

システム間インタフェース試論 — バリエーションの源泉 —

児玉公信[†]

複数の人間活動システムが、それぞれの仕事を独立に実行しながら、一方で連携する状況は、ビジネスのシステムではよく見られる。しかし、独立したシステムはそれぞれ微妙に異なる意味論を持っており、システム間のコミュニケーションにおいて誤解やムダを発生させる原因となっている。また、こうしたシステム間連携が、互いのビジネスプロセスにコミットすればするほど、意味論、プロトコル、媒体の多様化が必然となり、コミュニケーション形態の多様性は爆発する。これは、EDI (Electronic Data Interchange) の決定版がまだ存在しないことを見ても分かる。

本報告では、こうしたビジネスシステム間のコミュニケーションが本来、共約不可能性を含意することを前提としつつ、その多様性をどのように制御できるかについて、概念モデリングを援用して試論する。

A Tentative Assumption on Inter-Systems Co-working: the Source of Variation

Kiminobu Kodama[†]

The situation in which several Human Activity Systems cooperate while executing a work by themselves is frequently seen in business systems. However, each system has slightly different semantics, and it causes the misunderstanding and uselessness in the communications among the systems. Further, as such systems commit more tightly with each business processes, the diversification of semantics, protocols, and medium becomes more inevitable, and the diversity in the communications explodes. The fact that the definitive edition of EDI (Electronic Data Interchange) has not been present proves it.

In this report, assuming that the communication among such business systems implies the incommensurability, we discuss tentatively how the diversity can be controlled from the viewpoint of the conceptual modeling.

1. はじめに

本報告では“システム”という語を、情報システムの略ではなく、システム理論で言う本来の“システム”の意味で用いる。システム理論は、1950年代中頃に始まった複雑な対象をより合理的に扱うための考え方の体系である¹⁾²⁾。それは、システムの構成要素が相互作用することによって、システム全体のパフォーマンスが個々の要素のパフォーマンスの合計以上になるという“創発”や、構成要素間の相互作用の効率化を図るために自己組織化あるいはシステムの階層化がなされるといった性質で特徴づけられる。システムの中でもその振る舞いが非常に複雑で、将来の状態が論理的に予測できないものを“ソフトシステム”と呼び、複雑ではあっても予測可能な“ハードシステム”と区別する。人を構成要素として含むシステムはソフトシステムの典型であり、Checklandはこれを人間活動システムと呼んだ³⁾。社会システム、ビジネスシステム、情報システムはその例である。

複数の人間活動システムが、それぞれの仕事を独立に実行しながら、並行的に連携する状況は、ビジネスシーンにおいてよく見られる。人間活動システムは、それぞれにその発生、発展の歴史、知識の蓄積、世界観、ダイナミズムをもち、常に互いに異なるセマンティクスを持つ。これが、システム間のコミュニケーションにおいて誤解やムダを発生させる一因である。また、これらの人間活動システムはさらに上位の人間活動システムの構成要素(サブシステム)であると見ることもでき、上位システム自体も、歴史や知識の蓄積とダイナミズムをもち、それがサブシステムを規定する関係にある。

本報告では、こうしたシステム理論に沿って、ビジネスシステム間のコミュニケーションが本質的に異なるパラダイムの対話であって、共約不可能性⁴⁾を含意することを前提に、その多様性の源泉と制御について、概念モデリングを援用して試論する。

2. システム間連携

ビジネスにおける企業間の連携、いわゆるB to B (Business to Business)の取引は、受発注の例を挙げるまでもなく、古くから行われてきた。その連携の形が、近代産業の発展とともに、より緊密になってきたと感じる。たとえば、トヨタのジャストインタイムによる納入や、デザイン・インと呼ばれる設計の早期段階への納入業者の参加などである。この傾向は、自社のビジネスプロセスと顧客のビジネスプロセスとがより直接的に結びついてきたことを意味する。

本来、企業というシステムは、ビジネス領域が近いとしても、それぞれに歴史や価値観が異なり、それゆえ使用する言葉の意味、ビジネスプロセス、技術(technology)などが互いに異なっている。それが、連携の要請に基づいて、顧客のビジネスプロセ

[†] 情報システム総研
Information Systems Institute, Ltd.

スにコミットするなかで、多くは力関係に基づいてセマンティクスが形成されてきた。これが個別の連携における合意であったり、業界標準であったりする。

2.1 人による媒介

業種や業態によって異なるが、1980年代の前半までは、企業間のビジネス連携は人が負っていた。すなわち、営業担当者が顧客ごとに異なる意味論やビジネスプロセスを理解し、自社と顧客との差を調整していた。こうした顧客ごとの違いが大きく問題にされることはなく、むしろ、それを調整することこそが営業担当者の存在意義であった。その後、それぞれの企業で業務処理の情報システムが開発されても、それはその企業の内部システムでしかなく、外部との連携は、あいかわらず人と人が電話や文書によるコミュニケーションによって成り立っていた。これは、人を構成要素とする柔軟な協調のための非明示的なシステムであった（図1a）。

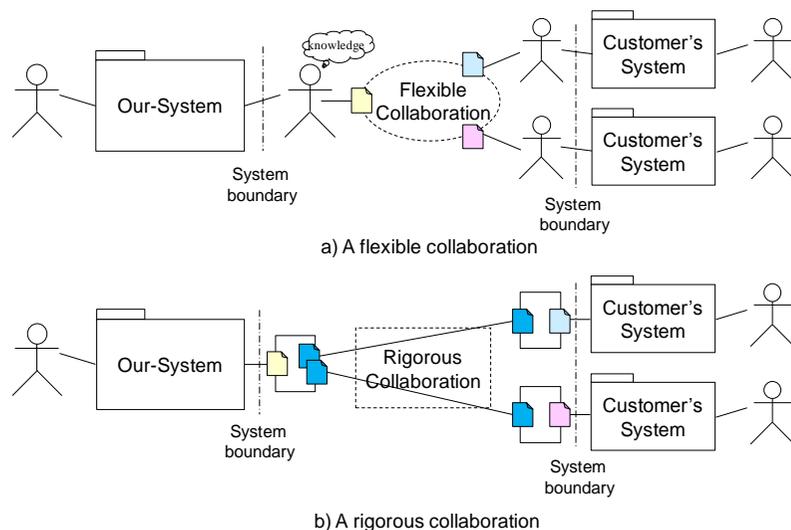


図1 柔軟な協調から厳密な協調へ

2.2 システム間連携

1980年代の後半から徐々に、コンピュータどうしをVANなどの通信ネットワークで直接つないで受発注プロセスを橋渡しするEDI (Electronic Data Interchange) が使われるようになってきた。当時の受発注EDIは、政府主導で品目コードや機関コードなどの共通辞書を定め、標準メッセージと標準プロトコルを制定し、政策的に誘導した

ものであった。大企業の多くはその時点で独自の受発注システムを構築していたため、このEDIを使うには、自社形式と標準形式の間で双方向に“トランスレート”する必要があった。その結果、標準コード、標準辞書、標準プロトコルで厳密に定められたやりとりしかできなかった。つまり、直結することが、かえって連携上の制約を強めることになった（図1b）。しかし、標準形式に単純に翻訳できないような独自のやりとりがビジネス上の必然である場合もあって、当時は、その窮屈さがEDIの普及を妨げていた一つの要因であったように思う。

近年は、インターネット、Webアプリケーション、XMLなどの技術の進展によりプロトコルの自由度が増す一方で、業界ごとの標準化も進み、国際取引も増えて、大企業でのEDIの普及率は上がってきた⁵⁾。しかし、その大企業の定めた標準形式に、力関係から合わせざるを得ない中小企業では、企業内の情報化の遅れもあって、普及率は伸び悩んでいる。

2.2.1 内部連携と外部連携

受発注プロセスは、明確にEDIの対象であると認識されているし、輸出入や金融決済業務などもEDIの拡張と考えられている⁶⁾。しかし、保守や検査などのサービス要求の受発注では、取引の形態が多様すぎてEDIの対象と認識されることは少ない。それは、EDIがSCM (Supply Chain Management) と概念拡張されても変わらない。しかし、システム連携の目的がビジネスプロセスの連携の効率と正確性の向上を目指すのであれば、サービス要求もEDIの対象となるべきである。

図2は、児玉・水野⁷⁾が提案した製造業におけるビジネス連携のモデルである。ここでは、製造業の技術プロセスは大きく設計、製造、保全（保守サービス）と呼ぶ三つの局面に分けられるとする。企業内部では三つの局面の知識業務が知識の展開や共有のために情報が連携される。この連携を促進するための仕組みを内部連携メカニズムと呼ぶ。企業外部とは三つの局面の実施段階で、連携相手の三つの局面のどれかの実施段階と接続する。この接続は、外部連携メカニズムと呼ぶ一種のフィルタを通して制御される^{a)}。

2.2.2 相手のビジネスプロセスにコミットすること

相手のビジネスプロセスにコミットする例として、組立型の製造業の一部で行われているJust-in-time納入を取り上げる。これは部品などを数量、時刻、場所を厳密に指定して納入させる制度である。これによって、発注者が在庫を持たないことで経営上は有効だが、この制度が成り立つには、発注者の製造実施が計画どおりに進められるほど製造力が備わっていることと、受注者がそれに合意するほど発注者に対する忠誠心が強いことなどの厳しい条件を満たしている必要がある。この制度は事実上、発注者の製造プロセスが、物流を介して受注者にはみ出していることを示している。これ

a EDIはむしろ、この外部連携のあり方の一つである。

は、図2左下にある製造実施どうしが連結しているイメージである。この図では、発注者と受注者が唯一のペアとして書かれているが、通常、1人の発注者は複数の受注者と連携する（供給源の多重化）。一方、これほど緊密な連携が標準語の使用を強いる共同のEDIシステムで行われることは少なく、発注者主導で独自のコード（概念構造と意味論）、記号（外部表現あるいはビュー）、媒体（プロトコル、対話制御）などの「対話インタフェース」に基づく連携システムが構築・運営され、受注者はこのシステムに囲い込まれていく。

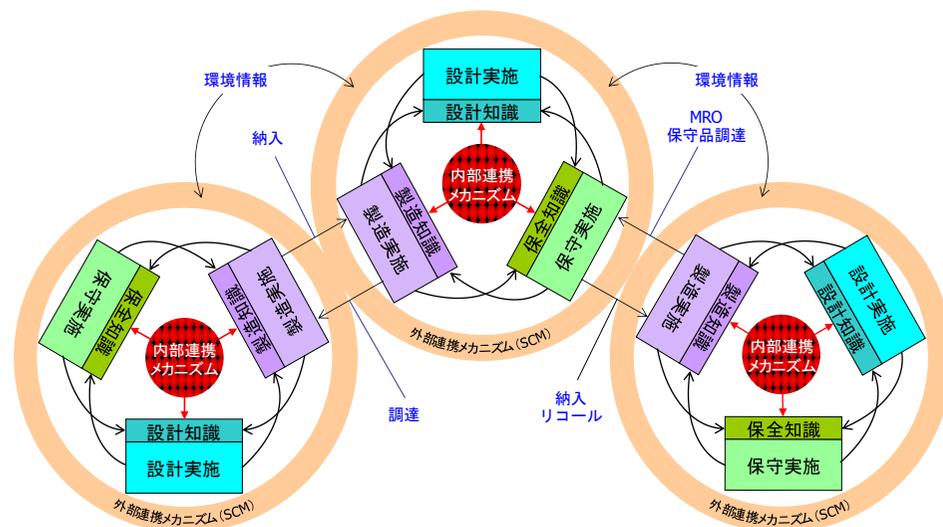


図2 製造業におけるビジネス連携モデル⁷⁾

しかし、逆に多くの受注者は複数の発注者と連携する。このとき、受注者は対話インタフェースを発注者ごとに用意する必要があり、媒体変換、記号変換などのトランスレーションは煩雑なものとなる。このため、緊密な連携先以外とは、コードが類似している業界EDIシステムに参加するようになってきた⁴⁾。

一方、サービスビジネスでは1人の受注者は多数の発注者と緊密に連携する。顧客のビジネスプロセスにコミットすることがサービスの本質だからだ。その分、対処すべき対話インタフェースは膨大になり、その管理もきわめて煩雑なものとなる。しかし、業界EDIシステムに参加することはない。それは受注者の命取りであり、むしろ発注者の対話インタフェースに個別に対応していくことこそが、逆の顧客の囲い込みになると考えられているからである。

3. 外部連携メカニズムの構想

このような対話インタフェースのバリエーションを合理的に吸収する外部連携メカニズムを構想する。これは外部のさまざまな対話インタフェースから内部の標準インタフェースにトランスレートし（あるいはその逆も）、異なるビジネスプロセスを連携させて協働（co-work）させる仕組みである。ここではそれぞれのビジネスプロセスは非同期に動く。このため外部連携メカニズムには、注文が正しく処理され、結果が報告されたことを追跡する機能も必要となる。

ここでは簡単のために、注文書が与えられ、それに応じて行ったサービスの報告書を返す一連のやりとりを扱うことにする。注文書は連携先ごとに異なる対話インタフェースを通して受け取られる。注文書は一旦、標準の内部形式（XMLファイル）に変換され、サービスが実施された後、報告書が標準の内部形式（XMLファイル）で作られる。報告書も連携先ごとに異なる対話インタフェースに逆変換して返す。その中には、注文書にある発注番号などの連携先が必要とする情報を盛り込む必要がある。

3.1 外部知識

対話インタフェースは、連携先ごとに異なるだけでなく時間とともに変化（バージョンアップ）する。また、内部の標準インタフェースも時間とともに変化するので、内部-外部のインタフェースを一律の手続きで変換することは難しい。まず、この変換をルールとして規定することを考える。

3.1.1 Translation ルール

Translation ルールは、類似の個別の変換手続きを、変数を取り出すことで関数化したものである。これを連携先が使用する多数の Translation ルールとそれぞれの変数値をセットで持つ（図3）。変換がこのような関数で扱えないほど個性が高い場合、連携先ごとにその変換ロジックをデータとして記述する。これらの処置によって、連携先ごとのバリエーションを一か所に集中して管理できる。

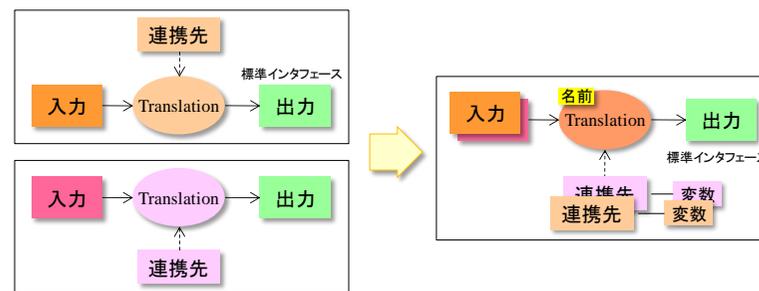


図3 関数化と連携先の関連づけ

3.1.2 Mapping ルール

Mapping ルールには、Translation ルールをどの入力フィールドに対して適用し、変換結果をどの出力フィールドに格納するかを記述する。

3.1.3 Validation ルール

注文書の受付において、入力データの妥当性チェック (validation) は必須の機能である。これもルールとして記述し、一か所で管理したいが、未検討である。

3.2 概念モデル

外部連携メカニズムの主要なオブジェクトの構造を、概念レベルのクラス図 (概念モデル) で表わす。

3.2.1 注文書のトランスレータ

図 4 は、注文書の外部形式を内部の標準形式 (XML) に Translation するインタフェース機能の概念モデルである。このモデルは、ファイル、OCR、端末からのメッセージなどのさまざまな媒体で受け取られた文字列を指示された位置で区切る、あるいは区切り文字を走査して切れ目を見つけて取り出して、Mapping ルールで指定された内部形式の XML のタグにセットすることを述べている。フィールドのうちいくつかは必須であり、他のフィールドの Mapping の変換内容に影響を及ぼすことがある。Mapping ルールが連携先によって異なることの表現については、次項で述べる。モデル中の「ドメイン」とは、アプリケーションのドメイン層を指し、内部標準形式のデータがオブジェクトとして取り込まれることを示す。

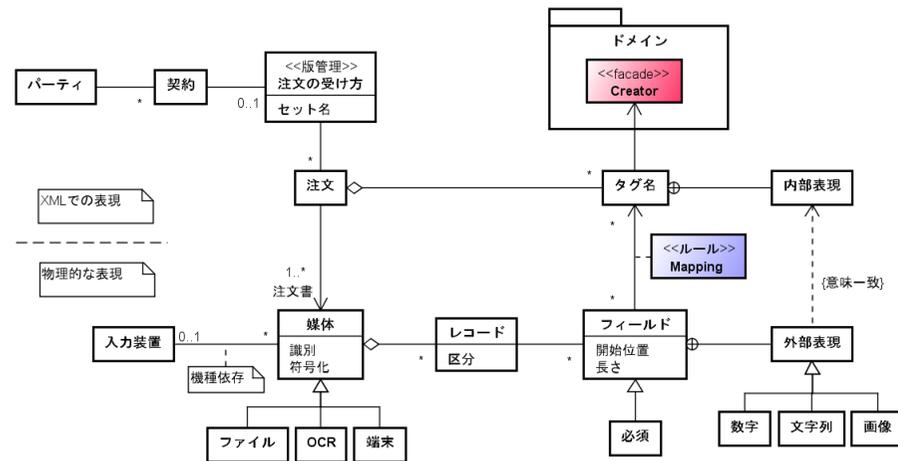


図 4 注文書のトランスレータの概念モデル

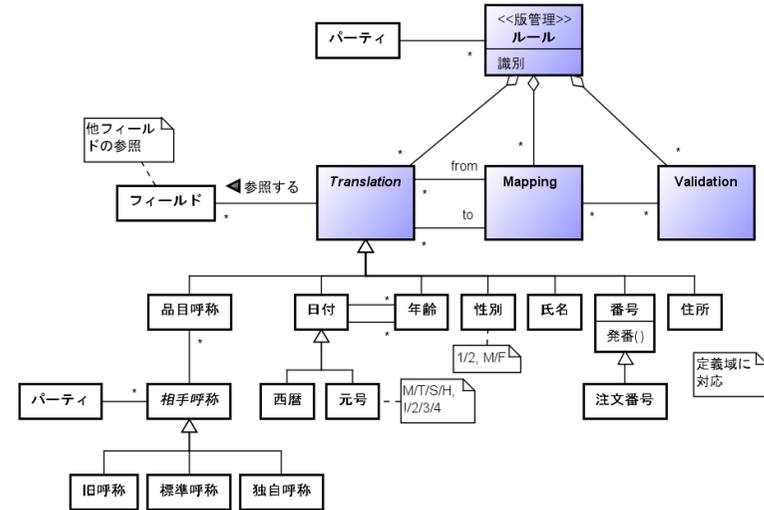


図 5 ルールの表現方法

3.2.2 ルールの表現

図 5 は、図 4 のステレオタイプ《ルール》で表した外部知識の本体の概念モデルである。モデル上は連携先を「パーティ」と表現している。ルールは、Translation, Mapping, Validation の 3 つの下位ルールからなる。このうち、Translation のルールを詳細に示している。Translation ルールは Mapping ルールから必要に応じて呼び出され、所定の変換をして返す。報告書処理の変換も同じルールを逆に用いて変換できる。

Translation ルール自体は抽象クラスで、その実体をサブクラスとして持つ。品目呼称は、連携先ごとに設定されている呼称あるいは業界標準の呼称と、内部の標準呼称の対応関係を持つ。日付の変換ルールはさまざまなフィールドで使われる。元号は内部的には西暦で持つが、外部表現はいくつかのバリエーションがあるので、それに対応した関数を用意する。年齢は生年月日と注文日付から導出されるが、年齢が明示されている場合は、生年月日との整合性をチェックする。性別の表現にも、1か2、あるいはMかFかなどのバリエーションがある。住所の表現も、地番の表し方などにバリエーションがある。番号は発番機能をもつ。

3.2.3 報告書のトランスレータ

図 6 は、報告書の内部の標準形式を外部形式にトランスレートする機能の概念モデルである。報告書は、帳票のほかに HTML や XML フォーマットなどきわめて多様だ

が、ここで扱う外部形式は帳票のみに限定する。報告書のトランスレータは、内部形式のXMLのタグを基に、Mappingルールで指定された帳票上の位置にTranslationルールで指定された形式にして配置する。品目呼称などは、内部形式から連携先ごとの呼称に読み替えられる。フィールドによっては、グラフ表示に変換することも考えられる。

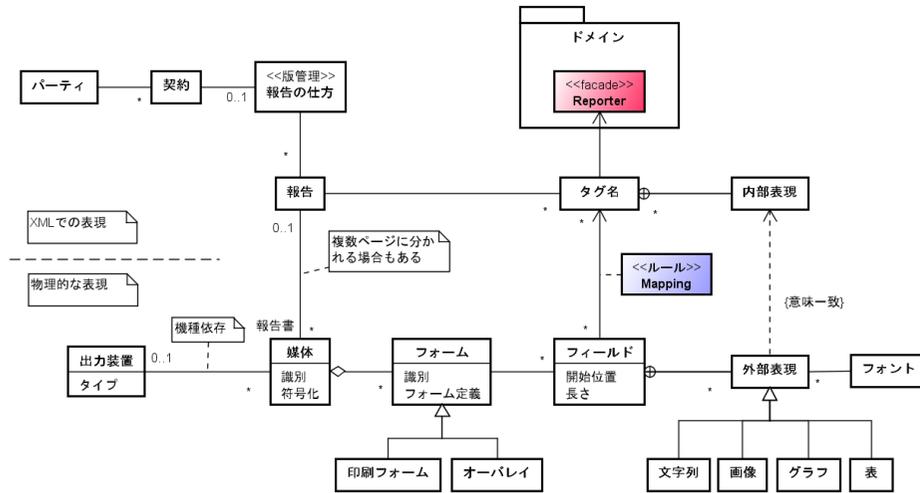


図6 報告書のトランスレータの概念モデル

3.3 外部連携メカニズムの実装

現在、注文書トランスレータのMappingルールとTranslationルールの試作をしている。試作の範囲では、構想どおり動作している。図5のルールのモデルは、試作結果を反映している。試作の評価については、あらためて報告したい。

4. バリエーションのコントロール

連携先ごとのバリエーションに対する従来の対処方法は、オペレータ向け操作マニュアルへの注意書き、プログラムへの個別ロジックのハードコーディング、発注者マスタファイル上に多数のオプションフラグを設定するなど、ルールの定義を散在させていた。このような状況では、対話インタフェースの変更要求に適時に追従することは困難であった。これを、概念モデルで表したように、Mapping, Translation, Validationの3種のルールに区分し、それぞれを共有可能な部品にすることで、集中管理しやす

くなり、変更要求に対してもすばやく対応できるようになる。

しかし、これは受け身の対応である。バリエーションに対応することの価値が逆の顧客の囲い込みにあるとすれば、この価値を積極的に売っていく方法も考えてみる。

受注者から見てバリエーションが発生する原因は、受注者がコミットする連携先のビジネスプロセスがそれぞれ異なるからであった。それでは、なぜビジネスプロセスが異なるのか。それは連携先企業の歴史や、扱っている商品、顧客が異なっているためである。では、企業の歴史は無理でも、扱っている商品のタイプや顧客層などで、連携先のビジネスパターンを整理できないか。一方で、ルールのパターンも整理しておき、ビジネスパターンとルールのパターンの対応関係を付けていくことができないか。

この構想の概念モデルが図7である。ビジネス変数は、商品タイプ、客層、連携タイプ(Just-in-time納入、「かんぱん」)などである。これらの変数の組合せによってビジネスパターンが定義できる。バリエーションのパターンは、注文の受け方、処理の仕方、報告の仕方をあらかじめ組み合わせたサービスセット(既製品)としてまとめておく。ビジネスパターンから見てサービスセットとのマッチングの良さを、営業担当者に評価してもらい、それをマッチングルールとして蓄えておく。新規の連携先に対しては、ビジネスパターンを分析し、サービスセットのマッチングの良さからふさわしいバリエーションタイプを選んで提案する。この時点で外部連携メカニズムは既に動作可能であることに注意。もちろん、提案したサービスセットからの変異があれば、その分についてルール部品を差し替えるか、新規に作る。作るにしても小規模な開発であろうし、開発した部品は後に他の連携先でも使うことができる。

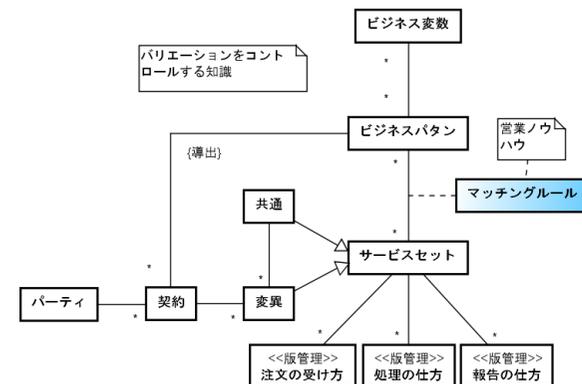


図7 バリエーションをコントロールする

5. おわりに

古くは人が仲介していたビジネス連携を、情報システムどうしを直接つなぐことで、より緊密にビジネスを効率的に連携させようとする動きが活発である。これをサービスの提供という切り口で見た場合、そのビジネス連携は、もともと異なる人間活動システムを連携するものであり、共約不可能生も含めて、コミュニケーションにおける未解決の課題が残されている。課題の一つはビジネス連携の多様さへの対処であり、もう一つは対話インタフェースの多様さへの対処である。対話インタフェースはシステム間をつなぐ手段であるが、それ自身、媒体、記号、コードからなるシステムであり、ダイナミズムを持っている。

本報告では、対話インタフェースというシステムを外部連携メカニズムとして設定し、バリエーションを集中的に管理することを提案した。その鍵となるのが連携のルールである。これは Mapping, Translation, Validation の3つのルールからなる。それぞれのルールをフィールドごとの処理部品に分けて保持し、連携先ごとに組み合わせで利用できるようにする。さらに将来的には、バリエーションを受け身的に扱うのではなく、顧客ビジネスのパタンに合わせて積極的に設計し、売り物にするという構想を述べた。

参考文献

- 1) Von Bertalanffy, L., "General Systems Theory," George Braziller, 1968. 長野敬, 太田邦昌訳, 「一般システム理論」, みすず書房, 1973
- 2) Weinberg, G. M., "An Introduction to General Systems Thinking," John Wiley & Sons, 1975. 松田武彦監訳, 「一般システム思考入門」, 紀伊國屋書店, 1979
- 3) Checkland, P., "Systems Thinking, Systems Practice," John Wiley & Sons, 1981. 高原康彦ほか監訳, 「新しいシステムアプローチ」, オーム社, 1985
- 4) Kuhn, Thomas, "The structure of scientific revolutions," The University of Chicago Press, 1996. 中山茂訳 『科学革命の構造』みすず書房, 1971
- 5) 次世代 EDI 推進協議会, 「広がる EDI」, http://www.jipdec.or.jp/dupc/jedic/edi/extend_edi.html, (2010年10月17日)
- 6) 財団法人中小企業総合研究機構, 「中小企業の次世代 EDI の導入等に関する調査研究」, 2005, <http://www.jsbri.or.jp/new-hp/work/business/h17-5.html> (2010年10月17日)
- 7) 児玉公信, 水野忠則, 「部品表の統合に関する一考察」, 情報処理学会 第70回全国大会, 2008/3/15