

# 統合型 CASE 評価法に関する一考察

## — 開発工程間を引き継ぐ情報を可視化 —

藤 尾 好 則

本論文は、事務処理系の情報システムを構築する場合に使用する統合型 CASE について上流から下流工程まで引き継がれる情報の流れに着目、その自動化の程度を可視化して評価する方法である。

まず強度付き相関マトリックスを用いてエンティティと処理をモジュール化する方法を示す。次に統合型 CASE を用いて事務処理系の基幹となる情報システムを構築する場合の開発工程の概要を説明する。

情報システム・アーキテクチャ、すなわち横軸に情報システムの開発工程を記述し、縦軸に開発工程で使用する主要な情報をエンティティ・タイプとして記述したマトリックスを作成する。次に開発工程とエンティティ・タイプの相関が一番高くなるように、提示したモジュール化の方法に基づき縦軸のエンティティ・タイプを入れ替えて情報システム・アーキテクチャを完成する。

情報システム・アーキテクチャを用いれば、情報の流れや CASE による自動化の範囲を可視化して表すことができる。このため統合型 CASE を評価・考察するには効果的な手法である。この手法は CASE ツールを開発する場合に最適な設計自動化の範囲を決定したり、開発対象システムに応じて開発工程を統合するなどツールや開発方法論を改善することにも応用できる。

## 1. はじめに

事務処理系の情報システムを構築するとき、開発方法論とその方法論に基づいた統合型 CASE ツールを使用する。統合型 CASE は上流から下流工程まで一貫して設計を支援することを特徴とする。一貫して設計を支援する度合いが高い CASE は設計の自動化が多くなされているツールである。しかし現状では CASE による設計の自動化の度合いを客観的に比較評価する方法がない。<sup>1)</sup>

情報（データ）を意味的に独立した固まりとして取り出したものをエンティティと定義する。本論文は情報システムの開発に必要なエンティティを抽象化してまとめたエンティティ・タイプに関し、CASE 中の開発工程の間を引き継がれるデータとアクティビティを可視化する方法を「4. 情報システムアーキテクチャの作成」で示す。又、可視化したデータとアクティビティに基づき統合型 CASE の設計の自動化の度合いを評価する。

CASE により情報システム構築の開発工程が自動化されている度合を具体的に検証するには、情報システム構築の2大要素であるデータとアクティビティに関するエンティティ・タイプが、どの工程の機能で CASE に入力され、どのような工程の機能でエンティティに情報が付加され、その成果物がどの工程の機能で出力されるかを図示して確認することである。多くの主要なエンティティ・タイプが上流工程から CASE に取り込まれ、下流工程まで連続的に処理される CASE ほど設計の自動化の度合いが高い。又、複数の CASE を用いて開発工程を自動化している場合は、エンティティ・タイプが複数のツール間を連続的に自動化されて処理される割合である。

又、この情報システム・アーキテクチャは、情報戦略計画のためのツールである相関マトリックス<sup>2)</sup>を発展させたものである。

以上が結論であるが、ここに至るまでの記述の展開を次に述べる。

現在、開発対象としている事務処理系の情報システムはデータ中心設計が中心で、このような情報システムを取り上げる。

データとアクティビティの流れを可視化するにはエンティティ・タイプと開

発工程の関連を調べる。これまで構造化設計のプログラム設計においモジュール化<sup>3) 4)</sup>の重要性が論じられてきたが、データ中心設計における可視化作業にも開発工程と対応したエンティティ・タイプをモジュール化 (またはグループ化) することが重要である。

論理的に数値化してモジュール化する方法をエンティティと処理を関連づけて「2. 相関マトリックスによる処理の統合」で示す。この関連づけの方法を「4. 情報システムアーキテクチャの作成」ではエンティティをエンティティ・タイプに対応させ、処理を開発工程に対応させてモジュール化を計っている。

## 2. 相関マトリックスによる処理の統合

システムの処理を統合する場合、どの処理を互いに統合すれば効果的であるかを決定するために相関マトリックスを用いる。

エンティティを縦軸 (Y軸) に、処理を横軸 (X軸) とするマトリックスを作成し、各エンティティがどの処理で作成 (CREATE = C)、更新 (UPDATE = U)、読取り (READ = R)、削除 (DELETE = D) されるかをマトリックスに記述する。例えば図1のA点は座標 ( $X_i$ 、 $Y_i$ ) で表される。具体的には  $X_i = 2$ 、 $Y_i = 5$  は座標 (2、5) である。又、エンティティに対する処理の作用である C、U、R、D をエンティティに対する処理作用の強度係数 (重み付け) に置き換える。C の強度係数を  $m$ 、U の強度係数を  $n$ 、R の強度係数を  $o$ 、D の強度係数を  $p$  として、 $m = 3$ 、 $n = 2$ 、 $o = 2$ 、 $p = 1$  の値を与える。作成 (C) はエンティティが存在するためには必須の処理であり、エンティティへのアクセスが複雑 (アクセス回数が多い等) で迅速性が要求されるため削除 (D) の3倍の強度係数値とした。更新 (U) と読取り (R) は、作成 (C) に較べエンティティへのアクセスが複雑でないため削除 (D) の2倍の強度係数値とした。すなわち相関係数値 (相関の強さ) を削除 (C) を1個分とした個数に置き換える。作成 (C) は削除 (D) の3倍で3個分、更新 (U) と読取り (R) は削除 (D) の2倍で2個分に相当するものとする。

エンティティ (0, 0)	処理							
	発注する 1	給料を再検討 2	仕入先を追加 3	販売する 4	利益を再検討 5	仕入れる 6	社員を採用 7	社員が退社 8
仕入 1				D		C		
販売 2				C				
利益 3					U		C	D
発注 4	C					D		
給与 5		U					C	D
人 6		R			R		C	U
製品 7	R		R	D		C		
仕入先 8	R		C					

Y A点  $(X_i, Y_i) = (2, 5)$   
 相関係数は  $\gamma = -0.279$  である。

図1 処理／エンティティ相関マトリックス

C、U、R、Dを記述した座標を元に、エンティティ（X軸）と処理（Y軸）の相関係数  $\gamma$  を求める。

$$\gamma = \frac{\sum_i F_i (\bar{X}_i - X)(\bar{Y}_i - Y)}{N \delta_x \delta_y}$$

$F_i$  は強度、 $N$  は集団の大きさ、 $\delta_x$ 、 $\delta_y$  は  $X$ 、 $Y$  についての標準偏差である。但し、 $X_i$  における強度  $F_i$  は  $C = m$ 、 $U = n$ 、 $R = o$ 、 $D = p$  である。

$$N = m \times C \text{ の個数} + n \times U \text{ の個数} + o \times R \text{ の個数} + p \times D \text{ の個数}$$

$$\delta_x = \sqrt{\frac{\sum_i (\bar{X}_i - X)^2 F_i}{N}} \quad \delta_y = \sqrt{\frac{\sum_i (\bar{Y}_i - Y)^2 F_i}{N}}$$

次にX軸のエンティティの項目間を互いに入れ替え、またY軸の処理の項目も互いに入れ替えて一番相関係数が大きい値になるように組み替える。このようにして図2に示すように項目をグループ化する。このグループ（図2のaとbのグループ）を1つの処理単位とすればグループ内は密結合、グループ間は粗結合となり効果的に処理を統合できる。

処理 エンティティ (0,0)		仕入先を追加 1	発注する 2	仕入れる 3	販売する 4	社員を採用 5	給与を再検討 6	利益を再検討 7	社員が退社 8
仕入先 1	C	R							
発注 2		C	D						
仕入 3			C	D					
製品 4	R	R	C	D					
販売 5				C					
人 6					C	R	R	U	
給与 7					C	U		D	
利益 8					C		U	D	

相関係数は  $\gamma = 0.822$  である。

図2 処理／エンティティ相関マトリックス（グループ化後）

### 3. 統合型 CASE を用いた設計

図3に示す計画工程から詳細工程までの各開発工程について、統合型 CASE を用いて設計する場合の主要な設計手順を説明する。<sup>5)</sup>

#### (1) 計画工程

環境把握では企業環境／経営方針／目標／事業環境や現状システムの把握、問題点／ニーズ／制約条件の予備調査を行う。さらにシステム化検討／企画書において、新システムの起案やプロジェクト計画を作成する。

#### (2) 分析工程

現状分析では現状調査／分析、機能情報関連分析を行い問題点／ニーズを明らかにする。

新システム要求仕様作成は統合型 CASE を用いた設計の中心に位置付けられ、現状分析の結果に計画工程で検討した新システムの企画内容を反映して新システムの要求仕様を作成する。

- ①情報構造設計は、機能情報関連分析で抽出した概念ファイルに要件を反映してエンティティ関係図を作成する。
- ②機能情報関連設計は、現状の機能階層や機能に新システムに対する要件を反映し機能階層図や機能情報関連図を作成する。
- ③業務遂行要求、④処理要求、⑤ヒューマンインタフェース (H I) 要求
- ⑥DB／ファイル要求を明確化にしてシステム要求仕様書を作成。

#### (3) 概要設計工程

システム要求の実現方法の明確化してプログラムレベルまで機能の分割を行う。

- ①プログラム・パターン (P P)、部品の適用・設計はプログラム・パターンや部品の機能を調べ「適用か?、カスタマイズか?、新規に作成するか?」を決定する。
- ②処理設計は機能階層図、機能情報関連図をもとに処理の流れを計算機処理フロー図として作成。さらに詳細化して複数プログラム間の関連等を明らかにしてプログラム網図を作成する。

③ヒューマンインタフェース（HI）設計、④DB／ファイル設計。

(4) 詳細設計工程

プログラム設計では各プログラムの内部構造を設計、必要な機能を追加してプログラム仕様書を完成する。

#### 4. 情報システム・アーキテクチャの作成

統合型 CASE を用いた設計法の各開発工程の主要な作業内容と、情報システム開発で使用される情報（エンティティ・タイプ）との相関関係を表すマトリックスを作成する。ここで示すエンティティ・タイプはエンティティ間で類縁性が高いもの同士を汎用化して、まとめたものである。図3はN社の統合型 CASE に関して作成した開発工程／エンティティ・タイプ相関マトリックスである。この相関マトリックスは開発工程がエンティティ・タイプにどのように作用するかを表す記号、作成（C）、更新（U）、読取り（R）、削除（D）を記入している。図3の開発工程とエンティティ・タイプ間の相関係数は  $\gamma = 0.753$ 。

この相関マトリックスにおいて「2. 相関マトリックスによる処理の統合」で説明した方法に基づき相関係数を用いて相関が一番高い開発工程／エンティティ・タイプ相関マトリックスを作成する。これを情報システム・アーキテクチャと呼ぶ。具体的に相関が一番高い開発工程／エンティティ・タイプ相関マトリックスを作成する方法について述べる。図3に示すように横軸（X軸）は上流の開発工程の機能を左側に記述し、右側に行くに従って下流の開発工程の機能を記述している。本論文では横軸の開発工程は固定しているものとした。相関を一番強くするに、縦軸（Y軸）のエンティティ・タイプを入れ替える。上流の開発工程で作成・更新するエンティティ・タイプを上段に、下段に行くに従って下流の開発工程で作成・更新するエンティティ・タイプになるように並べ換える。又、相関係数はエンティティ・タイプの並べ替えを理論的に行うために用いており、並び替えを自動化することも可能である。

このようにして、論理的な結びつきが強いエンティティ・タイプ同士をグ

開発工程	計 画		分 析				概 要 設 計				詳細		
	環境把握		現状分析		新システム要求仕様書		システム設計						
	経営方針／目標／事業環境	現状システム	問題点／ニーズ／制約条件	システム化検討／企画書	現状調査／分析	機能情報関連	問題点／ニーズ	システム作成	新システム設計	システム要求仕様書		システム要求仕様書	システム設計
エンティティ・タイプ	C												
企業目的	C												
成功要因	C												
技術	C												
情報技術	C												
組織単位	C												
企業所在地	C												
ファンクション(機能)	C												
サブジェクト・エリア	C												
エンティティ・タイプ	C												
関係	C												
プロセス	C												
現行システム	C												
現行データストア	C												
情報	C												
情報ニーズ	C												
現行プロシジャ	C												
システム概念モデル	C												
エンティティ・サブタイプ	C												
属性	C												
イベント	C												
基本プロセス	C												
外部オブジェクト	C												
情報フロー	C												
ビジネス・モデル	C												
ビジネス・システム	C												
プロシジャ	C												
プロシジャ・ステップ	C												
対話フロー	C												
フロー条件	C												
プログラム仕様	C												
ワールド	C												
位置情報	C												
特性	C												

図3 開発工程／エンティティ・タイプ関連マトリックス (N社の統合型CASE)



ループ化することができる。

図4は図3のエンティティ・タイプをグループ化して作成した情報システム・アーキテクチャである。相関係数は最大の $\gamma \approx 0.830$ 。

一方、図7はT社の統合型 CASE のエンティティ・タイプをグループ化した情報システム・アーキテクチャである。<sup>6) 7)</sup> 図7の相関係数は $\gamma \approx 0.815$ 。

## 5. 情報システム・アーキテクチャを用いた統合型 CASE の評価

情報システム・アーキテクチャでは、開発プロセスを構成する開発工程と引き継がれる情報の関係が上流から下流まで整然とまとめられている。このため図4、図7の情報システム・アーキテクチャを用いれば情報（データとアクティビティ）の流れを容易に追跡できる。

### (1) データの流れ

情報フローが多く情報システムアーキテクチャが見にくいいため、データの流れに関係しない情報を隠蔽して図4から図5、図7から図8の情報システム・アーキテクチャ（データの流れ）を作成する。

図5について考察する。黒で塗りつぶした箇所がデータの流れにおいて、この CASE で自動化された機能部分である。計画工程では企業情報や現行データストアのデータを抽出して、情報ニーズを加味してエンティティ・タイプを作成する。エンティティ・タイプと現行プロシジャを合わせて後続のシステム概念モデルを作成する。さらに情報構造関連設計ではエンティティ・サブタイプや属性を追加してエンティティ関係図を完成する。これはシステム要求仕様書やDB/ファイル設計に使用される。

次に図8について考察する。図5と同様に黒で塗りつぶした箇所がデータの流れにおいて、この CASE で自動化された機能部分である。計画工程では、企業情報からデータに関する情報を抽出してサブジェクト・エリア図を作成する。このサブジェクト・エリア図からエンティティ関係図を作成して後続の工程で詳細化し、エンティティ・サブタイプや属性を追加する。エンティティ関係図を検証するため、現行システムと突き合わせる。エンティティ関係図はプ

□ ツールによる自動化  
 □ 自動化されず

開発工程	計		画		分			析			概要設計			詳細	
	環境把握		システム化検討/企画書		現状分析		新システム要求仕様書		システム要求仕様書			システム設計			プログラム設計
	経営方針/目標/事業環境	現状システム	問題点/ニーズ/制約条件	システム化検討/企画書	現状調査/分析	機能情報関連	問題点/ニーズ	システム構成	システム基設計	システム仕様書	システム要求	DB/ファイル設計	HI設計	処理設計	PP/部品の適用・設計
エンティティ・タイプ															
企業目的	C														
成功要因	C														
情報技術	C														
組織	C														
企業所在地	C														
企業所在地	C														
ファンクシヨ(機能)															
現システム															
現システム															
現システム															
情報ニーズ															
サブジェクト・エリア															
エンティティ・タイプ															
関係															
プロセス															
現システム															
現システム															
システム概念モデル															
エンティティ・サブタイプ															
属性															
基本プロセス															
情報フロー															
イハント															
外部オブジェクト															
ビジネス・モデル															
ビジネス・システム															
プロシジャ															
プロシジャ・ステップ															
対話フロー															
フロー条件															
フィールド															
位置情報															
特性															
プログラム															

図4 情報システム・アーキテクチャ (N社の統合型CASE)

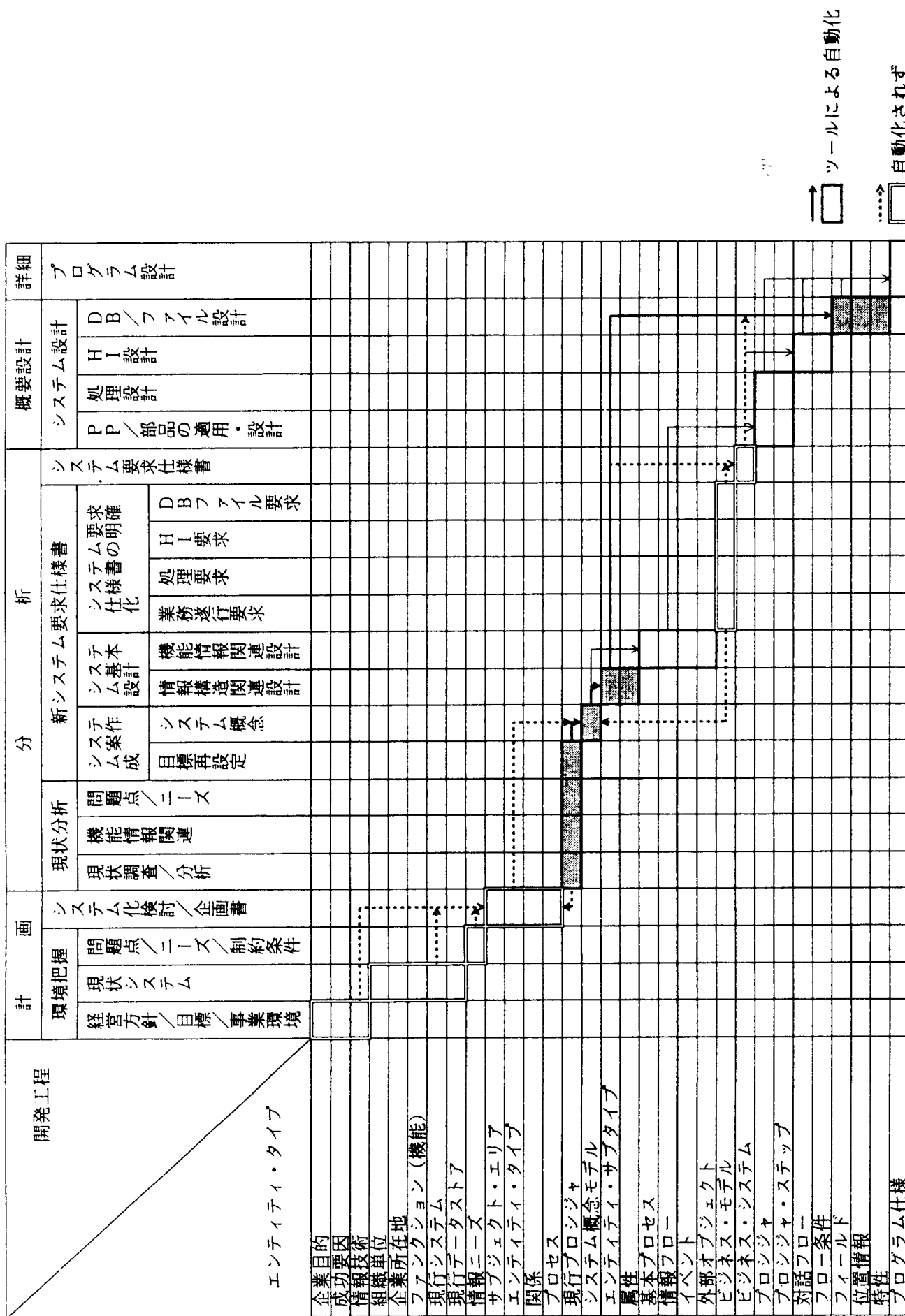


図5 情報システム・アーキテクチャ (データの流れ)

ロセス論理を記述するアクション図の作成、画面や帳票のレイアウト設計に使用される。

### (2) アクティビティの流れ

アクティビティに関係しない情報を隠蔽して図4から図6、図7から図9の情報システム・アーキテクチャを作成する。

図6について考察する。黒に塗りつぶした箇所がアクティビティの流れにおいて、このCASEで自動化された機能部分である。計画工程では企業情報や現行システムからアクティビティに関する情報を抽出して情報ニーズを加味し、プロセスの機能階層図を作成する。現状分析において現行プロセッサと突き合わせ具体化し、さらに要求機能を反映したシステム概念モデルを作成する。機能情報関連設計では基本プロセスを抽出して機能情報関連図を作成。この基本プロセスは実行単位として処理設計においてプロセッサに組み込まれたり、パターンや部品に置き換えられる。さらにH I設計の対話フローによって実行順序を決定してプログラム仕様に引き継ぐ。

次に図9について考察する。計画工程では企業情報からアクティビティに関する情報を抽出してファンクション階層図を作成する。この階層図からプロセス階層図やプロセス依存関係図を作成。プロセス階層図は分解して基本プロセスを抽出する。現行システムと突き合わせプロセス依存関係図を検証する。基本プロセスは、アクション図によってプロセス論理を記述して実行単位とし、プロシッサやプロシッサ・ステップに組み入れられる。プロシッサやプロシッサ・ステップは対話フローによって実行順序が決定される。

### (3) CASE ツールによる設計自動化の範囲

情報システムアーキテクチャはCASE ツールによって自動化される開発工程を可視化できる。このためCASE ツールによる設計自動化の範囲を比較する場合にも利用できる。このときエンティティ・タイプは出来るだけ同じものを用い、開発工程はそれぞれの方法論の開発工程を記述する。<sup>注1)</sup>

図4を考察するとN社の統合型CASEの計画工程は自動化されていない。分析工程から詳細設計工程のプログラム作成に至る範囲が自動化されている。又、システム要求を取り込みビジネス・モデルを作成する手段などは自動化さ

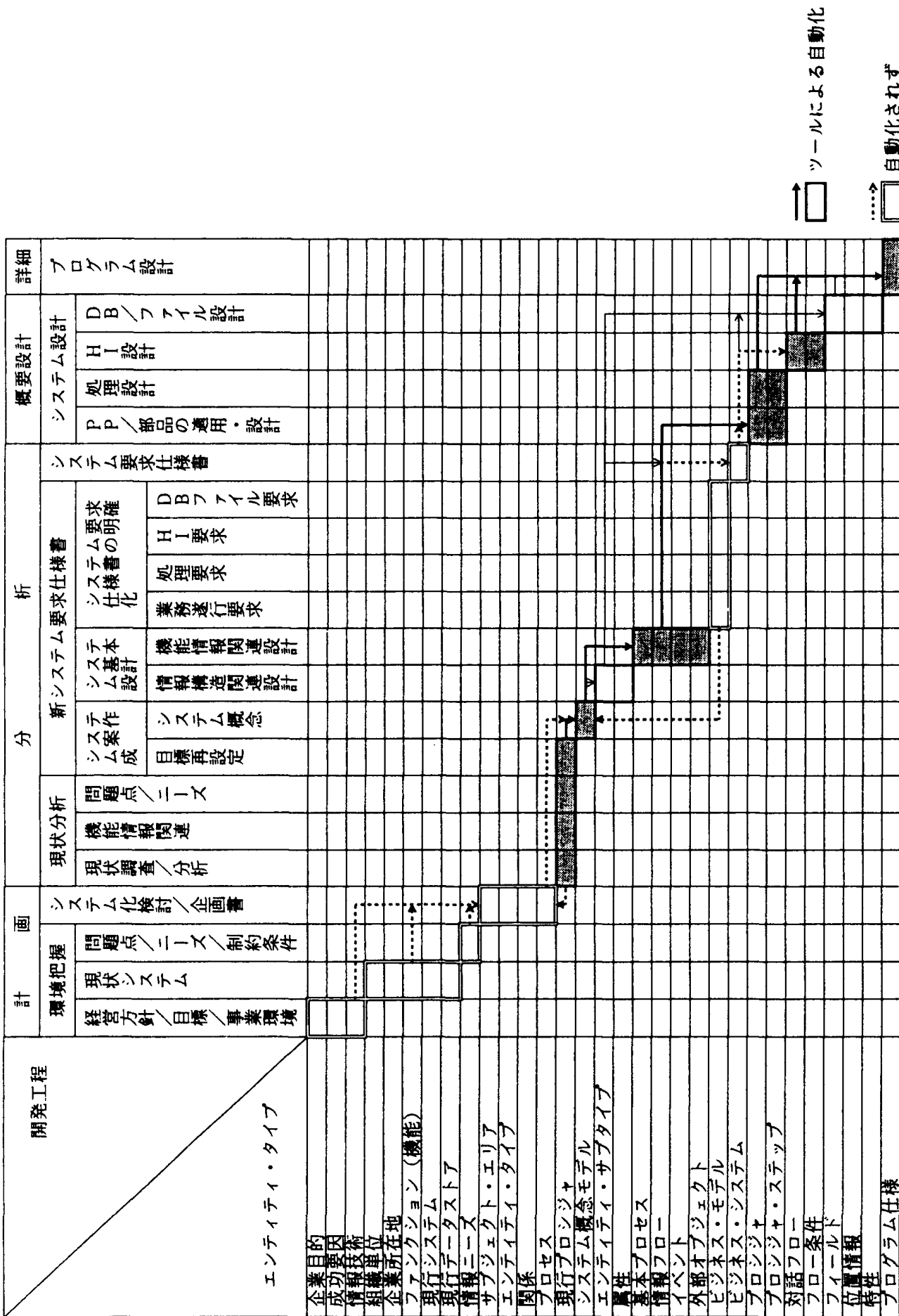


図6 情報システム・アーキテクチャ (アクティビティの流れ)

ツールによる  
自動化  
自動化されず



開発工程	計			画			分			析			設			計				
	企業情報の調査	情報アーキテクチャの特徴	現行システムの調査	業務アーキテクチャの作成	技術アーキテクチャの作成	計画プロジェクトの終了	分析プロジェクトの準備	データ分析	アクティビティ分析	相互作用分析	現行システム分析	ビジネス・モデルの検証	ビジネス・システムの定義	設計プロジェクトの準備	プロシヤの定義	対話設計	レイアウト設計	プロトタイピング	プロシヤ論理の設計	設計プロジェクトの完了
エンティティ・タイプ																				
企業目的	C																			
成功要因	C																			
組織単位	C																			
企業所在地	C																			
企業ニーズ	C																			
技術情報	C	R																		
フロンクシヨン		R	U																	
サブジエクト・エリア		U	R																	
エンティティ・タイプ		R	R																	
関係		C	C																	
プロセス		C	C																	
現行システム		C	C																	
現行データストア		C	C																	
ビジネス・エリア		C	C																	
エンティティサブタイプ		C	C																	
属性																				
基本プロセス																				
イベント																				
外部オブジェクト																				
情報フロー																				
アクション文																				
現行プロシヤ																				
ビジネス・モデル																				
ビジネス・システム																				
プロシヤ																				
プロシヤ・ステップ																				
対話フロー																				
フロー条件																				
フィールド																				
位置情報																				
特性																				

図7 情報システム・アーキテクチャ (T社の統合型CASE)

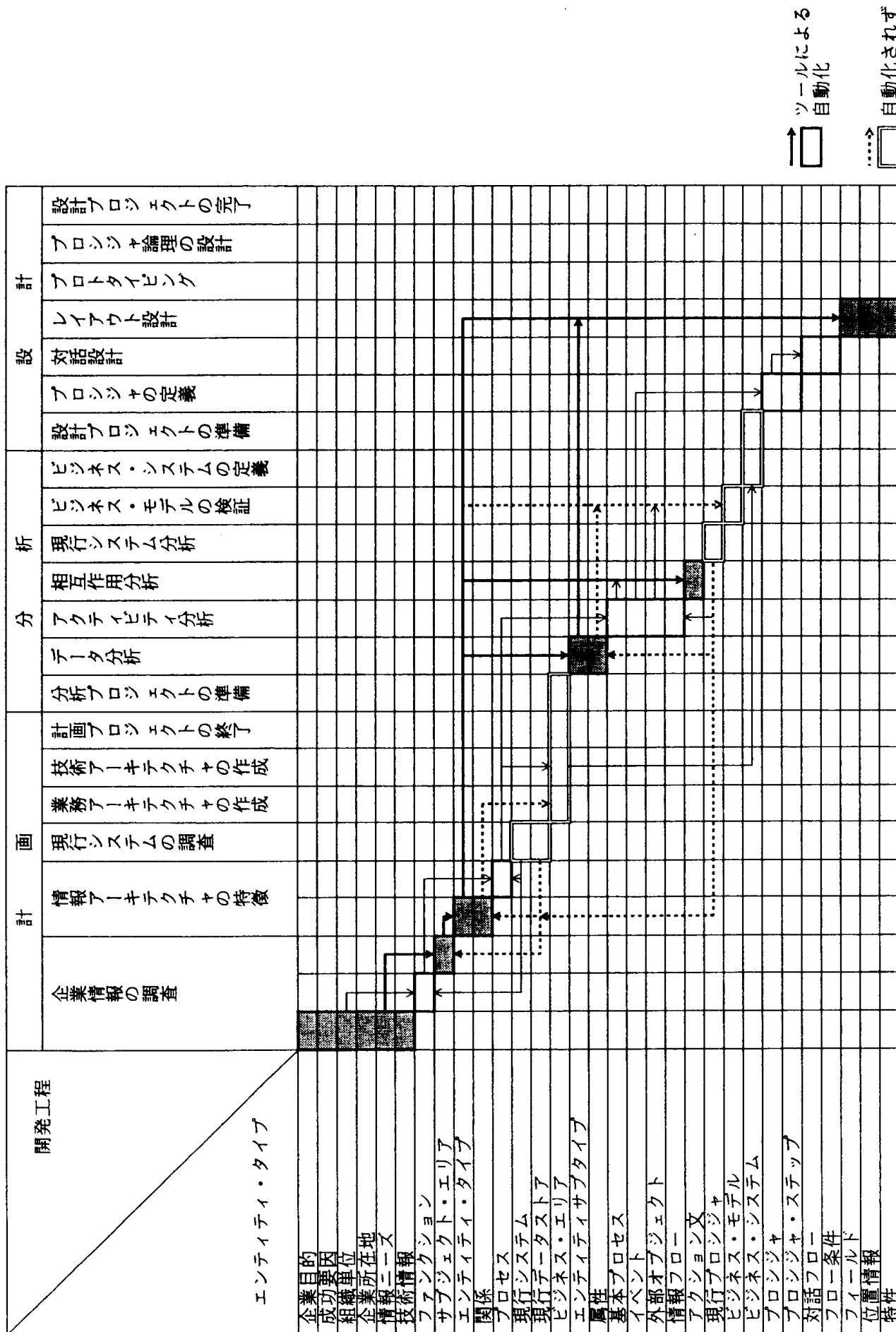


図8 情報システム・アーキテクチャ (データの流れ)

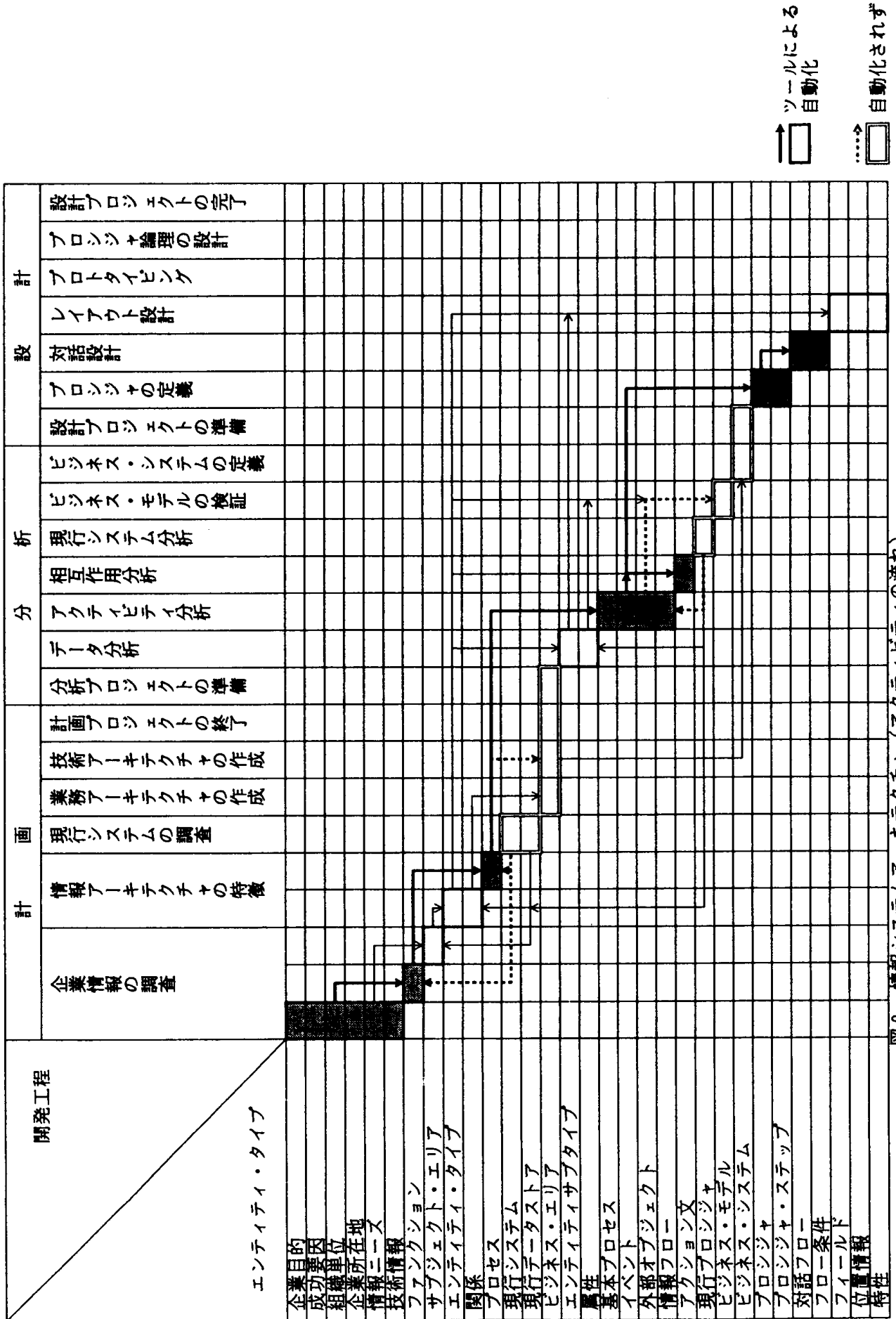


図9 情報システムアーキテクチャ (アクティビティの流れ)



れていないことが分かる。情報の流れから判断すると現状分析に重きを置き、既存の AP を活用する生産性を重視したツールであることが読み取れる。

次に図 7 の T 社の統合型 CASE は計画工程から設計工程に至るまで自動化されている。しかし、現行システムのプロセスを作成するための手段などが自動化されていない。計画工程に重点を置く情報システム構築ツールであることが分かる。<sup>8)</sup>

このように開発工程とエンティティ・タイプの対応関係を、情報システム・アーキテクチャを用いて表現すれば、データの流れやアクティビティの流れを図示して各開発工程の自動化の程度を可視化することができる。この可視化の手法を他の CASE にも適用して CASE の自動化の程度を比較できる。

## 6. おわりに

このようにして作成した情報システムアーキテクチャはデータとアクティビティに関する情報が上流から下流工程に CASE の中を引き継がれる様子を図示できる。このため情報システムアーキテクチャを用いて統合型 CASE を、次のようにして比較評価することができる。

① CASE ベンダー各社のツールを情報システム・アーキテクチャ上にマッピングして CASE を比較する。

- ・ 主要なエンティティ・タイプに関し、情報システム構築の 2 大要素であるデータとアクティビティの流れに着目して比較する。
- ・ CASE の元になっている方法論の内部関連を図示できるため方法論を体系的に外観する。

② CASE の設計自動化の程度を明確にする。

- ・ 何が（どのエンティティ・タイプに関する成果物）自動化されていないかにより CASE 自動化の程度を評価する。
- ・ 複数の CASE を組合せて情報システムを開発する場合においても、各 CASE や開発工程間をとおして設計自動化の程度を把握できる。

開発工程／エンティティ・タイプのような相関マトリックスは、一般に開発

工程等のように順序性がある項目とエンティティ・グループ等のように順序性が無い項目との間の相関を強くするため、順序性がない項目の順序を入れ換え相互の関連を明確にする場合に効果的な方法である。

しかし相関の強度係数は項目相互の処理作用の強さを表すもので、項目間における処理や作用の複雑性、迅速性、順序性、必要性等を多面的に考慮して決定すべきである。強度係数の値を目的に応じてどのように設定するかは今後の課題である。

相関マトリックスに関し、開発工程の順序性を考慮しなくてよい場合はエンティティ・クラスをまとめた仕方と同様にして開発工程をグループ化することができる。グループ化された開発工程を単位にツールを作成すればツール内は密結合であるが、ツール間のインターフェースは粗結合になる。このためCASE ツールを作成するとき、ツールが自動化する開発工程の範囲を決めたり、既にあるツールをどのようにして結合すればよいかを決定する場合にも活用できる。さらに発展させれば、開発方法論そのものの評価や改善の足がかりを示すものとなる。

注1) N社の統合型CASE (CASEWORLD) 及びT社の統合型CASE (IEF) の開発工程は大枠で参考文献<sup>9)</sup>の体系に基づき工程のレベルでそろえている。

#### 参考文献

- 1) 平成3年度シグマ会 CASE 委員会編：日本及び米国におけるCASE 導入・利用実態調査報告書、株式会社シグマシステム、東京(1992)
- 2) J. Martin: *Information Engineering Book II*, Prentice - Hall International Editions, New Jersey (1990)
- 3) G. J. Myers: *Reliable Software Through Composite Design*, Petrocelli/Charter, New York (1975)
- 4) G. J. Myers: *Composite/Structured Design*, Litton Educational Publishing, (1978)
- 5) 日本電気株式会社著：システム開発方法論 STEPS (作業標準・作業手順編), 日本電気株式会社, 東京(1990)
- 6) Texas Instruments, Inc. 著, 日本テキサス・インスツルメンツ株式会社訳：CASE 実用ガイド, 日経 BP, 東京 (1992)
- 7) 92年度シグマ会 CASE 委員会編：開発技法検討部会報告書, 株式会社シグマシステム, 東京(1993)
- 8) 藤尾好則：「日本の情報システム開発方法論の研究」、1995情報学シンポジウム講演論文集、日本学術会議(1995)
- 9) 共通フレーム検討委員会編：システム作業体系と管理基準に関する調査研究報告書、情報処理振興事業協会(1994)