

XMLによる情報統合とプロセス統合： Enterprise Process Integration Calculus

若山 俊弘[†], Igor Djakovacki[†], 陌間 端[‡]
[†]国際大学 大学院国際経営学研究科
[‡]富士ゼロックス株式会社 IT事業開発センター

Web/XMLの急速な普及により組織全体さらに組織をまたがった大規模な情報統合が可能となりつつある。本稿は、情報統合・プロセス統合による業務プロセスの構造的改革を実現するため業務プロセス記述言語E P I C (Enterprise Process Integration Calculus)を提案する。情報とプロセスの二元性を有効活用するように設計されたE P I Cの主たるの特徴と機能について示す。さらに、E P I Cともに用いるネットワーク情報技術と経営学上の手法・知見の橋渡しをする中間概念としての方法論について考える。

Information Integration and Process Integration via XML: Enterprise Process Integration Calculus

Toshiro Wakayama[†], Igor Djakovacki[†], Tan Hazama[‡]
[†] International University of Japan
[‡] Fuji Xerox Co., Ltd.

The emerging environment of information interchange and processing based on Web/XML presents unprecedented opportunities for enterprise-wide and inter-enterprise information integration. EPIC (Enterprise Process Integration Calculus) is a process modeling language which, among other things, aims to facilitate large-scale business process integration through information integration enabled by this emerging technology and standard. This paper introduces main features and functions of EPIC. We also discuss, briefly, a conceptual framework for process design methodology which is intended to pair with EPIC.

1 はじめに

Web/XML [XML98]の出現により、全社的、さらには企業組織の壁を超えるスコープで情報統合がなされる環境が整いつつある。E P I C (Enterprise Process Integration Calculus)はこのような情報統合スコープの飛躍的拡大を業務プロセス上の改善、革新に結び付けることを狙った業務プロセス記述言語である。このことをサポートするために、E P I Cは「プロセス構成」と「ドキュメント構成」からなり、両者は「情報文脈化 (Contextualization)」「情報脱文脈化 (Decontextualization)」というパイプでつながっている。(図1)

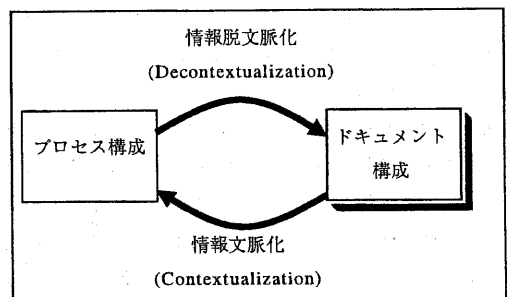


図1: E P I Cモデルの構成

業務プロセス記述言語は、言うまでもなく、すでに様々なプロセス支援のソフトウェアにみる事ができる(プロセス定義ツール、プロセスシミュレーションツール、ワークフローソフトウェア等)。これら既存の

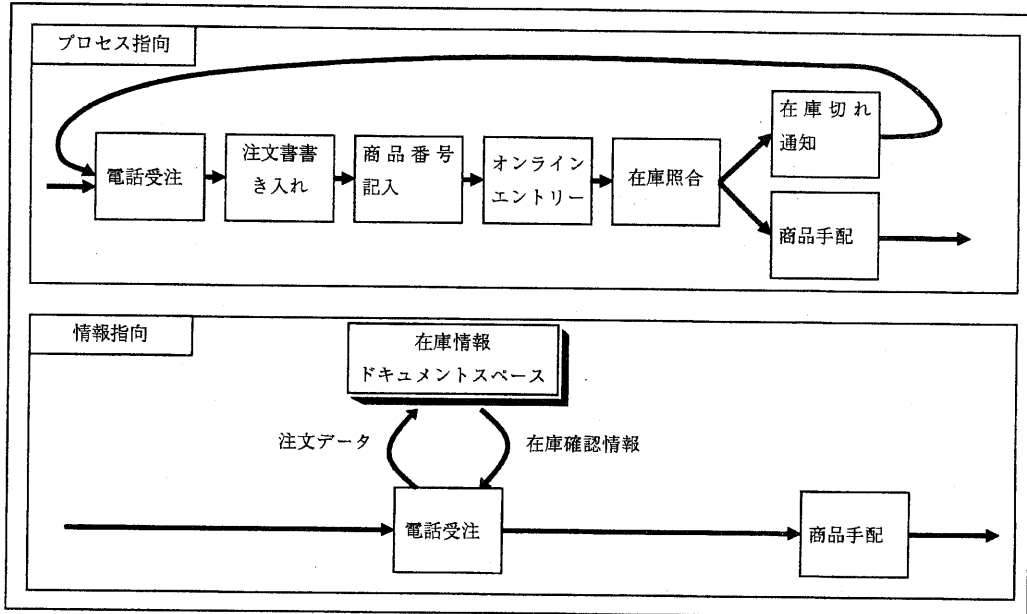


図2: 情報統合によるプロセス統合

記述言語は様々な問題点も指摘されているが⁽¹⁾、E P I Cを新たな記述言語として提唱せんとする最も基本的な理由は以下の2点である。

- ▶ 既存プロセス記述言語では、「プロセス」とそのプロセス内で生成・活用される「情報」の二元性(Duality)が十分に捉えられていない。
- ▶ 「情報とプロセスの二元性」の有効活用は、XMLを中核とした次世代Web技術の進展により、全く新たな局面を向かえている。

ここで言う「情報とプロセスの二元性」[Wakayama 98]とは、両者の形態にある種のトレード・オフが存在することを指す。いくつかのタスクから構成される受注・発送業務を例にとりて考える。このような業務をデザインするにあたり、1つの可能性は、個々のタスクがアクセス、処理、生成する情報を「小刻み」にし、それを補うようなかたちで数多くのタスクを導入することである。たとえば、受注・発送業務では、「電話受注」、「注文書書き入れ」、「商品番号記入」、「オンラインエントリー」、「在庫照合」、「在庫切れ通知」、「商

品手配」などのタスクが考えられる。これと対照的なデザインとして、個々のタスクが扱う情報を大きく区切り、そのことによって、必要タスク数を最小にする、ということも可能である。受注・発送業務の例で言えば、たとえば、「電話受注」時に在庫データをアクセスできるようにすることによって、「電話受注」、「在庫照合」そして「在庫切れ通知」を1つの統合化されたプロセスとして扱うことが可能となる。

この場合、「電話受注」時に顧客から入ってくる「注文データ」と、すでに存在する「在庫データ」から「在庫確認情報」が生成されなければならない。この生成が、たとえば自動化などによって、タイムリーに行われれば、受注-在庫確認のサイクルが短縮され、注文してきた顧客の取り逃がしを防ぐことにもなる。逆にいうならば、プロセス-タイムリーに生成された「在庫確認情報」(さらには、「在庫切れ代替品情報」)は、「注文してきた顧客は取り逃がすな」という業務上の「意図」を表現することになる。このように、既存の情報(「注文データ」、「在庫データ」)から、業務上、経営上の意図・価値を表現する第三の情報(「在庫確認情報」)を生成することを、ここでは「情報統合」と呼ぶ。(図2)

(1) たとえば、多くの場合「意味論」が欠落しあるいは不明確で、このような言語で記述されたプロセスデザインはそのデザインの意味・意図が正確に伝わらない、というような難点がある。

したがって、ここでいう「情報統合」は、単に情報(部品)を収集・再構築することではなく、その作業効果がプロセス文脈の中で意味・価値を持つことである。その意味・価値がプロセスに還元され、より有効なプロセス・デザインが可能となる。

このように、プロセス構造とそのプロセスで生成・活用される情報には密接な関連があり、これまでプロセス的に表現されていた部分(「電話受注」業務、「在庫照合」業務及び「在庫切れ通知」業務)を、より統合化された情報(「注文データ」と「在庫データ」からプロセス-タイムリーに生成される「在庫確認情報」として置き換えてやることができる。これが「情報とプロセスの二元性」である。しかしながら、現存のプロセスウエア(プロセス定義ツール、プロセスシミュレーションツール、ワークフローソフトウェア等)に組み込まれているプロセス記述言語は、このような二元性を十分にサポートするものではない。これがE P I C 提唱の背景にある、上記の第1点である。

続いて、上記の第2点に移る。第2点でいう、「情報とプロセスの二元性」にXMLがもたらす「新局面」とは、ひとことで言うならば、「データ」共有から「ドキュメント」共有への変遷を差す。ここで言う「データ」とは、リレーショナルデータベースで扱われるような情報及びその他の比較的、事実表現的(factual)な情報を指す。これに対し、このような「情報」に、その情報が使われる組織におけるその情報の「解釈」、「利用意図(Use Intent)」などが加味されたものを「組織知」と呼ぶ。上記の「ドキュメント」とは、このような「組織知」が何らかのメディア上に表現されたものを指す。XMLの出現は、「ドキュメント」共有という視点からみて、以下の意味をもつ。

- ▶ 対SGML: Web上で広くXMLが使われることによって、「ドキュメント」共有のスコープが飛躍的に拡大する。このことは、「情報とプロセスの二元性」を考えた場合、プロセス統合のスコープ拡大も意味する。
- ▶ 対HTML: XMLはメタ言語であり、ユーザがその意図にあったタグをデザインできる。タグの1つの機能は、そのタグ間にある情報コンテンツの「利用意図(Use Intent)」の表現にあ

る。タグはまた、ユーザの定義する属性も伴うことができ、このような属性を通し、様々な情報処理機能も表現できる。企業が情報を獲得、集計、分析し、さらには、二次加工、三次加工と、新たな企業価値を持つ情報を生成していく上で、XMLのこのようなメタ言語性は、「ドキュメント」共有・統合に新たな可能性を開く。

E P I Cは、このような、XMLの持つ情報統合・プロセス統合の可能性を充分に実現せんとする、これまでになかった「業務プロセス記述言語」の試みである。以下では、E P I Cの特質、主要機能を概略する。

2 E P I Cの特質

2.1 Syntax

現存のプロセス記述ツールで表現されたプロセス・モデルは、そのツールでしか見ることができない。これに対し、E P I Cで書かれたプロセス・モデルはXML文書として、Web上で自由に配布できる。これはE P I CのシンタクスがXMLのDTDとして定義されているからである[若山 98]。したがって、E P I C DTDのレイアウトが定義されていれば、E P I CモデルはWebブラウザで見ることができる。

さらに、E P I Cモデル(schema)のシミュレーション及び実行時の「状況」(Situational Instance)そのものもXML文書として記述される(上記E P I C DTDから派生する「E P I C Situation DTD」下におけるXML文書として)。この「E P I C Situation DTD」のレイアウトが定義されていれば、シミュレーション及び実行時の動的推移もWeb上で見ることができる。

2.2 Semantics

プロセス記述言語は、その言語によって記述されたモデルがその実行時にどのような業務上の動的推移を指示しているのか明確に伝えることができないなければならない。この「動的推移の指示」を与える部分がプロセス記述言語のセマンティクスであるが、既存のプロセス記述言語の場合、このセマンティクスが明確でない。記述されたモデルをシミュレーションあるいは実行して初めてそのセマンティクスが見えてくるといのが現実である。これでは、モデル実行時にそのモデ

ルがいつその設計者の意図から乖離した(場合によっては、設計者の意図と対立した)「動的推移の指示」を出してしまうかわからない。このような言語に、複雑な企業システムのモデル化及び実行のサポートを託するわけにはいかない。

E P I Cはそのセマンティクスが体系的・自己完結的に(つまり、シミュレーション・エンジン、実行エンジンに頼らず)定義されているので、特定シミュレーション・エンジン、実行エンジンに拘束されることなく、「プロセス知」の媒体として Web 上で機能できる。E P I Cはこの点において、既存のプロセス記述言語と一線を画する。

2.3 ネットワーク・ビジネス言語

ネットワーク情報技術の急速な進展により、企業システムがオンライン上に再構築される可能性が現実のものとなりつつある。ここで必要となるのは、このような企業システムを分析、設計、そして討議するための「ビジネス言語」である[Haeckel 93]。このような「ビジネス言語」のポイントには以下の2点がある。

➤ 企業業務システムが記述の対象であり、このようなシステムの最適設計がサポートされなければならない。既存のワークフローシステムの「プロセス記述言語」は、ワークフローの自動化がその視座の中心にあり、企業システムの最適化サポートという視点からは必ずしも満足できるものではない。業務システムとして最適化されていないシステムを自動化しても、そのコストに見合う結果が得難いのはよく指摘されるところである。

➤ 「ビジネス言語」が業務システムを記述するための「語彙」を備えていなければならない一方、Web時代の「ビジネス言語」はネットワーク情報技術の提供する様々な機能とうまく連携し、その効果を最大限取り込むことのできるものでなければならない。

第一点にある「企業システムの最適設計」とは、個々の業務で派生する様々な「ローカルな要請」を業務全体の「グローバルな方針・制約」のもとに調整、システム化することである。E P I Cでは、個々のプロセスで派生する「ローカルな要請」(たとえば、個々の

受注業務から派生する「注文データ」)を、それらプロセスに付随する「記帳属性」で捉え、情報脱文脈化のパイプで「ドキュメント構成」内に収集する。収集された「ローカルな要請」は、「ドキュメント構成」内に存在する「グローバルな方針・制約」を表現する情報(たとえば、「在庫データ」)と照合され、その結果ある種の調整情報(たとえば、「在庫確認情報」)が生成される。生成された調整情報は、情報文脈化のパイプを通して関連プロセスに配布され、その関連プロセスに付随する「参照属性」の値としてプロセス担当者に提供される。

続いて上記第二点の「ネットワーク情報技術をフル活用する Web 時代のビジネス言語」に移る。ネットワーク情報技術進展の一つのハイライトは、XMLによる情報交換・情報合成である。E P I Cはネットワーク情報技術の進展・普及の意味を、業務システムの観点から、「情報統合・プロセス統合」と解釈し、それが十分にサポートされるように設計されている。ネットワーク情報システムが機能的に整備される一方で、情報システムとその上に構築される業務システムとの乖離(後者が前者の機能を十分に反映できない、など)が指摘されている。E P I Cでいう「情報統合・プロセス統合」とは、情報システム、業務システムの間位置する「中間概念」である。E P I Cは、この中間概念を通して、業務システムが情報システムを十分に有効活用できるように設計されている。

2.4 定型的、非定型的業務ともに対応

E P I Cにおける「プロセス構成」は基本的には、様々なタスク及びそのタスク間の依存関係が階層的に表現されたものである。したがって、その依存関係の総体が比較的定型的な場合には有効であるが、このような定型性が弱い時はタスク間の連携を「プロセス構成」内で捉えることには無理がある。通常のプロセス記述言語の場合、このような、タスク間の非定型的な連携は記述できない。E P I Cでは、タスク間の関連は、「プロセス構成」内で明示的に記述できなくても、「ドキュメント構成」がそのようなタスクの「共有情報スペース」として機能するので、「ドキュメント構成」を介して実現できる。

3 E P I Cの基本機能

3.1 基本機能の概要

情報システムと業務システムの間に位置する「情報統合・プロセス統合」という「中間概念」を通して、E P I Cは機能する。どのようなメカニズムによってそのような特質が得ることができるのか基本機能によって示す。E P I Cは、プロセス記述言語ではあるが単に業務プロセスが記述できればよいとは考えていない。E P I Cは、特定の情報システムを想定しているわけではないが、実際の業務プロセスを革新するためのメカニズムとしての情報システムが必要となる。E P I Cの基本機能は、情報システムによって実現されることを前提としている。

E P I Cの基本機能には大きく2つの側面がある。状況の管理と動的推移の管理である。状況は、ある定義されたE P I Cモデルの実行時の状態、すなわちプロセスの状態、エンティティの内容、属性の値等の全てを表している。これらは、モデルによって与えられた条件に合致している必要がある。「電話受注」と「商品手配」の結ばれた2つのプロセスの間にはエンティティ「確認済み受注データ」が存在しなければならない。動的推移は、プロセスに割り当てられた実行者(人間に限らず機械やプログラムも含む)が実際に業務を行った結果を状況の変化として捕らえることである。その際にモデルに照らして妥当性を検証する必要がある。「在庫確認」を行わずに「商品手配」を行ってはならない。

3.2 プロセス構成の機能

プロセス構成は、既存のプロセス記述言語と同様にプロセスをサブプロセスから成る階層構造として設計する。プロセスとプロセスの間にはポートが存在する。ポートは、前工程のプロセスが出力したエンティティを一時的に保持する場所で、後工程のプロセスは入力ポートにエンティティが存在する場合に発火することができる。このようにE P I Cでは、ポートとエンティティを媒介としてプロセスが結合される。エンティティは、プロセスを定義するにあたってそのプロセスの行う業務に対する意図を表現したものである。すなわち、「入力となるエンティティを受け取った時点

で、出力として定義されるエンティティを後工程に引き渡すことのコミットメント」がそのプロセスの意図として表現されている。エンティティは、XML文書、文書型定義(DTD)、あるいは、物理的な機械部品であってもかまわない。

プロセス構成の状況は、各プロセスからなる構造がモデルと照らし合わせて妥当でなければならない。あるプロセスが、いくつかのサブ・プロセスに分解される場合、親プロセスのコミットメントを子プロセスの結合によって実現する。そのとき、親プロセスのコミットメントは、入力エンティティに対して指示されるエンティティを出力することであるので、サブプロセスは、必ず入力エンティティを入力とするプロセスから始まり、親プロセスに指定されているエンティティを出力するプロセスで終わるように構成しなければならない。その途中の各サブプロセス間は、出力されるエンティティが別のサブプロセスの入力となることによって結合される。エンティティは、そのプロセスが他のプロセスと結合するに必要なものであれば、実体を伴わないシグナルであっても差し支えない。実体のある何ももの引き渡されなくとも、前のプロセスが終了したことを確認した後に開始するプロセスは、前のプロセスから終了シグナルをエンティティとして受け取っていると定義し得る。

プロセス構成の動的推移は、コミットメントの生成・消滅によって行われる。実行時にプロセスは、コミット可能、コミット終了可能、どちらでもない親プロセス3つの状態をとる事ができる。ポートからエンティティが入力できる状態であればコミット可能状態である。ポートにエンティティを出力できる場合がコミット終了可能である。まず、コミット可能な親プロセスのコミットメントが生成される。そのとき1つ以上のサブプロセスがコミット可能となる。次にコミット可能なサブプロセスのうち一つがコミットメントを生成する。それが階層構造を辿っていきリーフレベルのプロセスであれば、コミットメントは終了可能である。コミットメントを消滅させ、ポートに定められたエンティティを出力する。そのポートに結合された別のプロセスがコミット可能となる。サブプロセスがコミットメントを終了し、親プロセスの出力したいエン

ティティをポートの用意できた時点で、親プロセスもコミット終了可能となる。コミットメントの推移をE P I Cは管理することによりプロセスの統合を実現する。

3.3 ドキュメント構成の機能

ドキュメント構成は、プロセス間の情報共有を支援する。プロセス構成として、構造化されていない情報の流れについては、ドキュメント構成を用いた共有機能が用いられる。また、外部関数を呼ぶように指定することによって、文書部品間の関係を的確に表現することができ、どのような情報をどのように加工して共有するかを記述することができる。

また、トラッキングツリーと呼ぶ構造化された共有情報の表現により、プロセス内で生じるプロセスが進行するに従って変化する追跡すべき情報を再構成することができる。受注・発送処理の場合、各注文がどのプロセスでどの状態にあるのかといった情報や、さらにそれを再構成し、各顧客別の注文状況の経緯と比較といった複雑なものも追跡するように定義することができる。

ドキュメント構成の状況とは、ドキュメント構成として定義されている文書と文書部品およびそれら属性のある時点の全てのコンテンツと値である。モデルに定義されている文書クラスに従ったインスタンスが値でなければならない。

各文書、文書部品間には依存関係が付けられている場合、それらの依存関係の連鎖をすべて評価することが動的推移の一単位となる。プロセス内に定義されている記帳属性の値が設定・変更された場合、それに依存している全ての連鎖を順に評価していく。それによって、トラッキングツリーをはじめとするドキュメント構成中の文書や文書部品が更新され、さらに参照属性としてドキュメント構成中の文書を参照していたプロセスの属性値が変更される。これらの一連の動作がドキュメント構成の動的推移を成す。

4 E P I C方法論

BPRは業務プロセスの構造的改善に主眼を置くものであるが、その成功率は低いといわれている。これには様々な理由がある。一つには、ネットワーク情報技

術の急速な進展にも拘わらず、その情報システムとしての機能をBPRなど経営学上の手法・知見に結び付けるための「中間概念」の欠如がある。情報システム上に構築されたこれら中間概念が体系的に整理されたものを「メソドロロジー・ウエア」、「ユース(Use)アーキテクチャー」等というが、業務プロセスの構造的改善においてはこの「メソドロロジー・ウエア」の不備・欠如が指摘されている。

情報インフラにはハードウエア・ソフトウエア等比較的目につく部分とそれ以外に「メソドロロジー・ウエア」などいわば「見えざるインフラ」の部分がある。日本が欧米先進国に対し情報インフラで遅れているというときは、たいてい、従業員100人に対してコンピュータが何台などという「見えるインフラ」の部分をさす。しかし、この遅れはまさに目に見えるがゆえに、認識もされやすいし対策もまともやすい。これに対し「見えざるインフラ」には、

- その概念そのものが十分に把握されていない、
- 文化・組織を超えて移植することが単純ではない、
- 個人のみならず集団・組織のレベルで理解され、受け入れられなくてはならず、時間がかかる、

というような特徴・状況があり、その開発、浸透には「見えるインフラ」にはない難しさがある。このような「見えざるインフラ」の形成においては、米国は数年前より確実にそのノウハウの蓄積期に入っている。この背景には、米国においては、このようなインフラが民間主導で形成されていく文化的・社会的土壌が整っている、ということがある。情報技術とその有効活用における日米のギャップは、単に最新のハードウエア・ソフトウエアをとりいれるだけで埋まるものでないことは銘記されなければならない。日本がこのような状況に対処するためには、「見えざるインフラ」の形成手法が体系的に整理・構成され、詳細な実用例も豊富に盛り込んだ「手引書」の開発が有効、不可欠と確信する。

以上が、業務プロセスデザイン方法論構築の一般的な背景である。E P I C言語を想定し、その上に構築される業務プロセスのデザイン方法論をデジタルプロ

セスハンドブックとして体系化したいと考える。一般に、EPICも含め、デザイン言語とその言語に則ったデザイン方法論は緊密な関係にあり、両者が一体となりデザイン作業の実際のニーズに答えることになる。たとえば、データデザインの実際には何らかの形で階層構造のサポートが要求されるわけであるが、これはデザイン言語の中に組み込んでも、あるいはデザイン方法論にとりいれてもよい。前者の例がXMLで、後者の例がリレーショナルデータベースにおけるER (Entity-Relationship) 言語である。ER言語の場合、階層構造の使い方も含め、包括的はデザイン方法論が確立、普及している。リレーショナルデータベースが数年の間に情報産業の一翼を担う技術に成長し市場を開拓できたのも、このような方法論の整備・普及がその一因として背景にある。

プロセスデザイン方法論の具体的な表現としては、大きく分けて以下の二つのタイプがある。

- プロセスプレート型：プロセスプレート、特に特定応用領域に特化したプレートを提供する。提供されたプレートがその領域におけるプロセスの組み方の「手本」あるいは「可能性」を表現する。
- メタプロセス型：プロセスをデザイン・改善するためのプロセス、つまりメタプロセスを提供する。プロセスプレートはこの過程で生成、あるいは参照されるものとして位置づけられる。

前者の例としては、マサチューセッツ工科大学、Center for Coordination Science におけるプロジェクト、「プロセスハンドブック」[Malone 93]がある。EPICに基づく「デジタルプロセスハンドブック」は後者のメタプロセス型が主体になる。理由は以下の通りである。

- 第三者のデザインしたプレートはわかりにくい。
- テンプレートも含め、プロセスをデザインする過程で生まれるプロセスに対する理解、特にデザインチームが共有できる理解が大切である。デザイン過程が「組織知工学」の実践でもある。
- メタプロセス型ハンドブックはXML用途技術

の対象として位置づけられる。メタプロセスもプロセスであり、XMLによる情報統合、プロセス統合の対象となる。

- メタプロセスとしてもっとも高度に様式化されたものとして、日本で確立し、世界に知られるようになった「Kaizen」がある。メタプロセス型ハンドブックは日本の「Kaizen」資産から学ぶことができる。

5 謝辞

最初のEPICワークショップに参加し、その後も多くの議論と助言をいただいたの Srikanth Kannapan 博士 (IBM Global Services, Bangalore, India) と Bharat Jayaraman 助教授 (State University of New York, Buffalo, New York, USA) に深く感謝する。

6 参考文献

- [XML 98] W3C Recommendation 10-February-1998 : Extensible Markup Language (XML) 1.0, <http://www.w3.org/TR/1998/REC-xml-19980210>
- [Wakayama98] Toshiro Wakayama, Srikanth Kannapan, Chan Meng Khoong, Shamkant Navathe and JoAnne Yates: Information and Process Integration in Enterprises: Rethinking Documents, Kluwer Academic Publishers, 1998.
- [若山 98] 若山俊弘, 陌間端: Enterprise Process Integration Calculus: Syntax and Semantics (執筆中)
- [Haeckel93] S.H. Haeckel and Richard L. Nolan: Managing by Wire. Harvard Business Review, September-October 1993.
- [Malone 93] Thomas W. Malone, et al: "Tools for inventing organizations: Toward a handbook of organizational processes," In Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises, 1993, <http://ccs.mit.edu/CCSWP198/>