合、X社が“出荷が無効とする”イベントを記録し、続いて正しい発注番号で“出荷”イベントを記録することで、通常EPCISイベントを用いて修正することが可能である。しかし、これは全体的なトレース、特にすでに入荷イベントが存在することを考慮すると、あまり望ましくない処理といえる。

このような状況に対応する手段として、前のイベントによるアサーションが間違いだとする意味のイベントを作成するメカニズムがEPCISにはある。そうしたイベントを“エラー宣言イベント”と呼ぶ。

以下の節で、誤りのさまざまな訂正方法をもう少し詳しく説明する。

5.9.1 例 1:通常イベントを使用した訂正 – シンプルな追加

この例では、X社には、シリアル番号101、102、103の製品をY社に出荷したEPCISイベントが記録されている。Y社はそれらの製品を受領したが、シリアル番号101、102、103に加えシリアル番号104も到着していることを発見した。X社に問い合わせた結果、シリアル番号104は確かに出荷されており、その出荷イベントに間違いがあったことが確認された。

修正方法としては、X社は、元のイベントと同様の情報でシリアル番号104が出荷されたという新しいEPCISイベントを記録することになる。

この2つのイベントは以下ようになる：

表5-21 訂正ビジネス・ステップとともに通常イベントを追加することによるエラー訂正の例

ディメンション	データ要素	V1	V2
	説明	4つ目があることに気が付かず、3つの製品インスタンス出荷を記録	4つ目のインスタンスも出荷されていたことを認識する追加のイベント
	Event Type	Object Event	Object Event
	Action	OBSERVE	OBSERVE
When	Event Time	15 July, 10am	15 July, 10am
What	EPC List	GTIN X, シリアル番号 101, 102, 103	GTIN X, シリアル番号104
Where	Read Point	メーカーの積込ドックのSGLN	メーカーの積込ドックのSGLN
	Business Location	(なし)	(なし)
Why	Business Step	Shipping (CBV)	Shipping (CBV)
	Disposition	In Transit (CBV)	In Transit (CBV)
	Source	Owning Party (CBV): X社のGLN	Owning Party (CBV): X社のGLN
	Destination	Owning Party (CBV): Y社のGLN	Owning Party (CBV): Y社のGLN

5.9.2 例 2:通常イベントを使用した訂正 – 訂正ビジネス・ステップ

この例では、X社には、シリアル番号101、102、103の製品をY社に出荷したEPCISイベントが記録されている。Y社はそれらの製品を受領したが、そこにはシリアル番号101と102しかなかった。X社に問い合わせた結果、シリアル番号103は出荷されておらず、X社の在庫に残っていることが確認された。両社はシリアル番号103に対する請求を差し戻すことで合意した。

修正方法としては、X社は、シリアル番号103の出荷が無効となる新しいEPCISイベントを記録することになる。ここでは、こうした目的のために定義されたビジネス・ステップvoid_shippingを使用する。新しいイベントはシリアル番号103のみを参照するため、101や102といった他のシリアル番号の出荷イベントには影響を及ぼさない。したがってこれらのシリアル番号のプロセスは、まだvoid_shippingイベントが記録されていない状態でも続行可能である。

この2つのイベントは以下ようになる：

表5-22 通常イベントの追加によるエラー訂正の例

ディメンション	データ要素	V1	V2
	説明	実際には1つ少なかったことに気が付かず、3つの製品インスタンス出荷を記録	3つ目のインスタンスは実際には出荷されていなかったことを示す追加のイベント
	Event Type	Object Event	Object Event
	Action	OBSERVE	OBSERVE
When	Event Time	15 July, 10am	18 July, 2pm
What	EPC List	GTIN X, シリアル番号 101, 102, 103	GTIN X, シリアル番号103
Where	Read Point	メーカーの積込ドックのSGLN	メーカーの積込ドックのSGLN

ディメンション	データ要素	V1	V2
	Business Location	(なし)	メーカーの倉庫のSGLN
Why	Business Step	Shipping (CBV)	Void Shipping (CBV)
	Disposition	In Transit (CBV)	In Progress (CBV)
	Source	Owning Party (CBV): X社のGLN	Owning Party (CBV): X社のGLN
	Destination	Owning Party (CBV): Y社のGLN	Owning Party (CBV): Y社のGLN

ここで留意すべきは、イベント・タイム、リード・ポイント、ビジネス・ロケーション、ディスポジションは、出荷無効化のプロセスを反映するという点である。つまり、イベント・タイムは、出荷が無効になった日時、ビジネス・ロケーションは、出荷が無効となった後にシリアル番号103が置かれる倉庫、ディスポジションは、シリアル番号103が出荷されていない場合は“in progress”となる。しかし、void_shippingイベントのソースとデスティネーションは、無効にしたビジネス移転の情報を反映して、元の出荷イベントのものと同じになる。

5.9.3 例 3: 訂正イベントなしで、既存のイベントのエラーを宣言

この例では、X社には、シリアル番号101の製品Xが破棄されたとするEPCISイベントが記録されている。このイベントは、アクションがDELETEのオブジェクト・イベントである。後日、シリアル番号101がまだ倉庫に残っており、破棄されていないことが発覚した。この場合、オブジェクト・イベントにとってアクションDELETEは「そのオブジェクトは...その後のイベントに現れることがない」ことを意味しているため、通常イベントで履歴を修正することはできない。

修正方法としては、エラー宣言イベントを発行することになる。これは誤りのある元のイベントと同じように見えるが、エラー宣言のセクションが存在するという違いがある。

この2つのイベントは以下ようになる:

表5-23 エラー宣言イベントの追加によるエラー訂正の例

ディメンション	データ要素	V1	V2
	説明	実際は破棄されていないことに気が付かず、製品Xの破棄を記録	前のイベントが誤りであるとする追加イベント
	Event Type	Object Event	Object Event
	Action	DELETE	DELETE
Error Declaration	Declaration Time		17 July, 2pm
	Reason		Did Not Occur (CBV)
When	Event Time	15 July, 10am	15 July, 10am
What	EPC List	GTIN X, シリアル番号101	GTIN X, シリアル番号101
Where	Read Point	倉庫のSGLN	倉庫のSGLN
	Business Location	(なし)	(なし)
Why	Business Step	Destroying (CBV)	Destroying (CBV)
	Disposition	Destroyed (CBV)	Destroyed (CBV)

5.9.4 例 4:訂正イベントを用いて、既存のイベントのエラーを宣言

X社は、複数の製品が出荷されたとするEPCISイベントを記録しており、“why”ディメンションにはビジネス・トランザクションとして発注書番号123が記録されている。Y社はそれらの製品を受領し、入荷イベントを記録した。その後、その出荷イベントの発注参照に間違いがあり、発注番号が123ではなく456と記されていることが発覚した。この場合、X社が“出荷を無効とする”イベントを記録し、続いて正しい発注番号で“出荷”イベントを記録することで、通常のEPCISイベントを用いて修正することが可能である。しかし、これは全体的なトレース、特にすでに入庫イベントが存在することを考慮すると、あまり望ましくない処理といえる。

修正方法としては、訂正イベントと合わせてエラー宣言イベントを発行することになる。エラー宣言は、誤りのある元のイベントと同じように見えるが、エラー宣言のセクションが存在するという違いがある。訂正イベントは元のイベントの訂正バージョンである。任意で、訂正イベントにユニークなイベントIDを付与し、エラー宣言イベントと関連付けることもできる。

この3つのイベントは以下ようになる：

表5-24 エラー宣言イベントの追加によるエラー訂正の例

ディメンション	データ要素	V1	V2	V3
	説明	発注番号が間違っていることに気が付かず、製品を出荷	前のイベントが誤りであるとする追加イベント	訂正された出荷イベント
	Event Type	Object Event	Object Event	Object Event
	Action	OBSERVE	OBSERVE	OBSERVE
	Event ID			UUID 692...6bd
Error Declaration	Declaration Time		17 July, 1pm	
	Reason		Incorrect Data (CBV)	
	Corrective Event IDs		UUID 692...6bd	
When	Event Time	15 July, 10am	15 July, 10am	15 July, 10am
What	EPC List	GTIN X, シリアル番号101, 102, 103	GTIN X, シリアル番号101, 102, 103	GTIN X, シリアル番号101, 102, 103
Where	Read Point	倉庫のSGLN	倉庫のSGLN	倉庫のSGLN
	Business Location	(なし)	(なし)	(なし)
Why	Business Step	Shipping (CBV)	Shipping (CBV)	Shipping (CBV)
	Disposition	In Transit (CBV)	In Transit (CBV)	In Transit (CBV)
	Business Transactions	PO #456	PO #456	PO #123

5.9.5 エラー宣言イベントと訂正イベントを取得するタイミング

5.9.4 節の例は、1つのエラー宣言に1つ以上の訂正イベントが付随する可能性があることを示している。イベント・データを受け取るEPCISアクセス・アプリケーションに訂正イベントが送られる場合、エラー宣言を認識しているかどうかの問題となる。そうでなければ、このアプリケーションは元の(誤りのある)イベントと訂正イベントに矛盾があるものと認識してしまう恐れがある。

したがって、エラー宣言イベントより前に訂正イベントがEPCISキャプチャ・インタフェースに送られることのないようにする必要がある。一方、もしエラー宣言イベントがcorrectiveEventIDsフィールドを使用してその後の訂正イベントを参照するようにしていた場合、その訂正イベントは、エラー宣言イベントが生成された時点でEPCISアクセス・アプリケーションに認識されると考えられる。この2つの懸念を合わせて解決するには、エラー宣言とそれに関連する訂正イベントが同時に(つまり、EPCISキャプチャ・インタフェースに送られるドキュメントの同じイベント・リストで)取得されるようにすることが推奨される。

ここで留意すべきは、上記は当該イベントのイベント・タイム・フィールドとも、宣言タイム・フィールドとも関連がないということである。エラー宣言の宣言タイムはそのエラー宣言が生成された日時であり、エラー宣言のイベント・タイムは誤りのある元のイベントのイベント・タイム(イベント・タイム自体が元のイベントのエラー箇所でない限り、通常は宣言タイムより前)と一致しており、訂正イベントのイベント・タイムはそのイベントが実際に発生した日時(イベント・タイム自体が元のイベントのエラー箇所でない限り、通常は元のイベントのイベント・タイムと同じ)である。

5.9.6 エラーと訂正が存在している状態でイベントにクエリを出す

エラー宣言イベントにはErrorDeclarationセクションが含まれる。具体的に説明すると、E1というイベントがあり、このE1にエラーがあることを宣言するエラー宣言イベントE2があった場合、E2のイベント構造の内容はE1と一致しつつ、ErrorDeclarationという要素を含むこととなる。例えば、例3の“破棄”イベントのエラー宣言は、アクションをDELETEとするオブジェクト・イベントでもあるが、そこにはErrorDeclarationという要素が含まれる。一般的に、イベントEにエラーがあることを宣言する場合、新しいイベントの内容はイベントEと一致しつつ(レコード・タイムは異なる)、そこにはErrorDeclaration要素が含まれることとなる。

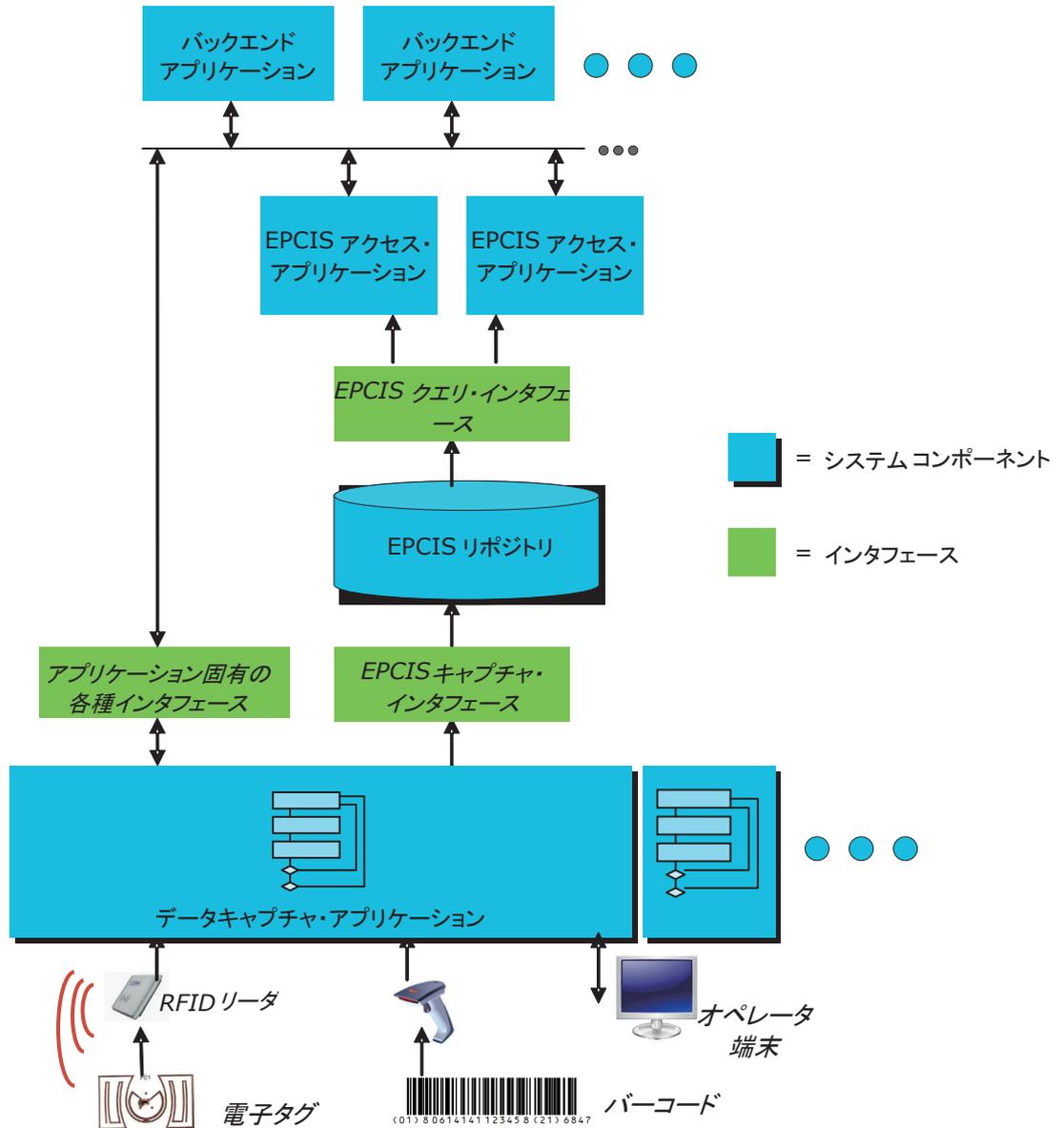
EPCISのエラー宣言イベントがこのように表現される理由は3つある。1つ目の理由は、誤りのあるイベントを示すのにイベントIDを必要としないことである。そのため、今後発生する可能性のあるエラー宣言のためにイベントごとにイベントIDを付与しておく必要もない。イベントIDは、エラー宣言イベントと訂正イベントをリンクするときに利用できるが、必ずしもイベントIDが必要なわけではない。2つ目の理由は、イベントと一致するEPCISクエリでは、エラー宣言があればそちらにも一致することである。これは、EPCISアクセス・アプリケーションは、エラー宣言がある場合、それを認識するのに特別なロジックを必要としないことを意味する。3つ目の理由は、EPCISアクセス・アプリケーションがエラー宣言イベントを受け取った際に、何らかの理由で元の(誤りのある)イベントのコピーがない場合でも、そのイベントのすべての情報はエラー宣言イベントにも存在するため、元のイベントを抽出する必要がないことである。

6 EPCISデータの共有

EPCISデータは、物理/デジタル・オブジェクトを取り扱うビジネス・プロセスの、*What, When, Where, Why* を記録するものである。このようなデータがあれば、さまざまな多くのビジネス・アプリケーションから利用することができる。この章では、1つの会社や組織の中での、アプリケーション間のEPCISデータの共有について説明した後、サプライチェーン全体や、会社を取り巻く環境全体でのプロセスの可視化を目指して、取引相手との間でEPCISデータを共有するにあたっての、実用的な側面を取り上げる。

6.1 1つの組織の中のEPCISデータの共有

EPCISデータを1つの会社や組織の中だけで使う場合、全体のシステム展開は次のようになるのが典型的である：



ここには、次のコンポーネントが見られる:

- **EPCISキャプチャ・アプリケーション**: EPCISキャプチャ・アプリケーションは、ビジネス・プロセスを見張って、プロセスの各ステップが完了するごとに、EPCISイベントを生成するソフトウェアである。大多数は、現場に配備したバーコード・スキャナや、RFIDリーダ、携帯端末、車載端末、装置コントローラなどを通して物理世界とやり取りする。特に、デジタル・オブジェクトを対象にしたEPCISイベントでは、EPCISキャプチャ・アプリケーションが、純粋にソフトウェア上のものになることもある(例えば、5.6 節のデジタル・クーポンの例)。1つの組織内に、EPCISキャプチャ・アプリケーションがいくつもあってもよく、それぞれで違ったビジネス・プロセスや、違った物理位置から、EPCISデータを取得する。
- **EPCISリポジトリ**: EPCISリポジトリは、EPCISデータの格納場所である。EPCISキャプチャ・アプリケーションで生成されるイベントを保存しておき、それらのイベントを、EPCISアクセス・アプリケーション(下記)から利用できるようにする。

- **EPCISアクセス・アプリケーション**: EPCISアクセス・アプリケーションは、EPCISデータを入力として消費して、何らかのビジネス情報機能を実行するソフトウェアである。EPCISイベントにリアルタイムに応答して、ビジネス・プロセスを自動化する場合もあり、あるいはリポジトリに保存されているEPCISイベントの履歴を使って、分析を実施する場合もある。多くの場合は、EPCISイベントをそのまま受け取る準備のない、従来のビジネス・アプリケーションに対してインタフェースとして働き、その際には、データ量を従来アプリケーションの処理できる水準まで縮小する役目も果たす。

実際に導入されるEPCISシステムは、少なくともEPCISキャプチャ・アプリケーションを1つと、EPCISアクセス・アプリケーションを1つは持つ。多くの場合は、それぞれいくつも存在する。EPCISの強みは、複数のキャプチャ・アプリケーションから集めたデータを総合して、複数のアクセス・アプリケーションの間で利用することにある。

EPCISリポジトリは、キャプチャ・アプリケーションからEPCISイベントを受け取り、それを格納して、アクセス・アプリケーションに入力として渡す機能を果たすだけの、独立したソフトウェア・コンポーネントとして導入するのが、ごく一般的である。場合によっては、1つのソフトウェア・コンポーネントの中に、EPCISイベントの格納のほかに、ビジネス機能を付け加えていることもある。この場合、ソフトウェア・コンポーネントはEPCISリポジトリとEPCISアクセス・アプリケーションの両方の役割を果たす。反対に、EPCISリポジトリを取り去って、EPCISキャプチャ・アプリケーションからEPCISアクセス・アプリケーションにデータを直送するだけの、EPCISデータのリアルタイム処理を専門にするシステムもある。これらのことから分かるように、EPCISキャプチャ・アプリケーションと、EPCISリポジトリ、EPCISアクセス・アプリケーションは、システム展開の中のさまざまなソフトウェア・コンポーネントが果たす役割であり、かならずしも独立したコンポーネントではない。

EPCIS標準は、上記のソフトウェアの各役割の間をつなぐインタフェースを2つ定義している:

- **EPCISキャプチャ・インタフェース**: EPCISキャプチャ・アプリケーションが、新しいEPCISイベントを、EPCISリポジトリに送るインタフェース(または、前記のようにEPCISアクセス・アプリケーションへ直接送るインタフェース)
- **EPCISクエリ・インタフェース**: EPCISアクセス・アプリケーションが、EPCISリポジトリからEPCISデータを入手するインタフェース

EPCISキャプチャ・インタフェースは非常に単純で、EPCISキャプチャ・アプリケーションから、1つまたは複数のEPCISイベントを収めたドキュメントを送り渡すためのインタフェースを用意するだけである。次節で説明するEPCISクエリ・インタフェースは、もう少し複雑になる。

6.2 EPCISクエリ

EPCISアクセス・アプリケーションは、EPCISクエリを使って、EPCISリポジトリからEPCISイベントを入手する。EPCISクエリは、アプリケーションによって指定されるイベントの照合条件のセットである。EPCISリポジトリは、指定された条件に一致するすべてのEPCISイベントを検索して結果を送り返す。

EPCIS標準8.2.7.1節に、40個余りの条件が定義されていて、これを使ってイベント・データのクエリを構成する。条件は、単独で使っても、複数を組み合わせて使ってもよい。各々の条件は、“パラメータ名”がEPCIS標準で定義されていて、ほとんどの場合、条件をどのように適用するかを詳しく指定する“パラメータ値”をとる。下の表に、よく使われる条件からいくつかを挙げている。詳しい一覧は、EPCIS標準8.2.7.1節を参照のこと:

表6-1 EPCISのクエリ条件の例

クエリ条件パラメータの名前	クエリ条件パラメータの値	説明
eventType	1つまたは複数のイベント・タイプ: ObjectEvent、 AggregationEvent、 TransactionEvent、または TransformationEvent	イベント・タイプが、パラメータ値で指定されたイベント・タイプのいずれかと同じイベントにマッチする。
EQ_action	1つまたは複数のアクション: ADD、 OBSERVE、DELETE	パラメータ値で指定されたアクションと同じアクションを持つイベントにマッチする。
GE_eventTime	日時値(タイムゾーンの指定子を含む)	イベント時刻が、パラメータ値で指定された日時以降のイベントと一致する。
LT_eventTime	日時値(タイムゾーンの指定子を含む)	イベント時刻が、パラメータ値で指定された日時より前のイベントと一致する。

クエリ条件パラメータの名前	クエリ条件パラメータの値	説明
MATCH_anyEPC	1 つまたは複数のインスタンス・レベル識別子、あるいはインスタンス・レベル識別子にマッチするパターン	What ディメンションのインスタンス・レベル識別子のどれか少なくとも1 つが、パラメータ値で指定された識別子またはパターンのどれかと一致するイベントにマッチする。 MATCH_anyEPC は、What ディメンション全体のインスタンス・レベル識別子を対象にする。EPCIS 標準の定義には、What ディメンションの特定の部分だけを調べるクエリ条件もある(例えば、アグリゲーション・イベントの親だけを調べて、子は除くものなど)。
EQ_readPoint	1 つまたは複数のロケーション識別子	リード・ポイント(Where ディメンション)が、パラメータ値で指定されたロケーション識別子のどれかと同じイベントにマッチする。
EQ_bizLocation	1 つまたは複数のロケーション識別子	ビジネス・ロケーション(Where ディメンション)が、パラメータ値で指定されたロケーション識別子のどれかと同じイベントにマッチする。
EQ_bizStep	1 つまたは複数のビジネス・ステップ識別子	ビジネス・ステップ(Why ディメンション)が、パラメータ値で指定されたビジネス・ステップ識別子のどれかと同じイベントにマッチする。
EQ_disposition	1 つまたは複数のディスポジション識別子	ディスポジション(Why ディメンション)が、パラメータ値で指定されたディスポジション識別子のどれかと同じイベントにマッチする。
EQ_bizTransaction_XXX	1 つまたは複数のビジネス・トランザクション識別子	ビジネス・トランザクション(Why ディメンション)を持ち、そのタイプが XXX で、かつビジネス・トランザクション識別子が、パラメータ値で指定された識別子のどれかと同じイベントにマッチする。 このパラメータを使うときは、パラメータ名の XXX をビジネス・トランザクション・タイプで置き換える。後述を参照。
EQ_XXX	1 つまたは複数の文字列	名前 XXX の拡張要素を持ち、かつその拡張要素の内容が、パラメータ値で指定された文字列のどれかと照合するイベントにマッチする。 このパラメータを使うときは、パラメータ名の XXX を拡張の XML 要素名で置き換える。後述を参照。

1回のクエリに複数の条件を入れることができ、その場合イベントは、**全部の**条件に一致しないと結果に入らない。例えば、GE_eventTime条件とMATCH_epc条件を持ったクエリには、指定のイベント時刻以降に発生して、かつ指定のインスタンス・レベル識別子のどれかを持つイベントだけが一致する。

ビジネスの問いに答えようとする場合、まず必要な情報ははっきりとさせてから、どんなEPCISイベントに、その情報が入っているかを考える。そして、それに従ってEPCISクエリを構成する。下の表は、いくつかの典型的な例で、それをどのように行うかを示している。

表6-2 ビジネス情報の要求と、そのためのEPCISクエリ条件の例

ビジネス情報の要求	関係する EPCIS イベント	EPCIS クエリ条件
EPC XXX が有効であることを確認して、その生成日付のほか、ロット、有効期限その他の関係事項を調べる。	EPC XXX を指した、ステップが commissioning の EPCIS イベント。このイベントが存在すれば、EPC は有効である。同じイベントに、イベント時刻と、ほかの問いに答えるインスタンス/ロット・マスターデータがある。	MATCH_epc: XXX EQ_bizStep: urn:epcglobal:cbv:bizstep:commissioning
2014年3月15日に、入荷ドック#23で入荷した全製品を調べる。	ビジネス・ステップが receiving、リード・ポイントがドック#23で、イベント時刻が指定の日付に入る、全部の EPCIS イベント	GE_eventTime: 2014-03-15T00:00:00Z (2014年3月15日深夜零時 UTC) LT_eventTime: 2014-03-16T00:00:00Z (2014年3月16日深夜零時 UTC) EQ_readPoint: (ドック#23の SGLN 識別子) EQ_bizStep: urn:epcglobal:cbv:bizstep:receiving
発注書#559に従って出荷された製品のシリアル番号を調べる。	ステップが shipping で、トランザクション識別子が発注書#559を指している EPCIS イベント CBVによる発注書のビジネス・トランザクション・タイプは urn:epcglobal:cbv:btt:po 発注書番号は、GLN を使ってビジネス・トランザクション識別子を作成する CBV のテンプレートでコード化。ここでは GLN を 0123456789012 とする。	EQ_bizTransaction_urn:epcglobal:cbv:btt:po: urn:epcglobal:cbv:bt:0123456789012:559

6.3 クエリ・モード:プル vs プッシュ

EPCISクエリを使って、EPCISリポジトリのイベント・データを、その情報を必要とするアプリケーションまたは取引相手に受け渡す。このデータ受渡しは、下記のようにいくつかのかたちのトリガーで発生する。

- **プル**: 要求/応答のやり取りによる受渡し。アプリケーションまたは取引会社が、EPCISクエリ条件を指定した要求を、EPCISリポジトリに出し、EPCISリポジトリが、その条件に一致するEPCISイベントの情報で応答する。
- **プッシュ**: 片方向のメッセージによる受渡し。データが必要なアプリケーションまたは取引相手に対して、EPCISリポジトリの方から、EPCISイベントの情報を送り付ける。この場合、さらに2つのかたちに分かれる:
 - **事前取決め**: 送り手と受け手の各当事者が、EPCIS標準の範囲にない何かのやり方で、どのデータをどんな条件のときに受け側に送るかを、事前に取り決めておく。それに従って、送り手が自身で判断した時点でイベント・データを送る。
 - **サブスクリプション**: 受け手が、送り手に**継続クエリ**を送って、継続的に情報を必要とすることを伝える。継続クエリには、“プル”の場合と同じようにEPCISクエリ条件を入れるとともに、イベント・データを送るトリガーになる条件を添える。トリガー条件としては、定期スケジュール(例えば、毎日午前3時)のほか、何らかのトリガー・イベントを指定できる。トリガー条件が発現するごとに、送り側がクエリ条件を評価して、条件に一致する新しいイベント(前回のトリガー以降のもの)を送る。

“プッシュ”の2つの方法は、どちらも送り手から受け手に一方方向の通信でEPCISイベントを送るが、事前取決めの形式では、いつどのデータを送るかが送り手ですべて決まってしまうのに対して、サブスクリプションの形式では、継続クエリを通して、受け手が随時にニーズを伝えることができる点に違いがある。なお、6.7節で述べるように、どの方法(“プッシュ”と“プル”)も、最終的には送り手に、どのデータを送るかの決定権がある。

EPCISデータを収集するビジネス・プロセスを設計する際には、それぞれの形式のクエリを、各々の条件に応じて使い分けることになる。一般的に言って、“プッシュ”は、EPCISデータの収集の要求が、繰り返し起こることが前もって分かる場合に使い、“プル”は、例外的な状況など、前もって分からないデータ収集の要求があるときに使う。それぞれの方法を、どんな状況で使うかの例を下の表に挙げている:

表6-3 ビジネス・シナリオと、それに対応するEPCISクエリ・モード

ビジネス・シナリオ	クエリ・モード	クエリ・モードの使い方
GTIN X、ロット Y がリコールになり、製造元 XYZ で、小売業者 ABC に納入した製品に、対象品がないかを調べる必要がある。	プル	製造元 XYZ から、ABC 社の EPCIS リポジトリに要求を送り、What ディメンションに GTIN X、ロット Y が入っていて、ビジネス・ステップが“receiving”のイベントすべてを問い合わせる。
地域の規則により、卸売業者 PQR から ABC 薬局に医薬品を納入すると、各シリアル番号の情報を、出荷から 1 時間以内に併せて送ることが求められる。	事前取決めによるプッシュ	PQR 社と ABC 薬局が事前に取り決めに交わし、さらに PQR 社がデータのメッセージを送るアドレスを、ABC 薬局から連絡しておく。PQR 社は、ABC 薬局向けに医薬品を出荷するごとに、EPCIS リポジトリから ABC 薬局にメッセージを送って、What ディメンションにその医薬品の SGTIN を持ち、ビジネス・ステップが“shipping”の EPCIS イベントの情報を送り渡す。
小売業者 ABC は、危険物質を含んだ製品 X の出荷が、製造元 XYZ からあるときには、荷の特殊な取扱いに備えるために通知を必要とする。	サブスクリプションによるプッシュ	小売業者 ABC から継続クエリを出し、その中で、What ディメンションに X の GTIN を持ち、ビジネス・ステップが“shipping”で、デスティネーション・リストに自社の GLN が入った全 EPCIS イベントを検索する条件と、毎日 1 回実行するトリガーを指定する。

6.4 EPCISクエリ制御インターフェース

EPCIS標準は、EPCISアクセス・アプリケーションまたは取引会社が、EPCISリポジトリとやり取りする標準のインターフェースを用意している。このインターフェースを通して、アプリケーションや取引会社は、“プル”のクエリの発行や、“プッシュ”のサブスクリプションの設定をはじめ、各種の操作ができる。

このインターフェースから実施できる操作を、下の表にまとめている:

表6-4 EPCISクエリ制御インターフェースの処理

処 理	説 明	要求の内容(EPCIS アクセ ス・アプリケーションまたは取 引会社から)	応答の内容(EPCIS リポジトリ)
Poll	“プル”のクエリの実行	EPCIS クエリ条件	クエリ条件に一致する EPCIS イベント
Subscribe	“プッシュ”のサブスクリプションの設定	サブスクリプション ID (要求側が指定)、EPCIS クエリ条件、トリガー条件、継続クエリの結果を送るアドレス	確認応答 その後 EPCIS リポジトリは、トリガーが起こるごとに、条件に一致するイベントを指定のアドレスに送る。
Unsubscribe	設定してあるサブスクリプションの解除	サブスクリプションの設定に使ったサブスクリプション ID	確認応答
getSubscriptionIDs	現在有効なサブスクリプションの問い合わせ	[添付情報なし]	要求側から設定してあるサブスクリプションのサブスクリプション ID のリスト
getStandardVersion	EPCIS リポジトリが対応している、EPCIS 標準のバージョンの問い合わせ	[添付情報なし]	リポジトリの対応バージョンに応じて 1.0 または 1.1
getVendorVersion	EPCIS リポジトリの実装についての、ベンダー固有の情報の問い合わせ	[添付情報なし]	EPCIS リポジトリのベンダーが設定している文字列
getQueryNames	EPCIS リポジトリが対応している、EPCIS クエリのタイプの問い合わせ	[添付情報なし]	EPCIS リポジトリが対応しているクエリのリスト。EPCIS 標準で定義されている SimpleEventQuery は必ずあり、SimpleMasterDataQuery を含むこともある。ほかにベンダー固有のクエリを持つこともある。

EPCIS標準は、EPCISクエリ・インターフェースで使うそれぞれの要求と応答のメッセージについて、XML表現を定義している。

6.5 コレオグラフィ・モデル: サプライチェーン全体のデータ共有

2つの取引会社の間だけでEPCIS情報を共有する場合、情報の流れはごく簡単である。すでにビジネス関係があるため、2社の間でどのデータを共有し、どのクエリ・モードを使うかを取り決める。

たくさんの会社に関わる取引環境では、もう少し複雑な状況になる。それぞれの当事者が多くの会社と取引を交わし、その取引関係の1つ1つにEPCISデータの交換が求められる可能性がある。さらに、直接の取引関係のない他社との間にも、EPCISデータの共有が必要になることがある。例えば、A社がB社に販売し、B社がC社に販売している場合、A社とC社の間に直接の取引関係はなくても、サプライチェーンの全体を知るためには、A社とC社間のEPCISデータの共有が必要になる。

ここで役に立つのが、コンテンツとコレオグラフィを分けて考えることである。コンテンツとはEPCISデータである。どのビジネス・ステップに可視化が必要か、そのステップの遂行をどのEPCISイベントで記録するか、イベントにどんなデータを持たせるかなどのコンテンツは、本書のこれまでの章で述べた方法に従って設計する(3章、4章、5章)。そこでは、各ビジネス・ステップのWhat、When、Where、Whyの情報を、正確にモデリングすることに焦点を当てる。その一方で、取引会社の間では、いつ、どのようにデータを受け渡すかも決めなければならない。これをコレオグラフィと呼ぶ。コレオグラフィで決めるのは、データをどこに置か、何をトリガーにして当事者間でデータをやり取りするか、プッシュを使うかプルを使うか、どのネットワーク・テクノロジーを使うかなどである。コンテンツをコレオグラフィから切り離すことで、取引相手の広がりや縮小、テクノロジーの進歩があった際にも、EPCISのコンテンツの設計を保ったまま、変化に適応していくことができる。

コレオグラフィには、いろいろなやり方が考えられる。その多くは、大きく分けて次の3つのどれかに入る：

- **中央集中型コレオグラフィ**：このモデルでは、サプライチェーンの中のいくつもの当事者からのEPCISイベントを、共有リポジトリに送る。サプライチェーン全体から情報を調べたいときは、共有リポジトリにクエリを出すだけでよい。
- **分散型クエリ・コレオグラフィ**：このモデルでは、EPCISデータを取得した各当事者が、自分のリポジトリでそのデータを保管する。サプライチェーン全体の情報を調べたいときは、それぞれのデータをリポジトリに保管している当事者を全部調べ上げて、クエリを出さなければならない。
- **分散型プッシュ・コレオグラフィ**：このモデルでは、EPCISデータを取得した各当事者が、自分のリポジトリでそのデータを保管する。しかし、ほかの当事者からデータにクエリがあるのを待つのではなく、取得した側から、サプライチェーンの中で、そのデータを必要としそうなほかの当事者に向けてデータを送る(プッシュする)。データのプッシュは、物

理/デジタル・オブジェクトと同じ経路を取ることが多い。
 例えば、A社が、商品をB社に出荷する場合、B社に向けて、そのEPCISデータも送る。

次からの節で、例を挙げて3つの方法をもう少し詳しく説明する。どの節も、次のようなシナリオを考える。A社が商品をB社に出荷し、B社がC社に商品を出荷していて、C社には、商品を受け取ると、A社とB社から出る上流側のEPCISイベントを調べる。

それぞれのコレオグラフィの相違点を、次の4つの問いを通して対比する：

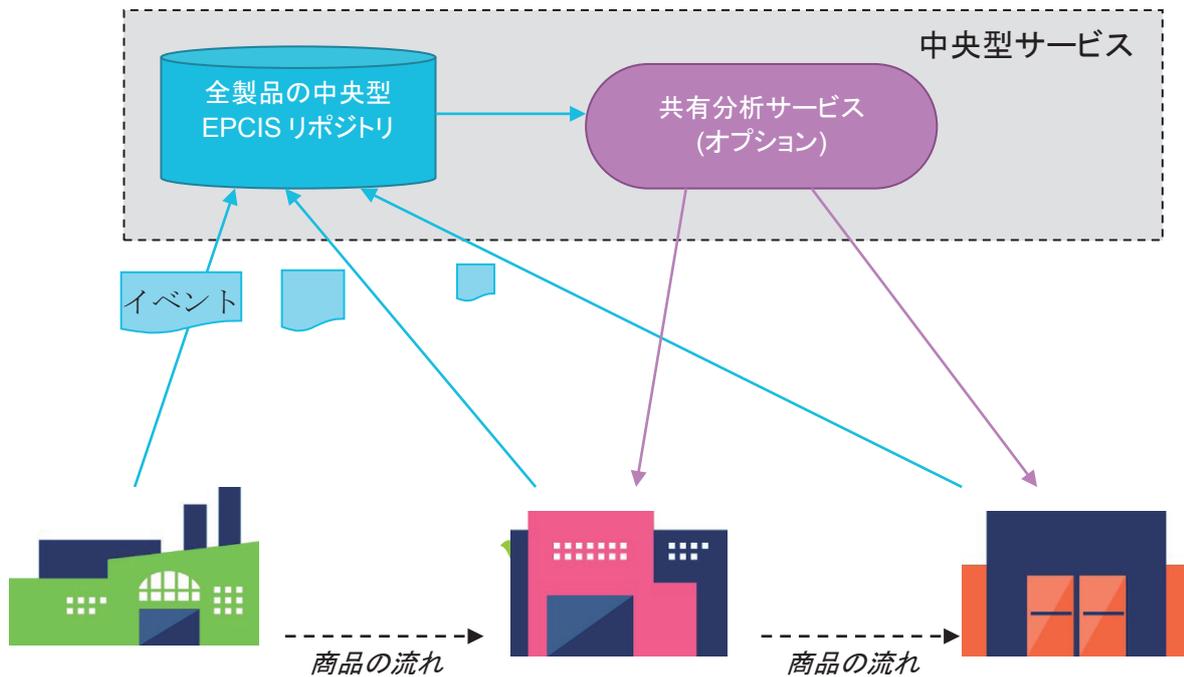
- EPCISイベント・データの生成側についての問い：
 - いつEPCISイベントをほかの当事者と共有するか？
 - 共有するデータを、どこに送るか？
- EPCISイベント・データの消費側についての問い：
 - 分析の際、複数の当事者が生成するイベントをどうやって集めるか？
 - イベントの収集と分析は誰の役割か？

それぞれの当事者は、いろいろなビジネス・プロセスの背景の中で、生成側としても、消費側としても振る舞うことがある。

6.5.1 中央集中型コレオグラフィ

最も簡単なコレオグラフィ・モデルは、1個のEPCISリポジトリを置いて、サプライチェーンのすべての当事者が共有するかたちである。

図6-1 集中型コレオグラフィ



中央集中型コレオグラフィのモデルの特徴は、次のようにまとめられる：

表6-5 中央集中型コレオグラフィの特徴

問 い	中央集中型コレオグラフィ
いつ EPCIS イベントを共有するか？	生成側が、イベントを取得したとき、または製品を出荷するとき
共有するデータをどこに送るか？	生成側から中央リポジトリに送ってデータを共有
分析するイベントをどうやって集めるか？	全イベントが中央リポジトリにあるため、特にイベントを集める処理は不要
イベントの収集と分析は誰の役割か？	消費側が中央リポジトリにクエリを出して、自身で分析を実施するか、あるいは中央リポジトリが分析サービスを提供して、消費側に代わって実施

中央集中型は、イベントがすべて1カ所に集まるため、分析のためのデータ収集が簡単になることが長所である。また、

そのため共有型の分析サービスを提供する場所としても自然である。

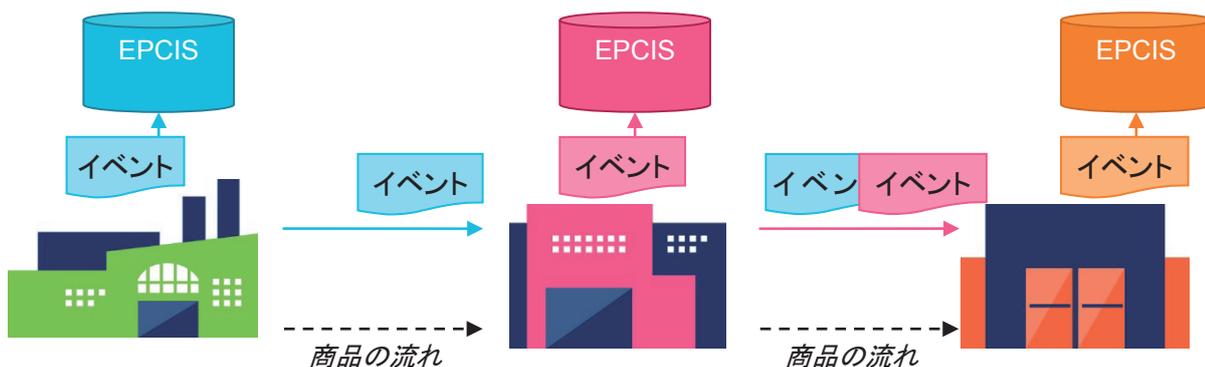
中央集中型の短所の1つは、サプライチェーンの全当事者が、同じリポジトリ・サービスを使うことに合意しなければならない点である。そのため、非常に大きなサプライチェーンでは現実的ではないかもしれない。中央集中型の変形として、共有リポジトリをいくつも用意する方法も考えられる – これは*セミ中央集中型*と呼ぶ。リポジトリが複数になるため、分析に必要なデータが、全部1つのリポジトリにあるとは限らない。セミ中央集中型では、この対処のため追加の処理が必要である。例えば、次のような方法が考えられる：

- 共有リポジトリの間で連携し、互いにデータをコピーして同期を取るか、必要に応じてクエリをほかのリポジトリに転送する。
- クエリが、1つのGTINなど、特定のEPCクラスのエントの収集に限られる場合は、それに合わせてリポジトリを切り分ける。この場合、それぞれのEPCクラスを特定のリポジトリ(たいていの場合、EPCの作成当事者が決める)に割り振って、何らかの照会サービスに登録しておき、EPCクラスと特定のリポジトリの対応が分かるようにしておく必要がある。この目的には、オブジェクト・ネーム・サービス(ONS)が利用できる。下流側の当事者は、照会サービスを使って、どのリポジトリでデータを共有すべきかを知る。

6.5.2 分散型プッシュ・コレオグラフィ

分散型プッシュ・コレオグラフィでは、サプライチェーンの各当事者が、取得データを自分のEPCISリポジトリに保管しながら、同時に物理オブジェクトのフローに沿って、下流側にもEPCISイベントのコピーを送る。EPCISクエリを使うことはない。

図6-2 分散型プッシュ・コレオグラフィ



分散型プッシュ・コレオグラフィのモデルの特徴は、次のようにまとめられる：

表6-6 分散型プッシュ・コレオグラフィの特徴

問 い	分散型プッシュ・コレオグラフィ
いつ EPCIS イベントを共有するか？	生成側が、物理オブジェクトを下流側当事者に出荷するとき
共有するデータをどこに送るか？	下流側当事者と、さらにその下流側当事者
分析するイベントをどうやって集めるか？	下流側当事者は、上流側の全イベントを受け取るため、特にイベントを集める処理は不要
イベントの収集と分析は誰の役割か？	消費側の当事者

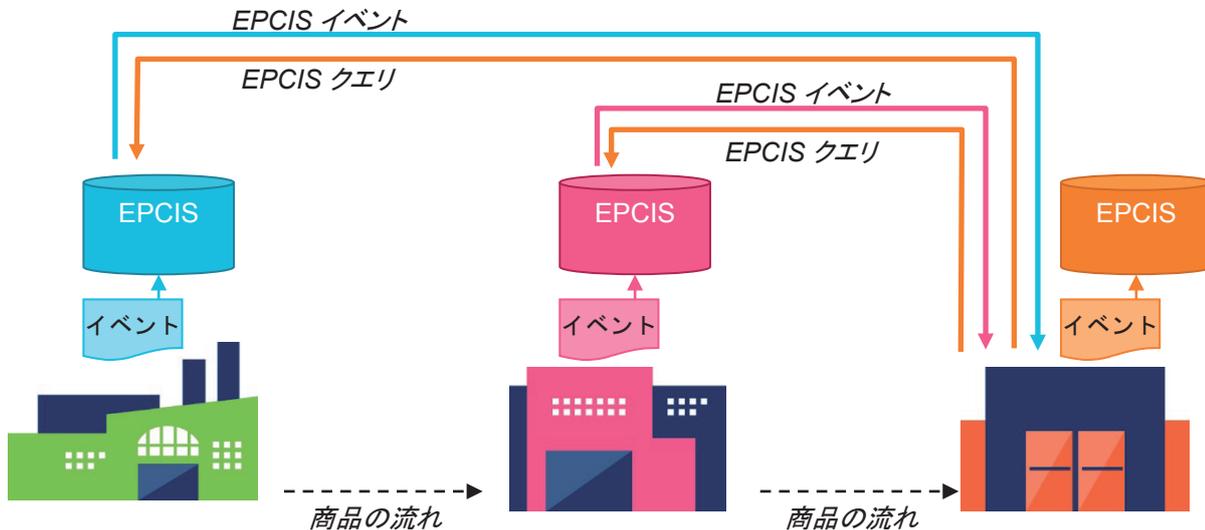
分散型プッシュは、消費側が必要なデータを事前に受け取り、クエリを出す必要がないのが長所である。消費側は、ほかの当事者のEPCISサービス(あるいは、いっさいの共有サービス)の稼働状態に左右されないため、安定した処理ができる方法でもある。この方式の短所は、データが結局は必要にならなくても、イベントを受け渡す点である。また経路途中の当事者は、まったく使うことのないイベントでも中継しなければならない。

上記のように、分散型プッシュ方式は上流側のイベントを下流側の当事者に送っていくが、下流側のイベントを上流側の当事者が受け取って、逆方向にデータを送ることも可能である。

6.5.3 分散型クエリ・コレオグラフィ

分散型クエリ・コレオグラフィでは、サプライチェーンの各当事者が、取得データを自分のEPCISリポジトリに保管する。ほかの当事者のデータを必要とするときには、そこにクエリを出す。

図6-3 分散型クエリ・コレオグラフィ



分散型クエリ・コレオグラフィのモデルの特徴は、次のようにまとめられる：

表6-7 分散型クエリ・コレオグラフィの特徴

問 い	分散型クエリ・コレオグラフィ
いつ EPCIS イベントを共有するか？	ほかの当事者からクエリが出たときのみ
共有するデータをどこに送るか？	データを要求した当事者
分析するイベントをどうやって集めるか？	各々の当事者の EPCIS リポジトリにクエリを出す。このためには、クエリを出す EPCIS リポジトリのディスカバリのための仕組みが必要
イベントの収集と分析は誰の役割か？	消費側の当事者

分散型クエリの方法の長所は、各当事者が自分のデータを確実に管理して、データを必要とする当事者にだけ、データを直接に送ればよい(ほかの当事者を中継しない)点である。共有サービスにも頼らない。

この方法で課題になるのはディスカバリの手段、つまり、EPCISデータの消費側がクエリを出すEPCISリポジトリを、どうやって見つけるかである。ディスカバリ処理は、次のように分解される：

- 消費側が求める情報に関するデータを、どの当事者が持っているか(または、持っている予想されるか)を調べる。
- クエリの送り先の当事者の、EPCISサービスのネットワーク・アドレスを入手する。
- クエリの応答側から認証を受けて信頼関係を確立し、クエリの応答側が、要求側の求めるデータへのアクセスを安心して許可できるようにする。これは、要求側と応答側の間に直接のビジネス関係がない場合、例えば、サプライチェーンの中で1ステップ以上離れているときなどには複雑になる。

ディスカバリの問題にはさまざまな解決策が考えられるなかで、例えば次のものが挙げられる：

- **所有者チェーン・トークン**：サプライチェーンの各当事者が、次の下流側の当事者に短いメッセージを送り、その中で、出荷した物理オブジェクトに関するEPCISデータのアクセスを認める認証トークンと、EPCISサービスのネットワーク・アドレスを知らせる。サプライチェーンの中の各当事者は、下流側の当事者に、自分の所有者チェーン・トークンを渡すとともに、上流側の当事者から受け取ったトークンを送り渡す。これによって、下流側の当事者は、受け取った物理オブジェクトについてのデータを持つ、上流側の全当事者にアクセスするトークンを入手する。

前にも触れたように、このような場合、下流側の当事者が上流側のデータについて知ることができても、上流側の当事者に下流側のことは分からない。そこで、別のトークンを上流側に向けて送り渡していけば、上流側の当事者にも、下流側のデータのことが分かるようになる。

- **ディスカバリ・サービス**: 中央集中型の“ディスカバリ・サービス”を稼働させて、関連するすべてのEPCISデータの置き場所へのインデックスを提供する。EPCISデータを取得した当事者は、ディスカバリ・サービスにメッセージを送って、自分のEPCISサービスのネットワーク・アドレスと、そこにデータを置いている物理オブジェクトを知らせる。その後、消費側の当事者からディスカバリ・サービスにクエリを出すと、必要なデータを置いているEPCISサービスを調べ上げることができる。ディスカバリ・サービスに送る情報には、消費側の当事者が生成側にクエリを出すときに、信頼関係を確立するための認証情報を加えてもよい。

ディスカバリの問題は、送る情報がEPCISデータのポインタであるという点を除けば、EPCISイベント自体を配布する問題と同じである。そう考えると、ディスカバリ・サービスは、ポインタ・データの中央集中型モデルであり、所有者チェーンの方法は、ポインタ・データの分散型プッシュ・モデルである。ただし、EPCISデータのポインタは、EPCISイベント本体よりもデータが小さいため、EPCISイベントの中央集中型コレオグラフィと分散型プッシュ・コレオグラフィに比べると、集積するデータやプッシュするデータの規模は小さくなる。

6.6 マスタ・データの同期

EPCISイベントの*What*と*Where*のディメンションは、全世界でユニークな識別子のかたちを取る。例えば、*What*ディメンションならばSGTIN(グローバル・トレードアイテム・ナンバー+シリアル番号)、*Where*ディメンションならばGLN(グローバル・ロケーション・ナンバー)である。ビジネス・アプリケーションがEPCISイベントのビジネス上の意味を解釈する際には、通常それぞれの識別子に関係付けられた追加の記述情報が必要になる。GTINの記述情報としては、例えば製品名、ブランド名、外形寸法など、GLNの記述情報としては、例えば場所の番地や、地理座標などである。このような記述情報が、“マスタ・データ”と呼ばれる。

EPCISイベント・データに比べて、マスタ・データは静的である。イベント・データと違って、取引が行われたというだけでは、マスタ・データは増えない。ただし、まったく固定しているのではなく、例えば、新製品の発売や、新しい施設の建設など、ビジネスの拡大があると、新規のマスタ・データが生成される。しかし、一般に識別子は、同じ内容のマスタ・データを関係付けただけで、いくつものEPCISイベントに現れる。そのため、マスタ・データを1つ1つのEPCISイベントに入れるよりも、各識別子につき1回だけ、前もって知らせておく方が能率的である。

識別子を作成した当事者から、そのマスタ・データを必要とするほかの当事者に、マスタ・データの内容を伝える方法はいくつか考えられる。例えば、次のような方法がある：

- マスタ・データの効率的な伝達のために考えられた、次のシステムを利用する：
 - GS1グローバルデータ同期化ネットワーク(GDSN): 商品(GTIN)のマスタ・データと、GLNのマスタ・データ
 - GS1 GLNレジストリ: GLNの詳しい補足情報
- EPCIS 1.1標準8.2.7.2節で規定されているEPCISマスタ・データ・クエリを利用する。
- イベントに直接マスタ・データを入れる、EPCISのインスタンス/ロット・マスタ・データ(ILMD)機能を利用する。これは、GTINの特定のロットや、特定のインスタンス(SGTINや、その他のインスタンス・レベル識別子)の、マスタ・データが対象になる。
- 標準のEPCISヘッダーのVocabularyList要素を使って、マスタ・データをEPCISドキュメントに入れる。
- GS1標準の範囲にない、その他の手段。

これらの中で、GDSNとGLNレジストリは、すべて標準で定められているため、相互運用性が最も優れている。EPCISのマスタ・データ・クエリ、ILMD機能、標準のEPCISヘッダーのVocabularyList要素は、標準化されたインタフェースをマスタ・データに用意し、コア・ビジネス・ポキュラリで定義されたマスタ・データ属性と一緒に使用されることもある。

6.7 EPCISイベント・データのリダクション

EPCISの基本原則の1つとして、EPCISデータを取得する当事者は、そのデータを所有し、ほかのどの当事者がそのデータを受け取るかについて支配権を持つ。したがって、ある当事者が、ある条件に一致するEPCISイベントを問い合わせるクエリを送ったというだけで、それを受け取った側が、条件に一致するイベントをすべて応答する義務はない。応答側は、ビジネス・ルールに基づいて、要求側が受け取るデータを制限してもよい。これを“リダクション”(編集)と呼ぶ。

一般にEPCISサービスは、クエリに応答する場合でも、その他のトリガーによる場合でも、ほかの当事者にデータを送る際、受け取り側の身元を考えてビジネス・ルールを適用することで、データにリダクションを加えてよい。リダクションとしては、次のようなことが考えられる。これは、EPCIS 1.1標準8.2.2節の記載に手を加えたものである。

- サービスは、セキュリティ例外で応答することで、要求全体を拒否してよい。
- サービスは、要求されたよりも少ないデータで応答してよい。例えば、指定期間内のすべてのオブジェクト・イベントのインスタンスを要求するクエリがあった場合、100個の一致するイベントがあっても、サービスはその全部を送らなくてもよい(例えば、要求側当事者の事業者コードの入った、SGTINのEPCを持つイベントだけを送るなど)。
- サービスは、要求されたよりも粗い粒度の情報で応答してよい。具体的には、クエリの応答にロケーション識別子が含まれる場合、詳しい位置の情報を、付近全体の場所の情報で置き換えてもよい(例えば、特定の入荷ドックのSGLNの代わりに、施設全体のGLNを送るなど)。
- サービスは情報を隠してよい。例として、オブジェクト・イベントのインスタンスを要求するクエリに対して、応答からbizTransactionListフィールドを取り除くこともできる。ただし送る情報は、必ずこの仕様と業界ガイドラインに従った、正しい形式のEPCISイベントにならなければならない。例えば、アクションがADDのAggregationEventで、parentIDフィールドを隠そうとすると、アクションがADDの場合はparentIDが必須のため、形式の整わないイベントになる。このような場合には、parentIDを残すか、イベント全体を応答から差し控える。
- サービスは、クエリの対象範囲を、特定のクライアントの取得したデータに制限してよい。こうすると、データを個別に管理しなければならない、相互に無関係な複数のユーザの間で、1つのEPCISサービスを区分けして利用することができるようになる(いわゆる、“マルチテナント”実装)。

EPCISの実装にあたっては、上記のどの措置を、どのような手段でクエリの処理に取り入れるかを自由に判断してよい。また、認証規則は、EPCIS標準の範囲を外れる。EPCIS標準では、どのように認証を行うかについて、いっさいの立場を取っていない。EPCISの実装は、認証についてどんな複雑なビジネス規則でも自由に採用してよい。

7 データの検証とシステムの相互運用性

7.1 EPCISイベントの検証

EPCISを使った可視化システムの働きは、EPCISイベントのデータ品質に大きく左右される。そのため企業や組織では、データ検証の仕組みを取り入れるとよい。検証はテクニカル検証、コンテンツ検証、インテグリティ検証の3つの種類に分かれる:

- **テクニカル検証**とは、EPCISイベントを、形式の上で現行のEPCIS標準に準拠させることを意味する。つまり、EPCIS 1.1標準で規定されているXMLスキーマ(XSD)に従ったXMLフォーマットで、イベントを送ることである。ユーザ/ベンダー拡張を加える場合には、ベストプラクティスとして、必要なXML要素と、拡張分の名前空間に対するXSDを作成して、それを加えてテクニカル検証を実施する。
- **コンテンツ検証**では、各々のイベントが、ビジネスの観点から意味の通るものであることを検証する。例えば、ある現場の特定のプロセス・フローの中で、パレット積みの梱包イベントに、SSCCのほか、LGTINで表される梱包箱の個数を収めることが規定されていれば、コンテンツ検証では、梱包イベントがその構造の内容を持ち、ほかのかたち(構文上は正しくても、現場の作業に合わないもの)にはなっていないことを確かめる。また、EPCISのキャプチャ・アプリケーションでも、日付検証のような意味的チェックを実施できる。例えば、eventTimeの値が将来の時刻になっていないかなどである。

- **インテグリティ検証**では、可視化システムが、実プロセスと対応付けて設計した通りにエンド・ツー・エンドで働き、所要のビジネス目的を果たす結果を生成することを検証する。例えば、商品に見られた欠陥を、同じ施設内の入荷段階までトレースすることを目指したシステムならば、それが達成されていることを確認する。

テクニカル検証とコンテンツ検証は、一般にEPCISキャプチャ・インタフェースからEPCISイベントが送られる時点で、ただちに行うことができる。データ品質の重要度が高い場合、EPCISのリポジトリやキャプチャ・アプリケーションは、あらかじめ決めておいたテクニカル検証とコンテンツ検証の条件が満たされない限り、EPCISイベントの受け取りを拒否してもよい。あるいは、テクニカル検証に合格する限りはイベントを受け取って、コンテンツ検証に問題があれば警告を出すなどのかたちでもよい。

インテグリティ検証については、エンド・ツー・エンドのプロセス全体でイベントがすべて取得された後に、さかのぼって実施するしかない。イベントが正しいシーケンスで取得されていない場合、可視化イベント・データを消費する側のビジネス・アプリケーションは、あらかじめ決めておいた規則に従って処置を取る。例えば、そのビジネス・プロセスにならないイベントが欠落している場合、警報を出して注意を促してもよい。また、明らかな重複イベントや意味のないイベントは、無視することも考えられる。

7.2 認証プログラム

EPCglobalソフトウェア認証プログラムは、EPCglobalコミュニティによって策定された、標準ベースの準拠試験プログラムであり、EPC/RFIDソフトウェア製品の試験について、中立で権威ある典拠となるとともに、認証製品とその開発ベンダーの情報も提供する。

7.3 プログラムの認証要件

EPC Information Services(EPCIS) 1.0仕様準拠要件：
<http://www.gs1.org/gsm/kc/epcglobal/epcis>

7.4 データ検証ポータル

GS1で用意している自己テスト式のEPCIS検証ポータル：<http://www.epcisvp.com>

7.5 ソフトウェアの認証全般

認証についてのその他の事項：<http://www.gs1.org/software-certification-program>

8 参考文献

[CBV] GS1, "Core Business Vocabulary Standard, Release 1.2," GS1 Standard, September 2016,
<http://www.gs1.org/sites/default/files/docs/epc/CBV-Standard-1-2-r-2016-09-29.pdf>.

[EPCIS] GS1, "EPC Information Services (EPCIS) Standard, Release 1.2," GS1 Standard, September 2016,
<http://www.gs1.org/sites/default/files/docs/epc/EPCIS-Standard-1.2-r-2016-09-29.pdf>

[GenSpecs] GS1, "GS1 General Specifications," GS1 Standard, January 2016,
http://www.gs1.org/docs/barcodes/GS1_General_Specifications.pdf

[GS1Arch] "The GS1 System Architecture," GS1 technical document, April 2016,
http://www.gs1.org/sites/default/files/docs/architecture/GS1_System_Architecture.pdf

[TDS] GS1, "EPC Tag Data Standard, Version 1.9," GS1 Standard, November 2014,
http://www.gs1.org/sites/default/files/docs/epc/TDS_1_9_Standard.pdf

A 付録:XMLの例

この節では、本文中に表形式で挙げたEPCISイベントに対する、XML表現の例を掲載する。それぞれのXML例は、EPCIS 1.1規格9.2節のXMLスキーマによるEPCISDocumentである。各ドキュメントの中のイベントは、表の左から右の順番に並んでいる。

本文中の表に記載した例の多くは、簡略化のためにいくつかのディメンションを省略するとともに、実際の識別子を使わずに、“GTIN X”などのかたちで表記している。XML例では、これらの細部もサンプル値を使ってすべて加えている。

A.1 表3-1のEPCISイベントのXML

```
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T09:05:08.720-04:00">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2012-09-23T05:10:12.000Z</eventTime>
        <eventTimeZoneOffset>-05:00</eventTimeZoneOffset>
        <epcList>
          <epc>urn:epc:id:sgtin:0614141.112345.12345</epc>
        </epcList>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:5012345.67890.D123</id>
        </readPoint>
        <bizLocation>
          <id>urn:epc:id:sgln:5012345.67890.0</id>
        </bizLocation>
        <bizTransactionList>
          <bizTransaction
            type="urn:epcglobal:cbv:btt:po">urn:epcglobal:cbv:bt:5012345000015:ABC123</bizTransaction>
          <bizTransaction
            type="urn:epcglobal:cbv:btt:inv">urn:epcglobal:cbv:bt:0614141000012:XYZ987</bizTransaction>
          </bizTransactionList>
          <extension>
            <sourceList>
              <source
                type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00001.0</source>
            </sourceList>
            <destinationList>
              <destination
                type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:5012345.00001.0</destination>
            </destinationList>
          </extension>
        </ObjectEvent>
      </EventList>
    </EPCISBody>
  </epcis:EPCISDocument>
```

A.2 表4-6の例のXML

```
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T09:33:00.597-04:00">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
```

```

    <eventTime>2014-03-15T10:11:12.000Z</eventTime>
    <eventTimeZoneOffset>-05:00</eventTimeZoneOffset>
    <epcList>
      <epc>urn:epc:id:sscc:0614141.0123456789</epc>
    </epcList>
    <action>OBSERVE</action>
    <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
    <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
    <readPoint>
      <id>urn:epc:id:sgln:0614141.11111.2</id>
    </readPoint>
    <bizTransactionList>
      <bizTransaction
type="urn:epcglobal:cbv:btt:po">urn:epcglobal:cbv:bt:5012345678900:1234</bizTransact
ion>
      <bizTransaction
type="urn:epcglobal:cbv:btt:inv">urn:epcglobal:cbv:bt:0614141111114:9876</bizTransac
tion>
    </bizTransactionList>
    <extension>
      <sourceList>
        <source
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.11111.0</source>
      </sourceList>
      <destinationList>
        <destination
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:5012345.67890.0</destinati
on>
      </destinationList>
    </extension>
  </ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.3 表5-3の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T09:46:05.724-04:00">
  <EPCISBody>
    <EventList>
      <AggregationEvent>
        <eventTime>2015-03-15T10:00:00.000+01:00</eventTime>
        <eventTimeZoneOffset>+01:00</eventTimeZoneOffset>
        <parentID>urn:epc:id:sscc:5012345.0678901234</parentID>
        <childEPCs>
          <epc>urn:epc:id:sgtin:5012345.177777.1001</epc>
          <epc>urn:epc:id:sgtin:5012345.177777.1002</epc>
          <epc>urn:epc:id:sgtin:5012345.177777.1003</epc>
          <epc>urn:epc:id:sgtin:5012345.177777.1004</epc>
          <epc>urn:epc:id:sgtin:5012345.177777.1005</epc>
        </childEPCs>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:packing</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:5012345.00000.123</id>
        </readPoint>
        <bizLocation>
          <id>urn:epc:id:sgln:5012345.00000.0</id>
        </bizLocation>
      </AggregationEvent>
    </ObjectEvent>
    <eventTime>2015-03-15T10:11:00.000+01:00</eventTime>
  <eventTimeZoneOffset>+01:00</eventTimeZoneOffset>

```

```

    <epcList>
      <epc>urn:epc:id:sscc:5012345.0678901234</epc>
    </epcList>
    <action>OBSERVE</action>
    <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
    <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
    <readPoint>
      <id>urn:epc:id:sgln:5012345.00000.124</id>
    </readPoint>
  </ObjectEvent>
  <AggregationEvent>
    <eventTime>2015-03-17T15:00:00.000Z</eventTime>
    <eventTimeZoneOffset>+00:00</eventTimeZoneOffset>
    <parentID>urn:epc:id:sscc:5012345.0678901234</parentID>
    <childEPCs>
      <epc>urn:epc:id:sgtin:5012345.177777.1001</epc>
      <epc>urn:epc:id:sgtin:5012345.177777.1002</epc>
      <epc>urn:epc:id:sgtin:5012345.177777.1003</epc>
      <epc>urn:epc:id:sgtin:5012345.177777.1004</epc>
      <epc>urn:epc:id:sgtin:5012345.177777.1005</epc>
    </childEPCs>
    <action>OBSERVE</action>
    <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
    <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
    <readPoint>
      <id>urn:epc:id:sgln:5099999.00000.789</id>
    </readPoint>
    <bizLocation>
      <id>urn:epc:id:sgln:5099999.00000.0</id>
    </bizLocation>
  </AggregationEvent>
  <AggregationEvent>
    <eventTime>2015-03-17T15:10:00.000Z</eventTime>
    <eventTimeZoneOffset>+00:00</eventTimeZoneOffset>
    <parentID>urn:epc:id:sscc:5012345.0678901234</parentID>
    <childEPCs>
      <epc>urn:epc:id:sgtin:5012345.177777.1003</epc>
      <epc>urn:epc:id:sgtin:5012345.177777.1005</epc>
    </childEPCs>
    <action>DELETE</action>
    <bizStep>urn:epcglobal:cbv:bizstep:unpacking</bizStep>
    <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
    <readPoint>
      <id>urn:epc:id:sgln:5099999.00000.678</id>
    </readPoint>
    <bizLocation>
      <id>urn:epc:id:sgln:5099999.00000.0</id>
    </bizLocation>
  </AggregationEvent>
  <AggregationEvent>
    <eventTime>2015-03-17T15:25:00.000Z</eventTime>
    <eventTimeZoneOffset>+00:00</eventTimeZoneOffset>
    <parentID>urn:epc:id:sscc:5012345.0678901234</parentID>
    <childEPCs />
    <action>DELETE</action>
    <bizStep>urn:epcglobal:cbv:bizstep:unpacking</bizStep>
    <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
    <readPoint>
      <id>urn:epc:id:sgln:5099999.00000.678</id>
    </readPoint>
    <bizLocation>
      <id>urn:epc:id:sgln:5099999.00000.0</id>
    </bizLocation>
  </AggregationEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.4 表5-4の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T10:06:18.000-04:00">
  <EPCISBody>
    <EventList>
      <AggregationEvent>
        <eventTime>2015-03-15T10:00:00.000+09:00</eventTime>
        <eventTimeZoneOffset>+09:00</eventTimeZoneOffset>
        <parentID>urn:epc:id:sgtin:0614141.111111.101</parentID>
        <childEPCs>
          <epc>urn:epc:id:sgtin:0614141.011111.1</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.2</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.3</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.4</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.5</epc>
        </childEPCs>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:packing</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.00000.100</id>
        </readPoint>
        <bizLocation>
          <id>urn:epc:id:sgln:0614141.00000.0</id>
        </bizLocation>
      </AggregationEvent>
      <AggregationEvent>
        <eventTime>2015-03-15T10:05:00.000+09:00</eventTime>
        <eventTimeZoneOffset>+09:00</eventTimeZoneOffset>
        <parentID>urn:epc:id:sgtin:0614141.111111.102</parentID>
        <childEPCs>
          <epc>urn:epc:id:sgtin:0614141.011111.6</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.7</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.8</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.9</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.10</epc>
        </childEPCs>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:packing</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.00000.100</id>
        </readPoint>
        <bizLocation>
          <id>urn:epc:id:sgln:0614141.00000.0</id>
        </bizLocation>
      </AggregationEvent>
      <AggregationEvent>
        <eventTime>2015-03-15T10:10:00.000+09:00</eventTime>
        <eventTimeZoneOffset>+09:00</eventTimeZoneOffset>
        <parentID>urn:epc:id:sgtin:0614141.111111.103</parentID>
        <childEPCs>
          <epc>urn:epc:id:sgtin:0614141.011111.11</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.12</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.13</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.14</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.15</epc>
        </childEPCs>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:packing</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
  
```

```

    <id>urn:epc:id:sgln:0614141.00000.100</id>
  </readPoint>
  <bizLocation>
    <id>urn:epc:id:sgln:0614141.00000.0</id>
  </bizLocation>
</AggregationEvent>
<AggregationEvent>
  <eventTime>2015-03-15T10:25:00.000+09:00</eventTime>
  <eventTimeZoneOffset>+09:00</eventTimeZoneOffset>
  <parentID>urn:epc:id:sgtin:0614141.211111.1001</parentID>
  <childEPCs>
    <epc>urn:epc:id:sgtin:0614141.111111.101</epc>
    <epc>urn:epc:id:sgtin:0614141.111111.102</epc>
    <epc>urn:epc:id:sgtin:0614141.111111.103</epc>
  </childEPCs>
  <action>ADD</action>
  <bizStep>urn:epcglobal:cbv:bizstep:packing</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.00000.100</id>
  </readPoint>
  <bizLocation>
    <id>urn:epc:id:sgln:0614141.00000.0</id>
  </bizLocation>
</AggregationEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.5 表5-6の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T10:24:01.152-04:00">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2015-03-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <epcList>
          <epc>urn:epc:id:sscc:0614141.0123456789</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.11111.0</id>
        </readPoint>
        <extension>
          <sourceList>
            <source
              type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00000.0</source>
            <source
              type="urn:epcglobal:cbv:sdt:possessing_party">urn:epc:id:sgln:0614141.00000.0</source>
            <source
              type="urn:epcglobal:cbv:sdt:location">urn:epc:id:sgln:0614141.11111.0</source>
          </sourceList>
          <destinationList>

```

```

    <destination
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:9999999.00000.0
    </destination>
    <destination
type="urn:epcglobal:cbv:sdt:possessing_party">urn:epc:id:sgln:5012345.00000.
0
    </destination>
    <destination
type="urn:epcglobal:cbv:sdt:location">urn:epc:id:sgln:5012345.11111.0
    </destination>
  </destinationList>
</extension>
</ObjectEvent>
<ObjectEvent>
  <eventTime>2015-03-17T10:00:00.000Z</eventTime>
  <eventTimeZoneOffset>+00:00</eventTimeZoneOffset>
  <epcList>
    <epc>urn:epc:id:sscc:0614141.0123456789</epc>
  </epcList>
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:5012345.11111.123</id>
  </readPoint>
  <bizLocation>
    <id>urn:epc:id:sgln:5012345.11111.0</id>
  </bizLocation>
  <extension>
    <sourceList>
      <source
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00000.0</s
ource>
      <source
type="urn:epcglobal:cbv:sdt:possessing_party">urn:epc:id:sgln:0614141.00000.
0</source>
      <source
type="urn:epcglobal:cbv:sdt:location">urn:epc:id:sgln:0614141.11111.0</sourc
e>
    </sourceList>
  </destinationList>
  <destination
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:9999999.00000.0
  </destination>
  <destination
type="urn:epcglobal:cbv:sdt:possessing_party">urn:epc:id:sgln:5012345.00000.
0
  </destination>
  <destination
type="urn:epcglobal:cbv:sdt:location">urn:epc:id:sgln:5012345.11111.0
  </destination>
</destinationList>
</extension>
</ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.6 表5-7の例のXML

```
<epcis:EPCISDocument
```

```

  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T10:35:38.116-04:00">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2015-07-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <epcList />
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:creating_class_instance</bizStep>
        <disposition>urn:epcglobal:cbv:disp:active</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.00000.100</id>
        </readPoint>
        <bizLocation>
          <id>urn:epc:id:sgln:0614141.00000.0</id>
        </bizLocation>
        <extension>
          <quantityList>
            <quantityElement>
              <epcClass>urn:epc:class:lgtin:0614141.011111.12</epcClass>
              <quantity>20</quantity>
            </quantityElement>
          </quantityList>
        </extension>
      </ObjectEvent>
      <ObjectEvent>
        <eventTime>2015-07-16T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <epcList />
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.00000.200</id>
        </readPoint>
        <extension>
          <quantityList>
            <quantityElement>
              <epcClass>urn:epc:class:lgtin:0614141.011111.12</epcClass>
              <quantity>10</quantity>
            </quantityElement>
          </quantityList>
        </extension>
      </ObjectEvent>
      <ObjectEvent>
        <eventTime>2015-07-17T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <epcList />
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.00000.200</id>
        </readPoint>
        <extension>

```

```

    <quantityList>
  <quantityElement>
    <epcClass>urn:epc:class:lgtin:0614141.011111.12</epcClass>
    <quantity>10</quantity>
  </quantityElement>
</quantityList>
</extension>
</ObjectEvent>
<ObjectEvent>
  <eventTime>2015-07-25T10:00:00.000-04:00</eventTime>
  <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
  <epcList />
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:5012345.00000.111</id>
  </readPoint>
  <extension>
    <quantityList>
  <quantityElement>
    <epcClass>urn:epc:class:lgtin:0614141.011111.12</epcClass>
    <quantity>10</quantity>
  </quantityElement>
</quantityList>
</extension>
</ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.7 表5-8の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T10:45:30.662-04:00">
  <EPCISBody>
    <EventList>
      <AggregationEvent>
        <eventTime>2015-03-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <parentID>urn:epc:id:sscc:0614141.0123456789</parentID>
        <childEPCs />
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:packing</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.00000.100</id>
        </readPoint>
        <bizLocation>
          <id>urn:epc:id:sgln:0614141.00000.0</id>
        </bizLocation>
        <extension>
          <childQuantityList>
            <quantityElement>
              <epcClass>urn:epc:class:lgtin:0614141.011111.12</epcClass>
              <quantity>10</quantity>
            </quantityElement>
            <quantityElement>

```

```

    <epcClass>urn:epc:class:lgtn:0614141.022222.52</epcClass>
    <quantity>20</quantity>
  </quantityElement>
</childQuantityList>
</extension>
</AggregationEvent>
<ObjectEvent>
  <eventTime>2015-03-16T10:00:00.000-04:00</eventTime>
  <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
  <epcList>
    <epc>urn:epc:id:sscc:0614141.0123456789</epc>
  </epcList>
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.00000.200</id>
  </readPoint>
</ObjectEvent>
<ObjectEvent>
  <eventTime>2015-03-20T00:00:00.000Z</eventTime>
  <eventTimeZoneOffset>+00:00</eventTimeZoneOffset>
  <epcList>
    <epc>urn:epc:id:sscc:0614141.0123456789</epc>
  </epcList>
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:5012345.00000.300</id>
  </readPoint>
  <bizLocation>
    <id>urn:epc:id:sgln:5012345.00000.0</id>
  </bizLocation>
</ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.8 表5-10の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T10:45:30.662-04:00">
  <EPCISBody>
    <EventList>
      <AggregationEvent>
        <eventTime>2015-03-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <parentID>urn:epc:id:sscc:0614141.0123456789</parentID>
        <childEPCs>
          <epc>urn:epc:id:sgtin:0614141.011111.101</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.102</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.103</epc>
        </childEPCs>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:packing</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>

```

```

    <id>urn:epc:id:sgln:0614141.00000.100</id>
  </readPoint>
  <bizLocation>
    <id>urn:epc:id:sgln:0614141.00000.0</id>
  </bizLocation>
  <extension>
    <childQuantityList>
      <quantityElement>
        <epcClass>urn:epc:class:lgtin:0614141.022222.12</epcClass>
        <quantity>10</quantity>
      </quantityElement>
      <quantityElement>
        <epcClass>urn:epc:idpat:sgtin:0614141.033333.*</epcClass>
        <quantity>20</quantity>
      </quantityElement>
    </childQuantityList>
  </extension>
</AggregationEvent>
<ObjectEvent>
  <eventTime>2015-03-16T10:00:00.000-04:00</eventTime>
  <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
  <epcList>
    <epc>urn:epc:id:sscc:0614141.0123456789</epc>
  </epcList>
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.00000.200</id>
  </readPoint>
</ObjectEvent>
<ObjectEvent>
  <eventTime>2015-03-20T00:00:00.000Z</eventTime>
  <eventTimeZoneOffset>+00:00</eventTimeZoneOffset>
  <epcList>
    <epc>urn:epc:id:sscc:0614141.0123456789</epc>
  </epcList>
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:5012345.00000.300</id>
  </readPoint>
  <bizLocation>
    <id>urn:epc:id:sgln:5012345.00000.0</id>
  </bizLocation>
</ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.9 表5-14の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T11:14:25.411-04:00">
  <EPCISBody>
    <EventList>
      <extension>

```

```

<TransformationEvent>
  <eventTime>2015-03-15T00:00:00.000-04:00</eventTime>
  <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
  <inputEPCList>
    <epc>urn:epc:id:sgtin:0614141.011111.10</epc>
    <epc>urn:epc:id:sgtin:0614141.011111.45</epc>
    <epc>urn:epc:id:sgtin:0614141.011111.97</epc>
  </inputEPCList>
  <inputQuantityList>
    <quantityElement>
      <epcClass>urn:epc:class:lgtn:0614141.022222.12</epcClass>
      <quantity>10.0</quantity>
      <uom>LTR</uom>
    </quantityElement>
    <quantityElement>
      <epcClass>urn:epc:idpat:sgtin:0614141.033333.*</epcClass>
      <quantity>100</quantity>
    </quantityElement>
  </inputQuantityList>
  <outputQuantityList>
    <quantityElement>
      <epcClass>urn:epc:class:lgtn:0614141.044444.999</epcClass>
      <quantity>100</quantity>
    </quantityElement>
  </outputQuantityList>
  <bizStep>urn:epcglobal:cbv:bizstep:creating_class_instance</bizStep>
  <disposition>urn:epcglobal:cbv:disp:active</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.00000.100</id>
  </readPoint>
  <bizLocation>
    <id>urn:epc:id:sgln:0614141.00000.0</id>
  </bizLocation>
</TransformationEvent>
</extension>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.10 表5-15の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T11:14:25.411-04:00">
  <EPCISBody>
    <EventList>
      <extension>
        <TransformationEvent>
          <eventTime>2015-03-15T00:00:00.000-04:00</eventTime>
          <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
          <inputEPCList>
            <epc>urn:epc:id:sgtin:0614141.011111.10</epc>
            <epc>urn:epc:id:sgtin:0614141.011111.45</epc>
          </inputEPCList>
          <inputQuantityList>
            <quantityElement>
              <epcClass>urn:epc:class:lgtn:0614141.022222.12</epcClass>
              <quantity>5.0</quantity>
            </quantityElement>
          </inputQuantityList>
        </TransformationEvent>
      </extension>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>

```

```

    <uom>LTR</uom>
  </quantityElement>
</quantityElement>
  <epcClass>urn:epc:idpat:sgtin:0614141.033333.*</epcClass>
  <quantity>40</quantity>
</quantityElement>
</inputQuantityList>
<transformationID>urn:epcglobal:cbv:xform:0614141000005:123
</transformationID>
<bizStep>urn:epcglobal:cbv:bizstep:creating_class_instance
</bizStep>
<disposition>urn:epcglobal:cbv:disp:active</disposition>
<readPoint>
  <id>urn:epc:id:sgln:0614141.00000.100</id>
</readPoint>
<bizLocation>
  <id>urn:epc:id:sgln:0614141.00000.0</id>
</bizLocation>
</TransformationEvent>
</extension>
<extension>
  <TransformationEvent>
    <eventTime>2015-03-15T01:00:00.000-04:00</eventTime>
    <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
    <outputQuantityList>
      <quantityElement>
        <epcClass>urn:epc:class:lgtn:0614141.044444.999</epcClass>
        <quantity>30</quantity>
      </quantityElement>
    </outputQuantityList>
    <transformationID>urn:epcglobal:cbv:xform:0614141000005:123
    </transformationID>
    <bizStep>urn:epcglobal:cbv:bizstep:creating_class_instance
    </bizStep>
    <disposition>urn:epcglobal:cbv:disp:active</disposition>
    <readPoint>
      <id>urn:epc:id:sgln:0614141.00000.100</id>
    </readPoint>
    <bizLocation>
      <id>urn:epc:id:sgln:0614141.00000.0</id>
    </bizLocation>
  </TransformationEvent>
</extension>
<ObjectEvent>
  <eventTime>2015-03-15T02:00:00.000-04:00</eventTime>
  <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
  <epcList />
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.00000.200</id>
  </readPoint>
  <extension>
    <quantityList>
      <quantityElement>
        <epcClass>urn:epc:class:lgtn:0614141.044444.999</epcClass>
        <quantity>30</quantity>
      </quantityElement>
    </quantityList>
  </extension>

```

```

    </extension>
  </ObjectEvent>
</extension>
  <TransformationEvent>
    <eventTime>2015-03-15T03:00:00.000-04:00</eventTime>
    <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
    <inputEPCList>
      <epc>urn:epc:id:sgtin:0614141.011111.97</epc>
    </inputEPCList>
    <inputQuantityList>
      <quantityElement>
        <epcClass>urn:epc:class:lgtin:0614141.022222.12</epcClass>
        <quantity>5.0</quantity>
        <uom>LTR</uom>
      </quantityElement>
      <quantityElement>
        <epcClass>urn:epc:idpat:sgtin:0614141.033333.*</epcClass>
        <quantity>60</quantity>
      </quantityElement>
    </inputQuantityList>
    <outputQuantityList>
      <quantityElement>
        <epcClass>urn:epc:class:lgtin:0614141.044444.999</epcClass>
        <quantity>70</quantity>
      </quantityElement>
    </outputQuantityList>
    <transformationID>urn:epcglobal:cbv:xform:0614141000005:123
  </transformationID>
  <bizStep>urn:epcglobal:cbv:bizstep:creating_class_instance
</bizStep>
  <disposition>urn:epcglobal:cbv:disp:active</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.00000.100</id>
  </readPoint>
  <bizLocation>
    <id>urn:epc:id:sgln:0614141.00000.0</id>
  </bizLocation>
</TransformationEvent>
</extension>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.11 表5-16の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.1" creationDate="2015-04-14T11:32:14.865-
  04:00">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2015-07-15T10:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <epcList>
          <epc>urn:epc:id:sgcn:0614141.11111.12345</epc>
        </epcList>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:commissioning</bizStep>
        <disposition>urn:epcglobal:cbv:disp:active</disposition>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>

```

```

    <readPoint>
      <id>urn:epc:id:sgln:0614141.00000.100</id>
    </readPoint>
  </ObjectEvent>
  <ObjectEvent>
    <eventTime>2015-07-16T10:00:00.000+02:00</eventTime>
    <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
    <epcList>
      <epc>urn:epc:id:sgcn:0614141.11111.12345</epc>
    </epcList>
    <action>DELETE</action>
    <bizStep>urn:epcglobal:cbv:bizstep:decommissioning</bizStep>
    <disposition>urn:epcglobal:cbv:disp:inactive</disposition>
    <readPoint>
      <id>urn:epc:id:sgln:0614141.00000.1002</id>
    </readPoint>
  </ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.12 表5-21の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.2" creationDate="2015-04-14T10:24:01.152-
  04:00">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2016-07-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <epcList>
          <epc>urn:epc:id:sgtin:0614141.011111.101</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.102</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.103</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.11111.0</id>
        </readPoint>
        <extension>
          <sourceList>
            <source
              type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00000.0</s
              ource>
          </sourceList>
          <destinationList>
            <destination
              type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:9999999.00000.0</d
              estination>
          </destinationList>
        </extension>
      </ObjectEvent>
      <ObjectEvent>
        <eventTime>2016-07-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <epcList>

```

```

    <epc>urn:epc:id:sgtin:0614141.011111.104</epc>
  </epcList>
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.11111.0</id>
  </readPoint>
  <extension>
    <sourceList>
      <source
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00000.0</s
ource>
    </sourceList>
    <destinationList>
      <destination
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:9999999.00000.0</d
estination>
    </destinationList>
  </extension>
</ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.13 表5-22の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.2" creationDate="2015-04-14T10:24:01.152-
04:00">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2016-07-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <epcList>
          <epc>urn:epc:id:sgtin:0614141.011111.101</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.102</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.103</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.11111.0</id>
        </readPoint>
        <extension>
          <sourceList>
            <source
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00000.0</s
ource>
          </sourceList>
          <destinationList>
            <destination
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:9999999.00000.0</d
estination>
          </destinationList>
        </extension>
      </ObjectEvent>

```

```

<ObjectEvent>
  <eventTime>2016-07-18T14:00:00.000-04:00</eventTime>
  <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
  <epcList>
    <epc>urn:epc:id:sgtin:0614141.011111.103</epc>
  </epcList>
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:void_shipping</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.11111.0</id>
  </readPoint>
  <bizLocation>
    <id>urn:epc:id:sgln:0614141.11111.0</id>
  </bizLocation>
  <extension>
    <sourceList>
      <source
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00000.0</s
ource>
      </sourceList>
      <destinationList>
        <destination
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:9999999.00000.0</d
estination>
        </destinationList>
      </extension>
    </ObjectEvent>
  </EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.14 表5-23の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.2" creationDate="2015-04-14T10:24:01.152-04:00">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2016-07-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <epcList>
          <epc>urn:epc:id:sgtin:0614141.011111.101</epc>
        </epcList>
        <action>DELETE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:destroying</bizStep>
        <disposition>urn:epcglobal:cbv:disp:destroyed</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.11111.0</id>
        </readPoint>
      </ObjectEvent>
      <ObjectEvent>
        <eventTime>2016-07-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <baseExtension>
          <errorDeclaration>
            <declarationTime>2016-07-17T14:00:00.000-04:00</declarationTime>
            <reason>urn:epcglobal:cbv:er:did_not_occur</reason>
          </errorDeclaration>
        </baseExtension>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>

```

```

    </errorDeclaration>
  </baseExtension>
  <epcList>
    <epc>urn:epc:id:sgtin:0614141.011111.101</epc>
  </epcList>
  <action>DELETE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:destroying</bizStep>
  <disposition>urn:epcglobal:cbv:disp:destroyed</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.11111.0</id>
  </readPoint>
</ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

A.15 表5-24の例のXML

```

<epcis:EPCISDocument
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  schemaVersion="1.2" creationDate="2015-04-14T10:24:01.152-04:00">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2016-07-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <epcList>
          <epc>urn:epc:id:sgtin:0614141.011111.101</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.102</epc>
          <epc>urn:epc:id:sgtin:0614141.011111.103</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.11111.0</id>
        </readPoint>
        <bizTransactionList>
          <bizTransaction
            type="urn:epcglobal:cbv:bt:po">urn:epcglobal:cbv:bt:0614141000005:456</bizT
ransaction>
          </bizTransactionList>
          <extension>
            <sourceList>
              <source
                type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00000.0</s
ource>
            </sourceList>
            <destinationList>
              <destination
                type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:9999999.00000.0</d
estination>
            </destinationList>
          </extension>
        </ObjectEvent>
      <ObjectEvent>
        <eventTime>2016-07-15T10:00:00.000-04:00</eventTime>
        <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
        <baseExtension>

```

```

    <errorDeclaration>
      <declarationTime>2016-07-17T13:00:00.000-04:00</declarationTime>
      <reason>urn:epcglobal:cbv:er:incorrect_data</reason>
      <correctiveEventIDs>
        <correctiveEventID>urn:uuid:6926712e-599f-4c4e-b6e9-8dd888c906bd
      </correctiveEventID>
      </correctiveEventIDs>
    </errorDeclaration>
  </baseExtension>
  <epcList>
    <epc>urn:epc:id:sgtin:0614141.011111.101</epc>
    <epc>urn:epc:id:sgtin:0614141.011111.102</epc>
    <epc>urn:epc:id:sgtin:0614141.011111.103</epc>
  </epcList>
  <action>OBSERVE</action>
  <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
  <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
  <readPoint>
    <id>urn:epc:id:sgln:0614141.11111.0</id>
  </readPoint>
  <bizTransactionList>
    <bizTransaction
type="urn:epcglobal:cbv:btt:po">urn:epcglobal:cbv:bt:0614141000005:456</bizT
ransaction>
    </bizTransactionList>
  <extension>
    <sourceList>
      <source
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00000.0</s
ource>
      </sourceList>
      <destinationList>
        <destination
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:9999999.00000.0</d
estination>
        </destinationList>
      </extension>
    </ObjectEvent>
  <ObjectEvent>
    <eventTime>2016-07-15T10:00:00.000-04:00</eventTime>
    <eventTimeZoneOffset>-04:00</eventTimeZoneOffset>
    <baseExtension>
      <eventID>urn:uuid:6926712e-599f-4c4e-b6e9-8dd888c906bd</eventID>
    </baseExtension>
    <epcList>
      <epc>urn:epc:id:sgtin:0614141.011111.101</epc>
      <epc>urn:epc:id:sgtin:0614141.011111.102</epc>
      <epc>urn:epc:id:sgtin:0614141.011111.103</epc>
    </epcList>
    <action>OBSERVE</action>
    <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
    <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
    <readPoint>
      <id>urn:epc:id:sgln:0614141.11111.0</id>
    </readPoint>
    <bizTransactionList>
      <bizTransaction
type="urn:epcglobal:cbv:btt:po">urn:epcglobal:cbv:bt:0614141000005:123</bizT
ransaction>
      </bizTransactionList>
  </ObjectEvent>

```

```

    <extension>
      <sourceList>
        <source
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00000.0</s
ource>
          </sourceList>
        <destinationList>
          <destination
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:9999999.00000.0</d
estination>
          </destinationList>
        </extension>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>

```

9 過去バージョンの協賛者

以下は、本ガイドラインのバージョン1.0の協賛者である。

氏名	所属
Andrew Kennedy、ワーキンググループ共同議長	FoodLogiQ
Michele Southall、ワーキンググループ共同議長	GS1 US
Gena Morgan、ワーキンググループ取りまとめ役	GS1 本部
Ken Traub, Editor	Ken Traub Consulting
Kerry Angelo	GS1 本部
Paul Arguin	r-pac international
Robert Besford	GS1 UK
Karla Biggs-Gregory	Oracle
Karolin Catela	GS1 Sweden
Robert Celeste	GS1 US
Mario Chavez	GS1 Guatemala
Hussam El-Leithy	GS1 US
Heinz Graf	GS1 Switzerland
Tany Hui	GS1 Hong Kong
Jianhua Jia	GS1 China
Peter Jonsson	GS1 Sweden
Janice Kite	GS1 本部
Jens Kungl	METRO Group
Jean-Luc Leblond	GS1 France
Paul Lothian	Tyson
Fargeas Ludovic	Courbon
Noriyuki Mama	GS1 Japan
Kevan McKenzie	McKesson
Reiko Moritani	GS1 Japan

氏名	所属
Alice Mukaru	GS1 Sweden
Falk Nieder	European EPC Competence Center GmbH (EECC)
Juan Ochoa	GS1 Columbia
Ted Osinski	MET Laboratories
James Perng	GS1 Taiwan
Zhuoqiong Qin	GS1 China
Craig Alan Repec	GS1 本部
Chris Roberts	GlaxoSmithKline
Thomas Rumbach	SAP AG
John Ryu	GS1 本部
Chuck Sailor	Frequentz
Michael Sarachman	GS1 本部
Roxana Saravia	GS1 Argentina
Hans Peter Scheidt	GS1 Germany
Michael Smith	Merck & Co.
Steve Tadevich	McKesson
Hristo Todorov	Axway
Ralph Troeger	GS1 Germany
Geir Vevle	HRAFN
Elizabeth Waldorf	TraceLink
Ruoyun Yan	GS1 China
Tony Zhang	FSE
Mike Zupec	Abbvie

