

- 私家版：コンサルティング手帳 -

第1回「ビジネス・アーキテクチャ:ビジネスのCSモデル」



EST コンサルティング本部 桐山 俊也

1. はじめに

一定以上の規模を持ったシステムを設計・開発する際には、アーキテクチャ設計が欠かせない。綿密に箱と針金をつなげ、OS やツール、DBなどを慎重に配置して、システムの全体構成を作り上げる。なにしろ、このシステム・アーキテクチャの善し悪しが、情報システムの善し悪しを大きく左右するのだから。

けれども一般にSEは、適用対象となる事業や業務（以後、一括して「ビジネス」と記す）については、それほど綿密にアーキテクチャを設計しているとはいえない。どちらかといえば、「アーキテクチャは実装技術だ」と、括り出してしまう傾向があるように思える。システム・アーキテクチャは綿密に組み上げるけれど、うっかりすると、ビジネスは、乱暴に「ありのまま」に合わせているだけですませてはいないか。どうして、そんな極端な差が出るのだろうか。

SEはしばしば、機能、道具、組織などの「見えるもの」に飛び付きがちだ。業務システムの要求分析・定義などをする段になると、いきなり、次のような項目に目を向けようとする。

- ・「システム機能ニーズ」
- ・「資金部の業務概要」
- ・「営業店の数と場所」
- ・「データ量と項目数」
- ・「ハードと開発ツール選定」

その一方で、次のような開発指針を声高に唱える。

- ・「業務システム間の連携」
- ・「データの重複排除」

- ・「共通機能の標準化」
- ・「最適な機能分散」

そうはいつでも、いきなり細部に入り込んでしまっているのだから、全体が見えるわけではない。上記の各指針は行き場を失ない、開発の途中で妥協・断念されるケースが多いのではないだろうか。

どうやら私も含めてSEは、ビジネス・アーキテクチャに触れないまま、システムを作ろうとする傾向を持っているようだ。個々の作業やその手順、組織の構造や機能などの、見た目だけでビジネスを解釈してしまっているように思われる。けれども、ビジネスにもアーキテクチャがあって然るべきなのだ。

アーキテクチャの基本ともいえるクライアント/サーバー・モデルを使って考えていこう。「クライアント/サーバー」の“そもそも”を慮ると、クライアント(=顧客)と、サーバー(=サービス提供者=企業)の関係がベースだといえる。アカデミカルには違う由来があるのかもしれないが、今はここから出発したいと思う。

SEが、業務システムのクライアント/サーバー設計を行う際には、ビジネスに内在するクライアントとサーバーとの関係を、きっちりと踏まえておく必要がある。ビジネスのアーキテクチャと、システムのアーキテクチャとが、きちんと噛み合っていれば、声高に唱えた開発指針も生きてくる。逆に、ビジネス・アーキテクチャという鳥瞰的な視点がなければ、情報システム開発は、いつまでも「やつつけ仕事」に陥ってしまう危険があるだろう。

2. ビジネス・プロセスの外枠とは

ビジネスを「プロセス」として表現する基本的なモデルの一つに、入力と、それに対する出力、そして処理そのも

の、という3要素で捉える方法がある(図1: Fig.1)

I/O だけに着目した機能主義でみるか、内部構造を掘り下げる構造主義でみるかはともかくとして、多くのコンピュータ・プログラムの基本的なパターンは、IPO(Input ~ Procedure ~ Output) のモデルで考えられる。IPO にもとづく「処理」を「なにかをすること」ぐらい大胆に解釈するならば、私たちが『お化けを見て(入力) 驚き(処理) 「ぎゃ!」と声を上げる(出力)』という一連の行動も、IPO モデルで表現できる。だとしたら、もう少し手前に戻して、「営業」とか「事業」といった、社会的行為も、「処理」に対応できるはずだ。

仮に、問屋(卸業)を想定して、その会社のビジネスを分析してみる。会社全体の「処理」は、注文(入力)を受けて、商品を提供(出力)する一連の仕事、すなわち「卸し」と表現できる(図1: Fig 2)。さらに前後をみていくと、「卸し」という処理のトリガーになっている処理は「発注」で、「卸し」の後を引き継ぐ先は「受領」...と定義できる(図1: Fig 3)。

この段階で、モデルは、処理 処理 処理という一連の流れを表わしているが、まだ、プロセスではない。プロセスは、次の基本構成を満たしている必要がある。

- ・ Start
何らかのインプットを受け取る...どこから?
インプットにもとづいて、処理を行う...なにを?
結果をアウトプットとして引き渡す...どこへ?
- ・ End

そこで、一旦、注文の発信元を「発注者」、商品の受け取り手を「受領者」として、From/To を実体に置き換えて定義し直すと、上記の Start ~ End が満たされた全体像が表現できる。すなわち、これが、「卸し」というビジネス

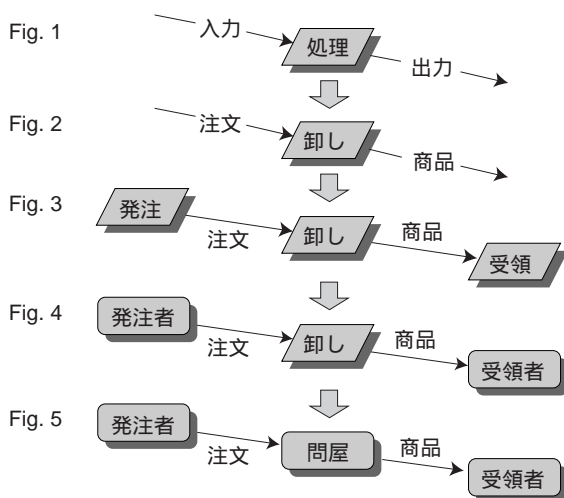


図1 ビジネス・プロセスの基本モデル

ス・プロセスの外枠になるわけだ(図1: Fig 4)

3. ビジネスからクライアント/サーバーへ

ここまでで、問屋ビジネスをIPOからはじめて、単純なモデル化による一つのビジネス・プロセスを次のように定義できた。

- ・他の処理主体から情報なりメッセージなりを受け取って、業務処理(作用というか、メソッドというか...)を行い、その結果を他の処理主体へ引き渡す。

このとき、「卸し=問屋のオヤジが行う事業」は処理の内容、「発注者」/「受領者」は処理の主体である。さらに一歩進めて、「問屋=卸事業を行っているオヤジの会社=企業」という主体化を行えば、全体が処理主体に統一された関係図が得られる(図1: Fig 5*)。

さて、特別な場合を除けば、発注者と受領者は同一なので、Fig 4, Fig 5に示している発注者と受領者は、一括りに「顧客」にまとめられる(図2: Fig 6)。

また、注文や商品といった入出力は「リクエスト」「サービス」として一般化できる。そして問屋も「企業」として一般化できるので、それらを皆、一般化(というか抽象化というか...)してみると、次の2つの関係に整理できる(図2: Fig 7)。

- ・リクエストを受け取って(IN) サービスを提供(OUT)する「企業」。
- ・サービスを受け取る(IN)のために、リクエストを出す(OUT)「顧客」。

冒頭で、「クライアント/サーバーの“そもそも”はビジネスである」と記したが、それだけでは必ずしも、ビジネスが本質的にクライアント/サーバーだ、という逆説が成立するとは限らない。しかし、今までの整理にのっとり

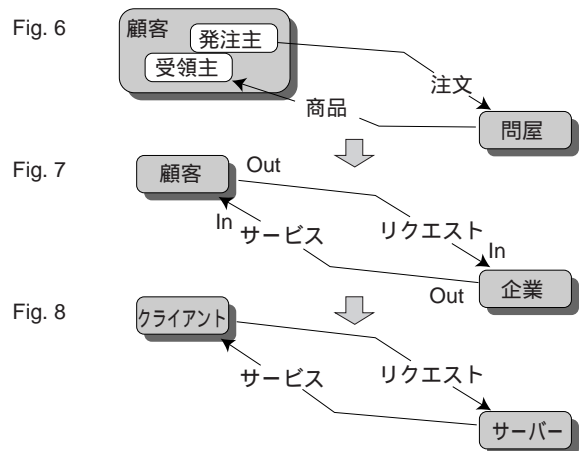


図2 「問屋」と「クライアント/サーバー」構造の類似

* 1) 巧みなクラスへのすり替えが起きていますが...おっと、手の内がバレバレ(^ ^ ;)

ば、「顧客」という処理主体を汎化した「クライアント」クラスへの変換は容易だ。同様に、「企業」という処理主体を汎化した「サーバー」クラスへの変換も容易だ(図2: Fig 8)。ここまでくれば、少なくとも、問屋というビジネスの本質に、クライアント/サーバー構造があるという結論が導きだせる。

4. 今度はプロセスの内側へ

ところで、ビジネスを全企業レベルで捉えたシステム開発の機会、それほど多くない。日常のシステム開発の大部分は、企業内の特定の事業(部門)をターゲットとした業務を扱うものではないだろうか。そこで、全企業レベルの定義はひとまず置いておき、再びビジネス・プロセス(図3: Fig 4)まで立ち戻って考える。

問屋のビジネス・プロセス全体をさらに詳細化してみたら、図3: Fig 9のようになったとする。威勢の良いオヤジが、一人で木材問屋を営んでいる...などという場合でも

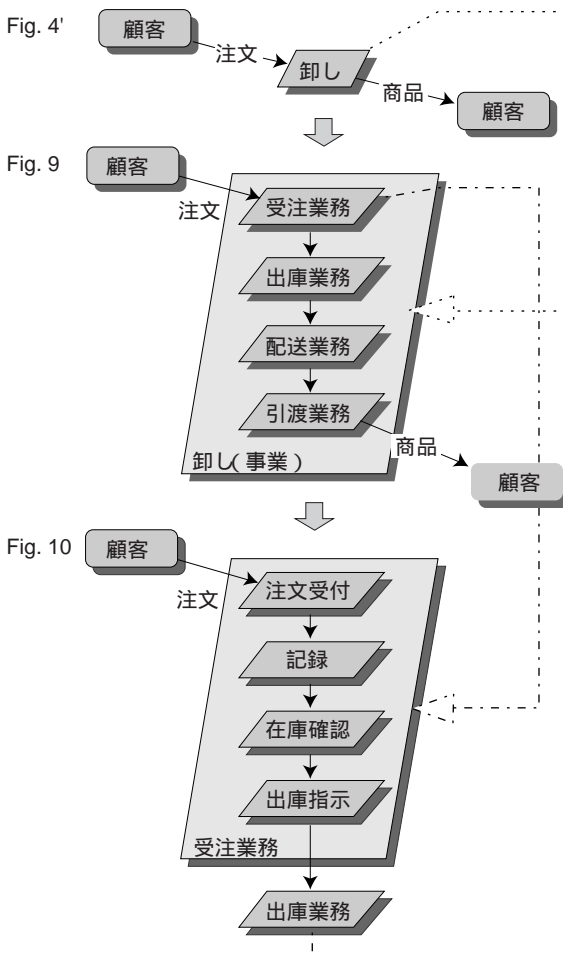


図3 ビジネス・プロセスの詳細

ない限り、どんな事業にも、ある程度は明確な流れがあるはずだ。この問屋の場合は、顧客から注文を受けて(受注業務) 蔵や倉庫から品物を出してきて(出庫業務) それをトラックなどに載せて顧客のところまで持って行き(配送業務) 荷下ろしして引き渡したら、代金をもらい、納品書にハンコをもらうとか(引き渡し業務)する、という一連の流れが見いだせる*2。

「卸し」処理は、さらに詳細ないくつかの処理によって成立していると定義できる。さらに、「受注業務」について、もう1段階の詳細化をすると、図3: Fig.10のように、個々の作業に近づく。もちろん、「受注業務」の最初の処理である「注文受付」をさらに分解すれば、次のような具体的なアクションに落ちる。

- ・電話で対応し、顧客からの注文を聞く
- ・扱い品目の台帳を見て、品物の発注番号を確認する
- ・受注伝票に、日付/品物/数量/納品先を記入する
- ・受注伝票のコピーをとって保管し、原票を主任に提出する

ここまでくると、「受注伝票を作成したあとに『記録業務』があるので、作業が二重になっている」とか、「記録と在庫確認をシステム処理に置換すれば、電話で顧客に対応しながら、出庫指示まで一人でできる可能性がある」とか、いろいろ問題や改善ポイントが見えてくるだろう。

つまり、企業全体の活動は、いくつかに分けられる個別の処理の総体であり、それぞれの個別処理も、さらに詳細な具体的な処理の総体である、というフラクタルな関係が、事業レベルから担当者の行動レベルまで続いているわけだ。これらをすべて書き連ね、それを引き伸ばすと、一本の長い線(プロセス)になる。きわめて多くの作業や業務が連なって、はじめて、一つの「卸し」という事業が成立する。しかし、仔細にはどうあろうとも、全体としては常に、「顧客から注文を受け取って、商品を顧客に引き渡す」プロセスにほかならない。そしてそのプロセスを成立させているのは常に、処理主体の連携によるクライアント/サーバー構造なのである。

5. 誰がやっているか?

ここまで私は、「その処理を実際にやっている」人や部署を「処理主体」と呼んできた。ある処理を誰がやっているかは、ビジネス・プロセスの分析や定義の上で非常に重要である。しかし、企業を分析していくと、しばしば、現実の組織や担当者とプロセスが一致しない。必ずしも、1処理: 1主体とは限らないし、処理の流れと組織の担当分けが一致しているとも限らない。

* 2) 現実には、もっと組織だっているはずですが...うっ...オヤジのイメージから離れられない(un;)

問屋モデルの場合（図4：Fig .11）受注業務は販売部門が担当している。出庫業務は、出庫部門ではなく*3、仕入部門が担当している。仕入部門は、その会社における品物の入りを一括して処理している。また、配送業務と引き渡し業務は、配送部門が一括担当している。商品を配送すれば、最後に引き渡しが行われるので、一括りに「届ける」と考えれば自然だろう。このように考えてくると、業務処理の流れ（ビジネス・プロセス）と組織・担当者とは、必ずしも1対1にはならない。

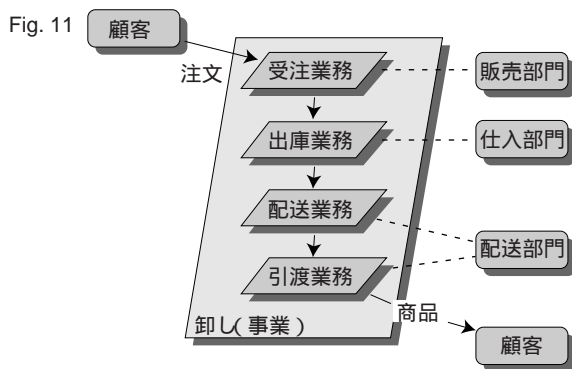


図4 問屋モデルでの部門間の流れ

私たちは、得てして、すでに存在している組織の形に目を奪われてしまいがちだ。例えば、次のような発想をしていないだろうか。

- ・財務部があるから、財務を管理する業務があって、財務処理がある。
...だから、財務管理システムを作ろう。
- ・業務部があるから、「業務」という業務があって、業務処理がある。
...だから、業務システムを作ろう。
- ・関西統括部があるから、関西を統括する業務があって、統括処理がある。
...だから、関西統括システムを作ろう。

しかし、頭の中では理解しているつもりでも、よくよく考えると、次のようにブラックボックス化している部分もある。

- ・じゃ、管理...ってなに？
- ・じゃ、業務...ってなに？
- ・じゃ、統括...ってなに？

案外、ブラックボックス化した時点で安心し、明確な解析や定義を省略してしまっていないだろうか。

仕入部門などの「組織」を出発点にすると、得てして、部門が担う数多くの業務から頭を切り替えにくくなる。つまり、一連のビジネス・プロセスの上で、個々の部門がど

のように関わっていて、その結果、全体としてどのような処理の流れになるかといった全体像を見失いがちだ。すると、例えば仕入部門の役割や業務処理を定義する際、顧客から販売部門を経て注文が入ってくる流れや、配送部門によって商品が顧客に届くまでの流れなどが把握できなくなる。

受注～納品プロセスを一本の流れとして捉えてきた最初の分析には、仕入業務は現われなかった。しかし、現実の仕入れ担当部門は、単一業務だけでなく、商品の入りと商品の出との両方を担っている。なぜ、ビジネス・プロセスと担当組織との間に、こういう相違が生じるのだろうか。それは、次の2つが、そもそも《異なるビジネス・プロセス》だからである。

- ① 注文に応じて商品を卸すビジネス・プロセス
- ② 在庫に応じて商品を仕入れるビジネス・プロセス

現実の組織や担当分けを静的に捉えてみると、それは得てして、いくつものプロセスをその部門が関わる範囲だけで「ぶつ切り」したものだ。企業内にプロセスが、ただ一つしかないならば、プロセスと組織（担当者）との関連は交錯しないが、現実には、問屋の仕入部門のように、一つの部門や担当者がいくつかの業務処理をこなしている場合がほとんどである。すると、「ある部門の業務システムを開発する」からといって、その部門の業務だけをどんどん深堀りしていくと、たくさんの雑多な機能や、どこにどう繋がるのかが意味不明な処理ばかりが出てきて、收拾がつかなくなるといった問題が起こりやすくなる。

業務分析（設計）でビジネス・プロセスを作った後に、それぞれの処理の主体をマッピングすれば、業務の流れと組織（担当者）との対比が可能になる。業務を処理に置き換えて、頭から最後までの流れとして把握すれば、ぶつ切りの組織構造に惑わされにくいので、業務の流れと組織の役割との整合性を把握しやすくなる。

6 . ビジネスのクライアント/サーバー・アーキテクチャー

これまでの分析で、問屋会社の処理を全体として大掴みにすれば、「顧客」と「問屋」のクライアント/サーバー構造で示せる（図5：Fig 6'）と分かった。

そして、その処理は、いくつかの階層に詳細化できる処理の総体だと分析した。ビジネス・プロセスとして処理をみる限り、それは一連の流れである。けれども、処理の主体に着目して、それぞれの主体の関係を整理すると、処理は、必ずしも単純な流れではない。例えば、販売部門から仕入部門に出庫依頼が出た段階で、仕入部門からは、ACKかNAK*4が返ってきているはずである。また、仕入部門

* 3) だいたい、そんな部門はほとんど考えられない^^ ;

と配送部門の間には、荷受けと引き渡しに相当するやり取りがあるだろう。そして、配送部門と販売部門の間にも、次のようなやり取りがあるはずだ。

「仕入部に在庫してくれるよう言ったから、荷受けしてね」

「はい。荷受けして、お客に届けといたかんね」

このような関係は、プロセスを何段階か詳細化していく中に現われてくる。それぞれの処理主体の関係をまとめると、図5：Fig.12のようになる。

この図は処理の流れではなく、処理主体の関係を表わしているので、順序や時間軸は表現されていない。けれども、お互いにクライアント/サーバー関係にある販売部門、配送部門、仕入部門それぞれの関係が表わされている。そして、個々のクライアント/サーバー関係を内包したまま、「顧客と問屋」という、全体のクライアント/サーバー関係も、崩れずに成立している。

これが、基本的なビジネス・アーキテクチャである。ただし、このクライアント/サーバー関係は、顧客との関係からスタートしているので、顧客から見える問屋の顔、ともいべき部分にしか焦点があたっていない。図5：Fig.12のモデルは、厳密にいうと問屋が行う、いくつかの処理の一側面を捉えているに過ぎない。多くの場合、現実の企業はもっと複雑である。例えば、クレーム対応や不良品の引き取りなどの処理はどうなっているだろう。生産者からの仕入れそのものや新製品のプロモートなども、ここ

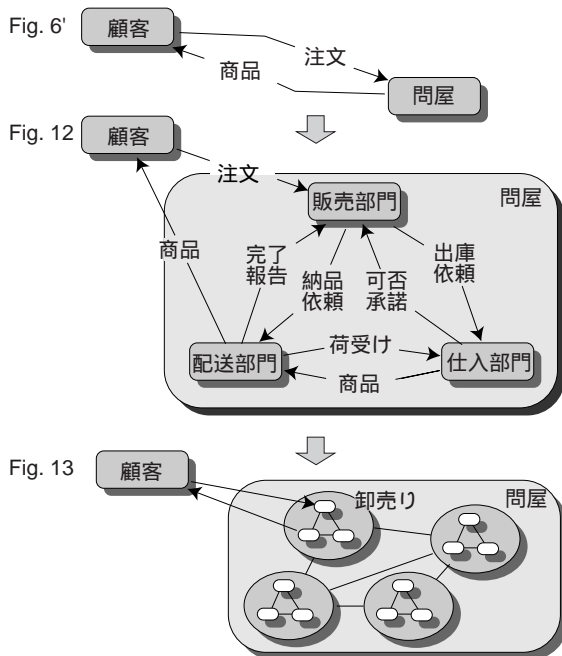


図5 基本的なビジネス・アーキテクチャ・モデル

には現われてきていない。さらに、金銭や小切手、手形などの処理は、どこに隠れているのだろうか。

このように考えていくと、一つの企業には、お互いに直接は運動しない、複数の異なるビジネス・プロセスが存在していると想像できる(図5：Fig.13)。モデルが悪いわけではない^{*5}。もともと、一つのビジネス・プロセスを採り上げているモデルなので、全体が表現されていないだけである。換言するなら、次のようにいえるだろう。

「一つの事業が、二つ以上の業務領域(ドメイン)を持っているなら、その業務領域ごとに、ビジネス・プロセスを表わす、別のモデルが必要だ。」

図5：Fig.13は、卸売りという事業を表わしている。しかし、これは問屋が持つたくさんのビジネス・プロセス群の一つに過ぎず、同時に、その会社の内外に存在する、いくつかのドメインの一つに過ぎないわけである。

7. じゃあ、全体のアーキテクチャは...?

さらにビジネス・プロセスを定義していけば、図6：Fig.14のような全体アーキテクチャが得られる。

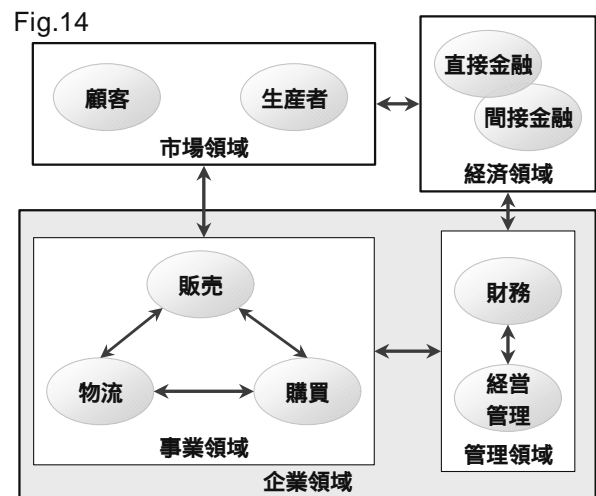


図6 全体アーキテクチャ

卸売り(=販売)は、物流、購買、などととともに、大きく事業領域としてまとめられるだろう。一方、企業の内部を隈なく見回すと、財務や会計に相当する部門がある。これらの部門は事業領域とは別途に、管理領域として括れると思う。すると、この問屋モデルには、事業領域と管理領域があって、それぞれがクライアント/サーバー関係にあると表現できる。

* 4) ACK: Acknowledge: 承認応答。「オ~ケー!(^ ^)」って意味ですね。

NAK: Negative Acknowledge: 否認応答。「んだめっ!!(´ `)」って意味ですね。

* 5) と、私は思っています...弱腰(un;)

また、事業領域は、顧客だけでなく取引業者（生産者）などとも関係を持っている。管理領域は、社内のみならず、銀行や証券会社などとも関係をもっている。だから、この問屋は、社外との関係を数多く持っているという状態も表現できるわけである。図6：Fig .14では、仮に「市場領域」と「経済領域」と命名したが、それぞれの顧客や銀行の中にも、この問屋と同様に、いくつかのドメインがあるはずだ。つまり、クライアント/サーバー関係は、何処まで行ってもフラクタルに繋がっていると考えられるのである*6。

もちろん、普通はそこまでやる必要がなく、企業領域の中を把握すれば十分である。業務改革やシステム開発の規模によっては、今回扱ったような、卸売りの範囲だけに限定してもよいだろう。

ともかくにも、この段階で、問屋の内外を含めた、全体的なビジネス・アーキテクチャが描かれた。そしてそれは、いくつかのクライアント/サーバーによって構成されているものだと分かった。ビジネス・アーキテクチャにみられるクライアント/サーバー構造は、何段階かに詳細化できる、より小さなクライアント/サーバー構造の総体である。これを詳細化に分析していけば、たくさんのクライアント/サーバー構造が、分子のように絡み合っ、問屋というビジネスを成り立たせているという全体構造が明らかにできるだろう。

8 . おわりに

今回、「会社全体のビジネス」という視点から演繹を開始し、途中から折り返して帰納するという方法で、問屋さんのビジネス・アーキテクチャを導き出した。しかし、これは説明の展開上、たまたま踏んだ道筋であって、必ずしも、実際の業務分析（定義）作業でたどるべき手順ではない。

実際の業務分析（設計）では、対象となる領域の大きさも位置付けも、不定である。対象のビジネス領域を詳細化するだけで、必要十分なアーキテクチャを把握できるかもしれないし、業務に精通していれば、若干の資料とヒアリングだけで、いきなり全体アーキテクチャを描けてしまうかもしれない。だから、分析の手順も場合によりけりで、一概にきめられないのである。

機能から入ると、全体が見えない。組織から入ると、部門にクローズした“ぶつ切り業務”の深みにはまってしまう。システムはスパゲティのようだ。手順やフローは得てして、大昔のビジネス・ルールや、古くからの単なる慣例などをいっぱい抱え込んでいる。データ分析から入ると、種類と量と例外と重複が多く、大抵は、とんでもない量があって、しかもグジャグジャになっている。

ただ一つの原則は、ビジネス処理の相互関係から入ることだ。これが、システム・アーキテクチャとビジネスのアーキテクチャを効果的に結び付ける、唯一の方法だと考えられるからである。

* 6) それを考えると気が遠くなりますが...。(+ + ;)