同時エンジニアリングの為の情報統合

IDEF3　プロセス記述獲得手法に関する報告

Information Integration for Concurrent Engineering (IICE)

IDEF3 Process Description Capture Method Report

Richard J.Mayer, Ph.D.
Christopher P.Menzel, Ph.D.
Michael K.Painter
Paula S.deWitte, Ph.D.
Thomas Blinn
Benjamin Peralath, Ph.D.

Knowledge Based System, Incorporated
Human Resources Directorate Logistics Research Division

September 1995

Interim Technical Report For Period April 1992 - September 1995

この文章は、プロセス記述獲得の為の IDEF（Integration Definition）手法（IDEF3）の概要と実践（practice)、記述の使用、そして言語についての参照に関するものである。IDEF3 は、既存もしくは作成予定のシステムの、プロセスのドキュメント作成と分析を補助する為のものである。確かなガイドラインと情報獲得の為の言語により、プロセス情報の獲得と、下流での使用の為の組織化を補助している。IDEF3 は、プロセス中心並びにオブジェクト中心での、知識の獲得戦略を支援している為、人間的表現の共通形式それぞれに並行なやり方で、現実世界のプロセスとイベントについての主張（assertion）を、ユーザーが獲得することが可能である。IDEF3 は、システムがどのように機能しているかを複数の視点から獲得し、構造的に記述する能力も持っている。そして他の IDEF ファミリー手法のように、IDEF3 単独でも、他の IDEF 手法との併用でも、経営に重要なプロセスのドキュメント化、分析、そして改善を行う事が可能である。

目次：

第１章	はじめに	8
１．１	動機	8
１．２	IDEF3 の設計特性	11
１．３	適応性	15
１．４	利益	16
１．５	論文の構成	17
１．６	概要	18
第２章	IDEF3 の概要	19
２．１	シナリオ：IDEF3 プロセス記述の為の、組織化構造	19
２．２	プロセス中心視界：プロセススキーム図	21
２．３	オブジェクト中心視界：オブジェクトスキーム図	25
第３章	IDEF3 プロセス記述言語	30
３．１	IDEF3 プロセス記述の基本要素	30
３．２	プロセススキーム図	33
３．２．１	行動ユニット（UOB）	33
３．２．２	リンク	35
３．２．２．１	単純順序リンク	36
３．２．２．２	活動のプロット	37
３．２．２．３	制約順序リンク	38
３．２．２．４	点線リンク	40
３．２．２．５	リンク番号	41
３．２．２．６	交差点が無いプロセススキーム図の活動意味論	41
３．２．３	交差点	45
３．２．３．１	交差点の種類	45
３．２．３．２	基本的な交差点の文法	47
３．２．３．３	交差点番号の採番要領	48
３．２．３．４	基本的な交差点の意味論	49
３．２．３．５	交差点の連結	52
３．２．４	具体的な例	59

３．２．５	UOB の分解	62
３．２．６	UOB 参照番号の採番要領	65
３．２．７	不完全な記述	67
３．２．８	参照指示	68
３．２．８．１	呼出続行参照指示	70
３．２．８．２	呼出待機参照指示	73
３．２．８．３	IDEF3 プロセススキーム図での、参照指示の使用	76
３．３	オブジェクトスキーム図	78
３．３．１	オブジェクトとオブジェクト状態	78
３．３．２	変移スキーム図	80
３．３．３	条件	83
３．３．４	IDEF3 オブジェクトスキーム図での参照指示の使用	85
３．３．４．１	変移リンクに取り付けられた参照指示	86
３．３．４．２	オブジェクト状態に取り付けられた参照指示	94
３．３．４．３	複数の参照指示を持つオブジェクトスキーム図	95
３．３．５	複合変移スキーム図	97
３．３．６	変移交差点	100
３．３．７	オブジェクト状態情報の遮蔽	104
３．３．８	拡張変移スキーム図	105
３．３．８．１	一階スキーム図	106
３．３．８．２	構成スキーム図	114
３．３．８．３	二階スキーム図	117
３．３．８．４	分類スキーム図	120
３．３．８．５	構成と分類情報の遮蔽	125
３．３．８．６	拡張変移スキーム図の作成	127
３．４	詳細説明	130
３．４．１	詳細説明言語の幾つかの例	133
３．４．２	プロセススキーム図での詳細説明言語の使用例	134
３．４．３	オブジェクトスキーム図での詳細説明言語の使用例	138
３．５	注記	141
３．６	確率論的プロセスの表現	143

第４章	IDEF3 記述の開発	144
４．１	IDEF3 記述進化サイクル	144
４．２	IDEF3 記述獲得活動	146
４．２．１	プロジェクトの定義	146
４．２．２	目的の定義	148
４．２．３	背景の確立	150
４．３	データ収集の為の組織化	150
４．４	データの収集と分析	152
４，４．１	インタビューの準備	153
４．４．２	分野専門家へのインタビュー	156
４．４．３	オブジェクト名の収集	156
４．４．４	活動名の収集	157
４．４．５	プロセス上の発生物に関連する、事実と制約の収集	157
４．４．６	状況記述の収集	158
４．４．７	素材資材の収集とカタログの作成	159
４．４．８	収集されたデータの分析	162
４．４．９	必要に応じた追加情報の取得	163
４．５	IDEF3 スキーム図をまとめる	164
４．５．１	プロセススキーム図をまとめる	165
４．５．１．１	UOB の識別	166
４．５．１．２	最初のプロセススキーム図の作成	168
４．５．１．３	UOB、交差点、リンクについての詳細説明の作成	170
４．５．１．４	選択した UOB（行動ユニット）の分解	179
４．５．２	オブジェクトスキーム図をまとめる	180
４．５．２．１	興味の対象となるオブジェクトの選択	181
４．５．２．２	最初のオブジェクトスキーム図の作画	182
４．５．２．３	オブジェクト状態、オブジェクト、交差点、リンクの詳細説明の作成	185
４．６	IDEF3 プロセス記述の漸次的な改良と検証	193
４．６．１	動機	193

４．６．２	検証の種類	194
４．６．３	キットの構築と配布	194
４．６．４	キット批評プロセスでのメンバーの役割	195
４．６．５	IDEF3 キット批評サイクル	196
４．６．６	IDEF3 キットの種類	198
４．６．７	分析者とコメンターの為のガイドライン	199
４．６．７．１	コメンターのガイドライン	199
４．６．７．２	分析者とコメンターの間のやり取り	200
４．６．７．３	会議のルール	200
４．６．８	IDEF3 キットの構成物	201
４．６．８．１	キットを準備する際の一般的なガイドライン	201
４．６．８．２	キットの表紙	202
４．６．８．３	キットスキーム図フォーム	205
４．７	批評の進展に伴う調整の実行	208
４．８	プロセス記述獲得での、他の IDEF 手法の併用	209
４．８．１	IDEF3 と共に IDEF0 を使用する場合	209
４．８．２	IDEF3 と共に IDEF1 と IDEF1X を使用する場合	210
４．８．３	IDEF3 と共に IDEF5 を使用する場合	211
第 5 章	IDEF3 の開発：資材発注プロセスの例	213
５．１	目的と背景の定義	213
５．２	データの収集	214
５．２．１	分野専門家へのインタビューと、初期記述の取得	214
５．２．２	データ識別の為の記述の分析	216
５．３	プロセススキーム図をまとめる	219
５．３．１	最初のプロセススキーム図の作画	219
５．３．２	詳細説明の作成	223
５．３．３	分野専門家によるプロセススキーム図の批評	229
５．４	オブジェクトスキーム図をまとめる	233
５．４．１	興味対象となるオブジェクトの選択	233

５．４．２　オブジェクト状態の識別　234
５．４．３　最初のオブジェクトスキーム図の作画　235
５．４．４　必要に応じた交差点の追加　236
５．４．５　参照の取付け　237
５．４．６　詳細説明の作成　238
５．４．７　分野専門家によるオブジェクトスキーム図の批評　241

第 6 章　IDEF3 プロセス記述の読解　244
６．１　記述の読解ステップ　244

６．２　IDEF3 プロセス記述の速読　246
６．２．１　大局　246
６．２．２　スキーム図の流し読み　247
６．２．３　記述の理解　247

参考文献（省略）　253

付録 A　IDEF3 の精巧言語　253
A.1　詳細説明の基本部分の記述　254
A.1.1　基本的な統語論種類　254
A.1.2　演算子　255
A.1.3　項　256
A.1.4　文　260
A.1.5　定義　263

A.2　詳細説明言語の基本部分で用いられている BNF　265
A.2.1　アルファベット　266
A.2.2　基本的な構文種類　266
A.2.3　演算子　267
A.2.4　項　267
A.2.5　文　268
A.2.6　定義　268

A.3 基本的な定義、公理、公理型 269
A.3.1 基本的な意味論分類 270
A.3.1.2 アリティ 272
A.3.1.3 Null オブジェクト 274

A.4 基本的な状況理論 275
A.4.1 状況とインフォン 275
A.4.2 種類、UOB、プロセス 278
A.4.3 基本的な状況理論での関係 280

A.5 IDEF3 詳細説明の正式な言語 282
A.5.1 詳細説明言語の基本部分の拡張 282
A.5.2 基本状況理論意味論分類 283
A.5.2.1 一階関係 284
A.5.2.2 インフォン項の公理 284
A.5.3 基本状況理論関係 286
A.5.4 基本時間関係 287
A.5.5 状況発生期間 290
A.5.6 分類変数の使用 290

付録 B IDEF3 用語集 292

第１章　　はじめに

プロセスや状況を記述する為の､最も一般的な情報伝達機構の１つが､ストーリー（story）として、イベントや活動を順番に語る事である。例えば、あるエンジニアが、最近開発された製品に関するストーリーを語る事で、そのエンジニアの会社における設計プロセスを記述する事が良くある。同様に、定盤（shop floor）の責任者は、その定盤での製品の組立プロセスを記述する事によって、その定盤の製造システムの運用について記述している。

IDEF3 プロセス記述獲得手法（Process Description Capture Method）は、活動の順番の記述を獲得する事に特化された作りになっている。IDEF3 のかかげる主な目標は、その分野の専門家が特定のシステムもしくは組織の運用に関する知識を表現可能な、構造化された手法を提供する事である。知識獲得は、現実世界のプロセスとイベントに関する主張（assertion）を、獲得するのに最も自然な形で、直接獲得することによって成し遂げられる。これには、プロセスに関係する対象物に関する主張、対象物を支援する事に関する主張、そして環境内のプロセスとイベント（event）の間にある優先度と因果関係の獲得が含まれている。

IDEF3 は、このような種類の知識の習得（aquiisition）を、信頼性が高く上手く構造化されたプロセス知識獲得アプローチと、非常に強力であるにも関わらず使いやすい情報獲得言語・情報表現言語を提供する事で、支援している。実践を確かなものにしてくれる手続と、表現豊かで強力な言語という IDEF3 の 2 つの方向性（dimension）は、それぞれが共に機能する事により、対象となるプロセスの適切な部分にユーザーの注意を集中させ、プロセスの本質と構造に関して明確に表現された情報に必要となる表現力を提供してくれる。

１．１　動機（Motivation）

一般的な意味合いにおいて、プロセスとは、イベント（event）を順番に並べただけのものである。そして人間によって作られたシステムでは、プロセスを構成している複数のイベントは、何らかの期待された成果を出すように、意図して並べられている。特にビジネスプロセスでは、人、資材、エネルギー、装備といったものをも巻き込んだイベントが、定義されたビジネス成果を達成できるように、順序良く並べられている[(Davenport &

Short, 1993, 103),(Pall, 1987)]。ビジネスプロセスの重要性は、それ自身が実証している。ビジネスプロセスは、ビジネスが何をしているのかを定義しているだけでなく、ビジネスがそれをどれだけ上手く実行しているのかも決定しているのである。

IDEF3 の開発を動機付ける幾つかの要因が存在する。中でも傑出した要因には、以下のようなものがある。

（1）ビジネスシステム分析の生産性を向上させる

IDEF3 の最も大きな開発動機の 1 つは、ビジネスシステムのモデリングプロセスの速度向上の必要性である。ビジネスシステム分析は、問題となる状況に関する正確な記述を取得することから開始されることが多い。対象となる分野専門家は、回想した状況を、順番に並べたイベントもしくは活動（activity）によって表現する。更に分野専門家は、一般的に、彼らの関わる活動と対象物を関わりあったように記述してしまう傾向にある。その為、移り変わるビジネス活動とプロセス記述の獲得を促進する「手法」と、獲得された知識を保存し、取扱う事が可能な「表現媒体」の両方が必要となる。IDEF3 は、分野専門家によって記述された、そうしたプロセス情報を伝達する為の構造化されたアプローチを準備する事で、こうした要求を満たしているのである。

（2）設計データライフサイクルの運用支援（management）の促進

手法に必要な物を識別するための初期の研究から、設計データライフサイクル（design-data life cycle）の記述の獲得手法が不足している事が判明した（Mayer 1987）。エンジニアリング設計データライフサイクルの記述には、以下の物を記述しなければならない：

①設計情報人工物（design information artifact）　図面、CAD データ等
②設計情報人工物の状況変移（state transition）
③状況変移を決定した、決定ロジックもしくは決定プロセス

IDEF3 は、このデータライフサイクル情報を記述する為の仕組みを提供してくれる。

（３）プロジェクト運用管理プロセスの支援

プロジェクト運用管理の技法（technique）は、様々な適用分野で、プロジェクトを監視、管制するために使用される。このようなプロジェクト運用管理技法を支援してくれるソフトウェアツールが幾つかあるが、その多くは、プロジェクト運用管理状況で発生する複雑性を十分に獲得できるだけの表現性を欠いている。IDEF3 は、あるプロジェクトの活動間に存在する数々の制約（リソースや、一時的な関係性を含む）を獲得する仕組みを提供してくれている。IDEF3 言語はまた、プロジェクト活動に関係する、プロジェクト活動で生成される、もしくは使用される対象物に関した、詳細な情報を表現することが可能である。更に、自動化ツール（訳注：ソフトウェア）により支援された IDEF3 ダイヤグラムを活用する事により、リアルタイムでプロジェクト活動の監視と管制を行う手段も提供されることになる。

（４）システム要求定義プロセスの促進

もう 1 つの IDEF3 の開発動機は、「システム要求記述」を構築する為に必要となる、概念、文法、手続といったものを提供する必要性があることである。こうした記述は、完成したシステムが受け入れ可能かどうかの決定を行えるように、十分に詳細化されていなければならない。こうした必要性から、IDEF3 手法は、以下に挙げるような項目についての記述を支援できなければならない。

①組織的活動のシナリオ
②そうした組織的活動内での、ユーザーの種類の役割
③ユーザー機能（function）レベルでの、ユーザーシナリオもしくは、
　情報システムとユーザーの相互作用
④ユーザー機能に対するシステムの反応
⑤ユーザークラスと、ユーザークラスの描写（delineation、表現）
⑥タイミング、順序、リソース制限の描写
⑦ユーザーインターフェースオブジェクト
　（メニュー、キーワード、スクリーン、ディスプレイ、等）

（５）支援協調活動と作業の統合（Support Coordinated Acticity and Integration of Effort）

ビジネスシステムのプロセス視界（view）は、ビジネス意思決定プロセス（decision-making process）にとって、必要不可欠（critical）である。しかし、この視界は、他の視界（機能視界、情報視界、組織視界、等）との協調が特に必要である。この必要性を考慮し、IDEF3 手法は、単独でも、異なった分野にそれぞれ集中する他の手法（IDEF0 機能モデリング手法、等）との併用でも、IDEF 手法ファミリーに補足的に付加されることで、十分に機能するように作られている。

１．２　IDEF3 の設計特性（Design Feature of IDEF3）

これまでに挙げた必要性から、手法設計者である IDEF3 チームに対して、特別な要求が出されることになった。効率的なプロセス知識の獲得と表示機構を提供する必要から、IDEF3 は以下の項目を実現できるように作られている：

（１）技法を構造化することにより、殆ど訓練されない人でも学習・使用が容易となると共に、技法を使用する事でもたらされる実践も、一貫性と高い信頼性を持つようになる。

（２）下流での使用も考慮し、プロセス情報の保存、運用管理、再利用が可能である。

（３）個人利用であっても、チームによる使用であっても、素早く、信頼性のある、コスト効率の良い運用管理に必要となる、効果的手法を提供可能である。

（４）ユーザーが、選択（alternative）プロセス構成の間に存在する重要な差異部分を認識することが可能である

（５）規模の大小に合わせて導入する事が可能であり、また詳細の度合いに依らずプロセス情報を収集可能である。

（６）エンジニアリング、製造、供給、経営といった、複数の異なるシステム分野に跨った使用においても、機能を十分に発揮可能である。

（7）他の IDEF 手法と併用する際に、統合作業を支援してくれる。

IDEF3 は、システム運用の記述獲得にユーザーを集中させ、既存のモデリング手法よりも簡易なものとし、そして収集したプロセス情報を下流で最大限に利用できるようにすることで、上に挙げた目標を達成している。このような IDEF3 の特性により、ユーザーは、モデリング特性の最適化作業に時間を浪費することも、決断に苦労することなく、ビジネスシステムの運用形態を十分に観察し、確信を得る事に集中できるのである。それとは対照的に、モデリングを意図的に行わなければならない（既存の）手法は、数学的理想形、つまりは、予め設定された状態の組み合わせの下で、システムがどのような挙動を示すかを予想する為に使用するモデルの開発を、補助するように作られている。

「記述（description）」と「モデル（model）」の違いは、微妙ではあるものの重要である。統合定義（Integration Definition）手法という観点から見ると、この 2 つの用語は、精密な技術的意味をそれぞれ持っているのである。「記述」という用語は、経験的観察物の記録という意味に割り当てられた（？reserved）技術用語である。つまり「記述」は、観察や経験そのものから得られた、もしくはそれらを基にして得られた知識が、記録されたものなのである。一方の「モデル」という用語は、出来事（affair）の実体もしくは状況を理想化した物、という意味で使用されている。

つまりモデルは、ある適切な側面で、対象となる現実世界のシステムの特性を模倣する為に作られた、対象物、特性、関係から成る、理想化されたシステムを構成しているのである。摩擦の無い平面であったり、曲りの全くない本体、点質量という仮定といったものは、この「モデル」としての特性の例を示している。

一方で記述は、個人の知識もしくは経験の分野内にある物に関する事実もしくは信念も記録されており、一般的に不完全である。何故なら、システムを記述する過程において、記述者にとって無意味だと考えた物、もしくは書き忘れてしまった物が省略されてしまうからである。更には、記述者の専門分野や状況の観察方法の違いにより、記述が異なったものになる。

IDEF3 では、このような起こり得る問題に対して、同一のシナリオもしくはプロセスに関する記述の選択物を獲得し、組織化する特性を提供することで対応している（図 1-1）。

モデリングの場合、相反したり一致しない視界（view）の解決に、記述の獲得よりも余分な手順を踏む必要がある。モデラーは単一の視点を選択・作成するか、もしくは作為的

なモデリング誤差を用いることで、このギャップを埋めざるを得ないのであるが、これでは直接的な知識や経験を入手することが不可能になる。しかし記述はモデルとは異なり、検証可能な理想化された状態である必要が無く、そして単純な意味での正確性から離れていても構わないのである。

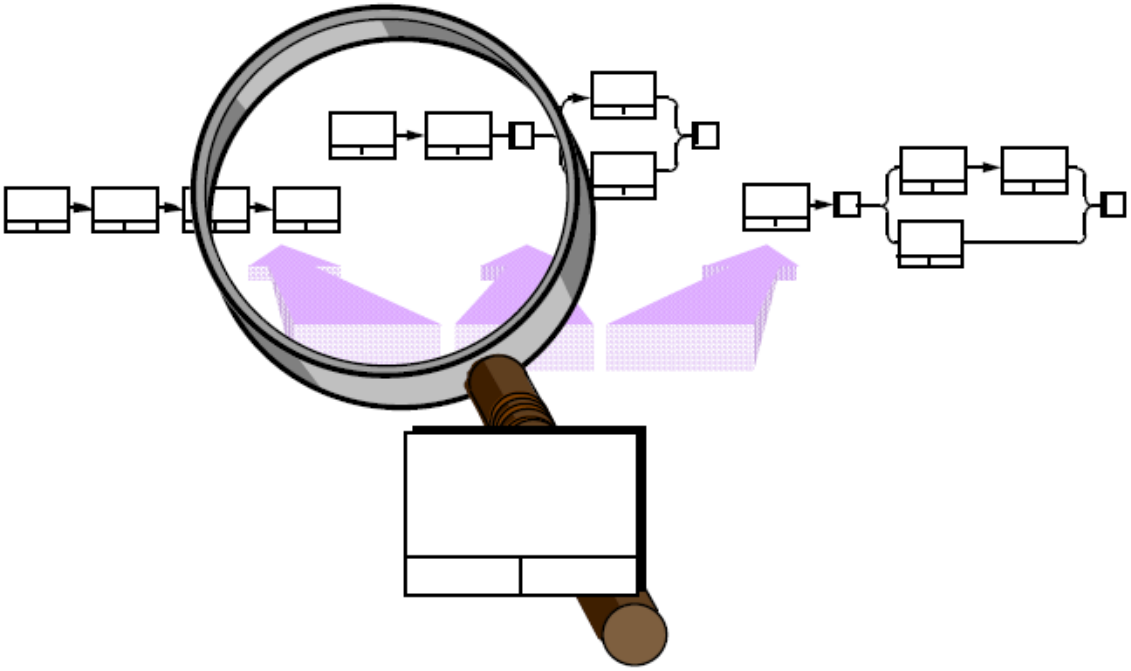


図 1-1　IDEF3 は 1 つのプロセスを複数の視点から獲得する

記述獲得の目的には、単なるプロセス知識の記録や伝達だけではなく、重要プロセスの実際の運用方法に関する認識の、人による相違を識別することも含まれている。ユーザーは記述獲得手法を使用する事で、実行可能なモデルを作成するという慣習（convention）（精度や内部整合性、論理的一貫性、非冗長性、完全性、等を保証する慣習）を学ぶことも適用することも必要なくなるのである。その上、モデルを作成するには、「モデル設計」という思考を持たなければならない上に、自身の分野の経験的知識をモデルに正確に反映できなくなるという危険性をも負う必要がある。

記述獲得はまた、モデル作成の際にも実行されることになる。暗黙的に行われる場合であっても、明示的に行われる場合であっても、記述はモデル作成の素材なのである。その為、記述の使い勝手は、複数の理想化やモデル構築での再利用を通して、実感させられることになる。

興味深い事に、モデルは記述の一形式であるが、その逆は正ではなく、記述はモデルにはならない。分析データの作成にモデルが使用されるが、記述から分析データを作成する事は不可能である。しかし、記述は分析データの一形式として扱われている。例えば、バス路線と時刻表についての記述は、公共交通システムのモデル開発用データの作成に便利であるが、記述だけでモデルを構成することはできない。同様に自動車についての記述は、例え他の用途として便利であったとしても、有限要素分析データの作成には使用できない。

モデル構築と比較すると、記述獲得（description capture）は幾つかの理由から、知識獲得（knowledge aquisition）戦略として魅力的である。まず第一に、実行者は、自身の専門分野の記述を作成するに当たり、モデルの作成と比べて、一般的により少ない訓練で実行可能である事が挙げられる。2 つ目に、対象となる状況の記述は、モデル構築（機能モデルや状況モデル等）も含めた多様な用途に再利用することが容易である。IDEF3 は、こうした必要性を直接取扱う、記述組織化と記述獲得の為の手法である。

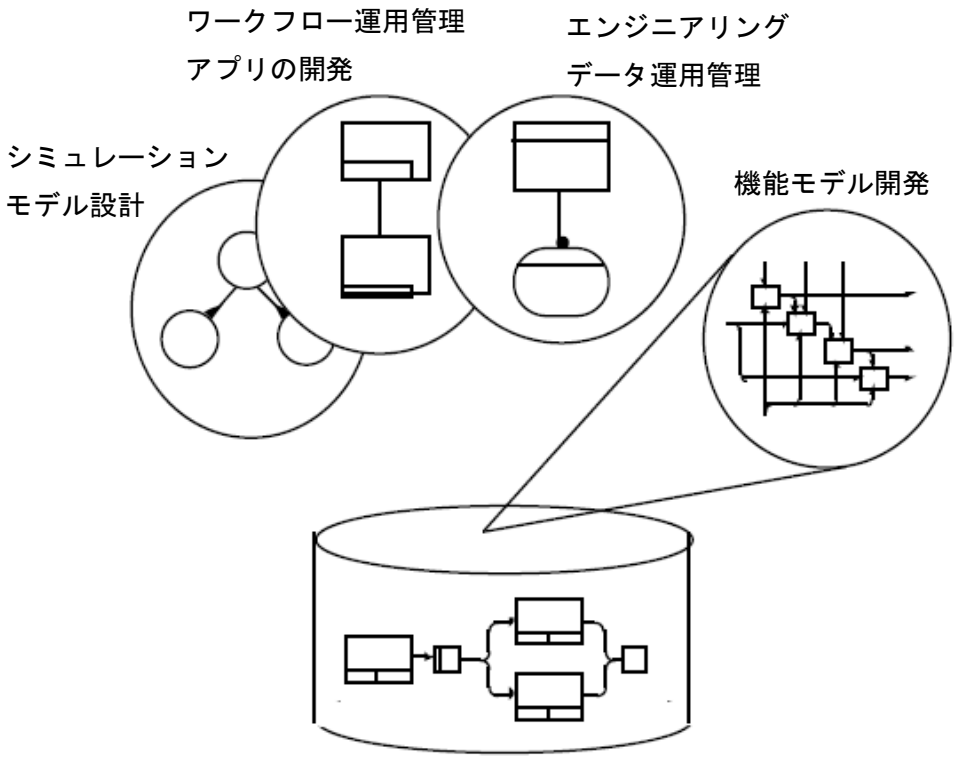


図 1-2　IDEF3 は記述獲得に焦点を置いている為、再利用性が最大となっている

１．３　適応性（Applicability）

IDEF3 は当初の目標範囲での適用が成功し、次第に適用範囲を経営の全てへと広げつつある。また、IDEF3 は図 1-3 に示すように、システム開発プロセスや経営改革プロセスを通した利用に便利であるように作成されている。

既存プロセスのドキュメント化
↓
重要なプロセス知識の識別と獲得
↓
既存プロセスの分析
↓
新しいプロセスの設計
↓
選択案内での評価と選択
↓
実行すべき経営事例の作成
↓
適用する変更に関する認可の取得
↓
プロセス変更の計画と実行
↓
継続的なプロセス改善を通じた、競争促進環境の維持

図 1-3　IDEF3 によって促進可能な経営改善フロー

１．４　利益（Benefit）

IDEF3 の適用によってもたらされる利益は、コスト削減や日程の正常化（schedule gain）、品質向上、組織的能力の向上、組織文化の変化の持続、といった成果として現れて来る。IDEF3 は以下のような目的に使用されている：

（１）組織間の不明瞭な繋がりの識別
（２）冗長的な活動や無価値な活動のあぶり出し（highlight）
（３）新しいプロセスの手早い作成

また、IDEF3 のメカニズムとしての使いやすさによって得られつつある追加の利益には、以下のようなものがある。

（１）地理的に分散した部署間での、生産プロセス知識（ハッブル望遠鏡の加工プロセス、等）の獲得と分配

（２）企業の主要な運営シナリオにおける、組織の情報リソースの重要性の判断

（３）人間システムの相互関係に対する、実装に依存しない仕様（implementation-independent specification）の提供

（４）データ構成（data configuration）の、運用管理・変更管制ポリシーの定義

（５）重要な共有データの状態とライフサイクルに影響を与える、決断手続のドキュメント化

（６）高品質な IDEF0 機能モデルの開発速度の向上

（７）シミュレーションモデルの開発と検証の速度の向上

（８）事実（fact）、決定点（decition point）、作業分類（job classification）の明確な定義を行うメカニズムを提供する、リアルタイム管制ソフトの開発

（９）ワークフロー運用管理システム・アプリケーションの挙動の定義

（１０）組織内で達成される変化を用いた、プロセスの規定

１．５　論文の構成（Document Organization）

この論文は、以下の 8 つの章で構成されている：

第 1 章　はじめに
第 2 章　IDEF3 の概要
第 3 章　IDEF3 プロセス記述言語
第 4 章　IDEF3 記述の開発
第 5 章　IDEF3 の開発：資材発注プロセスの例
第 6 章　IDEF3 プロセス記述の理解
付録 A　IDEF3 の精巧言語（elaboration language）
付録 B　用語集

第 2 章では、IDEF3 プロセス記述の開発に使用されている基本的な構築ブロックの記述を含む、IDEF3 手法の簡単な概要を述べている。第 3 章では、IDEF3 言語とその意味論（semantics）について、経験豊富なユーザーの先進的概念と共に、詳細に説明して行く。第 4 章と第 6 章では、IDEF3 手法をシステム的に適用する為の、実践ガイドラインを提供する。そして第 5 章では詳細な例を紹介する。

付録 A では、IDEF3 の精巧言語（elaboration language、対応する日本語は無い。ELL と呼ばれている）を説明し、また付録 B では、IDEF3 手法の主要な専門用語を定義する。精巧言語は、IDEF3 アプリケーションを通じて獲得された情報を再利用する為に用いる、コンピューター処理可能な媒体（medium）である。

著者は、この論文を幅広い目的に使用できるように記述している。その為、読者が論文全体を読まなくとも、必要な情報を得ることが可能なような形式を採っている。その為、目的に応じて以下のようなガイドラインが提案されている：

（1）経営者の為の概要（executive overview）としては、第 1 章と第 2 章を読めば良い

（2）IDEF3 プロセスフロー記述を正確に開発できる専門家を目指すのであれば、資料全体を読むべきである。特に第 2～4 章、第 6 章を重点的に読むべきである。

（３）IDEF3 の経験がある分析者は、第 3 章を言語の参考文献として使用する事が可能である

（４）IDEF3 プロセス記述の批評（reviewing）の専門家を目指すのであれば、第 6 章を精読し、第 2 章と第 3 章を流し読みすれば良い

（５）IDEF3 のプロジェクトリーダーは、第 3 章と第 4 章を詳細に研究しておけば良いが、なるべくなら、手法の全体についても理解しておくべきである。

１．６　概要（summary）

IDEF3 は、既存システムもしくは構築予定のシステムの、最重要プロセスの獲得と分析を補助する為に作成されたものである。供えられたガイドラインと、単純で使いやすい図形言語構造により、幾つもの下流工程で使用可能な、プロセス情報の獲得と組織化とを実現している。複数の視点から 1 つのシステムの挙動状態に関する記述を獲得、構成する能力を始めとする IDEF3 の独自機能は、システム挙動を近似するエンジニアリングモデルの構築においてユーザーを指導してくれるだけでなく、システムの挙動についての知識に精通した専門家により伝達された情報を、ユーザーが獲得できるようにしてくれる。この特性は、IDEF3 と、その他の選択手法との大きな相違点の一つである。そして他の IDEF ファミリー手法と同じく、IDEF3 は単独でも、他の IDEF との併用でも、経営の重要プロセスの識別と開発に適用することが可能である。

第 2 章　IDEF3 の概要（Overview）

この章では、IDEF3 手法の幅広い概要と、幾つかの例について述べる。組織化構造について説明を行うには、基本的な IDEF3 要素[3]を知る必要があるのだが、完全な定義については第 3 章の「IDEF3 プロセス記述言語」で説明することとして、ここでは簡単に触れるのみとする。

本文注 3：　IDEF3 要素は、IDEF3 の基本言語を構成している物で、**UOB、接合点、リンク、対象物状態、参照先、**といったものがある。

２．１　シナリオ：IDEF3 プロセス記述の為の、組織化構造（organizing structure）

IDEF3 プロセス記述の基本的な組織化構造には、シナリオもしくはストーリーという観念が用いられている。シナリオは、繰り返される状況や、組織やシステムによって指摘された問題の典型的なクラスについて記述した状況のセット、もしくは、そのプロセスが発生する設定として考える事も可能である。シナリオは、記述の焦点（focus）と境界状態（boundary condition）を確立する。このようにシナリオを使用する事で、人間というものが、対象となるシナリオもしくは状況を背景としている活動の順序通りに、自分の知っている事を記述するという傾向を上手く利用している。シナリオを、プロセス中心知識（process-centered knowledge）の収集物を組織化する際の媒介物として利用すると、組織化が便利になる。

シナリオの主な役割が、IDEF3 プロセス記述の背景（context、文脈）のまとめであることから、シナリオ名を適切なものにすることが重要となる。シナリオ名は命令形（「発注書の発行」や「取付テスト（Test Fit)」といった動詞もしくは動詞句）となることが多く、また時には、動名詞（「実行一致確認（Performing Consistency Checks)」のような名詞として機能する動詞）の形をとることもある。シナリオ名を正しく選ぶことで、記述を行うユーザーは、記述された現実世界の状況を正しく理解することができるのである。シナリオを正確に識別、特徴付け、命名することは、プロセス中心の IDEF3 プロセス記述を作成する為に必要なステップである。以下に、典型的なシナリオ名の例を挙げる：

①側面開口パネルのプレス型設計（Develop Die Design for Side Aperture Panel）
②顧客からの苦情処理（Processing a Customer Complaint）
③エンジニアリング変更要求の適用（Implement Engineering Change Request）

IDEF3 プロセス記述は、「プロセス中心」戦略と「オブジェクト中心」戦略の、2 つの知識獲得戦略を用いて作られている。プロセス中心戦略は、プロセスと、プロセス間の時間関係（temporal relation）・因果関係（casual relation）・論理関係（logical relation）に焦点を置き、プロセス知識を体系化（組織化）してゆく。2 つ目の軸（オブジェクト中心戦略）は、オブジェクトと、オブジェクトの状態が 1 つ以上のシナリオを通して変化してゆく挙動に焦点を置き、プロセス知識の体系化を行う。

IDEF3 ユーザーは、この 2 つのプロセス知識獲得戦略の片方もしくは両方を用いて、IDEF3 プロセス記述を作成する。2 つの戦略は、どちらも IDEF3 言語の基本要素を使用して、記述を形成する主張（assertion）の獲得と表現とを行っている。プロセス記述に含まれる情報の図形的投影には、IDEF3 の図形言語を用いている。図形言語はプロセス情報の直接的な記録装置と、プロセス情報の表示機構の、2 つの目的で用いられている。このような図形的投影は、「スキーム図（schematic、系統図、概略図、回路図等）」と呼ばれている。

2 つのプロセス知識獲得戦略には、それぞれ別々の IDEF3 スキーム図が割り当てられている。IDEF3 の「プロセススキーム図（Process Schematic）」は、プロセス中心的視界（process centered view）からシナリオを表現している。一方の「オブジェクトスキーム図（Object Schematic）」は、オブジェクト中心情報（object centered information）の図形表示を支援している。オブジェクトスキーム図は、オブジェクト中心視界（object centerd view）から、あるシナリオを表現しているが、これは「変移スキーム図（Transition Schematic）」と呼ばれている。そして、追加のオブジェクトとオブジェクト関係を表示する事で、背景設定情報（context-setting information）を表現した変移スキーム図は、「拡張変移スキーム図（Enhanced Transition Schematic）」と呼ばれている。また、複数のシナリオに跨ったオブジェクト中心情報を表示したオブジェクトスキーム図は、単純に「オブジェクトスキーム図」と呼ばれている。

1 個の IDEF3 プロセス記述には、0 個もしくは 1 個以上のプロセススキーム図と、0 個もしくは 1 個以上のオブジェクトスキーム図とが含まれている。例えば、ある特定のオブジェクトが、ある分野で、そこへの参加者によって認識されたことを記録する作業は、その分野の記述の一部であると考えられている。十分に識別されたオブジェクトは、記述内

のそのオブジェクトと関係するオブジェクトスキーム図を持つ可能性があるが（？may or may not have)、こうしたオブジェクトも記述の一部であると考えられている。シナリオという概念は、プロセス中心視界と、オブジェクト中心視界の、両方を体系化（組織化）する為に使用されている。収集されたシナリオと、そのシナリオを体系化する際に使用された情報を構成したものが、IDEF3 プロセス記述となっている。

この 2 つの IDEF3 スキーム図の記述表現概念と記述表現文法について、これから簡単に紹介して行きたい。

２．２　プロセス中心視界：プロセススキーム図
（Process-Centered Views: The Process Schematics）

IDEF3 プロセススキーム図は、プロセス中心知識の獲得、運用管理、表示を行う為の、主要な手段である。このスキーム図は図形情報媒体として、対象分野の専門家と、別の適用範囲の分析者との間で行われる、プロセスに関する知識の情報交換を補助してくれる。プロセススキーム図には、イベント（event）や活動（activity）についての知識、イベントや活動の発生に関係するオブジェクト、それらの発生の挙動を統括する制約関係も含まれている。

プロセス中心記述は、IDEF3 スキーム図言語の基本組立ブロック（basic building block）を使用し、それらを異なった方法でリンクさせることで、システム的に構築される。この組立ブロックは、それに関係する特定の意味論（sematics）を持っている。つまり組立ブロックは、現実世界での特定の種類の活動もしくは関係を表現する為に使用されている。この組立ブロックの詳細仕様は、第 3 章で説明する。この第 2 章では幾つかの重要な組立ブロックについて紹介し、また同時に IDEF3 プロセススキーム図の作成において、この組立ブロックをどのように使用しているかを例を挙げて説明してゆく。

図 2-1 は、資材発注（Order Material）という名前のシナリオについての、プロセススキーム図の例である。IDEF3 では、シナリオは記述の背景（context）を固めている（bound）が、これは異なった面から似たような状況を記述する際に、便利な機構になっている。こ

こで挙げる例では、経営者は、新人労働者の教育補助と、社内での発注標準の強化という 2 つの目的で資材発注プロセスをドキュメント化を行う際に、IDEF3 を使用している。経営者は、新人の為に、どのようにして購入要求書（Purchase Request）が処理されているのかを、特に記録したいと考えていた。プロセスの記述を要請された際に、経営者は以下のように説明している（訳注：読みにくく、箇条書きにでもすべきなのだが、実際の聞き取り作業の雰囲気を残す為か、わざとこのような形に書かれているようである）。

まず最初に我々が行わなければならないのが、購入要求書の書式を使用して、資材を要求する事である。それが行われると、購買部門（Purchasing department）は、発注した資材について現在取引を行っている供給者か、候補となると思われる供給者の一覧を識別する。もしも必要な物品について取引中の供給者が無い場合には、購買部門は候補となる供給者に対して入札を要求し、その中から最も良い価格を出した供給者を候補に決めなければならない。供給者が決まれば、購買部門は要求された資材を発注してゆくが、資材の要求には、まず購入要求書（Purchase Request）を準備しなければならない。それから資材要求者は、会計責任者（Account Manager）もしくは代理人から購入の許可を取らなければならない。また、会計責任者に提出する購入要求書には、その購入予算を賄うプロジェクトの会計番号を割当てなければならない。この時、会計責任者もしくは代理人は、担当するプロジェクトの会計に関する全ての購入を担当（responsible）し、認可しなければならない。会計責任者が購入を認可したら、承認者の署名（authorization signature）が必要になる。利害関係が衝突する事を防ぐ為に、資材要求者と会計責任者・購入承認者は別人にしなければならない。直接費プロジェクト（Direct project）の購入要求書には会計責任者の署名が必要だが、間接費プロジェクト（Indirect project）の場合には署名は不要である。必要となる全ての認可署名が揃ったら、資材要求者は署名された購入要求書を購買部門へと提出する。それを受け、購買部門は要求された資材を発注する。資材発注後、購入要求書は発行済の発注書（Purchase Order）として追跡される。

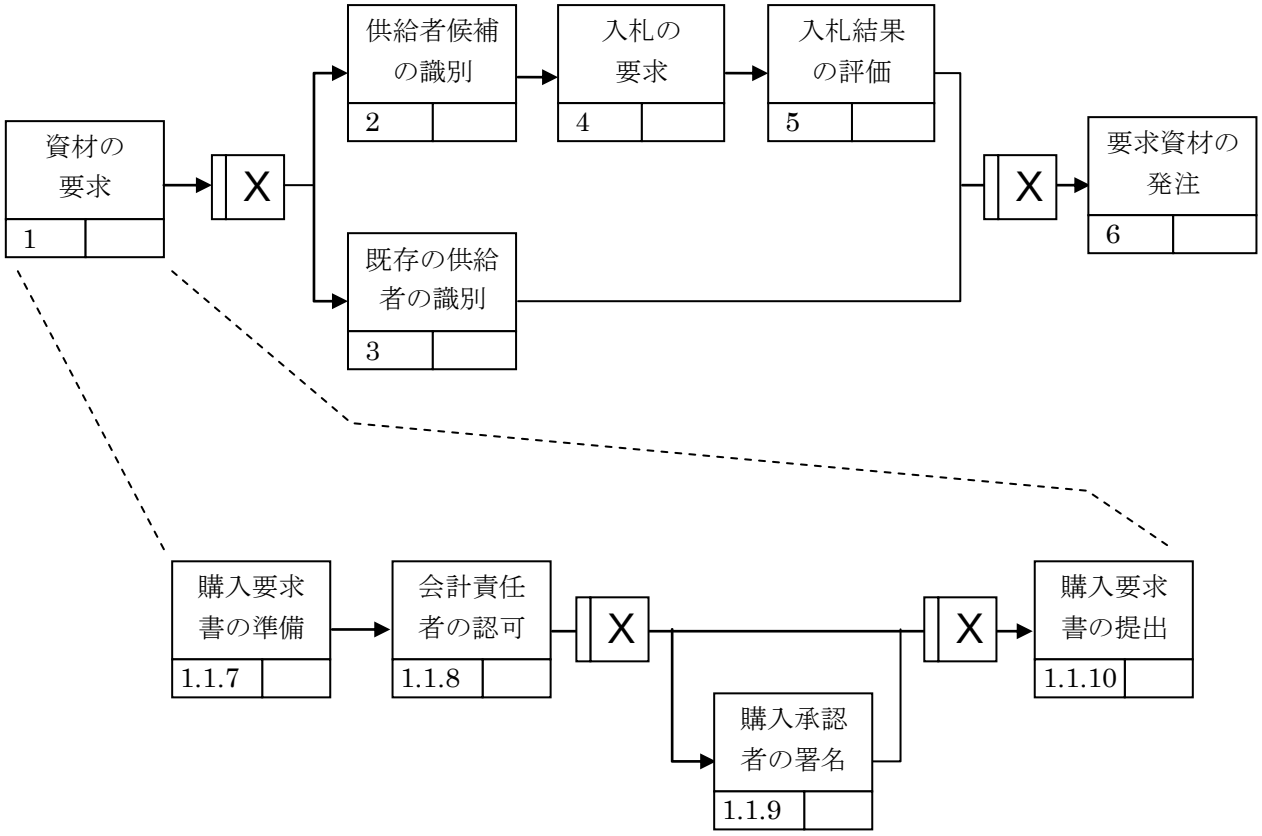


図 2-1　プロセススキーム図の例

経営者の記述内のプロセスは、スキーマ図上の、1 から 10 までの番号付された箱として表現されている。それぞれの箱は、イベント（event）や決定、行動（act）、プロセスに関する情報といったものを区別可能なパケット（packet、小包、小箱）として表わされている。つまりは、個々の箱は出来事（happening）の種類を表現している。このような出来事は、中立用語（neutral term、訳注：分類学上で明確な分類階層に当てはまらないグループやフォームがそれに当るらしいのだが、良くわからない）である「行動ユニット（UOB, unit of behavior）」と呼ばれている。個々の UOB 箱は、それぞれが現実世界のプロセスを表現している。1 つの UOB について記録された情報には、次の物が含まれている。

①UOB が何を表現しているかを表わした名前（動詞ベースの物が多い）
②プロセスに関わるオブジェクトの名前と、そのオブジェクトの特性
③オブジェクト間に張られた関係

図 2-1 上の、箱の間に接続された矢印（リンクと呼ばれている）は、記述されているプロセス間に結ばれている前後関係（precedence relation）（もしくは、より一般的に「制約」）を示している。つまり、リンクの起点にある UOB のインスタンス（instance、具体的なオブジェクト）は、同じリンクの終点にある UOB のインスタンスの前に終了することになる。例えば、「入札の要求」というラベルの UOB は、「入札結果の評価」というラベルの UOB が開始される前に、終了することになる。「X」の入った小さな箱は、「交差点（junction）」を示している。交差点では、1 つのプロセスが複数のパスに分岐するか、もしくは複数のパスが合流している。交差点は、プロセスの活動ロジック（activation logic）の制約（もしくは制約の効果）を示している。例えば図 2-1 の上左にある最初の交差点は、記述された 2 つのプロセスの中のどちらかのパスのみが活動することを示している。

IDEF3 手法は、ユーザーによる記述の獲得を、「分解（decomposition）」と呼ばれるメカニズムを提供する事により、様々な抽象化レベルで実行可能にしている。この分解は、UOB のより詳細な記述を組織化する手段を提供している。分解されたスキーム図は、シナリオと同一の文法ルールに従っており、同一の IDEF3 要素を使用している。1 つの UOB は、異なった複数の分解物セットとなる可能性を持っているが、それらは全て同一の階層上に存在している。同一の UOB に対して、2 つ以上の（別々の）分解を行う事で、異なった視点の表現や、UOB に関係するプロセスの更なる詳細化を行う事が可能である。

図 2-1 上の「資材要求」という UOB は、UOB7～10 へと分解されている。UOB 箱の左下角にある 7 から 10 までの番号の前には、UOB1 への参照（1 桁目）と、分解回数（UOB1 を 1 回分解）とが付け加えられている。これは、記述内の各詳細レベル間での追跡性を明確なものにする IDEF3 の付番手法（numbering scheme）を表わしている。図 2-1 に描かれたプロセス記述は、経営者という特定の視点から見た、資材発注プロセスを示している。この同じプロセスについて、異なった視点、例えば会計管理者の視点に立ったプロセススキーム図を作成する事も可能である。異なった視点から記述された各視界は、それぞれが独自のラベルと番号を持つ、異なった分解を表現することになるだろう。

図 2-1 のプロセススキーム図は、資材発注プロセスを、プロセス中心視界（view）で表現したものである。この視界は、発生したプロセスとその順序に関する主張（assertion）に焦点を置いている。場合によってはプロセス中心視界からではなく、オブジェクト中心視界から状況の記述を組織化した方が都合の良いこともあるだろう（関係するオブジェクトやオブジェクトセットに注意を向けたい場合など）。そこで次は、オブジェクト中心視界を使用して、IDEF3 がどのようにプロセス記述獲得を促進しているかについて説明して行くことにする。

２．３　オブジェクト中心視界：オブジェクトスキーム図
（Object-Centered View: The Object Schematics）

IDEF3 オブジェクトスキーム図は、

・様々な種類のオブジェクトが、プロセスによって、どのように他の種類のオブジェクトへと変換されるのか、

・対象となる種類のオブジェクトが、プロセスによって、どのように状態を変化させるのか、

・プロセスにおける、オブジェクト間の重要な関係に関する背景設定情報（context-setting information）

といった、プロセスのオブジェクト中心記述の、獲得、運用管理、表示を行う。

IDEF3 でのオブジェクトは、分野への参加者により、その分野内で何が起こっているのかという記述の一部として認識、参照されている、何らかの具体的もしくは概念的物体である。オブジェクト中心 IDEF3 プロセス記述の作成において、オブジェクトの識別、特徴づけ、命名は、正しく行われなければならない。オブジェクト名は、名詞もしくは名詞句であることが多く、それに状態記述子（state descriptor、下例の「：」の右側）が付くこともある。下に挙げるのは、典型的なオブジェクト名の例である：

（１）水：沸騰
（２）調達発注書：認可済
（３）車体

単体のシナリオを背景として作成されたオブジェクトスキーム図は、そのシナリオ内のイベント上にあるオブジェクトで発生する状態変移を、特徴付けている。このようなスキーム図は変移スキーム図（Transition Schematics）と呼ばれており、これを用いる事でユーザーは、対象となるシナリオで発生しているオブジェクト状態間の変移を統括するルールを明確にすることができる。

またこれとは別に、より日和見的なやり方（opportunistic fashion）で進化し、複数のシナリオを跨いだ、オブジェクトやオブジェクト状態、それらの変移についての記述を獲得する、オブジェクトスキーム図もある。このやり方で作成されたオブジェクトスキーム図は、１つのシナリオ内のイベント上の、オブジェクト状態挙動変化の構造を定義するという意図を持っていない。こうした複数シナリオに跨ったオブジェクトスキーム図の開発ア

プローチは、より詳細な焦点を当てるべきオブジェクト中心プロセス情報を探索する時や、ある記述に出現するオブジェクトに関する背景設定情報（context-setting information）を発見しようとしている時に、便利であることが多い。（こうした複数のシナリオに跨る）オブジェクトスキーム図は、背景設定シナリオ名（context-setting scenario name）の存在の有無から、より特殊化された変移スキーム図（並びに強化変移スキーム図）と区別できる。

一般的には、IDEF3 のオブジェクトスキーム図は、１つの特定プロセスもしくは特定シナリオの、オブジェクト中心記述を提供する為に作成されている。その為、IDEF3 オブジェクトスキーム図の作成においては、変移スキーム図が注目されることが多い。

図 2-2 のスキーム図は、経営者の記述から作成された、「資材発注（Order Material）」シナリオのオブジェクトスキーム図である。この例では、変移スキーム図として描かれているが、これは資材発注シナリオで発生するオブジェクト状態変移の本質と構造とを特徴づけしているからである。

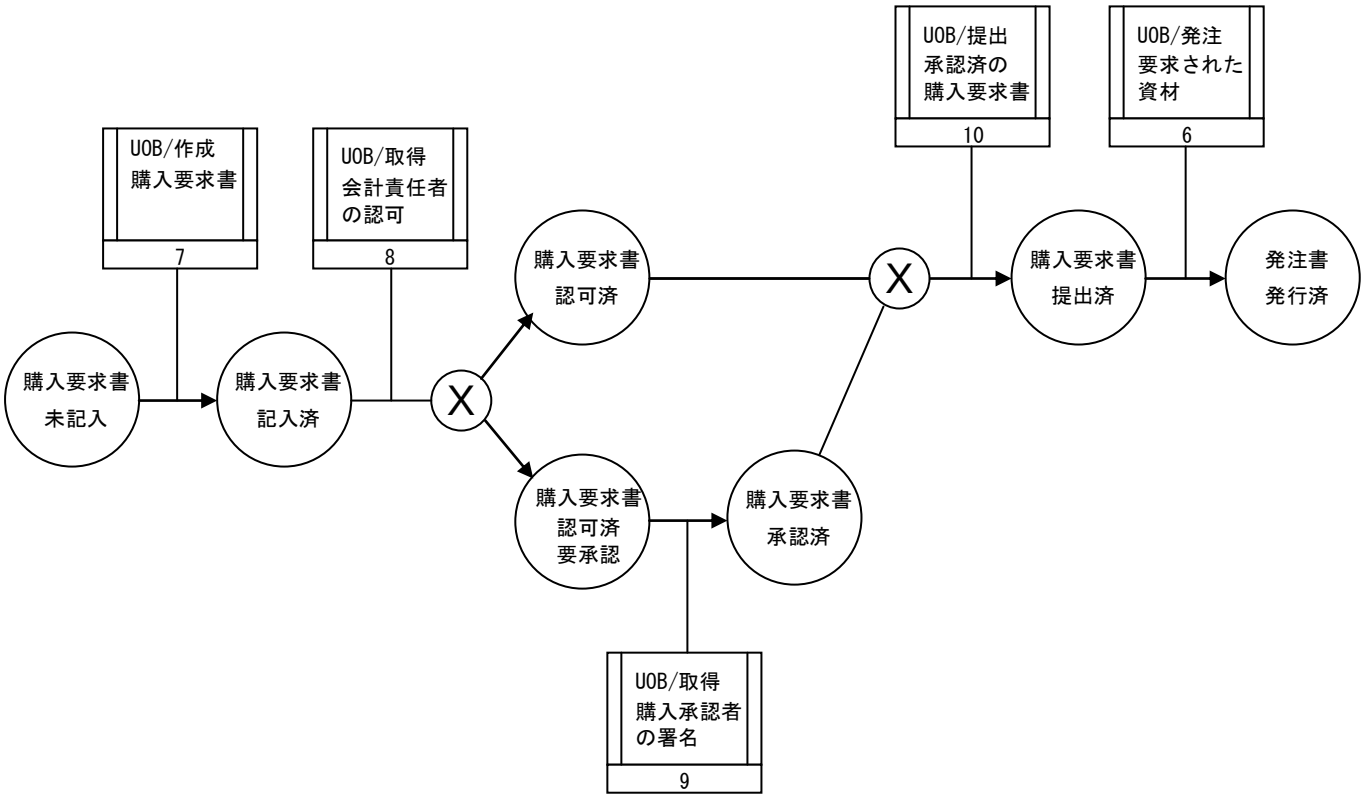


図 2-2　変移スキーム図の例

このプロセスで重要な部分は、「購入要求書」の書式である。この書式は、「資材発注」プロセスを通して、発注書へと次第に変化している。

オブジェクトの名前を含んだ円は、何らかの種類のオブジェクトを表わしている（購入要求書、会計責任者、プロジェクト、等)。こうしたラベルの付いた円は、「種類（kind)」シンボルとして知られている。何らかの状態にあるオブジェクトの何らかの種類は、種類そのものと、それに一致した状態の両方を持つラベルの付いた円で表現されている。これにより、(対象となるプロセスの）ある状態にあるオブジェクトの種類（もしくはクラス）を表現している。例えば、認可を受けた購入要求書は、「購入要求書：認可済」、承認された購入要求書は「購入要求書：承認済」、といったように表される。オブジェクトスキーム図の作成の第一歩は、オブジェクトが変化し得る状態を識別する事である。現実世界に存在するオブジェクトは、状態が連続しつつ変化する物も多いが、オブジェクトスキーム図上では、分野専門家にとって興味が切り替わる状態へと分割し、それぞれに焦点を当てている。円の間を結ぶ変遷弧（transition arc）（黒塗りの三角形の付いた矢印）は、ある状態から次の状態へと変化した活動である「状態変遷（state transition)」をシンボル化したものである。そして、あるオブジェクトが、対象となる状態（state）である場合に確立される：

・どのようにして、その状態（state）であり続けているのか、
・どのようにして、状態間の変移が発生するのか、
・どのようにして、新しい状態（state）へと変化するのか、

という状態（condition）が、特殊な形式で記録されている。矢印に（参照（referent）と呼ばれているもので）結び付けられた箱は、あるシナリオの発生に関係する、オブジェクト状態や UOB、シナリオ、他の変移スキーム図の間の関係性の記述を補助している。例えば、オブジェクト「購入要求書」が、会計責任者によって批評される為に準備された状態（購入要求書：記入済）から、認可された状態（購入要求書：認可済、もしくは購入要求書：認可済・要承認）へと変移する間に、UOB「取得　会計責任者の認可」によって表されるプロセスが開始され、そして終了されなければならない。「X」(排他的論理和、「どちらか」の意味) が書かれた変移交差点 (transition junction) は、幾つかの発生し得るパスの中で、厳密に 1 つのパスが選択される、ということを示している。

このように、図 2-2 は「未記入」状態から「記入済」状態、「記入済」状態から「認可済」状態もしくは「認可済　要承認」状態への、購入要求書の変移を表わしている。購入要求書に署名が必要な場合には、「提出済」状態へと変移させる前に、「承認済」状態へと変移させられる。そうでなければ、そのまま「提出済」状態へと変移する。購入要求書が「提

出済」状態へと到達すると、オブジェクトは「発行済」状態の発注書へと変移する。状態間の変移に関係する UOB やシナリオ、他の変移スキーム図は、適切にラベル付された、オブジェクトスキーム図（訳注：プロセス中心のスキーム図）への参照（referent）を付与することにより、指示されている。そして、変移スキーム図上の参照の相対的位置関係は、それらの発生する順序を表わしている。例えば、図 2-2 における UOB「購入要求書　作成」の位置は、シナリオの発生時、スキーム図によって参照されている他の全ての UOB の前に、開始・終了している、ということを表わしている。

興味深いのは、発生し得る状態変移の中に、要求の失敗が反映されていないということである。これは単に、元々の聞き取り（dialog）に、そうした（失敗に関する）状況に関する情報が含まれていないからである。このことは、IDEF3 の使用において重要な点となる。IDEF3 は、その分野専門家によって行われた主張を構造化する為のメカニズムであることを意図している。仮説の「モデリング」に含まれる部分的情報を、完全なものにするように強制されることはない。

図 2-2 のスキーム図に追加の背景設定情報（context-setting information）を追加する事で、図 2-3 のように装飾することができる。この図では、変移スキーム図は追加情報を提供するオブジェクトと適切な関係リンクで補強されている。例えば、「購入要求書：作成」と「直接費プロジェクト」、「購入承認者の署名」の間にある 3 か所の関係は、直接費プロジェクトに関する購入要求書には購入承認者の署名が必要であることが表現されている。更に、承認された購入要求書それぞれに購入承認者の署名が含まれていること、そして要求者は購入要求書の認可もしくは購入要求書への署名を行う人物であってはならないということも表現されている。

ここで紹介した例について、第 5 章で更に詳細に説明している。特に、プロセススキーム図とオブジェクトスキーム図の双方の作成に使用されている各プロセスを、一つ一つ説明している。次の第 3 章では、IDEF3 の基本的なスキーム図要素とその意味論（semantic）についての詳細な説明を続けることにする。

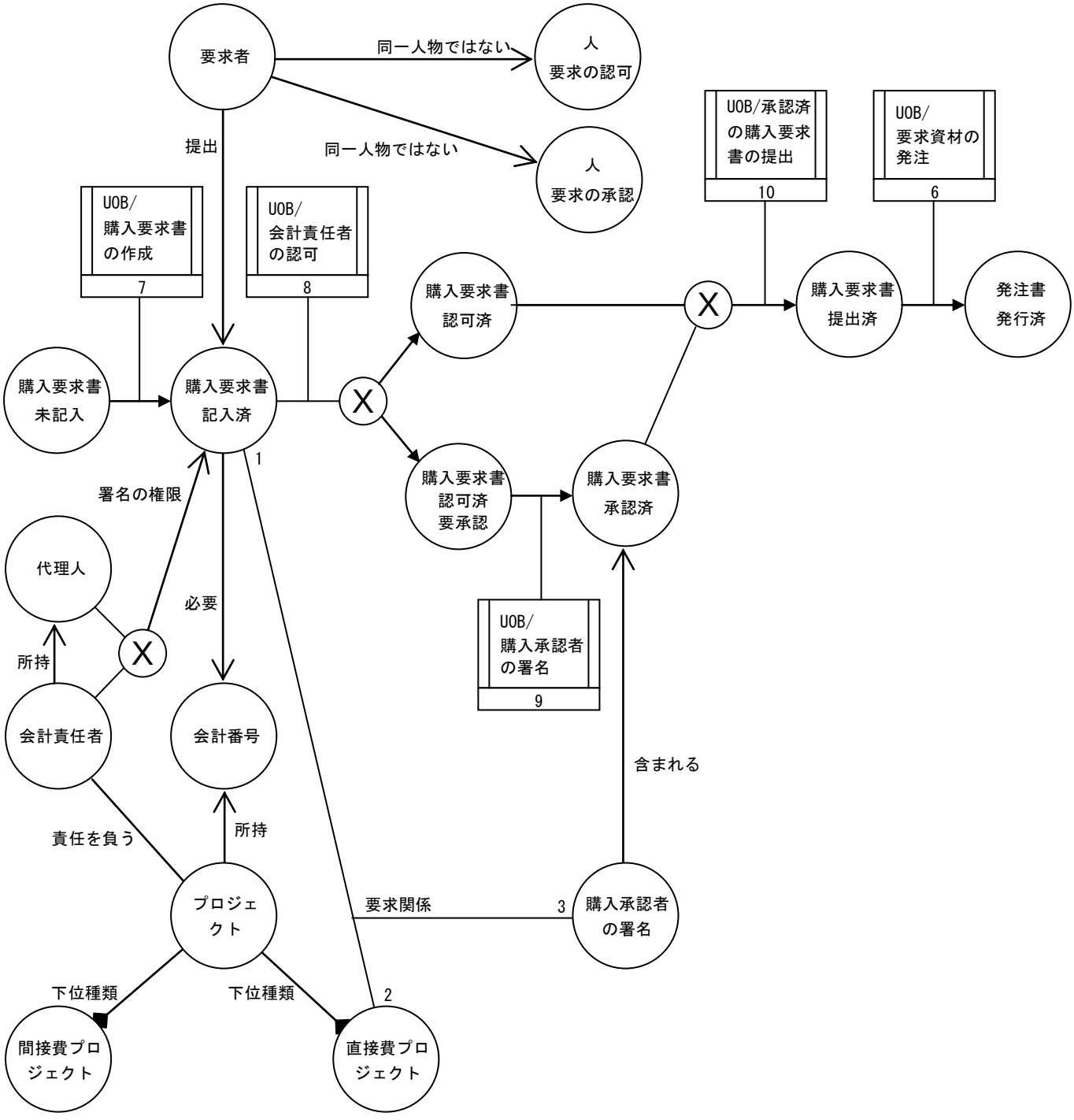


図 2-3　強化変移スキーム図の例

第 3 章　IDEF3 プロセス記述言語

この章では、IDEF3 プロセス記述言語の基本的な要素と、それらの要素をどのようにしてまとめ上げ、意味論に富んだ動的システム記述を形成してゆくのか、ということについて説明して行く。

１つの IDEF3 プロセス記述は、ある特定のシナリオ内の、状況間の関係ネットワークを組織化している。この IDEF3 記述は、プロセス中心とオブジェクト中心という、2 つの異なった切り口から作成される。この 2 つのアプローチは完全には排他的でない為、この異なったアプローチから相互参照することにより、IDEF3 は複雑なプロセス記述を表現する事が可能となるのである。

3.1 章から 3.5 章までは、IDEF3 プロセス記述言語の文法要素について説明している。3.6 章では、その文法要素を組み合わせることによりプロセス記述を形成する方法について説明する。そして 3.7 章では、プロセスのオブジェクト中心記述に用いる、IDEF3 オブジェクト状態変遷記述（object state transition description）の文法要素について説明する。こうした言語の其々で行われている宣言（statement）を用いた相互参照のメカニズムについては、個々の言語仕様の説明の中で紹介している。そして基本的な文法要素をどのようにして組み合わせて IDEF3 スキーム図を構築して行くかについての例は、それぞれの節（3.1 章から 3.7 章）に分散して紹介している。

３．１　IDEF3 プロセス記述の基本要素

IDEF3 プロセス記述言語の基本的な文法要素を、図 3-1a に示す。図 3-1b は、一階関係（first-order relation）に用いる、選択シンボル（alternative symbol）の取り決めについて説明している。

こうしたシンボルの非公式な文法や意味論と、それを用いて構築可能な、より複雑な構造については、次の節で説明して行く。

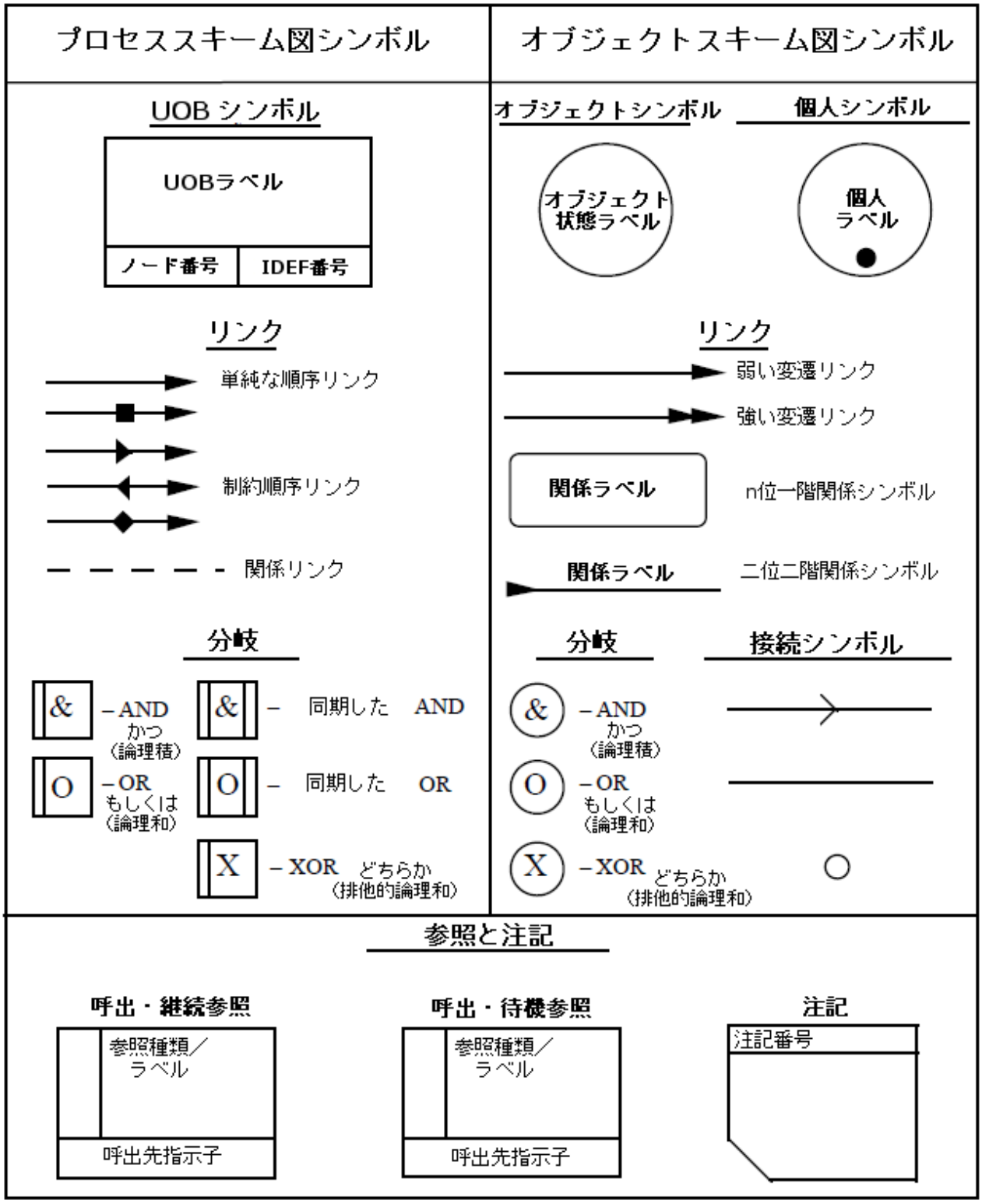


図 3-1a　IDEF3 プロセス記述スキーム図で使用されているシンボル一覧

選択一階リンクの取り決め

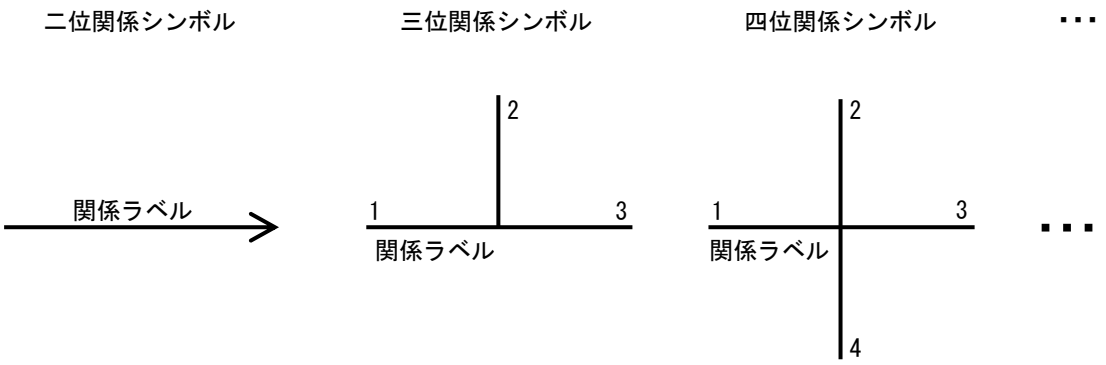
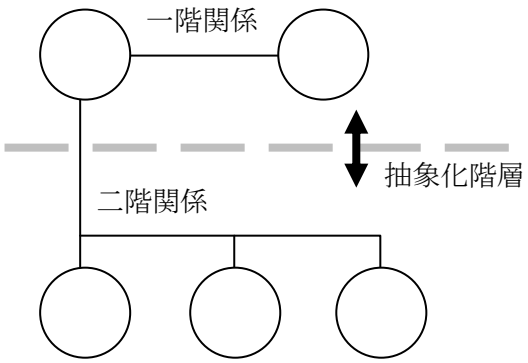


図 3-1b　一階リンクの選択シンボルの取り決め

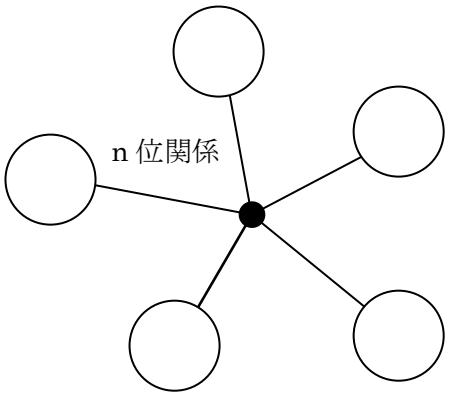
訳注：　「階（order)」と「位（place)」

本文中に order という単語と place という単語が出てくる。それぞれ「階」と「位」と訳され、論理式等で用いられている。wiki 等で論理式について調べてみたものの、とても理解できそうにも無く、また本文中での用いられ方を見ても、論理式の世界で用いられているような厳密で複雑なものではない。

そこで、恐らく本文中で用いられている意味はこうではないのか、という推測に基づいて、階（order）と位（place）について簡単にまとめてみる。



同一の抽象化階層内の関係：一階関係？
抽象化階層を 1 つ跨ぐ時：二階関係？
跨ぐ階層の数が N 個：N 階関係？



n 位＝関係の分岐数？

３．２　プロセススキーム図

プロセススキーム図は、IDEF3 手法の中で、最も親しまれ、広く使用されているものである。このプロセススキーム図は、シナリオのプロセス中心記述で用いる視覚的メカニズムを提供してくれる。プロセススキーム図を構成する図形要素には、行動ユニット（UOB）箱、順序リンク（precedence link）、交差点（junction）、参照（referent）、そして注記（note）がある。参照と注記は、プロセススキーム図とオブジェクトスキーム図で共通の構造をしている。プロセススキーム図の作成で用いられている、こうした図形的要素それぞれの用法について、より複雑な宣言（statement）がどのように定式化されているのかという説明を添えつつ紹介して行きたい。

３．２．１　行動ユニット（Unit of Behavior、UOB）

UOB（＝UOB 箱）の意味について明確にする為に、まずは「種類（type）」と「インスタンス（instance）」とを区別しておかなければならない。この区別のやり方は、データベース設計の分野で行われている事が多い。

データベーススキーム（scheme、方向性、基本構造）の設計を行う為に、データベース設計者は、対象となるシステム上に見つかった個々ののオブジェクトを抽象化することで、「クラス（class）」もしくは「種類（type）」を分離してゆく。この時、見つけられた個々のオブジェクトは、分離されたクラスもしくは種類の「インスタンス（instance）」となる。同様に、データベース設計者は、個々のオブジェクトが持つ、個々の属性を抽象化して行き、同一の種類に属する全てのインスタンスに共通な属性の種類（色、型式、硬さ、等）を識別する。

次にこの情報を用いて、設計者が情報を維持したいと望むオブジェクトの、ある特定の種類についての関係スキーマの設計を行う。この関係スキーマは、実体関係性（Rntity Relationship、ER）もしくは IDEF1X モデリング言語の中で、データベーススキーマによって表現されている情報の一種であり、対象となるシステム上のオブジェクトの「種類」や、陳列されている全ての対象種類の持つ属性オブジェクトの「種類」、それらを１つに束ねるロジック的拘束、といったものを記述している。

同様な理由で、対象となるシステムで「何が進行しているのか」を獲得する時には、特定の時間にシステムで実際には発生している事象を記述するのではなく、むしろ、システムで「一般的に（in general）」発生している事象を記述する。つまり、システム内で何度も発生し得る、一般的な動的「パターン（pattern）」や、状況（situation）の一般的な「種類」を記述するのである。その為、「行動の計画」や「決断（Make Decision）」、「購入の決定（Buy Decision）」、「契約締結イベント（Contract Award Event）」といった UOB は、状況の「種類」を記述しており、UOB インスタンスは単純にその UOB の発生（occurrence）を表わしている。データベーススキーマのように、プロセス記述は種類という面からシステムを記述しているのである。言い換えるならば、プロセス記述はシステム上で発生し得る状況（プロセスや機能等）の種類と、それらをまとめるロジック的・時間的制約を、表しているのである。

図 3-1a で図示されているように、UOB は固有の「ラベル」が付けられた、特種な箱によって表現されている。UOB 箱（a UOB box）と、UOB 箱によって表現された UOB（the UOB）の違いを、名前そのものと、名前によって示された個人とを区別する事が重要であるのと同様に心に留めて置かなければならないにも関わらず、実際には「UOB」という用語は曖昧に用いられることが多く、スキーム図に描かれている UOB 箱を指すこともあれば、シナリオで表現されている UOB を指すこともある。しかし、通常は文脈（context）が十分に存在することから、どちらを意味しているのかを区別可能であり、更に多くの場合、スキーム図と、そのスキーム図によって表現されたシナリオとの間に構造的類似性が存在する為、この曖昧さは問題にはならない。

（訳注：この段落も何が言いたいのかよくわからない。例えばプロセス記述の中で「行動の計画」という UOB について述べられている時に、スキーム図上の「行動の計画」というラベルの付いた UOB 箱の事を指しているのか、それともスキーム図で表現しようとしている「行動の計画」という UOB を指しているのか曖昧である、ということだけなら理解可能なのだが、「スキーム図とそれによって表現されたシナリオに構造的類似性があるので曖昧であっても問題ない」という事が何を言っているのか理解できない。誤った方として獲得してしまった場合でも、結果同じ事を言っているので問題は無い、とでもいうのだろうか？）

３．２．２　リンク（Link）

「リンク（link)」は、動的プロセスの表現を形成する為に、UOB 箱を接続する接着剤である。リンクは主に、プロセス上の UOB の間の重要な関係性を強調する為に使用されている。

IDEF3 リンクによって強調することが可能な関係の種類の例として、時間的関係、ロジック的関係、因果的関係、本質的関係、因習的（通常？）関係等があるが、主に用いられているのは時間的順序であることから、この関係の表現専用の特殊なクラスを用意している。「順序リンク詳細ドキュメント（precedence link elaboration document)」により、ユーザーは特定の順序リンクに関する追加詳細を獲得する事が可能となる。そしてリンクは、UOB 箱もしくは交差点シンボル上のいかなる点からも、開始もしくは終了することができる。IDEF3 プロセススキーム図で使用されているリンクは、基本的に実線の順序リンク（precedence link）と点線の関係リンク（relational link）の 2 種類である。それぞれの種類の表現するシンボルは、図 3-2 の通りである。

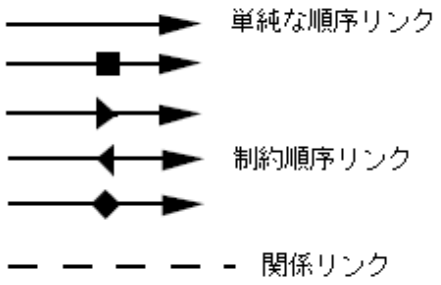


図 3-2　IDEF3 リンクの種類

３．２．２．１　単純順序リンク（Simple Precedence Links）

「順序リンク（precedence link)」は、2 つの UOB のインスタンス間の時間的順序関係を表現している。順序リンクは最も使用されるリンクであり、ソリッド矢印によって表示され、多くの場合には図 3-2 のように柄の部分に追加のマーカーが加えられている。図 3-3 で示されている UOB 箱間を接続する順序リンクは、単純順序リンクである。

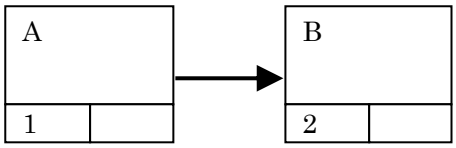


図 3-3　基本的な順序リンク文法

リンクの後端の「A」というラベルが付けられた箱 1 はリンクの「起点（source)」、リンクの前端の「B」というラベルが付けられた箱 2 はリンクの「終点（destination)」である。IDEF3 スキーム図として見る場合、箱 1 は箱 2 の（直接（immedeate)）先駆（predecessor）であり、箱 2 は箱 1 の（直接）後駆（successor）ということができる。（訳注：predecessor / successor を先駆／後駆として良いのかは自信が無いが、他に良い日本語も見当たらない）

IDEF3 スキーム図一般におけるスキーム図の意味とは、スキーム図で発生する可能性のある「活動（activation、起動、活性化)」という用語で理解する事が可能である。スキーム図の活動は、スキーム図によって表現されているプロセス内にある、その時間的特性とロジック的特性がスキーム図で明確にされた状態（condition）を満たした UOB の、一部もしくは全てのインスタンスを集めたものである。一般に、対象となるスキーム図 1 つに対して、幾つもの異なったパターンの活動が存在している。例えば、図 3-3 にあるような単純な 2 つの箱で構成されたスキーム図での、発生し得る活動パターンの 1 つとして、**UOB「A」の単体インスタンス「a」の後に UOB「B」のインスタンス「b」が続く**、ということを挙げることができる。これはより正確に、**（UOB）A と B のインスタンス a と b のいかなる組み合わせにおいても a が終了する前に b が開始されることは無い**、と言うことが可能であり、これが図 3-3 の活動の正規的な表現（legitimate activation）となるのである。

３．２．２．２　活動のプロット（activation plots）

活動の表現の一つとして、活動プロットが用いられる。ある活動における UOB インスタンスは、その活動の最初のインスタンスの開始から、最後のインスタンスの終了まで持続する、1 本の有限線状のものとして発生することになる。ある UOB インスタンスの集合が、対象となる記述の活動かどうかを判断するには、インスタンスの発生の一般的なパターンを、発生する期間をそのままプロットしてみるのが便利である。

記述作成が目的の場合には、一般的なパターンを発見する為に、対象となる期間内に観察された集合物を基にして活動パターンをプロットするのが便利であることが多い。この際、UOB の名前をリストにまとめ、各 UOB のインスタンスをそのインスタンスの発生時間と期間に従ってプロットする事で、この作業を仮想的に行う事が可能である、例えば、図 3-3 のスキーム図の活動プロットは、図 3-4 のようになる。この時、各 UOB の名前の右側にある線分は、その UOB のインスタンスが発生していた期間を表現している。このプロットでの、2 つの線分の投影が水平方向に重なっていないという事実は、インスタンス B はインスタンス A が終わる前に開始しない、という図 3-3 のスキーム図によって要求されている条件を表現していることから、このプロットは図 3-3 の活動の表現そのものであると言える。

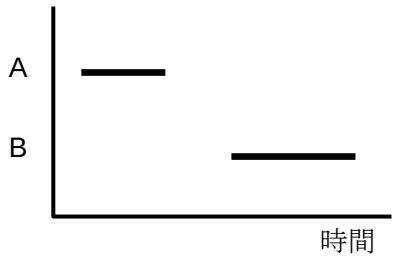


図 3-4　図 3-3 のスキーム図の活動プロット

３．２．２．３　制約順序リンク（Constrained Precedence Links）

図 3-3 のような単純順序リンクは、片方の UOB のインスタンスが、それの対となるインスタンスの記述の無いシステムで発生可能かどうかについて何も言及していない。図 3-3 によって示されている事は、A のインスタンスは B のインスタンス無しでも発生し得ること、もしくは B のインスタンスは A のインスタンスの前に発生し得ること、の 2 つだけである。このように、単純順序リンクの意味論はかなり大雑把（rather permissive）である。

制約順序リンクは、こうした単純順序リンクの活動意味論（activation semantic）の上に制約を加えたものである。制約順序リンクには幾つかの種類が存在するが、ここで最初に説明する制約順序リンクは、**起点 UOB の全てのインスタンスは、終点 UOB のインスタンスによって追従（be followed by）されなければならない**、ということを表現したものである。これは、リンクの「指向性（directionality、方向性）」であり、制約は、「左から右へ」だけとなるように強制される。

例えば、単純順序リンクとして見るならば、図 3-5a のスキーム図の活動は、「出勤簿への署名（Sign timesheet）」インスタンスは、「出勤簿の認可を取る（Obtain timesheet approval）」インスタンスによって**追従されている**、という構成になっている。しかし、このリンクに制約が加えられることによって、全ての「出勤簿への署名」インスタンスは、「出勤簿の認可を取る」インスタンスによって**追従されなければ**ならない、ということを表現するようになる（訳注：署名された出勤簿は、必ず認可を取らなければならない）。この場合、「出席簿の認可を取る」インスタンスが存在しないということは、システム記述との「不一致」となる。つまり、課せられた制約から外れるイベントの集合は、IDEF3 記述の活動には分類されないことになる。

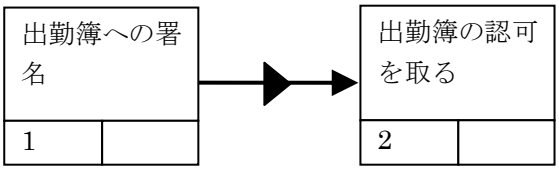


図 3-5a　制約順序リンクを含むスキーム図の例

図 3-5a のリンクのような指向性が与えられていても、「出勤簿の認可を取る」インスタンスのみが単独で存在する事は禁止されていない。例えば、対象となる給与支払期間の出勤簿が提出される前に従業員が辞めてしまった場合に、こうしたイベントが発生することになる（この場合、その従業員の署名の無い出勤簿が認可され、従業員が辞めた後に認可されたタイムシートに従業員が署名することはない）。

（訳注：制約順序リンクは、リンクで結ばれた 2 つの UOB がセットで存在している事までは保障はしていない、ということか？）

それに続く 2 つの制約順序リンクは図 3-5b に図示されているもので、1 つ目の制約順序リンクの逆向きになったマーカーと、両方向のマーカーが付いたものである。

上側のスキーム図は（もう一度繰り返すが、単純順序リンクの活動意味論に追加されたもの）、B のインスタンスは A のインスタンスによって**先導（be preceded by）されなければ**ならない、ということを示しており、下側のスキーム図は、A の全てのインスタンスは B のインスタンスによって**追従されなければ**ならず、かつ、B のインスタンスは A のインスタンスによって**先導されなければ**ならない、という事を示している。

こうした制約は、システム記述に「規範的（normative）」構成を追加してくれる。この規範的構成とは、例えば、システムの挙動がどのように観察されたかということだけでなく、どのような挙動を示すべきかということも表現する、ということである。その為、制約されたリンクは、IDEF3 によってシステムの挙動に関する信念（belief）と観察物を記録する場合だけでなく、システムのモデリングに IDEF3 を使用する場合にも、容易に使用することが可能である。

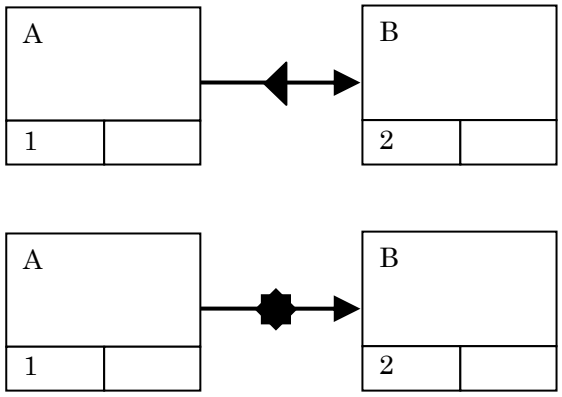


図 3-5b　制約順序リンクの例の続き

以上説明した 3 つのリンクだけでは、UOB 間に結ばれる可能性のある制約の全てを表わすことができないのは明白である。

例えば、図 3-3 の単純な順序意味論に、A の全てのインスタンスのどのような活動であっても、その終了時間から 5 分以内に、そのインスタンスに追従する B のインスタンスが開

始されなければならない、という制約を追加する場合を考えてみよう。最後に挙げる制約順序リンクは、このような一般的制約を表現するものであり、その為このリンクは「一般（general）」制約順序リンクと呼ばれており、図 3-6 のように図示する。こうした種類の制約の持つ本質は幅広く変化する事から、順序リンク詳細説明ドキュメント（precedence link elaboration document）を別途付加し、内容を明確に記録しておかなければならない。（順序リンク詳細説明ドキュメントについては、後で詳細に説明する）

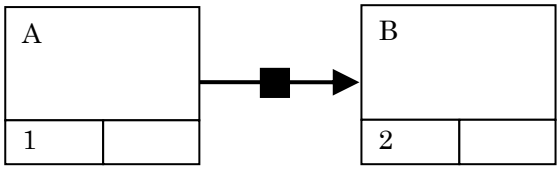


図 3-6　一般制約順序リンク

３．２．２．４　点線リンク（Dashed Link）

「点線リンク（dashed link）」の意味論は、事前に定義されていない。その為、ユーザー定義リンクもしくは関係リンクと呼ばれている。このリンクは、2 つの UOB 間の（制約的関係と成り得る）関係性の存在を強調する。

例えば図 3-7 の関係リンクは、「出勤簿への署名」と「出勤簿の認可を取る」の 2 つの UOB の間に、「自分自身の出勤簿に認可を出さない」という制約を表わしている（訳注：下図中にはリンクの詳細特性については書かれていないが）。関係リンクによって表される関係性の詳細な特性については、別途、関係リンク詳細説明ドキュメント（Relational Link Elaboration document）で明確にする。

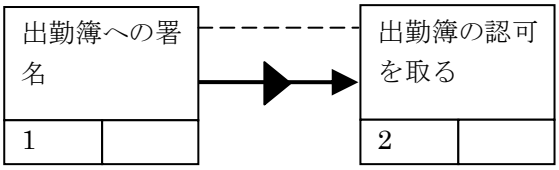


図 3-7　関係リンクの例

３．２．２．５　リンク番号

全てのリンクには、詳細説明の為の固有のリンク番号が割り振られている。順序リンク番号には頭に「PL」(Precedence Link) の文字が、関係リンク番号には「DL」(Dashed Link) の文字がそれぞれ付けられており、例えば順序リンクの場合だと PL1、PL2…となる。UOB の採番方法と同様な手続きを用いて、リンク番号の固有性を保証しなければならない（訳注：別のリンクに同じ番号を採番しないこと）。その為、リンク番号は著者（author）によって予約された番号から順番に割り振られることになる。プロセススキーム図へのリンク番号の表示はオプションであり、必ずしも必要ではない。

３．２．２．６　交差点が無いプロセススキーム図の活動意味論（Activation Semantics）

交差点について説明を行う前に（IDEF3 は、分岐したプロセスの構造も記述可能である）、これまでに挙げた例よりも大きな交差点の無いスキーム図で、異なった種類のリンクの意味論を一般化しておくと理解しやすい。そこで図 3-8 のような、委員会報告書に関する会議を開催するプロセスについて考えてみる。

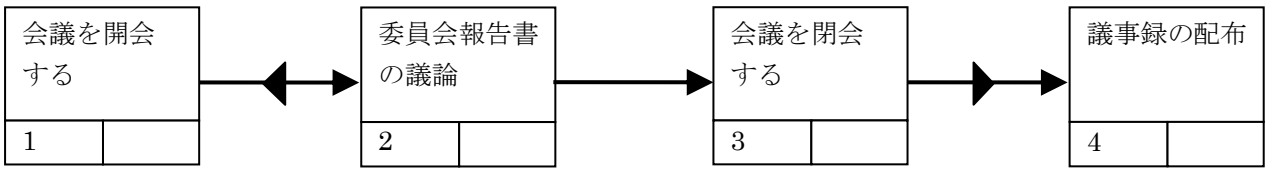


図 3-8　分岐しない IDEF3 スキーム図

一般的な IDEF3 スキーム図と同様に、このスキーム図の基本意味論は、そこに記述されている活動の発生し得るパターンという形で理解することが出来る。言換えるなら、対象となる背景（context）の中で、実際には何を会議と「見なしている」のかを、スキーム図は明確にしているのである。2 つの箱だけの簡単なものの場合、活動は一般的に以下のようなパターンとなるだろう：

・「会議を開会する」インスタンスは「委員会報告書の議論」インスタンスによって追従されており、更に「会議を閉会する」インスタンス、「議事録の配布」インスタンスによって追従されている。

・順番に並んだ各インスタンスは、それぞれの先駆インスタンスが終了した後にインスタンスが開始される

（訳注：ここでは順序リンクのみ言及されている）

図 3-8 のスキーム図は、全ての交差点の無いスキーム図（実際には、分裂分岐（disjunction branch）の無い全てのスキーム図）と同じく、その典型的な活動パターンは図 3-9 の活動プロットのようになる。

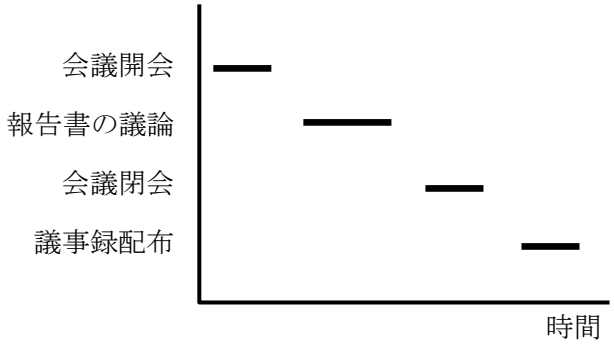


図 3-9　図 3-8 の活動プロット

制約順序リンクは、プロセス上の更なる制約について指示している：

・委員会報告書についての議論は、会議開会前に行ってはならない

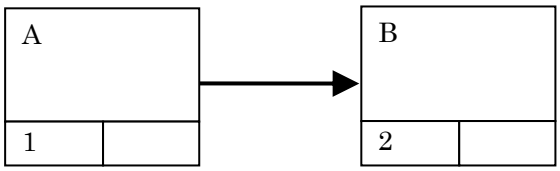
・会議閉会後に議事録を配布しなければならない

この例では、2 つ目と 3 つ目の UOB の間に制約が課せられていないが、制約が存在しないことにより、「委員会報告書の議論」UOB が終了する前に会議を閉会する可能性を残してくれている。そのような場合、中断された会議は、記述されたプロセスの中で再開され

ることになるが、それはどの制約にも違約することなく、スキーム記述の内容との一貫性を保つのである。制約リンクによって表された制約は、どの活動からも「独立（independent）」したものであると理解できる。その為、「委員会報告書の議論」UOB が終了しないまま会議が閉会されたとしても、最後の 2 つの UOB の間の制約は生きており、委員会報告書の議論が終了していなくとも、会議が閉会すれば、会議の議事録の配布が要求されるのである。

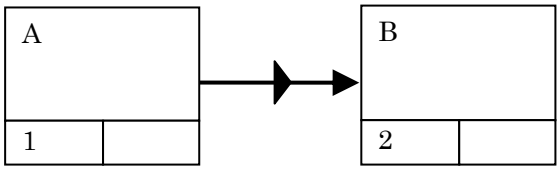
訳注：順序リンクと制約の補足

単純順序リンク



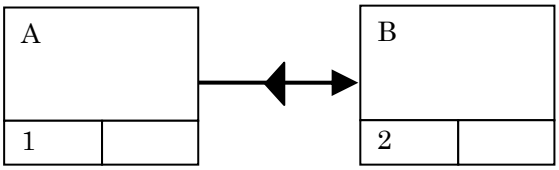
A と B の順序関係のみ記述
・A のみで B 無し、
・B のみで A 無し
の可否については触れていない

制約順序リンク



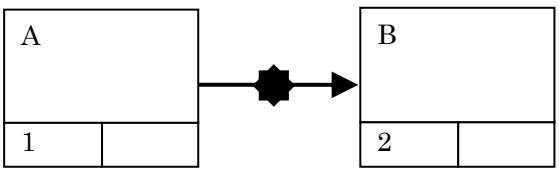
B のみで A 無しは OK
A のみで B 無しは NG

A の全てのインスタンスは、B のインスタンスによって
追従（A be followed by B）されなければならない



A のみで B 無しは OK
B のみで A 無しは NG

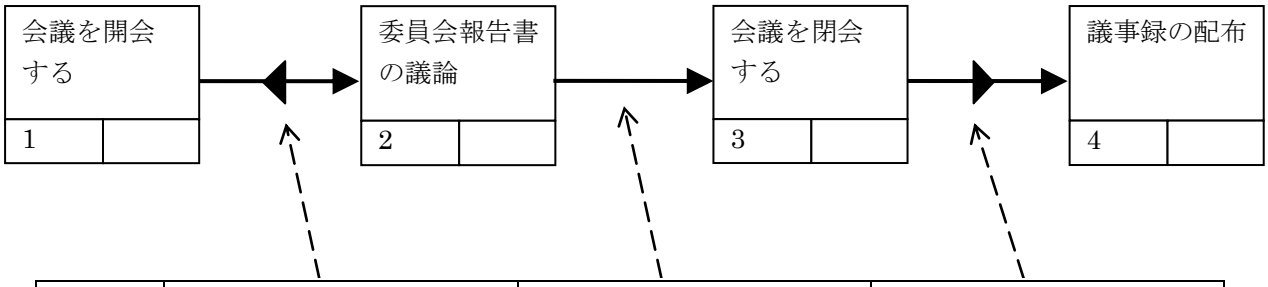
B のインスタンスは A のインスタンスによって
先導（A be preceded by B）されなければならない



A、B は両方必要

A の全てのインスタンスは B のインスタンスによって**追従されなければ**ならず、
かつ、B のインスタンスは A のインスタンスによって**先導されなければ**ならない

訳注：　図 3-8 のスキーム図での制約：



	制約順序リンク（先導）	単純順序リンク	制約順序リンク（追従）
制約的表現	委員会報告書の議論は、会議が開会されてから行われる	会議の閉会は、議論開始よりも後に行われる	会議が閉会されたら、議事録が配布される
可能性のある状況	会議を開会しても、委員会報告書の議論が行われないこともある	議論終了前に、会議が閉会されることもある 会議開会後、議論の無いままに会議が閉会することがある	会議が閉会されないまま、議事録が配布されることもある

３．２．３　交差点（Junctions）

IDEF3 の交差点（junction）は、プロセス分岐のロジックを明確にするメカニズムを提供してくれると共に、複数のプロセスパス間の関係性が持つ、順序とタイミング（時間的関係）の獲得を容易にしてくれる。

３．２．３．１　交差点の種類

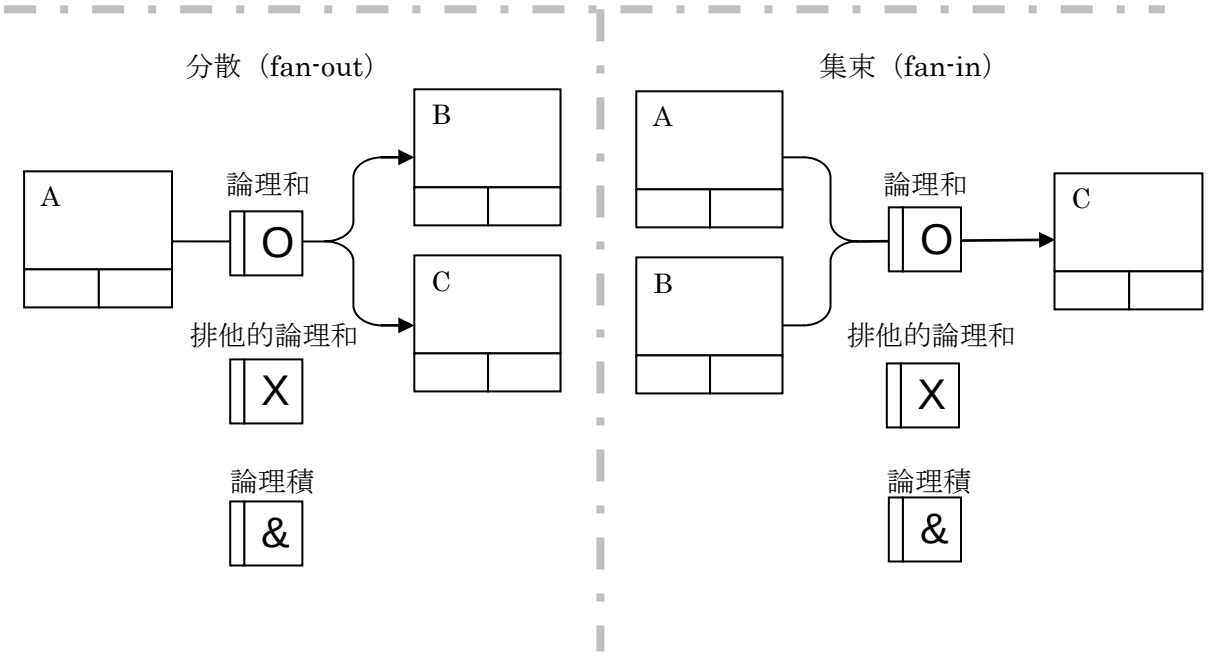
一般的に IDEF3 スキーム図は、複雑なプロセスを種類階層で記述したものである（？be type-level descriptions of complex processes）（例えば、プロセス「種類」等)。その為、プロセスが線形（linear、分岐なしの 1 本の流れ）となることは稀である。殆どの場合、次に挙げる 4 種類の分岐点の何かしらが含まれている：

（１）1 本のプロセスが、複数の並行サブプロセス（parallel subprocess）へと
　　分離（diverge）する交差点
（２）1 本のプロセスが、複数の（非排他的（nonexclusive）なものも含まれる）
　　選択肢サブプロセス（alternative subprocess）へと分離する交差点
（３）複数の並行サブプロセスが、1 本の「スレッド」へと合流（converge）する交差点
（４）複数の選択肢サブプロセスが、1 本のスレッドへと合流する交差点

IDEF3 では、4 種類の交差点を用いて、4 通りの分岐点を表現している。最初の 2 つは、分散（fan-out）交差点によって表現されている。論理積（conjunction AND）分散交差点は、複数の並行サブプロセスへと分離する点を表わしており、論理和（disjunction OR）分散交差点は、複数の選択肢サブプロセスへと分離する点を表わしている。残り 2 つは、集束（fan-in）交差点によって表現されており、論理積集束交差点は、複数の並行サブプロセスが集束する点を表わしており、論理和集束交差点は、選択肢サブプロセスが集束する点を表わしている。論理積交差点（AND 交差点）は「&」文字で表現される。論理和交差点には更に、選択肢の質問が完全に排他的かどうかによって、論理和（inclusive disjunction、OR）と排他的論理和（exclusive disjunction、XOR）の 2 つの種類に分かれている。以上の交差点の分類は、図 3-10 のようになる。詳細については後に説明する。

交差点	集束 （fan-in）	論理和	論理和（OR）	もしくは
			排他的論理和（XOR）	どちらか１つ
		論理積（AND）		かつ
	分散 （fan-out）	論理和	論理和（OR）	もしくは
			排他的論理和（XOR）	どちらか１つ
		論理積（AND）		かつ

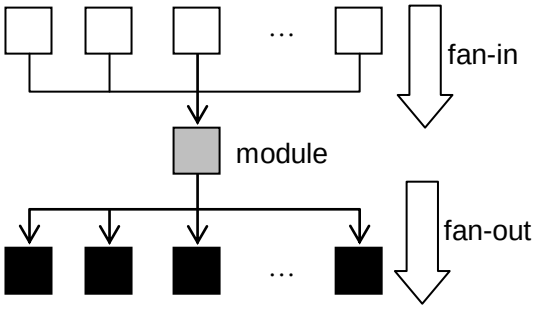
図 3-10　交差点の種類の分類



訳注：fan-in と fan-out

fan-in：ロジック入力数。
特定の１つのモジュールを制御しているモジュールの数
（モジュール依存度の尺度）

fan-out：ロジック出力数。展開する。
１つのモジュールの制御下にあるモジュールの数。



３．２．３．２　基本的な交差点の文法

一般的なプロセスにおいて、交差点は、プロセスの 1 本の「スレッド」が複数の（並行もしくは選択的）スレッドへと分離するか、もしくは複数のスレッドが 1 本のスレッドへと合流する分岐点を表現している。IDEF3 におけるプロセスの分離は、複数の順序リンクの起点の役割を果たしている 1 個の交差点によって、またプロセスの合流は、複数の順序リンクの終点の役割を果たしている 1 個の交差点によって、それぞれ表現されている。複数の並行サブプロセスからの合流と、それらへの分離は、図 3-11 に示されているように、AND 交差点を使用して表示する。

同様に、複数の選択肢サブプロセスの分離と合流もまた、図 3-11 のように表現されるが、AND 交差点の代わりに OR 交差点もしくは XOR 交差点（選択肢が完全に排他的な場合は XOR、そうでない場合は OR）が用いられる。

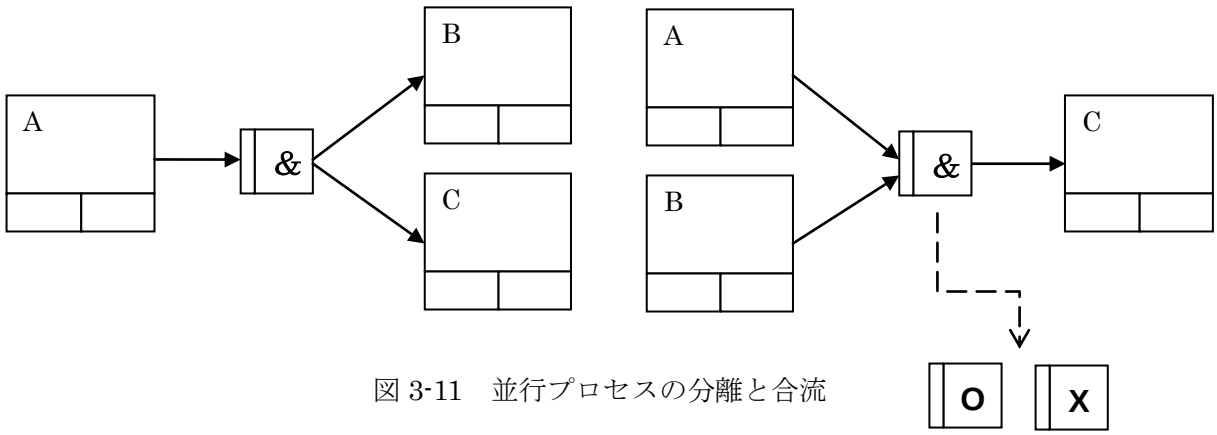


図 3-11　並行プロセスの分離と合流

OR もしくは XOR の場合

慣習として、分散交差点へと入る順序リンクは、それが 1 本の場合には、矢印の頭を省略する。また、分散交差点から出て行く順序リンクは、柄を 1 本にし、また折れ角部を角ではなく丸くする。集束交差点も同様の慣習が用いられる。図示すると、図 3-11 のスキーム図は、図 3-12 のようになる。

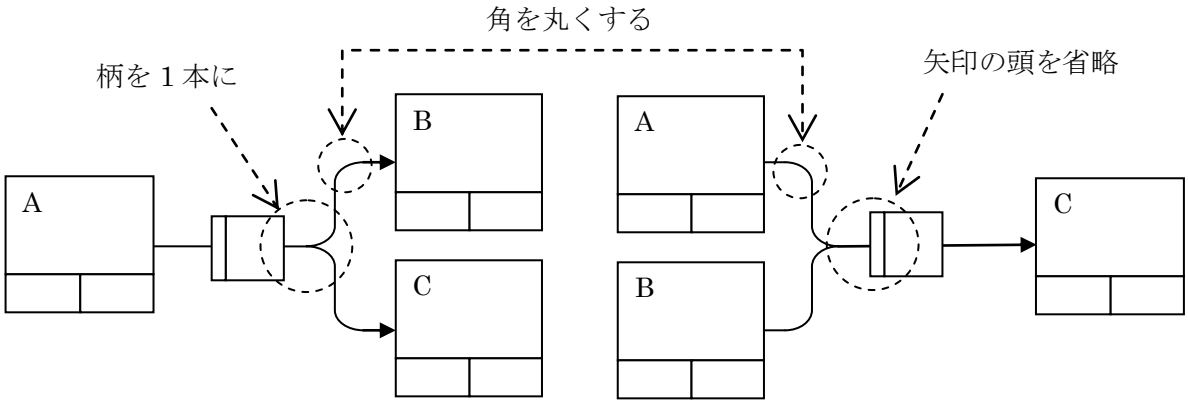


図 3-12　交差点への順序リンクの接続の図示における慣習

交差点シンボルは、分散でも集束でも基本的に同じものを使用する。スキーム上の交差点シンボルによってもたらされる「出来事（occurrence)」の分散・集束の区別は、複数パスが起点側にあるか、終点側にあるかによって判別する。

３．２．３．３　交差点番号の採番要領（numbering scheme）

IDEF3 スキーム図内の交差点への参照を明快にする為に、IDEF3 の交差点用の識別要領（identification scheme）が提案されている。順序リンクに PL を頭文字とする固有の番号が振られていたように、交差点も J を頭文字として、J1、J2…のように割り振るという識別番号要領に従っている。そしてリンクと同様に、異なった 2 つの交差点に同一の番号を振ることは禁止されている。

３．２．３．４　基本的な交差点の意味論（semantics）

（訳注：以下、そもそも難しい内容である上に、書かれている英語が滅茶苦茶で意味不明。他の IDEF3 の解説資料と内容を突合せ、**相当に**意訳している）

スキーム図での分散 AND 交差点が意味する事は、「スキーム図上のプロセスが分散 AND 交差点に至ると、交差点の直後にある全ての UOB 箱が表現する UOB のインスタンスが発生する」ということである。それが同期 AND 交差点であれば、全ての UOB インスタンスは同時に開始されなければならない。

同様に、スキーム図での集束 AND 交差点が意味する事は、「スキーム図上のプロセスが集束 AND 交差点を通過するまでは、交差点の直前にある UOB 箱が表現する全ての UOB インスタンスが存在している」ということである。それが同期 AND 交差点であれば、直前まで存在していたインスタンスは同時に終了しなければならない。

そのため、図 3-13 の左側のスキーム図が表現する活動は、UOB「A」のインスタンスが、「B」と「C」の両方の UOB のインスタンスによって追従された構成になる。同様に、図 3-13 の右側のスキーム図が表現する活動は、UOB「C」のインスタンスが、「A」と「B」の両方の UOB のインスタンスによって先導された構成になる。その際、交差点に「同期」AND 交差点が使用されていれば、スキーム図の活動は、A と B は同時に終了しなければならない。

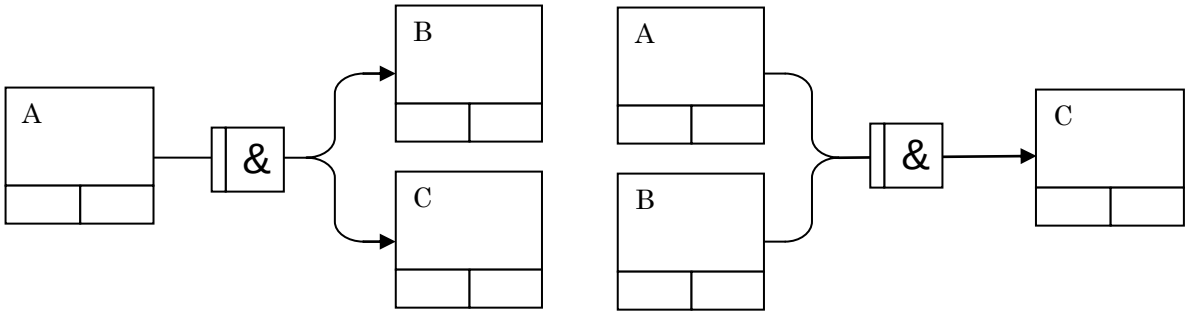


図 3-13　AND 交差点の意味論を説明したスキーム図の例

スキーム図の分散 OR 交差点は、その点を通過する際に、その交差点の右側に接続した UOB の中の少なくとも 1 つのインスタンスが発生することを意味している。同様に、スキーム図の分散 XOR 交差点は、その点を通過する際に、交差点の右側に接続した UOB の中の厳密に 1 つだけの UOB のインスタンスが発生することを意味している。交差点が「同期」OR 交差点であれば、発生するインスタンスは全て同時に開始されなければならない（XOR 交差点の場合は、存在するインスタンスが 1 つのみであるため、この制約は適用できない）。

同様に、スキーム図の集束 OR 交差点は、交差点を通過する時点で、交差点の左側に接続する UOB の内の少なくとも 1 つの UOB のインスタンスが存在しているということを意味している。「同期」OR 交差点の場合には、これらのインスタンス（2 つ以上存在している場合のみ）は、全て同時に終了しなければならない。

そのため、図 3-14 の左側のスキーム図の活動は、UOB「A」のインスタンスが、UOB「B」もしくは「C」もしくはその両方のインスタンスによって追従された構成になる。同様に、図 3-14 の右側のスキーム図の活動は、UOB「C」のインスタンスが、UOB「A」もしくは「B」もしくはその両方のインスタンスによって先導されているという構成となる。もしも図 3-14 のスキーム図が XOR 交差点を用いたものであれば、「B」もしくは「C」のどちらかのインスタンスのみによって先導／追従されることになる。

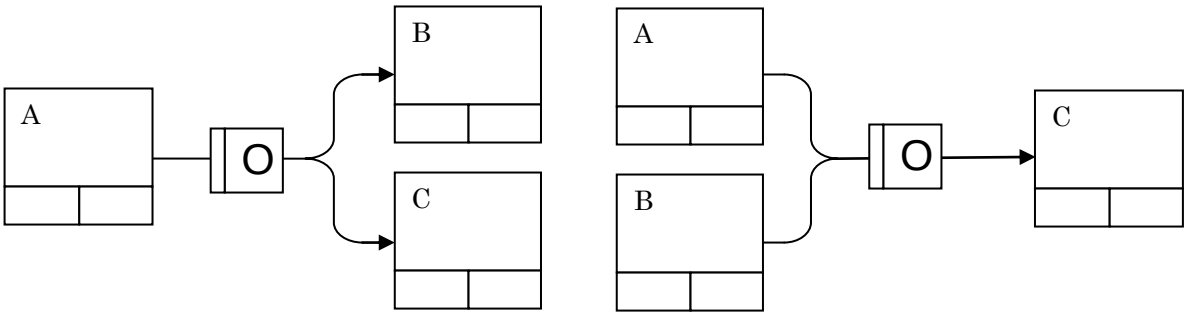


図 3-14　OR 交差点の意味論を説明したスキーム図の例

交差点の実際の意味論からは外れるが、IDEF3 のスキーム図上の交差点は、それに関連する「決定ロジック（decision logic）」を持つものが多い。この交差点の決定ロジックは、後駆の UOB の開始タイミングや順序を決定している。また OR 交差点と XOR 交差点の決定ロジックでは、活動の時点で、プロセスがどのようにして分岐するのかについて説明し

ている。AND 交差点でも同様なロジックが用いられることがある（単なる同時発生でない場合など)。こうした交差点の決定ロジックは、交差点の詳細説明（elaboration）内に記録されている。

プロセスの分離分岐と合流分岐のどちらにも意味と取れてしまう可能性があることから、IDEF3 では､1 つの UOB 箱から複数の順序リンクを引き出すという表現は絶対にしない。また、このような構造は、その分岐がプロセスを並行な複数のサブプロセス（concurrent subprocesses）へと分離するものなのか、1 つの分岐のみがインスタンス化される条件付き分岐（conditional branch）なのかの違いについても、意味論的に不明瞭となる。

しかし交差点を使用する事により、分岐の意味が完全に明確になる。ただし、UOB 箱が複数の矢印の終点となる場合には、許容される例外（「ループバック」と呼ばれることが多いが）が存在する。

（訳注：原文では上文と同じ内容が繰り返されていたので全文端折った）

３．２．３．５　交差点の連結（Combining Junction）

IDEF3 は、複数の並行・選択スレッドが入り組んだ複雑なプロセスを表現する際に、その真の威力を発揮する。その際、プロセスを有りのまま表現する為には、交差点の正しい組合せを見つけ出すことが特に重要となる。そこで、最も基本的な交差点の組み合わせについて、これから説明して行くことにする。

１本のスレッドが複数のスレッドへと分離し、その後で再び１本のスレッドへと合流するプロセスは、一般的に良く見られるものである。IDEF3 では、こうしたプロセスを、分散交差点と集合交差点を組み合わせることで表現している。図 3-15 は、１本のスレッドが並行なサブプロセスへと分離し、その後合流するプロセスを表現している。このサブプロセスは並行に走っていることから、AND 交差点が用いられている。

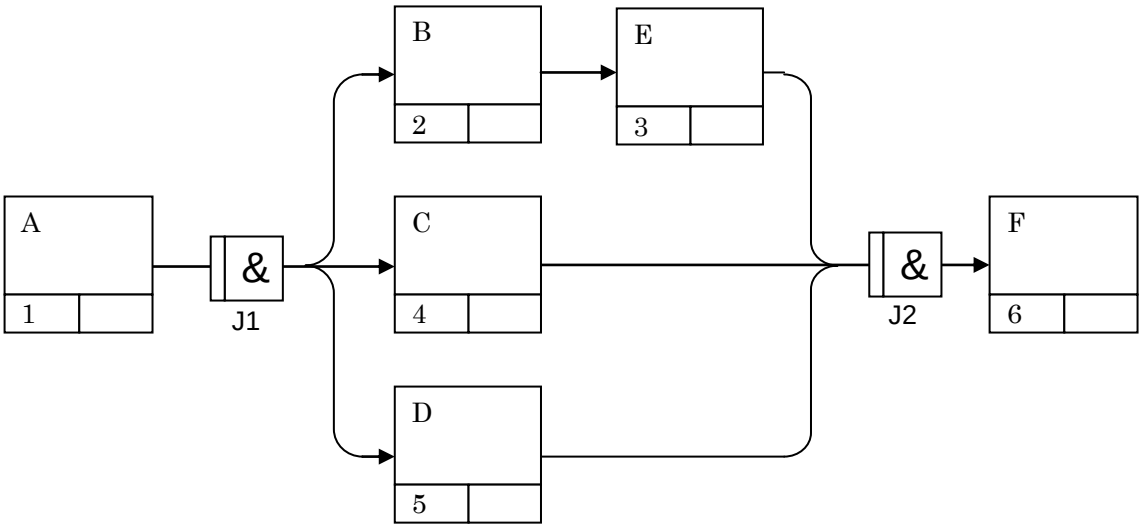


図 3-15　非同期 AND 交差点を用いたスキーム図

図 3-15 のスキーム図の活動は、次のような手順で処理される。

①UOB「A」のインスタンスが終了した後に、B、C、D の 3 つの UOB がインスタンス化される。この時、交差点 J1 は非同期である為、B、C、D の各インスタンスは、どのような順序で開始されても構わない。

②3 つ全てのパスは J2 交差点で合流することから、UOB「F」は E、C、D の全ての UOB インスタンスが終了した後になって、ようやくインスタンス化されることになる。交差点 J2 も非同期であることから、前にある 3 つのインスタンスが終了する順序やタイミングは自由である。

図 3-15 のスキーム図の活動のパターンは、図 3-16 のプロットのようになる。

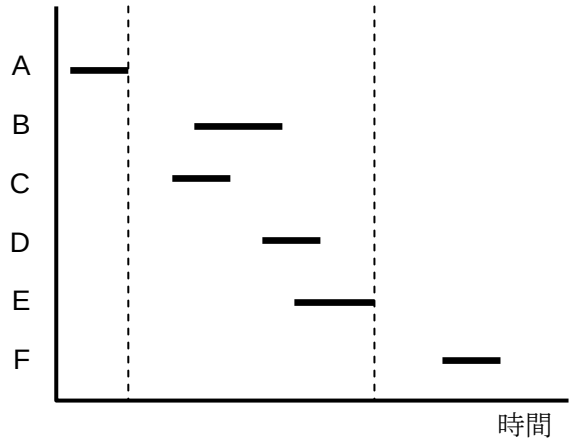


図 3-16　図 3-15 の活動プロット

図 3-17 の順序リンク L1 は、図 3-15 と同じく、UOB「A」のインスタンスが、その後駆の UOB がインスタンス化を開始する前に終了することを要求している。ただし図 3-15 と異なるのは、図 3-17 は縦線を 2 本持った交差点箱で表現される同期ロジック（synchronous logic）で構成されている事である。この同期 AND 交差点 J1 は、UOB「B」、「C」、「D」が同時にインスタンス化されるということを示している。同様に、同期 AND 交差点 J2 は、交差点 J2 を通過して UOB「F」のインスタンスへと引き継がれる前に、UOB「B」、「C」、「D」のインスタンスが同時に終了するということを表わしている。

図 3-17 のスキーム図の活動プロット（図 3-18）には、同期制約による一時停止構造が表われている。

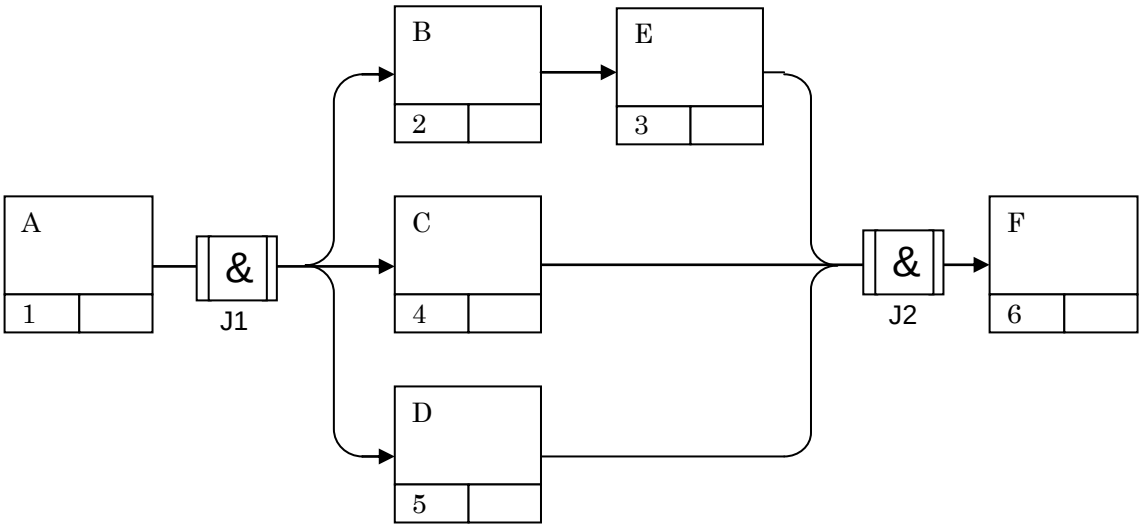


図 3-17　同期 AND 交差点

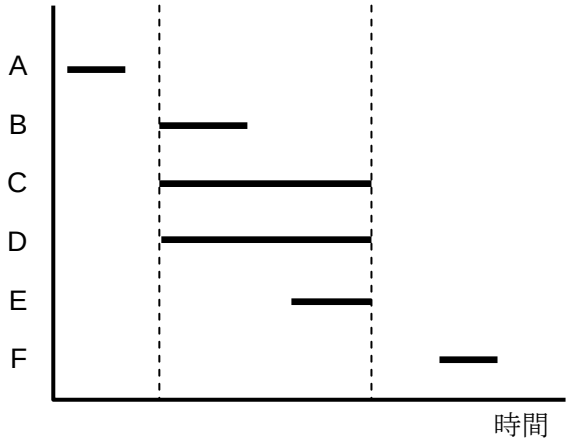


図 3-18　図 3-17 の活動プロット

図 3-19 は、図 3-15 の交差点 J1 と J2 が、非同期 OR 交差点になったものである。この中で交差点 J1 は、インスタンス A の後で UOB「B」、「C」、「D」の 1 つ以上でインスタンス化が行われるように指示をしており、それにより 1 本から 3 本の「スレッド」が活動することになる。そして J2 も非同期 OR 交差点であることから、これらのスレッドの内の 1 本でも終了すれば、インスタンス F がインスタンス化されることになる。

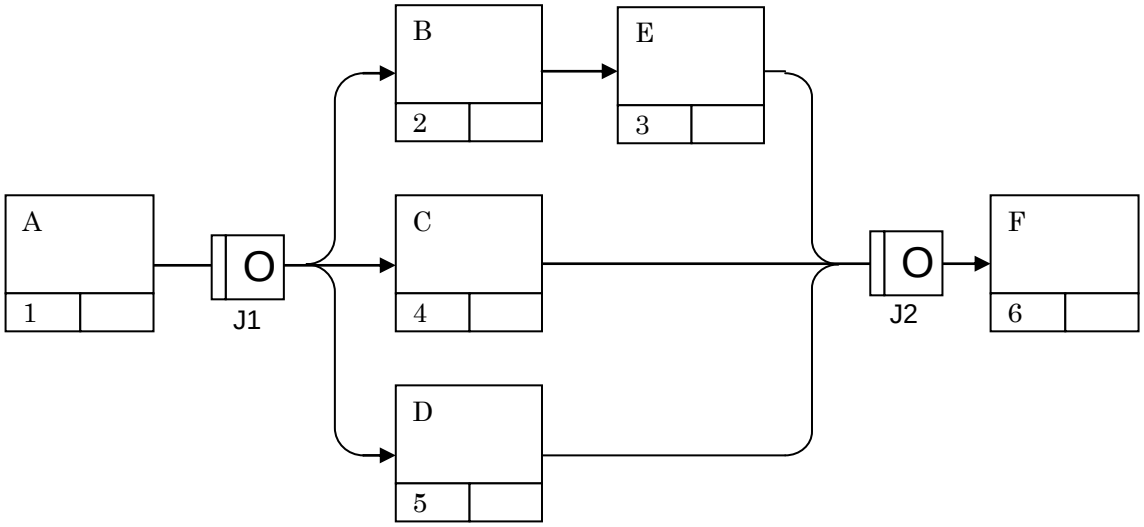


図 3-19　非同期 OR 交差点

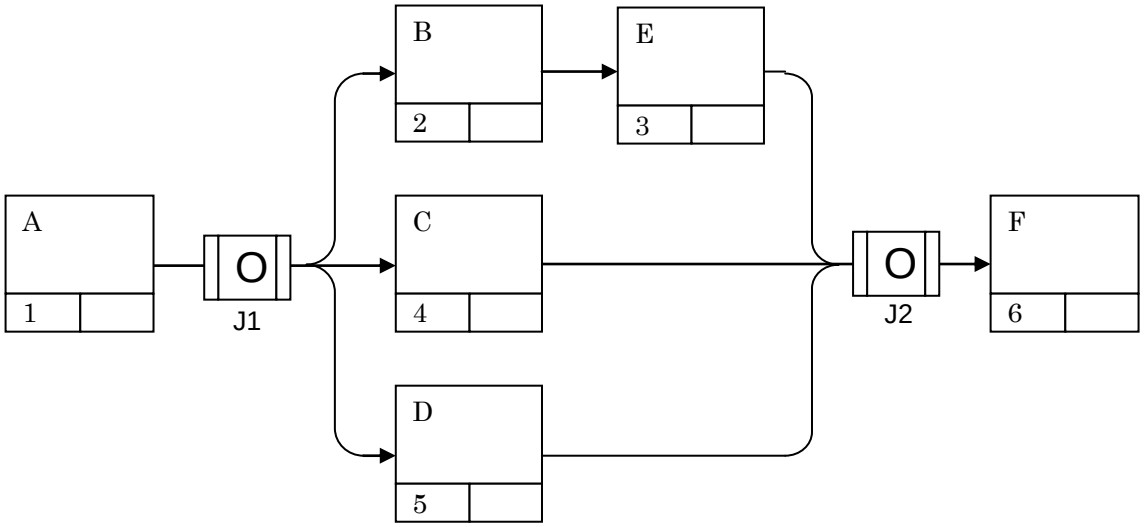


図 3-20　同期 OR 交差点

図 3-20 は、2 つの同期 OR 交差点を組み合わせたものである。分散 OR 交差点は、UOB「B」、「C」、「D」の 1 つ以上のインスタンスが、インスタンス A の後に開始されることを示している。

交差点が同期であることから、インスタンス A の後に開始される 1 つ以上のインスタンスは、同時に発生する。仮にそれが UOB「B」のインスタンスだった場合、図 3-21 の左の活動プロットで示されているように、インスタンス B を追従する UOB「E」のインスタンスは、インスタンス B と同時に開始された別のインスタンス D と同時に終了することになる。UOB「B」がインスタンス化されなかった場合の活動は、図 3-21 の右の活動プロットのようになる。このプロットにおいて注目すべきは、交差点 J1 と J2 の両方の制約により、UOB「C」と「D」の 2 つのインスタンスの開始と終了とが同時となる事である。

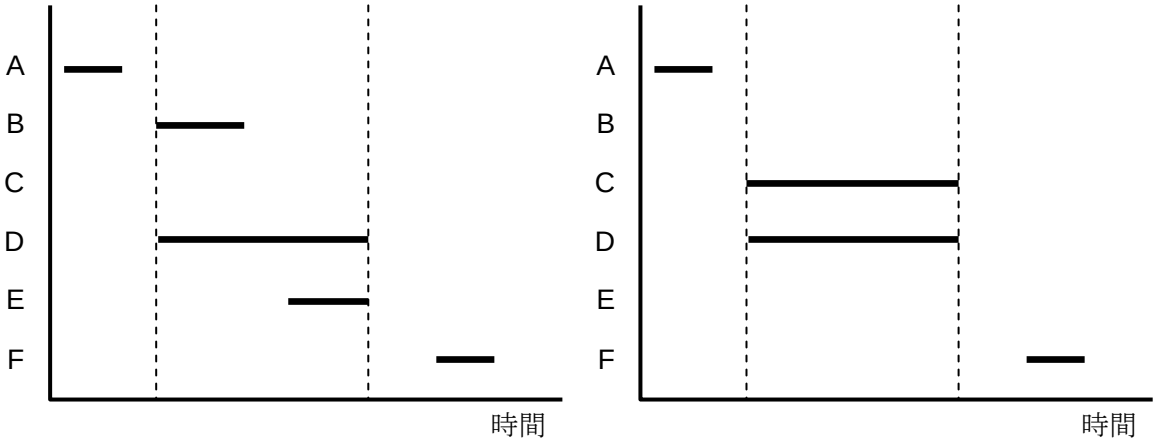


図 3-21　図 3-20 の活動プロット

図 3-22 は、より自由な活動のタイミングと順序を表現する為に、異なった種類の 2 つの交差点を組み合わせた例を示している。

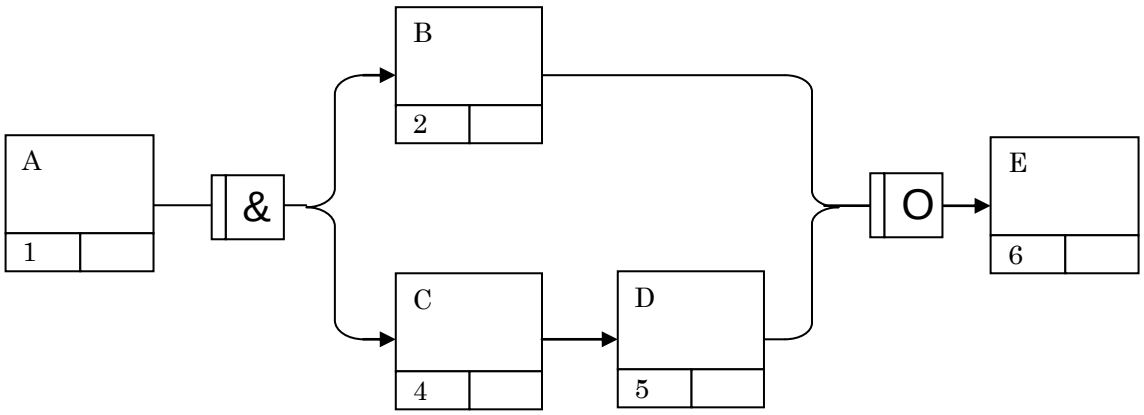


図 3-22　集束 OR 交差点に追従された分散 AND 交差点

図 3-22 の表している活動では、UOB「B」と「C」の 2 つのインスタンスは、インスタンス A の終了後に発生するが、インスタンス B もしくは C のどちらかが終了するか、もしくはそれらがインスタンス化すら行われていない場合であっても、集束 OR 交差点とそれに続くインスタンス E の発生を処理する場合も有る。そこで 2 本のスレッドが合流する交差点に非同期集束 OR 交差点を使用する事で、そうした活動を表現している。

プロセス活動が終了するには、両方のスレッドが、同時であろうと別々であろうと終了していなければならないものの、インスタンス E の開始条件は、片一方のスレッドの終了のみで十分である。

図 3-23 は、図 3-22 のスキーム図での、基本的な 3 種類の活動パターンをプロットしたものである。

左のプロット図は、OR 交差点が AND 交差点であっても許容される活動パターンである（もしくは、OR 交差点が同期交差点だった場合)。ここでは、インスタンス B と D（インスタンス C もまた）の両方が、インスタンス E の前に終了している。

真ん中のプロット図では、インスタンス B が終了する前に（もしくは開始すらされない内に)、インスタンス E が開始されている。

右側のプロット図では、インスタンス D が終了する前に、インスタンス E が開始されている。

もちろん、活動に制約を更に追加する事で、発生し得る活動の組み合わせを狭める事も可能である。例えば、「活動内のインスタンス B は、常にインスタンス E が終了した後に開始される」という制約を追加する事で、真ん中のプロット図のようなパターンのみに絞ることができる（インスタンス E の終了後に、インスタンス B が開始されている)。

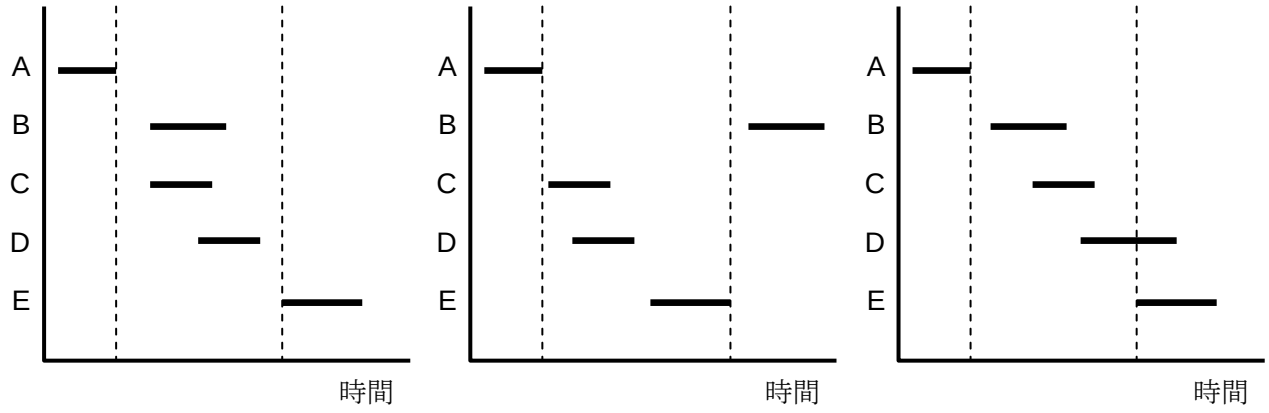


図 3-23　図 3-22 の活動プロット

３．２．４　具体的な例

これまでに紹介した構造を更に詳細に説明する為に、幾つかの例を挙げる。

図 3-24 では、提案の受領が、コスト評価と技術的評価によって追従されたシナリオを示している。それぞれの評価は、契約締結の前に終了されなければならない。交差点が非同期である事から、2 つの評価の開始と終了のタイミングの関係性には何の制約も課せられていない。2 つの評価は、単純に提案の受領に追従し、契約締結を先導しているだけである。

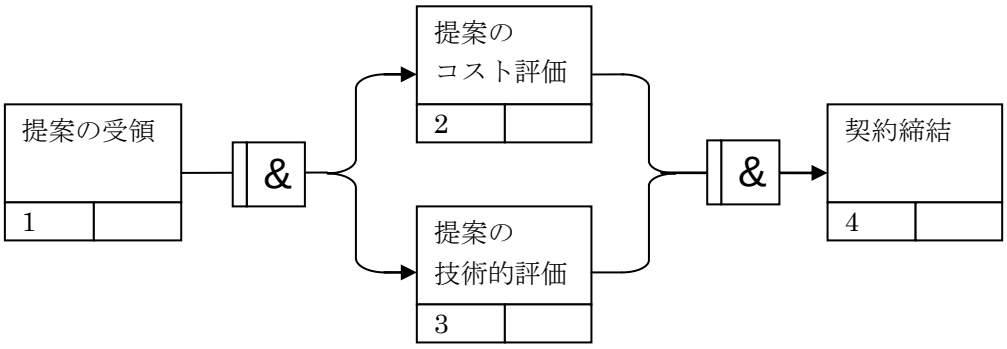


図 3-24　非同期 AND 交差点の例

これとは対照的に、図 3-25 のシナリオでは同期 AND 交差点を使用することで、コスト評価と技術的評価は同時に開始しなければならないが、終了は別々でも構わないという状況を表現している。終了も同時でなければならない場合には、もう一方の交差点も同期 AND 交差点を用いる事で、意図したプロセスを表現することが出来る。

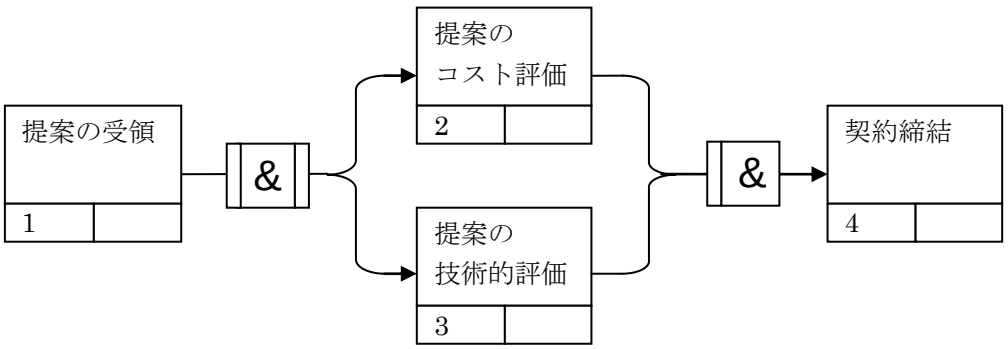


図 3-25　同期 AND 交差点の例

図 3-26 では、契約者を選択するシナリオが記述されている。このプロセス記述では、評価が行われた後で提案が拒否されるか、もしくは提案本体のみ受託、提案オプションのみ受託、その両方の受託が契約の締結前に行われる、という状況を表わしている。

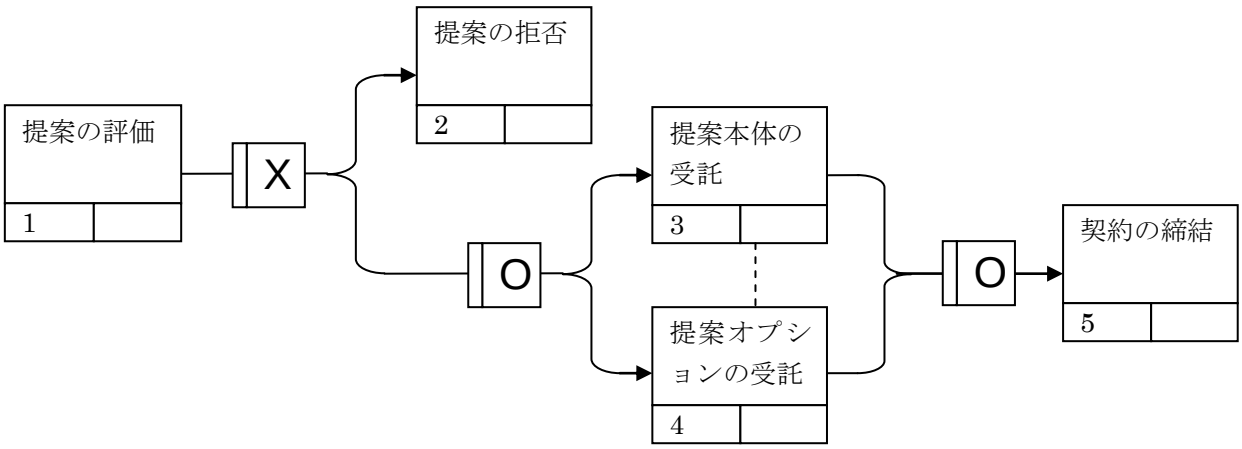
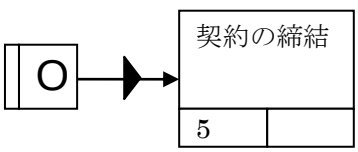


図 3-26　非同期 OR 交差点の例

図 3-26 で図示されているシナリオには、「提案の拒否」という中断活動が含まれているが、他の 2 つのどちらか、もしくは両方の活動は、契約締結へと至っている。また、UOB「提案本体の受託」と UOB「提案オプションの受託」の間に引かれた関係性リンク（点線）は、2 つの UOB 間に存在する何らかの関係性を示している。そしてこの記述は、交渉が失敗した際に何が発生するのかが示されていないことから、未完成であるといえる。例えば、年金積立基金（the funding agency）では、契約の締結は殆どの場合、契約者による契約の容認（contractor acceptance of the terms）に依存しており（depend upon）、交渉プロセスの一部として、契約者に再提案（resubmit the proposal）を要求している。こうした状況は、図 3-26 のスキーム図に何かを追加するか、もしくは「契約の締結」を分解して詳細を記述することにより、IDEF3 でも簡単に表現する事が可能である。このスキーム図のままでは「契約の締結」を要求するものは何も存在していない。（左から右の方向で）「制約」順序リンクが、集束 OR 交差点から UOB「契約の締結」への接続に使用されている場合にのみ（訳注：下図）、契約の締結が強制される。



制約順序リンク（追従）：
「契約の締結」のみで「O」無しは OK
「O」のみで「契約締結」無しは NG

交差点の組み合わせによっては、誤ったプロセスロジックを表現することもある。特に図 3-27 のような形で、発散 XOR 交差点が集束 AND 交差点の前にあるとプロセスに不一致が生じてしまい、プロセスが活動できなくなってしまう。

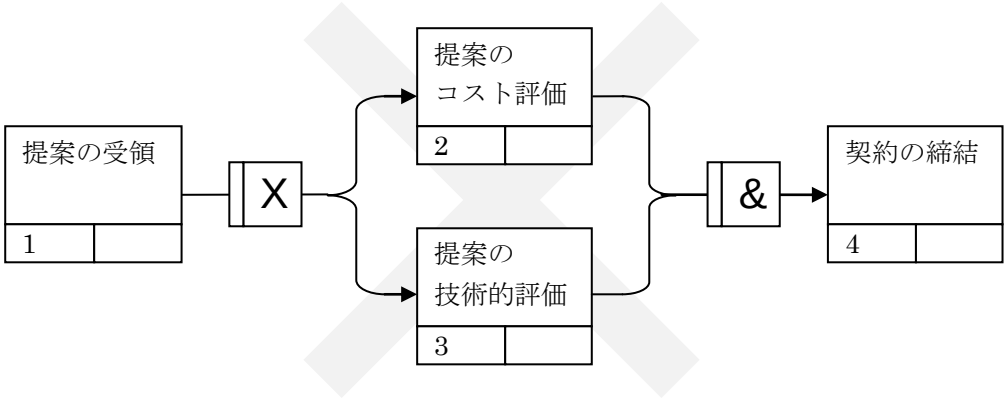


図 3-27　誤った XOR／AND 交差点構造の例

図 3-27 では、「提案の受領」UOB の後に XOR 交差点があり、その後で「提案コスト評価」UOB もしくは「提案の技術的評価」UOB の 2 つの UOB へと続いている。これは、どちらか一方の UOB のみが実体化され、スキーム図上で活動化されるということを示している。しかし、2 つの UOB の後を受けている AND 交差点は、両方の UOB が処理されなければ実体化されない為、その後に続く「契約の締結」UOB は決して活動化しないことになってしまう。こうしたスキーム図は誤った構造を表現しているものの、既存（AS-IS）のプロセス内にある未発見の矛盾点を獲得するには便利である。図 3-27 のような特徴を持つ状況では、恐らく契約が締結されることは決してないことから、IDEF3 スキーム図は、組織的な問題点を識別し、矛盾の解決を可能にしてくれることになる。こうした誤った種類の構造は、理想（TO-BE）のシステム、組織構造、プロセスの表現には決して使用してはならず、こうした構造が存在していたとしても、記述検証プロセスによって、IDEF3 スキーム図のエラーとして識別、訂正されることになる。

３．２．５　UOB の分解（Decomposition）

詳細説明（elabration）によって、プロセスについての詳細な知識の獲得と構造化とを行う。対象となるスキーム図上の 1 つの箱によって表現されている UOB が非常に複雑な物であった場合、その UOB を、その UOB を構成する複数の UOB へと判り易く「分解（deconpose)」すると便利である。IDEF3 のこうした表現方法を用いると、分解元の箱と相互関係を持つ別の IDEF3 スキーム図上で UOB が「爆発的」に増加してゆくことになるが、このような構成を採る事により詳細階層を用いた UOB の表現が可能となる。この詳細説明を行うスキーム図は、元 UOB 箱の「分解（decomposition)」として知られている。分解を行うことにより、ユーザーは様々な抽象化階層で記述が可能となると共に、複雑性の取扱いの強力なメカニズムである「分割による征服（devide and conquer)」原理を適用する事も可能となる。この原理を繰り返し適用する事により、どのような詳細階層においても、プロセス記述を構成することが可能となる。また、分解を用いる事で、異なった知識ソースや視点から、同一のプロセスをモデル化してゆくことも可能になるのである。こうしたことが可能になるのは、IDEF3 では、異なった分解や「視野（view)」での UOB それぞれが固有の番号を持っているからである。またこの IDEF3 の能力は、対象となるプロセスが、複数の機能的組織に跨っている分野においても有用である。

分解された UOB は、そのまま別の IDEF3 スキーム図となる。図 3-28 は、契約の運用管理分野での分解の使用例を示している。分解を行う UOB「契約の受領と実行（Receive and Activate Contract)」を表わした UOB 箱 3 は、「親 UOB 箱」と呼ばれている。同じ UOB を参照する別の UOB 箱が存在しない場合、等の曖昧な状態でなければ、「箱」を略して単に「親 UOB」と呼ぶことも可能である。

親箱の分解物は、それぞれが「子」分解物となる。それぞれの子分解物には、同一の親 UOB の分解によって発生したものであることを識別可能な、ラベルと固有番号が与えられている。分解物の中にある UOB 箱が、更に分解されることもある。（図に有るように、分解には特殊な参照番号の採番要領が要求されている。この採番要領については次節で説明する。それまでは、各 UOB 箱の番号の右端の番号を UOB 番号としておく。その左にある 2 つの番号は、IDEF3 スキーム図の読者に追加情報を提供している）

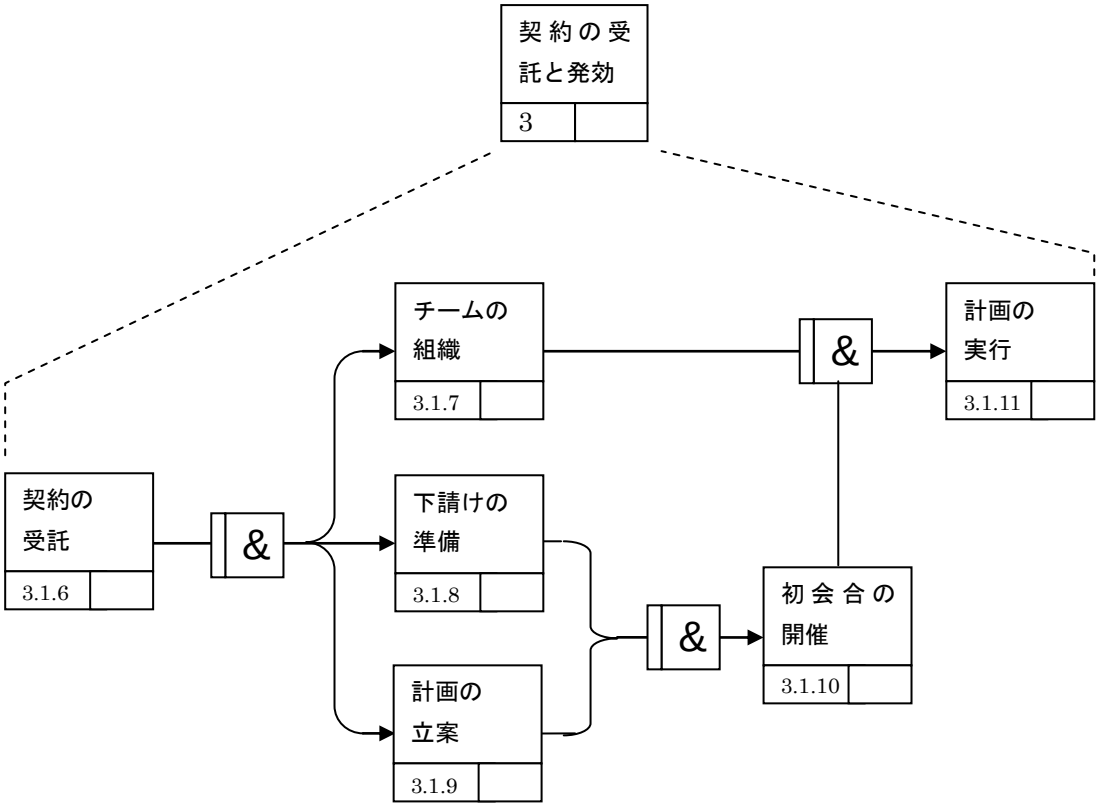


図 3-28　UOB「契約の受託と実行」の分解 3.1

複数の視界（view）によって行われた別々の分解は、客観的視界（？objective view、目的別視界？）へと、まとめられることになる。図 2-29 で表された視界は、UOB「初会合の開催（Hold Kick-off Meeting)」の客観的視界の例を示したものである。しかし、契約のプロジェクトマネージャーは、このプロセスに関して独自の見方（perspective）を持っている事から、IDEF3 ではプロジェクトマネージャーの視点による UOB の分解を別途表現することを可能にしている。この際のプロジェクトマネージャー視点で UOB「初会合の開催」を分解したスキーム図を、図 3-30 に示す。

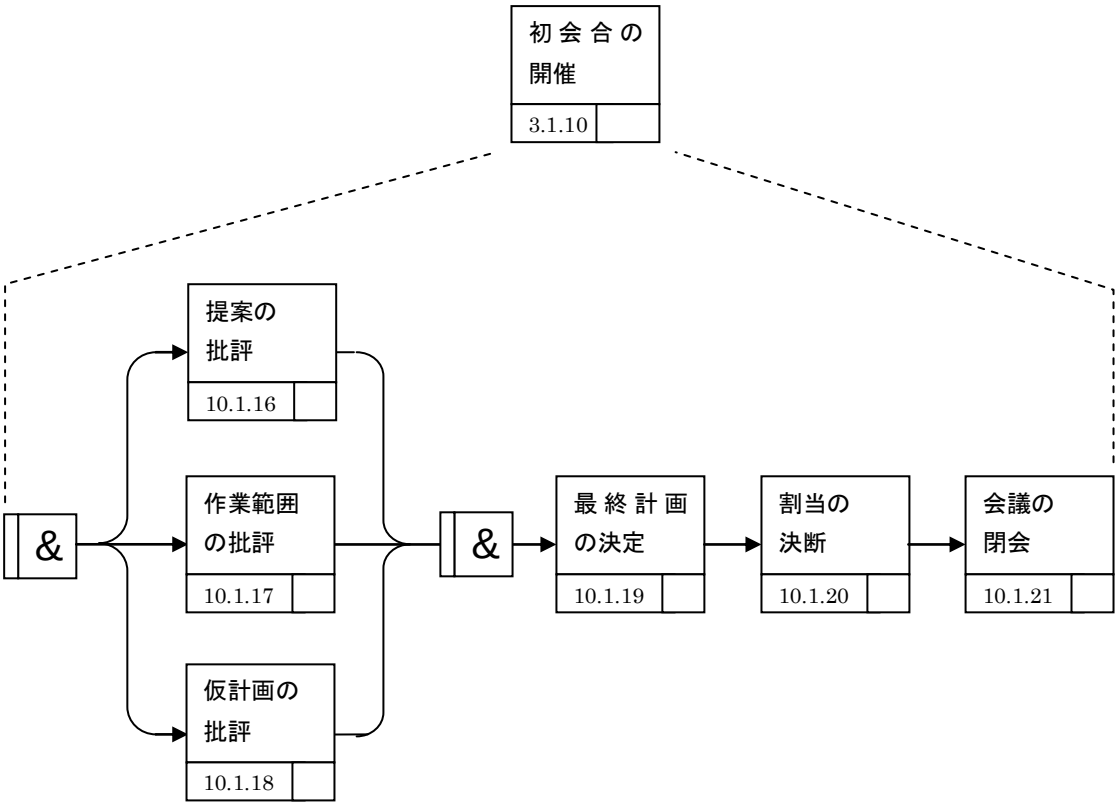


図 3-29　UOB「初会合の開催」の分解 10.1

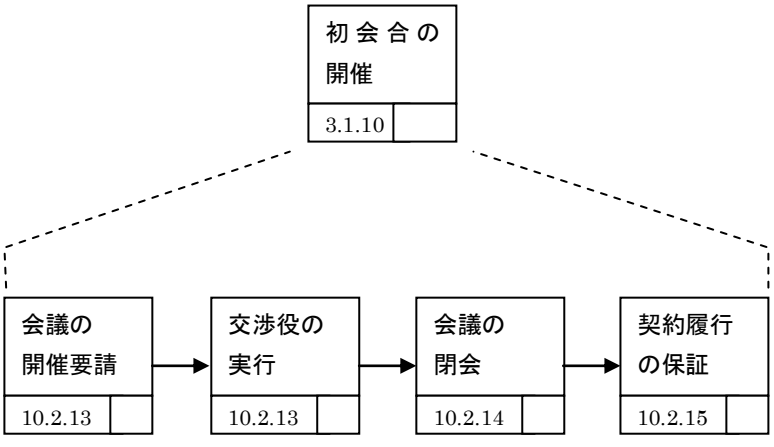


図 3-30　プロジェクトマネージャーの視界での分解

３．２．６　UOB 参照番号の採番要領（UOB Reference Numbering Scheme）

IDEF3 プロセス記述では、個々の UOB 箱に対して UOB 箱番号が割り当てられている。一般には、1 つの IDEF3 記述には多くの UOB 箱が含まれており、そしてそれぞれの UOB 箱が更に複数回分割されることで、非常に複雑なものとなる可能性が有る。こうした複雑なスキーム図では、単純な番号を箱に割当てて個々の箱を識別できるようにしただけでは、そこから十分な情報を得ることは困難である。特に単体の数字だけの UOB 番号は、その UOB についての背景（context)、例えば全体のプロセス記述のどの部分に UOB が存在しているのかという情報等について、何も与えてくれない。この背景情報を提供する為に、IDEF3 のスキーム図では、より複雑な採番システムが使用されている。分解の最上層だけは特別に、箱の UOB 番号と参照番号が同一なものとなっている。それよりも下の分解階層では、UOB 箱の参照番号はピリオドで区切られた 3 組の数字によって構成されている。左端の数字は、その UOB 箱の親 UOB の参照番号の右端の数字である。真ん中の数字は、その UOB を生成する固有の親箱分割に対して割り当てられている数字である。(一般には、数字は分解や UOB 箱が作成された順番に割り振られているが、任意の番号を割り振っても良い)。そして右端の番号は、その UOB 箱の通しの箱番号である。以上のような採番要領により、UOB 箱の通しの箱番号と、箱がどの分割に所属しているのか、そして元になった親 UOB の番号の、3 つの情報を表現している。参照番号の割当ては、図 3-31 のようになる。

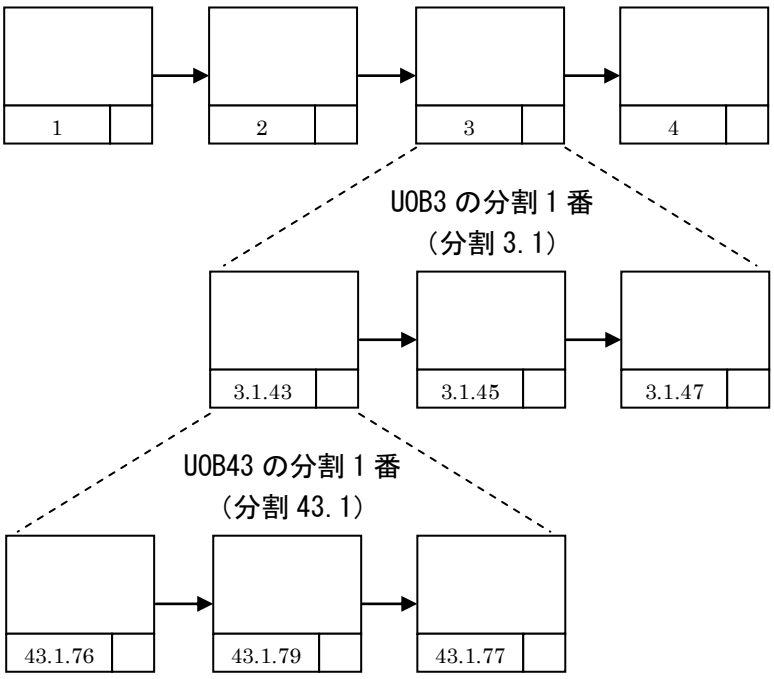


図 3-31　行動ユニット（UOB）の採番要領

記述の作成に複数人が関わる場合には、全ての UOB 箱に固有の箱番号と参照番号が間違いなく割り振られるようにする為に、採番に制限を設けるべきである。例えば、作業者毎に番号のセット（例えば山田さんは 1 から 99、佐藤さんは 100 から 199、等）を割当て、作業者は自分に割り当てられた番号セットの中から UOB 箱の番号を選択しておき、最初に割り当てられた番号セットを使い果たした場合には、必要に応じて番号セットを追加する、というやり方である。こうした採番手続きを適用する事で、開発作業チームの分析責任者（lead analyst）は、最終的に記述を 1 つにまとめてゆく際に、全ての UOB が固有の箱番号と参照番号を持っている事を保証できるのである（※4）。

※4：

UOB の参照番号採番要領により、協調したチーム作業が促進され、記述内の複数の視界や異なった詳細階層の中での行き来が容易になっている。この参照番号採番要領は、紙ベースの環境や、ソフトウェアツールが統合を限定的にしか支援してくれない場合に、特に重要となる。一方で、表示には便利であっても、UOB 参照番号の表示や UOB の番号付を、スキーム図に表示されている順番（左から右、上から下へ、等）にすることは避けるべきである。

３．２．７　不完全な記述（partical Descriptions）

UOB 箱はリンクによって結合されている。IDEF3 は記述獲得に焦点を置いていることから、図 3-32 で図示された例のように、IDEF3 スキーム図の他の部分とのリンクを持たない UOB を表現する事も可能である。こうした単独表示は、一般的に対象分野の専門家が相互関係に言及すること無く、イベントや活動の存在を指摘してゆかなければならない、事実収集活動の初期段階で用いられている。

リンクの無い UOB 箱により、実際の状況を表現したり、対象分野の専門家のリンクの有無についての知識の不確かさを反映することが可能である。図 3-32 の UOB4 の単独表記の場合、スキーム図は実際の状況を表現している。この図 3-32 では、スキーム図内の「プロジェクトマネージャーが進捗と日程を比較する」UOB が、「プロジェクト日程」オブジェクトをスキーム図の他の UOB の間で共有しているという概念が、表現されている。IDEF3 手法は、このような独立した UOB の作成を許容することにより、「不完全な記述（partial descriptions)」の作成を促進しており、ユーザーは自身が認知している作業範囲（state of the world）を、不完全なままで表現する事が可能となっている。実際に、記述の開発の中で発生するエラーの多くは、本質的に不完全な記述のセットを「無理やり完全なものに」しようとすることで発生しているのである。

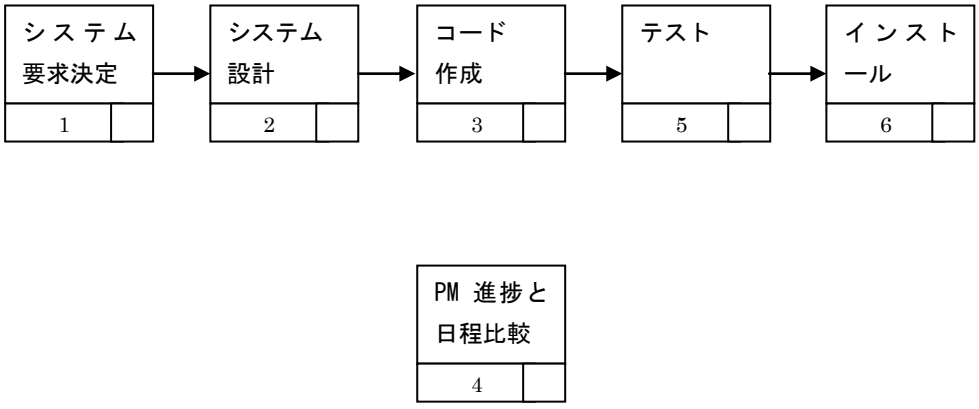


図 3-32　非接続の UOB の例

３．２．８　参照指示（Referents）

参照指示（referent）は、プロセススキーム図とオブジェクトスキーム図の両方に対して、理解の強化と意味の追加、そして構造の単純化（散らかりの最小化）を実現してくれる。IDEF3 のプロセススキーム図とオブジェクトスキーム図で使用されている参照指示は、以下の事を行っている：

（１）既に定義済の UOB に対して、UOB を二重定義すること無く、その UOB の別のインスタンスをプロセス内の特定の点で発生させるように指示することで（ループバック無しで）、参照を行う。

（２）プロセスの中での、管制の受け渡し（？transfer control）、もしくはループバックの指示

（３）プロセススキーム図とオブジェクトスキーム図の間での、参照やリンクの形成

図 3-33 は、参照指示の 2 つの基本的な図形シンボルである。どちらの種類の参照指示も、プロセススキーム図とオブジェクトスキーム図の両方で使用されているが、プロセススキーム図では呼出続行（Call-and-Continue）形式の参照指示が広く使用されている傾向があるようである。呼出続行参照指示を使用すると、参照先の要素が開始されれば、その要素の進行が終了していなくとも、焦点が置かれている IDEF3 要素（すなわち、参照を行う IDEF 要素）が最終状態まで進行可能になる。一方、呼出待機（Call-and-Wait）参照指示を使用すると、開始された参照先の要素が終了するまでは、焦点が置かれた IDEF3 要素は待機させられる。

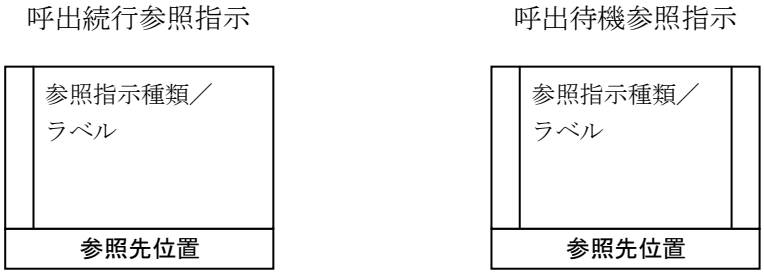


図 3-33　参照指示のシンボル文法

参照指示によって表現されている参照指示種類（referent type）:「UOB」、「シナリオ（SCENARIO)」、「TS（変移スキーム図)」、「GO-TO」の何れか１つの後に「／（スラッシュ)」を挟み、後ろに参照指示によって表現されている物のラベルを書く（例：UOB／作業範囲分析の実行)。参照指示には、表現されている物の位置（locator）を示す欄も含まれている。参照指示種類とラベル付け要領を、図 3-34 にまとめておく。

参照指示種類	参照先ラベル	位置の指定手段
UOB	UOB ラベル	UOB 番号
シナリオ（SCENARIO）	シナリオラベル	シナリオ番号
TS（Transition Schematic、変移スキーム図）	変移スキーム図ラベル	変移スキーム図番号
GO-TO（プロセススキーム図のみで使用）	UOB ラベル	UOB 番号／シナリオ番号 もしくは、 分解番号（ID 発生時）
	シナリオラベル	シナリオ番号
	交差点種類（&、O、XOR）	交差点番号／シナリオ番号 もしくは、 分解番号（ID 発生時）

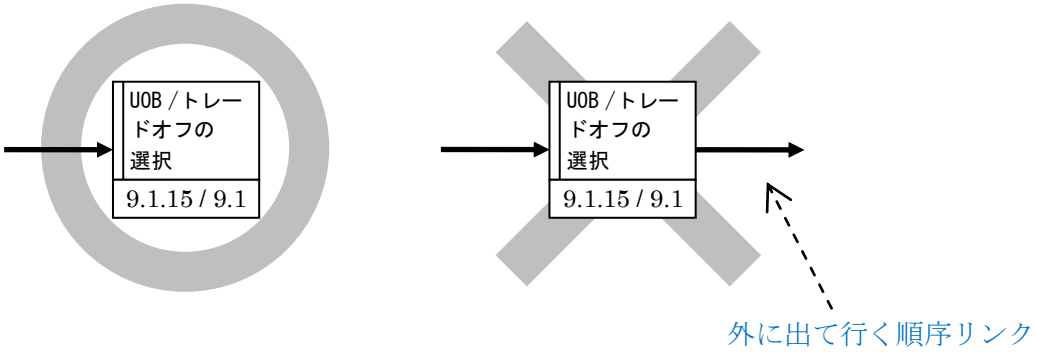
図 3-34　参照指示シンボルの構成

次の節では、プロセススキーム図とオブジェクトスキーム図の両方で使用可能な、参照指示の意味論（sematics、意味論）についてまとめている。参照指示に関する全ての意味論を知りたければ、オブジェクトスキーム図について更に詳しくならなければならない。次に挙げる、プロセススキーム図での使用について特に注意を置いて書かれた、呼出待機参照指示と呼出続行参照指示について説明を読むことも、その手助けとなるだろう。オブジェクトスキーム図についての基本的な理解を得た後で、もう一度、呼出続行参照指示と呼出待機参照指示の説明を読み返すことで、理解を完全なものにすることができる。

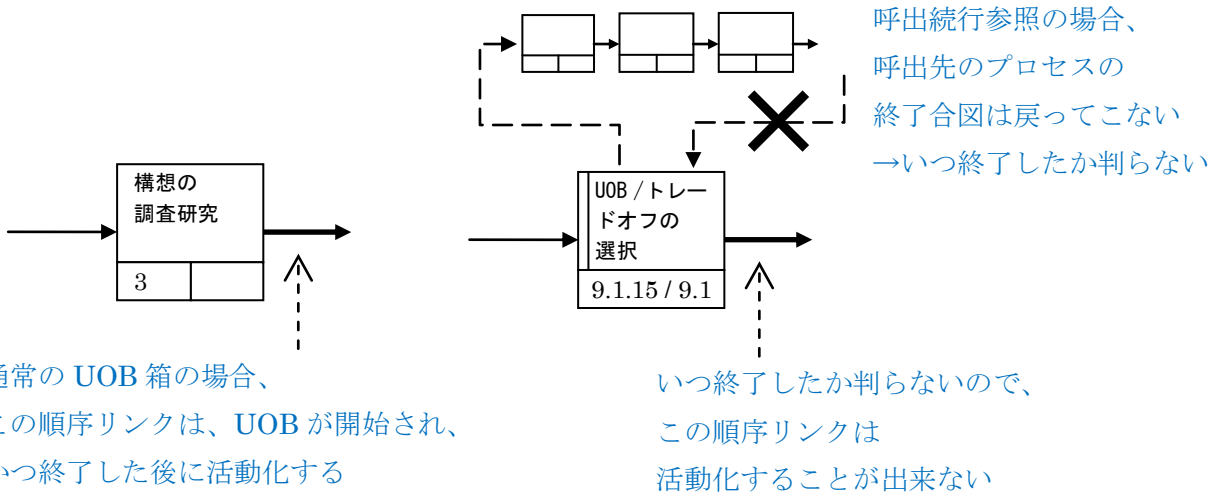
３．２．８．１　呼出続行参照指示（Call-and-Continue Referents）

（訳注：全体的に意味不明な部分が残る）

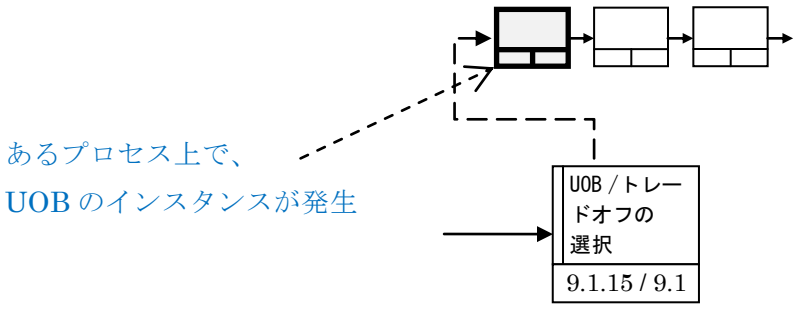
呼出続行参照指示の種類が「UOB」、「シナリオ」、「GO-TO」の何れかの場合には、外に出て行く（outgoing）順序リンクを持つことは無い。出してしまうと、順序リンクの意味論と一致しなくなる（訳注：下図）。



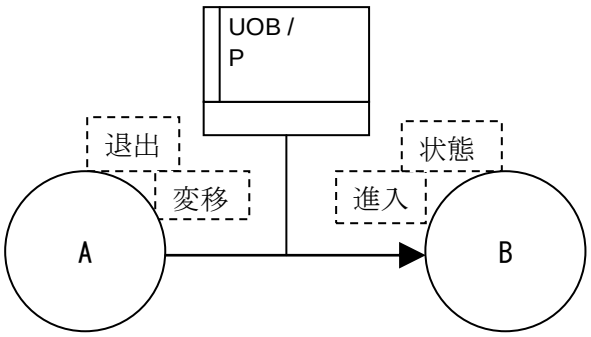
呼出続行参照指示では、例えば参照先の UOB のインスタンスが開始、終了に関係無く、呼出元のプロセスは続行される。しかし順序リンクの制約では、UOB のインスタンスが開始され、かつ終了しなければ、プロセスを続行できないのである。つまり、2 つの異なった意味論を、文法を壊すことなく同時に適用する事は不可能であることから、呼出続行参照指示に対して、外に出て行く順序リンクを与える事はできないのである（訳注：下図）。



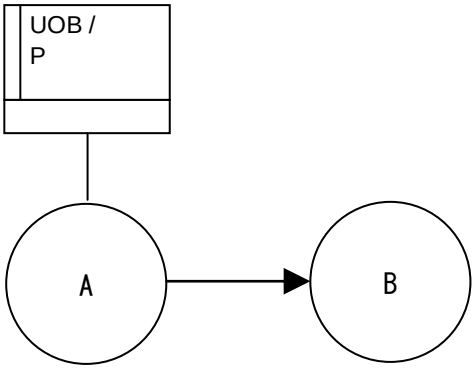
参照指示の種類が「UOB」だった場合、ラベルは UOB のラベルと同じにしなければならない。参照指示の種類が UOB である場合には、予め定義されていた UOB の別のインスタンスが、（ループバックの無い）プロセス上の、ある特定の点で発生している事を意味している。



オブジェクトスキーム図上にある変移ライン（transition arc）に、この種類（UOB）の参照指示が付けられている場合（下左図）には、参照先の UOB の活動は、状態変移（state transition）が許可される前に（退出条件が満たされた後で、変移条件が満たされる前？）開始されなければならない（オブジェクトスキーム図について説明されている小節を参照のこと）。



訳注：「図 3-41　基本状態変移スキーム図上の UOB 参照指示」を引用、追加

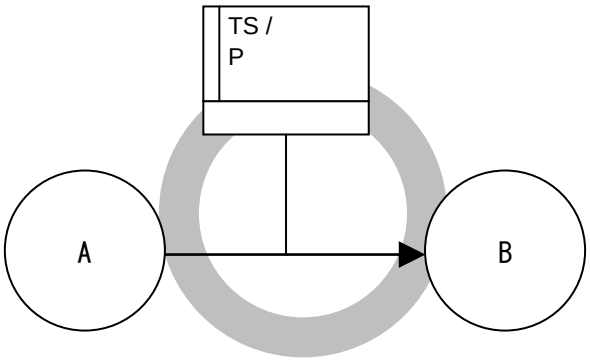


訳注：「図 3-46　オブジェクトをある状態に保ち続けている構成」を引用

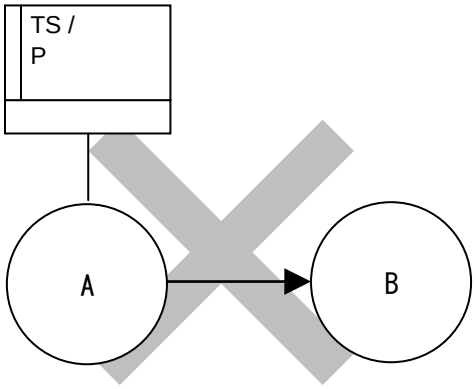
また、オブジェクトスキーム図上にあるオブジェクト状態（object state）に、この種類の参照指示が付けられている場合（上右図）には、参照先の UOB はオブジェクトをその状態のまま維持していることを示している。オブジェクト状態に付けられた「シナリオ」種類の参照指示も、これと同じである。詳細については、「オブジェクト状態に付けられた参照指示」という題名の小節を参照のこと。

参照指示の種類が「シナリオ」の場合も、参照指示のラベルはシナリオのラベルと同じでなければならない。この種類の参照指示がプロセススキーム図で使用されている場合、プロセスフローにおいて、参照先のシナリオ活動が、次に発生することを示している。また、この種類の参照指示がオブジェクトスキーム図上の変換線に付けられている場合には、参照先のシナリオの活動は、状態変移が許可される前に開始されなければならない（UOBの参照指示を参照）。

参照指示の種類が「TS（Trasition Schematic、変移スキーム図）」である場合もまた、参照指示のラベルは変移スキーム図のラベルと同じにしなければならない。この種類の参照指示がプロセススキーム図上で使用されている場合には、参照指示の UOB への取付は、単純な接続リンクを用いなければならない（順序リンクは使用不可である）。また、この種類の参照指示がオブジェクトスキーム図上で使用されている場合には、状態（state）間に張られた変換線（transition arc）上のどこかの点に取り付けなければならなず、オブジェクトやオブジェクト状態に取付けてはならない（下図）。状態間に張られた変換線に取り付けられた呼出続行 TS 参照指示は、状態変移が許可される前に、参照先の変移スキーム図の複数の状態を経由する変換を開始しなければならない（オブジェクトスキーム図について説明されている小節を参照のこと）。

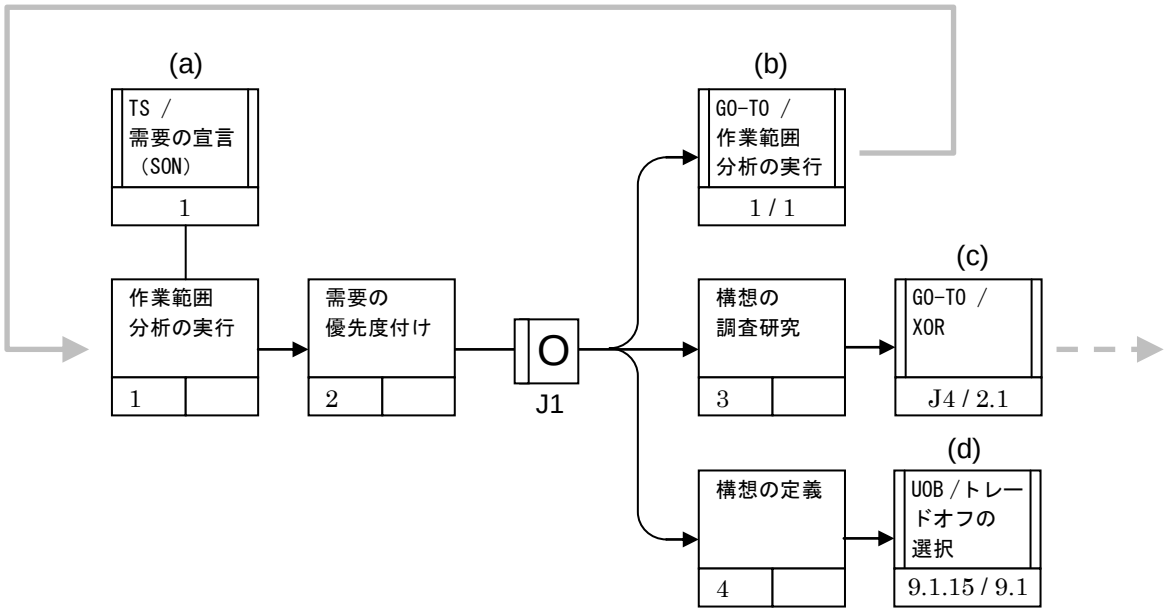


訳注：「図 3-41　基本状態変移スキーム図上の UOB 参照指示」を引用、追加



訳注：「図 3-46　オブジェクトをある状態に保ち続けている構成」を引用

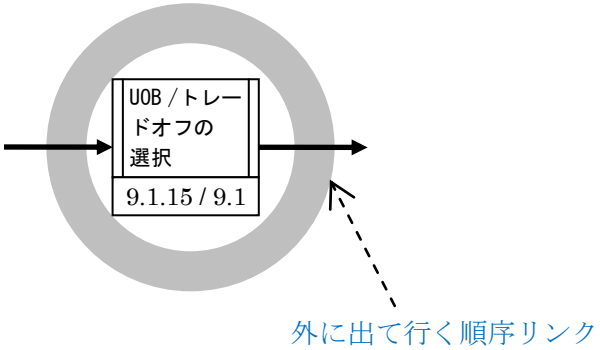
参照指示の種類が「GO-TO」で、UOB を参照している場合には、プロセスの次のイベントは参照先の UOB の発生となる。この種類の参照指示は、プロセス内のドキュメントループ（？document loop、調べたが意味不明）に使用されることが多い。また、参照指示の種類が「GO-TO」で、交差点を参照している場合には、プロセスの次のイベントは、参照先の交差点の次にある UOB の発生である。GO-TO 参照指示は、常に呼出続行参照指示となる。



訳注：　図 3-35　GO-TO 参照指示のあるプロセススキーム図

３．２．８．２　呼出待機参照指示（Call-and-Wait Referents）

参照指示の種類が「UOB」か「シナリオ」の場合には、外に出て行く（outgoing）順序リンクを持っている（下図）。また呼出待機「GO-TO」参照指示は、成立しない。



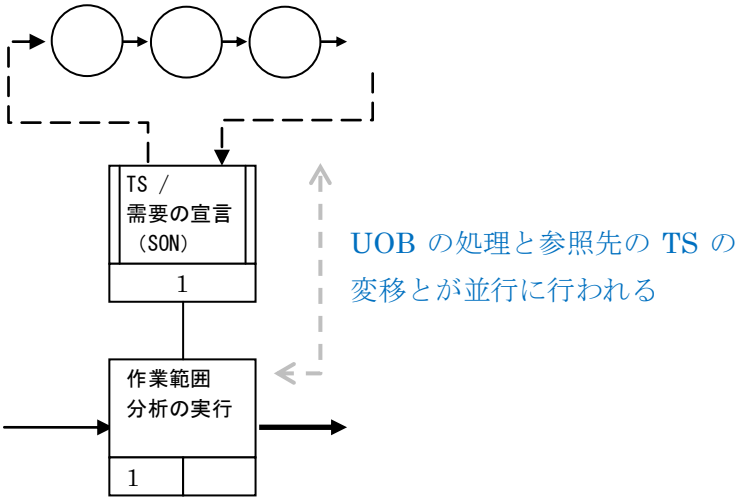
参照指示の種類が「UOB」の場合には、参照指示のラベルは UOB のラベルと同じにしなければならない。これは、前に定義された UOB の別のインスタンスが、（ループバックの無い）プロセス上の、ある特定の点で発生している事を意味している。

オブジェクトスキーム図上の変換線に取り付けられている場合には、状態変移が許可される前に、参照先の UOB の活動を開始、終了させなければならない（オブジェクトスキーム図について説明した小節を参照のこと）。この種類の参照指示がオブジェクトスキーム図上のオブジェクト状態に取り付けられている場合には、参照先の UOB が、その期間を通して、オブジェクトをその状態に保っているということを示している。更に、呼出待機参照指示を使用することにより、それに続くオブジェクト状態が、その UOB によって表現されているプロセスが完了する前に実体化してはならない、という制約を追加している。オブジェクト状態に取り付けられたシナリオ参照指示の場合も同様である。詳細については「オブジェクト状態に取り付けられた参照指示」という題名の小節を参照のこと。

参照指示の種類が「シナリオ」の場合には、参照指示のラベルはシナリオのラベルと同じ物にしなければならない。プロセススキーム図で「シナリオ」種類の参照指示が使用されている場合には、プロセスフローの次のイベントは、参照先のシナリオの活動の発生となる。つまり、その名前のシナリオに含まれる全ての分解物は、プロセスフローの次のイベントが発生する前に、開始され、かつ終了していなければならない。種類がシナリオである参照指示がオブジェクトスキーム図上の変換線に取り付けられている場合には、参照先のシナリオの活動は、状態変移が許可される前に完了していなければならない（オブジ

ェクトスキーム図について説明している小節を参照のこと)。

参照指示の種類が「TS（変移スキーム図)」の場合、ラベルは変移スキーム図のラベルと同じものにしなければならない。この種類の参照指示がプロセススキーム図で使用されている場合には、UOB への取付は、単純な接続リンクを用いなければならない（順序リンクは使用不可)。この種類の参照指示が使用されている場合には、参照先の変移スキーム図を通したオブジェクトの変移の終了時に、取り付けられた UOB も終了する、ということを意味している。



この種類の参照指示がオブジェクトスキーム図で使用されている場合には、状態間に張られた変換線上に取り付けなければならない（オブジェクトやオブジェクト状態へは取付けられない)。状態間に張られた変換線に取り付けられた呼出待機 TS 参照指示は、そのオブジェクトが、状態変移が許可される前に、参照先の変移スキーム図上の複数の状態を変移しなければならない、ということを意味している（オブジェクトスキーム図について説明している小節を参照のこと)。

３．２．８．３　IDEF3 プロセススキーム図での参照指示の使用

この節では、プロセススキーム図での参照指示の使い方について説明する。オブジェクトスキーム図での参照指示の使い方については、基本状態変移スキーム図の説明が終わった後で行う。このように段階を踏んで説明して行く理由は、同一のプロセスを基にして作られたプロセス中心視界とオブジェクト中心視界の、統合方法の説明を容易にする為である。

図 3－35（次ページ）は、立案プロセスに必要な手順を説明したプロセススキーム図である。プロセススキーム図に書かれた(a)～(d)の参照指示は、以下のような意味合いを持っている：

（a）この参照指示は、「需要の優先度付け（Prioritize Needs)」UOB がプロセスの活動内で開始可能となる前に、TS（変移スキーム図番号 1）を横断（traverse）しておかなければならない、ということを示している。この構成は、作業範囲分析（Mission Area Analysis）の作業中に実現されていなければならない状態のセットが、変移スキーム図「需要の宣言（SON、Statement of Needs)」によって横断されている、という観念（notion）を反映している。

（b）GO-TO 参照指示を使用することで、「作業範囲分析の実行（Perform Mission Area Analysis)」UOB へのループバックの可能性を表わしている。

（c）交差点参照指示であり、「構想の調査研究（Explore Concept)」UOB の後、プロセスが分解 2.1 の交差点 J4 へと移動する、ということを示している。

（d）この参照指示は、他の場所にある UOB 箱によって表わされている状況が、ここに複製されていることを表わしている。この例では、「構想の定義（Define Concept)」UOB を横断しているプロセスパスのインスタンスが、「トレードオフの選択（Perform Alternative Trade-offs)」UOB（分解 9.1 内にある番号 15 の UOB）で発生しているプロセスを複製したものに、追従された状態にある。

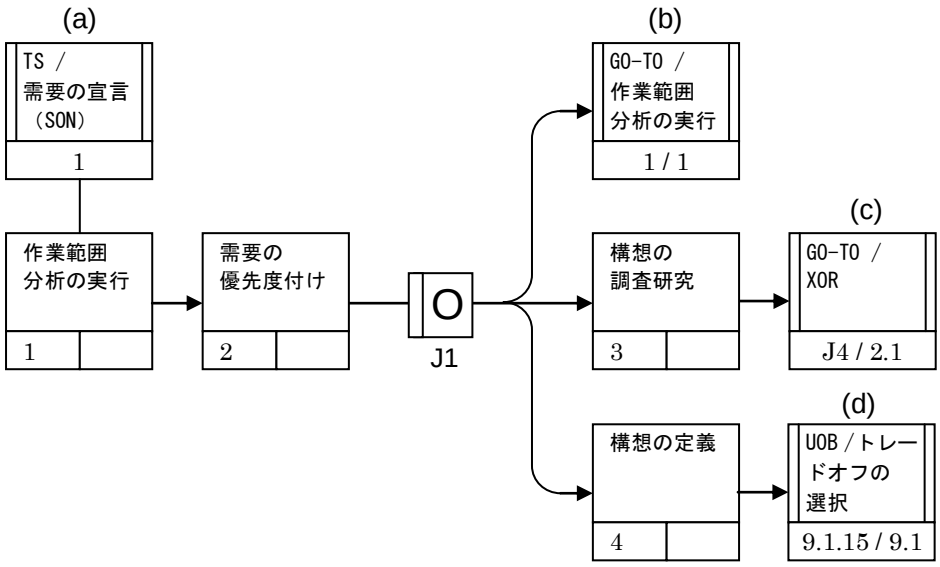


図 3-35　GO-TO 参照指示のあるプロセススキーム図

３．３　オブジェクトスキーム図（Object Schematics）

この小節では、オブジェクト中心プロセスの詳細な情報、つまりは、様々な種類のオブジェクトが、プロセスを通して、どのように他の種類のオブジェクトへと変化して行くのかという情報や、対象となる種類のオブジェクトが、プロセスを通して、どのように状態を変化させて行くのかという情報を、どのように表現して行くのか、といった事に関して説明して行く。例えば、自動車製造においては、ドライブトレイン（drive train、駆動系）とシャシー（車台)、車体とを組立てると自動車になるということが、また加熱プロセスにおいては、水の性質が冷凍状態から冷水、温水、熱水、沸騰へと状態を変化させることが、オブジェクト中心プロセスである。その為、オブジェクト中心表現では、物体の種類だけでなく、多様な状態にある物体の種類についても獲得する必要がある。

３．３．１　オブジェクトとオブジェクト状態（Object and Object States）

例えば自動車のシャシーのような、ある特定の「種類（kind)」のオブジェクトは、図 3-36 に示されたような、適切なラベルを持つ円によって簡単に表現される。この円は「種類」シンボルと呼ばれている。

図 3-36　種類シンボル

ある種類のオブジェクトが、ある状態にあることを表現するには、オブジェクトの種類とそれに対応した状態か、もしくは（対象のプロセス内で）その状態にあるオブジェクトの形式（type）もしくはクラス（class）による表現を記したラベルを持つ円によって表現される。例えば、氷のラベルは「水：凍結」、冷水は「水：冷たい」というようになる。この新しい構成は、「オブジェクト状態」シンボルと呼ばれ、図 3-37 のようになる。

図 3-37　オブジェクト状態シンボル

種類シンボルとオブジェクト状態シンボルによる複雑な表現構成は、「オブジェクトスキーム図（object schematics）」と呼ばれている。この節では、こうした表現構成の文法と意味論（意味論）の説明をするまでに留めて置く。この表現構成と、より一般的な存在論（オントロジー、ontology）の背景についての詳細な説明については、IDEF5 手法の報告書（KBSI 1994）を参照して欲しい。

３．３．２　変移スキーム図（Transition Schematics）

最初に説明する最も基本的な構成は、図 3-38 で示される「基本状態変移スキーム図（basic state transition schematics）（もしくは単純に、変移スキーム図）」である。

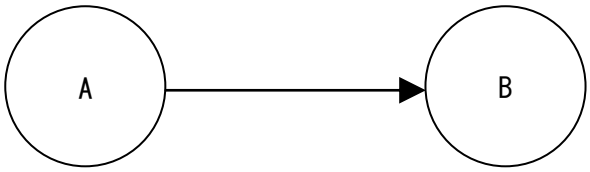


図 3-38　基本状態変移スキーム図

直感的に、基本状態変移スキーム図は、イベント（event）の特定のパターンや、発生し得る状態の特定の形式（type）、すなわち、対象となる状態 A であるオブジェクト a が、状態 B にあるオブジェクト b によって追従されているという状態、を表現したものである。

この場合、一般的にオブジェクト a は、ある時間の中で、改良・変換されて b になるか、もしくは b を生産する為に消費される。この a と b は、個体状態から液体状態へと変移する水であったり、未塗装状態から塗装状態へと変化する自動車の車体のように、通常は同一のオブジェクトであることが多い。しかし、常に a と b が同一のオブジェクトであるとは限らず、例えば燃焼プロセスでは、木片が消費されて灰の山が生産されるという状態変移が発生している。その結果として、一般的に基本変移スキーム図のデフォルトの意味論は弱く、2 通りの解釈（reading）が出来てしまう。その為、同一のオブジェクトが状態のみ変移させているという解釈を、より強く、明確に表現するには、図 3-39 のような矢が 2 つ付いた矢印を用いている。

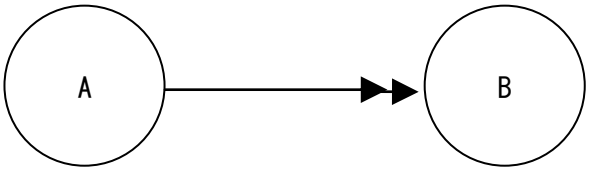


図 3-39　強変移リンクを持つ、基本状態変移スキーム図

A から B へと変移するだけであれば、一般的にはオブジェクトは同一のままで、状態 A にあるオブジェクトが A であることを止め、その後で B へと変移する。水が個体から液体へと変移したり、車体が未塗装から塗装済へと変移するといった例が、これに当る。
しかし、常にこうでなければならないと言うわけでもない。例えば、1 人が居る（1 人以上居る）部屋が、10 人が居る（10 人以上居る）部屋へと変移する可能性がある。また 10 人以上居る部屋が、 1 人以上が居る部屋へと変移する可能性も有る（訳注：意味不明。オブジェクト自体が全く変化せずに、オブジェクトの状態のみが変化する例なのか？）。

基本状態変移スキーム図が示す一般的な意味論は：
・指示された変移が発生した際に、
・先に状態 A にあるオブジェクト a が存在しており、
・「続いて」状態 B へと変化して行くことになるオブジェクト b が存在する、
というものである。

つまり、「オブジェクト b が状態 B へと変化する前に、オブジェクト a が状態 A であること」が要求されているのである。

この時、あまり一般的では無いものの、
・状態 A にあるオブジェクトと、状態 B へと変化してゆくオブジェクトとが
別個のものであること
・オブジェクト b が状態 B に変化した後でも、オブジェクト a が状態 A のままで
あること
が、それぞれ「許容」されていることになる。

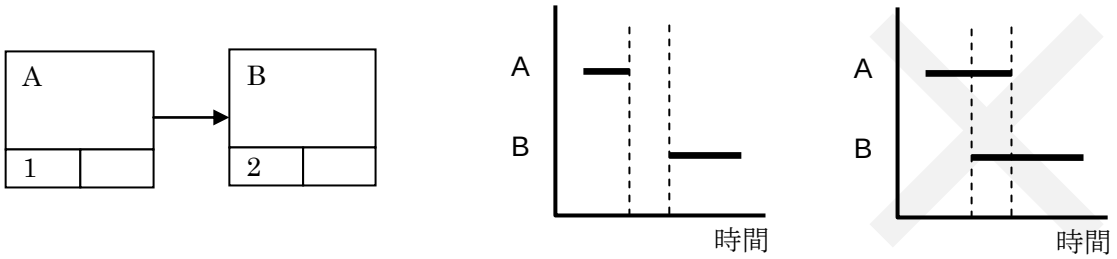
オブジェクトスキーム図上の中塗り三角矢印（solid-tipped arrow）の意味論は、見た目は殆ど同じではあるが、プロセススキーム図上の矢印の意味論とは異なっているということに気を付けなければならない。

プロセススキーム図では、矢印は全ての時間における順序を意味している。矢印の尾部側にある UOB のインスタンスは、矢印の頭部側にある UOB のインスタンスが開始されるまでに、終了していなければならないのである（次ページ訳注図上）。

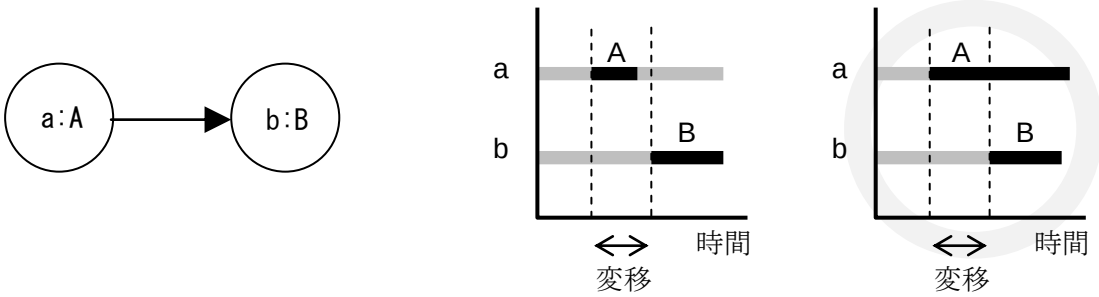
一方、オブジェクトスキーム図では、矢印は始点のみを考慮した順序を意味している。矢印の尾部側（始点）の状態にあるオブジェクトは、矢印の頭部側（終点）の状態にあるオブジェクトへと変移する前に、その状態であることを開始しておかなければならない。（下の訳注図下）

このように状態変移スキーム図上で、順序関係が希薄になっている理由は、前に説明した通りである。変移は、ある状態にあるオブジェクトが、別の状態にあるオブジェクト（このオブジェクトが変化前のオブジェクトと同一物かどうかは関係ない）へと変化することのみに関与しているのである。つまり、一般的ではないが、変移の最初の状態にあるオブジェクトは、変移の後になっても、その状態であることを終了させる必要が無いのである。この変移形式を許容する為に、オブジェクト変移スキーム図上の矢印の意義が、弱められているのである。

訳注：分かり難いので、以下に図示する



プロセススキーム図上での矢印の制約



オブジェクトスキーム図上での矢印の制約

３．３．３　条件（Conditions）

対象となる種類の（もしくは対象となる状態にある）オブジェクトの特徴と、そのオブジェクトの別の種類への変化の仕方（どのうようにしてその状態へと変移するか、その状態から変移するか、等）を統制する条件（condition）もしくはルールとを、区別する事が重要である。（この節での説明の中心は状態であることから、今後、種類への明確な参照は省略する）

IDEF3 ではこの条件を、進入（entry）、変移（transition）、状態（state）、そして退出（exit）の、4 つの一般的なクラスへと分類している。この内、状態条件と退出条件は、内因的（intrinsically、本質的）に状態と関係し、進入条件と変移条件は、状態と変移リンクとの間のインターフェースに関係しており、それらは図 3-40 のようになる。従って、状態条件と退出条件はオブジェクト状態の詳細説明（elaboration）に、そして進入条件と変移条件は変移リンクの詳細説明に、それぞれ付加されている。（図 3-40 は、4 つの状態がオブジェクトスキーム図のどの点に適用されている物かを説明する為に描かれたものであり、これらはオブジェクトスキーム図の新しい図形要素というわけではない）

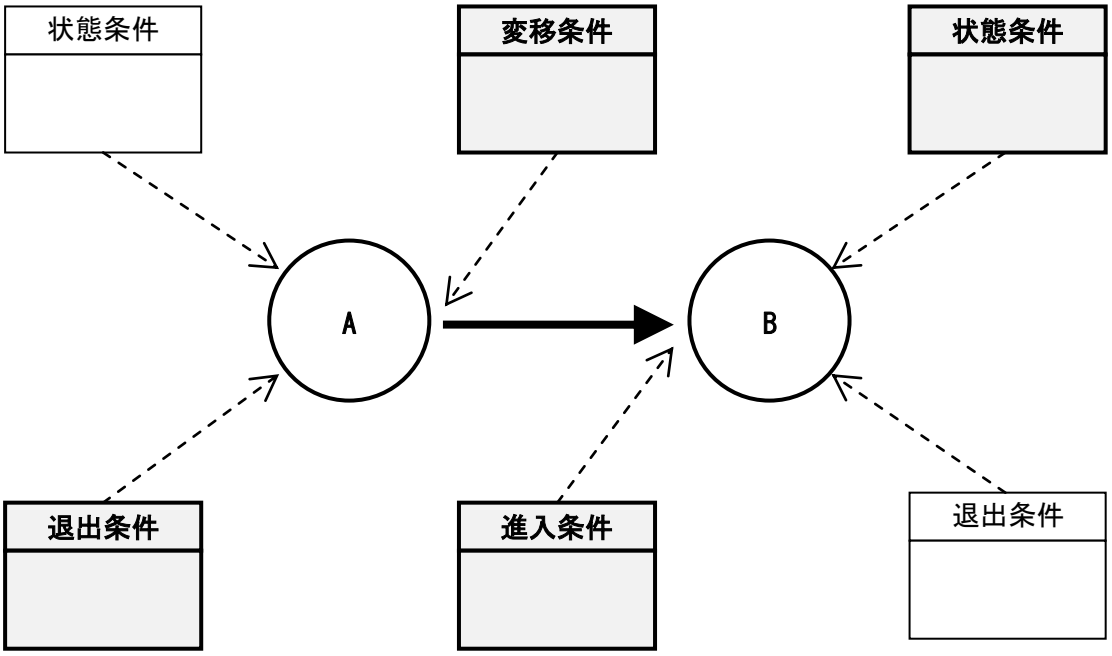


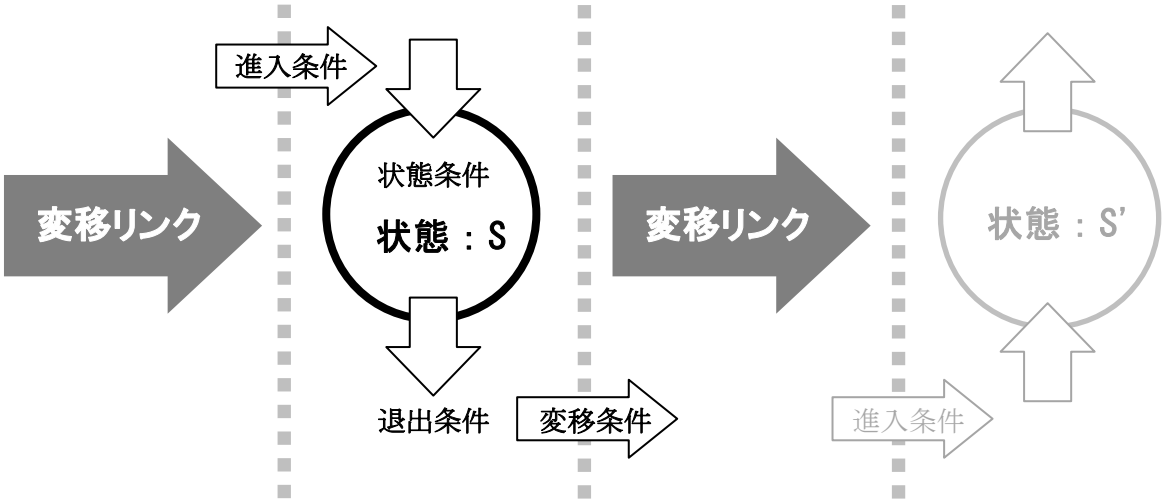
図 3-40　オブジェクトスキーム図の条件

状態条件（state condition）は、オブジェクトが、対象となる状態である為に必要となる条件である。凍結状態（平均海面）で例えるなら、水は 0℃以下という条件を満たさなければならない（塩分がある一定の濃度よりも低い等といった他の条件もまた、水が凍結する条件として満たす必要があるが）。注意しなければならないのは、この情報は、水がどのようにして冷凍状態へと移行するかということとは、独立しているいうことである。

退出条件（exit condition）は単純に、オブジェクトが、対象となる状態であることを終了し始める（その状態から脱し始める）条件のことを示している。例えば、温度が 0℃以上になると、水は凍結状態であることを終了する。どの状態かどうかとは関係なく（例えあるとしても）、対象となる S 状態にあるオブジェクトは、状態 S の退出条件が満たされることにより変移を開始する。これが、退出条件と、次に述べる変移条件との本質的な違いになっている。

変移条件（transition condition）は、状態と、そこから出発しているリンクとの間の「インターフェース」（訳注：つまりは変移リンク）に対して適用される。そしてこの変移条件は、対象となる状態（図 3-40 の「A」）にあるオブジェクトが、関係したリンクの終着点（destination）の状態にある（恐らく別の）オブジェクト（図 3-40 の「B」）へと変移する為の、独立した必要条件もしくは連帯した必要条件により構成されている。

進入条件（entry condition）は、ある状態と、その状態へと入力されるリンクとの間のインターフェースに対して提供されている。変移リンクの始点にある（恐らくは別の）オブジェクトは、変移条件を満たすことで変移を開始するが、このオブジェクトが次の状態へと進入する為に満たすべき条件によって構成されている。



訳注：　4 種類の条件？

オブジェクトスキーム上の、ある状態から別の状態への変移がどのようなものであっても、上で説明した 4 種類の条件を明確にしたり、識別する必要は無い。例えば、相対的に単純なスキーム図において、状態のラベルと UOB のシンボルから、意味論は明白であるからである（？）。

３．３．４　IDEF3 オブジェクトスキーム図での参照指示（referent）の使用

プロセススキーム図と同じく、ある状態変移に関するより細かな詳細は、オブジェク状態の詳細説明（elaboration）の中で行われている。しかし、変移線（arcs）に参照指示（referent）を取付ける事で、状態変移についての便利な情報を、それに一致するスキーム図へと明確に付加する事が可能である。

（訳注：この資料では、ここに至るまで、詳細説明（elaboration）についての説明が無い。詳細説明については、この後にある「３．４　詳細説明」を参照のこと）

３．３．４．１　変移リンクに取り付けられた参照指示
（Referents Attached to Transition Links）

オブジェクトスキーム図の参照指示は、UOB スキーム図、シナリオスキーム図、変移スキーム図の、いずれかに対して参照させることが可能である。

UOB 参照指示もしくはシナリオ参照指示である場合、その参照指示は直観的に、指示した変移が発生している間に実行されるプロセスか、もしくは最低でもその変移に関係したプロセスである事を表わしている。

一方で、参照指示が変移スキーム図である場合（下図 3-44、状態 C から D への変移スキーム図）、この変移（C→D）が、参照先の変移（図 3-41、状態 A から B への変移スキーム図で表される変移）を中間状態（intermediate state）として、行われているということを示している。

最も一般的な場合における文法は、図 3-41 のような、基本変移スキーム図上の 1 本の変移線に、1 つの参照指示が取り付けられたものである。

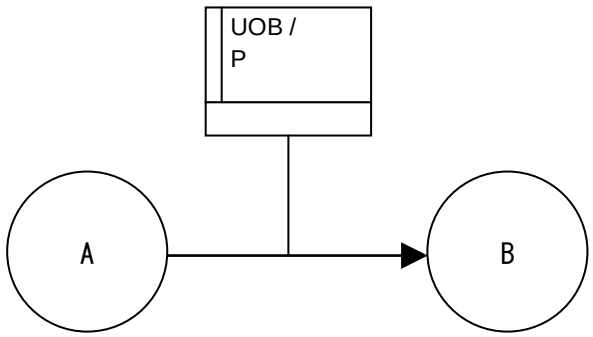


図 3-41　基本状態変移スキーム図上の UOB 参照指示

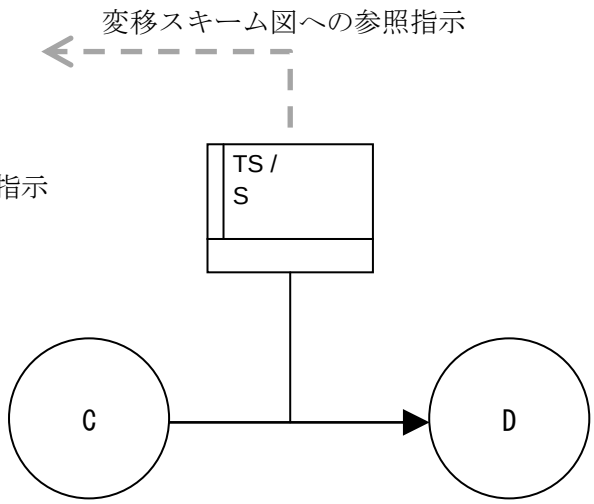


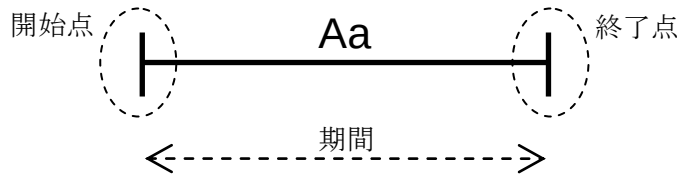
図 3-44　呼出続行参照のある、変移スキーム図

（訳注：原文ではここに無いが、上の説明文を理解するには必要なので、ここに持って来て図 3-41 と組み合わせた）

一般的に、(図 3-41 の) P は、指示した側の変移が発生している間のプロセスとなる。その為、指示されたプロセスが発生すると、参照先プロセス P のインスタンス p が発生した時点で、変移側には既に状態 A であるオブジェクト a が存在しており、また P が開始された後のある時点において、オブジェクト b が存在していることになる。しかし、これまでに説明したように、図 3-41 の参照指示は、状態 A から B への変移に、そのプロセスが関係しているということのみを表わしているのである。つまり、図 3-41 の標準的な意味論として要求されている事は「参照指示の付いた変移の発生において、P のインスタンスの開始時もしくは開始前のある時点で、状態 A のオブジェクトが存在しなければならない」という 1 点だけなのである。

この変移スキーム図の意味論は、図 3-42 のような「期間ダイヤグラム（interval diagram)」という形で表現する事が可能である。このダイヤグラムは、オブジェクトスキーム図によって表現されるイベントパターンのインスタンスで発生し得る、様々な状況の間の時間的関係性を説明している。

期間ダイヤグラムの横線は、対象となる UOB もしくはシナリオが発生している期間か、もしくは特定のオブジェクトが対象の状態にある期間を表わしている。期間ダイヤグラムの縦線は、期間の開始点と終了点を示している。「Aa」は「状態 A にあるオブジェクト a」を、「Bb」は「状態 B にあるオブジェクト b」を、それぞれ短縮したものである。



このようなダイヤグラムは、基本状態変移スキーム図であったとしても、複数の「具体化パターン（instantiation pattern、インスタンスパターンでも良いのか？)」、つまり現実世界のイベントをスキーム図のインスタンスとして見なすことを可能にする複数の並行な行き方を許容していることから、便利である。図 3-42 の（1）から（4）の期間ダイヤグラムは、いずれも図 3-41 のスキーム図の、文法的に正しい（legitimate）具体化パターンを図示したものである。

（1）P のインスタンス p が始まる前に状態 A の a が存在しており、そして変移によって発生したオブジェクト b が、P の終了後も、状態 B のまま在り続けている。

（2）オブジェクト a の状態 A から B への変移が、瞬間的（最小時間単位に近い）に行われている。

（3）ここでは 2 つの重要な可能性が示されている。1 つ目は、オブジェクト a が状態 A となったと同時か、それ以降にインスタンス p が開始可能となる事であり、2 つ目は、オブジェクト a が状態 A であることを終了する前に、オブジェクト b は状態 B となる事である。このような場合は、一般的に a と b は別のオブジェクトであり、そしてインスタンス p の間にオブジェクト a が状態 A になる事が、オブジェクト b が状態 B となることの必須条件となる。例：回路（オブジェクト a）に電源が入る（状態 A）ことが、警告ライト（オブジェクト b）が作動する（状態 B）ための必須条件。

（4）インスタンス p が開始される前に、オブジェクト a は状態 A を終了し、インスタンス p が終了した後に、オブジェクト a が状態 B となる。

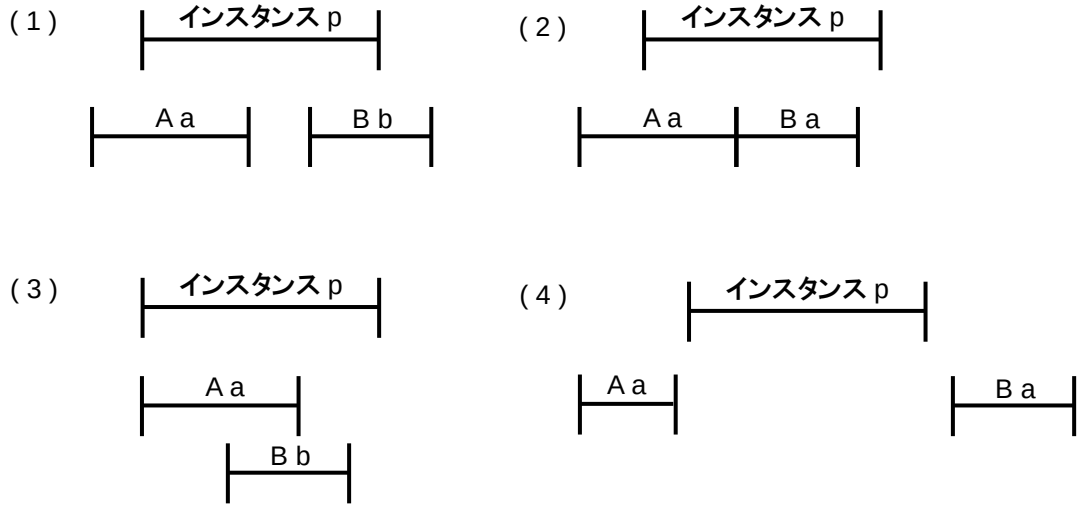


図 3-42：　図 3-41 のインスタンスを表現した、期間ダイヤグラム

基本変移スキーム図上の参照指示は、一般的に、指示された変移の発生の切っ掛けとなるプロセスを示したものであることから、（4）の期間ダイヤグラムの構造は、不適当なもののように見える。しかし、もう一度、図 3-41 の参照指示を見返してみると、変移に関係するプロセスのみを指示する必要はあっても、変移プロセス全体を指示する必要な無いのである。例えば、A が凍った水、B が気化した水、P が加熱プロセス（加熱要素と関係している）であるとする。この時、指示されたこのプロセスの中で、氷の塊が自然に融解して水になるのを待ち、それから水を P により沸騰する直前まで加熱し、沸騰寸前の熱湯が蒸発によって自然に気化するのを待つ、という場合も考えられるのである。

この弱められた意味論（weaker semantics）の重要な点は、IDEF3 というものは、そもそもがプロセス（と状態変移）の記述獲得手法である、ということである。ある変移を記述している時に、人は変移プロセス全体の事を知っているわけではなく、変移内の幾つかの中間プロセスしか知らないのである。この意味論は、これを考慮しているのである。
（訳注：IDEF3 にとって、知識を収集する事が第一の目的であり、収集した知識の辻褄が合っているかどうか、収集した知識が正しいかどうかは後で考えれば良く、それよりも収集中にあれこれ考えている内に知識を取りこぼしする方が困る、というスタンス）

対象となる表現の意味論で、何がルール外となっているかを理解する事は、何が許容されているのかを理解する事と同じくらいに重要である。本質的に、対象となるイベントの流れの中から除外されている唯一の事は、その構成状態の開始点の順序に関するものである。例えば、下に挙げる図 3-43 の 2 つの期間ダイヤグラムは、図 3-42 の正当な具体化パターンには含まれない。（5）では、P のインスタンス p が開始される前に、b が状態 B になっている。また（6）では、インスタンス p が、オブジェクト a が状態 A となる前に開始されている。

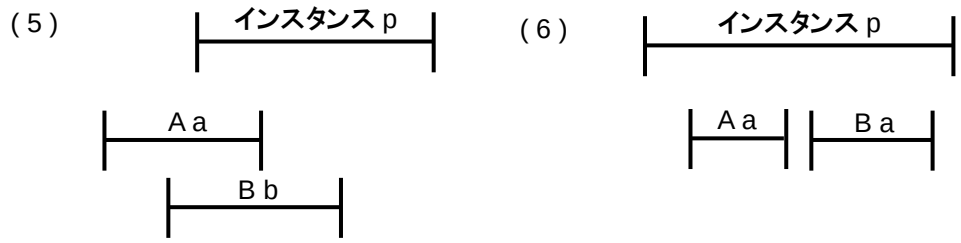


図 3-43　図 3-41 の意味論から除外されているパターン

この意味論では、参照指示が UOB かシナリオについては考慮されていない。全てのシナリオは、UOB のより詳細な（finer-grained）分解物として考慮する事が可能であることから、区別を付けない事は論理的である。このことは TS（変移スキーム図）の参照指示の場合でも、似たようなものになっている。図 3-44 のスキーム図の参照指示は、図 3-41 のスキーム図を参照している。

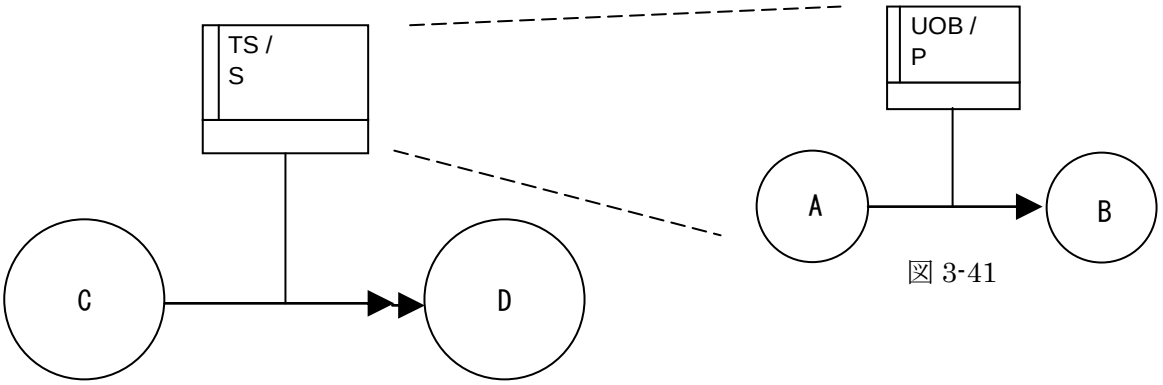


図 3-41

図 3-44　呼出続行参照のある、変移スキーム図

このスキーム図は、次のようなことを表わしている：

参照指示の取付けられた変移のインスタンスにおいて、まずオブジェクト c が状態 C である。そしてその時点で、図 3-41 のインスタンスが開始され、それによってオブジェクト a がプロセス P のインスタンスを経由した状態 B への変移を開始する。そして、この状態 A から状態 B への変移が開始された後のある時点において、c は状態 D へと変移する。

対象となる場合において、開始された変移が何を意味するのかについて、分析者は正確に判断しなければならない。

ここで重要となるは、以下の 2 点である：

（1）TS（変移スキーム図）参照指示が指示する一連の変移は、少なくとも（？in some sense)、C から D への変移が完了してしまう前に、開始されていなければならない

（2）C から D への変移は、参照指示の指示する一連の変移が完了したかどうかに関係無く、発生する事が可能である。

呼出待機参照指示を使用する事で、状態変移に関係するイベントの時間的順序情報を追加する事が可能である。呼出待機参照指示のシンボルは、図 3-45 のように、呼出続行参照指示の名前の右側に 2 本目の縦線が追加されたものになっている。

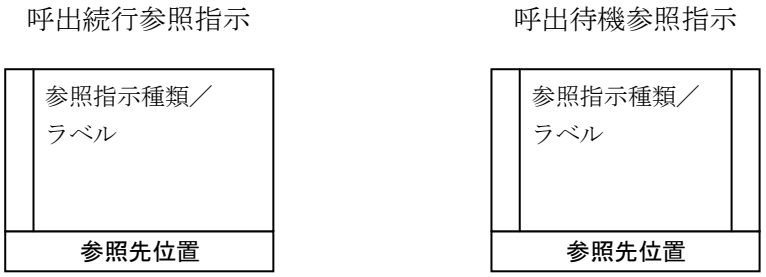
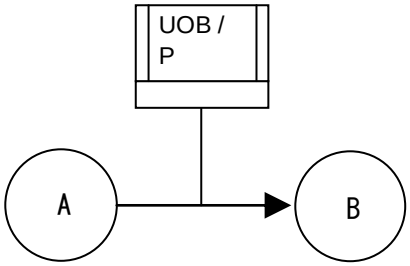


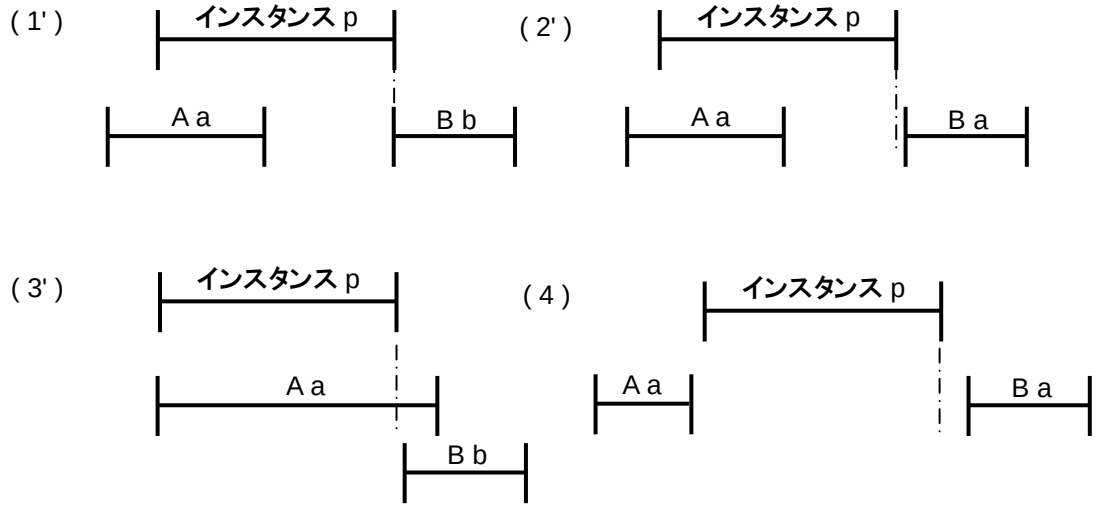
図 3-45　呼出待機参照指示のシンボル文法（右）

呼出待機参照指示は、「呼び出された状況（situation）は、変移の次の状態が発生可能となる前に、終了しなければならない」ということを意味している。図 3-41 の場合で説明するならば（次ページ）、UOB「P」のインスタンス p は、状態 A にあるオブジェクトが状態 B へと変移可能となる前に、終了しなければならない、ということである。ここで注目すべき事は、「インスタンス p の終了が、対象となる変移の完了と同時であっても良い」ということである。

これは、呼出続行参照指示を用いた状態変移スキーム図には含まれていない事である。むしろ、参照指示によって指示されたプロセスは、変移が完了する前に開始さえしていれば良いのであり、変移が終わった後も延々と続けることが可能である（図 3-42 参照）。



訳注：図 3-41 の呼出待機 Ver



訳注：　呼出待機の場合の期間ダイヤグラム？

同様に、対象となる呼出待機参照指示が TS（変移スキーム図）参照指示である場合、指示された連続する変移は、スキーム図で表現されている変移が完了する前に、完了しなければならない。その為、もしも図 3-44 の TS 参照指示が呼出待機で、図 3-41 のスキーム図を参照している場合には、オブジェクト c が状態 C から D への変移を完了する前に、A から B への変移のインスタンスは完了していなければならない。

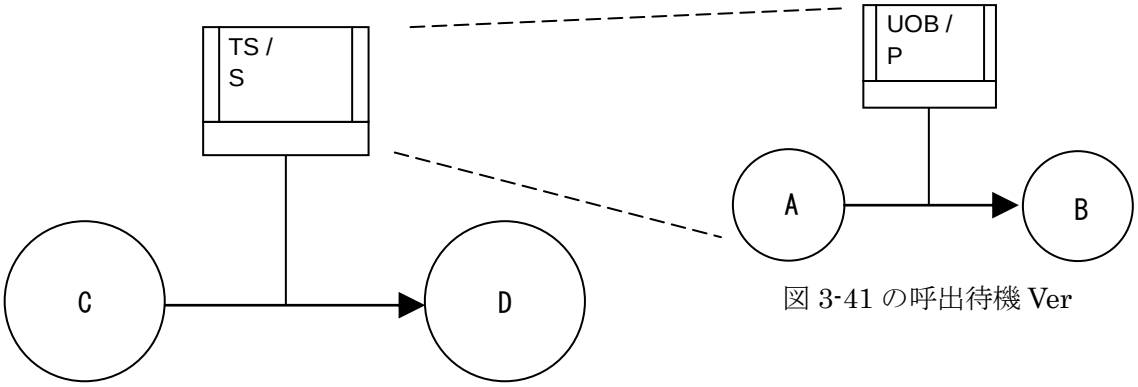
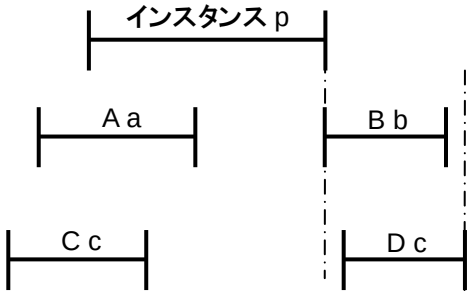


図 3-41 の呼出待機 Ver

訳注：　図 3-44 の呼出待機 Ver



訳注：　図 3-44 の呼出待機 Ver の期間ダイヤグラム？

３．３．４．２　オブジェクト状態に取り付けられた参照指示

対象となる状態にオブジェクトを「保つ」という状況は余り一般的なものでは無いが、例えば冷凍プロセスは、物質を個体の状態に保っている。こうした形式の状況は、図 3-46 のような構成で表現する事が可能である。

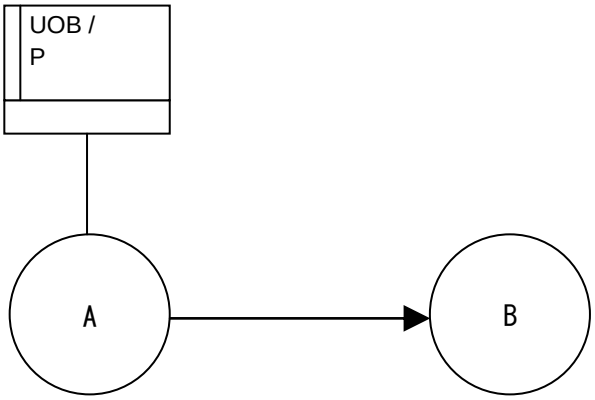


図 3-46　オブジェクトをある状態に保ち続けている構成

図 3-46 について、より一般的に説明するならば「UOB「P」のインスタンス p が存在し、そしてこの p が持続する間、オブジェクト a は状態 A である」となる。これを満足させるには、p が開始された時点に a が存在していなければならない。しかし、a は p が開始される前でも状態 A に成り得ることから、p の前に他の何らかのプロセスによって状態 A となり（実際には、物質が化学反応により個体になる、など)、その後、p によってその状態を保っている、ということも可能である。その為、図 3-46 の構成に対して、図 3-47 の示される 2 つのパターンが両立することになる。

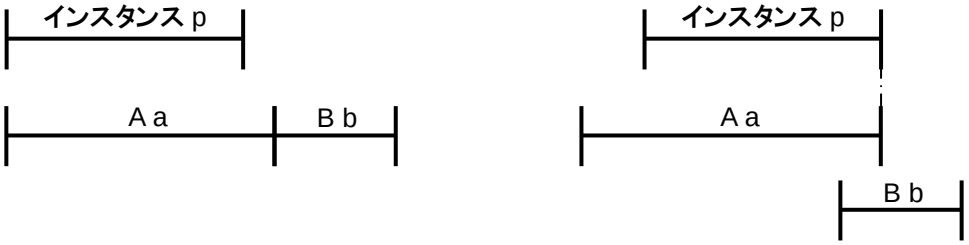


図 3-47　　図 3-46 のインスタンスのパターン

注意すべき事は、右側の期間ダイヤグラムでは、P のインスタンス p が終了する前に、オブジェクト b が状態 B になっているが、図 3-46 の呼出続行参照指示を呼出待機参照指示へと変更すると、これは除外されることになる。その場合には左側のダイヤグラムのみが、正しい具体化パターンを表わすことになる。

３．３．４．３　複数の参照指示を持つオブジェクトスキーム図

ある状態から別の状態への変移が、より複雑な複数のイベントを経由する場合、1 個の参照指示だけで表現できない事が多くある。そうした複雑なイベントの構成の詳細は、別々のプロセススキーム図によって表現すれば良いのかもしれないが、一連のイベントの並びは、1 つのオブジェクトスキーム図として表現する方が判り易くなる。そこで、1 本の変移線（arc）に複数の参照指示を取付けられるようになっている。

そうした複雑なスキーム図を上手く表現する為に、基本状態変移スキーム図での変移線を、指示された変移が発生している期間を表わす時間線に近い物（rough timeline）として考えるのである。つまり、ある状態変移スキーム図 D 上の変移線の、参照指示が取り付けられた位置は、D によって表現されている変移インスタンスでの、指示された UOB やシナリオ、連続した変移が開始される相対的な時間的順序（relative temporal order）を表現しているのである。例えば図 3-48 では、状態 A から B への変移において、まず UOB「P」のインスタンス p が先に発生し、次に UOB「Q」のインスタンス q が続くことになる。また、最初の参照指示は呼出待機であるので、インスタンス p はインスタンス q が開始される前に終了しなければならない。

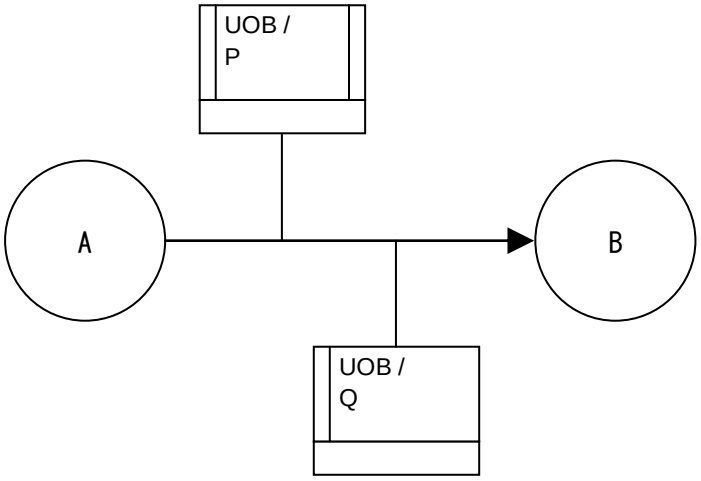


図 3-48　複数の、時間的順序の参照指示を持つオブジェクトスキーム図

図 3-49 は、2 つの UOB のインスタンス p と q が、同時に開始される変移を表わしたものである。左側の参照指示が呼出待機であることから、インスタンス p は状態 B への変移が完了する前に終了していなければならないが、右側の参照指示は呼出続行であることから、インスタンス q は、p のような拘束は受けない。

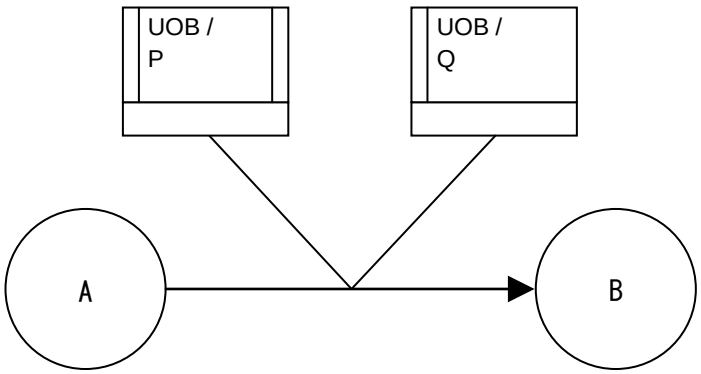


図 3-49　複数の、同時の参照指示を持つオブジェクトスキーム図

そして最後に図 3-50 は、時間的不確実性マーク（temporal indeterminacy marker：変移リンク上の小さな丸）を追加することで、変移に関係する複数の UOB 間の時間的順序が（判明している範囲で）定義されていない、ということを表わしている。P、Q、R のそれぞれのインスタンスは、指示された変移と関係している事は判っているが、それらの時間的順序関係は不定であり、どのような順序にも成り得るのである。そして時間的順序関係が定まっていないことから、通常の参照指示（呼出続行等）のみが使用される。呼出待機参照指示を使用しても、それは意味を持たない。

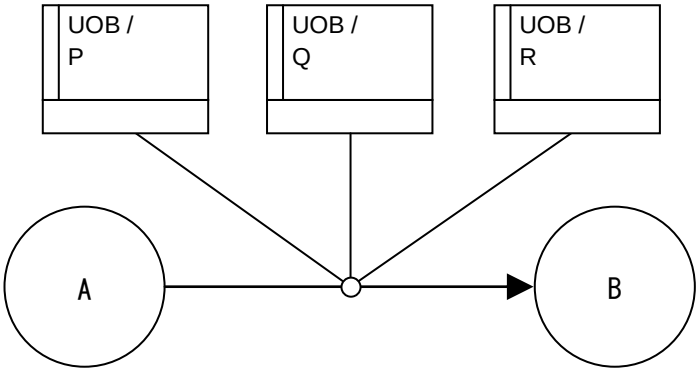


図 3-50　時間順序が未定義の参照指示を持つオブジェクトスキーム図

３．３．５　複合変移スキーム図（complex transition schematics）

オブジェクト中心視点から記述やモデリングを行おうとしているプロセスは、基本変移スキーム図（basic transition schematic）だけでは十分に獲得できないくらいに複雑である事が多い。そこで、複数のオブジェクト状態シンボルを持ったスキーム図等の、複合変移スキーム図（complex transition schematics）の構築が可能になっている。

複合変移スキーム図は、複合プロセススキーム図と、大まかな所で一致している。これにより、プロセスのオブジェクト中心視界の提供という変移スキーム図の中心的役割が支援されることになる。
（訳注：IDEF3 において非常に重要な事である。オブジェクトを処理するプロセスと、プロセスによって処理されるオブジェクトとは、当然に裏表の関係であり、同一のプロセスをプロセス視点、オブジェクト始点でフローを獲得すると、共有される部分が各所に現れる。その共有される部分を軸にして、プロセス中心視点とオブジェクト中心視点から獲得・分析を行う事で、対象となるプロセスを、より的確にとらえる事が可能となるのである）

またこの事から、プロセススキーム図と変移スキーム図は、構造的に相似するようにしておくべきである。ただし、変移スキーム図は、IDEF3 で構築可能なオブジェクトスキーム図全体の中のサブクラスの１つでしかないことも忘れてはならない。

まずは図 3-51 で示された、複合変移スキーム図について考えて行くことにする。

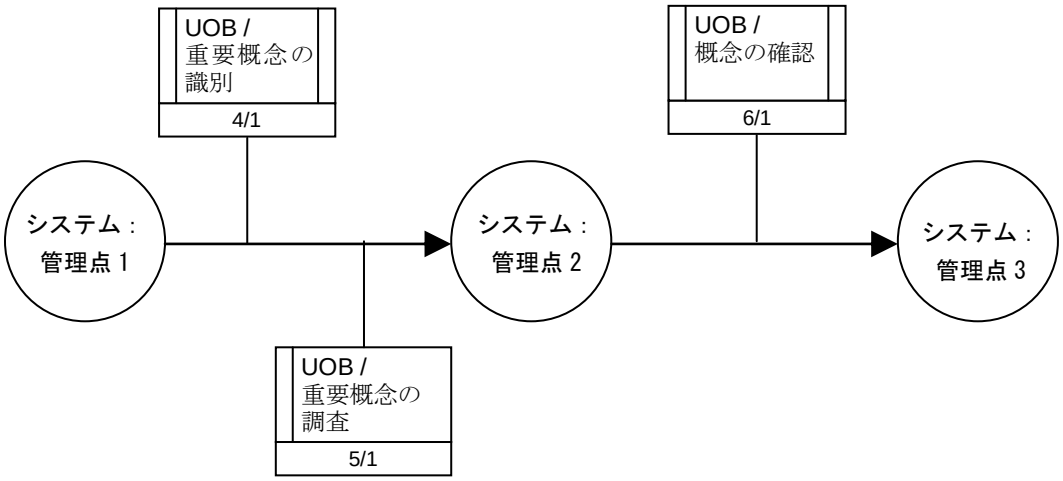


図 3-51　複合変移スキーム図

この変移スキーム図で記述されているプロセスは、あるシステム（図では単純に「システム」と呼ばれている）に関したものであり、「管理点 1（milestone1)」、「管理点 2」、「管理点 3」の 3 つの状態を経由して変移している。

1 番目の参照指示が呼出待機であることから、システムが管理点 1 から管理点 2 へと変移するには、参照指示で指示されている UOB「重要な概念の識別」が完了していなければならない。UOB「重要概念の調査」は、その後で開始されなければならないが、参照指示が呼出待機ではないので、管理点 2 へと変移する前に完了させる必要は無い。例えば、識別された重要概念の殆どについて調査が行われていれば、それで十分とすることも可能である。

UOB「概念の確認」は、管理点 2 への変移の後に開始しなければならず、そしてシステムが管理点 3 への変移に成功する前か、少なくとも成功した時に、この UOB「概念の確認」は終了していなければならない。ここで注意すべきは、単に変移線上の取付点の相対的な位置関係が重要なのであり、変移線上の取付点の距離は、それがゼロ（同一点）でない限り（同じ点上に参照指示が取り付けられている場合は、同時発生を意味する）無関係である、ということである。

プロセススキーム図と同様に、変移スキーム図も構造が全てであり、対象となる複合プロセスに関係した 1 つ以上のオブジェクトの、複数の状態変移構造（もしくは、状態変移の代表的なセットを最低 1 組）を、1 つの一般的な流れとして（in a general way）で記述している。一般的に、1 つの変移スキーム図が、状態変移の「連続物」もしくは「ネットワーク」を表現している為、変移スキーム図を情報損失無しに、より小さな部品へと分解する事は一般的に不可能である。例えば、図 3-51 のスキーム図を分割した図 3-52 の 2 つの基本変移スキーム図は、元のスキーム図と比べると情報量が劣っている。

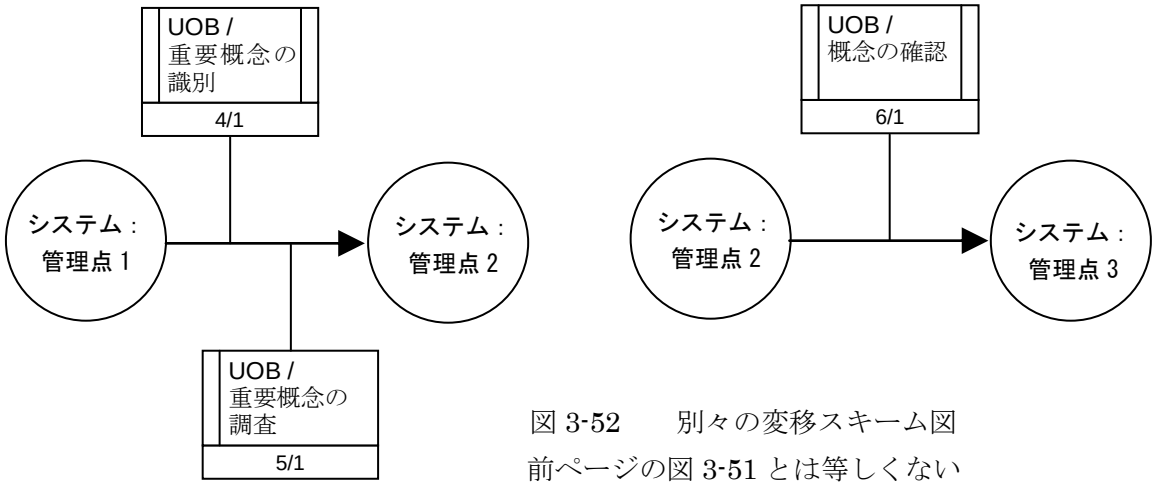
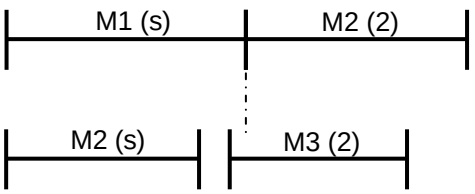


図 3-52　　別々の変移スキーム図
前ページの図 3-51 とは等しくない

これは、管理点 1 から 2 までと、管理点 2 から 3 までに、スキーム図を分割しただけのものである。このようにスキーム図を分割してしまうと、これらの 2 つの変移は必ずしもシステムの中で継続されることが無くなってしまうのである。何故なら、基本変移スキーム図の意味論には、管理点 1 と管理点 2 は互いに排他的な状態である（2 つの管理点が実際に排他的であるかどうかは無視して）という事が含まれていない為、この 2 つのスキーム図で表される特性は、図 3-53 のような管理点 1 から管理点 2 への変移の前に、管理点 2 と管理点 3 への変移が存在するという分割前には有り得ない状況も、含んでしまうのである。（訳注：結果として、2 つの基本変移スキーム図へと分割してしまうだけで、管理点 1、2、3 の時間的順序の制約情報が失われてしまう事になる）



M1(s)：システムが管理点 1 の状態にあるという意味

図 3-53　　図 3-52 の分割スキーム図で発生し得るインスタンスのパターン

３．３．６　変移交差点（Transition Junctions）

変移交差点を用いる事で、状態変移の挙動の中で、潜在的に複数のパスを持っているロジックを明確にすることができる。例えば図 3-54 は、論理和（inclusive disjunction、OR）交差点を使用することで、あるオブジェクトの状態 A が、幾つかの状態 Bn の中から選択された状態（複数可）へと変移する、ということを表わしている。

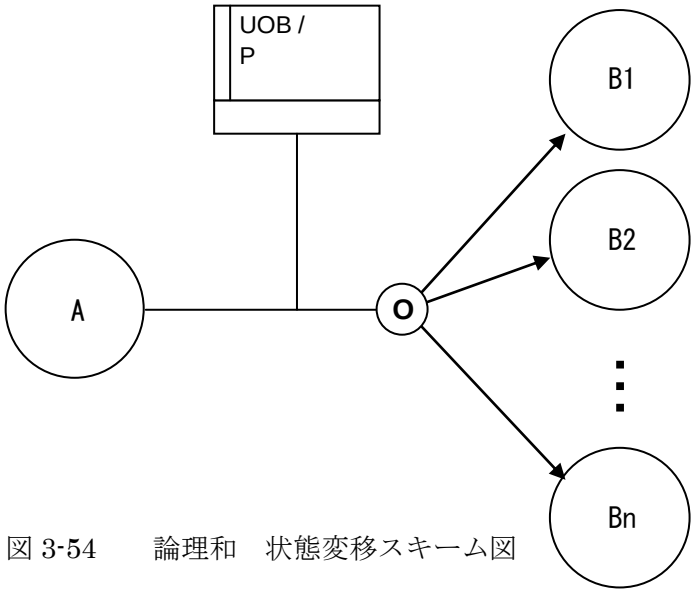


図 3-54　　論理和　状態変移スキーム図

論理和交差点ではなく、図 3-55 のように排他的論理和を指定する事で、変移先を Bn の中から 1 つの状態のみに絞ることができる。

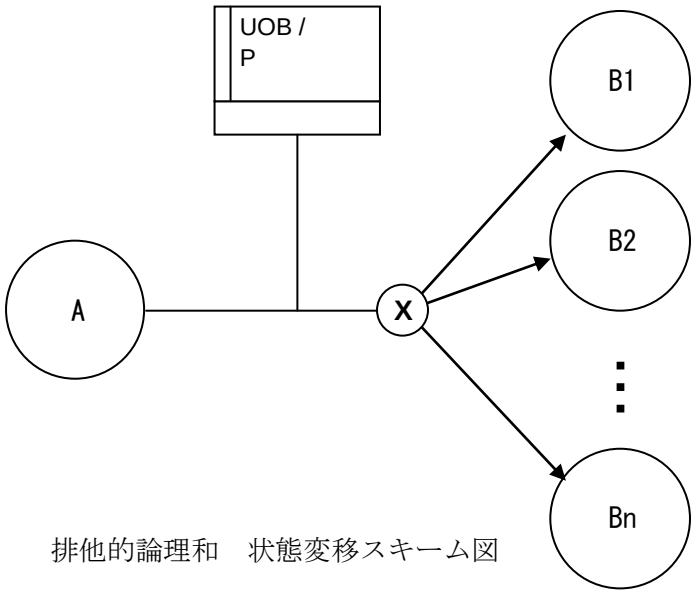


図 3-55　　排他的論理和　状態変移スキーム図

同様に、論理積交差点を用いると、図 3-56 のように継続する状態 Bn の全ての状態への変移を指示する事もできる。

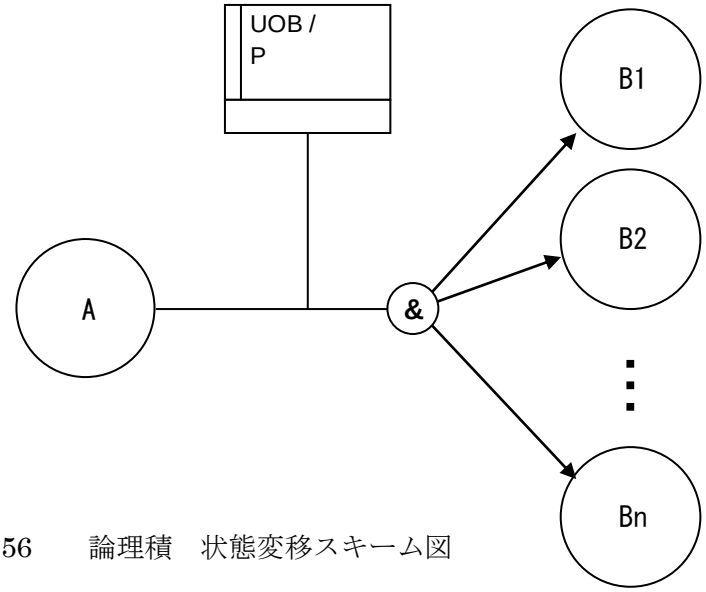


図 3-56　　論理積　状態変移スキーム図

論理的交差点を含むスキーム図の意味論は、基本変移スキーム図の意味論を総合化したものとなる。図 3-56 のスキーム図は、状態 A にあるオブジェクト a が、状態 B1 にあるオブジェクト b1、…　状態 Bn にあるオブジェクト bn へと変化する、プロセス P に関係する変移での状態の種類を表現したものである。基本変移スキーム図と同じように、オブジェクト a は、インスタンス p の開始の前か、もしくは少なくともそれと同時に存在していなければならず、そしてオブジェクト bi（i = 1 ... n）は、インスタンス p の開始後に状態 Bi となるか、もしくは状態 Bi への変移を開始ししなければならない。

オブジェクト bi の開始点と終了点の考慮され得る範囲は、図 3-57 の一般インスタンスパターン図上の、点線によって表現されている。

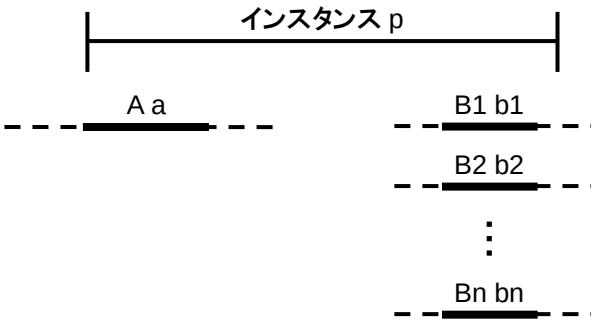


図 3-57　　図 3-56 の総合化された意味論

このような論理的スキーム図には「逆方向」のものもあり、それを表わしている図 3-58 のスキーム図上の「*」に、O、X、＆の何れかの記号を用いる事が可能である。

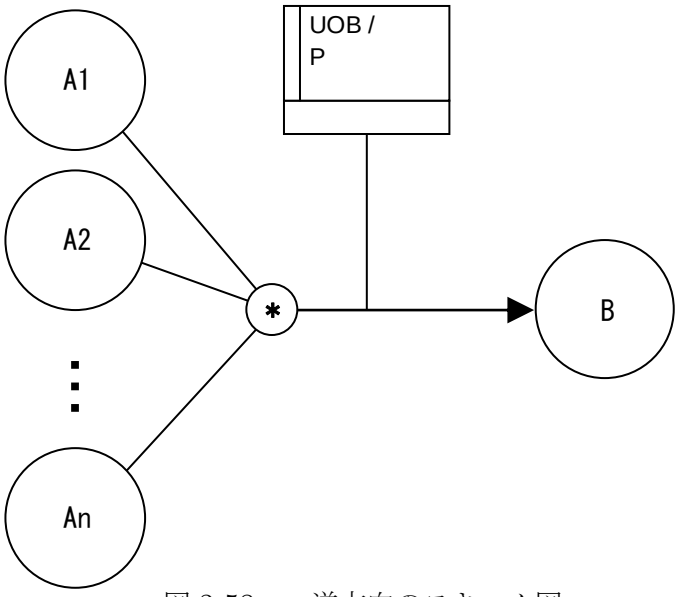


図 3-58　　逆方向のスキーム図

更に、包括的なプロセスの開始と終了の両端に複雑なロジックを持つ変移を表現する事も可能である。例えば、「状態 A1 と A2 であるオブジェクトが、プロセス P を通して、状態 B1 もしくは B2 のいずれかへと変移する」というものである。こうした変移の一般的な文法は図 3-59 のようになる。図中の「*」と「#」には、O、X、＆の、いずれかの記号が入る（両端が同じ記号でも良い)。

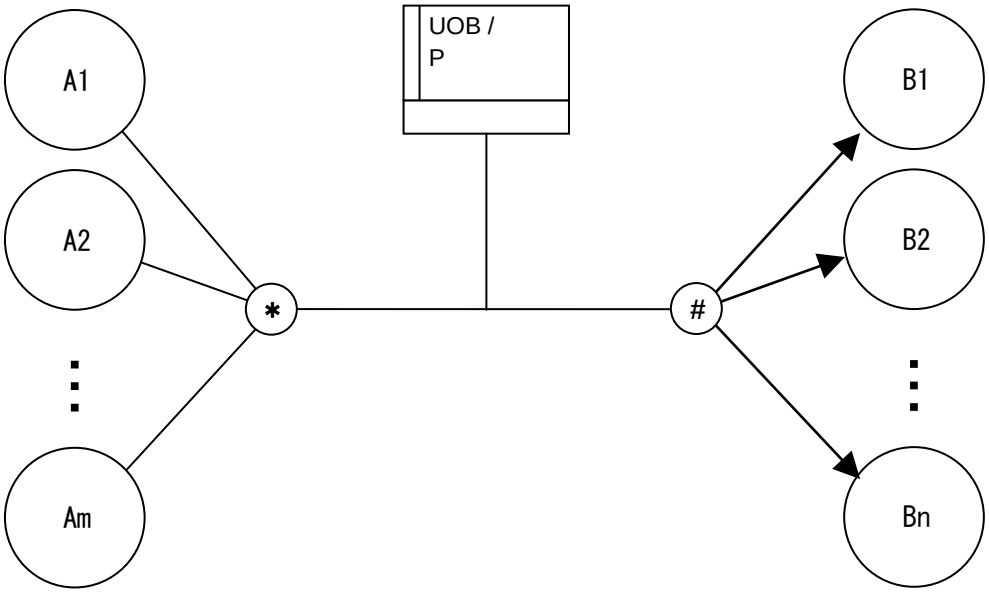


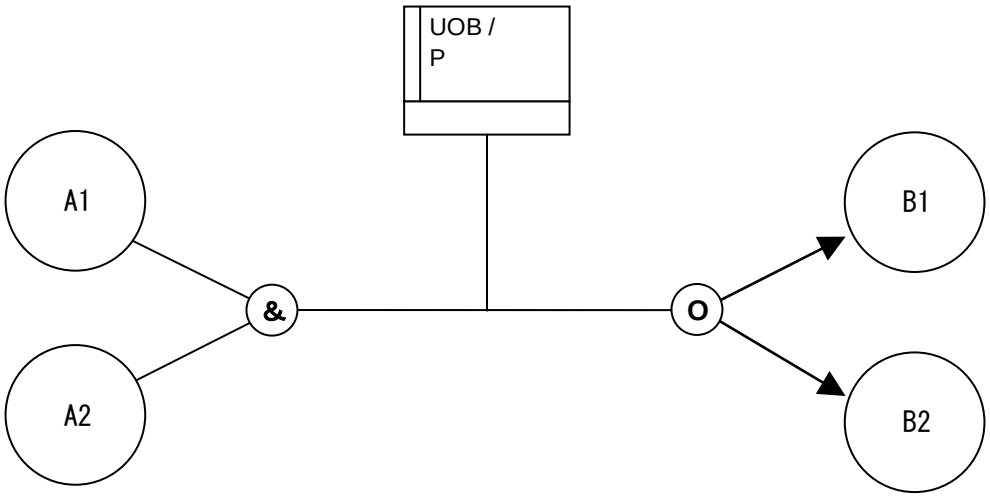
図 3-59　複数の交差点シンボルを用いた、複雑な変移ロジック

こうした種類のスキーム図は、一般的に不明瞭である。例えば、図 3-59 で、「*」が「&」、「#」が「O」の場合、n=m=2 とすると（下図)、インスタンス P によって状態 A1 と A2 であるオブジェクト a1 と a2 が：

（1）オブジェクト（a1 と a2）の両方が揃って状態 B1 もしくは B2 へと変移する

（2）オブジェクト a1 は状態 B1 に、オブジェクト a2 は状態 B2 に変移する

の、どちらの意味にも採ることができてしまうのである。そこで、このような不明瞭さは、リンクに詳細説明を加える事で解決することになる。



訳注：　図 3-59 の両端が＆と O、n=m=2 の場合

３．３．７　オブジェクト状態情報の遮蔽

スキーム図の構成やクラス階層と同様に、オブジェクトスキーム図上の情報を遮蔽することが可能である。ある目的に対して複雑すぎる状態変移情報を、単純なオブジェクト状態へと丸めて行く際に、この遮蔽が便利である。例えば、図 3-60 に示すような、水を凍結状態から沸騰状態まで加熱するプロセスについての、一連の状態変移があるとする。

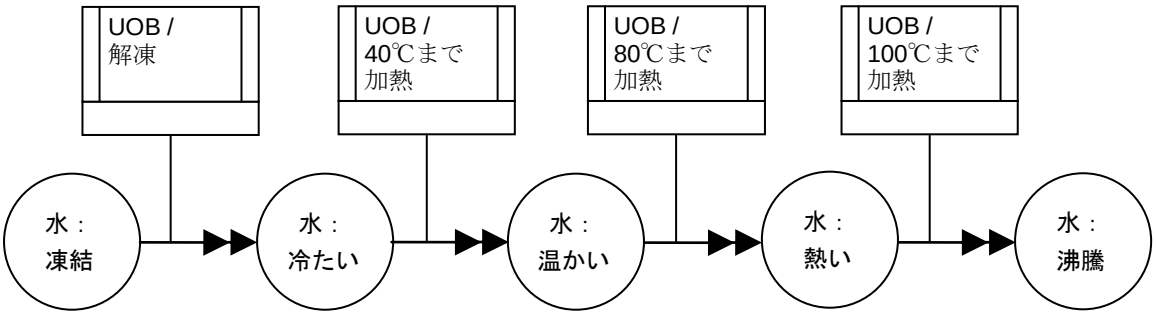


図 3-60　加熱プロセスにおける、オブジェクト変移

ある視野（perspective）において、凍結と沸騰の間にある変移が無意味である場合、こうした変移を図 3-61 にあるように、大まかな（course-graind）に「加熱されている水」と 1 つの状態へとまとめることで、情報の遮蔽を行う。このまとめた状態には二重丸を使用し、またこの場合であれば、このまとめられた状態が遮蔽された情報の形式が状態変移（state transition）についてのものであることを明示しておく為に、「S」の文字（state transition の頭文字）を付加しておく。

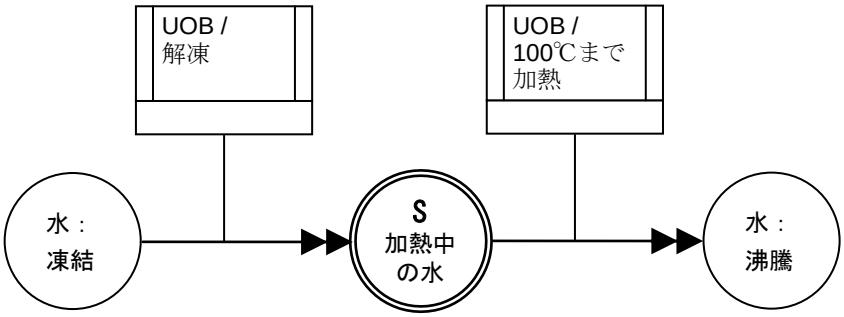


図 3-61　状態変移情報の遮蔽

詳細な（finer-grained）スキーム図から、粗い（coarse-grained）スキーム図へと、一般化（抽象化）を行う手続きは、これといって決まっておらず、アルゴリズム化できない。

例えば、「加熱中の水」という状態シンボルで直接置き換えられているのは、図 3-60 のスキーム図の真ん中の 3 つの種類シンボルと、それらを繋いだリンクとで構成された部分である。しかし、図 3-60 内の具体的な変移スキーム図を、より抽象的な状態変移スキーム図へと置き換え、置き換えたものに最適なラベルを付けるという代替作業を正確に行おうとすれば、表現されているプロセスの本質を考えながら行わなければならず、一般的には非アルゴリズム的な作業となるのである。

３．３．８　拡張変移スキーム図（Enhanced Transition Schematics）

オブジェクト変移の周辺に存在する背景情報（contextual information）は、実際の変移に対しては本質的なものでは無いものの、それと密接に関係していることから、オブジェクト変移を記述して行く際に、そうした背景情報を提供できる手段があると非常に便利である。更に、こうした背景設定（context-setting）を行っているオブジェクトと関係性の目録を作成する事は、便利であるどころか必須ですらある。

背景情報の提供を実現する為には、IDEF5 存在論（ontology）獲得手法の幾つかの構成物を、IDEF3 オブジェクトスキーム図の言語で利用できるようにしなければならない。この IDEF5 構成物の使用は、全くのオプションである。分析を行おうとする人が、単に変移のみを記述したいのであれば、これから説明する追加の構成物は不要である。しかし、こうした IDEF5 構成物に詳しくなる事で、表現力を向上させることができるのである。

以下の小節では、IDEF5 の追加構成物を、変移スキーム図とは独立したものとして説明している。この追加構成物と変移スキーム図との統合については、一通りの説明の後で行う。

３．３．８．１　一階スキーム図（First-Order Schematics）

（以下、訳わからん）

個々のオブジェクト（individual objects）、即ち個体（i.e., individuals）の論理種類（logical type、※下記訳注参照）と、そうした個体（those individuals）の持つ特性（property）の論理種類は、異なったものである。

特性（property）は、別々の複数の個体（distinct individuals）によって「共有」されている、抽象的で一般的な特徴（feature）であり、それを共有する個体を同一であると判断する際に「考慮（respect）」されるものである。またそれと同様に、関係（relation）は、別々の複数の個体の組合せ（2 個以上の個体で構成される組合せ）によって共有可能な、一般的な関連性（association）である。

特性と関係は、個体（individuals）が持つ固有の特徴（feature）を抽象化することにより識別されている。その為、特性と関係は、それらを例示している（exemplify）個体そのものよりも高階な（higher）、即ちより粗く、より抽象的な（i.e., roughly, more abstruct）論理種類（logical type）であると、特徴付け（characterize）されていることが多い。

このような理由から、個体は「一階オブジェクト（first-order objects）」、そしてこの一階オブジェクトの特性と関係は「一階特性と一階関係（first-order properties and relations）」と呼ばれることが多い。「～へと変移する（transition-to）」という関係は、一階関係の典型例の一つである。

※訳注：

※※論理種類（Logical Types）について：　http://behavenet.com/logical-types

Alfred North Whithead と Bertrand Russel によって提唱された「論理種類理論（Theory of Logical Types）」によれば、クラス（クラスセット）とクラスの要素とを区別しなければならないとされている。クラスを参照する宣言は、クラスの要素を参照する宣言よりも、より高位の抽象化（higher level of abstraction）、即ち、より高位の論理種類を表わしている。この区別は、参照がクラス全体に対するものか、それともクラスの要素に対するものかを、外観（outset）から判断不可能なように 2 つの宣言が表現されている時には、特に重要となる。

例えば「人（man)」という言葉は、ある個人（individual being）を参照することも、全ての人間が属している人というクラスを参照することも可能である。後者の場合、「人」という概念は、より高い抽象化階層におかれており、それはより高い論理種類と一致している。

※※Logical Type を「論理型」とすると、プログラミング用語の boolean（True/False）と混同してしまうので、「論理種類」と訳しておく。

「一階スキーム図（First-Order Schematics)」の意味

オブジェクト間に存在する関係や、オブジェクトの特性を表現するスキーム図？
（一階オブジェクトと一階関係・一階特性の範囲内で作図されるスキーム図なので、一階スキーム図？）
　変移スキーム図は、一階スキーム図の一種？

2 つのオブジェクト記号を一階関係記号によって接続した一階関係は、図 3-62 のようになる。

図 3-62　基本的な一階スキーム図の標準形式

こうしたスキーム図には、デフォルトの意味論（即ち、詳細説明（elaboration）言語による追加の説明が無くとも理解可能な、最も広く知られた意味）が必要である。この目的の為、図 3-63 のような具体的な例について考えてみる。

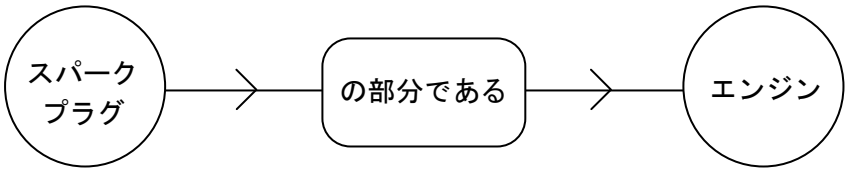


図 3-63　基本的な一階スキーム図の例

簡単に説明するならば、この構成物のデフォルトの意味は「~の部分である（Part-of)」関係の「型式仕様書（type specification)」である。つまり、このように記述することで、スパークプラグとエンジンが、その関係（「~の部品である」）を正当に持ち得るものであることを明確にしている。

この時、例えば「全てのスパークプラグがエンジンの一部である」と言う事も、また「全てのエンジンがスパークプラグを持っている」という事も、どちらも言及されておらず、対象となる分野（domain）には、スパークプラグの外されたエンジンや、プラグレスのエンジンが存在する可能性もある。むしろ、その基本となるデフォルトの意味では、『「スパークプラグ」は、「エンジン」「の部分」に成り得る』という事実を単純に述べているにすぎないのである。より強い意味合いを持たせたいのであれば、IDEF3 の詳細説明言語により明確にすることが可能である。

上のスキーム図の代替となる文法として、2 つの接続シンボルと関係シンボルを、図 3-64 のように、関係のラベルと同じラベルを持つ 1 本の矢印で置き換える事が可能であり、またそうすることが推奨されている。これだと変移スキーム図と混乱してしまう可能性もあるが、スキーム図の他の詳細説明で使用されている「黒塗りの」矢印頭部（訳注：▼）ではなく、「開いた」矢印頭部（訳注：＞）を用いる事で、区別している。

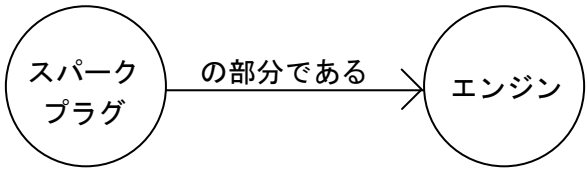


図 3-64　基本的な一階スキーム図の略図の例

2 つの実体の間で結ばれた「の部品である」のような関係は、関係内の論点（argument）の数、もしくは関係の「項数（arity、関数内の引数の数）」が 2 であることを示していることから、「二変数関係（2-place relation）※」と呼ばれることが多い。しかし、関係の「項数」には、何の論理的境界線も存在しておらず、例えば「の間（between）」という関係は、3 つのオブジェクトの間でも結ぶことが可能である。より精巧で便利な関係を、4 以上の論点で定義する事も容易である。

※訳注：二項関係という言葉は binary relation の訳語であるのが一般的なので、ここでは二変数関係としている

二変数（一階）関係記号を含んだ一階スキーム図の意味論は、n 変数関係記号を含んだスキーム図として一般化されている。例えば図 3-65 は、「~へ運搬する」関係のインスタンスが、「コンベア」、「車体」、「下塗り塗料槽（Paint primer vat）」の 3 つと関係可能である事のみを表現している。

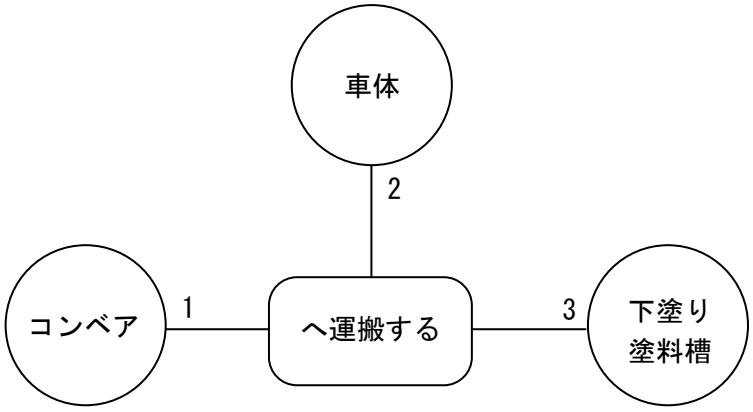


図 3-65　基本的な三変数一階スキーム図の例

二変数（2-place）の場合にはシンボルは矢印で接続されていたが、n 変数化する際には、関係から延びる枝（spoke）に番号を追加することにより、一般化を行っている。この例では「コンベア」、「車体」、「下塗り塗料槽」のそれぞれに、「~へ運搬する」関係の論点位置番号 1、2、3 が割り当てられているが、これは『「コンベア」は運ぶ、「車体」を「下塗り塗料槽」へ（a Conveyer conveys a Car body to a Paint primer vat.）』というラベルの、英語読みによって発生する順番になっている。

二変数の場合には関係記号を省略し、ラベルの付いたリンクで単純化する事が可能であったが、これは n 変数の場合も同様であり、図 3-66 のようになる。この論文では、このような簡易的な表記法を推奨している。

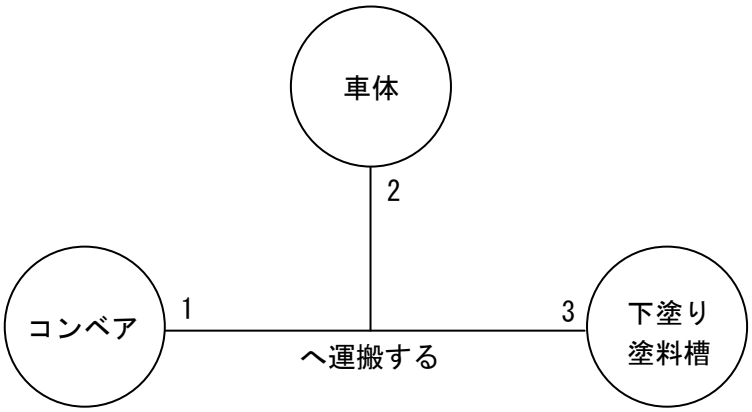


図 3-66　　図 3-65 の代替文法

個体記号（indivisual symbol）を使用する事で、図 3-66 のようなスキーム図に存在する不明瞭さを幾らか解決することが可能である。例えば、図 3-66 に示されるような状況では、複数の塗装槽の使用が容認されてしまうことになる。しかし、ある特定の塗装槽のみに焦点を絞りたい場合であれば、図 3-67 のように個体記号を用いる事で、それを表現にすることができる。

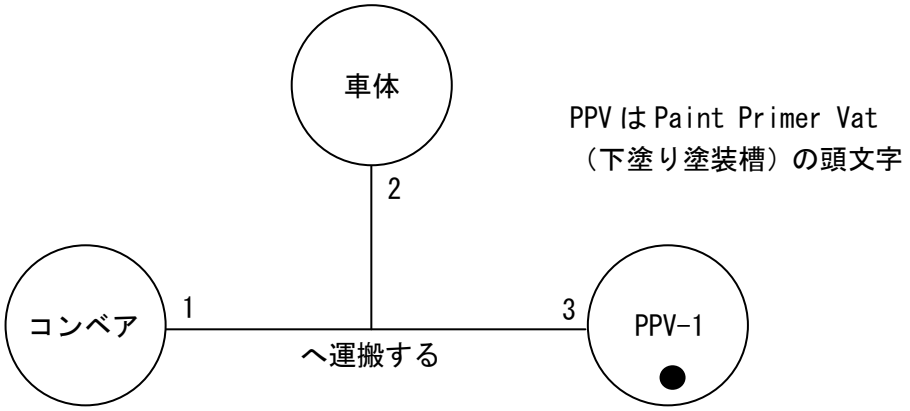


図 3-67　　個体記号の使用例

図 3-67 のスキーム図は、個体記号で指示する事により、あるコンベアが、ある車体を、下塗り塗装槽「PPV-1」へ運搬する事が可能である、という事を表現している。これは図 3-66 と比べて、より明確な主張をしている。

個体記号のみを使用することで、不明瞭さを完全に排除することができる。図 3-68 のスキーム図は、CB-J27-S121 という特定の車体は、コンベア Conv-2 によって、下塗り塗料槽 PPV-1 へと運搬される、ということを表現している。

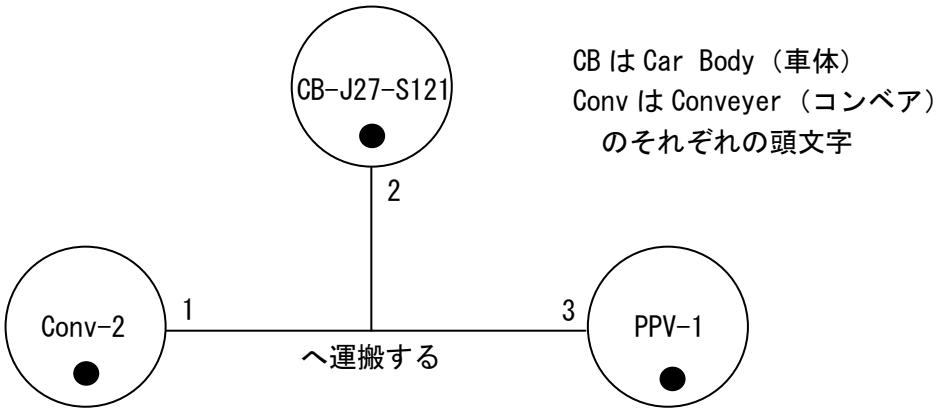


図 3-68　　全てを明確化した例

複数の円を別々の矢印で同一の円へと接続することで、複雑なスキーム図を作成する事が可能である。一般的に、複雑なオブジェクトスキーム図に変移リンクが含まれていなければ、本当に便利である。図形要素の再利用が簡単になる上に、単体の複雑スキーム図で複数の主張を可能になる。例えば、スパークプラグはエンジンの部品と成り得るという事と、エンジンは自動車の部品と成り得るという事の、2 つを同時に表現する場合であれば、「エンジン」という種類の表現に円は 2 つも必要ではなく、図 3-69 のように 1 つに結び付ける事で、より上手く表現する事が可能となる。

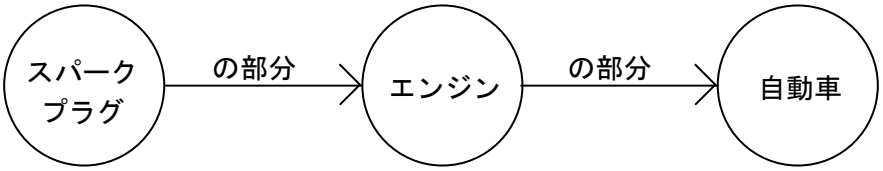


図 3-69　小規模な複雑スキーム図

同様に対象となる分野に「自動車はデトロイトで製造される」、「デトロイトから販売店へと出荷される」、という情報を追加する事も可能である。この情報は図 3-70 のように簡単に表現することができる。

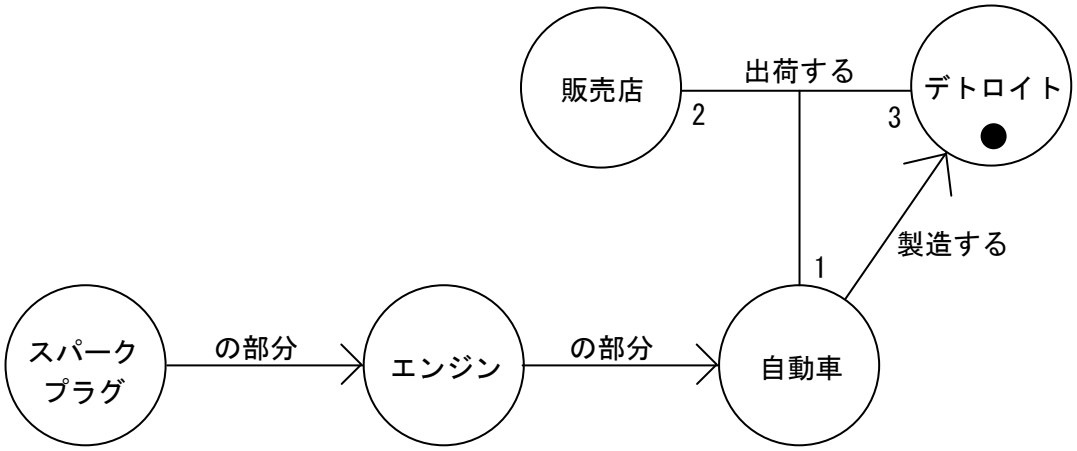


図 3-70　複数の関係を持つ複雑スキーム図

それと同時に､オブジェクトスキーム図内の関係を１種類のみにすることも可能である。この場合、スキーム図を読み易くする為に、分析者はスキーム図の下端に関係シンボルの意味を簡単に記述し、ラベルを省略することが可能である。

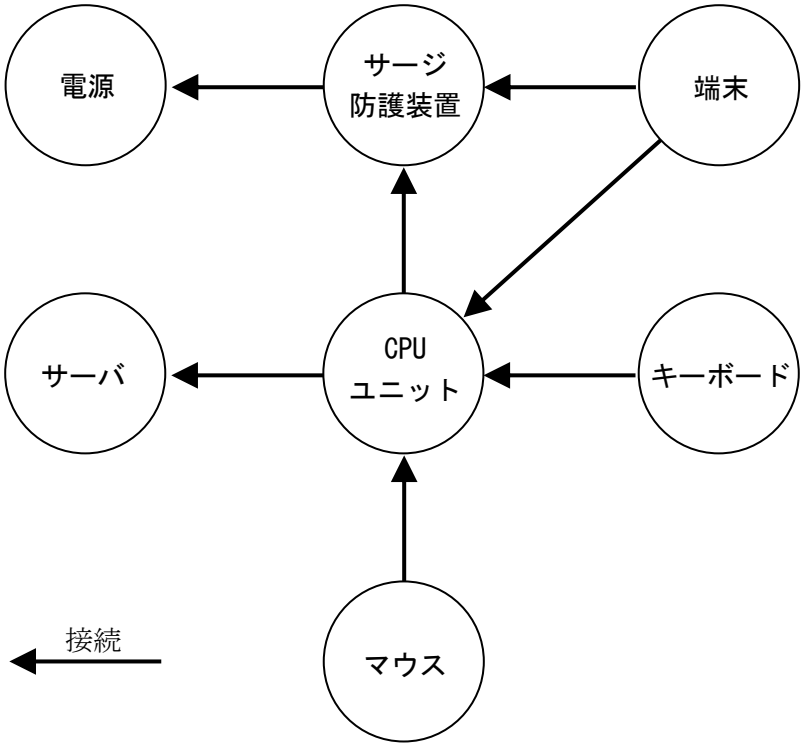


図 3-71　パソコンへの周辺機器の接続

３．３．８．２　構成スキーム図（Composition Schematics）

「の部分である（part-of)」という関係は、設計やエンジニアリング、製造において、余りにも一般的であることから、「の部分である」ラベルと、それに関連する公理（axiom）は、IDEF5 言語に特別に用意されている。部品表（Bills of Material、BOM、素材一覧表？）はこの書式表現の一般的な例である。これにより、ユーザーは、オブジェクトの対象となる種類の構成についての事実を表現する事が可能である。一般に、オブジェクト間の関係の構成の表現は、図 3-72 のような書式のスキーム図によって行われている。

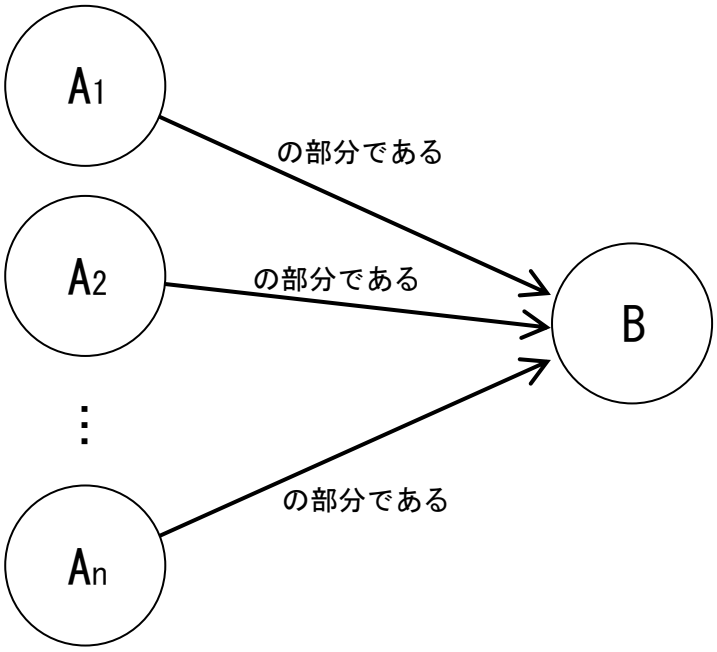


図 3-72　　構成スキーム図

図 3-72 のデフォルトのスキーム図は、インスタンス A1 は B の部分と成り得る、インスタンス A2 は…、インスタンス An は B の部分と成り得る、という事を意味している。しかし、「の部分である」の文脈では、より強い解釈（reading）が望まれることが多い。例えば部品表（BOM）では、単純に「インスタンス A1 は B の部分と成り得る」などとは言わずに、「全ての B は実際にインスタンス A1、A2、等によって構成されている」と言う。例えば、ある種類のボールペンについての構成物構造を表現しようとすると、図 3-73 のようになる。

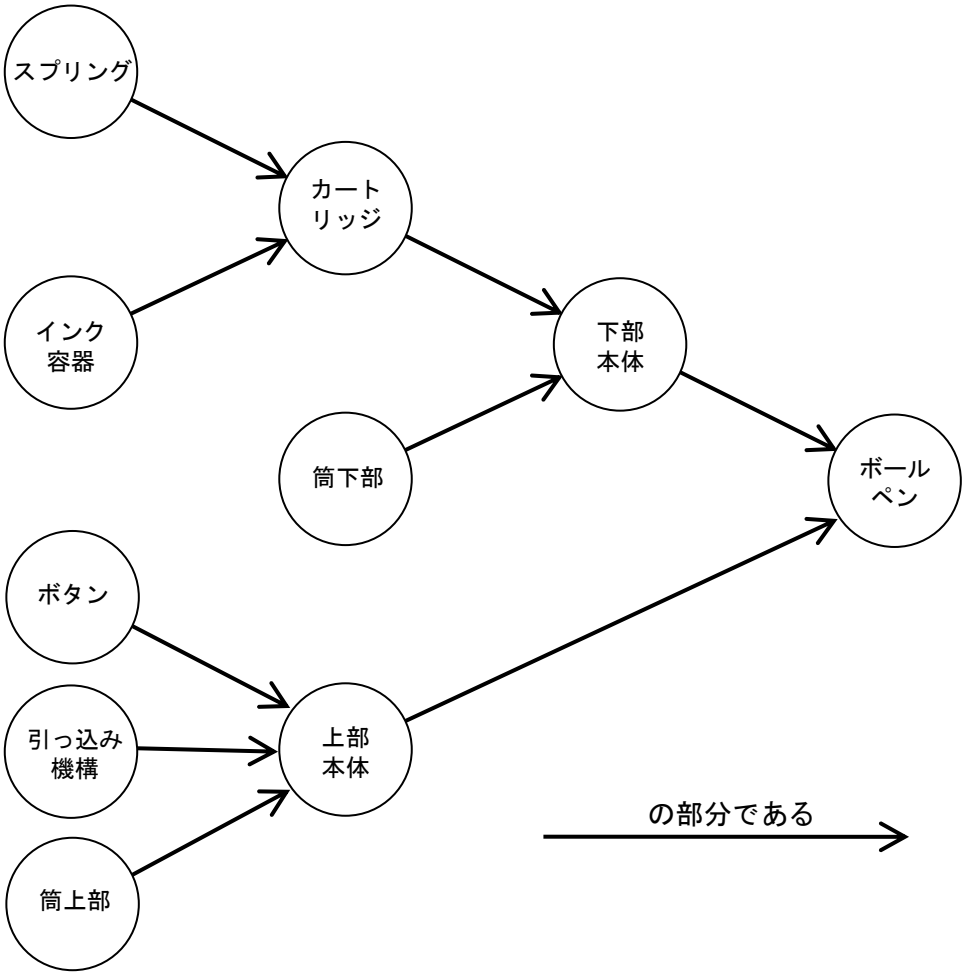


図 3-73　構成スキーム図

これを強調する為には、通常は記述（note）や詳細説明言語（elaboration language）[7]に頼らなければならなかった。そこで、図 3-73 のスキーム図のような強い意味論を用いる事により、分野の問題となっている部分で：

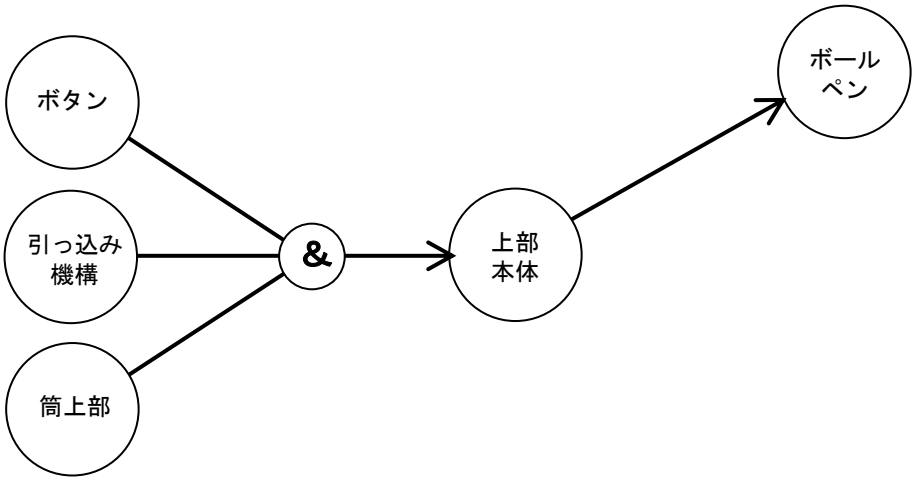
・ボールペンは上部本体と下部本体の両方を持つ
・上部本体はボタンと引っ込み機構（retraction mechanism）と筒上部（upper barrel）によって構成されている
・下部本体は筒下部とカートリッジによって構成されている
・カートリッジはスプリングとインク容器によって構成されている

という事を表現している[8]。

[7]：種類 B のインスタンスが種類 A1、A2、A3 の 3 つの部分を持つ場合、詳細説明言語文では次のように記述する

(forall ?x (-> (B ?x) (exist (?y_1 ?y_2 ?y_3) (and (A_1 ?y_1) (A_2 ?y_2) (A_3 ?y_3) (part-of ?y_1 x) (part-of ?y_2 x) (part-of ?y_3 x)))))

[8]：構成スキーム図に交差点を追加する事で、解釈の可能性の範囲を狭くすることが可能である。例えば「&」交差点を複数の「の部分である」リンクの「合流部」に使用することで、構成から 1 個以上の付加オブジェクトが排除されてしまう可能性を防ぐことが出来る。上のような（[7]）詳細説明言語文が無い場合、図 3-73 だけだとスプリングや引っ込み機構の無いボールペンを許容してしまう事になる。そこでスキーム図に交差点を追加することで、スプリングとインク容器の両方がボールペンのカートリッジの部分であり得るが、スプリングもしくはインク容器の無いカートリッジはあり得ない、という事を表現する事が可能になるのである。



訳注：　&交差点を用いた構成スキーム図？
（例が無いので、本当にこういう風に描くのか不明）

３．３．８．３　二階スキーム図（Second-Order Schematics）

オブジェクトの個体間で保持されている特性と関係は、例え抽象的なものであっても、それら自身をオブジェクトとして識別することができる。ただし、それらは通常の一階オブジェクトよりも１つ高い抽象レベルにある、上位の「論理種類（logical type）」から発生したものであることから、オブジェクトとしては「二階（second-order）」オブジェクトとして分類される。

一階特性と一階関係は、（二階）オブジェクトとして取り扱われる場合には、特性を持つことが可能である。このような一階特性と一階関係が持つ特性は、二階オブジェクトに属することから、一般的に「二階特性」として知られている。そして一階特性や一階関係のような二階オブジェクトもまた、お互いに関係を持つことが可能である。

このように、個々のオブジェクトに適用された種類（kind）、特性（property）、そして関係（relation）は、個体、つまりは「一階」オブジェクトよりも「上位（higher）」の、より普遍的な論理階位（logical order）にあることから、一般には「二階」オブジェクトとして知られている。個体と同様に、二階オブジェクトもまた、他の一階もしくは二階オブジェクトと関係を持つことが可能である。わかりやすい例では、種類間で結ばれた「~の下位種類（sub-kind-of）」や、個体と種類（もしくは一般的には特性）の間で保持されている典型的な関係である「~のインスタンスである」などが挙げられる。

訳注：結局、以下のようになる？

一階スキーム図：

オブジェクトのインスタンス（個体）間の関係・特性を説明するスキーム図

一階オブジェクトと、その関係・特性（一階関係・一階特性）が登場するので、「一階」スキーム図と呼ばれている？

二階スキーム図：

関係・特性の間の関係・特性を説明するスキーム図

関係・特性（一階関係・一階特性）をオブジェクトとして見立てた物（二階オブジェクト）と、それらの間の関係・特性（二階関係・二階特性）が登場するので、「二階」スキーム図と呼ばれている？

一階関係と二階関係のどちらも円に接続されているものの、それぞれの意味論が非常に異なることから、二階関係を表現する為に、一階関係とは別の種類の矢印が必要となる。その為、二階意味論の基本形式は、関係を表わす際に図 3-74 に示されているような二階関係矢印と呼ばれている表現が用いられている。

図 3-74　基本的な二階スキーム図

二階スキーム図の意味論は、殆どの一階スキーム図の意味論よりも、遥かに限定的（明確、具体的）である。特に、二階スキーム図は、インスタンス（具体化されたオブジェクト）についてではなく、種類について表現したものになっている。図 3-74 のスキーム図では、左側の円によって表現された種類と、右側の円によって表現された種類とを矢印で結ぶことで、2 つの種類の間の（二階）関係を表現している。

その上、デフォルトの意味論（default semantics）は一階スキーム図程、限定されていない。一般的な一階スキーム図の意味論は、「対象物が対象分野上にどのように存在しているのか」ということを表わしているが、二階スキーム図の意味論は、そうしたことは殆ど触れられておらず、むしろ 2 つの種類が実際にどのように関係しているのかについて示している。

図 3-75 は、アメリカ国民の方がカナダ国民よりも多い、という事を表している。（より文法的に表現するならば、「アメリカ国民」という種類は、「カナダ国民」という種類よりも、多くのインスタンスを持っている、となる）

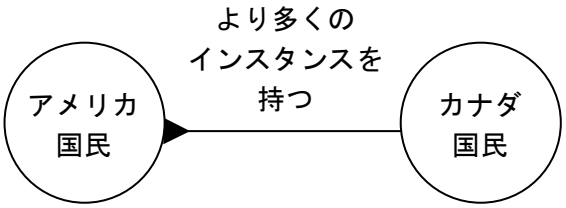


図 3-75　基本的な二階スキーム図の例

図 3-76 は、二階関係「の下位種類（subkind-of)」を用いたスキーム図である。意味論によって、種類「六角ボルト」は種類「留め具」の下位種類である、ということだけが示されている。

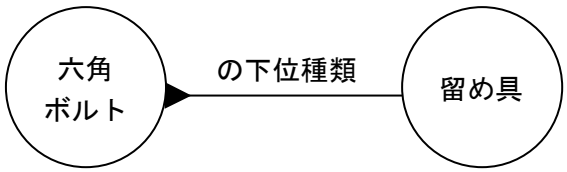
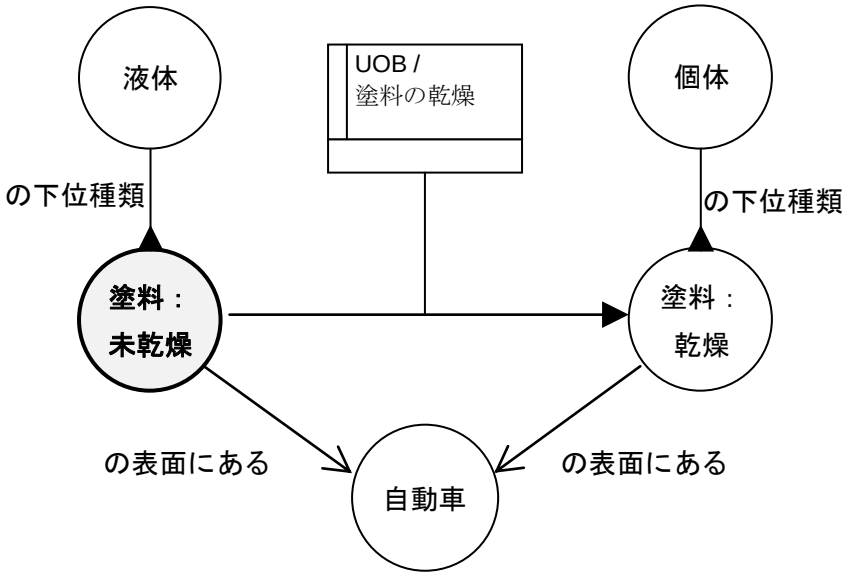


図 3-76　「の下位種類」を用いた二階スキーム図の例

訳注：
状態変移、一階関係、二階関係の 3 種類のオブジェクトスキーム図が紹介されたが、そうした区分けをわざわざしている理由は、下図のように 3 つを混合して描かれたスキーム図を見ると理解しやすい。

「塗料：未乾燥」は：
- 乾燥前の塗料という特性として、「液体」との間に、「の下位種類」という二階関係を持っている
- （未乾燥）塗料という個体として「自動車」との間に、「の表面にある」という一階関係を持っている
- 未乾燥状態にある塗料として、UOB「塗料の乾燥」を介する状態変移により、乾燥状態にある塗料へと変化する可能性を示している



訳注図：図 3-82　オブジェクト変移構造を持ったオブジェクトスキーム

３．３．８．４　分類スキーム図（Classification Schematics）

下位種類関係（subkind relation）が最も一般的であることから、ラベルが付いていない二階関係矢印は、デフォルトで下位種類関係という意味で用いられている。これにより、ユーザーはスキーム図上で「の下位種類」というラベルを繰り返し記述する手間を省くことが出来る。このように下位種類関係が二階関係のデフォルトに設定されたのは、知識表現の為の一般的なメカニズムは、分類学（taxonomy）ダイヤグラムの中に見受けられる（Brachman, 1985）という観察が基になっている。

知識獲得作業に従事している分野専門家は、「A は B である（is）」や、「A は B の類型である（is a type of）」、「A は B の一種である（is a kind of）」といった宣言を行う事が多い。こうしたやり方で行われている知識の組織化における認識活動は、「分類（classification）」と呼ばれている。

分類には、幾つかの同一視可能な類別（identifiable varieties）が存在している。その中の主要なものとして「記述包摂（description subsumption）」と「自然種分類（natural kind classification）」の 2 つがある。

記述包摂では、以下の 2 つの事が成立している：

（１）他の全ての下位種類の特性はもちろん、分類の中で「最上層」に位置している種類 K の特性は、種類 K に属する為の資格（membership）として必要となる、厳密な必要十分条件（rigorous necessary and sufficient condition）を構成している。

（２）個々の種類の特性が K の特性を継承している（entail）ことから、全ての下位種類の特性は、種類 K の特性によって「包摂されている」。つまり、K の定義済み特性は、最も一般的な概念を構築している。

それとは対照的に、自然種分類では、最上層の種類 K に属する為の資格（membership）としての、厳密な同一視可能な必要十分条件（identificial necessary）の存在が保証されていない。その代り、様々な方法による特殊化により K の下位種類の生成を繰り返す事で出来上がった、潜在的に構造化された特性の存在が保証されている。この分類手法の最もわかりやすい例としては「金属」や「ネコ科」といった純粋に自然に存在している種類（genuine natural kinds）があるが、他にも「自動車」や「NC マシン」のような人工物の持つ種類も挙げる事が可能である。

記述包摂と自然種類分類の分類の例を、図 3-77 に図示する。

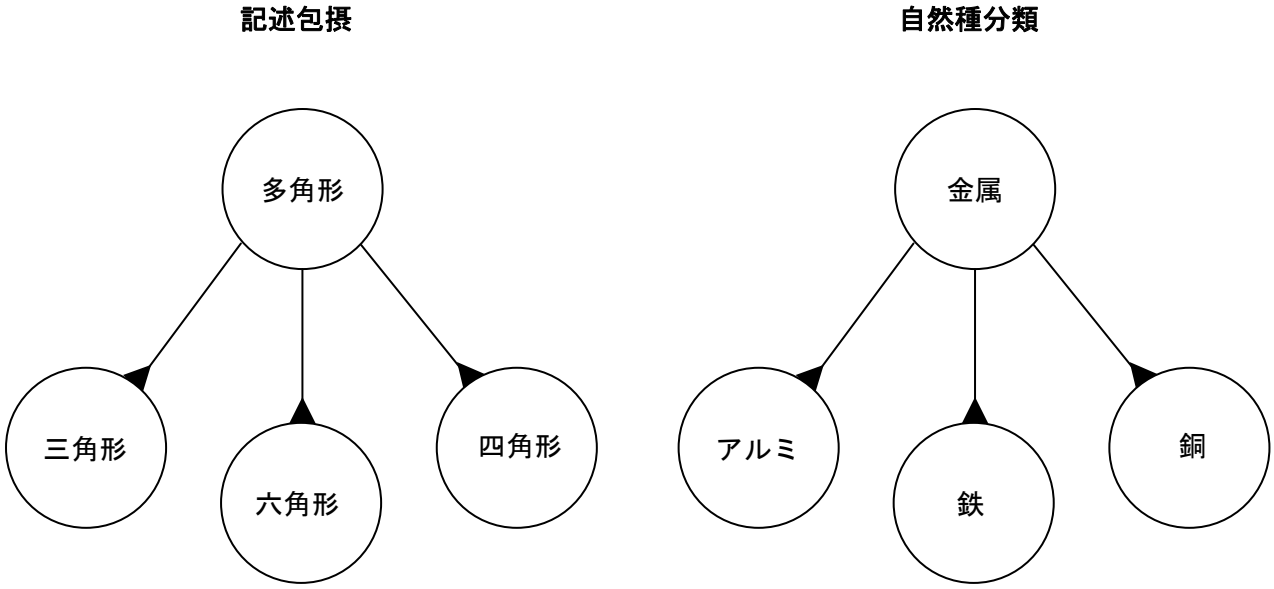


図 3-77　分類の異なった種類

訳注：記述包摂（description subsumption）について

左上の多角形の例を拡張したものを下に挙げる。それを基に考察してみると：

・種類毎に持つ特性の定義が厳密である

・新たに特性を追加する事により下位種類が定義されてゆく

　その際、ある階層種類の特性は、そこより下にある全ての下位種類に継承されるということか？。

<table>
<tr><td colspan="5">多角形
n 本の互いに交差しない線分によって構成された閉じた二次元図形</td></tr>
<tr><td colspan="2">三角形
n=3</td><td>六角形
n=6</td><td colspan="2">四角形
n=4</td></tr>
<tr><td>二等辺三角形
ある 2 辺の長さが同じ</td><td>直角三角形
2 辺間の角が 90 度</td><td rowspan="5">正六角形
辺の長さが全て等しい</td><td colspan="2">台形
向かい合う 1 組の辺が互いに平行</td></tr>
<tr><td rowspan="4">正三角形
辺の長さが全て等しい</td><td rowspan="4">直角二等辺三角形
等辺の為す角が 90 度</td><td colspan="2">平行四辺形
残りの 1 組の辺も平行</td></tr>
<tr><td>長方形
頂点の角が全て 90 度</td><td rowspan="2">菱形
辺の長さが全て等しい</td></tr>
<tr><td>正方形
辺の長さが全て等しい</td></tr>
<tr><td colspan="3"></td></tr>
</table>

上層の特性は、全ての下位種類に厳密に継承されている

訳注図：多角形グループの分類

訳注：自然種分類（natural kind classification）について

自然種（natural kinds）の分類法

自然種：水や金属といった天然に存在する物？

　本文中では犬や猫も自然種として挙げられているが、生物学ではこうした生物種は自然種とは別なものという認識が持たれるようになっているしい。また下記の文章では、本文中で挙げられていた自動車等の人工物（時間と共に進化してきた人工物）もまた、自然種ではないとしている。
（http://www.wakate-forum.org/data/2013/resume1.php）

…幾つかネットに転がっている、こうした自然種や分類学についての日本語のドキュメントを見てみたが、言っている事がバラバラで、また私には理解できない内容で、どうしたらいいのか判らなくなってしまった…

訳注：結局のところ…

この本文で言う所の、2 つの主な分類方法は以下のようになる？？。

（1）分類の基準が先に明確に定義され、それを基にして対象物が仕分けられ、その分類を基にして対象物が取り扱われる。もしくは、ある分類基準の下で対象物の製作や設定が行われる。分類の基準とされる特性は、後から対象物に付け足されている。
　記述包摂。多角形や東京都民等。Classification（次ページ）。

（2）複数の対象物が先天的に持つ共通の特性らしきものを見つけ出し、それを分類の基準とする。分類の基準とされる特性は、対象物が先天的に持つものである。
　自然類分類。金属や（時間軸を固定した状態での）生物種、Categorization（次ページ）

種類というものの主要な観念を考慮すれば、一般的なオブジェクトスキーム図言語を自然に適用することが、即ち分類学ダイヤグラム（taxonomy diagrams）の作成となることは明白である。今後は、この分類学ダイヤグラムを「分類スキーム図（classification schematics)」と呼ぶことにする。

一般的に分類は、例の提示（example suggest）よりも遥かに詳細である。殆どの分類体系（classification scheme）は、体系内の、より一般的な種類の「下に」、より特殊な下位種類を持つ複数階層を伴った構造をしている（「~の下位種類（subkind-of)」と「~のインスタンス（instance-of)」はどちらも、意味ネットワーク（semantic net）や他の図形言語の中で「~は~である（is a)」関係によって、曖昧に表現されていることが多い。こうした体系は「~は~である　階層構造」と呼ばれている事が多いが、「~は~である」は努めて使用しないよう推奨されている。その代りに、意図しようとする意味に応じて、「~の下位種類」関係、もしくは「~のインスタンス」関係を使用するべきである)。

例えばプロジェクト計画では、プロジェクトを成功させる為に必要となる「リソース」の種類を整理（categorize）する作業が必須である。非公式ではあるが、リソースを、活動を実行する為に消費し、使用し、必要とするオブジェクトとして、定義する事が可能である。リソースはプロセスにおいて実行の必需物（enabling）の役割を果たしている。図 3-78（次ページ）の例のような分類スキーム図は、プロセスに必要なリソース整理（categorizing）の本質的な手法（natural way）の一つである。

訳注：categorization と classification の違い

	Categorization	Classification
プロセス	状況（context）もしくは知覚された類似性を基にした、実体の独創的合成体	必要十分な特徴の分析を基にした、実体のシステム的配列
境界	どのグループも資格が縛られていない為、境界は「曖昧」	クラスは完全に排他的で重なりが無い為、境界は固定されている
資格	柔軟： カテゴリー資格は、一般化された知識や直接的状況（immediate context）を基にしている	厳密： ある実体が特定のクラスに属するか、属さないかは、クラスに内包（intension）されているかどうか、を基にしている。
登録基準	状況（context）への従属性・独立性の両方を基準とする	予め定義されたガイドラインや原理を基準とする
典型性	同一カテゴリー内のメンバーを、典型性によって順位づけすることが可能	あるクラス内の全てのメンバーは平等
構造	実体のグループは、階層構造を形成することもある	固定されたクラスによる階層構造

https://pdfs.semanticscholar.org/774e/ab27b22aa92dfaa9aeeeafbe845058e85f58.pdf
「Classification and Categorization: A Difference that Makes a Difference」ELIN K.JACOB 著　より

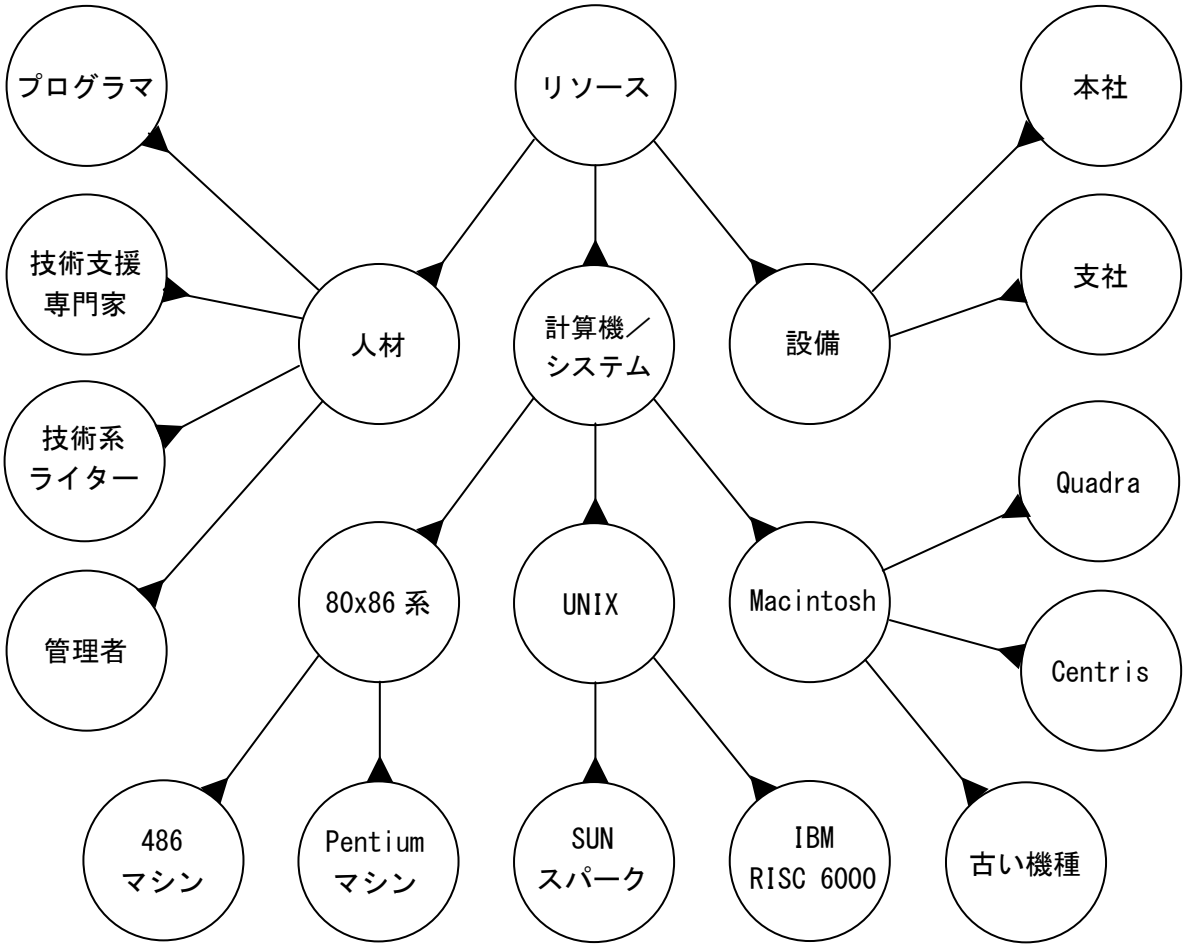


図 3-78　リソースの分類

３．３．８．５　構成と分類情報の遮蔽
（Hiding Composition and Classification Information）

これまでにも説明してきたように、構成物の関係を用いる事で、極めて詳細なスキーム図を生成する事が可能である。しかし、こうした詳細さは、同時に煩雑さを生む原因にもなる。例えば、種類「ボールペン」の構成構造の記述以外にも、その種類と種類のインスタンスに関する他の多くの関係についても、説明したいことがあったとする（例えば、「そのボールペンは Washington 州 Sequim で生産する事が可能である」とか、「万年筆は一般的にボールペンよりも高価である」とか、「ボールペンはペンの下位種類である」等）。しかし、種類の構成構造が不適切となる文脈（context）も多く存在する為、そうした場合に、その文脈に対して不適切な情報を遮蔽することができると便利である。そこで、図 3-70 で図示されているように、（通常のものは一重丸であるのに対して）二重丸を使用し、また遮蔽された情報の種類を区別する為に、円の上に「P」（の部分（part-of）という意味）を添えることで、ダイヤグラム上の遮蔽された情報を表現することにする。

また下の例では、一階関係記号と二階関係記号が、同一のスキーム図で用いられている。

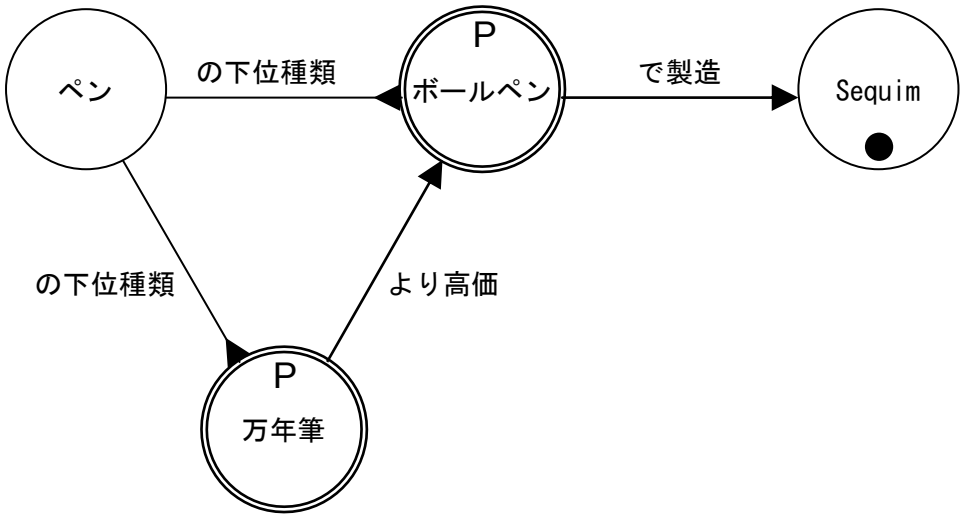


図 3-79　構成情報の遮蔽

訳注：
価格の上下、生産地、下位種類といった事についての説明において、ボールペンや万年筆そのものの構造がどうあるのか、ということは不必要である為、情報の遮蔽を行っている。

同様に、オブジェクトスキーム図においても、分類の詳細を遮蔽することが便利である事が多い。例えば、設備と人材を協調させる必要がある場合のような文脈において、計算機とコンピューターシステムに関する情報は明確にする必要が無い。構成関係で行ったのと同様に、遮蔽された情報を二重丸にし、そしてこの場合には、上に「C」（分類（classification））の文字を記入する。このようにして、図 3-78 のコンピューターシステムの下位種類に関する情報を遮蔽し、その一方で施設に関する情報を追加した物が図 3-80 となる。

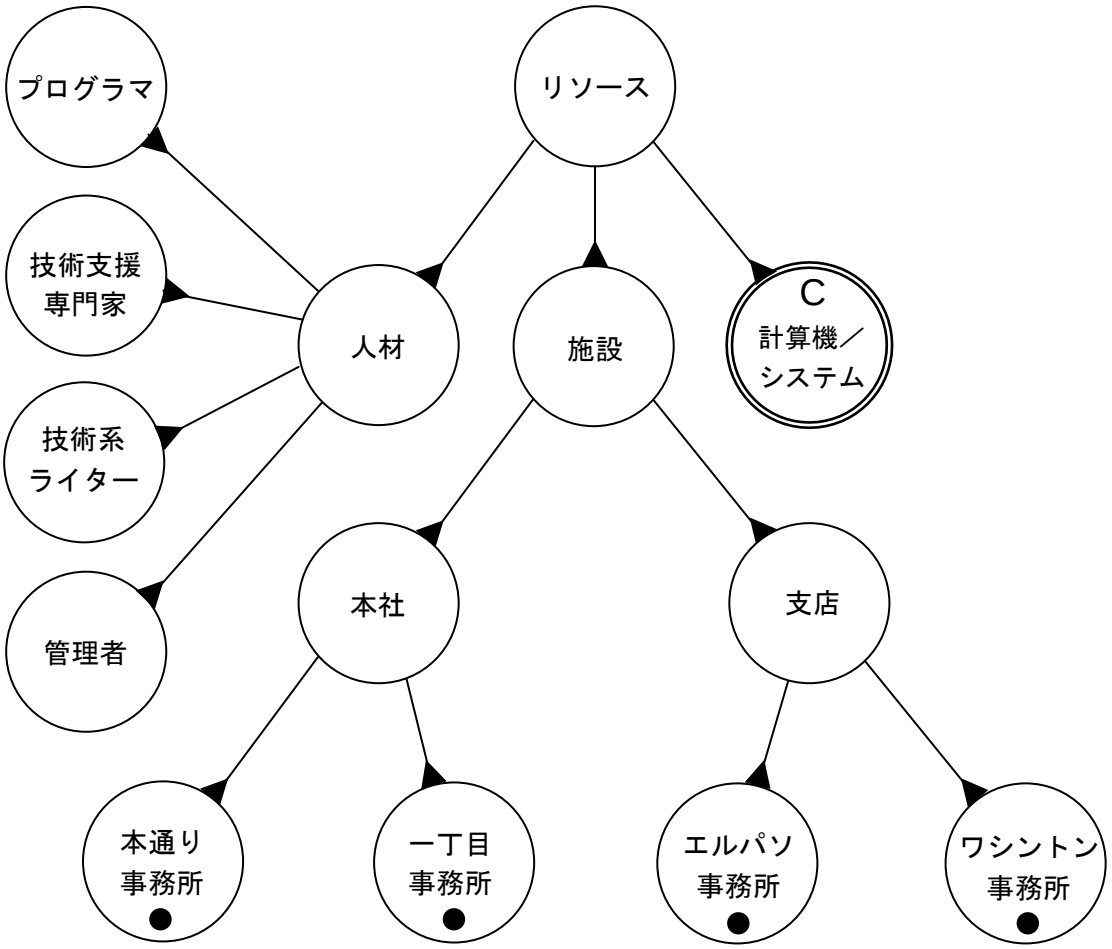


図 3-80　情報遮蔽を用いたリソースの分類

３．３．８．６　拡張変移スキーム図の作成（Creating Enhanced Transition Schematics）

一般的なスキーム図構造に、記述作成作業の背景と目的にとって便利な情報を追加する事で、変移スキーム図の拡張を行うことが可能である。変移スキーム図を中心にし、これに背景設定情報を追加することで、図 3-81 のようになる。この図では、加熱プロセスでの水の状態変移が表現されているが、これに「の下位種類」関係を追加されることで、「水」という種類が、色々なオブジェクト状態によって表現された下位種類を持っている事が説明されている。

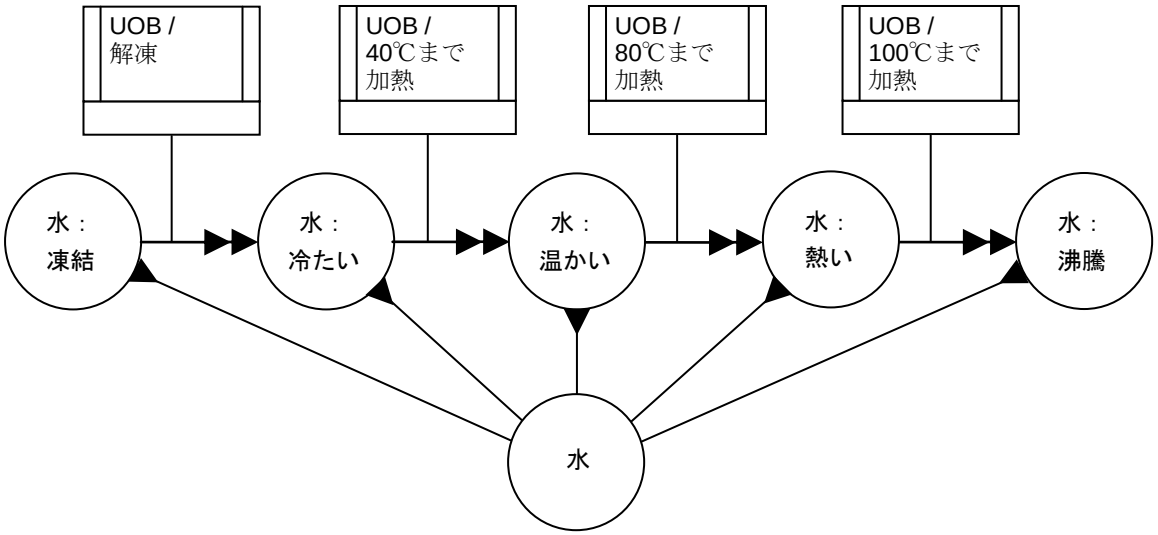


図 3-81　状態変移を中心に表現した合体スキーム図

別の例として、図 3-82（次ページ）のように、変移スキーム図と一般的なオブジェクトスキーム図構造を統合したものがある。

この例では、乾燥プロセスを経て、ある分量の塗料が未乾燥状態から乾燥状態へと変移している事に加えて、関係記号を使用する事で、状態「塗料：未乾燥」と状態「塗料：乾燥」もまた、矢印上のラベルに示されているように、「液体」と「個体」という他の種類とも関係している事を示している。

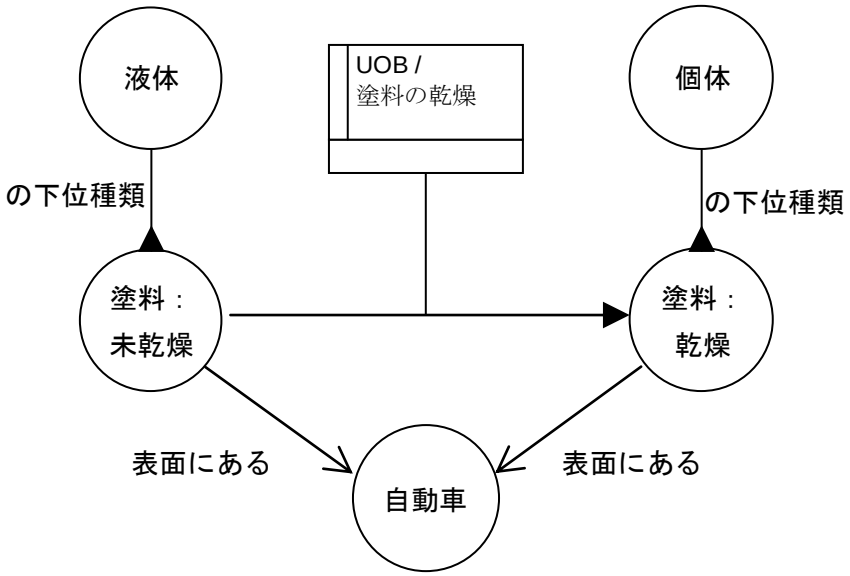


図 3-82　オブジェクト変移構造を持ったオブジェクトスキーム図

更に複雑な例として、図 3-83 のようなスキーム図を考えてみる。

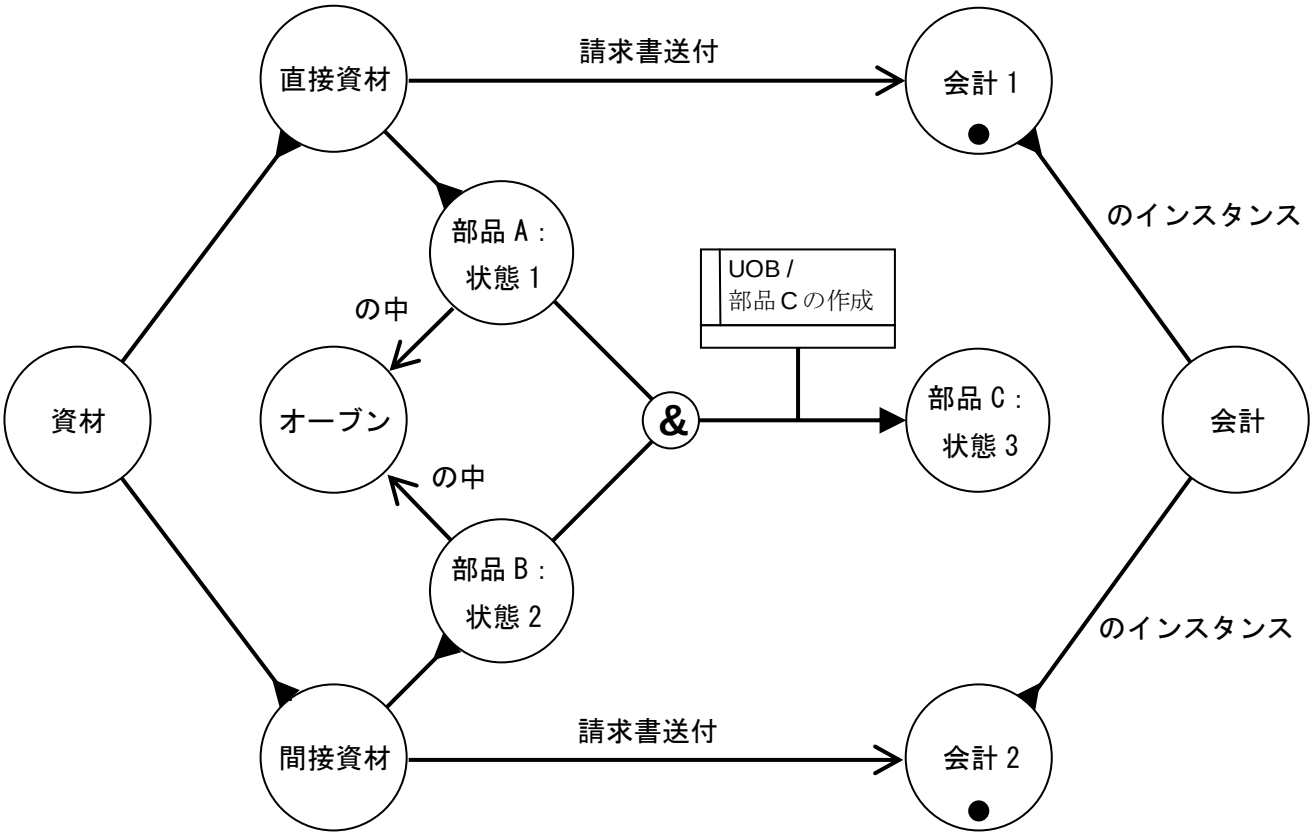


図 3-83　オブジェクト変移構造を持った複雑なオブジェクトスキーム図の例

訳注：元図では、部品 A～C は、それぞれ Frammitz、Widget、Grommet となっていたが、ニュアンスを日本語に翻訳不能だった為、理解しやすいように部品 A～C に置き換えている

このスキーム図の中心部では、「状態 1 にある部品 A」と「状態 2 にある部品 B」が、「部品 C の作成」プロセスを経て「状態 3 にある部品 C」が生成されるという変移が表現されている。それに加えて関係リンクを使用する事で、部品 A と部品 B が、このプロセスの中で「オーブン」に入れられることを表現する事が可能である。更に、次のような文脈情報（contextual information）も追加可能である：

（1）部品 B は「間接（費）資材」、部品 A は「直接（費）資材」である
（2）間接資材も直接資材も、どちらも「資材」の下位種類である
（3）間接資材と直接資材が、どの会計として処理されるのか

上に挙げた、一般的なオブジェクトスキーム図構造を用いる事により、分析者は、対象となる変移の周辺に存在する、様々な種類の文脈となる詳細物を補完することができる。

その一方で、図 3-83 のスキーム図についての上記のような解釈が、唯一のものでは無いと言う事にも注意しなければならない。特に、変移スキーム図に追加された情報のある部分が、対象となっている変移にのみ関係しているものなのか、それとも一般的に関係しているものなのかということを、どのようにして判断するのかという点が重要となる。

例えば、「部品 A：状態 1」と「オーブン」との間のリンク「の中」は、「表現された変移（のインスタンス）において」、変移中の部品 A はオーブンの内部にある、という意味に解釈されている。しかし一方で、部品 A は常にオーブンの中に存在し、そして常に状態 1 であるという意味にも解釈できてしまうのである。

それと同様に、このスキーム図について、部品 A は常に直接資材であると考慮されているのか、それとも対象となっている変移のインスタンスとして使用されている部品 A のみが直接資材であると考慮されているのかについても、決定することもできないのである。もしかすると、別の文脈（で同一状態）の部品 A は、直接資材ではなく間接資材として考慮されるかもしれないのである。

IDEF3 では、一般的な解釈は「デフォルト（default）」として取り扱われる。デフォルトとであるということは、他に指示されていない場合には、スキーム図上に記録された関係は、最も可能性の高い文脈によって獲得される、と言う事を意味している。可能性の低い文脈を意図しているのであれば、記述（note）か、より公式な詳細説明言語によって、この事を記録することが可能である。

（訳注：例えば図 3-83 において、左半分にある 4 本の二階関係矢印にはラベルが書かれていない。これらは、二階関係で最も一般的な「の下位種類である」というデフォルトの解釈を行われる）

３．４　詳細説明（Elaborations）

スキーム図要素（UOB 箱や交差点、リンク、オブジェクト記号等）に付加できる「詳細説明（elaboration）」は、対象となっているスキーム図要素によって示されている実体（entity）についての詳細な特徴（characterization）を伝えてくれることから、プロセス記述を理解する為に非常に重要なものである。こうした詳細な特徴は、「詳細説明ドキュメント（elaboration document）」によって表現される。詳細説明ドキュメントには、一般的に、次の物が含まれている：

（１）スキーム図上の要素の名前、ラベル、番号

（２）要素が意味する実体と関係した、オブジェクト種類、インスタンス、事実（fact）、制約（constraints）

（３）その実体の、文章による記述

事実と制約の間の区別について、説明しておく必要がある。事実（fact）とは単純に、あるプロセスの中にある少なくとも 1 つのインスタンスで、持続されていることが観察されている宣言（statement）である。例えば、塗装／乾燥プロセスで、「部品塗装」UOB のインスタンスは 4 分 30 秒かかる、とか、プロセスに投入される部品の色は灰色である、というものである。このような事実の記述は、対象となる UOB のインスタンスで何が発生しているかということを、単純に記録したものである。

記述された事実を大量に蓄えておくことにより、IDEF3 記述の構築の初期段階での作業が便利になる。蓄積された事実は、一般的に素材データ（raw data）の役割を果たし、プロセス記述をより明確で正確なものにしてくれる。プロセスの知識が増大するに従い、そのプロセスのインスタンスで何が発生しているのか、という事だけではなく、そのプロセスの全てのインスタンスで何が発生しなければならないのか、という事についても記録する必要が出てくる。こうした発生すべき事実を、IDEF3 では制約（constraints）と呼んでいる。維持すべき事実もまた、事実を代表したものであることから、制約は事実の特別な種類（a special kind of fact）であると言える。

制約は、「絶対的制約（absolute）」と「条件付き制約（conditional）」の 2 つに、大きく分ける事ができる。絶対的制約は、例えば「全ての郵便には、郵便番号が表示されていな

ければならない」といったような、条件を伴わない宣言である。一方の条件付き制約では、「もしも（IF)、A という状況が維持されていることが明確であれば（THEN)、B という状況も同様に維持されていることが明確である」という形式で条件が示されている。例えば、塗装／乾燥プロセスでの制約の場合であれば、「もしも（部品塗装 UOB のインスタンス内で）塗装されたオブジェクトが種類 K であれば、部品塗装インスタンスの期間は正確に 5 分でなければならない。」となる。この条件が示されている場合、別の種類のオブジェクトであれば、たった 4 分で塗装を終えても構わないことになる。

あるプロセス記述において識別された全ての要素は、その要素に関する詳細記述ドキュメントを持っている。詳細説明ドキュメントは参照番号のみで構成されているが、オプションでラベルを追加できる。この詳細説明ドキュメントにより、多くの情報を付加されることから、複雑なプロセスを構成する要素群の理解が促進される。詳細説明ドキュメントの詳細な説明と構成については、第 4 章の「IDEF3 記述の作成」で行う。

一般的に、詳細説明ドキュメントには、自然言語を用いた宣言（訳注：つまりは普通の文章）で形作られている（populated with)。詳細説明ドキュメントに対して、自然言語を用いた宣言よりも、構造的で正確でなければならない場合には、IDEF3 の詳細説明言語を用いる事ができる。次の小節では、この詳細説明言語の例について説明して行く。また付録 A には、詳細説明言語の完全な説明と、更に多くの例を収録している。

３．４．１　詳細説明言語の幾つかの例

詳細説明言語は、知識交換フォーマット（Knowledge Interchange Format, KIF）として知られている最近登場した情報共有標準のサブセット（subset、部分集合）に、幾つかの改善を加えたものをベースにした論理言語である（[Genesereth & Fikes, 1992]）。この情報共有標準のサブセットは、ほぼ全ての論理言語に必要とされている基本要素を内包した、詳細言語の「基本部分（core）」として有名であるが、これを、IDEF3 スキーム図言語によって表現されているプロセスと変移に関する、正確な情報を表現する為に作られた IDEF3 仕様構造（specific construct）を幾つも重ねることによって、拡張している。

しかし、これを効率よく行うには、言語についての明確な意味論（semantics）を持つことが必須となる。IDEF3 スキーム図の直観的意味論（intuitive semantics）は、最近開発された情報理論である状況理論（situation theory）を基にしている（[Barwise & Perry, 1983]：この理論の公式な概要については、付録 A の第 A4 章を参照のこと）。

詳細説明言語の中で、UOB やプロセス等の IDEF3 の基本概念は、状況理論に基づいたある基本的な意味論区分（category）によって識別されている。詳細説明言語の基本部分に追加された構造（constructs）は、これらの区分に一致している。

訳注：この辺、良くわからない。

３．４．２　プロセススキーム図での詳細説明言語の使用例

プロセススキーム図との組み合わせで詳細説明言語の使い方を説明する為、図 3-84 のようなスキーム図で表現されたプロセスについて考えてみる。このプロセスを「PQD（塗装／待機／乾燥プロセス、Paint/Queue/Dry Process)」と呼ぶことにする。

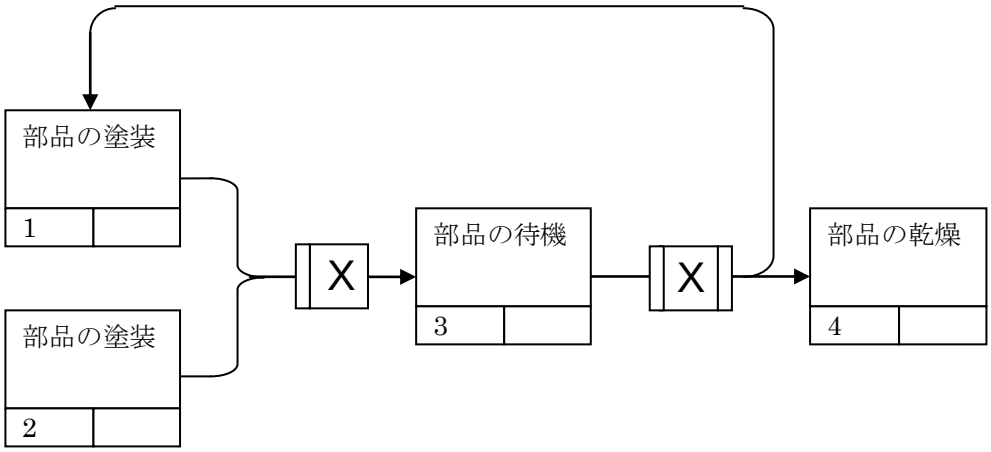


図 3-84　塗装／待機／乾燥　プロセス

スキーム図で指示された制約に加えて、図形言語では表現不可能なプロセス上の追加制約を、色々と作ることが可能である。例えば、以下のようなものを考えてみる。

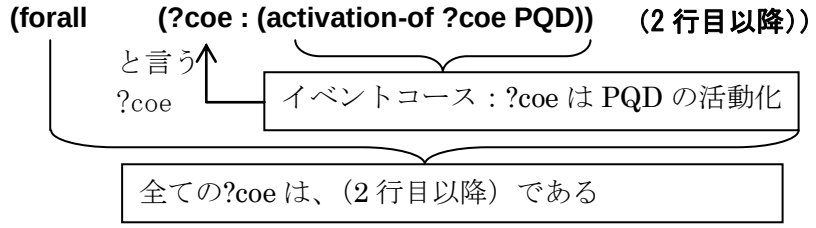
「PQD が発動した際、UOB「部品の塗装」がどのように発生していても、
厳密に 1 個の部品のみが塗装される」

この制約を詳細説明言語を用いて表現すると、以下のようになる：

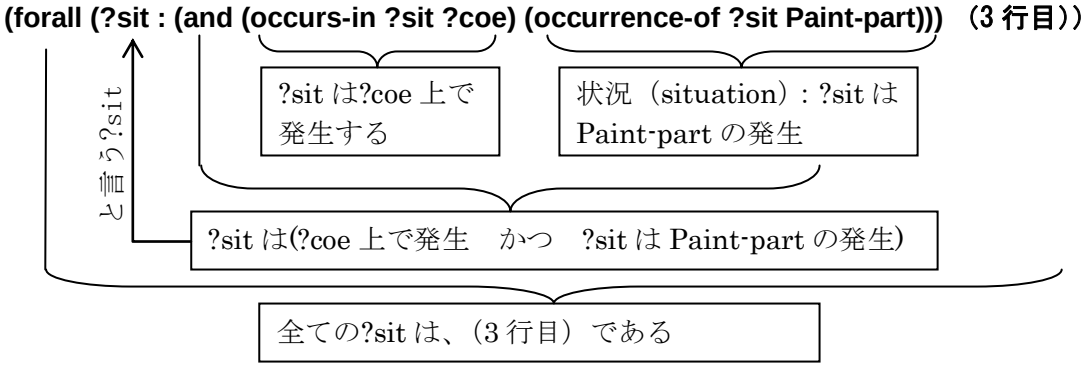
```
(forall       (?coe : (activation-of ?coe PQD))
    (forall (?sit : (and (occurs-in ?sit ?coe) (occurrence-of ?sit Paint-part)))
          (exist!-1 ?x (supports ?sit (painted ?x +)))))
```

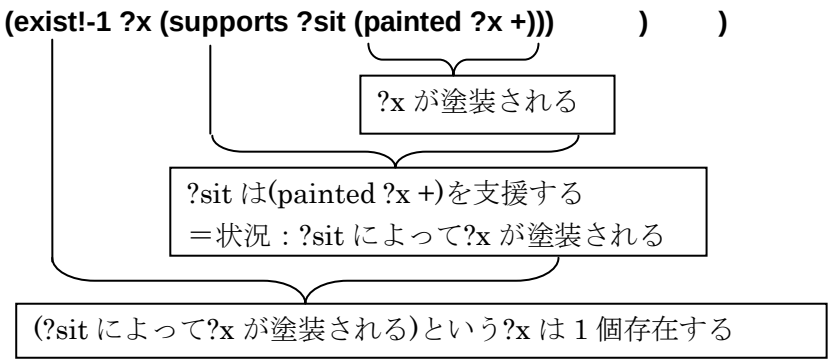
この制約文の中で、変数「?coe」は、PQD のような一般的なプロセスのイベントの過程

（cources-of events)、すなわち活動（activation）やインスタンス作成（instantiation）等の範囲に存在している変数である。この変数は、コロンの右にある　(activation-of ?coe PQD)　ーそうしたイベントの過程は、プロセス PQD が活動化したものであるー、という表現によって、更に制限されている。



そして、そうした全てのイベントコース c で、「部品の塗装」という UOB もしくは状況種類が発生するという状況（?sit）s の全てに対して、状況 s で塗装されるオブジェクト x（状況理論の言語では、x が塗装されるという情報を s が「支援（support)」している、という表現になる）が正確に 1 個存在する（「exists!-1」)、という事を、残りの 2 行で言っているのである。





注意すべきは、この制約が、その表現の通りに、ダイヤグラム上で図示されている PQD シナリオ全体に適用されている、ということである。しかし、対象となる PQD の発動の中の「部品の塗装」の発生の全てに対して適用することが意図されている事から、「部品の塗装」の特徴（characterization）に、この制約を直接付加する方が、より自然である。その為、制約の最初にある、一般的な全称量化（universally-quantified、1 行目と 2 行目にある **forall** の事）状態を省略する事が可能であり、次のように、一層単純で直接的に表現する事ができる。

```
(exist!-1 ?x (supports ?sit (painted ?x +)))
```

また、「部品の塗装」に課せられた制約としての状況変数「?sit」は、暗黙的な全称量化としてではなく、むしろオブジェクト変数「?x」と同じように、対象となる活動の中での「部品の塗装」の発生という役割を果たすパラメータとして、考えられているということである。（訳注：意味が良くわからない）

2 つ目の例では、以下の事が既に定義されていることを前提としている：

・「待機列にある（in-queue）」関係
　オブジェクトと待機列（queue）、期間（interval）の 3 つの間で、オブジェクトがその期間で待機列になっている場合にのみ、保持されている関係。

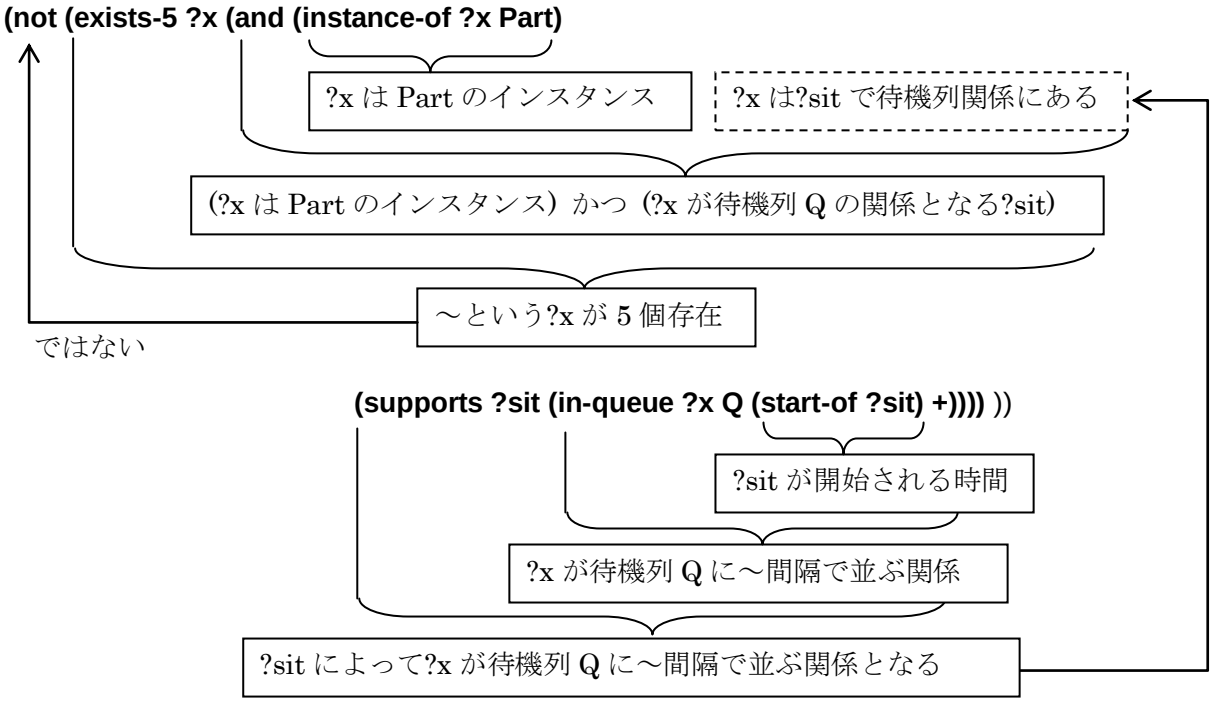
・「を開始する（start-of）」関数
　状況を、それが開始されるその時点（期間の変数）へと変換する関数

詳細説明言語は、複雑な定義も可能な強力な表現力を持っている。そこで、次のような制約を PQD に追加してみる：

「PQD が活動化された時点で、5 つのオブジェクトが待機列になっている場合、「部品の塗

装」のインスタンスは、どのような時でも開始されない」

(forall (?coe : (activation-of ?coe PQD))
(forall (?sit : (and (occurs-in ?sit ?coe) (occurrence-of ?sit Paint-part)))
(not (exists-5 ?x (and (instance-of ?x Part)
(supports ?sit (in-queue ?x Q (start-of ?sit) +))))))



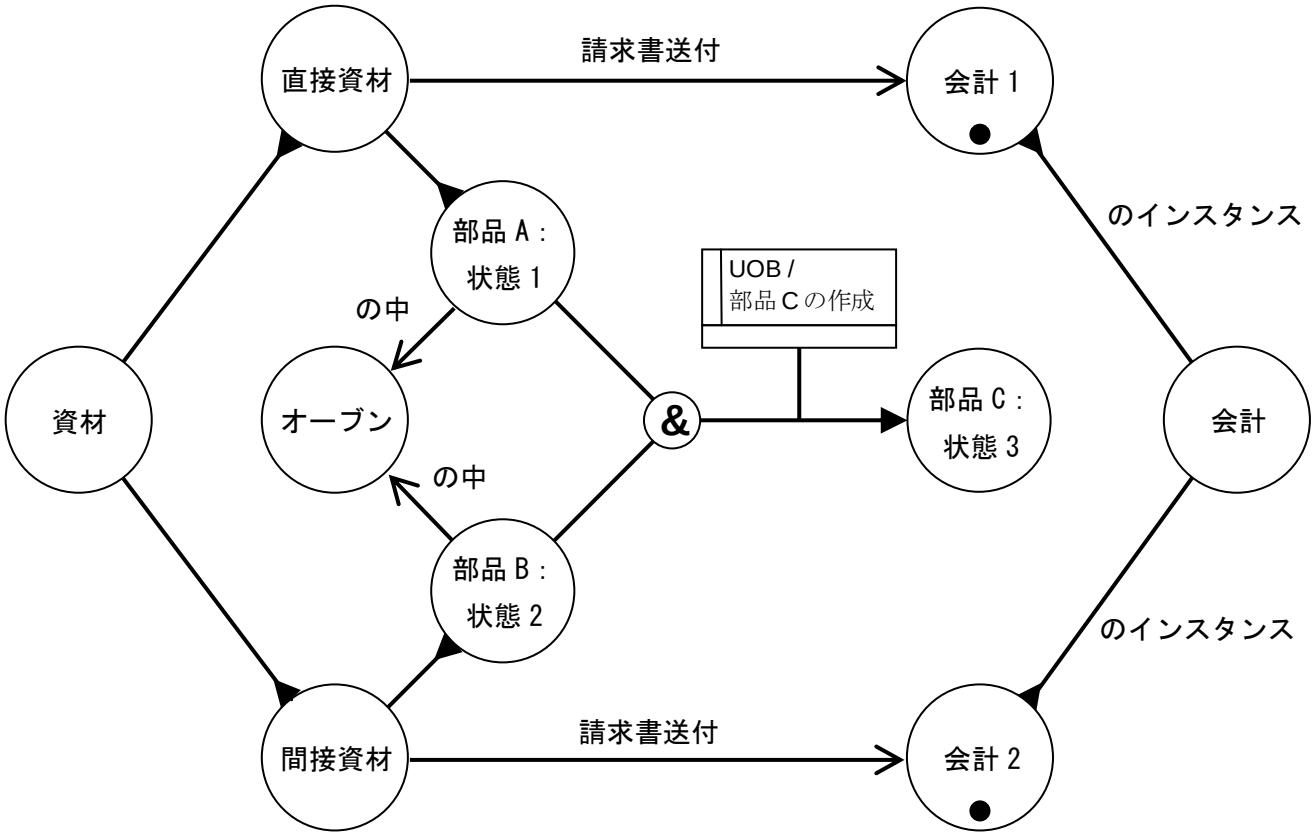
つまりは、PQD の活動の全てに対して、状況の開始時に、5 つ以上のオブジェクトが待機列になっていれば、その活動において「部品の塗装」という状況が発生しないのである。前の例と同様に、この制約は、PQD について一般的に表現された物であり、対象となる発動の中の「部品の塗装」の発生に適用し、「部品の塗装」の特徴に直接付加するのであれば、以下のように直接的に表現可能である。

(not (exists-5 ?x (and (instance-of ?x Part) (in ?x Q (start-of (interval-of ?sit))))))

３．４．３　オブジェクトスキーム図での詳細説明言語の使用例

オブジェクトスキーム図での詳細説明言語の使用例を説明する為に、図 3-83 の拡張変移スキーム図について考えてみる。

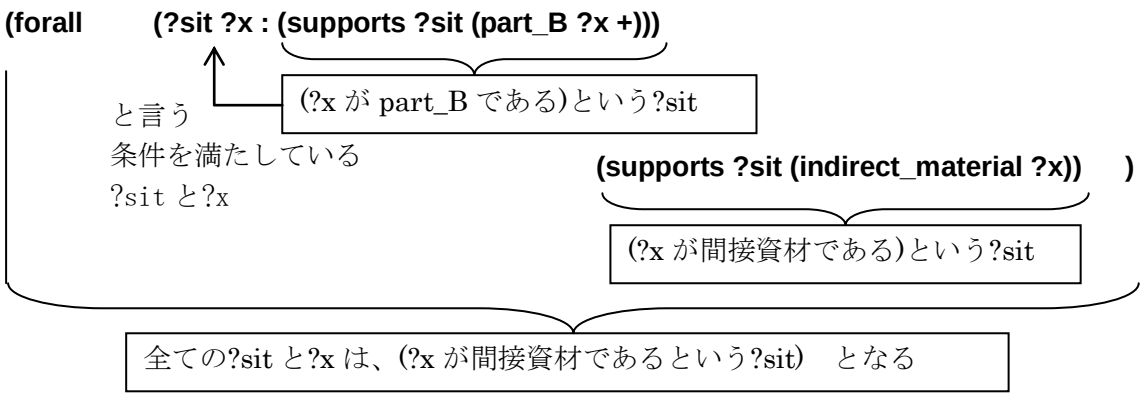
前にも説明したように、こうしたスキーム図では、周辺の文脈的情報に関係した意味論の不確実性が存在している。例えば、部品 B は「一般的に間接資材だと考えられているのか？」、それとも、「描写されたプロセスのような、より制限された文脈でのみ間接資材だと考慮されているのか？」という事は、このスキーム図だけでは判らない。そこで、こうした情報を、詳細説明言語を用いて明確にすることが可能である。



再掲：図 3-83　オブジェクト変移構造を持った別のオブジェクトスキーム図の例

そこで、スキーム図の詳細説明ドキュメントに、次のような制約を明確にしようとすると、次のような表現になる。

「部品 B は、いかなる状況においても、間接資材である」



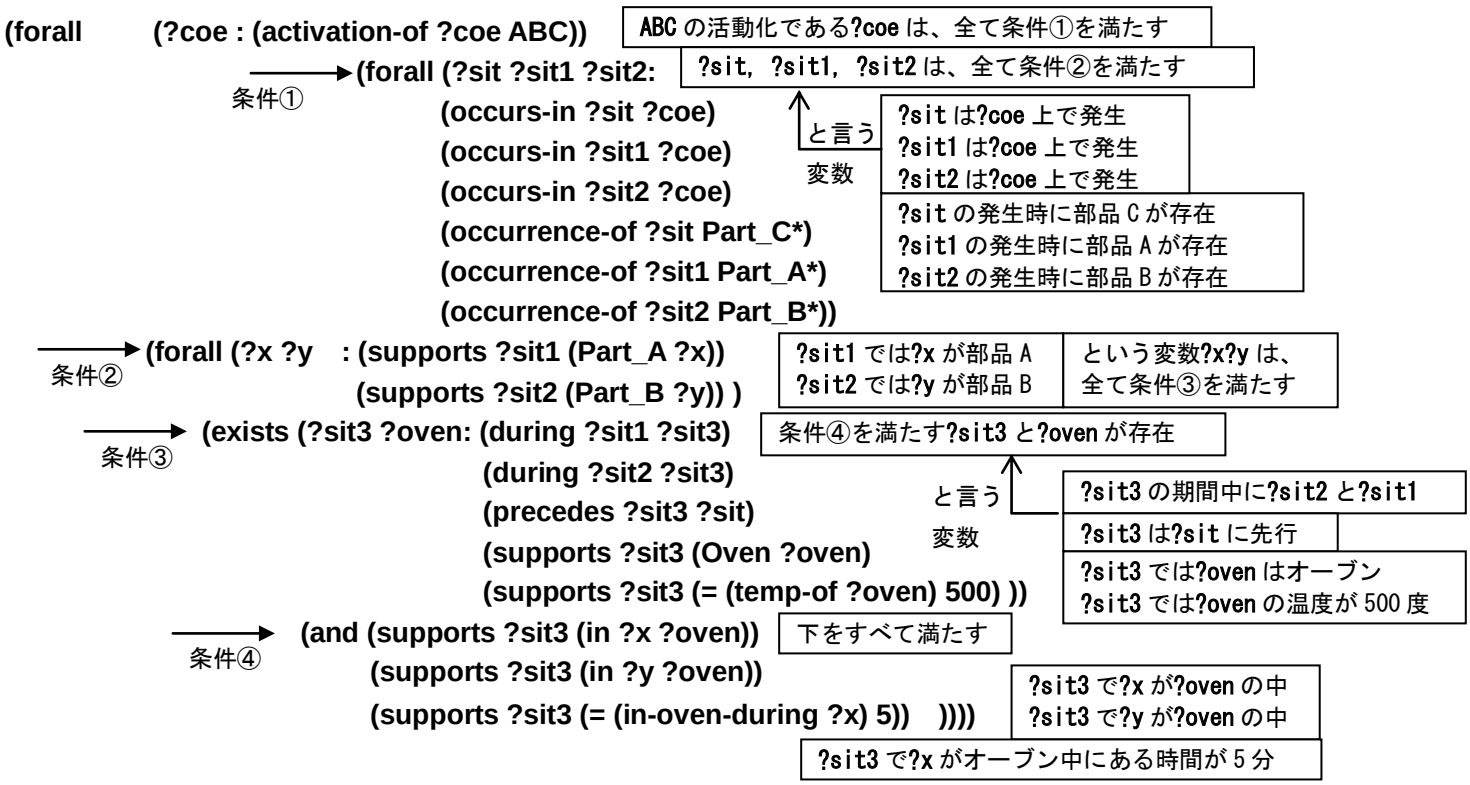
これにより、?x が部品 B であるという全ての状況において、同時に、?x が間接資材であるという状況になるのである。

前にも説明したように、変移スキーム図上のオブジェクト記号は、対象となっている状態（state）だけでなく、オブジェクトがその状態（state）となる状況の種類（type）も表現している。その為、拡張変移スキーム図は、オブジェクト記号の意味という点において多少曖昧になってしまっている。

例えば、図 3-83 で、はめ込まれている変移スキーム図（部品 A＋部品 B→部品 C の部分）の文脈では、「部品 A」記号は「部品 A が存在している状況の種類」を表している。ところが「直接資材」記号へとリンクされた記号という文脈においては、「部品 A は直接資材の下位種類である」ことが表現されているのである。そこで、こうした曖昧さを詳細説明言語で整理するために、「部品 A が存在している状況の種類」を意味する「Part_A*」という新しい用語を準備する。

この新しい用語を使う事により、以下のような制約の表現が可能となる：

「インスタンス ABC の部品 A と部品 B は、部品 C へと組み立てられる前に、500 度のオーブンの中で 5 分間置かれる」



訳注：上図では、元図の詳細言語の誤っていると思われる部分を幾らか修正している
（逆にその修正が間違っている可能性もあり）

これは即ち、ABC の発生 c においては常に、発生 c での?sit、?sit1、?sit2 がそれぞれ Part_C*、Part_A*、Part_B*の発生であり、かつ Part_A*と Part_B*で?x と?y がそれぞれ Part_A と Part_B であるとすれば、以下の条件を満たすオブジェクト?oven と状況?sit3 が存在する：

（1）?sit3 の期間中に?sit1 と?sit2 が発生する
（2）?sit の前に?sit3 が存在する
（3）?sit3 では?oven は温度が 500 度のオーブンであり、?x と?y はオーブンの中に在る

ということになる。

訳注　Part_A*：部品 A が存在しているという状況

３．５　注記（Notes）

注記（notes）は、UOB や交差点、オブジェクト、リンク、参照に取り付けられる。この注記により、IDEF3 分析者は以下のような事が可能になる：

（１）注記を付けた UOB や交差点と、個々のオブジェクトもしくは関係との関与を強調する

（２）参照されたデータやオブジェクトの具体例を調和させる（tie in）
（画面配置等（e.g., screen layout）？？？）

（３）対象となる交差点の詳細説明と関係する、特殊な制約セットを強調する。
注記により、交差点の詳細説明へと注意を向けたり、その内容をリスト化することが可能である。（追加の事実や制約、その交差点がどのように稼働するのかということを記述した決定ロジック等）

また注記は、個々の IDEF3 モデル要素に関する追加情報の提供や、記述への説明図やテキスト、画面配置、コメント等の取付けにも用いる事もできる。新規の IDEF3 ユーザーにとって、アイデアや概念を表現するには、交差点種類や、点線矢印、制約言語宣言文といった物よりも、注記の方が便利である。

図 3-85（次ページ）の例は、交差点と特殊な制約セットとの関連を、注記を用いてどのように強調できるのか、ということについて説明したものである。この注記により、ある状況において、UOB「作業範囲分析の実行」の繰り返しが要求されることを宣言している。この例では、交差点 J1 に付けられた注記は、参照「UOB／作業範囲分析の実行」が発動される条件を表現している。

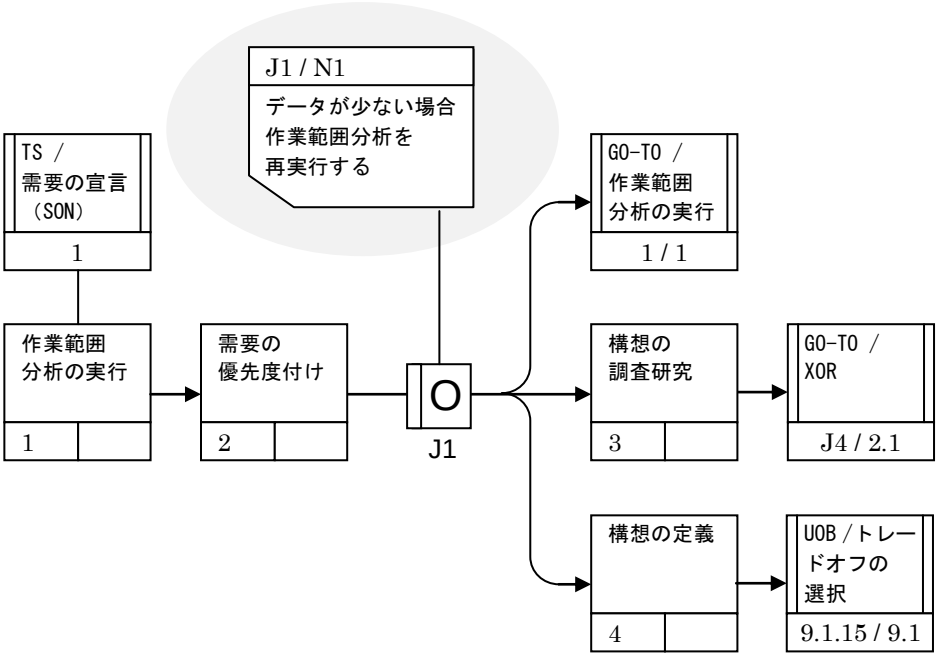


図 3-85　交差点に関連する注記

（訳注：元図では J1/N1 は UOB2 に付けられているが、誤りなので修正）

注記箱は 2 つの部分に分かれている。箱の上の左右に細長い部分は、注記の識別に使用されており、参照された要素番号と注記番号によって構成される注記識別番号が書かれている（J1 / N1 等）。注記箱の下側は、注記欄（note field）と呼ばれており、注記そのものを行う為に使われている。この欄の内容には公式な記述構成は存在していないが、プロジェクトの目的毎に、注記欄についての著作協定（authoring convention）を作ることが許可されている。例えば、「参照された 1 つの要素に関連する注記を、1 つの注記欄にまとて記述するようにし、個々の要素に 2 個以上の注記を付ける事を禁止する」、とか「注記欄での記述を、注記の種別（note classification）のようなものから書き始める構造にすることで、読者に対し、注記が何に焦点を置いているのかを強調する」といったものがある。

３．６　確率論的プロセスの表現（Representing Stochastic Process）

論理分岐のあるオブジェクトスキーム図を用いた表現を、マルコフ連鎖（Markov chains）等の確率論的プロセスによって置き換える事も可能である。こうした確率論的プロセスは、対象となるシステム S が成り得る複数の状態と、その中のある状態から別の状態（元の状態の場合も有る）へと変移する可能性のセットで構成されている。

システムは、一般に複合的な（complex）オブジェクトであることから、状態 A1 から他の状態への変移は、図 3-86 の、XOR 交差点から延びる変移リンクそれぞれに、変移の可能性を表現する実数を付けたスキーム図ような、排他的論理和（exclusive disjunction）の一種として表現するのが自然である。

状態 An 毎に、そうしたスキーム図を別々に準備しても良いのだが（訳注：図 3-86 のようなスキーム図を、n 個用意するということ）、それよりも、発生し得る全ての状態間の変移を収集し、その状態の間に線を双方向に直接引いた、より標準的な図形表現手法（訳注：図 3-87 のようなもの）を用いて、プロセス全体を表現する方が、より効率的で、読みやすくなる。

変移スキーム図の文法は、こうした表現を慣習として表現することを禁止していない。つまりは、図 3-86 のような変移スキーム図を全て集積し、それを省略形によって表現するのである。IDEF3 変移リンクを使用することで、対象となっている変移に影響を与えているプロセスの実際に記述を用いて、集積スキーム図を、増強する事が可能である。こうした慣習的表現の例を図 3-87 に図示する。読みやすくする為に、ここでは確率値は省いている。

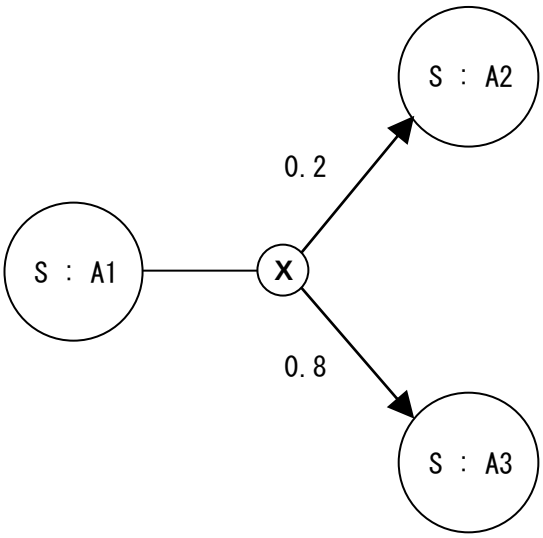


図 3-86　発生し得る複合的な状態変移論理を図示した変移スキーム図

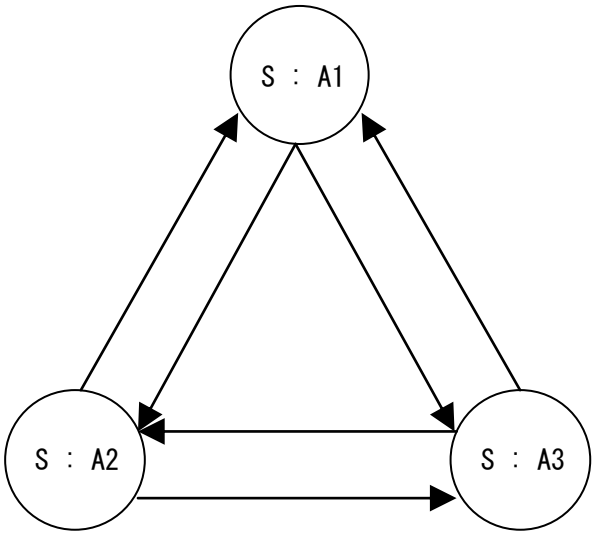


図 3-87　確率論的プロセスを表現した慣習的な変移スキーム図

第 4 章　IDEF3 記述の開発

この章では IDEF3 を、プロセス記述の獲得（capture）、統合（consolidation）、そして検証（validation）の手段として使用する為の手続（procedure）について説明して行く。ここで紹介する手続は、チームで行われる大掛かりな作業が必要なものを念頭に置いており、より狭い範囲で行われる小さなプロジェクトでは、ここで説明する物が全て必要となるわけではない。適用手続は、手法を使用する目的に大きく依存している事から、プロジェクトリーダーは、プロジェクトの開始時に詳細な手法適用ガイドを準備しておくことが推奨されている。

記述開発手続は、まずは IDEF3 記述を実行する過程で行われる進化サイクルの点から説明し、その後で、進行するプロジェクトに追従し、変化して行く機能的な記述について、それぞれ説明してゆくことにする。

４．１　IDEF3 記述進化サイクル（The IDEF3 Description Evolution Cycle）

（訳注：「サイクル」は、繰り返されるセット作業という意味でのサイクル）

IDEF3 記述の開発は、プロセススキーム図とオブジェクトスキーム図、そしてそれらと関連する詳細説明（elaborations）の作成によって行われる。この中の記述獲得と記述検証のプロセスは、非常に再帰的（recursive）で、かつ反復的（iterative）である。再帰的プロセスでは、プロセスを終了する基準、つまりは何時止めるべきかを知る事が重要となる。記述開発作業をどこで完了させるかという正確な基準を作る事は不可能であるので、幾つかの基本的なガイドラインを適用することになる。

最初に行われ、かつ最も重要である記述開発は、一般的に、何らかの目的を完遂する為に着手されたものである。この目的は、単にプロセスをドキュメント化するだけの事もあるが、この場合にはスキーム図と詳細説明の開発が完了する事が、終了の基準となる。

しかし殆どの場合、記述開発は、幾つかの発見活動もしくは決断活動を補助する目的で着手される。そうした状況では、記述開発に消費される時間と労力は、プロジェクトの情報需要によって左右されることになる。IDEF3 がプロセスのドキュメント化に使用される場合でも、発見・決断活動の補助に使用される場合でも、記述が完成して行くに従い、その構造や作業領域（scope）、詳細レベルで行われる変更の頻度が、次第に低減してゆく。

IDEF3 記述の開発は、対象となる組織の中で、活動がどのように実行されているかということに関する知識を獲得するプロセスである。こうした記述を収集し、体系化（organize）する為に IDEF3 を使用する場合には、一般的に以下の 5 つの段階（step）が再帰的に適用されている。

（１）収集（Collect）
プロセスの具体化（insatiations、実際に書き出してみる事）と、書き出されたプロセスの一般化を通して、観察と記述を行う。

（２）分類（Classify）
状況（situation）の種類（type）、オブジェクト、オブジェクト種類、オブジェクト状態（state）、そして関係を、個別化する（individuate）。

（３）体系化（Organize）　　組織化？
収取、分類されたデータを、IDEF3 構造を用いて組み立てる

（４）検証（Validate）
IDEF3 で行われた宣言（statements）の文法が正しいこと、そして収集された実際の状況や理想的な状況についての記述が、その宣言によって確実なものとなっている（corroborate）かどうかを、確かめる。

（５）改善（Refine）
新たに発見された情報の追加や、表現の簡素化、興味の対象となる重要な要素の強調を行うことで、既存の構造を調整する。

「再帰的に適用を行う」と言う事は、対象分野で利用可能な情報と知識を収集し尽し、それらを記述開発の終了条件を満足させた一つの構造へと組織化されるまで、同一の開発プロセスが繰り返される、ということを意味している。

４．２　IDEF3 記述獲得活動（IDEF3 Description Capture Activities）

IDEF3 の記述獲得を経験してみると、何度となく繰り返され、発見物に左右され（driven by finding）、そして作業の参加者によって様式化されることが多いことから、それが知識獲得（knowledge aquisition）や設計努力（design endeavor）に似ていることに気づくだろう。

この章で説明する活動は、連続した手順というよりも「思考法（modes of thought）」として考慮すべきものであり、ユーザーは記述獲得活動を、厳密に順序付けられた手法として適用できるなどと期待すべきではない。こうしたことを踏まえるならば、この章でこれから説明するフレームワークは、あくまでも IDEF3 の初心者の為のデフォルト構造なのである。

４．２．１　プロジェクトの定義（Define the Project）

プロジェクトが開始されたら、開発チームは可能な限り早く、記述獲得作業の目的（purpose）と背景（context）とを定めなければならない。

目的が宣言されることで、記述獲得作業の「完了基準（completion criteria）」が決定される。目的は、通常は以下の 3 つによって定められる：

（１）作業目標（objects）の宣言
（２）記述が満足すべき需要の宣言
（３）顧客が回答を希望する疑問（question）もしくは発見物（finding）

また、背景（context）が宣言されることで、プロジェクトが対象とする領域範囲が指定されることになる。この背景は、作業領域（scope）の宣言と記述獲得プロジェクトの初期シナリオの識別により確立される。

しかし、目的と背景が前もって完全に定義できることは稀である。顧客はデータ編集が開始された後でも、必要な発見物や質問のリストを頻繁に変更してしまうものであり、また分析者が回答であると考えていた部分が、実際には作業領域外だと考えていた別の部分

へ関係を持つ場合も多くある。一般的に目的と背景は、プロジェクト期間の殆どにおいて進化し続けるものである。

そして、このようにして決定された IDEF3 記述の目的と背景は、図 4-1 のような IDEF3 記述概要フォーム（Description Summary Form）へと書き込まれる。

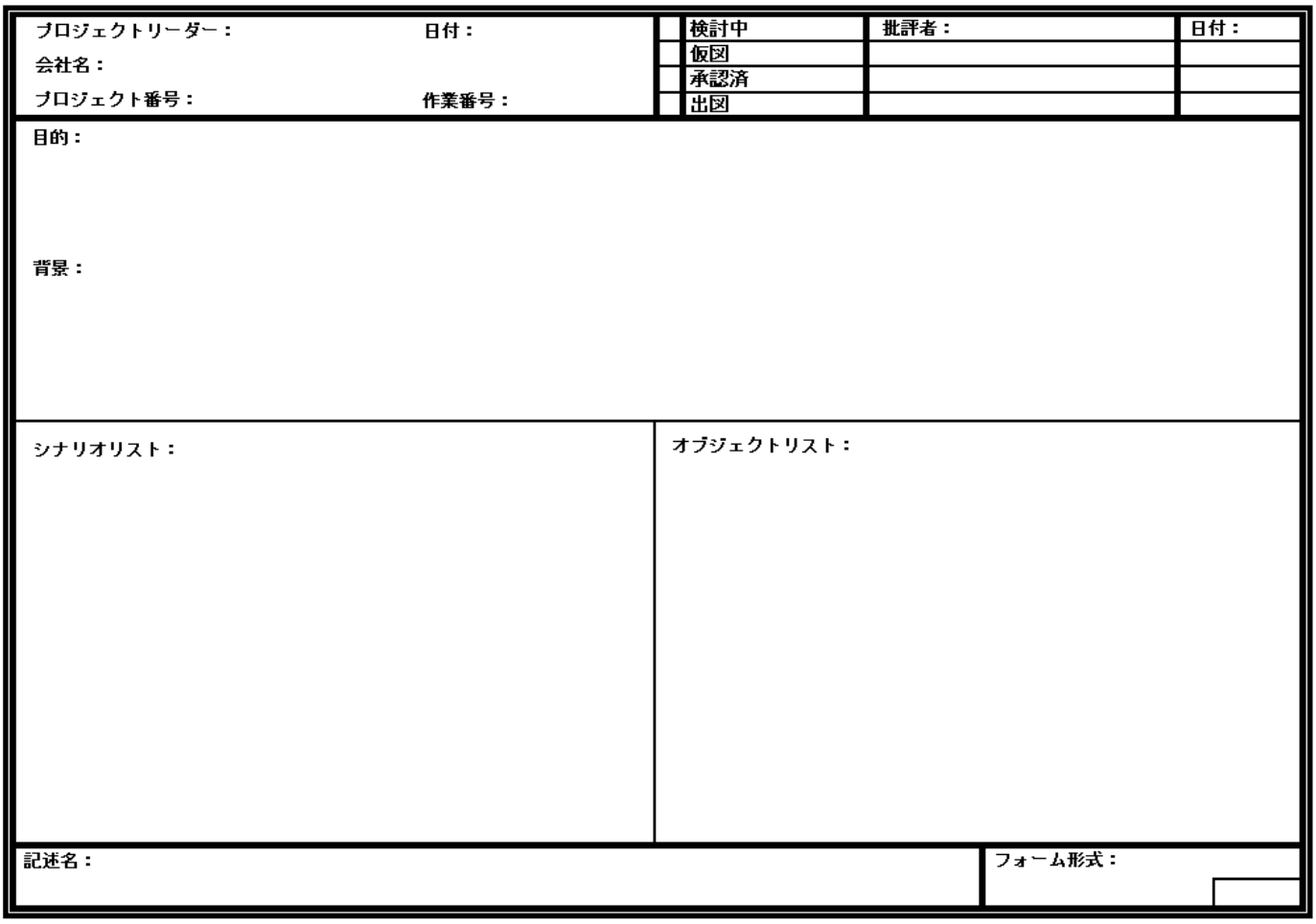


図 4-1　IDEF3 記述概要フォーム

４．２．２　目的の定義（Define the Purpose）

開発作業の初期段階において、目的の定義は重要な作業である。目的の宣言が行われなければ、作業に割り当てられた予算と時間だけが完了基準になってしまうことになる。

目標の定義は、大きく分けて 2 つの部分で構成されている。1 つは需要宣言（Needs Statement）、もう 1 つは、その記述された情報がどのように使用されるべきなのかという情報目標（information goal）の定義である。

需要宣言（Needs Statement）では、要求物の根源（source）（人やプロジェクト）の識別と、提示された顧客の目的の意訳（paraphrase）を行う。また情報目標（information goal）の識別は、次に挙げる質問に答えてゆく事で単純化できる：

（1）記述が利用可能となった時、誰がそれを使用するのか？
（2）顧客が回答を要望しているのは、どの疑問（question）か？
（3）どのような問題点（issue）が、プロセス記述の需要の裏に隠れているか？
（4）どのような決定が、プロセス記述の需要の裏に隠れているか？

４．２．３　背景の確立（Establish the Context）

作業目的の特徴が定まれば、作業領域（scope）の適用範囲（coverage）と詳細化レベルという点から、プロジェクトの背景（context）を定義する事が可能となる。

プロジェクトの背景の定義は、記述獲得作業の境界（boundary）を定義し、その境界を作業領域宣言（scope statement）のセットによってドキュメント化することから開始される。プロジェクトの作業領域の仕分け作業では、システムのどの部分を含め、どの分を排除するかが決定される。この際、顧客の需要と関係する作業領域のみを識別するのが理想的である。

プロジェクトの作業領域（scope）を定義する効果的なやり方（mechanism）は、考慮すべき重要な運営（operation）シナリオと、関係はしていてもプロジェクトの境界の外に出

てしまうシナリオとを、識別することである。シナリオの識別には、システム（組織）が遭遇する一般的に発生し得る状況もしくは問題の、題名と要約（paragraph description）を作成し、これを用いてチームメンバーの間で同意を取ることで行う。異なったシナリオが、実は、本質的には同一のプロセスを別々の視点から表現したものであることも多い。

また可能であれば､シナリオの中の開始 UOB と終了 UOB とを確立しておくべきである。更に、記述の背景の外に位置するにも関わらずシナリオに影響を与えたり、何かをもたらす活動を識別しておき、後で記述獲得作業の境界を改善する際に役立てるべきである。

目的と作業領域の宣言は、シナリオの識別を成功させるための便利なガイドラインではあるが、シナリオを実際に識別するには、その分野専門家の見識（insight）に依存せざるをえない。またプロジェクトリーダーは、この段階で識別されたシナリオは未だ仮の物であり、データの収集・分析の過程で変更される可能性が存在することに、注意しておかなければならない。

作業領域の定義と密接に関係しているのが、記述獲得作業の詳細レベルの決定である。要求される詳細レベルは、問題点の解決や、決断の実行、疑問の回答に、どれだけの詳細さが必要となるかを識別する事によって決定される。この詳細レベルの仕様は、通常は、例をセットにした形式でドキュメント化されている。

作業領域と詳細レベルの決定は、プロジェクトのこの段階においては仮の物であり、記述データが利用可能になれば、更新すべきである。プロジェクトリーダーは、獲得された記述データが、明確化された需要と情報目標に対して適正なものであるかどうかについて、定期的に評価を行うべきである。

４．３　データ収集の為の組織化（Organize for Data Collection）

初期のプロジェクト目標と背景とが決定されたら、データ収集の為の組織化作業を本格的に開始する事が可能となる。この時点で、プロジェクトチームの構成が固められ、チームメンバーの役割が確立し、シナリオ開発の責任所掌（responsibility）がチームメンバーに与えられる。

以下のものは、IDEF3 プロセスフローの記述獲得プロセスの関係者が果たすべき役割である：

（１）分析者（Analyst）
IDEF3 プロセスフロー記述の開発を主に行う、IDEF3 の専門家である。

（２）顧客（Client）
記述開発を依頼する、個人もしくは組織

（３）分野専門家（Domain Expert）
対象となる適用分野に関する知識を提供する人物

（４）主要仲介者（Primary Contact）
分析者と分野専門家との間を橋渡しする人物

（５）プロジェクトリーダー
記述開発作業全体の最高責任者

（６）批評者（Reviewer）
対象分野と IDEF3 手法の、どちらかもしくは両方の見識を持つ人物で、記述とドキュメントの下描き（draft）に対して、批評と承認を行う。IDEF3 スキーム図に批評文を記入する権限を持つ批評者は「コメンター（commentor）」、それ以外の批評者は「読者（reader）」と呼ばれている。チームメンバーだけでなく分野専門家も、批評者となることが可能である。（第 4 章を参照）

（７）司書（Librarian）
素材資料記録（source material log）やドキュメントファイルの維持管理、コピーの作成、IDEF3 キットの配布、記録の保持、等の責任所掌を担当する人物

（8）チームメンバー

IDEF3 プロセスフロー記述開発プロジェクトに関係する、全ての人物

チームメンバーに割り当てられる役割の中で重要となるのは、プロジェクト司書である。大掛かりなシステムの場合には、専門の司書が担当するが、より小規模な作業では、司書の役割は分析者が兼任する事が多い。この場合、プロジェクトリーダーはプロジェクトを通じて行われる、素材資料や IDEF3 キット、用語集、ファイル等の収集、カタログ化、管制、分配といった作業を、メンバーへと割当てることで司書の機能を確立してゆく。

それから、チーム作業を促進するために、外部情報源（external source、業界におけるプロセスベンチマークなど）からもたらされる参照モデルと参照資料をまとめる作業も、司書の責任所掌となる。

更に、分析者が専門分野（discipline）や工業分野（indutory sector）、会社（company）、会社内の部門（company segment）のそれぞれで使用される専門用語の独自の意味を理解するための用語集もまた、司書によって維持管理される。司書によって維持管理されている場合でも、分析者間で非公式に共有されている場合であっても、用語集はプロジェクトの進行とともに、増大し、改善されてゆく。

データ収集作業を組織化する上で中心となるのは、対象分野（domain）に関する知識と情報の主要な入手先（key source）を識別する作業である。主要仲介者やプロジェクトリーダー、そして分析者は、共同して専門家のインタビューリストを作成してゆく。このリストを作成する際、各専門分野に関する背景情報を主要仲介者から得ておくと便利である。作成される専門家についての情報には、専門家の責任所掌や現在の配員、そしてその専門家が経験や知識を持つ、対象分野に関係する他分野についての情報も含まれる。インタビューを行った専門家の名前、配属先、電話番号もまた、記録しておかなければならない。

データ収集作業を通して、これ以外にも様々な情報源（source）が発見、識別されることになる。新たな情報源となる可能性を持つものとして、運用指示書（operating instruction）や、手続マニュアル、従業員ハンドブック、基準書（regulation）、方針マニュアル（policy manual）、プロジェクトファイル、再利用可能な IDEF モデル、他の手法や技法によって得られたモデル、等が挙げられる。

プロジェクトリーダーは、こうしたチーム構造の組織化に加えて、チームの活動も組織化しなければならない。このプロセス記述獲得活動の組織化作業は、一般的な IDEF3 手続を、プロジェクト独自の需要に合わせた、より形式化された手法適用ガイドへと改造（cast）

することから始まる。この手法適用ガイドに従い、作業の需要に合うよう変形させた IDEF3 手法を、プロジェクトに合わせた方法で適用してゆく。この手法適用ガイドには、守るべきモデリングの慣習、分野専門家へのインタビューの概略標準、手法とツールのインターフェース仕様書、プロジェクトライブラリ利用手続、そして標準用語集、といった項目が含まれることになる。そしてこのガイドは、プロジェクト計画（plan）によって完成する。

一般的にプロジェクト計画は、明確な作業（task）と節点（milestone）、仮出図（intermediate deliverable）と最終出図、チームメンバーの配員、公式・非公式の報告形式、といった項目を用いて、踏むべき作業の段階を概説してくれる。

４．４　データの収集と分析（Collect and Analyze Data）

この段階に来て、ようやく実際のデータ獲得が行われる。チームが利用可能な主要な情報源は、分野専門家と、組織が持っている素材ドキュメント（source document）である。その際、分析者は、記述開発作業に関係するデータを効率よく獲得する為に、分野専門家と密接に協力しつつ作業を行わなければならない。

データ収集プロセスは繰り返し行われ、また個々のプロセスは相互作用をもたらす。予備（準備）データは、知識獲得作業の為のガイドラインを提供してくれる。分析者は、分野専門家とやり取りすることで、調査中のプロセスを書き下し、かつ言語化した初期記述を得る。活動の名前と、登場するオブジェクトの名前は、こうした初期記述から抽出されることになる。同じプロセスを異なった面を通して知っている複数の分野専門家に別々にインタビューを行ったり、分野専門家にフォローアップインタビューやマルチキット批評（multiple kit review、複数の異なるキットを用いてインタビューした内容を突き合わせて批評すること？）を行う必要がある事も多い。このようなプロセスを経て集められたデータは、完成した記述からでも分野専門家の観察を正確に反映したものであるという裏付けを簡単に取れるように、慎重に記録しておかなければならない。

４．４．１　インタビューの準備

IDEF3 手法では、データ収集の様式は明確に規定されていないが、分析者はインタビューを行う前に、仮の協議事項（agenda）と幾つかの明確な質問を準備しておくべきである。

分析者は、以下の 5 つの手短な概要を準備しておくことが望ましい：

（１）その専門家にインタビューを行う目的
（２）インタビューの対象とすべき項目
（３）求めている情報の種類
（４）インタビューを申し込む先の責任者
（５）議論を促進する質問

大規模なプロジェクトでは、プロジェクトリーダーはインタビューの準備ガイドラインと、標準的インタビュー計画シート、質問のひな形、用語集といった手法適用ガイドの標準を、より公式なものにしておくことが望ましい。

インタビューの準備に寄与する活動は幾つもあるが、その内のどれを実行するかは分析者の裁量に委ねられており、プロジェクト毎の需要や、関係する制約を見つつ選択してゆく。一般的には、以下に挙げるような準備活動の後に、インタビューが行われている：

（１）インタビューの日程調整と、必要な後方支援（logistics）の準備
　（訳注：logistics は軍隊における「兵站」の意味であるが、日本では軍隊以外でこの言葉を用いる事が無い為、近くて意味の通りそうな「後方支援」としておく。相手方の責任者との折衝、根回し、といったことか）
（２）インタビューの目標の確立
（３）候補となる質問の準備
（４）インタビューを受ける人が抱くだろう疑問と懸念を予め想定しておき、
　その懸念を解決できるように準備しておく

この時、追加の準備作業を行っておくことも望ましい。例えば、分析者は、顧客の公的なシステムやプロセスについて記述されたドキュメントを予め収集・分析し、それを基にして、議論の基点となる IDEF3 記述を準備しておいた方が良い。他にも、分析者は、類似システムのベンチマークモデルを使用することで、分野専門家と共に類似点と相違点を互いに識別し合える機会を持つようにすべきである。

インタビューを行う専門家のリストが出来上がったら、インタビュー日程の作成を行う。分野専門家との日程の交渉は、通常は主要仲介者を通して行われる。主要仲介者を介した場合でも直接交渉した場合でも、分析者はインタビューの時間と期間とが、インタビューを行う対象者とその上司の都合に合うように調整すべきである。

インタビューを行うに適した場所の準備と、必要となる物品の手配といった追加的な後方支援（logistics）の考慮も、インタビューを成功させるためには重要である。また分析者は、専門家的な見た目を装うことによってインタビュー相手（interviewee）の緊張をほぐす為、インタビューを行う際にどういう服装をするか、ということも考慮しておくのが一般的である。

インタビューの目標（複数可）を事前に確立しておくべきである。分析者が、何故このインタビューが行われるのか、少なくともどの情報を分野専門家から引き出したいかを明言することにより、インタビュー目標の確立が行われる。インタビューでの質問の流れ（interview line of questioning）に対して、一般的な方向性を与えられるように、目標を可能な限り簡潔なものにしておきたい場合には、目標宣言文を作成すると便利であることが多い。

インタビューの目標が確立したら、次に候補となる質問（candidate quetions）をまとめて行く。候補となる質問は、一度書出し、それらを筋が通るような順番へと構成して行くべきである。分析者は実績と経験を積むことにより、明確で、インタビュー相手の教育レベルや文化的背景にあった言葉や言い回しを使用し、答へ導くと言うよりも答えを請うような質問を作成する事ができるようになる。分析者は、準備過剰とならないように慎重であるべきだが、質問の書出しや、質問の構成のやり方について分析を行う事により、インタビュー技能が向上するだろう。一方で、この活動に費やす時間は、まとめられた質問が実際に使用される可能性に見合う、バランスの取れたものでなければならない。想定していなかった新しい情報がインタビュー中に発見された場合、準備していた質問は中止しなければならない。さもければインタビューは、事前に準備していない議論にはまり込んでしまう事になる。

準備項目が見過されることも多い。分析者は、インタビュー相手に対して、何故インタビューされているのか、インタビューで提供する情報によって何が行われるのか、そしてインタビューの見返りとして何が得られるのか、といったことをインタビュー相手自身が理解するために必要な情報を提供する必要がある。全てのインタビュー、特に最初のイン

タビューでは、分析者が自分自身に必要な情報の収集を急ぐ前に、そうした項目をインタビュー対象者が十分に理解できるようにすることから開始すべきである。以下に挙げるリストは､分析者が予め知っておくべき質問と配慮とを示したものである(Harrington, 1991)

（1）インタビューを行う目的は？

（2）インタビューの主催者は？

（3）他に誰をインタビューするのか？

（4）インタビュー相手は、どのように、誰に、選ばれたのか？

（5）インタビューで得られた情報はどのように使用されるのか？

（6）インタビュー相手は匿名にしたいのか？

（7）インタビュー相手の名前を、調査結果概要の中に出して良いのか？

（8）インタビュー相手はどのような意見（？feedback）を受けるのか？

（9）インタビュー相手は、このプロセスの結果に、どのように関係することになるのか？

（１０）インタビュー相手はどのような利益を得るのか？

（１１）インタビューとプロジェクトを成功させるために、特に詳細で正確な情報が
重要となる理由は？

（１２）重要なプロセスで、インタビュー相手がどのようにして主要な役割を
果たしているのか？

４．４．２　分野専門家へのインタビュー（Interview Domain Experts）

インタビューは、追加情報の収集や、以前に収集された情報の明確化、分野専門家によるIDEF3モデルの検証といった目的の為に､プロジェクトの期間全体を通して実行される。

専門家へのインタビューは、非常に重要である。分析者（インタビュー者）は、インタビューに前向きで友好的な雰囲気を醸成すべきである。インタビュー者は、専門家に対して、組織の為に、組織から要求された記述を作成することで問題を解決すべく、共に働いているという感覚を抱かせるように試みるべきである。分析者は常に、専門家はプロセスがどのように動くべきか、もしくは動いているかを知っている人である、ということを念頭に置くべきである。一般的に専門家は協力的であり、またインタビュー者が考えてもいなかった、調査の方向性や質問を提供してくれることが多い。

専門家が､「現在の」プロセスで使用されているドキュメントやフォームのコピーを提供してくることも多い。

この専門家から提供されたドキュメントは、実際にプロセスフローの概要を表わしていることもあるが、どちらかというと「するべき（Should-Be)」プロセスフローを示している。インタビュー者は、こうした実際に実行されているかどうか不明な公式ドキュメント上の手続よりも、実際に実行されているプロセスに、主要な焦点を当てるべきである。現在、プロセスがどのように実行されているかということに焦点を置く際には、分析者は、分野専門家の回答を偏ったものにしないために､「あるべき（To-Be)」システムについての会話を避けるように注意すべきである。

４．４．３　オブジェクト名の収集（Collect Names of Objects）

通常の環境下において、専門家がもたらしてくれる最初の情報の種類は、分野に関連したオブジェクトの名前である。インタビュー者は、こうしたオブジェクトを注意深く書き留めておくべきである。インタビュー後に行われる分析で、分析者（もしくはインタビュー者）は、得られたオブジェクト名をリストにまとめてゆく。こうして出来上がったリスト（オブジェクトプール（pool、蓄積された候補))は、分野に関係する UOB と分野のオブジェクトとを関連付けるために、後で再び分析されることになる。

４．４．４　活動名の収集（Collect Activity Names）

専門家からもたらされる活動の名前も、注意深く記録しておくべきである。こうして得られた活動名は、プロセススキーム図を構成する UOB の名前になる事が多い。活動の名前が収集されたら、活動の順序と構成についての観念を決定し、書き留めておくべきである。

そしてインタビュー後に行われる分析作業の中で、分析者（もしくはインタビュー者）は、得られた全ての活動名をリスト化してゆく。こうして作られた活動名リストは、IDEF3 スキーム図の為の潜在的な UOB の「プール（pool、蓄積された候補）」となる。

４．４．５　プロセス上の発生物に関連する事実と制約の収集
（Collect Facts and Constraints Related to Process Occurrences）

UOB とオブジェクトに関係する事実（fact）と、オブジェクトと事実もしくはオブジェクトと UOB の各間の制約関係、そして UOB 間に跨る事実も、インタビュー中に書き留めておくべきである。インタビュー中に焦点を置くべき情報の種類は、次の通りである：

（１）プロセスの開始（initiation）を統括（govern）する制約

（２）プロセスの期間中、保持されるべき状態

（３）プロセスの終了（termination）を知らせる状態

（４）プロセスの開始もしくは終了の切っ掛けとなるプロセス

（５）プロセスの発生（occurrence）についての特性（property）
（プロセスの発生期間、中断の可能性、等）

（６）エージェント（agent：発動者、行為者）、情報、リソース（resource：原材料、発生源、ある事に必要となる物）、もしくは生産物として、
プロセスに参加するオブジェクト

（７）オブジェクトの特性（例えば、入荷頻度や仕損品率といった、
　　個々のプロセスに関連する特性）

（８）単一プロセス内での、オブジェクト間の関係性（relations）、
　　もしくは関連性（associations）

（９）プロセス間のオブジェクトの関係性、もしくは制約（共有されたリソース等）

（１０）プロセスに参加するオブジェクトに関係して、満足していなければならない状態

（１１）プロセスの発生した際の、通常状態と例外状態との区別

以上のような情報セットのそれぞれが「事実（fact）」となる。

４．４．６　状況記述の収集（Collect Situation Description）

IDEF3 における状況記述（situation description）とは、プロセスの発生についての特徴（characterization）である。この特徴には、プロセスの発生期間中に明確な関係を持っているオブジェクトの集合と活動との関連性や、ある活動とその活動の前後に発生する他の活動との関連性も含まれている。

状況記述は、実行中のプロセスを観察することによって得られることが多い（ある部品が製造されている工場を訪問する等）。しかし、そうした直接的な観察からだけでは、短期間の状況の通常プロセスの情報しか得られない。その為、長期間の状況下での通常プロセスと、例外的なプロセスの双方についての特別な見識については、専門家に依存せざるを得ない。

分析者は、こうした状況記述の分析作業中、これまでに発見していたオブジェクト名リストと活動名リストに追加を行う。状況記述の分析作業を行う事で、活動の順序や事実のリスト、そして記述を行うプロセスと関連する制約に対して、必要となる見識が得られてゆくのである。

４．４．７　素材資材の収集とカタログの作成（Collect and Catalog Source Material）

必要に応じて、分析者は、分野専門家の記述に含まれているプロセスの情報物（information artifact、訳注：情報を含んだ人工物一般のこと、フォームやスクリーン、メール、スケッチ、各種ファイル等）を参照するように、読者に対して要請をすべきである。更には、こうした情報物のコピーを将来の分析の為に収集、保管しておくべきである。プロジェクトを通して収集された全てのデータは、図 4-2 で示すように、IDEF3 素材資料記録（Source Material Log）に記録しておくべきである。

使用先：	分析者：山田　太郎　　日付：1995/2/8 プロジェクト：サンプルプロジェクト 注記：　1 2 3 4 5 6 7 8 9 10　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

素材資料番号	素材資料名	収集先	収集者	収集日
SM1	購入請求書／フォーム PI-R6 4-72	一般購買担当者		
SM2	手続 #079-003 ／改正00 「請求書の作成」	一般購買担当者		
SM3	手続 #079-001 ／改正00 「調達要求書の作成」	社内方針・手続マニュアル		
SM4	手続 #101-506 「調達コード」	社内方針・手続マニュアル		
SM5	購買業務　商品コードリスト	一般購買担当者		
SM6	購買業務　製品コードリスト	一般購買担当者		
SM				
SM				
SM				
SM				
SM				

背景設定参照：	記述項目：	フォーム形式： 素材資料記録

図 4-2　素材資料記録

素材資料記録で参照されている個々の素材資料は、図 4-3 のような素材資料記述（Source Material Description）を用いて、より詳細な情報を記録しておく。

使用先：	分析者：山田　太郎　　日付：1995/2/8 プロジェクト：サンプルプロジェクト 注記：　1 2 3 4 5 6 7 8 9 10　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

素材資料番号 SM	素材資料名： 補助： コメント： 概要：
素材資料番号 SM	素材資料名： 補助： コメント： 概要：

背景設定参照：	記述項目：	フォーム形式： 素材資料記述

図 4-3　素材資料記述フォーム

素材資料記述フォーム上の個々の登録欄（entry）は、素材資料番号と素材資料名によって、その欄がどの素材資料について説明しているのかということを識別できるようになっている。こうした仕組みにすることで、記述開発チームのメンバーが、候補となるプロセス記述要素（candidate process description element）を個性化する際に、それぞれの要素の素材資料への追跡が可能となるのである。フォームに追加された欄には、以下の物がある：

（1）補助（Supports）

この欄には、この素材資料が補助している IDEF3 要素番号（UOB 番号やオブジェクト状態番号等）が記入されており、記述要素からその要素の素材資料への追跡性を上げている。

（2）コメント

この欄は、後日その資料がカタログ化される際に、参照しておくべき特殊な特性もしくはコメントを記録する為に使用される。

（3）概要（Abstruct）

素材資料内で議論されている主要な概念についての概要を記入する。

４．４．８　収集されたデータの分析

データ収集を終えたら、インタビューの記録を分析し、素材資料を調査し、素材資料記録を作成する。そして最初の発見物を、プール（pool）と呼ばれるリストへとカタログ化しておく。IDEF3 では（１）オブジェクトプール、（２）シナリオプール、（３）UOB プール、（４）オブジェクト状態プールの、4 種類のプールが使用されている。図 4-4 は、オブジェクトプールの例である。それ以外のプールも、図 4-4 と基本的には同じ構成をしている。

使用先：	分析者： 山田　太郎　　日付：1995/2/8 プロジェクト：サンプルプロジェクト 注記：　1 2 3 4 5 6 7 8 9 10　　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

オブジェクト		素材資料番号	オブジェクト		素材資料番号
ID No.	名称		ID No.	名称	
01	調達請求書	SM1	018	資材表	SM38
02	購買担当者	SM2	019	ルートシート	SM40
03	販売業者	SM3	020	発送先	SM41
04	調達要求	SM15	021	受取人	SM43
05	発送	SM6			
06	要求元	SM11			
07	部門	SM12			
08	パターン	SM21			
09	部品	SM26			
010	調達請求項目	SM23			
011	商品	SM30			
012	調達請求項目	SM31			
013	作業	SM34			
014	会計	SM36			
015	製品	SM37			
016	資材表ページ	SM38			

背景設定参照：	記述項目：	フォーム様式： オブジェクトプール

図 4-4　IDEF3 のオブジェクトプールの例

４．４．９　必要に応じた追加情報の取得
（Acquire Additional Information as Required）

収集したデータの分析を行ったり、IDEF3 スキーム図を初めて構築する際に、追加データが必要であることが判明する事が多い。追加情報の収集アプローチには以下のようなものがある：

（１）問題を解決し、追加の素材資料を識別する為のフォローアップインタビューを行う

（２）キットの評価プロセスを使用する（訳注：キットについては後ほど説明）

（３）問題のあるプロセスやシナリオを、直接観察できる機会を手配する

（４）新しい焦点で分析をやり直し、素材資料を改定する

（５）促進ワークショップ（facilitated workshop）を開催する

どのアプローチを用いるのか、どのアプローチを組み合わせるのかについては、必要となる情報の本質と、IDEF3 が使用される目的の両方を考慮して決定する。ワークショップは、グループでのブレインストーミングにより「あるべき（To-Be）」プロセス設計のアイデアを出し、グループ内でどのアイデアを選択するかについての同意形成を行うような場合にのみ、開催される。

４．５　IDEF3 スキーム図をまとめる（Formulate IDEF3 Schematics）

これまでに説明してきた 2 種類の IDEF3 スキーム図には、プロセス記述情報の収集と表現を行う機能（mechanism）が備わっている。

プロセススキーム図では、対象となるシナリオを、プロセスを中心とした視界（view）から見る事が可能である。またオブジェクトスキーム図では、単体、もしくは複数のシナリオを、オブジェクトを中心とした視界から見ることが可能である。どちらのスキーム図も、分野専門家の記述を図形的に投影したものである。

IDEF3 スキーム図を構築して行く上で重要な事は、スキーム図の開発を行う際に、精度不足は論外としても、理想的で検証可能な状態を満足しなければならないという制約に囚われないことである。例えば、開発の初期段階のプロセススキーム図のフローで辻褄が合わないことは、当然なことである。プロセススキーム図の描き始めは、接続が殆ど無い UOB 箱が描かれているだけの事も多い。その理由は、この時点では全体図がまだ獲得できていないからである。記述とは結局のところ、分野専門家が持つ知識や経験の実体の中に存在する何らしらかの物についての、事実や信念を記録したものなのである。こうした記述のようなものは、一般的に不完全になる。なぜなら、記述を作成している人は、その人が関係していないと考えた部分や、システムを記述する過程で忘れてしまった部分、知識を持ち合わせていない部分については、事実を省略してしまうからである。信じられないかもしれないが、誰一人として理解していない、もしくは知りもしない要素を内在しながら、稼働しているシステムが数多く存在している。

本文注釈：

分野専門家が、何らかの活動やオブジェクト、関係が存在している事を知り、そして彼らがそれが何であるのかを知っているのであれば、IDEF3 スキーム図要素を使用して、その知識を表現する事は簡単である。また分野専門家が、その中のある要素が存在しない事を知っていた場合でも、事実を簡単に獲得することが可能である。分野専門家が知らない物が含まれる知識を明確にドキュメント化する為に、例えば「UOB 間に引かれた波線の矢印は、その UOB の間に、良くわからない UOB の集合が存在していることを示している」とか、雲形の宣言は、「良くわからないものの、とにかく宣言が存在しているということを示している」といった、特別な文法慣習（syntactic convention）を作ることで対応することも可能である（ソクラテス派の情報とプラトン派の情報の違い、等（訳注：意味不明））。ただし、そうした慣習は、使用されるとしても記述開発の初期段階でしか使用すべきではない。

あるスキーム図が、非接続の要素を含んでいるという事実を、過大に懸念すべきではない。幾つものスキーム図にギャップが残ったままで、獲得プロジェクトが成功裏に終わることも珍しくない。こうしたことは、プロジェクトの目標を達成する為に、スキーム図上に存在するギャップの補填が不要である場合に発生し得る。

更に、現在の環境記述を獲得する目的で IDEF3 を使用している場合には、IDEF3 ユーザーはシステムをデザインしているのではなく、むしろシステムの挙動に関する既知の事実をまとめているのであり、その結果として出来上がった記述は、モデルを作成する際の素材に用いられるだけである。このように、IDEF3 記述は、プロセス知識を獲得し保存する為の、精密で上手く構成された道具（mechanism）として利用可能なだけでなく、並行な複数の理想形や、システム挙動のシミュレートや予測に使用するモデルの構築に、再利用することも可能なのである。

４．５．１　プロセススキーム図をまとめる（Formulate Process Schematics）

プロセススキーム図は、プロセスを「プロセス中心」視界で見たものである。このスキーム図は、プロセスとシナリオの背景にある一時的関係や、普遍的関係、そして論理的関係に焦点を当て、プロセス知識をまとめたものである。

プロセススキーム図の構築は、次のような段階を経て行われる：

（１）UOB の識別

（２）UOB を適切なシナリオと関連付ける

（３）シナリオ上での、UOB の組合せ間に存在する順序的制約を識別し、
最初のスキーム図を作図する

（４）論理的記述の箇所に、交差点を加える

（５）必要なら、制約順序リンクを加える。

（６）必要なら、UOB や交差点、リンクの、詳細説明（elaboration）を作成する

（７）選択された UOB を分解する

（８）関係リンクを追加し、追加された関係性を強調する

４．５．１．１　UOB の識別

最初のデータ収集活動が完了したら、分析者は対象オブジェクトや活動、事実、そして制約のリストを作成すべきである。このデータと状況記述を使用し、分析者は UOB の識別を行い、そして IDEF3 スキーム図の一般構造の組織化を開始する。

この作業の為の最初の準備を行うと同時に、プロジェクトの背景を確立するが、それまでに、分析者は対象のシナリオの識別しておかなければならない。識別された 1 つのシナリオに対して、少なくとも 1 つのプロセススキーム図を作成するのが一般的である。プロセススキーム図の背景として使用されたシナリオは、分析者が UOB 候補を探索する際の作業領域（scope）となる。

UOB の識別とは、IDEF3 の方法論（paradigm）の知識と分野専門家から収集した事実とを使用して、UOB の候補セットの識別と改良を行うという、知力による分類活動である。分析者は、このプロセスを実行する際に、人がプロセスを記述する（自然な）やり方に合わせると便利である。組織や複雑なシステム内で「何が進行しているのか」という記述を獲得するには、幾つもの自然言語概念に言及する必要がある。以下に挙げる概念は、「この世界で発生している何か」を記述する為に使用されている日常的な用語（everyday language）である。

１．機能（Function）	４．活動（Activity）	７．行動（Action）
２．プロセス（Process）	５．運営（Operation）	８．イベント（Event）
３．シナリオ（Scenario）	６．決定（Decision）	９．手続（Procedure）

こうした概念はそれぞれ、何らかの境界線を引く挙動（circumscribed behavior）を伴っている。例えば、「立案活動」、「内製か購買の決定」、「契約締結式」といった UOB への参照は、世界を時空的な塊へと切り分けることにより、残りの世界から分離されたその塊の中で「何が進行しているのか」という記述を行っているのである。IDEF3 では、一般的な情報単位（packet、情報伝達を行う際の情報通信単位）、もしくは UOB は、上に挙げたような概念をカプセル化（encapsulate、内包し外から直接見えないようにしている）している。

訳注：よくわからない

UOB 候補を探す際に、プロジェクト定義段階で作成されたシナリオ候補のリストを、最初に参照すると良い。例えば、プロジェクトの作業領域設定（scope）に使用された候補シナリオの初期セットを分析することで、分析者は、シナリオが「1 本に繋がって」いないか、リストにある他のシナリオ候補によって包摂されていないのか、ということを確認するのである（本文注：リストにあるシナリオ候補の幾つかは、同一の状況種類の代替名でしかない場合もある）。

最初に作成した事実セットもまた、UOB 候補の有用な捜索先である。UOB 候補の識別は、分野専門家の記述の中から、命名されたプロセス（購入プロセス、等）や、命令形の動詞（資金の工面（budget funds）、等）、動名詞（運用上の需要の識別、等）を捜索する事から始まる。この分析プロセスを通して、追加シナリオと UOB 候補の両方が識別されることになる。より一般化する傾向を示すシナリオとは異なり、UOB はより特殊化する傾向にあり、更に明確な時間的制限を受けている。つまり強力な因果的依存性か、時系列的依存性が識別されたならば、シナリオではなく UOB が識別された可能性があるということである。その為、「国家安全保障政策の転換に『呼応して』、それまでに割り振られた作業の再定義が要求される」という句からは、「呼応して（in response to）」という因果関係に立脚した、2 つの UOB 候補（「割り振られた作業の再定義」と「国家安全保障政策の改正」）が生成されることになる。

擬物化（objectification、対象化、客観化）と換喩法（metonymy：訳注「食卓」、「霞が関」、「手が足らない」、等）を拡張使用することで、名前を使わずにプロセスを記述するこ

ともできる。つまり、殆どのプロセスは、殆どのオブジェクトと同じように、名前を持っていないのである。

プロセスの擬物化とは、単に、あるプロセスをオブジェクトとして記述する、ということを意味している。例えば、「兵器システムを取得する（Acquire Weapon System）」プロセスは、議員からは「兵器システムの取得（weapon system acquisition）」、もしくは単純に「システム取得（system acquisition）」と呼ばれている。

換喩法は、その物を表わす限定的な記述子（descriptor）として記述されている物と関連する、重要なオブジェクトや関係性（に属する、等）の名前を使用した、表現形式の一つである（訳注：「食卓」という言葉は、ただのテーブルそのものを表現するだけでなく、換喩法においては「食事をすること」や「家族の団らん」といったものを意味している。食事をする場所、家族が団らんする場所は他にも存在し得るが、その代表的なものとして、「食卓」という言葉が用いられている）。例えばプロセスは、そのプロセスの主要な生産物の名前で呼ばれていることが多い（需要決定ミッション、マイルストーン１決定、等）。

こうした表現形式を認識しておくことで、分析者は、UOB 候補の識別と、それらの適切なシナリオとの関連付け、そして初期プロセススキーム図の構築プロセスの作成といった作業を、簡単に実行することが可能となる。

４．５．１．２　最初のプロセススキーム図の作成（Layout Initial Process Schematic）

最初に作成されるプロセススキーム図は、分析者が専門家から収集した情報をどのように理解しているのかを表現する為に作成される。この最初のスキーム図を使用し、分析者は分野専門家と共に批評を行い、この記述が正しい事を検証して行く。この最初のスキーム図はまた、分野専門家に更なる経験を呼び起こさせる手助けにもなる。最初のデータの収集プロセスは、分野専門家が自身が内部に持つ知識をどれだけ呼び起こせるか、ということに大きく依存している。

専門家から適度な精度・完成度の記述を獲得するには、分析者のスキーム図が分野専門家の知識に合致するまで作図プロセスを繰り返す必要がある。場合によっては、分析者が作成した下描きの記述を、後から分野専門家が批評するのではなく、分析者と分野専門家

が共同で記述を作成しても良い。共同で作成することにより開発時間が短縮され、最初からより完全に近い記述を作成する事が可能となる。

プロセススキーム図概略フォーム（図 4-5）は、分野専門家との批評活動の調整や、生データからのプロセススキーム図の作成を行う際に、分析者を補助してくれる。テキストによる記述（textual description）やプロセススキーム図の用語集も、このフォームに含まれている（is part of)。このフォーム上のテキスト情報には、スキーム図の目的が含まれるべきであり、また他の欄にすぐさま当て嵌められない情報をここに記入しておく事も出来る。分析者は、テキストによる記述の他にも、スキーム図で参照している UOB や他の IDEF3 要素（UOB、シナリオ、変移スキーム図）を記録できる。このフォームを完成させることは、プロセススキーム図構築に関連する分析活動の一部である（is part of)。

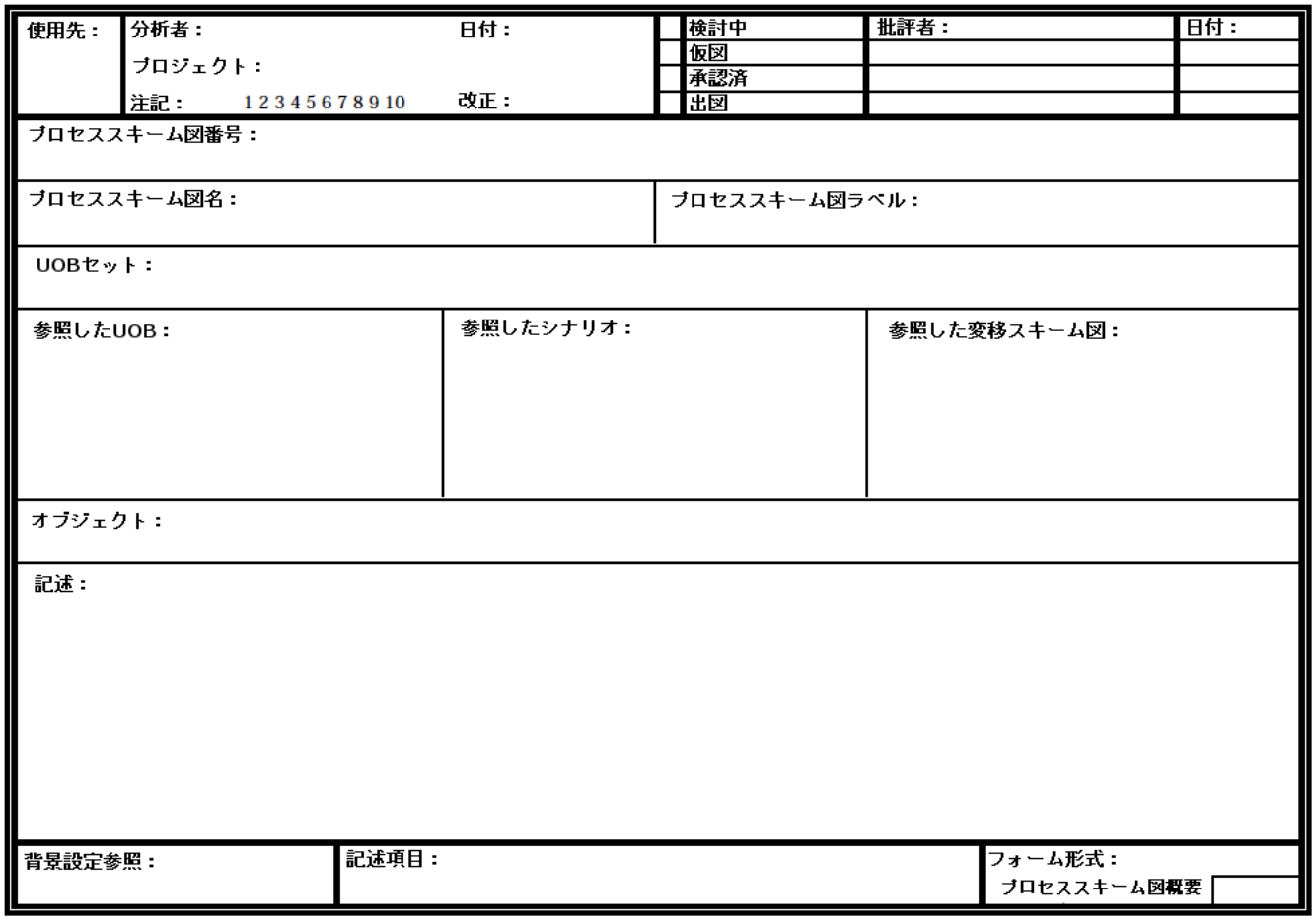


図 4-5　プロセススキーム図概要のフォーム

４．５．１．３　UOB、交差点、リンクについての詳細説明の作成
（Develop Elaborations for UOBs, Junctions, and Links）

IDEF3 プロセススキーム図は、プロセス内で発生している UOB の見地から図形的にプロセスを記述したものであり、個々の UOB は、それぞれが 1 個の UOB 箱によって表現されている。しかし、スキーム図上に描かれた一連の UOB 箱を流し読みしただけでは、記述されているプロセスの全体像を理解することは出来ない。そこで詳細説明（elaboration）を用いて、スキーム図内の IDEF3 要素の詳細な特徴を表現する。この詳細説明フォームから、専門家の記述の最も詳細な特徴を読み取ることができる。IDEF3 要素の詳細説明は、殆どの目的に対しては自然言語で十分であるが、自然言語を用いた詳細説明よりも構造的で精密であることを要求するものであれば、IDEF3 詳細説明言語（付録 A を参照）を使用することもできる。

UOB や交差点、順序リンクの詳細説明は、インタビューデータから作成され、そして記述が表現対象となる知識を持つ分野専門家により批評される。初めの内は、こうした詳細説明は単純な用語集の項目のように見えるが、データ分析が進むに従って、詳細説明はより構造的で、より簡潔（concise）なものへと変化して行く。詳細説明に含まれている典型的な情報は、関係するオブジェクトとその役割、対象となる事実、そして制約がある。こうした自然言語による詳細説明は、詳細説明フォームへと記入される（図 4-6 と図 4-9 を参照）

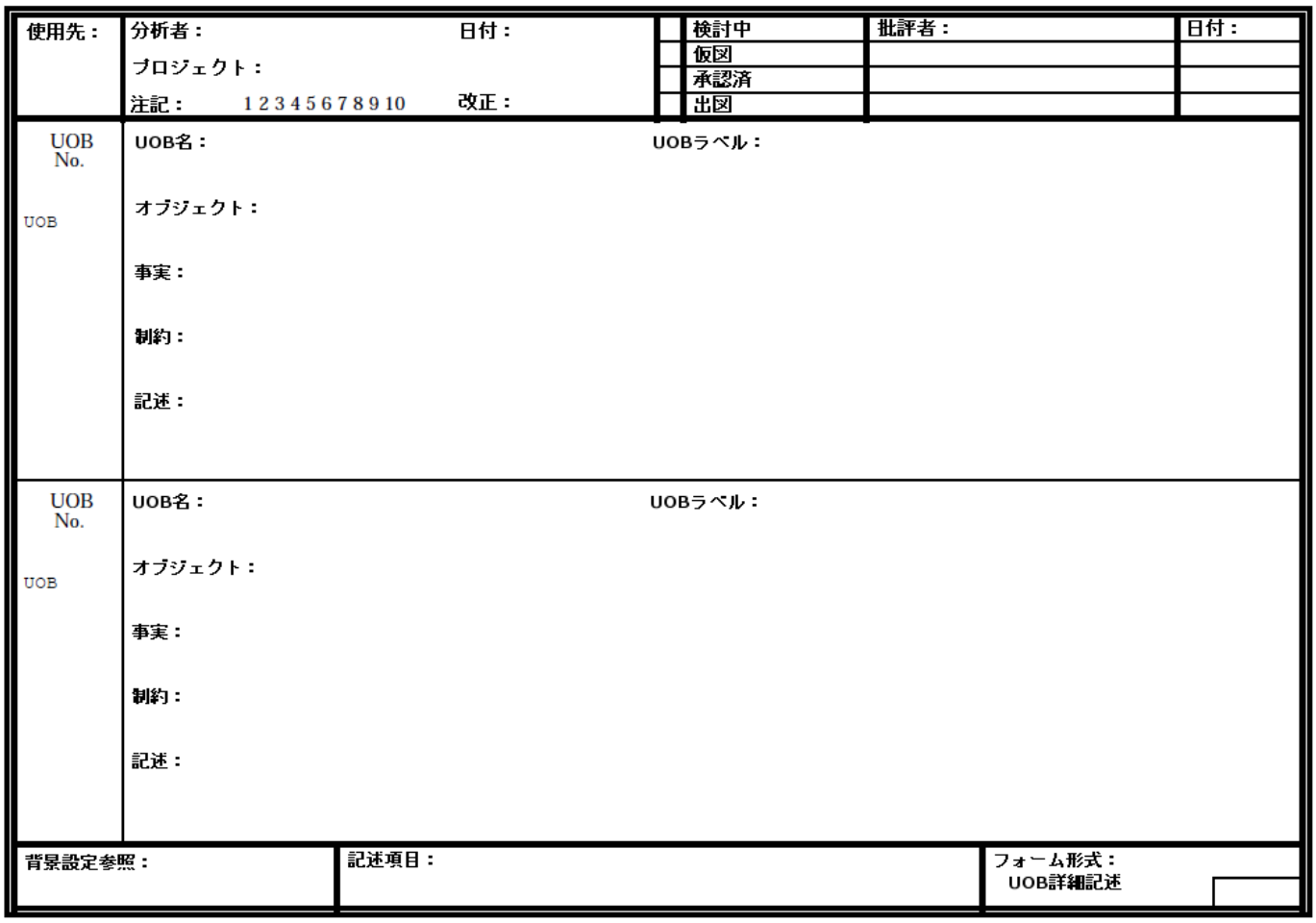


図 4-6　UOB 詳細説明フォーム

UOB 詳細説明ドキュメント（図 4-6）には、以下のような欄が用意されている：

（1）UOB 番号

関連する UOB 番号を記載する。この UOB 番号によって、
この詳細記述ドキュメントに関連する UOB を特定している。

（2）UOB 名

UOB 名を記載する。

（3）UOB ラベル

UOB ラベルを記載する（UOB 名、UOB 名の一部分、UOB 箱に書かれた略号、等）

（4）オブジェクト

UOB によって記述されているプロセスに参加している全てのオブジェクト（種類もしくはインスタンス）の名前の一覧を記載する。このオブジェクトは、物理的な物でも概念的な物でも構わない。また、プロセス上で作成、変更、削除されるオブジェクトも含まれる。オブジェクトは、以下のように発動物、受動物、参加物、生成物／削除物へと分類しておくと、便利である。

a）発動物（agent、エージェント、動作主）

オブジェクト（もしくは対象となる種類のオブジェクト）が、UOB のインスタンスで、主導的な役割（active causal role）を果たしている場合。

b）受動物（affected）

オブジェクト（もしくは対象となる種類のオブジェクト）が、UOB 活動のインスタンス内で変更される場合

c）参加物（participant）

UOB インスタンスで、オブジェクト（もしくは対象となる種類のオブジェクト）が変更されず、かつ主導性も持たない場合

d）生成物、もしくは削除物（created or destroyed）

オブジェクト（もしくは対象となる種類のオブジェクト）が、UOB のインスタンスで、生成もしくは削除される場合

（5）事実（fact）

UOB のインスタンスについての事実一覧を記載する。

（6）制約（constraint）

UOB の全てのインスタンスにおいて維持されるべき事実、等の
UOB に掛けられている制約の一覧を記載する。

（7）記述（description）

UOB の用語集の見出し（glossary entry）（テキスト記述（textual description））を記載する。用語集の見出しは、一般的には既にオブジェクト一覧、事実一覧、制約一覧に存在している情報を、テキストで詳細に説明したものである。

交差点の決定論理の獲得を促進するため、図 4-7 のような詳細説明ドキュメントを、プロセススキーム図内の交差点に添付することが可能である。

使用先：	分析者：　　日付： プロジェクト： 注記：　1 2 3 4 5 6 7 8 9 10　改正：		検討中	批評者：	日付：
			仮図		
			承認済		
			出図		
Junction No. J	交差点種類： オブジェクト： 事実： 制約： 記述：				
Junction No. J	交差点種類： オブジェクト： 事実： 制約： 記述：				
背景設定参照：	記述項目：		フォーム形式： 交差点詳細記述		

図 4-7　交差点詳細記述フォーム

交差点詳細記述の主要な記入欄は、以下の通りである：

（1）交差点番号

記述内の交差点に固有に割り振られた交差点番号（Junction（交差点）の頭文字である「J」を頭に付ける）を記載する。

（2）交差点種類

非同期 AND、非同期 OR、同期 AND、同期 OR、もしくは XOR（排他的論理和）

（3）オブジェクト

交差点に関連する、全ての重要なオブジェクト（種類もしくはインスタンス）。
一般的に、こうしたオブジェクトは、交差点に制約を与える発動物である。

（4）事実

交差点に関連し、制約を伴わない、記述すべき事実。
特に交差点に関連するオブジェクトについての事実。

（5）制約

決定論理の仕様や、それ以外の交差点に関係する制約を記載する。
IDEF3 詳細記述言語を用いて論理的に正確な形式で、この仕様を記述すると、理想的である。詳細記述言語については付録 A を参照。

（6）記述

その他の便利な注記や背景情報と共に、決定論理を非公式に記述したもの。

制約された順序リンクによって示されている特別な制約は、順序リンク詳細説明ドキュメント（図 4-8）に記録される。このドキュメントは、書式と目的において、UOB 詳細説明と似たものになっている。

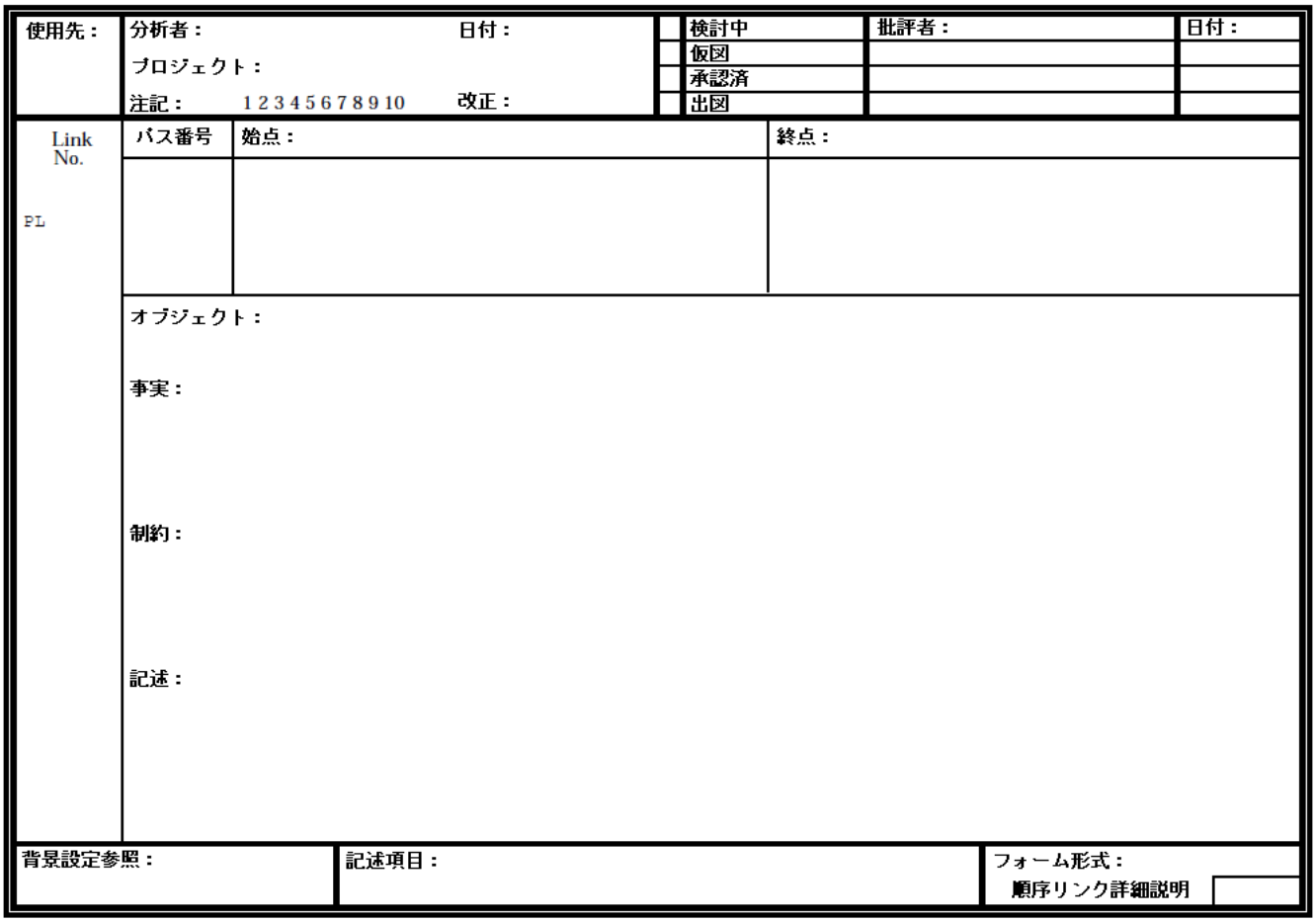


図 4-8　順序リンク詳細説明フォーム

順序リンク詳細ドキュメントの主要な記述内容は、以下の通りである：

（1）リンク番号

記述内の順序リンクに固有に割り振られた番号であり、頭に「PL」が付く。

（2）パス番号

リンクのパス番号。リンク番号と固有の整数、その間の「.（ピリオド）」によって構成されている。例えば、順序リンク「PL1」が交差点で 3 つの選択パスへと分離している場合、パス番号は PL1.1、PL1.2、PL1.3 となる。

（３）始点（source、出発地）

特定のパスの、リンクの始点となる IDEF3 要素（UOB もしくは参照）の名前

（４）終点（destination、目的地）

特定のパスの、リンクの終点となる IDEF3 要素（UOB もしくは参照）の名前

（５）オブジェクト

リンクが表現している関係に関わっている全ての重要なオブジェクト（種類もしくはインスタンス）。一般的には、リンクによって表現されている関係の始点もしくは終点にある構成要素である。

（６）事実

リンクによって表現されている関係に参加しているオブジェクトと関わっている、注目すべき、制約しない事実。

（７）制約

始点 UOB と終点 UOB の間で維持されているか注目すべき制約か、もしくはそれら制約オブジェクトの間で維持されている注目すべき制約。

（８）記述

リンクに関連する記述の用語集（descriptive glossary）。

ここ以外の欄（1～7）に記載できない記述情報も記載できる。

特定のリンクパスと独自な関連を持つオブジェクト、事実、制約についても、識別しておかなければならない。破線リンクによって表現される特別な制約は、図 4-9 の破線リンク仕様ドキュメントに記録する。

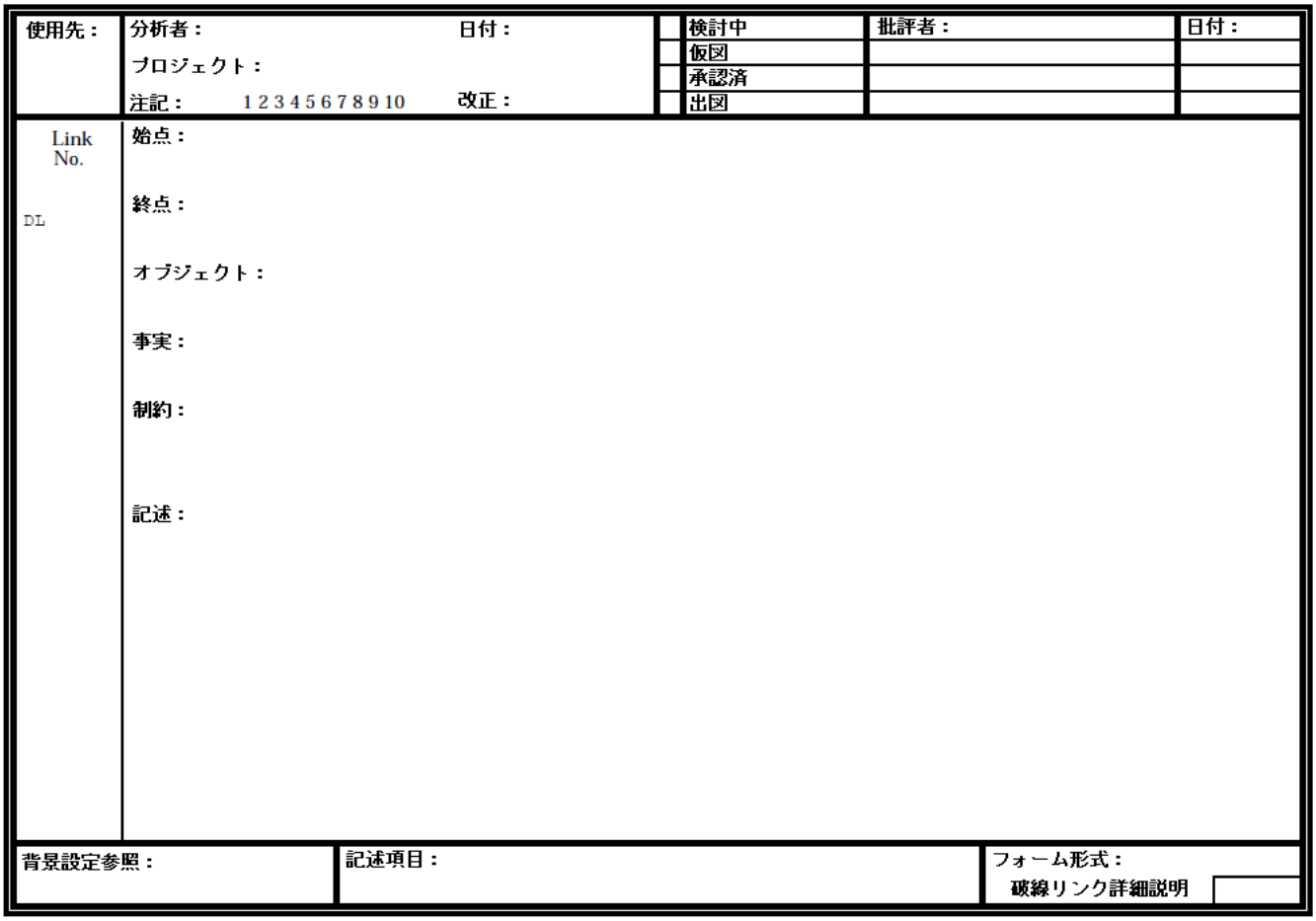


図 4-9　破線リンク詳細説明フォーム

以下は、破線リンク詳細説明ドキュメントの、主要な構成物である。

（1）リンク番号

「DL（Dash Link の頭文字)」で始まるリンク番号で、これにより、記述内のどの破線リンクのものであるかを識別する。

（2）始点（source）

リンクによって表現された関係の、始点の IDEF3 要素（UOB、参照等）の名前

（３）終点（destination）

リンクによって表現された関係の、終点の IDEF3 要素（UOB、参照等）の名前

（４）オブジェクト

破線リンクが表現する関係に参加する全ての重要なオブジェクト（種類もしくはインスタンス)。一般的には、こうしたオブジェクトは破線リンクによって表現される関係の、始点もしくは終点にある構成要素である。

（５）事実（fact）

破線リンクによって表現されている関係に参加するオブジェクトと関わっている、注目すべき、制約しない事実

（６）制約

始点要素と終点要素の間で維持されている注目すべき制約か、もしくはそうした制約オブジェクトの間で維持されている注目すべき制約

（７）記述

破線リンクに関連する用語集。ドキュメントの他の記入欄（1～6）に書きこめない記述情報も、ここに記載する。

４．５．１．４　選択したUOB（行動ユニット）の分解
（Develop Decompositions for Selected Unit of Behavior）

ある UOB の分解物（decomposition）とは、親 UOB によって表現されているプロセスに、ある断面に基づいた詳細情報を追加する UOB の集合体であると言える。シナリオレベル（ほぼ最上層）にある UOB は、分解を行うのが普通である。そして、このようにして作成された分解物を構成する UOB もまた、更に分解することができる。

活動中に何が発生しているのかということに対して、異なる分野専門家が持つ視界（view）は当然に異なる為、同一の活動を異なる分野専門家の視界で分解すれば、異なる分解物が生成されるのが普通である。また、どのオブジェクトの視界からプロセスを分解するかによっても、分解物が異なることになる。例えば、組織活動を支援するためには、情報システムに必要な処理ステップを示すような視界で、分解を行う事になる（？？）。

そして最後に、分析者が選択した UOB だけを分解することにより、スキーム図を判り易くすることもある。分解によって生成されたスキーム図は、視界をより詳細なものにし、また 1 つのプロセスを、明確に定義された視点からの異なる断面を通して見る事を可能にしてくれる。更に、1 つのプロセスが持つ選択可能な複数の視界を獲得したり、プロセス記述スキーム図を読みやすくする為に、分解を行う事も多い。

IDEF3 記述開発と同様に、分解作成（decomposition development）プロセスは、改善プロセスであり、最初の（？primary、主要な？）記述作成と同じ手続に従って行われる。この改善サイクルは、以下の 5 つの活動によって構成される。

①活動の分析

②追加データの収集

③関係する UOB の点から状況を記述

④批評

⑤必要なら、手続に従って１つ前のステップに戻る

４．５．２　オブジェクトスキーム図をまとめる（Formulate Object Schematics）

IDEF3 では、オブジェクトスキーム図はプロセススキーム図を補完するために用いられている。オブジェクトスキーム図は、オブジェクトや、オブジェクト状態、状態変移、オブジェクトの相互関係の特徴を、より詳細に記述してゆくことで、プロセスの「オブジェクト中心視界（object-centerd view）」を実現化している。

オブジェクトスキーム図の作成は、プロセススキーム図を作成する前、作成した後、もしくは作成中のいかなる時点であっても行われる。この節では、オブジェクトスキーム図の作成のガイドラインについて説明して行く。

オブジェクトスキーム図の構築ステップは、以下の通りである：

（１）興味の対象となるオブジェクトの選択

（２）オブジェクト状態の識別

（３）状態間で発生し得る変移の特徴を調査し、基本状態変移スキーム図を作成する。

（４）必要であれば交差点を追加し、選択可能な状態変移パスとオブジェクト構成ロジックとを反映させる。

（５）関係する UOB、シナリオ、そしてオブジェクトスキーム図に対する参照を、スキーム図上の適切な位置に取り付ける。

（６）必要であれば、詳細説明を作成する。

（７）選択したオブジェクト状態を分解する。

（８）別々の変移が、同一のオブジェクトを生成していないかどうか識別・注意する。

（９）背景設定情報が必要であれば、他のオブジェクトと関係とをスキーム図に追加する。

４．５．２．１　興味の対象となるオブジェクトの選択

ある記述の一部として、オブジェクトスキーム図を構築する際には、まずはどのオブジェクトを記述するのかを決定しなければならない。基本的に分析者は、システムに関する分野専門家の知識の中の、どのオブジェクトが重要な役割を果たすのかを識別しておかなければならない。プロセスに関係する全てのオブジェクトのリストは、膨大なものに成りかねない。それに対して、興味のあるオブジェクトだけをリスト化するのであれば、オブジェクトリストをより小さくすることが可能である。また一般的に、こうした興味対象となったオブジェクトは、プロセスが記述されていくに従って変更されて行くものである。

オブジェクトスキーム図の作成作業は、通常は 1 個以上のプロセススキーム図を作成した後に行われることから、対象となるオブジェクトの基になる主要な情報源（primary source）は、①UOB 詳細説明、②シナリオ記述、③シナリオが要求する情報モデル（他の IDEF モデルなど)、④インタビューの元データ、になる。オブジェクトの情報源が何であっても、オブジェクトは次の 2 つの共通の特性を持っている：

(a)プロセス上で、オブジェクトにとって重要な変化が進行中であること
(b)プロセス上の異なった点で、オブジェクトが幾つかの状態になっていること

オブジェクトは、理論上は物理的物体と概念的物体のどちらにも成り得ることから、どのオブジェクトが、対象領域内に存在してるのかを判断する科学的手法というものは存在しない（?)。しかし、IDEF3 を使用するような場合には、我々はシステムの運用で重要な役割を果たすオブジェクトに興味を持っているのが一般的である。また、そうしたオブジェクトは通常、名前を持っているものであることから、分析者はインタビューで収集した情報の中から、度々出てくる名前付きの語句を見つけて出してゆく事になる。この語句がどのようなものを表現しているとしても、オブジェクトの候補として考慮する事が可能である。

考慮すべき 2 つ目の事項は、興味対象のオブジェクトが、興味対象の状態にあるかかどうか、ということである。それには、以下のような条件を確認してゆくと良い：
①個々のオブジェクト状態が、対象領域内で共通に認識されている特徴を示しているか
②オブジェクトが、ある期間、その状態として存在していることを認識できるか
③認識された制約やプロセスによって、状態の変化が引き起こされたり抑制されたりしているか

そして選択された全てのオブジェクトに対して、少なくとも 1 つのオブジェクトスキーム図を作成して行くのである。

４．５．２．２　最初のオブジェクトスキーム図の作画
（Layout Initial Object Schematics）

オブジェクトスキーム図それぞれに対して、図 4-10 のようなオブジェクトスキーム図概要フォームを作成する必要がある。テキストによる記述（textual description）、もしくはオブジェクトスキーム図の用語集は、このフォームに書き込まれる。このテキストには、少なくともスキーム図の目的が明記されなければならないが、オブジェクトスキーム図に関する、他の記入欄には合わない情報をここに記入しても良い（存在論（ontology）情報は、後に IDEF5 モデルに組み込まれることになった）。テキストによる記述に加えて、分析者はオブジェクト状態と、このスキーム図で参照しているその他の IDEF3 要素（UOB、シナリオ、変移スキーム図）も記録する。オブジェクトスキーム図の作成に関係する分析活動の最初の作業が、このフォームを完成させる事である。このフォームを最初に完成させておくことで、分析者が生データ（raw data）からオブジェクトスキーム図を作成する作業が容易になる。

以下に挙げた記入欄リストには、オブジェクトスキーム図概要フォームのものだけでなく、変移スキーム図記述フォーム（？）の記入欄の説明も含まれている。

（1）オブジェクトスキーム図番号
オブジェクトスキーム図に割り当てられた固有の識別番号であり、変移スキーム図には頭に「TS」の文字を、オブジェクトスキーム図には頭に「OBS」の文字を付ける。

（2）オブジェクトスキーム図形式
オブジェクトスキーム図の種類は、変移スキーム図か拡張変移スキーム図、オブジェクトスキーム図の何れかになる。変移スキーム図（変移スキーム図と拡張変移スキーム図）

はオブジェクトスキーム図の特殊な形式であり、単体のシナリオによって確立された背景を持つものである。

（3）オブジェクトスキーム図名
オブジェクトスキーム図の名前。

（4）オブジェクトスキーム図ラベル
オブジェクトスキーム名か、名前の一部分、もしくは IDEF3 図形要素内に表示する際（参照として表示する際等）に便利な略称

（5）オブジェクト状態セット
スキーム図の種類が変移スキーム図である場合には、オブジェクトスキーム図によって表現される状態変移を構成しているオブジェクト状態のセットを記入する。一般的なオブジェクトスキーム図である場合には、オブジェクトスキーム図に含まれるオブジェクト状態の一覧を記入する。

（6）参照 UOB
オブジェクトスキーム図によって参照されている UOB のセット。

（7）参照シナリオ
オブジェクトスキーム図によって参照されているシナリオのセット。

（8）参照変移スキーム図
オブジェクトスキーム図によって参照されている変移スキーム図のセット。

（9）オブジェクト
オブジェクトスキーム図に含まれるオブジェクトの内の、背景設定の目的に用いるオブジェクトのセット。

（１０）記述
オブジェクトスキーム図に関連する、テキスト記述もしくは用語集。他の記入欄に当てはまらない情報も、ここに記入する。

図 4-10　オブジェクトスキーム図概要フォーム

オブジェクトスキーム図作成の次のステップは、個々のオブジェクト状態の記述と、状態変移の特徴づけである。これを完全に遂行する為に、分析者は以下に挙げる作業を実行する：

①個々のオブジェクト状態に対して定義されている特徴の識別

②個々の状態の退出条件の識別

③個々の状態の進入基準の識別

④その状態にあるオブジェクトが、変移を試み始める特別な条件の識別

⑤状態間で発生し得る変移の識別

⑥個々の変移の原因となる活動、変移を許容する活動、もしくは変移によってもたらされる活動の識別

４．５．２．３　オブジェクト状態、オブジェクト、交差点、リンクの詳細説明の作成

最初の 2 つの活動（前ページの①と②の作業？）の結果は、影響を受ける状態の、オブジェクト状態詳細説明フォーム（図 4-11）に記録してゆく。そして 3 つ目と 4 つ目の活動（③と④の作業？）の結果は、変移リンク詳細説明フォーム（図 4-12）へとまとめる。最後 3 つの活動（④と⑤と⑥の作業？）の結果は、スキーム図の配置へと反映する。こうした詳細説明フォームの構成と内容は、プロセススキーム図要素に関連する詳細説明と極めて似た物となっている。この詳細説明フォームの例は、図 4-11 から図 4-14 を参照のこと。

オブジェクト状態詳細説明ドキュメント（図 4-11）は、オブジェクトスキーム図で図示されている状態変移に関係する、オブジェクト状態の詳細説明を獲得する為に使用されている。このオブジェクト状態詳細説明ドキュメントは、オブジェクトスキーム図上の全てのオブジェクト状態に対して、個々に作成される。オブジェクト状態詳細説明ドキュメントフォームは、状態に詳細な特徴付けを可能とするだけでなく、状態と退出条件（第 3 章「IDEF3 プロセス記述言語」にて説明）についての情報も伝達している。

使用先：	分析者：　　日付： プロジェクト： 注記：　1 2 3 4 5 6 7 8 9 10　改正：		検討中 仮図 承認済 出図	批評者：	日付：
Object State No. OS	オブジェクト状態名： ラベル： オブジェクト状態からの変移： オブジェクト状態への変移： 事実： 制約： 状態条件： 退出条件： その他： 記述：				
背景設定参照：	記述項目：		フォーム形式： オブジェクト状態詳細記述		

図 4-11　オブジェクト状態詳細説明ドキュメント

オブジェクト状態詳細説明フォーム上にある記入欄は、以下の通りである：

（1）オブジェクト状態番号
オブジェクト状態の固有の識別番号で、頭に「OS」の文字が付く

（2）オブジェクト状態名
オブジェクト状態の名前

（3）ラベル
オブジェクト状態名か、その一部分、もしくは IDEF3 図形要素（参照等）上に表示しやすい略称

（4）オブジェクト状態からの変移
このオブジェクト状態から開始される、オブジェクト変移

（5）オブジェクト状態への変移
終了する事でこのオブジェクト状態へ至る、オブジェクト変移

（6）事実
この状態にあるオブジェクトについての事実

（7）制約
この状態にあるオブジェクトにかけられている制約。特に、以下の 3 つの種類の制約。

a) 状態条件
対象となる状態にオブジェクトを留めて置くために必要となる制約

b) 退出条件
対象となる状態からオブジェクトが離脱する際に満足する条件

c) その他
対象についての追加の制約

（8）記述
オブジェクト状態に関連するテキスト記述もしくは用語集。他の欄に記入できないドキュメントも、ここに記入する。

変移リンク詳細説明ドキュメント（図 4-12）は、オブジェクトスキーム図上の変移リンクの詳細説明を獲得する為に使用される。変移リンク詳細説明ドキュメントは、オブジェクトスキーム図上に表示されている全ての変移リンクに対して、それぞれ作成される。変移リンクそのものは、どのオブジェクト状態が別のオブジェクト状態へと変移可能か、ということを示しているに過ぎない。その為、変移リンク詳細説明は、次の 2 つの条件から構成されている：

①変移元の状態にあるインスタンスが、変移先の状態にあるインスタンスをもたらすような変移の開始を試みようとする、変移条件（transition conditions）

②変移元状態から離脱したオブジェクトが、変移先状態へと進入する為に満足させる必要のある、進入条件（entry conditions）

変移リンク詳細説明ドキュメントには、こうしたの情報に加え、リンクに割り振られた固有の変移リンク番号と、リンクを含有するオブジェクトスキーム図の名前（これは背景設定参照欄に含まれる）も含まれている。

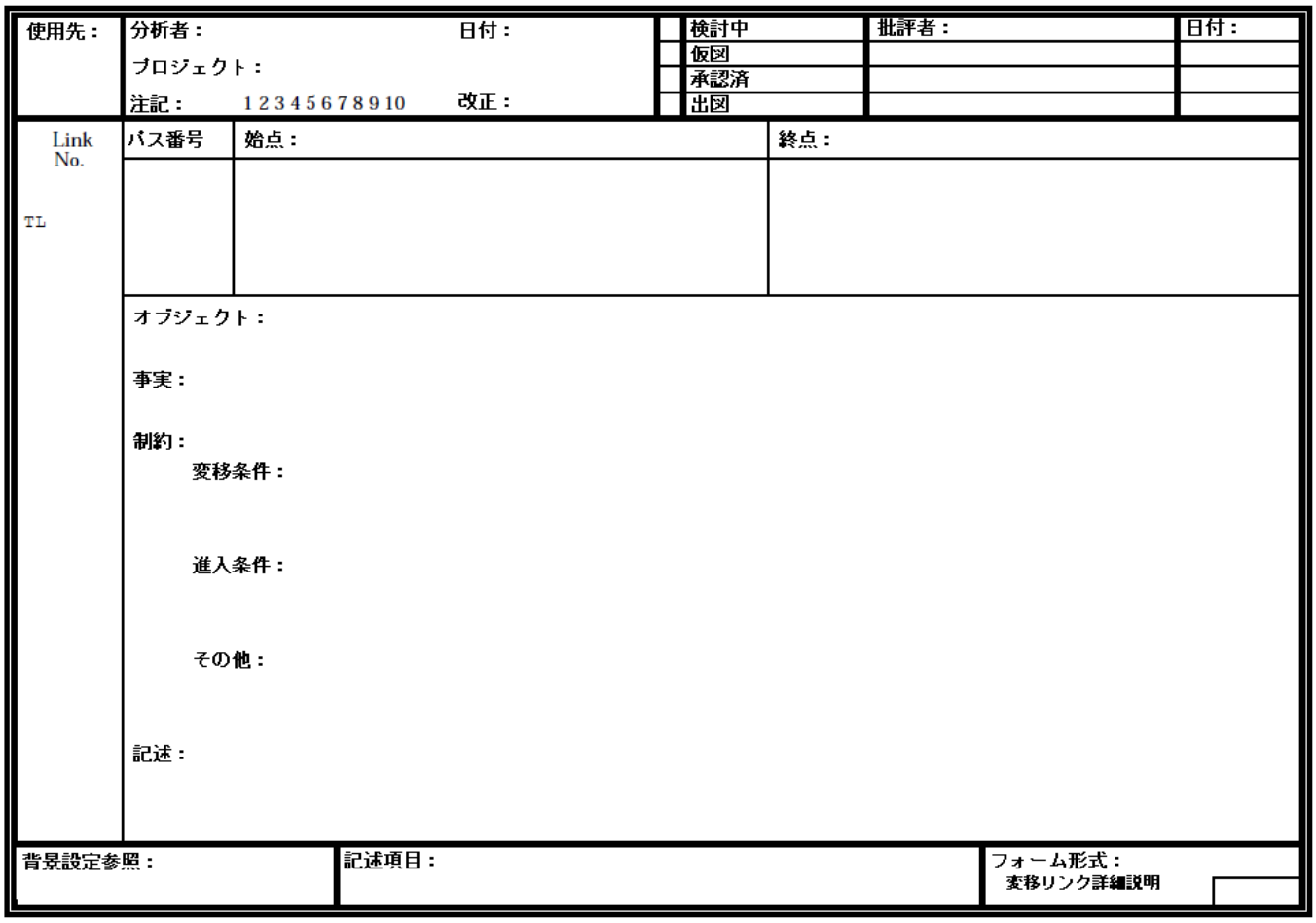


図 4-12　変移リンク詳細説明フォーム

変移リンク詳細説明フォームの記入欄は以下の通りである：

（1）リンク番号

変移リンクそれぞれに割り振られた固有の識別番号で、頭に「TL」を付ける。

（2）パス番号

リンクのパス番号で、リンク番号と固有の整数の間にピリオドを挟むことで構成している。例えば、交差点の後ろで 3 本の選択パスへと分岐する番号 TL1 の変移リンクのパス番号は、それぞれ TL1.1、TL1.2、TL1.3 となる。

（3）始点（source）

リンクの始点となる IDEF3 要素（オブジェクト状態、等）の名前。

（4）終点（destination）

リンクの終点となる IDEF3 要素（オブジェクト状態、等）の名前。

（5）オブジェクト

変移リンクによって表現されている関係に携わっている全ての重要なオブジェクト
（種類やインスタンス）

（6）事実

変移リンクによって表現されている関係性に携わっているオブジェクトに関する、制約以外の記録しておくべき事実

（7）制約

この状態にあるオブジェクトに掛けられた制約。特に、以下の 3 つの形式の制約がある：

a) 変移条件

そこにあるものが、始点から終点へ向かう変移を試みる為の、個別的な必要十分条件

b) 進入条件

適切な変移条件を満たした、そのリンクの始点の（恐らくは別の）状態にあるオブジェクトが、その状態に進入する為に満たすべき十分条件

c) その他

その他の興味対象となっている制約

（8）記述
変移リンクと関連する用語集。また、他の記入欄に記入できない全ての記述情報も、ここに記入する。

特定のリンクパスと独自に関連しているオブジェクト、事実、制約についても、必要に応じて識別して行く。
この時点で、変移に関係する、追加的な背景設定情報をもたらす可能性のある他のオブジェクトと関係も、識別しておくと便利である。そして、この識別作業を補助するものとして、オブジェクト詳細説明ドキュメントと、関係リンク詳細説明ドキュメントの、2 つの詳細説明フォームが用意されている。

オブジェクト詳細説明ドキュメントは、対象としている変移とは直接に関係してない変移スキーム図内に含まれている背景設定オブジェクトを、より一層特徴付けしてゆく為に使用されている。このオブジェクト詳細説明ドキュメントの例を、図 4-13 に示す。

以下は、オブジェクト詳細説明ドキュメント上にある、記入欄の一覧である：

（1）オブジェクト状態番号
オブジェクトに固有の識別番号で、頭に「O」の文字が付く。

（2）オブジェクト名
オブジェクト名

（3）ラベル
オブジェクトラベル（オブジェクト名や、その一部、もしくは略称）

（4）事実
オブジェクトのインスタンスに関する事実のリスト

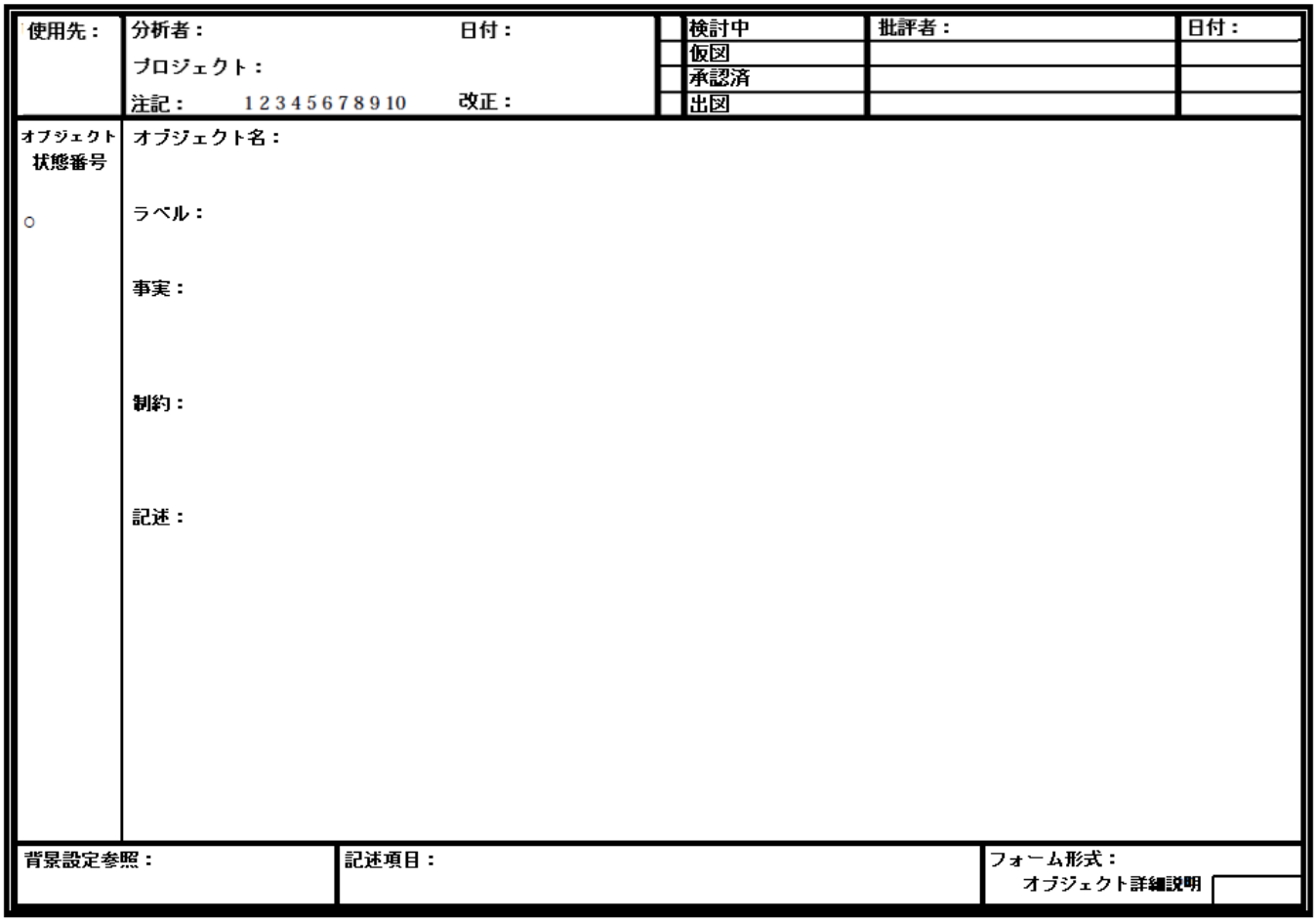


図 4-13　オブジェクト詳細説明フォーム

（5）制約

オブジェクトの全てのインスタンスで保持されなければならない事実等の、オブジェクトに課せられた制約の一覧。

（6）記述

オブジェクトの用語集項目（テキスト記述)。一般的に用語集項目（glossary entry）は、オブジェクトや事実、制約の一覧で既に説明済の情報を、テキストを用いて詳細に説明し直したものである。

関係リンク詳細説明ドキュメントは、オブジェクトスキーム図上にあるオブジェクトやオブジェクト状態の間に存在する関係（「~に変移」関係以外のもの）を、より詳細に特徴付けする為に用いられている。図 4-14 は、関係リンク詳細説明フォームの一例を示したものである。

関係リンク詳細説明フォームに含まれる記入欄は、以下の通りである：

（1）リンク番号
記述内の関係リンクを個別に識別する番号で、頭に「RL」の文字が付く。

（2）関係名
関係の名前。

図 4-14　関係リンク詳細説明フォーム

（３）関係種類

変数の数（number of place）と関係の階数（order）についての記述（二変数、一階、等）

（４）関係リンクに関わるオブジェクトとオブジェクト状態（即ち、引数（arguments））

関係リンクに関わる、オブジェクトとオブジェクト状態の一覧

（５）事実

関係リンクによって表現されている関係性に関与する（participate）オブジェクトについての、記録しておくべき、制約以外の事実。

（６）制約

関与するオブジェクトとオブジェクト状態の間、もしくはそれらの成分（constituent）オブジェクトの間で維持されている、記録しておくべき制約

（７）記述

関係リンクに関連する用語集。他の欄に記入できないような記述情報も、ここに記入する。

４．６　IDEF3 プロセス記述の漸次的な改良と検証
（Incrementally Refine and Validate IDEF3 Process Descriptions）

４．６．１　動機（motivation）

記述獲得の作業中に検証も行う場合に、IDEF3 の威力が特に大きくなる。既存のプロセスモデリング手法では、記述内に存在するギャップの上手く誤魔化し（gloss over）や、理想的な事実へと単純化に焦点を置くことが多いが、IDEF3 では、そうした制約を強要されることがない。IDEF3 は、システム運用に関して判明している事実の記録を行う、柔軟でありつつも几帳面な（flexible yet formal）メカニズムを持っている。

まず、スキーム図を作図することによってギャップと不一致が明確となり、分析者と分野専門家に対して注意を促してくれる。更に、１つのプロセスに対する複数の視点（viewpoint）を獲得することにより、違いが強調される。専門家と分析者の両者が、1 つの視界の内部と。複数の視界間のの両方で、ギャップと不一致を解消しようとすることにより、プロセスをより正しく理解することができるようになる。そしてこれにより、複数の専門者の間で、プロセスについての知覚がどのように異なっているのかを、理解できるようになるのである。

それとは対照的に、既存のモデリング技術では、専門家の記述について分析者が独自の理解を行った後でモデリングが行われる為、出来上がったモデルは、プロセスについての分析者の仮説となってしまう。そしてこのモデルが専門家に戻され、検証作業が行われるが、専門家はご都合主義（expediency）に陥ってしまったり、同調圧力（pressure for consensus）に屈して、完全に内容を理解しないままプロセスモデルを認可してしまう事が多い。

しかし IDEF3 を用いる事で、プロセス記述スキーム図を議論の焦点として使用することにより、プロセスの挙動についての異なった視点の間に発生する不一致を解決することが可能となる。

４．６．２　検証の種類（types of Validation）

「検証（validation）」は、構築された IDEF3 プロセス記述が、文法的にも意味論的にも正しいものであることをチェックし、保証するプロセスである。この定義から推測できるように、検証には「文法（syntactic）」と「意味論（semantic）」の 2 つの種類が存在する。

文法検証は、構築された IDEF3 スキーム図が IDEF3 言語の文法ルールを満足していることを保証することであり、また意味論検証は、IDEF3 記述上の宣言文が分野専門家の主張を正確に獲得したものであることを保証することである。

IDEF3 言語の文法にはスキーム図配置に対する制限が少ないことから、IDEF3 の手法を用いて、ユーザーはかなり自由に記述を構成する事が可能である。そして制約やルールの制限が少ない事により、構築された記述の構文（syntax）と意味論の中に、ユーザーの意図が十分に反映されることになるのである。

更に、こうした検証は、曖昧さの無い情報伝達手段（mean）として手法（method）の使い勝手（utility）を向上させるというやり方（manner）により、IDEF3 言語の潜在的ユーザー間の標準性を強化しようとしている。（訳注　意訳：検証がしっかりしていることにより、これから IDEF3 言語を使用しようかと考えている潜在的ユーザーに対して、IDEF3 言語が曖昧さの無い情報伝達手段である、という共通認識を持たせようとしている）

４．６．３　キットの構築と配布（Build and Distribute Kits）

IDEF3 プロセス記述を検証する主な手段は、キットの批評と承認を通して行われる。（本文注：このキット批評手続は、IDEF0 機能モデリングの報告書（AFWAL-TR-81-4023）から、直接持ち込まれたものであるが、これは複数の IDEF 手法の間で一貫性を保つために行われた。このキットは他の IDEF でも非常に重宝されている）

キットは、ほぼ完成状態となった記述の、一部分を表現している。キット批評作業は、追加事実の獲得を目的とした記述作成作業中であれば常に、また、分析作業の重要な段階が完了した時（UOB とオブジェクトの最初のリストが出来上がった時や、1 個以上のオブジェクトスキーム図が完成した時、プロセススキーム図が完成した時、等）に、実行する事が可能である。これから説明して行くキットの作成とその批評サイクルは規律付けられたものであり、それに従う事で、プロセス記述が正確なものとなる。

４．６．４　キット批評プロセスでのメンバーの役割

キット批評プロセスのチームメンバーの役割は、第 4 章で説明したチームメンバーの役割よりも、更に専門的になっている。キット批評プロセスに関係するメンバー各自の役割は、以下のようになっている：

（1）分析者

IDEF3 の専門家で、IDEF3 記述の主要な開発者でもある。批評プロセスは、分析者によって開始され、そして終了する。記述獲得とキット批評サイクルのどちらにおいても、記述の技術的な内容に関しては、分析者は分野専門家に依存している。

（2）批評者（reviewer）

IDEF3 キットの批評を行う全ての関係者

（3）コメンター（commentors）

コメンターは、キットの作業分野についての知識を持ち、また IDEF3 に精通し IDEF3 で決められた定型コメントを記入する事が可能な批評者である。コメンターは、分析者から提供された資料を読み、それが技術的に正確であることを検証するが、IDEF3 プロセス記述の間違いの発見や、改善の提案も行わなければならない。コメンターは、記述の目的が合致したものになっているか、間違いや見過しが存在するかどうかを判断してゆく。そしてコメンターは、批評プロセスの間、提案をキットに記入する権限を持っている。

（4）読者（reader）

情報を提供する目的でのみ、IDEF3 キットが配布される批評者。読者は、分析者が情報を得る為にインタビューを行った人であることが多い。

（5）司書（librarian）

プロジェクト関係のドキュメントと記述物のファイル管理や、コピーの作成、IDEF3 キットの配布、記録物の保管を担当する。

こうした「役割」は、個人の通常の役職とは関係ない。それどころか、同一人物が複数の役割を兼任する事もある。

４．６．５　IDEF3 キット批評サイクル

キットは、ほぼ完成状態になった IDEF3 プロセス記述の一部分を表現したものである。こうした記述の部分的下描きは、標準 IDEF3 キットの形で配布され、批評が行われる。図 4-15 の IDEF3 キットの批評サイクルは、他の IDEF 手法のキット批評プロセスを基にしている。これから行うサイクルの説明では、わかりやすくする為に、分析者とコメンターとの間のやり取りに焦点を置き、司書の役割を省略している。大きなシステムでは、司書の役割は必須であり専門に担当者が置かれるが、小さいシステムでは分析者によって兼任されている。

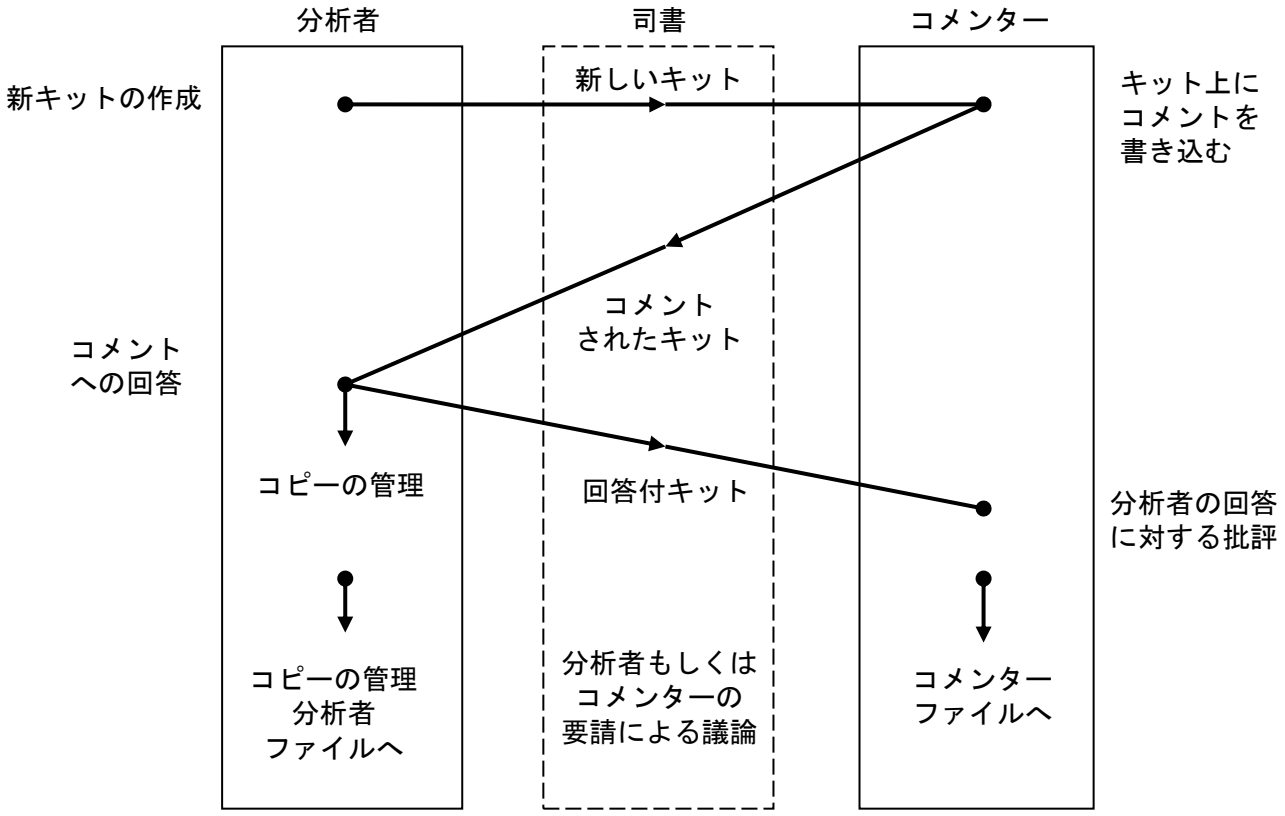


図 4-15　IDEF3 キットサイクル

以下は、IDEF3 キット批評サイクルの主な手順である。

（1）分析者は、キットを組み合せる（プロセススキーム図とオブジェクトスキーム図の付いた、プールキットや、シナリオキット、オブジェクトキット、記述キット、等)。分析者は一部のコピーを手元に置き、もう 1 部をコメンターに送付し、批評してもらう。

（2）コメンターは、決められた時間内でキットを調査する。この批評の主な目的は、できあがった記述が、記述作業の全体の目標と背景に従ったものであるかどうかを、判断する事である。コメントは、スキーム図やキットのその他のドキュメント、表紙に、直接記入する。コメントの入ったキットは、表紙に指示された期日までに、分析者へと返却しなければならない。

（3）分析者は、コメントへの回答を、キットのコメンター用コピーに直接記入する。分析者がコメントに同意したならば、その旨を作業用コピーに書き込み、次のバージョンの IDEF3 記述へと織り込む。もしもコメントに同意できなかった場合には、分析者は非同意の点をキット上に書き込み、コメンターへとキットを戻す。

（4）分析者からの回答に満足した場合には、コメンターは戻されたキットをファイルへと保存する。満足できなかった場合には会議を開き、コメンターと分析者との間で議論を行い、それによって解決する。

（5）このサイクルは、(分析者とコメンターの双方に）十分に受容可能な IDEF3 記述が出来上がるまで、繰り返される。

サイクルを通して、プロジェクトの司書は、分析者とコメンターの間で交わされる、IDEF3 キットのコピー、配布、ファイリング、運搬を執り行う（図 4-15)。

IDEF3 キットサイクルを実行することで、分析者とコメンターが共に関与した IDEF3 記述が出来上がる。もし必要となれば、経営者レベルでの対応が必要となる事項のリストも作成されることになる。また、この批評サイクルを通して蓄積される批評プロセスの履歴は、貴重な副産物となる。

４．６．６　IDEF3 キットの種類（Types of IDEF3 Kits）

IDEF3 キットは、他の IDEF 手法のキットと似た構造をしており、以下の 3 つの種類がある：

（1）シナリオキット

シナリオキットは、1 本のシナリオと、そのシナリオの全体もしくは一部分に関係するドキュメントを扱っている。シナリオキットは以下の物で構成されている：

a)　プロセススキーム図と、関連する全ての UOB 分解物。ただし開発プロセスの初期段階に作成される批評キットは、分解されない UOB も一部含まれる。

b)　利用可能な全ての UOB 詳細説明と、リンク仕様（link specification）。ただし開発プロセスの初期段階に作成される批評キットには、これらの一部もしくは全てを含まないものがある。

（2）オブジェクトキット

オブジェクトキットは、1 個以上のオブジェクトと、それに関連するオブジェクトスキーム図、オブジェクト記述、オブジェクト状態記述を扱っている。

（3）記述キット

記述キットは、開発作業の後半で作成される。記述キットは、対象プロジェクトのシナリオキットとオブジェクトキットの完成品を編集したものである。記述キットには、IDEF3 記述の全てのシナリオと、それに関連するドキュメントが含まれている。承認された記述キットは、開発作業における最終成果物（final deliverables）の 1 つとなる。

シナリオキットは、単一シナリオのプロセススキーム図から、全ての詳細説明と UOB 分解物スキーム図を含んだ完全なプロセス記述に至るまでの色々な詳細レベルで、作成する事が可能である。また記述キットも、色々な完成レベルで作成する事が可能であるが、シナリオキット中の単一シナリオの詳細レベルとは異なり、その完成レベルはプロジェクト全体の現状を反映したものになる。

４．６．７　分析者とコメンターの為のガイドライン

４．６．７．１　コメンターのガイドライン

主題となるものや様式、技術といったものは、幅広く変化してしまうことから、コメントを行う際の質問やルールには決まったパターンというものは存在しない。しかし、一方でコメントの品質を向上させる為のガイドラインは存在する。

品質に対する主な基準は、次のようなものになる：

・意図した相手に対して、十分に情報伝達できるのか？
・目的は達成されているか？
・対象となる背景範囲に対して、事実に基づき、正確であるか？

コメントを行う際の総体的なガイドラインを、以下に示す：

（１）注記は短く、完全で、明確なものとする。分析者が、簡潔さを優先させる為に正確さを低下させている事を自認している限り、情報伝達はより容易で、判り易いものになる。

（２）コメントを識別する為に、注記に「①」を用いる。「①」注記の書き方は、まず注記リストから空いている次の番号をチェックし、注記に番号を振って番号に丸を書き、注記した部分と番号との間に波線を引く。

（３）建設的な批判（criticism）を行う。後ろ向きなコメントだけを書くのではなく、その解決策も提案する。

（４）総体的なコメントを時間をかけて収集し、表紙もしくは別シートにまとめる（個々のページ内のみのコメントについては収集しない）。また、分析者とコメンターの間で開かれる会議の為の、協議項目もまとめておく。この協議項目は明確なものにしておく。

批判（critique）にかける時間は、主題に精通している度合いや、批評（review）を行った回数、コメンターと分析者の経験値等の、異なった複数の要因に依って変化する。また、コメントの無いIDEF3キットが分析者へと返却されるという事は、コメンターが分析者に完全に同意していることを意味している。コメンターは、仕事の品質に対する責任を分析者と共有しているということを認識しておくべきである。

４．６．７．２　分析者とコメンターの間のやり取り
（Analyst/Commentor Interchanges）

IDEF3 キットがコメンターから戻されたら、分析者は番号付き注記に対してチェック印「レ」とバツ印「×」を付けて行く。「レ」印は、分析者はコメンターのコメントに同意しており、次のバージョンの IDEF3 キットにコメントを織り込む、ということを意味している。一方の「×」印は、分析者はコメントには不同意であり、コメントを付けた理由を要求する、ということを意味している。分析者が全てのコメントに対して回答を付けた後、IDEF3 キットはコメンターへと返却される。

コメンターは、分析者からの回答を読んだ後、それでも非同意の点が残っていれば、分析者との会議を要求する。この非同意点リストは、そのまま会議での協議項目となる。

４．６．７．３　会議のルール

紙の上で（キットを使用して）コメントのやり取りを行うまでは、コメンターと分析者は直接意見交換を行うべきではない。

会議開催が要求されたら、次の手続きに従って行う：

（１）会議は、それぞれ制限時間を設けるべきである。

（２）セッション（項目別の小会議）はそれぞれ、明確な協議項目に従って開始し、議論が協議項目から逸脱しないように注意する。

（３）(疲労から) 会議の生産性が落ち、労力がより多くなったことに参加者が同意したら、セッションを中断すべきである。

（４）フォローアップセッションの日取りや、次回の協議項目が明確にされた行動項目リストに参加者が同意することにより、セッションが終了する。

（５）セッション（小会議）毎に「書記」を置き、会議時間の監視と、行動や決定、項目の記録を行わせる。

（６）未解決の重大な相違点については、上手く（professionally）取扱うべきである（双方の視点に立ってドキュメント化する、等）

会議の結果は、書面にされた問題の解決策か、もしくは適切な経営会議へと委託される問題リストへとまとめられる。参加者によって問題の解決策の更なる研究が行われる場合も有る。

４．６．８　IDEF3 キットの構成物

IDEF3 キットは、スキーム図やテキスト、用語集、決定概要（decision summary）、詳細説明、背景情報、その他関連資料を、批評（review）とコメントを行う為にパッケージ化した、技術ドキュメントである。

４．６．８．１　キットを準備する際の一般的なガイドライン

見落としを避ける為に、IDEF3 キットの見直しを行い（review）、入手可能な情報だけになるようにする。またその際、明確にしておきたい点を短い注記として、IDEF3 キットに追加しておく。IDEF3 キットに含まれる専門用語を定義した用語集は、補助資料として常に添付しておくべきである。

役に立つ資料を集め、コメンターのが利用できるように添付しておく。スキーム図そのものによって伝達すべき情報を、この追加資料で伝達してはならない。また読者が概念を

理解する為に重要となる詳細情報の表現には、可能な限り会話（most natural means of communication = speech、話し言葉？）を用いる。最後に、全ての資料をまとめ終わったら表紙を付けて完成させ、司書へと提出する。

４．６．８．２　キットの表紙

キットの表紙は、IDEF3 キットとしてまとめられた資料を、明確に区別してくれる。表紙には、分析者や日時、プロジェクト、ドキュメント番号、題名、状態、注記といった記入欄がある。ここでは、IDEF3 キット表紙（図 4-16）の各欄がどのような情報を与えてくれるかを説明する。

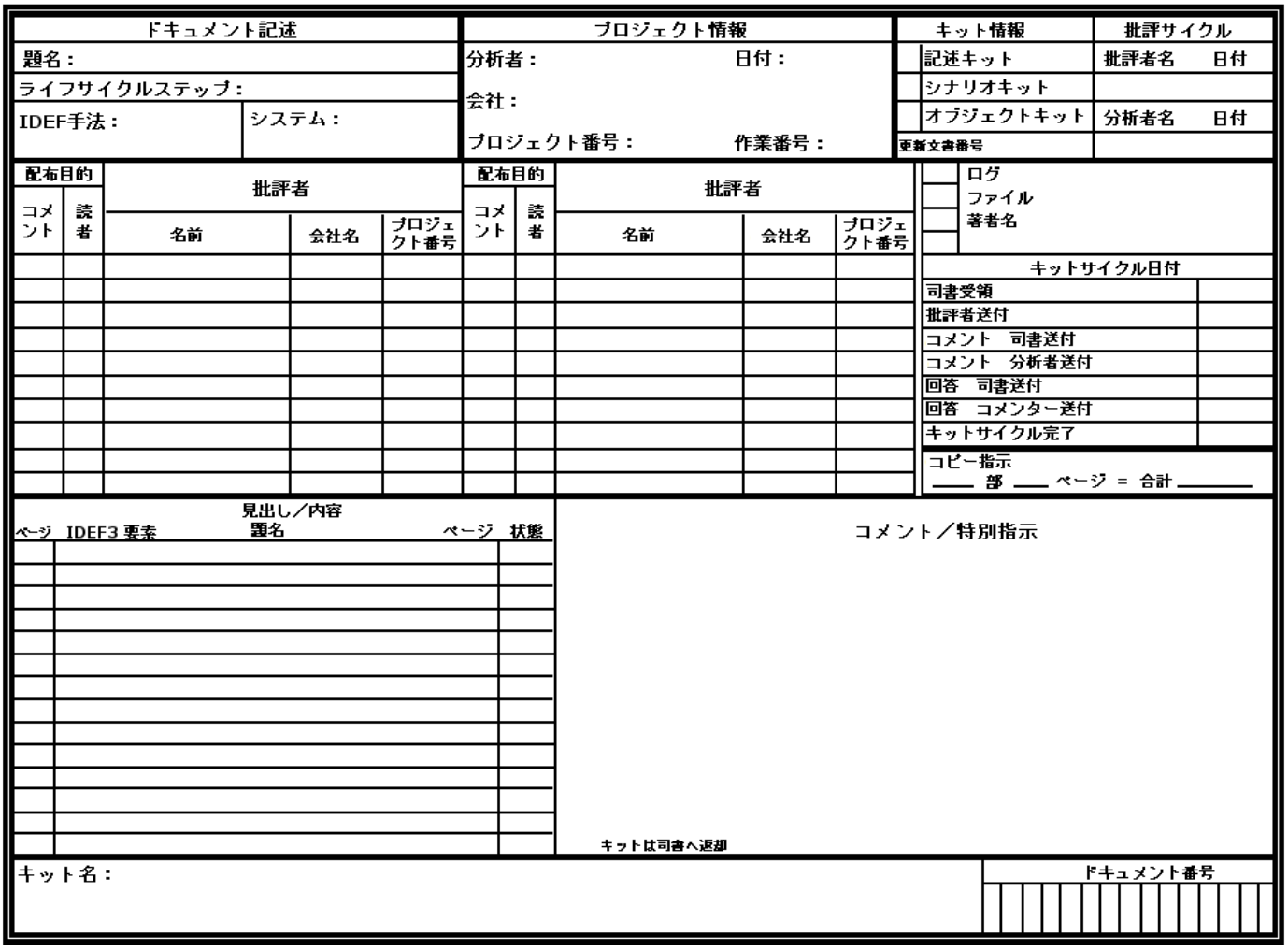


図 4-16　IDEF3 キット批評　表紙

（1）IDEF3 プロセス記述／ドキュメント記述

a) 題名

IDEF3 キットを言い表したものであるべきである。

b) ライフサイクルステップ

「現状（AS-IS)」もしくは「理想（TO-BE)」（キットに含まれている記述が、現状を示したものなのか、理想を示したものなのか、について）

c) システム（System、系列？）

システムもしくはサブシステムの頭文字

（2）プロジェクト情報

a) 分析者

IDEF3 キットを提出した人名

b) 日付

司書に送付された日付

c) 会社名

IDEF3 キットを提出した会社名

（3）IDEF3 キット情報

記述キット、シナリオキット、オブジェクトキットのいずれかにチェック「レ」を入れる。ドキュメント番号は司書によって割り振られる。

（4）評価サイクル

コメンターと分析者による批評の後に記入される、それぞれの署名と日付

（5）見出し／内容

シナリオ、記述、オブジェクト、オブジェクト状態（もしも関連するなら）の名前と、それのドキュメント内でのページ番号の一覧。記入欄が足らなければ、IDEF3 キット内容シート（図 4-17）と呼ばれる追加シートに記入し、添付する。

（6）コメント／特別指示

批評者の為の、その他の情報。この欄は、ドキュメントの取扱に関する司書への特別指示、司書から IDEF3 キットの受取人に対する特別指示としても使用する。

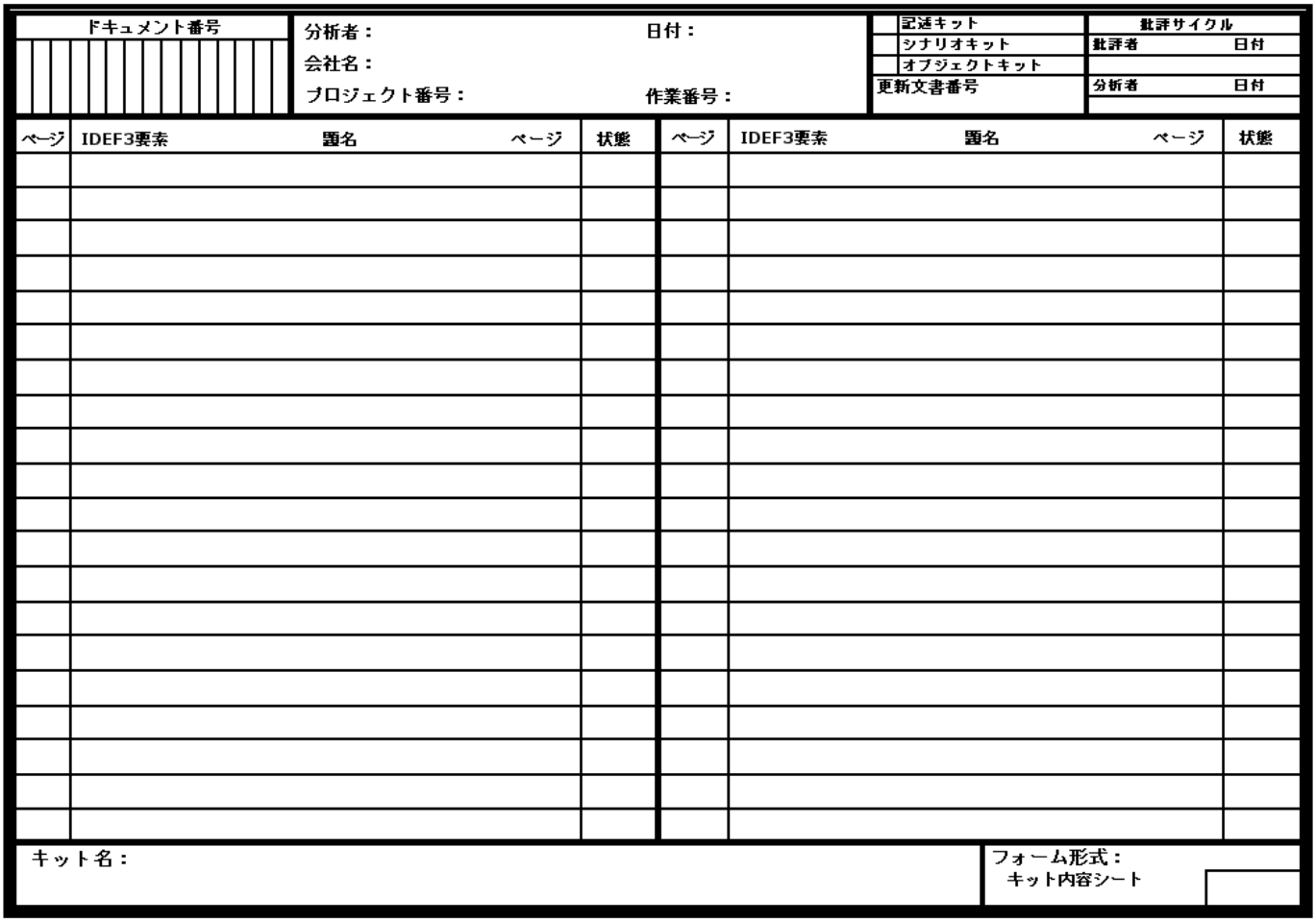


図 4-17　IDEF3 キット内容シート

４．６．８．３　キットスキーム図フォーム（The Kit Schematic Form）

図 4-18 のキットスキーム図フォームの構成は最小限であり、制約も少ない。このシートは、①視点（viewpoint）の確立、②書類のシート間の相互参照（cross-referencing between sheets of paper）、③各シートの内容に関する注記のドキュメント化（documenting）のような、構造化された分析作業を維持する為に重要となる機能についてのみ、補助するものである。

このフォームは、①作業情報欄（上部）、②メッセージ欄（中央）、③識別欄（下部）３つの主要な区分で構成されている。

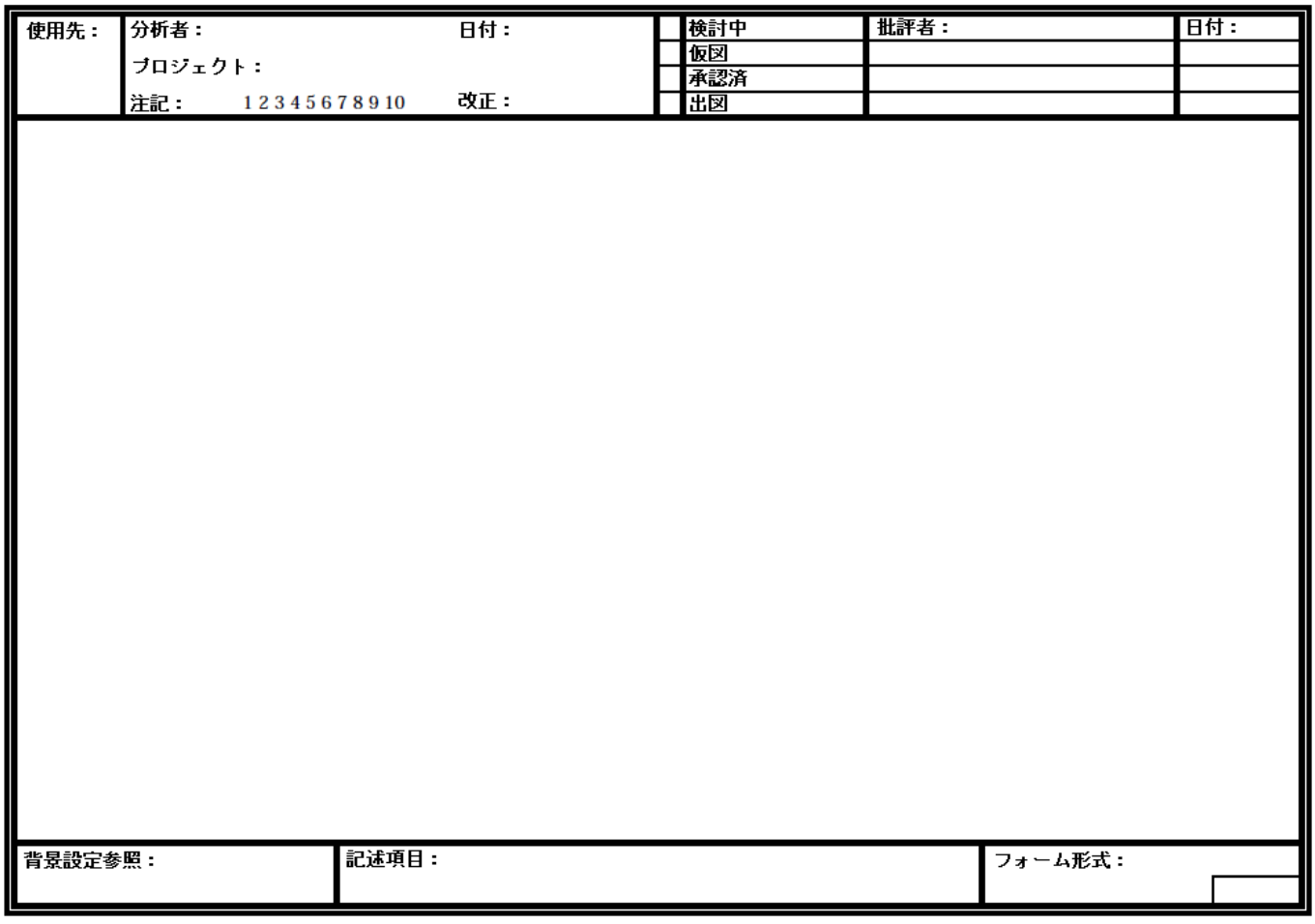


図 4-18　IDEF3 キットスキーム図フォーム

このフォームの上部にある作業情報欄は、最終的な「出図承認」版が完成した際に、切り離しできるように作られている。印刷されたドキュメントが使用されている時は、このスキーム図フォームを常に使用するべきである。作業情報欄内の記入欄には、以下の物がある：

（1）使用先

直接の背景としての用途以外で、このシートを何らかの用途に使用しているスキーム図のリスト

（2）分析者／日付／プロジェクト

このスキーム図の元々の作成者と作成された日付、その時のプロジェクト名を記述する。追加の日付は、元々の日付の下に記入する。この日付は、元シートに対する「改正（revision）」日付を表わしている。変更が無かった場合には、改正日付は記入されない。

（訳注：記入欄の中の「改正：」欄についての説明が無い。文脈からすると、上の改正日付は、この「改正：」の横に書くのではないかと思われる）

（3）注記

スキーム図シート上に書かれた注記番号（「①」等）を消し込む為に使用される。ページ上にコメントを記入したら、その注記番号に×を付けて行く。これにより、コメントの番号を素早くチェックすることが可能であり、またこのようなやり方で採番した注記番号は、特定のコメントに対して固有のものになり、番号の重複を防ぐことが出来る。

（4）状態

作成中、下書き、仮出図、出図済の 4 つの状態区分は、それぞれ承認のランクを表現している。

（4－a） 検討中（working）

このスキーム図は、以前の状態とは無関係に大きな変更が加えられた状態である。新しく作られたスキーム図はもちろんのこと、作業用コピーもこれにあたる。

（4－b） 仮図（draft）

このスキーム図は、前版のスキーム図から小さな変更が加えられており、そして関連する複数の読者によって承認された段階まで、合意された状態にある。仮図状態にあるスキーム図は、プロジェクトリーダーによって提案されているものの、プロジェクトチームからは承認されていないスキーム図である。

（４－c） 承認済（recommended）

このスキーム図と、それに添付されたテキストの両方が、プロジェクトチームによる批評と承認とを受けた状態であり、これ以上の変更は期待されていない（訳注：ほぼ完成状態であり、根本的なミス等による修正はあるとしても、小さな理由からの変更は行うべきではない、という意味）。

（４－d） 出図（released）

このページは、このまま最終リリース版として他で使用されるか、出図される状態にある。

（５）批評者／日付

この欄は、フォーム毎のコメンターのイニシャルと批評した日付を記入する。

フォームの中央部にあるメッセージ欄には、伝達すべき主要なメッセージが記入される。この欄には、通常はスキーム図が描かれるが、他の用途（用語集、チェックリスト、注記、スケッチ等）にも使用する事が可能である。

フォームの下部にある識別欄は、以下の物で構成されている：

（１）背景設定参照（context-setting reference）

この欄の情報は、メッセージ欄内の情報を解釈してゆく為に必要となる背景（context）の確立を手助けしてくれる。背景の確立は、固有な IDEF3 要素番号（シナリオ 1、分解 1.1、UOB43、PL31、J5、O6、OS22、等）を用いた参照識別子によって行われる。グローバル背景の場合には、識別子として「グローバル（Global）」という言葉を使用する。例えば、シナリオ詳細説明がメッセージ欄に表示されている場合には背景設定参照は「グローバル」となり、プロセススキーム図がメッセージ欄に表示されている場合には、背景設定参照はシナリオ番号もしくは親 UOB の番号となる。

（２）記述項目（item described）

この欄には、メッセージ欄に表現されている素材の名前を記入する。メッセージ欄にスキーム図が描かれている場合には、スキーム図の親 UOB 箱に書かれている名前をそのまま

正確に書かなければならない。

（３）フォーム形式（form type）

標準的なIDEF3キットフォームは、キットの表紙以外にも、記述の開発中に使用される全てのフォームの基本構成としても使用されている。その為、この欄によって標準的IDEF3キットフォームを何に使用されているかということを明確にする。標準的なIDEF3キットフォームの利用が推奨されるフォーム形式には、記述概要フォーム、キット内容シート、プロセススキーム図概要フォーム、オブジェクトスキーム図概要フォーム、素材資料ログ（Source Material Log）、素材資料記述フォーム、独立したプールフォーム（individual pool forms）、IDEF3要素詳細説明フォームがある。

４．７　批評の進展に伴う調整の実行（Review Progress and Make Adjustments）

記述獲得作業の過程で、プロジェクトチームは、プロジェクトの目的と作業領域（scope）の批評、そして進捗の評価とを、それぞれ頻繁に行わなければならない。またプロジェクトでの問題状況の誤解を防ぐ為に、作業領域の再定義を伴う調整が必要となることもある。プロジェクトの目的に向かって正しく進んでいることを頻繁に評価していくことで、高い代償（payoff）を招くような機会の早期発見を促進し、効率の悪い活動で浪費される時間と支出を制限することができる。その為、微小なものから劇的なものに至るまで、作業領域（scope）の変更が必要となることが多い。そして作業領域の変更が必要になった場合には、顧客に通知して承認を取るべきである。

また技術獲得作業の過程で、分析者は他の手法とIDEF3の併用を増大させる必要性を認識することになるだろう。

４．８　プロセス記述獲得での、他の IDEF 手法の併用

IDEF3 は、それ単体だけでも、他の IDEF 手法と協調させてでも、使えるように作られている。更には IDEF3 に、IDEF ファミリー以外の手法や技法を適用し、成功したプロジェクトも幾つか存在している。このように複数の手法技法を併用する場合には、手法毎の役割を明確にしておく必要がある。また、そうした色々な手法を適用する際に用いる慣習を、明確に定義しておくことも重要である。

プロジェクト毎の手法の適切な組み合わせは、利用可能な情報の本質と様式（form）、そしてプロジェクトの目標に依存している。IDEF の各手法は、固有の情報セットと認識支援アプリケーション（？cognitive support applications）として作り込まれている。例えば IDEF0 機能モデリング手法と IDEF1 情報モデリング手法は、複雑な状況の分析に便利であり、一方で IDEF5 存在論記述獲得手法は、オブジェクト構造とオブジェクト関係の記述に大きな力を発揮する。

1 つのプロジェクトに 2 つ以上の手法を適用することで、手法というものの本質が主に狭い作業範囲を支援する為に作られたメカニズムであり、一般的なものからプロジェクト特有なものに至るまでの、幅広い範囲のシステムエンジニアリングフレームワークに適用可能なものあるということを、実感する事ができるだろう（かなり意訳）。こうしたフレームワークは複数の作業間で必要となる統合の背景を確立し、そして結果的にそうした作業を支援する手法に必要な背景をも確立することになるのである。

４．８．１　IDEF3 と共に IDEF0 を使用する場合

IDEF0 モデルと IDEF3 プロセス記述の UOB とを比べた時、IDEF3 は IDEF0 の置き換えを意図していないことがわかる。分析を行うシステムが非常に大きなものである場合（航空機の製造、等）には、順序関係は明確にはならないことから、こうした場合には IDEF0 モデルの作成から開始した方が良い事が多い。IDEF0 モデルを分解して行くに従い、次第に活動間の順序関係が重要になるレベルへと至るのである。また一方で、収集した事実を綿密な流れ（cohesive story）へとまとめて行く場合には、IDEF3 プロセス記述を先に形にしておき、そこから IDEF0 モデルを抽出して行くのが一般的である。IDEF3 手法は、こうした相互関係を扱う為に作られたものであることを覚えておくべきである。

IDEF3 の文法では、IDEF3 の UOB から、その UOB に関係する IDEF0 活動への参照を張る事が可能であり、この関係性を明確にすることができる。UOB 箱には全て、IDEF0 モデルの活動や、類似した機能、もしくはプロセスモデル（論理データフローダイヤグラム内のノードや、HIPO 図、等）への参照を記入できる欄がある（図 4-19 の右下）

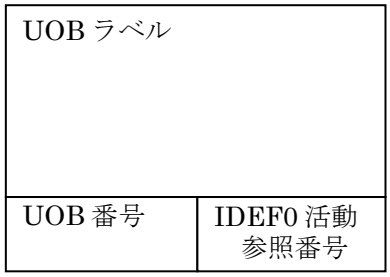


図 4-19　挙動ユニットの欄

IDEF3 の参照スキームでは、1 つの UOB に対して、0 もしくは 1 以上の IDEF0 活動をマッピングすることができる。IDEF0 活動の一部分のみに UOB がマッピングされている場合には、実際に関係している子活動のセットだけを、正確に参照指示すべきである（訳注：対象となる活動セットを正確に指示できるまで、IDEF0 モデルの分解を行う）。また IDEF0 モデルが十分な詳細レベルまで定義されていない場合には、マッピングを行う範囲を、UOB 詳細説明の中で説明しておくべきである。UOB が識別される度に、その中に IDEF0 参照を張っておくべきである。

４．８．２　IDEF3 と共に IDEF1 と IDEF1X を使用する場合

大型の IDEF3 開発プロジェクトでは、プロジェクトの開始前に IDEF1 モデルと、必要なら IDEF1X モデルも、先に準備しておくことが多い。それにより、オブジェクトスキーム図を作成する際に、オブジェクトの識別が容易になる。IDEF1 の場合では、オブジェクトやオブジェクト状態に関係する実体クラス番号か属性クラス番号を、IDEF1X の場合には実体番号もしくは属性を、オブジェクトスキーム図の用語集もしくは適切なオブジェクトスキーム図記述フォームの用語集で、参照しておくべきである。

プロセス記述獲得は、一般的に直線的で判り易いものであるが、情報システムによって支援されたビジネスルール（buisiness rules）の決定作業は、より困難である。また、経営の「非公式な」情報システムという、情報の落とし穴が隠されている事も多い。

情報モデリングの主要な役割の一つは、非公式・公式の両方の情報システムの定義を行うことである。そして IDEF3 記述は、情報モデルの開発前でも後でも、開発が可能である。

情報モデルの開発よりも先に IDEF3 記述の開発を行う場合には、分野専門家の注意が、彼ら自身のプロセスの支援に必要となる情報に集中することから、情報モデルの開発が促進されることになる。その結果開発された情報モデルは、経営に必要な情報を、プロセス中心視界（view）から見たものになっている。そして、こうして作られた複数のプロセス視界情報モデルを統合してゆく事により、経営規模でのデータ標準として利用可能な包括的情報モデルが、次第に出来上がってゆくのである。

４．８．３　IDEF3 と共に IDEF5 を使用する場合

IDEF5 では、領域存在論の定義においてプロセス種類が持つ重要性の為、オブジェクト種類と同様にプロセス種類への参照も可能になっている。

しかし、そうした参照が発生した際には 2 つの異なった背景が存在し、また、プロセス種類に関して維持されている情報は、背景によって異なったものになる。

もしもプロセス種類 P が 2 つのオブジェクト種類に関係する変移もしくは変質（？transformation or transition）についての記述の中で参照されている場合には、この P の「内部」特徴は、IDEF3 プロセス記述獲得手法に従って記述される。

これはつまり、それが関係するオブジェクト種類とそれらが持つ特徴、そして対象となるプロセスのインスタンスが生成された時にオブジェクト種類のインスタンス間で結ばれる適切な関係性という 3 つの面から、P が記述されているということである。この場合は特に、オブジェクト種類に関する情報、つまり特徴等を定義している一般的な情報は、プロセス種類向けのものではない。

一方で、分野を理解する為には、プロセスの内部構造で、オブジェクトがどのように関

係し合っているのかということを理解するだけでなく、オブジェクト種類おいては一般的なことであるが、その内部構造の詳細とは（一般的には）無関係に、あるプロセス種類と別のプロセス種類とが、論理的にどのように関係しているのかということを知ることが重要となる。

例えば、「製造プロセス計画」は、「計画」の下位種類（subkind、サブ種類）である。この場合、プロセス種類は、IDEF5 存在論獲得手法で提供された手続と言語構造を使用して特徴づけられている。（訳注：『「製造プロセス計画」は、「計画」の下位種類（subkind、サブ種類）である』という一文の定義が、IDEF5 の手続と言語構造で特徴付けられているのか、それとも後ほど詳細に説明される「製造プロセス計画」と「計画」の中で、特徴づけが行われているのか、良くわからない）

第 5 章　IDEF3 開発：　資材発注プロセスでの例

この章では、ある会社の発注プロセスの記述開発の例を用いて、共通な設定の範囲内で、IDEF3 手法の使い方について説明する。この例には、ユーザーが陥りやすい間違いを回避する為のヒントも含まれている。更に、記述開発への構造化プロセスの適用理由を、初心者に対するガイドとしてまとめている。

５．１　目的と背景の定義（Define Purpose and Context）

業務責任者（owner of business, business owner、普通は社長や経営者だが、部門長である可能性もある為、業務責任者としておく）が、新規雇用者の教育補助と会社の調達標準の強化を目的とした、IDEF3 による資材発注プロセスのドキュメント化に興味を持っているとする。そこで業務責任者は、「資材の発注（Order Material）」というシナリオの IDEF3 プロセス記述開発を分析者に依頼し、開発プロジェクトが開始された。

このプロジェクトは、調達要求の処理方法を記録し、それを新規雇用者の役に立てたいという業務責任者の願望に起因しているとする。こうした状況で IDEF3 を適用する利点の 1 つは、業務責任者自らがこの知識の伝達に時間を割かれることなく、新規雇用者が自分で IDEF3 プロセス記述を参照する事により、必要な資材をどのようにして入手しているのかということを簡単に理解可能なところである。

このプロジェクトの例では、プロセス記述の対象範囲は、社内活動に限定されることになる。また開発を通じて、新規雇用者が発注プロセスの作業を明確に理解する為に必要となる情報だけが獲得されることになる。

以上のような目的と背景は、IDEF3 プロセス記述概要フォームに記入されることになる。

記述開発プロセスのこの段階では、分析者は通常、候補となるシナリオを識別し、それをまとめて IDEF3 シナリオプールを作成する。このプールの内容は、プロジェクト期間を通じて改良、維持管理されて行くことになる。

このプロセス開発例で登場する IDEF3 プロジェクトチームの役割は、①分析者（IDEF3 専門家）、②分野専門家（業務責任者）、③顧客（これもまた業務責任者）の 3 つだけであ

る。この例では分野専門家と顧客を業務責任者が兼務しているが、通常は別々の人が分担している。そしてこの章では、登場人物はプロジェクトでの役割名（分析者、分野専門家、顧客）によって参照されることが多いことを念頭に置いておく必要がある。

５．２　データの収集（Collect Data）

プロジェクトの定義が終わると、分析者はデータ収集作業を準備し、それを実行する。このデータ収集の中で、最も効果を発揮する手段はインタビューである。この例では、インタビューによるデータ収集に、主に焦点を当てている。

５．２．１　分野専門家へのインタビューと、初期記述の取得（Interview Domain Expert and　Aquire Initial Description）

十分に計画され、上手く実行されたインタビューの効果が決定的なものとなることから、分析者は注意深く準備を行うべきである。インタビューは、分析者による「発注が必要となった際に、どのようにして発注を行っているのですか？」という問いかけから始まることになる。これに対して、分野専門家は次に様に回答したとする：

> **まず最初に、我々は購入要求書を使用して資材の要求を行います。すると購買部門（Purchasing Department）は、要求した種類の資材を担当する現在我々と取引のある供給者か、潜在的な供給者の候補の識別を行います。我々は、取引のある供給者と長期的な関係を結ぶことを望みます。これは可能な限り、現在取引している供給者と取引すると言う事です。それにより、高い品質の製品を、時間通りに、合理的な価格で供給してもらえます。現在、7 社の信頼のおける供給者と、正式に契約しているか、もしくは非公式な同意を結んでいます。もしも必要な資材を取り扱う契約者が居ない場合には、購買部は潜在的な供給者に入札させ、最適な入札値を評価します。そして供給者が決まれば、購買部は要求された資材をその供給者に発注します。**

インタビューの中で、分析者は、プロセスに関連する実資料（？source material）のコピーを見せてもらう方が判り易いと気づくことが多かった。また、方針マニュアルや手続マニュアル、以前に開発した（IDEF3）モデル、購入要求書、供給者リスト、入札案内（solicitation）、入札評価報告書、発注書といった物の、適当な部分についてのコピーも、入手しておいた方が判り易いことも判明した。

このインタビューの中で、分野専門家から購入要求書のコピーを提供されたとする。分析者は、その発注要求書の下に、「要求者」、「会計責任者（Account Manager）」、「購入承認者（Purchase Authorization）」の 3 か所の署名欄があることに注目した。

以上の事を基にして分析者は改めて質問を行い、それに対して分野専門家は以下のように回答したとする：

> **資材の要求を行う際には、我々はまず購入要求書（purchase request）を準備します。この購入要求書の作成に必要となる情報には、品目、個数、納期（必要な場合）、購入の会計番号、購入理由（written justification for stated need）、要求者の氏名と署名があります。購入要求書を作成したら、次に購入要求者は会計責任者もしくはその代理人から購入の認可を得なければなりません。この会計責任者もしくはその代理人は、担当するプロジェクトの会計で処理される全ての購入に対して責任を負い、その購入の認可を行わなければなりません。そして会計責任者による認可が下りたら、次は承認者（authorization）の署名が必要となります。この時、潜在的な利益相反行為（conflict of interest）を防ぐ為、購入者と会計責任者もしくは購入承認者は、同一人物が兼任できないようになっています。全ての署名を適切な人物から得られたら、要求者は署名の入った購入要求書を、購買部門（Purchasing）へと提出します。購買部門はそれを基にして、要求された資材を発注します。購入要求書は発注後も、発注済として引き続き追跡されます。**

2 回目の記述は、1 回目の記述よりも詳細なものになっているが、分野専門家にとって重要だと思われたことから最初の記述に含められていた、供給者がどのようにして識別されているかという記述が、全て省略されている。

実際に、インタビューによってもたらされる記述の完全性は、以下のような要因に依存している：

（1）分野専門家が、インタビューに貢献しようとしている（もしくは貢献が許されてい

る）時間の総計

（２）インタビューアーの持つ、経験と専門分野に関する知識

（３）記述を行うプロセスに関して、分野専門家が持っている知識

分野専門家へインタビューを行う過程で、分析者は、プロセスに関する文書類（witttten documentation）を含む、最初の記述を取得（acquire）する事になる。記述を取得する目的は、分野専門家が考えているところのシステムの挙動（もしくは分野専門家の考える理想的な挙動）ではなく、システムの実際の挙動を表現することである。その為、分析者は相互関係のある事実を、プロセスの直接観察を伴ったインタビュープロセスから獲得して行かなければならない。

また分析者は、システムはこう動いていている筈だという、自分自身で前もって考えていた知識により、記述を勝手に補完してはならない。その為、分析者も分野専門家も、記述とは本質的に不完全なものであることを理解し、自分達の考える理想的な記述へと補完しようとする欲求を抑制することが重要となる。

５．２．２　データ識別の為の記述の分析

インタビューが終了したら、分析者は記録物と観察物を慎重に調査してゆかなければならない。この分析作業により、記述内に存在するオブジェクト、活動、事実、制約といったものが識別される。この作業はリスト作成プロセスとなる。

プロセスの記述を行っている時、個人は、プロセス内で発生するイベント（event）や活動を実際に記述する前に、プロセス内の重要なオブジェクトと、そのプロセス内での役割に注目（集中）することが多い。

以下に挙げるのは、記述内で識別されたオブジェクトのリストである：

資材（Material）	購買部門（Purchasing Department）	契約（Contract）
注文（Order）	潜在的供給者（Potential Supplier）	入札（Bid）
既存の供給者（Current Supplier）	要求者（Requester）	購入要求（Purchase Request）
会計責任者（Account Manager）	代理人（Backup）	プロジェクト会計（Project Account）
発注（Purchase Order）	選定供給者（Chosen Supplier）	

分析者は、次のような理由から、IDEF3 オブジェクトプールに、オブジェクトのリストを明確に記録しておくことが重要となる：

（1）記述獲得プロセスの後の段階で、幾つかのオブジェクトをプールから排除しても良い

（2）最初の分析から得られたオブジェクトのリストには、プロセスの主要オブジェクトが含まれている事が多い。主要オブジェクトとは、オブジェクトスキーム図の作成に欠かすことのできない重要なオブジェクトである。

オブジェクトの識別の次は、インタビューの記録（note）を調査して、資材発注の中で発生している活動／プロセスを決定して行く。重要な活動は、記述で UOB として表現されてれいる、活動、行動、プロセスといったものの候補となる。ただし、開発のこの段階では、活動の順序は重要ではない。この作業の主な目標は、あくまでも（以下に挙げるような）UOB の候補リストを作成する事である。このようにして作成された UOB 候補は、IDEF3 の UOB プールにリストとしてまとめられる。UOB のリストは不完全である場合が多いが、この段階においては不完全であることはあまり問題にはならない。

最初の記述から、以下のような UOB 候補が作成される：

（1）資材の要求
（2）潜在的供給者の識別
（3）既存の供給者の識別

（４）入札の要求
（５）入札結果の評価
（６）要求された資材の発注

また 2 つ目の記述から、以下のような UOB 候補が追加される：

（７）購入要求書の作成
（８）会計責任者から認可を得る
（９）購入承認者の署名を得る
（１０）署名の入った購入要求書の提出

インタビュー分析の最後の作業は、事実（fact）の識別とリスト化と、分野専門家によって記述されたプロセスに関係する制約（constraint）の識別である。

事実はとは、オブジェクトについて成された主張（assertion）である。また制約とは、あるプロセス内にあるオブジェクト間、もしくはプロセスそのものの間で、維持されていると知られている、分別された条件（distingushed condition）である。

制約の発生を識別するために、記録された文字通りの記述の中から、（全ての、だけ、といった数量詞だけでなく）否定的な単語（ではない、決して~ない、等）を探してゆく。
事実と制約のリストは、開発の初期段階では不完全であることが多い。
分野専門家との更なるインタビューや対話を重ねて行くことにより、このリストは次第に完全なものに近づいて行く。
この例によって識別された、最初のリストは、以下の通りである：

（1）現在、7 社の供給者と付き合いがある
（2）会計責任者もしくはその代理人以外はのみが、担当する会計に関する購入を認可できる。
（3）購入要求者は、その購入を認可もしくは承認する人と同一人物であってはならない。

５．３　プロセススキーム図をまとめる（Formulate Process Schematics）

オブジェクトや活動、事実、制約の識別を行う最初の作業が完成に近づいたら、IDEF3 プロセススキーム図（スキーム図単体もしくはセット）をまとめる準備が整ったと言える。

インタビューの際に記録された観察物は、プロセススキーム図の作成の基礎として使用される。またデータ分析作業で作成された UOB の候補リストは、この段階で行われる UOB の構築に使用される。そしてインタビュー時の記録物（note）から識別された事実と制約は、UOB 詳細説明の構築に使用される。

プロセススキーム図の作成作業には、大きく分けて以下の 2 つの大きな段階が存在する：

（１）UOB を正しい順序に構築し直す

（２）UOB 詳細説明の作成

５．３．１　最初のプロセススキーム図の作画（Layout Initial Process Schematic）

UOB の識別と、その間の順序を明確にする作業は、幾つかの段階を踏んで行われる：

（ステップ 1）プロセス記述の最も左に来る UOB を識別する。この例では「資材の要求」

（ステップ 2）その次に来る UOB を識別する。この例では、「既存の供給者の識別」もしくは「潜在的供給者の識別」となる。

ステップ 2 では、プロセスフローの分割（split）を伴う事になることから、この分岐フロー（diverging flow）を表現する為に、分散交差点（fan-out junction）を使用する必要がある。分析者は、分離を開始する交差点種類を決定しなければならない。この例では、購買部門は 2 つの選択肢の内の 1 つだけ実行可能であることから、XOR 交差点が使用されることになる。この段階で、図 5-1 のような部分的なスキーム図を作成しておくと便利である。

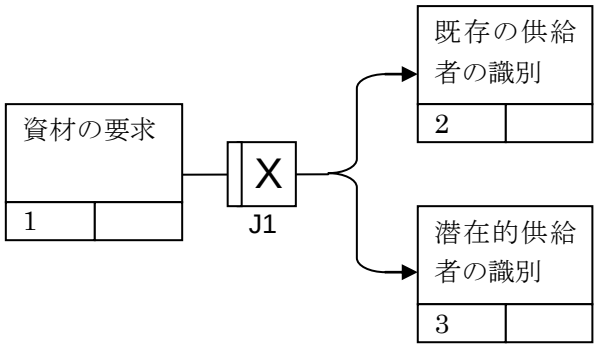


図 5-1　プロセススキーム図作成の最初の段階

もしもプロセスに分離が発生しなければ、分離が発生するまで、UOB 箱を順番につなげて行くことになる。分離が発生したら、分岐パスごとに別々にプロセスパスを作成してゆく。こうしたプロセスパスは、与えられた記述の文脈の中で、再び集束する場合もあればしない場合も有る。プロセスパスを作成する順番は、優先度の高い物から行ってゆく。

（ステップ 3）次に、UOB2 から始まるパスを作成する。このパスでは、UOB「入札の要求」、「入札結果の評価」、「要求資材の発注」と順番に続いて行く。その結果、スキーム図は図 5-2 のようになる。

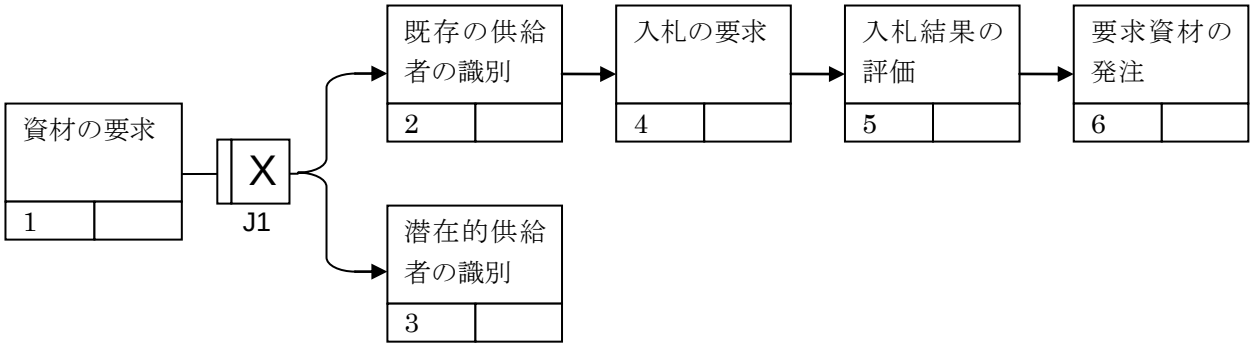


図 5-2　第 1 パスが完成したスキーム図

（ステップ 4）次に、図 5-2 の残りのパスも完成させる。この結果、プロセススキーム図は図 5-3 のようになる。この時、UOB への番号の割当ては、活動リストを基にして行われている。

2 つ目のパスもまた、要求資材の発注へと至ることになる。このことから、プロセスフローは集中（convergence）を伴う事になり、それを表現する為に集束交差点（fan-in junction）が必要となるが、分析者は、この集束交差点の適切な交差点種類を決定しなければならない。この例では、どちらか一方のパスだけが実行可能であることから、集束 XOR により、それぞれのパスを受け止める事になる。その結果、集束 XOR 交差点が使用されている。

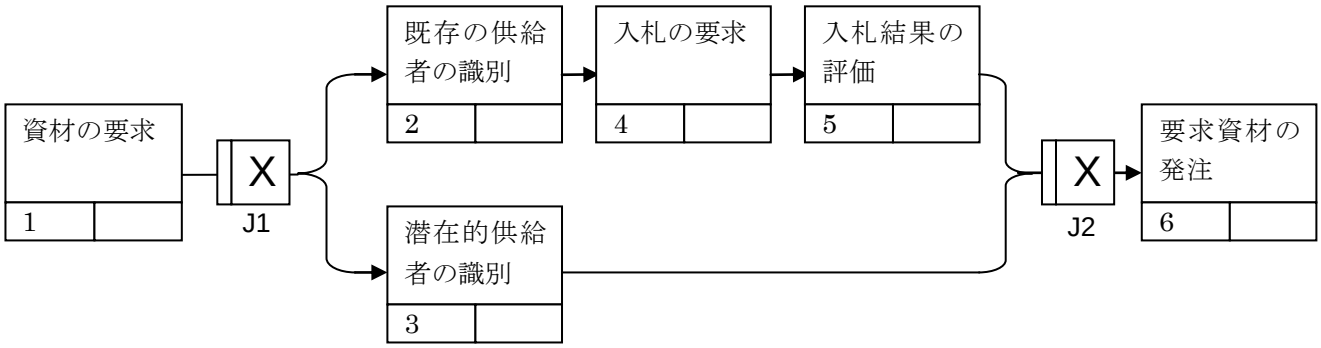


図 5-3　ほぼ完成に近くなったスキーム図

（ステップ 5）図 5-3 のようなスキーム図が完成した段階で、潜在的 UOB リストには、まだ 4 つの活動が残っている。この UOB7~10 は、資材要求 UOB に関する、分野専門家の記述の中に存在している。分析者によって、より詳細な階層でスキーム図の作成から開始し、後からこれを分解物の再構成を用いてスキーム図を単純化してゆく人も居れば、逆に最も抽象的な階層からスキーム図を作成し、後から分解物の作成によって詳細化したいと考えている活動と視点はクラス分けを行うだけに留めて置く方が向いている分析者も居る。
分解物を作成する事で、スキーム図を単純なものに保ちつつ、「資材の要求」活動がどのように行われているのか、ということについての詳細な記述を、別途追加することができる。
その結果、スキーム図は図 5-4 のようになる。

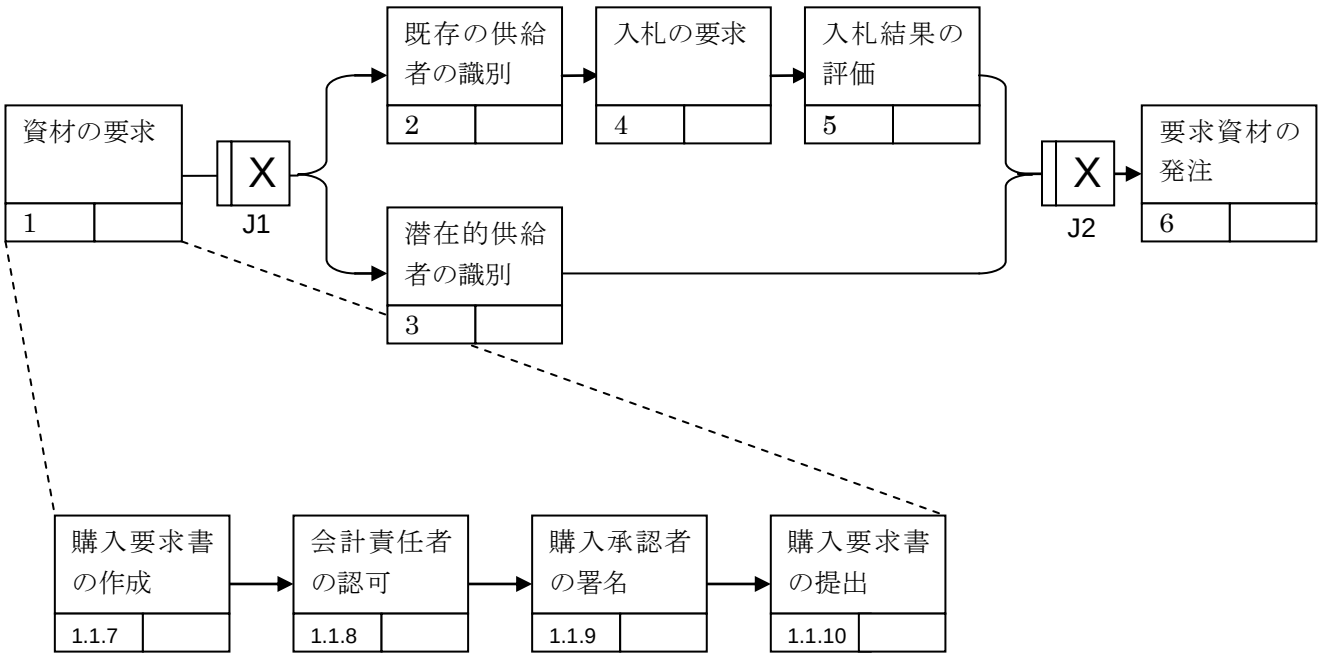


図 5-4　最初の批評前段階での、完成したプロセス記述スキーム図

５．３．２　詳細説明の作成（Develop Elaborations）

最初のプロセススキーム図が完成したら、次に図 5-5 から図 5-9 のように、各 UOB 毎の詳細説明を作成する。初めから完成したものを作る必要は無い。その理由の一つは、1 回目のインタビューでは、分析者は主にオブジェクトと活動に焦点を当てているからである。分析者が詳しくないプロセスや、大型で複雑なプロセスの作成においても同様である。

分析者がプロセス種類（process type）により詳しくなれば、1 回目のインタビューから、個々のプロセスについての情報を、より多く得る事ができるようになる。この精通の度合いは分析者の疑問へと反映され、1 回目のインタビューでも、同じプロセス種類の他のシステムと、プロセスの何が異なっているのかということを、分析者は判断できるようになる。また詳細説明の作成の際、システム種類についての分析者の個人的知識が、詳細説明に書かれる情報に影響を与えないように注意しなければならない。

詳細説明の作成順序は、あまり重要ではない。スキーム図の構成を手助けしてくれるような場合には、プロセススキーム図の作成と並行に詳細説明を作成しても構わない。しかしこの例では、プロセススキーム図が完成してから詳細説明の作成を開始している。

図 5-5 から図 5-9 は、この例での詳細説明である。この例では簡潔にする為に、これらの詳細説明に制約リストを含めていない。一般には、プロセススキーム図上の各リンクが、リンクされた個々の UOB の詳細説明の中で、制約実体を生成していることを忘れないでほしい。

使用先：	分析者： 山田 一郎　　日付：1995/2/8 プロジェクト：プロセス記述獲得 注記： 1 2 3 4 5 6 7 8 9 10　　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

UOB No.	UOB名： 資材の要求　　UOBラベル： 資材の要求
UOB 1	オブジェクト：資材 要求者 購入要求書 事実： 制約： 記述：

UOB No.	UOB名： 潜在的供給者の識別　　UOBラベル： 潜在的供給者の識別
UOB 2	オブジェクト：購買部門 購入要求書 事実： 制約： 記述：

背景設定参照：	記述項目：	フォーム形式：
シナリオ1	資材の要求UOB、 潜在的供給者の識別UOB	UOB詳細記述

図 5-5　UOB1 と UOB2 の詳細説明

使用先：	分析者： 山田 一郎　日付： 1995/2/8 プロジェクト：プロセス記述獲得 注記： 1 2 3 4 5 6 7 8 9 10　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

UOB No.	
UOB 3	UOB名： 既存の供給者の識別　UOBラベル： 既存の供給者の識別 オブジェクト：購買部門　資材 既存の供給者 事実： 現在、7社の供給者がある 制約： 記述：
UOB 4	UOB名： 入札の要求　UOBラベル： 入札の要求 オブジェクト：購買部門 潜在的供給者 事実： 制約： 記述：

背景設定参照：	記述項目：	フォーム形式：
シナリオ1	既存の供給者の識別UOB、 入札の要求UOB	UOB詳細記述

図 5-6　UOB3 と UOB4 の詳細説明

使用先：	分析者： 山田 一郎　日付： 1995/2/8 プロジェクト：プロセス記述獲得 注記： 1 2 3 4 5 6 7 8 9 10　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

UOB No.	
UOB 5	UOB名： 入札結果の評価　UOBラベル： 入札結果の評価 オブジェクト：購買部門　入札 潜在的供給者 事実： 制約： 記述：
UOB 6	UOB名： 要求資材の発注　UOBラベル： 要求資材の発注 オブジェクト：会計部門・財務部門　購入要求書　既存の供給者 購買部門　発注書　選定された供給者 事実： 制約： 可能な限り、既存の供給者を使用する 記述：

背景設定参照：	記述項目：	フォーム形式：
シナリオ1	入札結果の評価UOB、 要求資材の発注UOB	UOB詳細記述

図 5-7　UOB5 と UOB6 の詳細説明

使用先：	分析者： 山田 一郎　日付： 1995/2/8 プロジェクト：プロセス記述獲得 注記： 1 2 3 4 5 6 7 8 9 10　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

UOB No.	
UOB 7	UOB名： 購入要求書の作成　UOBラベル： 購入要求書の作成 オブジェクト：要求者　プロジェクト会計 購入要求書 事実： 制約： 記述：
UOB 8	UOB名： 会計責任者の認可　UOBラベル： 会計責任者の認可 オブジェクト： 要求者　プロジェクト会計　代理人 購入要求書　会計責任者 事実： 制約： 会計責任者もしくはその代理人のみが、担当する会計に対する購入を認可することができる。 要求者と、要求の認可者とが同一人物であってはならない。 記述：

背景設定参照：	記述項目：	フォーム形式：
シナリオ1	購入要求書の作成UOB、 会計責任者の認可UOB	UOB詳細記述

図 5-8　UOB7 と UOB8 の詳細説明

使用先：	分析者： 山田 一郎　日付： 1995/2/8 プロジェクト：プロセス記述獲得 注記： 1 2 3 4 5 6 7 8 9 10　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

UOB No.	
UOB 9	UOB名： 購入承認者の署名　UOBラベル： 購入承認者の署名 オブジェクト：要求者　プロジェクト会計 購入要求書 事実： 制約： 要求者と、購入承認者とが同一人物であってはならない。 記述：
UOB 10	UOB名： 購入要求書の提出　UOBラベル： 購入要求書の提出 オブジェクト： 要求者　購買部門 購入要求書 事実： 制約： 記述：

背景設定参照：	記述項目：	フォーム形式：
シナリオ1	購入承認者の署名UOB、 購入要求書の提出UOB	UOB詳細記述

図 5-9　UOB9 と UOB10 の詳細説明

５．３．３　分野専門家によるプロセススキーム図の批評（Review）

この例では、分析者は UOB9 と 10 の詳細説明が完成すると、評価を受ける為に、可能な限り早く分野専門家へと送付した。このようにして分野専門家から批評を受けることにより、作成されたスキーム図の構造が、シナリオについて分野専門家の持つ知識が上手く伝達されているかどうか評価されることになる。つまり、このプロセスにより、スキーム図と詳細説明の精度が検証されるのである。

分野専門家による批評の結果、獲得された記述に対して、何らかの変更が必要であることが判明することもある。その場合、オブジェクトや活動、事実、制約が新しく追加されたり、元のリストにあるそうした項目について、変更や削除が行われることになる。

分野専門家による IDEF3 記述の批評の結果、分析者は以下に挙げるような事を IDEF3 プロセススキーム図で変更しなければならない事に気が付いた：

（１）全ての完成した購入要求書に、購入承認者の署名が必要なわけではない

（２）直接費プロジェクトに関係する購入要求書は、資材購入を伴わない請求書の作成を防ぐ為に、会計と財務の購入承認者の署名が必要である。

（３）購入要求書の全ての必要な記入事項が要求者によって埋められていなければ、会計責任者による認可は成されない。

批評とそれに伴うインタビューとが終わると、新しく得られたデータを評価し、リストを更新する。それにより追加データが以下のような形式で記述へと織り込まれた。

リストに追加されたオブジェクト：

- 会計部門、財務部門
- 直接費プロジェクト（direct projects）
- 間接費プロジェクト（indirect projects）

追加された事実と制約：

- 全ての完成した購入要求書に、承認者の署名が必要なわけではない

・直接費プロジェクト関連の購入要求書には、承認者の署名が必要である
・購入要求書の全ての項目が正しく記入されていなければ、認可されない

分野専門家によって提案された、追加データと変更は、プロセス記述（スキーム図と詳細説明フォーム）へと織り込まれる。その結果、プロセススキーム図は図 5-10 のようになった。

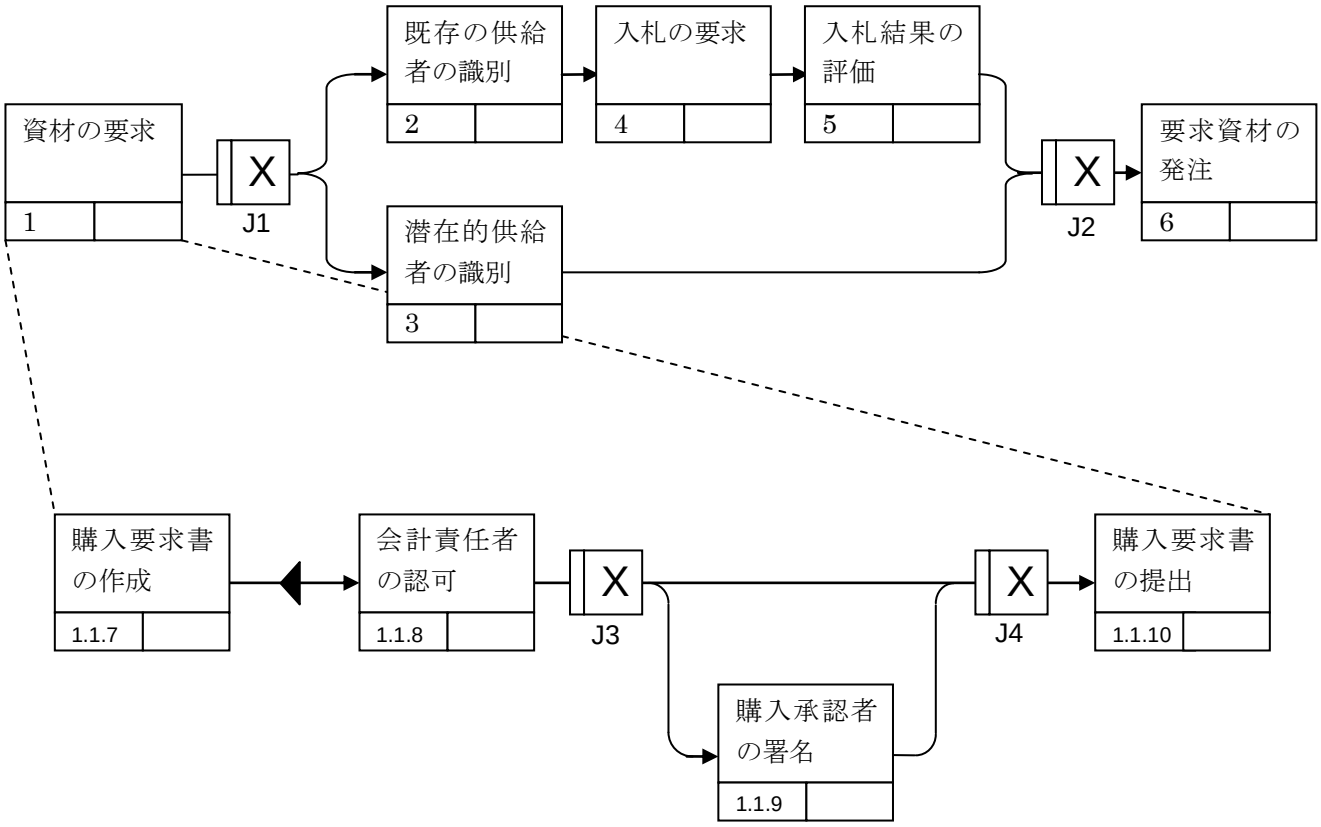


図 5-10　最終的なプロセススキーム図

UOB「資材要求」の分解物で、2 か所の構造変更が行われている。

1 つ目の変更は、必要な書類項目が要求者によって完成されていなければ、会計責任者は購入要求書を認可しない、という制約が、制約付き順序リンクとして追加されていることである。これにより、UOB「会計責任者の認可」の全てのインスタンスは、UOB「購入要求書の作成」インスタンスによって、常に先導（preceded）されることになるのである。

2 つ目の変更は、全ての購入要求書に、購入承認者の署名は必要でないという事実を表現した交差点を導入した事である。

最終的なスキーム図（図 5-10）の中で、交差点 J3 に関するロジックについて、より詳細な説明が必要になるが、これは交差点 J3 の詳細説明（図 5-11）を作成する事で対応する。この詳細説明フォームでは、ラベル欄は単純に交差点種類のみを識別している。そして番号欄には、交差点に割り当てられた番号を記入する（J3)。

交差点詳細説明フォームは、その交差点に関する決定ロジックを明確に定義できるような作りになっている。XOR 交差点の場合には、交差点から個々のパスを選択する際の決定ルールを完全に記述する事が可能である。

使用先：	分析者： 山田　太郎　　日付： 1995/2/8 プロジェクト：プロセス記述獲得 注記： 1 2 3 4 5 6 7 8 9 10　　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

Junction No. J1	交差点種類： XOR オブジェクト： 事実： 既存の供給者が存在する場合には、会社は常にその供給者から物品を購入する。 必要な物品を扱う既存の供給者が存在しない場合には、購買部門は潜在的供給者の識別と、 それらによる入札を行う。 制約： 記述：
Junction No. J3	交差点種類： XOR オブジェクト： 事実： 資材購入を伴わない直接費契約場合が存在する。 全ての購入要求書に、購入承認者の署名が必要なわけではない。 制約： 直接費プロジェクトには、購入承認者の署名が必要である。 記述： 資材購入を伴わない契約に対して請求書を発行する事を防ぐ為に、直接費プロジェクトでは 会計責任者による購入の認可の後、要求者は会計部門と財務部門の購入承認者から署名を もらわなければならない。

背景設定参照： シナリオ1	記述項目： XOR交差点J1、J3	フォーム形式： 交差点詳細記述

図 5-11　交差点詳細説明の例

このプロセス記述へのもう一つの追加物は、リンクの１つ（図 5-10 参照）に対する詳細説明である。このリンク詳細説明は、今回のような単純な場合には必ずしも必要な物ではないが、リンク詳細説明というものが、どのように特定のリンクと関連付けられているのかを説明する為に、敢えて作成している。対象となるリンクには、読者がリンク詳細説明とそのリンクとを結び付け易い様に、番号が割り振られている。このリンク詳細説明には、特定の UOB や参照の間に存在するリンクの、関連情報が含まれている。

使用先：	分析者：山田　太郎　　日付：1995/2/8 プロジェクト：プロセス記述獲得 注記：　1 2 3 4 5 6 7 8 9 10　　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

Link No.	パス番号	始点：	終点：
PL 6	PL6.1	会計責任者の認可（UOB8）	「何もしない」
	PL6.2	会計責任者の認可（UOB8）	購入承認者の署名（UOB9）

オブジェクト：購入要求書

事実：

制約：　(PL6.2) こうした承認の必要な購入要求書は、まず会計責任者もしくはその代理人によって認可され、署名されていることを確認しなければならない。

記述：

背景設定参照：	記述項目：	フォーム形式：
分解物　1.1	順序リンク　6	順序リンク詳細説明

図 5-12　順序リンク詳細説明ドキュメント

５．４　オブジェクトスキーム図をまとめる（Formulate Object Schematic）

プロセスに関係するオブジェクトを、詳細に特徴付けするには、オブジェクトスキーム図が便利である。オブジェクトスキーム図は、一般的には、プロセス記述の重要なオブジェクトについてのみ、作成されるものである。オブジェクトスキーム図を作成する事により、プロセスを異なった視界、つまりはオブジェクト中心視界から、見て行くことが可能となる。オブジェクトスキーム図はプロセススキーム図の後に作成されることが多いが、オブジェクトスキーム図の作成から開始するユーザーも存在する。

５．４．１　興味対象となるオブジェクトの選択（Select Objects of Interest）

オブジェクトスキーム図をまとめるには、オブジェクトもしくは興味対象となるオブジェクトを選択することから始めなければならない。発注書プロセスの場合、購入要求書が重要なオブジェクトとなる。この購入要求書を、記述されているプロセスの中で幾つかの状態を通して変移するものとして概念化すると、便利である。こうして作成されたオブジェクトスキーム図は、購入要求書が最初に作成されてから、最終的に発注書が発行されるまでのプロセスを、オブジェクト中心視界から調査する際の、興味対象を示してくれることになる。

以下に挙げる例は、こうした考えの下での、オブジェクトスキーム図の作成プロセスを説明したものである。

５．４．２　オブジェクト状態の識別（Identify Object States）

焦点となる主要なオブジェクトが選択されたら、分析者はオブジェクトスキーム図の作成を開始するが、最初の作業はオブジェクト状態の候補を識別である。
この例での興味対象となる重要なオブジェクトの購入要求書は、プロセスを通してその状態をどのように変化させているかを、更に詳しく調査する必要がある。顧客（訳注：この例では IDEF3 開発を依頼した業務責任者）によって提示された最終的な状態では、購入要求書は発注書へと置き換わる。発注書にも色々な状態が発生し得るが、顧客にとって発注書の様々な状態は対象とならない。

分析者は、まず最初に候補となるオブジェクト状態の識別に関する作業から開始する。情報源によっては、直接的に状態を識別できることもあるが、分野専門家に状態を明確に区別してもらった方が良い場合もある。情報源となるものには、分野専門家によって提供される生の記述や、UOB 名の候補、オブジェクトそのもの、UOB 詳細説明フォーム等がある。オブジェクトがプロセスの中で採り得る状態変化を識別する為には、分析者は、オブジェクト状態の候補名の吟味を、分野専門家と共に行う必要がある事が多い。

インタビューから得られたデータを批評する事により、分析者は以下のようなオブジェクト状態候補のリストを作成した：

購入要求書：未記入
購入要求書：記入済
購入要求書：認可済
購入要求書：認可済だが承認が必要
購入要求書：承認済
購入要求書：提出済
発注書：発行済

このリストは、論理的に一致した状態の識別や、名前の改良、そして無名だったか認識されていなかった状態の識別といったことを通して、変化し続けてゆく。

５．４．３　最初のオブジェクトスキーム図の作画（Layout Initial Object Schematic）

識別されたオブジェクト状態を、変移スキーム図へと構成して行くには、次のようなステップを経る：

（ステップ１）まず最初に、最初の状態、もしくはスキーム図で最も左に来る状態を識別する。この例では、最も左に来る状態は「購入要求書：未記入」となる。

（ステップ２）次に、最初の状態が変移可能な状態（複数可）を識別する。この例では「購入要求書：記入済」となる。

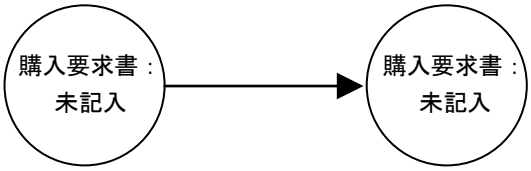


図 5-13　最初の変移スキーム図

（ステップ 3）後は、ステップ 1 と 2 を、全ての発生し得る状態変移が識別されるまで、繰り返して行く。

この時、全ての変移パスを一気にスキーム図に描きだそうとするのではなく、1 本の変移パス毎に識別、記述して行くのが一般的である。この例では、パスが初めて分岐する点は、購入要求書が会計責任者によって認可された時に発生する。この時点で、購入要求書は承認が要求されることになるが、もしも承認が不要である場合には、そのまま購買部門へと提出されることになる。以上により、図 5-14 と図 5-15 の変移スキーム図が作画される。

図 5-14　承認が不要な場合のパスを示した変移スキーム図

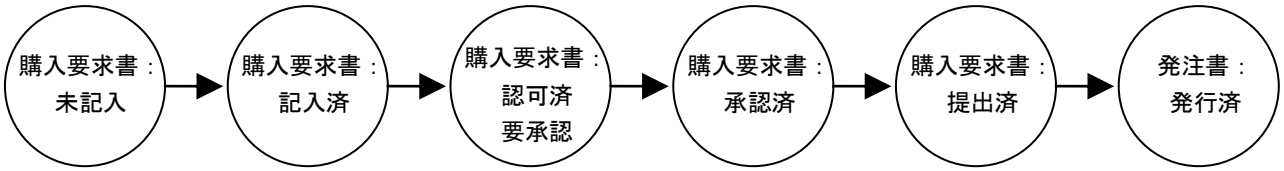


図 5-15　承認が必要な場合のパスを示した変移スキーム図

５．４．４　必要に応じた交差点の追加

次に、適切なロジック交差点を導入する事により、2 本の変移パスを 1 つのスキーム図へと合体させる。

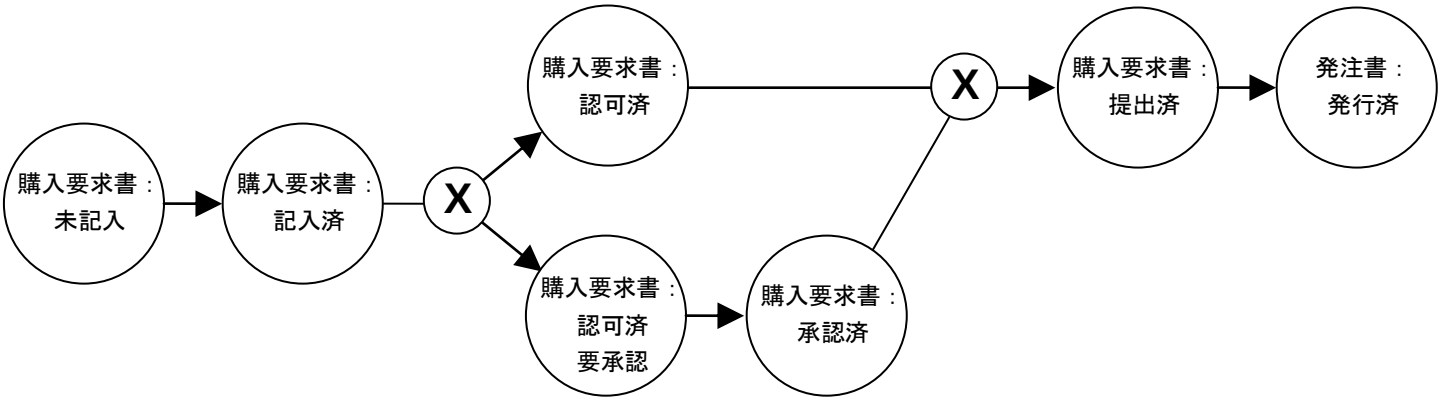


図 5-16　　図 5-14 と図 5-15 を合体した変移スキーム図

５．４．５　参照の取付け（Attach Reference）

発生し得る変移パスを識別し、それらを状態変移パスの分岐を反映させつつ統合したら、次に関係する UOB やシナリオ、他の変移スキーム図といったものへの参照を識別し、スキーム図上の適切な場所に取り付けて行く。その結果、図 5-17 のようになる。

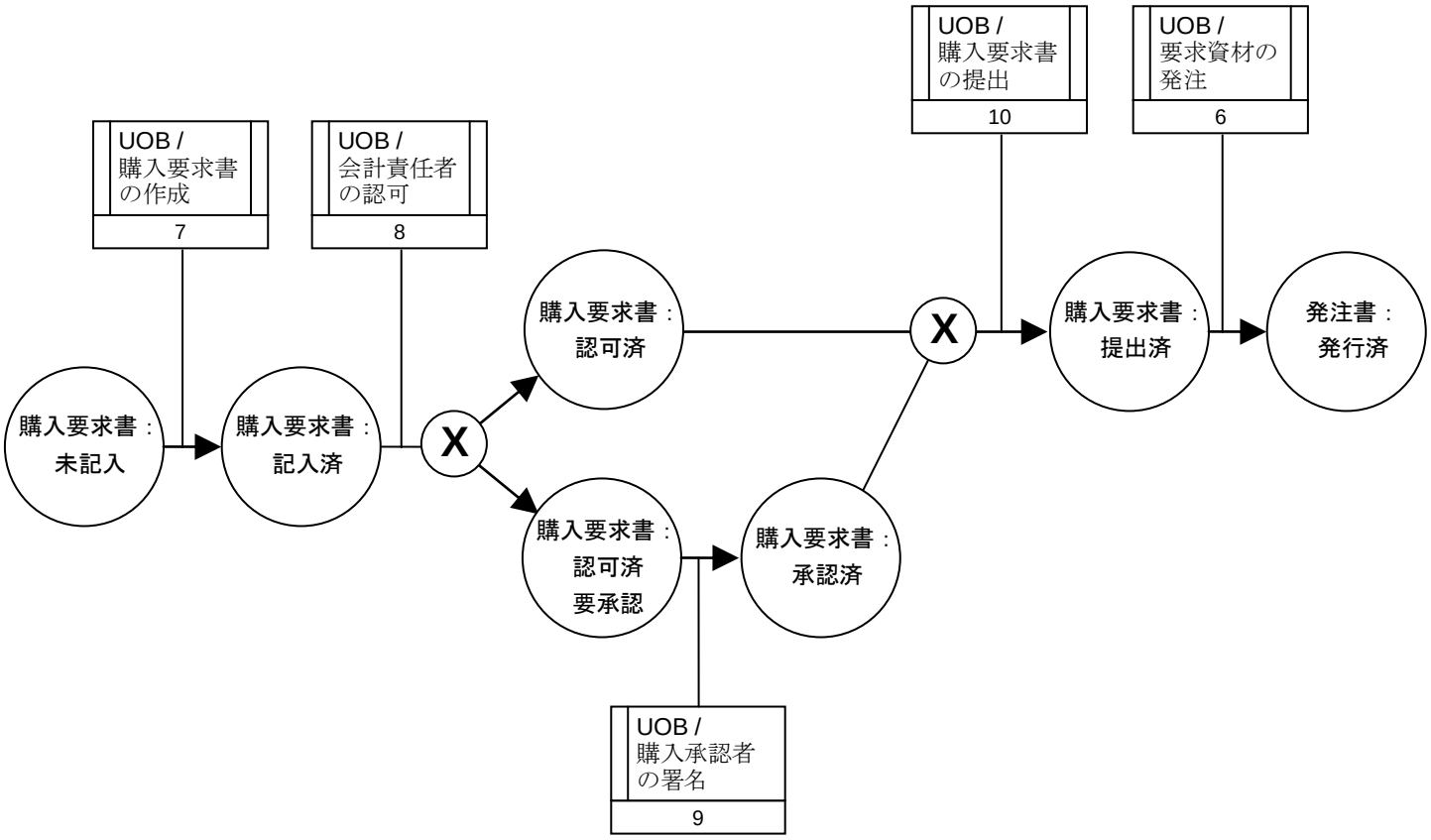


図 5-17　　最初の批評を行う前の段階での、完成したスキーム図

５．４．６　詳細説明の作成

最初の変移スキーム図が完成したら、識別されたオブジェクト状態の特徴付けをより完全なものとする為に、図 5-18 から図 5-20 に示したような詳細説明を作成する。詳細説明を作成してゆく際、分析者は自分の持っているシステム種類に関する知識が、詳細説明に配置される情報に影響を与えないように、注意する必要がある。

詳細説明を作成する順番は、必ずしも重要ではない。オブジェクトスキーム図の仕上げと、詳細説明の作成を並行に実施した方が、便利な場合も有る。特に、変移スキーム図とその詳細説明とを同時に作成することにより、前もって識別できていなかった状態の発見に繋がてゆくことことができる。

しかし今回の例では、変移スキーム図が完成した後で、詳細説明が作成されている。

使用先：	分析者： 山田　太郎　　日付： 1995/2/8 プロジェクト： プロセス記述獲得 注記： 1 2 3 4 5 6 7 8 9 10　　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

Object State No.

os 2

オブジェクト状態名： 購入要求書：記入済

ラベル： 購入要求書：記入済

オブジェクト状態からの変移： 購入要求書：記入済

オブジェクト状態への変移： 購入要求書：認可済
購入要求書：認可済　要承認

事実：

制約：

状態条件： 購入要求書フォームには、品目、個数、納期（必要な場合）、購入に際して発番される会計番号、購入理由、要求者の氏名と署名が含まれていなければならない。

退出条件：

その他：

記述：

背景設定参照： シナリオ1	記述項目： オブジェクト状態2 - 購入要求書：記入済	フォーム形式： オブジェクト状態詳細記述

図 5-18　オブジェクト状態「購入要求書：記入済」の詳細説明

使用先：	分析者： 山田　太郎　　日付： 1995/2/8 プロジェクト： プロセス記述獲得 注記： 1 2 3 4 5 6 7 8 9 10　　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

Object State No.

OS 4

オブジェクト状態名： 購入要求書：認可済　要承認

ラベル： 購入要求書：認可済　要承認

オブジェクト状態からの変移： 購入要求書：認可済

オブジェクト状態への変移： 購入要求書：承認済

事実： 全ての購入要求書に、購入承認者の署名が必要なわけではない。

制約：

状態条件： 直接費プロジェクトの予算による購入。
購入要求書に、会計責任者もしくはその代理人による署名がある。

退出条件：

その他：

記述：

背景設定参照：	記述項目：	フォーム形式：
シナリオ1	オブジェクト状態4 - 購入要求書：認可済　要承認	オブジェクト状態詳細記述

図 5-19　オブジェクト状態「購入要求書：認可済　要承認」の詳細説明

使用先：	分析者： 山田　太郎　　日付： 1995/2/8 プロジェクト： プロセス記述獲得 注記： 1 2 3 4 5 6 7 8 9 10　　改正：	X	検討中	批評者：	日付：
			仮図		
			承認済		
			出図		

Object State No.

OS 5

オブジェクト状態名： 購入要求書：承認済

ラベル： 購入要求書：承認済

オブジェクト状態からの変移： 購入要求書：認可済　要承認

オブジェクト状態への変移： 購入要求書：提出済

事実： 全ての購入要求書に、購入承認者の署名が必要なわけではない。

制約：

状態条件： 購入承認者が購入要求書に署名をしている。
購入要求書の承認者は、会計認可者と同一人物であってはならない。

退出条件：

その他：

記述：

背景設定参照：	記述項目：	フォーム形式：
シナリオ1	オブジェクト状態5 - 購入要求書：承認済	オブジェクト状態詳細記述

図 5-20　オブジェクト状態「購入要求書：承認済」の詳細説明

５．４．７　分野専門家によるオブジェクトスキーム図の批評（Review）

プロセススキーム図と同様に、オブジェクトスキーム図とそれに関係する詳細説明の正確性の確認（confirm）を、分野専門家の検証（validation）を通して行う。

この例の場合、変移スキーム図の批評により、分野専門家は、状態変移の中に要求が却下された状態が含まれていない事に気づいた。初期の段階で分野専門家によって提示された記述は一般的な場合を表わしたものであり、要求書が取り下げられたり、承認者によって却下された状態が含まれていなかった。そこで分野専門家は、以下のような 2 つの全く新しい状態の導入を提案した：

（1）購入要求書：不認可
（2）購入要求書：未承認

また分野専門家は、
オブジェクトの識別が準備されたものを通した
変移と、
オブジェクトが実際に全く異なったオブジェクトへと変移してしまう
変移も、識別した。

このような分野専門家のコメントを織り込み、分析者が修正したスキーム図が図 5-21 である。

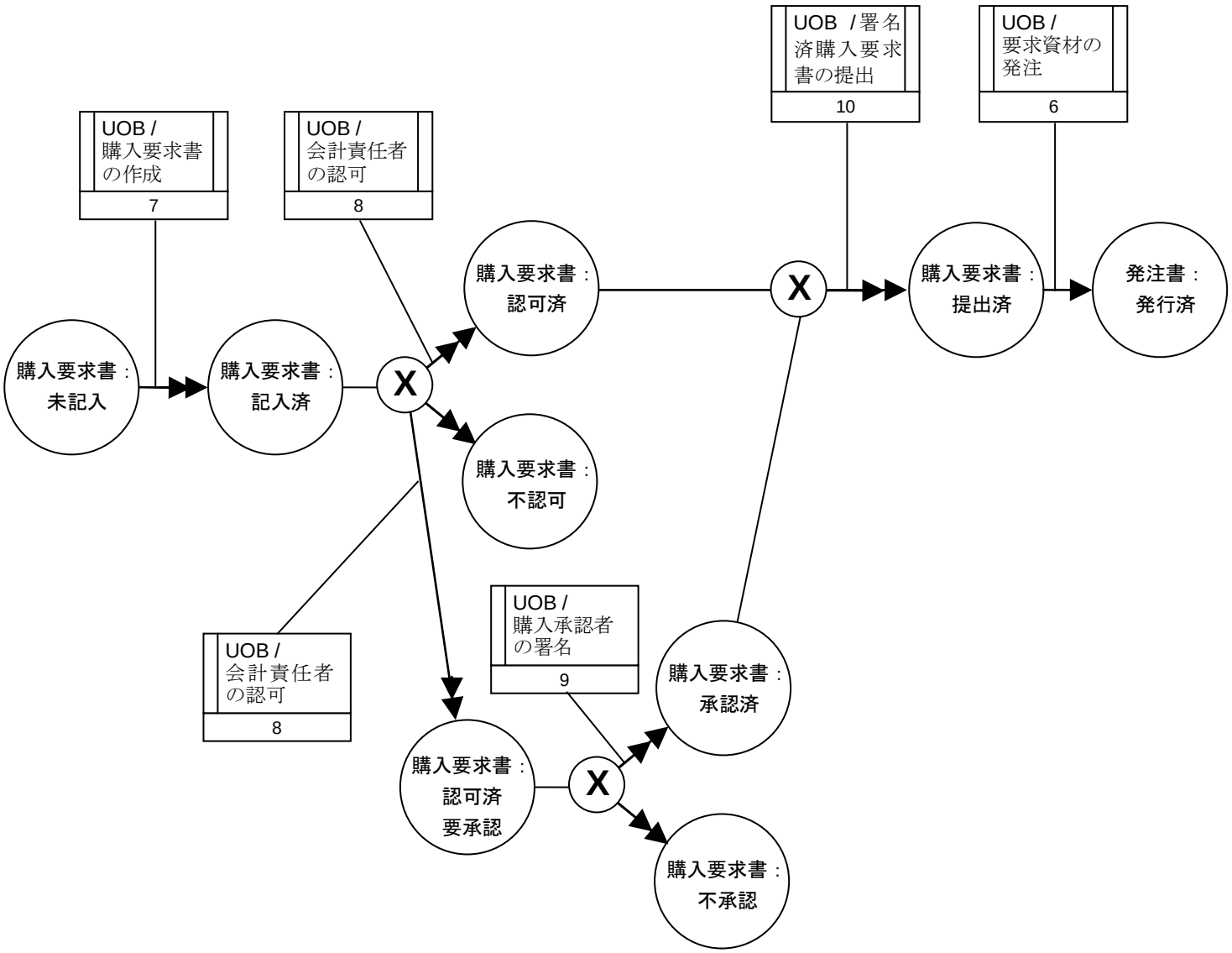


図 5-21　完成した変移スキーム図

必要によっては、変移スキーム図に背景設定情報を追加する事もある。例えば、直接費プロジェクトに関係する購入要求書には購入承認者の署名が必要である、という分野専門家の記述などが挙げられる。更に、会計責任者やその代理人は、担当するプロジェクトに関係する全ての要求書の処理を行わなければならない、という制約に関する議論についての記述も当てはまる。こうした情報は、スキーム図に直接記入される。そうした追加を行ったオブジェクトスキーム図が、図 5-22 である。

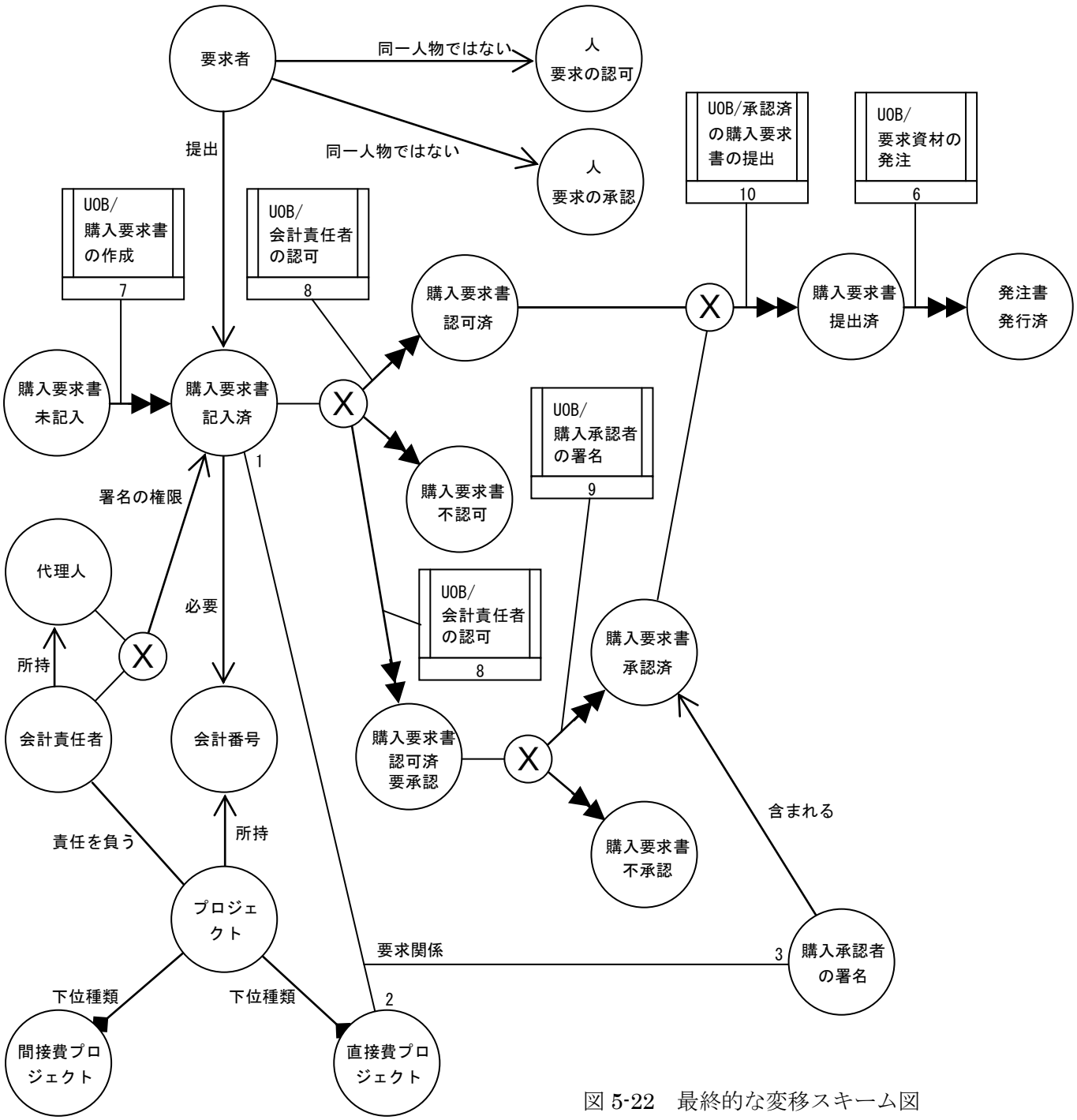


図 5-22　最終的な変移スキーム図

第 6 章　IDEF3 プロセス記述の読解

IDEF3 プロセス記述を作成する主な目的は、ある特定のシステムもしくは組織が、どのように動いているのかということを正確に表現する事である。IDEF3 プロセス記述によって、ある特定のシナリオに関係するプロセスフローとオブジェクト状態変移についての事実に関する記述を獲得することができる。

IDEF3 記述の批評者は、記述の作成は行わないが、記述内の事実を検証（validate）しなければならない。また IDEF3 記述の読者は、他人が作成した記述から、知識を得なければならない。そこでこの章では、IDEF3 プロセス記述の一般的な読み方と理解の仕方について説明して行く。

IDEF3 スキーム図は、それがプロセススキーム図であっても、オブジェクトスキーム図であっても、スキーム図の最も左側の要素から読み始めるのが普通である。慣習的に、スキーム図は左上から読み始められている。

記述されたシナリオの概略を理解するためには、スキーム図上の流れを脳内ウォークスルー（mental walkthrough）で辿ってゆく。これを行う中で、読者は UOB の順序関係やロジック的な位置関係に気づかされることになる。こうした読み方をすることで、読者はシステムの概要を理解するのである。そして個々の UOB やその間に張られたリンクを、それらについての詳細説明や記述と共に読み込むことで、記述をより詳細に知ることができる。またスキーム図上に存在するロジックをシステム的に研究する事で、IDEF3 プロセス記述を包括的に理解することができる。

６．１　記述の読解ステップ（Description Reading Steps）

システムについて収集された事実は、IDEF3 プロセス記述の中の、プロセススキーム図やオブジェクトスキーム図、そしてそれらに関する詳細説明言語で書かれた宣言文のセットとして構成されている。このような構成になっている IDEF3 スキーム図の読解方法は、読者と、読者が読み取りたいと望んでいる情報の総量とによって変化する。

スキーム図の読解プロセスは、場合に依って非常に変化することから、厳密なアルゴリズムの形で読解プロセスを表現する事は困難である。例えばある人は、最初にスキーム図

を走り読みし（scan)、それからスキーム図の理解が容易になるようにロジック部品へと分解して行く。またプロセススキーム図では、UOB やリンクの間の関係性を理解する為に、ある限定的な範囲の中でロジック的にグループ分けをすることもある。スキーム図のより小さな部分の意味を理解できたら、それらを交差点やロジックと結び付ける事により、より大きな範囲の意味が理解できるようになる。

IDEF3 プロセス記述の読解方法の概略は、以下のようになる：

（1）IDEF3 プロセス記述の、目的（purpose）の宣言（statement)、作業範囲（scope）の宣言、そのシナリオが記述された目的（objective)、そして視点（viewpoint）を、それぞれ注意深く読み解く。

（2）スキーム図上にある個々の要素（UOB やリンク、交差点、オブジェクト状態等）を、左上から流し読みすることで、書かれている内容の大まかな印象を得ると共に、記述の構造とロジックとを大まかに理解する。

（3）スキーム図上にある要素を、左上から順番に、ロジックグループやロジック構造へと分割して行く。ロジックグループとは、システム的な批評が容易になるようにスキーム図を分割した際に、その部分を構成している要素を集めたものである。こうしたグループ分けは、殆どの場合、プロセススキーム図の場合であればプロセスパスと、オブジェクトスキーム図の場合であれば変移パスと一致し、そしてこうしたパスはまた、ロジック的なサブグループを内部に持っている。記述をより良く理解するためには、こうしてできたグループとサブグループを、全てのダイヤグラムが分割されるまで、同じ様に分割してゆく。

（4）最も左手にあるグループの最初の要素から開始して、以下のようなガイドラインに従い、左から右へとスキーム図を読んでゆく：

①プロセススキーム図の場合、UOB とその詳細説明を読む。オブジェクトスキーム図の場合は、オブジェクト状態とその詳細説明を読む

②プロセススキーム図の場合には順序リンク、オブジェクトスキーム図の場合は変移リンクを、リンクに表示された制約やリンクの詳細説明の情報に注意しながら、詳細に調査する（examine)。

③選択したグループの範囲内にある、全ての参照と注記を調査する（study）

④基本的なグループ単位で、脳内ウォークスルーを実行する。

⑤交差点に行き当たったら、パスの選択に課せられた制約に注意しながら、それぞれのパスを辿ってゆく。

より慣れた読者は、もっと簡単な読み方をしている。この単純な読み方については、次の節で説明する。

６．２　IDEF3 プロセス記述の速読（quick reading）の例

IDEF3 プロセス記述に慣れた読者は、これから説明するものに近い方法で、プロセスを追っている。しかし、熟練した読者にこのような速読が可能である理由は、彼らが十分に経験を積んでいるからであり、その為、速読の方法は人に依存している。

ここでは、スキーム図の読み方の一例を紹介して行く。この例で紹介するスキーム図の読み方の概要（？outline）は、プロセススキーム図とオブジェクトスキーム図のどちらであっても、殆ど変らないやり方で、全ての分解物のに対して繰り返されてゆく。一般的には、親のスキーム図の読解と理解が終わった後で、分解物の読解を行うのが普通である。

６．２．１　大局（The Big Picture）

記述を読解するプロセスの中で最も重要となるステップは、（これから読もうとしている）記述された実在の状況が、一体どういうものであるのかという概略、つまり大局（big picture）を理解する事である。

この大局は、IDEF3 プロセス記述の中の目的の宣言や、作業範囲（scope）の宣言、それぞれのシナリオが記述された目的、そして視点を読解することによって得る事ができる。記述の中のこうした部分は、スキーム図の作業範囲（scope）をまとめ上げ、（特に、記述されたプロセスに慣れ親しんでいる）読者に、最上層のスキーム図を読むことで何が得られ

るかについて伝えてくれる。またこうしたものは、この記述は、どれくらい詳しく説明されているかと言う事についても知らせてくれる。

６．２．２　スキーム図の流し読み（Scan the Schematic）

スキーム図を左から右に流し読みする事により、読者はシナリオをより詳細に知ることができる。またこの流し読みにより、スキーム図上に描かれている個々の要素（UOB やリンク、交差点等）の詳細についても知ることができる。流し読みを行う目的は、スキーム図をより深く調査することではなく、むしろ記述されているプロセスについての大まかな印象を得て、シナリオのロジックの流れ全体を理解することである。

６．２．３　記述の理解

このステップで読者は、シナリオ、オブジェクト、スキーム図要素の分解物といったものと関連するスキーム図を、より詳細に理解して行く。この作業は、最も個人差が出て、そして最も時間のかかる情報伝達プロセスの 1 つである。この際、スキーム図を理解可能な部分へと分割すると、作業が容易になる。

このようなスキーム図の分割方法には、何が良くて何が悪いかという決まった物は無いが、スキーム図の構造に沿って分割すると読みやすくなる。この後で、スキーム図の分割のやり方をプロセススキーム図の例を用いて説明する。この例は、あるオブジェクトスキーム図に適用された同一のコンセプトについて説明したものである。

図 6-1 のような IDEF3 プロセススキーム図は、図 6-2 の円のように分割することが出来る。

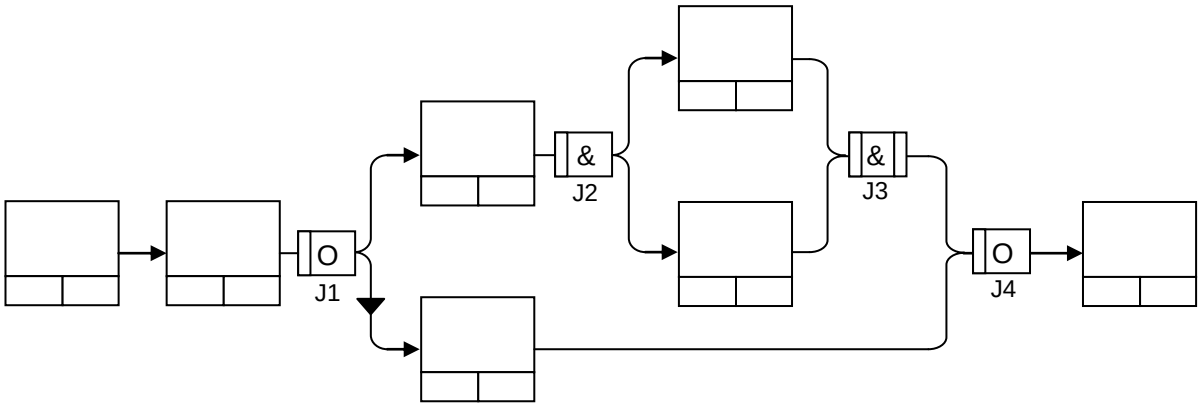


図 6-1　IDEF3 プロセススキーム図の例

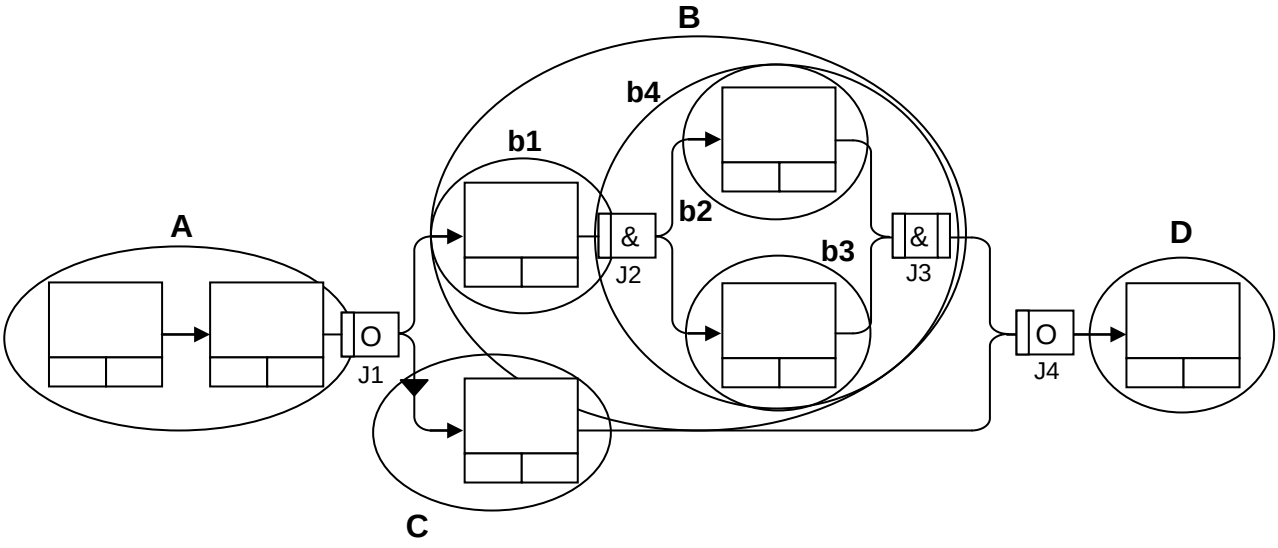


図 6-2　図 6-1 のスキーム図の分割例

図 6-2 で、スキーム図は A、B、C、D の 4 つのグループへと分割されている。これらの 4 つのグループは、4 本のプロセスパスを表現している。そしてこれらのグループを詳細に調査することにより、B は b1、b2、b3、b4 の 4 つのサブグループへと更に分割できることが判明した。望ましいグループ・サブグループへの分割が終了したら、それぞれの構造について分析を開始して行く。

図 6-3 のように分割したグループに 1 から 8 までの番号を付けたとすると、読者は、最も左端にはるグループ 1 から開始して、左から右に向かって、以下に挙げる活動をグループ毎に繰り返し実行して行くのが一般的である。

（1）UOB とその詳細説明の読解
（2）リンクを調査し、リンクの詳細説明で発見された情報を記録（note）する
（3）選択したグループの範囲内にある全ての参照と注記（note）を考慮する

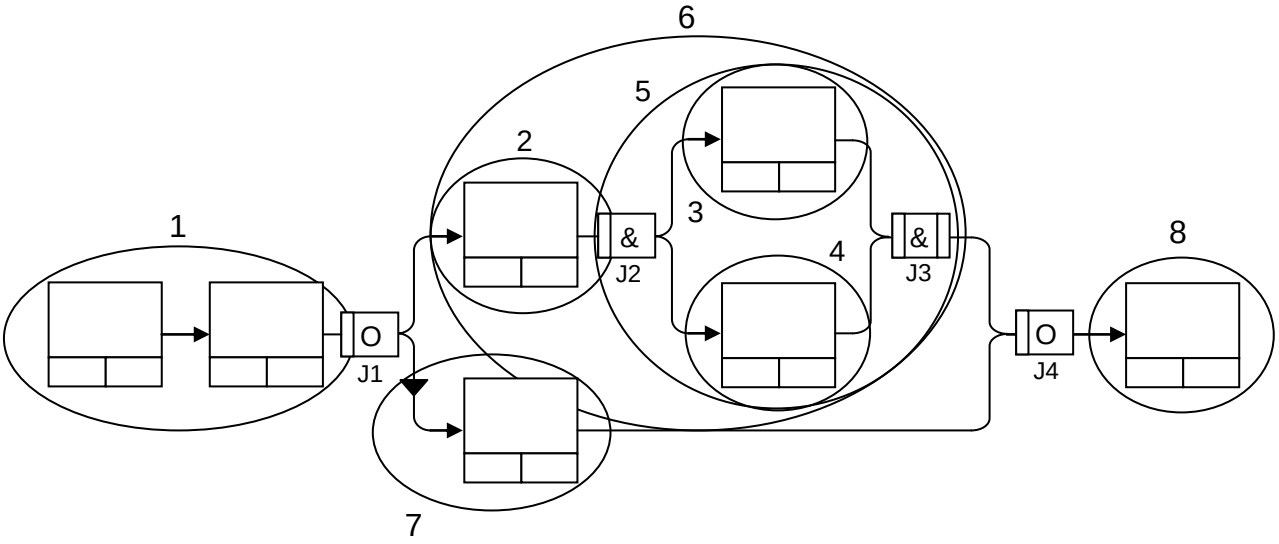


図 6-3　グループの分析

グループ 1 の理解が終わったら、読者はグループ 6 か 7 のどちらかの調査を行う。交差点 J1 については、この段階では直ぐに考慮しなくても良い。グループ 6 の調査を開始したら、そのサブグループ 2 と 5 について、それぞれ分析を行う。グループ 2 の分析とは、ここでは 1 個の UOB とその詳細説明を調査するということである。グループ 5 は複雑なグループであり、グループ 3 と 4 へと更に分割することができる。グループ 3 と 4 の調査が終わったら、グループ 6 全体について分析を行う。この分析では、グループ 2 からグループ 3、4 へと向かう J2 分散交差点を含んだグループロジックを理解することになる。交差点 J2 を理解するために、読者は交差点から出ている 2 本のパスを詳しく調査し、その 2 本のパスへと流れる条件に注目する。一般的に交差点のロジックは、その交差点に出入りしている全てのパスを辿り、そしてそれらのパスが選択される条件に注目する事によって、分析が行われる。グループ 5 の研究は、集束交差点 J3 のロジックの分析によって完了する。

その次に、読者はグループ 7 の理解を試みる。グループ 7 の分析が完了したら、続いて分散交差点 J1 の分析が行われる。読者はグループ 1 から始まるプロセスの脳内ウォークスルーを実行し、その交差点での分岐条件と各パスを統括する条件に注目しする。この場合読者は、関係する順序リンク詳細説明を綿密に調査する必要性を示している、制約順序リンクの存在に注目した。

次に、グループ 6 と 7 から発生したプロセスパスが合流する集束交差点 J4 について分析しなければならない。読者は交差点 J4 のロジックの分析に関係する脳内ウォークスルーを実行し、2 本のパスが合体する条件に注目する。

そして最後に、UOB8 とその詳細説明を読解し、UOB8 に取り付けられた参照を考慮するとにより、グループ 8 の分析が行われる。これを終えると、読者はダイヤグラム全体で、脳内ウォークスルーを実行したいと望むようになるだろう。そこでスキーム図の左端から開始し、グループ 8 に至るまで、全ての交差点を考慮しつつウォークスルーを実行する。

オブジェクトスキーム図の脳内ウォークスループロセスは、プロセススキーム図のそれよりも非常に単純なものになっている。図 6-4 のような IDEF3 オブジェクトスキーム図の例の場合、図 6-5 のように分割する事が可能である。

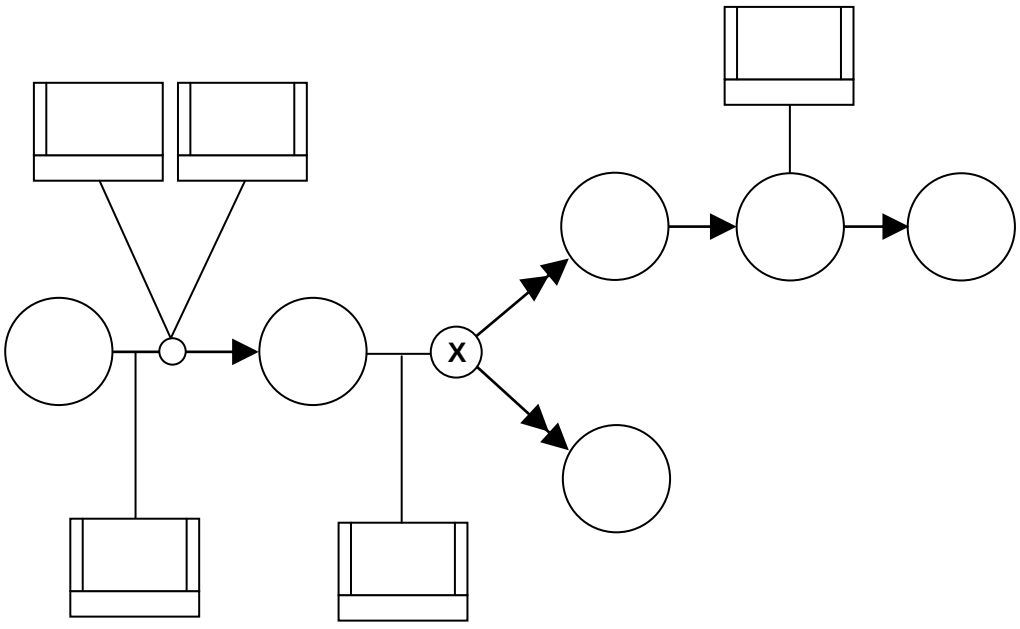


図 6-4　IDEF3 オブジェクト変移スキーム図の例

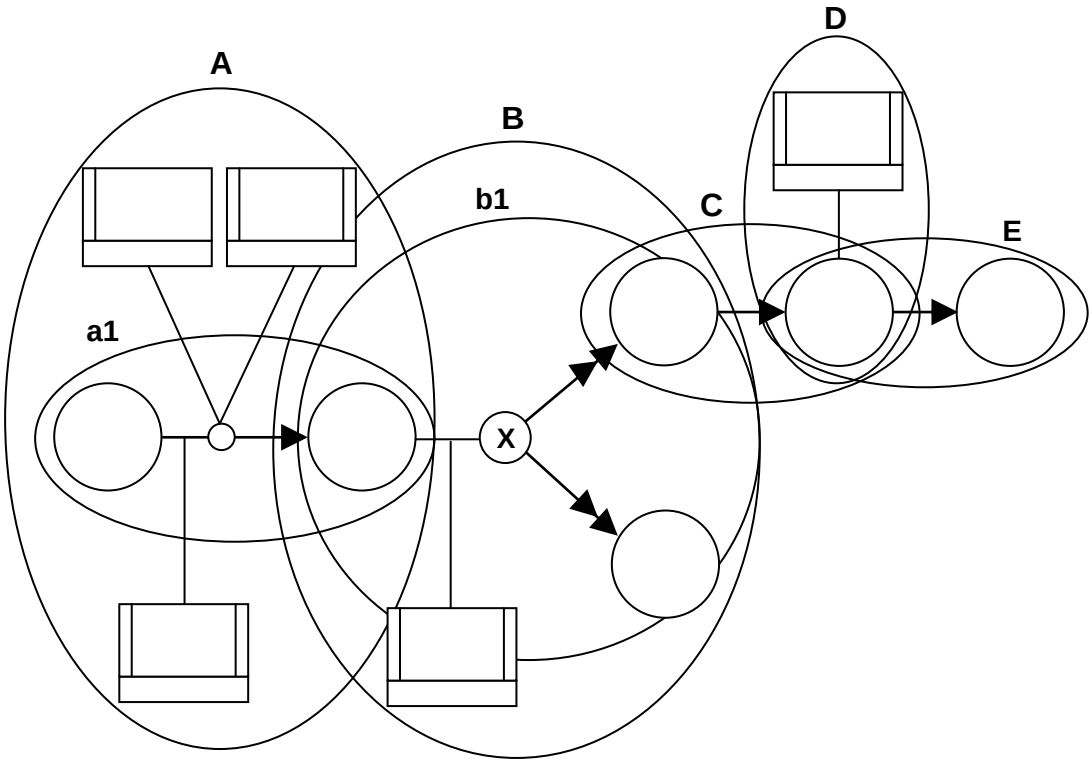


図 6-5　図 6-4 のスキーム図の分割例

まず、スキーム図を A、B、C、D、E の 5 つの大きなグループへと分割する。この時、これらのグループ分けの殆どが、4 本の変移リンクを中心に行われている（その内の 1 つは、分離する交差点を介して 2 本のパスを持っている）。グループ D の中心は、変移リンクではなく、単品のオブジェクト状態であり、状態とそれに取り付けられた参照を 1 つのロジック単位として扱っている。グループ A と B では、この例ではサブグループ a1 と b1 へと更に分割を行っている。

分割されたグループに 1 から 7 までの番号をふったとする。読者は左端にあるグループ 1 から開始して、通常は左から右に向かって以下に挙げるような活動をグループ毎に繰り返して行く：

（1）状態条件と退出条件とに特に注意を払いつつ、オブジェクト状態と
その詳細説明を読む。

（2）変移条件と退出条件とに特に注意を払いつつ、変移リンクとその詳細説明を
詳しく調査する。

（3）全ての、背景設定オブジェクト、非変移関係、そして選択されたグループの
範囲内に存在する注記を、考慮する。

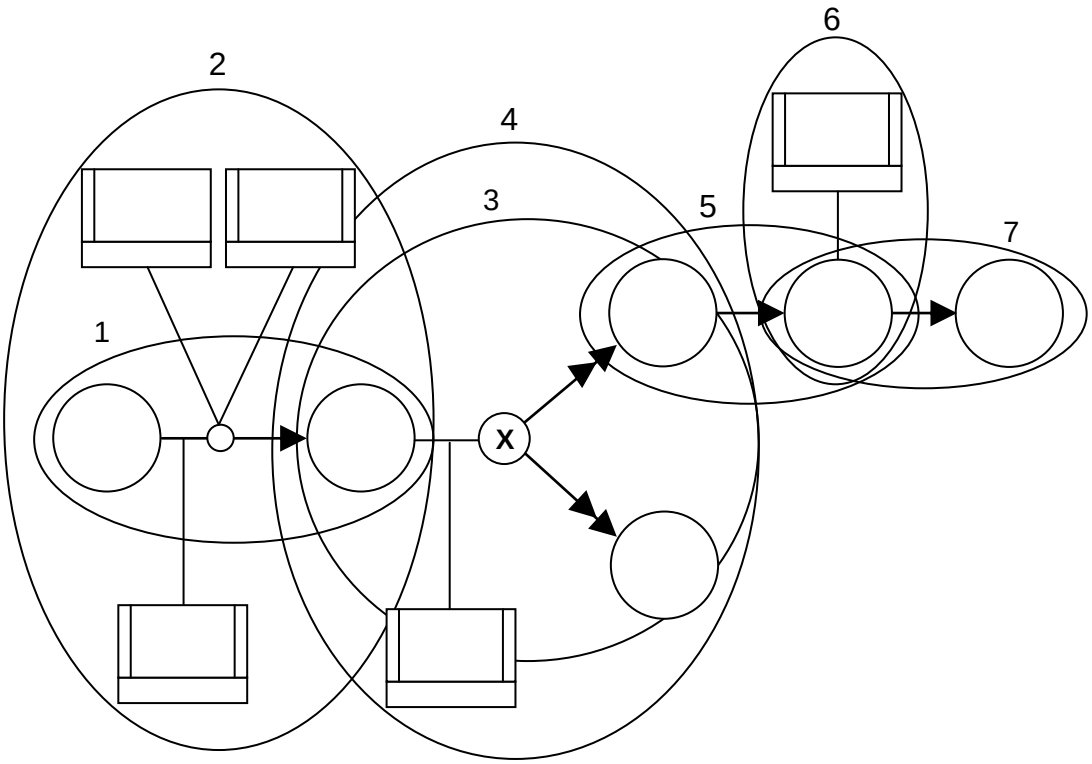


図 6-6　グループの分析

スキーム図を読む際に実施される作業は、プロセススキーム図を読む際に用いられたものと殆ど同じである。異なるのは、プロセススキーム図のプロセス分割では、グループとサブグループとが厳密な階層構造を構成しているが、オブジェクトスキーム図で分割されたグループは、他のグループと重なり合った形で階層構造を表現する傾向がある。

プロセススキーム図のグループを読解する為の背景（context）は、グループの階層内での位置によって確立されていたが、オブジェクトスキーム図のグループを理解するための背景は、グループ内の最も左端にあるオブジェクトの状態によって確立されている。
つまり、スキーム図を左から右へと処理してゆく中で、読者はオーバーラップしているオブジェクト状態の研究を繰り返すことになる。オーバーラップしたオブジェクト状態を最初に読む際は、主にそのオブジェクト状態へと至る変移リンクと関係する、変移状態と進入状態に焦点を当てる。2 回目にそのオブジェクト状態を読む際には、主に状態条件と進入条件に焦点を当てる。

このように、最初の読解では、グループの右端にある要素のオブジェクト状態を考慮し、2 回目の読解では、その隣のグループの左端にあるオブジェクト状態を考慮するという、2 通りの読解を実施する事により、批評者は、あるオブジェクトの状態変移の挙動について、明確に理解するのである。プロセススキーム図と同様に、スキーム図全体を別途批評し直すことにより、記述の理解がより深いものになる。

参考文献

（略）

付録 A：IDEF3 詳細説明言語

訳注：付録については、全般的に内容をよく理解できないまま翻訳を行っている

この付録では、バッカス・ナウア記法（Backus-Naur form、BNF）の仕様も含めた、IDEF3 詳細説明言語を詳細に説明している。

訳注：バッカス・ナウア記法（Wiki より）
文脈自由文法を定義する為に用いられているメタ言語であり、現在はこの BNF を拡張した EBNF が、ANS.1 や XML の構文定義で使用されている。

IDEF3 詳細説明言語は、大きく 2 つに分けて説明して行く。まずは、どのような言語であっても必要となる基本的な文法機構（syntactic apparatus）と、関連する論理的公理（logical axioms）を定義する言語の基本部分を説明する。そしてそれを、追加の文法機構や、状況や UOB、その他の IDEF3 の実体を表現する為の特殊用語の定義、更には追加物の基本的な使用方法を制約する機構によって拡張したものについて説明する。

言語の基本部分は、知識交換フォーマット（Knowledge Interchange Format, KIF）（Genesereth & Fikes、1992）を基にしたもので、理論的な設定に関する物を主に、幾つかの構成物を省略している。A.1 章では、言語の基本部分の個々の文法について、それぞれの文法の要素をどのように使用するかについて、例を挙げつつ説明している。正式な BNF については A.2 章で、幾つかの定義と付随する機構を用いて明確に説明している。状況理論（situation theory）の全体像については A.4 章で説明されており、導入された IDEF3 特有の拡張を動機付けている。こうした拡張の基本的な挙動については、A.5 章で解説を付けつつ公理化（axiomatize）している。

A.1　詳細説明の基本部分の記述（Description of the Elaboration Language Core）

A.1.1　基本的な構文種類（syntactic types）

詳細説明言語の基本部分（core）の基本的な構文種類は、極めて標準的である。

（訳注：言語論は良く理解できていないので、適切な日本語の訳語を当て嵌める事が不可能であり、なるべく原語を併記する）

訳語	原語	説明
文字	letter	文字はアルファベットで、大文字と小文字とがある。
数字	digits	0 から 9 までの数字。正の数字（positive digit）は 1 から 9 まで。
識別子	identifier	（訳注：プログラム言語における変数や配列、手続の名前として用いる文字列） 文字、数字、ダッシュ（dash、ASCII 記号ではマイナスと同じ記号。-）、アンダースコア（_）が使用できるが、文字で始まり、数字もしくは文字で終わらなければならない。
句読点	punctuation	句読点は、文字と数字以外の ASCII キャラクターで構成されている。（訳注：カッコやコーテーション等も含む）
正負記号	polarity	正負記号（+-）は、infon（状況理論の情報の部品）を表現するという特別な用途に用いられている。
？	posint	1 個以上の数字の並び。例えば正の整数。0 以外の正の数字から始まっている。
？	unsigned int	0 もしくは posint。（0 以上の整数）
整数？	int	unsigned int もしくは、posint の頭にマイナス（-）を付けた物。
ベキ指数	exponents	ベキ指数は、E もしくは e の後に int が続く
浮動小数点？	float	ベキ指数が後ろに付いた int か、もしくは、正負記号（正記号は無くとも良い）の後ろに int、その後に小数点、その後に数字、その後にベキ指数が続いたもの
数	number	整数もしくは浮動小数点（floating point）
文字列	stirng	文字列は、空白も含めた ASCII 記号の並びであり、前後をダブルコーテーション（"）で挟む

変数	variable	変数は、識別子の頭に疑問符（”?”）を付けたものである。IDEF3 詳細説明言語の基本部分の変数には、完全な形式が存在しておらず、殆どの論理言語では区別されている関係変数（relation variable）と個体変数（individual variable）の間の区別が無い。むしろ種類化（typing）は、IDEF3 の特殊公理の手法と、意図した意味論言語の基本的意味論カタログを導入し、その特性とそれらの間の関係性とを特徴づける定義とによって、強化されている。A.3 章を参照の事。

A.1.2　演算子（operator）

演算子は、特に項（term）や文（sentence）、定義といった、より複雑な表現に使用される表現方法である。IDEF3 詳細説明言語で使用されている演算子には、以下の物がある：

訳語	原語	説明
定義 演算子	Definition Operator	識別子の定義を形成する為に使用される演算子であり、次に挙げる言語の基本的な意味論カテゴリーの 1 つを用いてオブジェクトを表現している： define-indivisual, define-function, define-relation
項 演算子	Term Operator	複雑な表現に使用されている。次の 7 種類の演算子がある： listof, the, number-of, if, cond, lambda, kappa
文 演算子	Sentence Operator	複雑な文を形成する為に使用されている演算子である。 IDEF3 詳細説明言語には、共通な Boolean 演算子と、それに関係する演算子が含まれている： not, and, or, xor, =>（包含）, <=>（相互包含） また、標準的な全称量化記号（universal quantifier）である forall、存在量化記号（existential quantifier）である exists も含まれる。 数値化された存在量化記号には以下の物がある： exists-1：少なくとも 1 個存在。exists と同じ exists!-1：厳密に 1 個存在する exists-2：少なくとも 2 個存在する exists!-2：厳密に 2 個存在する

A.1.3　項（Terms）

IDEF3 詳細説明言語では、項（terms）、文（sentences）そして定義（definitions）の 3 種類の表現を使う事ができる。この内の項（terms）は、何らかの種類のオブジェクト（a object of some sort）を意味している。この時、それが何の種類なのかは言語によって明確に定義されておらず、むしろ特別な公理（axiom）（後で説明されている）という手段によって別々に定義されている。

項は以下のように区分されている：

（１）原子項（Atomic Terms）

この分類には、識別子（identifier）、変数（variable）、整数（int）、浮動小数点（float）、文字列（string）といった、言語の中心となる単純な「種類（type）」が全て含まれている。

（２）複雑項（Complex Terms）

複雑項は、究極的にはより単純な項と（恐らくは）文や演算子によって構成された項である。最も単純な種類の複雑項は、識別子のリスト（list）だけで構成されたものである。言語の文法は、そうした複雑項が何を意味しているのかに関係無く、その複雑項を構成する事を容認している。その為、「(Plato Socrates)」といった項を作ることができる。

しかし、このような n+1 個の項が含まれた複雑項は、最初の項が n 位関数（n-place function）である時だけ、意味論的に（semantically）意味を持つ（meaningful）のである。つまり詳細言語での意味論において「(Plato Socrates)」は（おそらくは「Plato」と「Socrates」という項は、哲学者の Plato と Socrates のそれぞれを意味していると思われるが）、それらは無意味であると考慮されるのである。より正確に言うならば、この項は null として知られている特別な「空白」を表わしているのである。

（訳注：上の例では、どちらも人名であり、最初の項が n 位関数、この例では 2 項なので 1 位関数、つまりは「○○を△△する」というような動詞である場合にのみ、意味を持つということか？）

項演算子は、複雑項の別のクラスの形成にも使用されている。listof は、その引数

（argument）から、リスト（list）もしくは並び（sequence）を形成する。その為、「(listof 2 3)」は、数字の 2 と 3 を、その順序で構成した 2 要素のリストを表現している。意味論的には、リストは有限の並びとして扱われており、例えば正の整数の最初の数字を返す関数（？ functions on proper initial segments of the positive integers）がある。

if と cond は条件項（conditional term）を形成する。この条件項は、その項の外延（denotation、語が指示する対象の可能域）が、保持されている幾つかの相互排他条件（mutually-exclusive condition）に依存している（？？）。

例えば下の項は、歳が 18 以下であれば Galen であり、そうでなければ Galen の姉である、ということを意味している。

(if (< (age-of ?x) 18) Galen (sister-of Galen))

cond も if と同様であるが、複数の条件を持つ仕様のみの使用が許可されている。Larry が誰か良くわからない政治家について話をしているとする。彼が話をしている政治家は、厳密に 1 人しかいないとするならば、その政治家と一致する項は、次のように完全に定義する事が可能である：

(define-indivisual POL := (the ?x (and (politician ?x) (talking-about Larry ?x))))

もしも Larry が話題にしているオブジェクトが、（1995 年 8 月の）大統領（President）か、副大統領（Vice-President）、もしくは下院議長（Speaker of the House）のいずれかであるということを追加で知ることになれば、次のような複雑項により、より明確にオブジェクトを表現する事ができる：（訳注：当時の米大統領はビル・クリントン）

(cond (predident POL) Bill)
(vice-pres POL) AL)
(speaker POL) Newt),

（訳注：上の式は文法的に間違っているようなのだが、どう直したら良いのかわからないので、そのままにしている）

この cond を含んだ複雑項は、Larry が大統領について話をしているのであれば、それはビル・クリントン（Bill Clinton）であり、副大統領について話をしているのであれば、それはアル・ゴア（Al Gore）であり、また下院議長について話をしているのであれば、それはニュート・ギングリッチ（Newt Gingrich）である、ということを表わしている。

こうした複雑項は、ある状況にあるオブジェクトが、適当な状態によって分類可能な幾

つかのオブジェクトの中の 1 つであるということのみ知っているような状況において、特に便利である。それが何に当るかを明確に知らない場合でも、条件に従って正しいオブジェクトを選択してくれる項を、作成する事が可能である。

残りの演算子は、変数連結（variable binding、変数結合）演算子である。この演算子は、ある 1 個の変数か、もしくはまとめてその外延（denotation）を明確にする複数の変数と、それに関係する何らかの状態とを使用している。

the は、対象となる記述を満たす特有なオブジェクトを表現する（denote）「限定記述（definite description）」項を形成している。例えば、(the ?x (current-U.S.-president ?x)) は、（1995 年であれば）ビル・クリントンを表現している。ただしこの時の限定記述項「(the ?x j)」（j は何らかの「記述」（文など。詳細は後ほど説明））で表現されている必要十分条件は、j によって記述された 1 つの物だけであるということに注意しなければならない。その為、「(the ?x (politician ?x))」という項の場合、この条件では多くの政治家が存在してしまうことから、表現できないか、もしくは空オブジェクト null を表現することになる。

同様に number-of は、ある記述を満足する物の数を表現するのに使われている。

```
(number-of ?x (and (integer ?x) (> ?x 0) (< ?x 5)))
```

例えば上の項は、整数で、1 以上で、5 未満という 3 つの条件を満たす物の個数を表現しており、つまりは「4」となる。

lamda と kappa は、意味論的には、関数と関係を表現する項を形成する為に使用されている。例えば下の文の中の「*（アスタリスク）」文字は掛け算（multiplation、乗法）を意味しているので、

```
(lamda (?x ?y) (+ ?x (* ?y ?y)))
```

は、数値 m と n があった場合、n の 2 乗と m を足す、2 位関数を表現することになる。

同様に kappa 演算子を用いる事で、複雑な特性（property）と関係を表わす項を形成す

ることが出来る。例えば：

(kappa (?x ?y) (and (person ?x) (house ?y) (owns ?x ?y)))

は、家の持ち主と、家との所有関係を表現していることになる。kappa を用いて項を形成する際に注意すべきことは、kappa は、特性もしくは種類（kinds）を表現する変数を 1 個しか連結できないことである。

また、オブジェクト状態（object state）は、種類のサブクラスと見られている事から、種類を特性として考える事も可能である。その為、種類と状態を表現する kind-of を慣習的に使用することが出来る。例えば：

(kind-of ?x (and (person ?x) (> (age-of ?x) 40) (< (age-of ?x) 60))

は、41 歳以上 60 歳未満の人、という種類を表現している。また：

(kind-of ?x (and (person ?x) (sleeping ?x))

は、状態変移スキーム図上の「person:sleeping」で指示されている状態を、詳細説明言語で表現したものになる。

一方で lamda や kappa のような演算子を含む、形式化されていない言語は、パラドックス（paradox、矛盾、逆説）に陥る危険性を持っている。例えば言語の文法は、下の項のような書式を制限している。

(kappa ?r (forall ?x (<=> (?r ?x) (?x ?x))))

これは、自分自身に当てはまる特性は、厳密にその特性自身に当てはまる（the property true of exactly those properties that are true of themselves）、という有名な Russell の非自己例証（？non self exemplification）の特性を表わしている。

このような特性は、矛盾の発生し得るロジックであるが、しかし特に下のような、素朴な包摂原理（naive comprehension principle）を含むロジックという文脈においてのみ、問題を生じる。

(<=> j[var/term] ((kappa var j) term))

この中の j[var/term]は、項 term の文 j 内の変数 var が、全く自由に発生するということを置き換えたものである（注 15)。このようなパラドックスが回避された、適切なロジックを適用するようにしなければならない。（注 16）

本文注 15：

文 e もしくは項 e 上の変数 var は、下のような記法表現が e 内に発生していない場合のみに、自由に発生する。

(OP var y)
(OP (var_1 ...var ... var_n) y)
(OP (var_1 ... var ... var_n : q) y)

この記法の OP は、最初の記法では、「the」もしくは、「exists-1」や「exists!-1」、「exists-2」、「exists!-2」といった数値量化記号（numerical quantifier）の内の 1 つ、3 つ共の記法では、量化記号「forall」や「exist」が当てはまる。

本文注 16：

より正確に言うならば、我々が知る限りで回避されなければならない。論理学者の Kurt Godel の有名な業績によって初めて明らかにされた理由から、（質問を乞うことなく）いかなる合理的で強力な原理であっても、矛盾から逃れることは出来ないのである。これについては、[Godel 31]もしくは、英語の資料なら[Enderton 72]の第 3 章を参照のこと。

訳注：

前ページの、パラドックスの危険性辺りから訳が分からない。ラッセルのパラドックス（Russell's paradox）という有名な話があり、それの絡みだと思われるが、このラッセルのパラドックスそのものも Wiki を読んでも理解できない。

ツールとして詳細説明言語を使用する立場であれば、「lamda や kappa のような演算子を含む、形式化されていない言語は、矛盾した記述にならないように注意しろ」くらいの意

味合いで理解しておけば良いのではないかと思われる。

A.1.4　文（sentences）

（訳注：sentence を「命題」と訳しているところもあるが、「文」にしておく）

直観的に言うならば、文とは真（true）か偽（false）と成り得る表現である。詳細説明言語では、文は基本的に、原子文（atomic sentence）、ブール文（Boolean sentence）、そして量化文（quantified sentence）の 3 つの種類に分けられている。

原子文とは、2 個以上の項で構成された単純な文である。これには、(= Mark_Twain Samuel_Clemens)といった等式（equality sentence）や、(/= Shakespeare Roger_Bacon)といった不等式（inequality sentence）のような、特別な場合も含まれる。

関数と同じく、丸括弧で括られた項の連続物も、適正な原子文となる。しかし、真（true）となる機会をも持たせる為には、そうした文は、n 個の項が後に続く n 位関係という構成になっていなければならない。

全ての非原子文（nonatomic sentence）は、原子文と論理演算子によって再帰的に構築されている。こうした複雑な文には、ブール文と量化文の 2 つの種類がある。標準的なブール文（Boolean sentence）は、ブール演算子と、他の真偽を返す関数演算子によって構築されている。また量化文（quantified sentence）は、量化記号（quantifier）によって構築されている。

ブール文の例は、これまでに既に出てきているが、

(and (person ?x) (> (age-of ?x) 40))

等である。しかしこれまでに出てきた例は、全て自由変数が関係したものであった。この自由変数を実際の名前によって置き換えてみると、(and (person Bill) (> (age-of Bill) 40))となり、?x だった場合の「40 歳以上の人」という意味とはではなく、「Bill は 40 歳を越した人である」という意味の閉文（？closed sentence）に変化してしまう。

訳注：closed sentence（閉文？）と open sentence（開文？）
closed sentence はどのような変数も自由に発生しない形式。その反対の open sentence は、自由変数を持った命題関数（propositional function）。とあるが意味はわからない。

量化文には、基本的に全称量化文（universially quantified sentence）と存在量化文（existentially quantified sentence）の 2 つの種類があるが、それらは更に幾つかの形式に分化している。

最も一般的な形式は、変数 1 個と文 1 個を伴う量化記号で構成されている。下の例は、21 歳以上の全ての人間は大人である、ということを表わしている。

```
(forall ?x (=> (and (human ?x) (>= (age-of ?x) 21))
               (adult ?x)))
```

また下の例は、100 歳以上の人間も存在する、ということを表わしている。

```
(exists ?x (and (person ?x) (> (age-of ?x) 100))
```

量化文の 2 つ目の形式として、幾つかの変数を、1 個の量化記号で合体させるものがある。以下の例は、全ての B-52 は、全ての F-16 よりも重い、ということを表わしている（訳注：F-16 の後ろは「?y」じゃないのかと思うのだが、もう 1 つの例文でも同じなので、そのままにしておく）。

```
(forall (?x ?y) (=> (and (B-52 ?x) (F-16 ?x))
                    (weights-more-than ?x ?y)))
```

詳細説明言語では、変数の範囲を直接に指定する条件を提示することにより、量化宣言（quantified statement）を表現するオプションも用意している。これを用いる事で、上の例文で表現された命題（proposition）を、次のように、もう少し簡潔に表現することも可能である。（訳注：この量化宣言が何なのか意味が分からない）

```
(forall (?x ?y : (=> (and (B-52 ?x) (F-16 ?x))) (weights-more-than ?x ?y))
```

簡素な存在量化記号は、ある明確な記述を満足する物が少なくとも 1 個存在するという事を単純に宣言しているだけであるが、数値存在量化記号は明確な数値 n を用いて、そうした物が最低でも n 個以上存在する事を宣言している。

その為、アメリカ合衆国に州が 50 あるということは、簡単に「(exists-50 ?x (state-in ?x US))」と表現する事が可能である。同様に、厳密に 1 人の大統領が存在するという事実は、「(exista!-1 ?x (US-president ?x))」と表わすことができる。

厳密には、数値量化記号は必ずしも必要ではないと言える。つまり、数値量化記号による表現は、原理的に通常の量化記号と識別関係によって表現する事が可能である。その為、「(exista!-1 ?x (US-president ?x))」は、次のような文で表すこともできる。

(exists ?x (and (US-president ?x) (forall (?y: (US-president ?y)) (= ?x ?y))))

上の例文は、少なくとも 1 人のアメリカ合衆国大統領が存在し、更に全ての大統領の数が 1 でなければならない、と表現されている。しかしこの表現方法では、ある記述を満足させる物の数が n 個ある場合、n 個の別々の変数が必要となってしまう為、その代わりに 1 個のみで表現できる数値量化記号は、非常に便利である（訳注：かなり意訳）。

A.1.5　定義（Definitions）

定義には、完全定義（complete definition）と部分定義（partial definition）の 2 種類がある。

完全定義を行う事で、定義対象によって、それぞれ次のような物が提供される：

- 個体（indivisdual）：定義された個体を選び出す項
- 関数（function）：定義された関数を選び出す項
- 関係（relation）：定義された関係が維持される為に必要十分な条件を合わせて明示する文のセット

部分定義は、個体、関数、もしくは関係に制約が加えるが、それによって定義された物は、一般的に完全定義から何かが欠けた状態にある。また、部分定義は、ある項を追加情報無しで言語に導入する為に使われているだけの事が多い（？）。

完全定義と部分定義では、演算子「:=」と「:=>」をそれぞれ用いている。

個体の完全定義では、個体の名前と個体を定義する項とが明確にされていなければならない。一方の部分定義では、明確にされるべきものは個体の名前だけである。オブションとして、個体の定義を厳密にする為に、部分定義に文を含める事も可能である。

定数（constant）は、完全定義を 1 つだけで定義されることはあっても、複数の部分定義によって定義されることは無い（？）。

例えば、関数「age-of」と、個体「Larry」は、次のように定義される。

```
(define-function age-of
        (forall ?x (and (integer (age-of ?x)) (>= 0 (age-of ?x)))))

(define-individual Larry
        (= (age-of Larry) 49))
```

「age-of」の部分定義（訳注：「(define-function age-of」の部分のこと？）の後ろにオプションの文を付ける事により、関数の返値が 0 以上の整数でなければならない事が明確にされている。そして、こうして定義された関数「age-of」は、「Larry」という項の部分定義の中で、彼の年齢を明確にする為に使用されている。

注意すべきことは、定義は関数と関係を表わす項を、lamda 項と kappa 項によって表現されている関数と関係と同じように、明確に導入する為に使用する事が可能であるということである。その為、例えば上で kappa 項によって参照されている、41 歳以上 60 歳未満の人の特性は、定義という手段により、適切な原子項（訳注：下の「middle-aged」？）を用いて明確に命名することが可能である。

```
(define-relation middle-aged :=
        (and (person ?x) (> (age-of ?x) 40) (< (age-of ?x) 60)))
```

A.2　詳細説明言語の基本部分で用いられている BNF（バッカス・ナウア記法）

この章では、詳細説明言語の基本部分（core）の基本文法である、拡張 BNF（Backus-Naur form、バッカス・ナウア記法）について説明する。これには次に挙げるような慣習が使用されている：

・垂直線「|」は、排他的論理和（exclusive disjunction、XOR）を示している。つまり、「C1|C2」は、C1 もしくは C2 が発生するということを示しており、C1 と C2 が同時には発生しない。この垂直線が無ければ、論理積（conjunction、AND）を示すことになる。

・構成（construct）の後ろに上付きになったアスタリスク「*」は（例えば「C^*」）、その構成 C のインスタンスが 0 以上存在し得る、ということを示している。

・構成の後ろに上付きになったプラスマーク「+」は（例えば「C^+」）、その構成のインスタンスが 1 以上存在し得る、ということを示している。

・角括弧（bracket）によって囲まれた、単一の構成もしくは組み合わされた構成は（例えば「[C1 | C2] 」等）、その構成もしくは構成の組み合わせが、オプションであると言う事を示している（訳注：省略しても構わない、ということ）。

これから説明して行く文法の中で、文法の終端記号（terminal、別の物に変換されない文字）、は、ボールド太字体（bold face）で書かれている。また、文法の非終端記号（nonterminal、他の文字に置き換えられる文字）は、「<」で始まり、「>」で終わっている。

例えば、変数用の識別子は「?」マークから開始されなければならないが、この時、構成（construct）は次のようになる。

<var> ::= **?**<id>

訳注：

この例では、「?」が終端記号で、「<var>」と「<id>」が非終端記号である。「<var>」と「<id>」は、具体的な変数や ID 番号に置き換えることができる。

また、この文章では、ボールド太字体の代わりに Arial 太字体を用いている。

A.2.1　アルファベット

詳細言語の基本部分に用いられるアルファベットは、93 個の標準的な印刷表現の ASCII 記号である。スペースについては、<space>として表現される。

ABCDEFCHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789()[]{}<>
=+-*/¥^~'`"_@#$%:;,.!?

訳注：日本語の変換系列では、バックスラッシュ「＼（これは全角）」が「¥」になってしまう。文法の記述でバックスラッシュは、その直後の記号を、記号そのものとして扱うという指示を行う。例えば「" (ダブルコーテーション)」は文字列をくくる記号であるが、「¥"」と書くことで、「"」という記号そのものを示すことになる。

A.2.2　基本的な構文種類（Basic Syntactic Types）

<letter> ::= **A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|**
a|b|c|d|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
<posdigit> ::=**1|2|3|4|5|6|7|8|9**
<digit> ::=**0** | <posdigit>
<id> ::= <letter> [[<letter> | <digit> | _ | -]*<letter> | [<letter> | <digit> | _ | -]*<digit>]]
<punct> ::= _|-|~|!|@|#|$|%|^|&|*| **(|)|+|=|`|:|;|'|<|>|,|.|?|/||| [|]|{|}**
<polarity> ::= **+** | **-**
<posint> ::= <posdigit><digit>+
<unsignedint> ::= **0** | <posint>
<int> ::= <unsignedint> | **-**<posint>
<exp> ::= **E**<int> | **e**<int>
<float> ::= <int><exp> | <int>.<digit>+[<exp>] | [**+|-**].<digit><digit>*[<exp>]

<number> ::= <int> | <float>
<string> ::= "[<letter> | <digit> | <punct> | <space> | ¥" | ¥¥]*"
<var> ::= **?**<id>

A.2.3 演算子（Operators）

<operator> ::= <defop> | <termop> | <sentop>
<defop> ::= **define-individual** | **define-function** | **define-relation** | **:=** | **:=>**
<termop> ::= **number-of** | **if** | **cond** | **the**
<boolop> ::= **not** | **and** | **or** | **xor** | **=>** | **<=>**
<quant> ::= **forall** | **exists** | **exists[!] -** <posint>
<sentop> ::= <boolop> | <quant>

A.2.4 項（Terms）

<term> ::= <atomterm> | <compterm>
<atomterm> ::= <id> | <var> | <int> | <float> | <string>
<compterm> ::= **(**<term> <term>$^{+}$**)** |
(if <sentence> <term> **[**<term>**])** |
(cond (<sentence> <term>**) (**<sentence> <term>**)**$^{+}$**)** |
(the <var> <sentence>**)** |
(number-of <var> <sentence>**)** |
(lambda <var> <term>**)** | **(lambda (**<var>$^{+}$**)**<term>**)** |
(kappa <var> <sentence>**)** | **(kappa (**<var>$^{+}$**)** <sentence>**)**

A.2.5　文（Sentences）

<sentence> ::= <atomsent> | <boolsent> | <quantsent>
<atomsent> ::= **(=** <term> <term>**)** | **(/=** <term> <term>**)** | **(**<term> <term>$^{+}$**)**
<boolsent> ::= **(not** <sentence>**)** | **(and** <sentence> <sentence>$^{+}$**)** |
(or <sentence> <sentence>**+)** | **(xor** <sentence> <sentence>$^{+}$**)** |
(=> <sentence> <sentence>**)** | **(<=>** <sentence> <sentence>**)**
<quantsent> ::= **(**<quant> <var> <sentence>**)** |
(forall (<var>$^{+}$ **[:** <sentence>$^{+}$**])** <sentence>**)** |
(exists (<var>$^{+}$ **[:** <sentence>$^{+}$**])** <sentence>**)** |
(exists[!]-<posint> **(**<var> **:** <sentence>**)** <sentence>**)**

A.2.6　定義（Definitions）

<definition> ::= <partial-def> | <complete-def>
<complete-def> ::= **(define-individual** <id> := <term>) |
(define-function <id> (<var>$^{+}$) := <term>) |
(define-relation <id> (var>$^{+}$) := <sentence>)
<partial-def> ::= **(define-individual** <id> **:=** <term>**)** |
(define-function <id> <setence>***)** |
(define-relation <id> **(**<var>***) :=>** <sentence>)

A.3　基本的な定義、公理（axiom）、公理型（axiom schema）

詳細説明言語の構文は、形式化が殆ど無されていない（highly untyped）。例えば文と関数項は、形式の強制が無い、ただの項のリストである。その代り、意味論としての形式化が強制されている。つまり公理（axioms）を用いる事で、構成している項（constituent terms）の外延（denotation）が正しい並び（right sort）で有りさえすれば、原子文が真になるように、また関数項がルール上正しい（legitimate）オブジェクトを表現するように、保証しているのである。この公理を用いた保証は、適切な意味論分類（semantic category）（例えば種類など）と、「define-relation」や「define-fucntion」といった宣言を並べた補助的な観念（notion）を最初に導入し、そして次に、こうした観念の公理化（axiomatizing）により、関係する種類分けの制約を獲得することで、達成されている。

注意すべきことは、こうした最初に行われている意味論分類は、IDEF3 に関係無く、完全に一般的な事だということである。こうした一般的な分類が行われ、公理化された後になって初めて、IDEF3 での意味論分類と定義とが明確に実行され、公理化されるのである（A.5 を参照）。

ここでの目的には明らかに不要であることから、基本部分となる詳細説明言語がモデル化される前の、KIF（知識交換フォーマット）の基本的な意味論分類の幾つか、とりわけセットとリストについての説明が、省略されているということにも、注目しておくべきである。

厳密に言うならば、集合論（set theory）は定義を正当化する為に必要であるが、それは定義したオブジェクトの存在を証明可能にしておかなければ（もしくは最低でもしようとしなければ）ならないからである（ただし、与えられた公理から、その存在を証明できないようなオブジェクトを、単に仮定したり導入したいだけである場合を除く）。

ここで集合論についての説明が省略されている理由は、特にユーザーが、強力な von Neumann-Godel-Bernays の KIF 集合論よりも弱い集合論を使用したいと望んでいる場合に、ユーザーが自分に合った集合論を選択できるオブションを用意すべきだ、という考え方が反映されている為である。関係する集合論の必要性を認知してはいるものの、この論文では、どの集合論を用いるかは読者に暗黙的に委託されている。

A.3.1　基本的な意味論分類（Basic Semantic Categories）

ここでは、基本的な意味論分類である individual（個体）、function（関数）、そして relation（関係）について紹介するが、それと同時に、正しい外延（genuine denotation）を持たないということを示すだけに使用されている特殊な null（空）オブジェクトについても説明する。

```
(define-relation individual (?x))
(define-relation list (?x))
(define-relation relation (?x))
(define-individual null)
(define-relation function (?x)
        :=> (relation ?x))
(define-relation interval (?x)
        :=> (individual ?x))
(define-relation integer (?x)
        :=> (individual ?x))
```

関数定義に関係する殆どの書式（formalization）は、ある明確な並び（？sort）という関係として関数を定義している。例えば、KIF（知識交換フォーマット）で行われているような集合論的取扱いでは、二位関係は順序を持つ対（ペア）で構成されたセットであり、その為、関数はただの二位関係となり、関係の対の 1 つ目の要素は、（対になっているオブジェクト以外の？）他のオブジェクトと対になることはないのである。

公式にも（formally）、この識別は非常に都合が良い。その為、この識別は、関数定義を行う公理として追加されている。同様に、期間（interval）と整数（integer）の両方が、個体（individual）の種類として取り扱われている。

関数としても、関係としてもとられてしまうと言う関数の二重性は、次のような一連の公理によって獲得されている。

「1 つ目のオブジェクトを適用した f の値が 2 つ目のオブジェクトになっている場合にのみ、2 つのオブジェクトが（二元）関数関係（(binary) functional relation）f にある」

(forall (?x ?y ?f: (function ?f)) (<=> (?f ?x ?y) (= (?f ?x) ?y))))

「2 つ目までのオブジェクトを適用した f の値が 3 つ目のオブジェクトになっている場合にのみ、3 つのオブジェクトが（三元）関数関係 f にある」

(forall (?x ?y ?z ?f: (function ?f)) (<=> (?f ?x ?y ?z) (= (?f ?x ?y) ?z))))

一般的に、以下のような公理型（axiom shcema）が成り立つことになる：

(forall ($?var_1$ ... $?var_{n+1}$?f: (function ?f))
(<=> (?f $?var_1$ ... $?var_{n+1}$) (= (?f $?var_1$ ... $?var_n$) $?var_{n+1}$))))

A.3.1.2　アリティ（Arity）

関数もしくは関係のアリティとは、それが幾つの引数を持っているかという事を示している。関数はそれがまた関係でもあることから、rel-arity と func-arity というアリティについて、2 つの別々の観念が必要となる。

rel-arity は、1 つの関係を引数として持ち、その値として正の整数を出力する関数である。

```
(define-function rel-arity
  (forall (?x ?y : (/= ?y null))
        (=> (= (arity ?x) ?y)
            (and (relation ?x) (integer ?y) (> ?y 0)))))
```

小文字のギリシャ体の文字「n」（訳注：Word のフォントに無いので、斜体にしておく）は、数 n の数字であるとする（？？）。この時、下に挙げる公理型（axiom schema）は、関係のアリティ rel-arity と、その関係を参照する項に関わっている原子文の真理（truth）との間の繋がり（connection）を表現している。すなわち、本質的に「1 つの原子文は、n 個の項がその後に続くアリティ n の 1 つの関係によって構成されている場合にのみ、真と成り得る」ということになる。

```
(forall   (?r ?var1 ... ?varn)
      (=> (?r ?var1 ... ?varn)
          (and (relation ?r) (= (arity ?r) n))))
```

「(?r ?var$_1$ ... ?var$_n$)」という図式的表現（schematic expression）が、詳細言語として正しい表現ではない事から、図式的文字（schematic letter）n を代用した変数?n の数を用いる量化（quantifying）によって、このような表現が不可能であることに注意しなければならない。（？？？）

rel-arity と同様に、func-arity は 1 つの関数を引数として正の整数を返値とする関数であ

る。

```
(define-function func-arity
   (forall (?x ?y : (/= ?y null))
         (=> (= (arity ?x) ?y)
               (and (function ?x) (integer ?y) (> ?y 0)))))
```

ある関数の func-arity と、その関数を参照する項と関係する関数項の外延（denotation）との間の連結は、本質的に次のような公理型から捉えられている。

「null オブジェクトを表わさない原子関数項は、n 位関数とその後に続く n 個の項を構成している」

(forall (?f $?var_1$ ... $?var_n$: (=/ (?f $?var_1$ ... $?var_n$) null)))
(and (function ?f) (=arity ?f n)))

また、次のような公理によって、アリティ関数に更なる制約が掛けられている。

「全ての関係と関数は、null ではないアリティを持っている」

```
(forall ?x (=> (or (relation ?x) (function ?x))
         (/= (arity ?x) null)))
```

全ての関数は関係でもあることから、rel-arity と func-arity は、共に関数として定義されてる。もちろん、それらの間には密接な関係（close relationship）が存在していることから、もしも f が関数として見られた場合には、func-arity は f に対して 1 個の値を返し、また f が関係として見られた場合には、rel-arity は別の値を返す。これをより厳密に表現すると、次のような 3 つの公理となる。

「ある関数の rel-arity は、その func-arity よりも 1 だけ大きい値になる」

```
(forall (?f : (function ?f)) (= (rel-arity ?f) (+ (func-arity ?f) 1)))
```

「n 位関係によって構成され、かつ n 以外の数 m（m≠n）の個数の引数項を持つ原子文は、必ず偽（false）を返す」

(forall (?var$_1$ ... ?var$_n$?r : (relation ?r) (= (arity ?r) m))
(not (?r ?var$_1$ ... ?var$_n$)))

「n 位関数項と、n 以外の数 m（m≠n）の個数の引数項によって構成された関数項は、必ず null オブジェクトを表現する」

(forall (?f ?var$_1$ ... ?var$_n$: (function ?f) (= (arity ?f) m))
(= (?f ?var$_1$ ... ?var$_n$) null)))

A.3.1.3　Null オブジェクト

以下の 2 つの公理型は、null オブジェクトと、それと関数・関係との連結を特徴付けている。

「null オブジェクトを表現する項を含む原子文は、必ず偽（false）となる」

(forall (?r ?var$_1$ ... ?var$_n$: (relation ?r) (or (= ?var$_1$ null) ... (= ?var$_n$ null)))
(not (?r ?var$_1$ ... ?var$_n$)))

「null オブジェクトを表現する項を含む関数項は、それ自身が null オブジェクトを表わす」

(forall (?f ?var$_1$ ... ?var$_n$: (function ?f) (or (= ?var$_1$ null) ... (= ?var$_n$ null)))
(= (?f ?s$_1$ ... ?s$_n$) null))

A.4　基本的な状況理論（Basic Situation Theory）

プロセススキーム図と状態変移スキーム図の両方に共通な根本的で直観的な意味論（underlying intuitive semantics）は、状況理論（situation theory）の上に成立している。IDEF3 詳細説明言語の目的は、対象となるプロセスや状況変移に追加する制約と情報を、極めて正確に表現する事であるが、この基本部分となる詳細言語は、特別に IDEF3 に合わせて構成された状況理論によって拡張されている。こうした構成の暗黙的な理論は、本格的な状況理論よりも力不足なものに思えてしまうが、しかし直観的な意味論に加えて、経験からも、この理論が経営プロセスの獲得とモデリングに必要不可欠なものであるということが証明されているのである。

ここでは、状況理論の概略を紹介し、それによってプロジェクトリーダーによる詳細説明言語を用いた詳細説明宣言の作成を可能にする。

A.4.1　状況とインフォン（infon、状況理論における形式化された部分情報）

「状況（situation）」の観念（notion）は、状況理論の最も基本的な観念である。この観念は、知識表現の文法（literature of knowledge representation）に近い。

状況は（典型的に）具体的で（concrete）、空間的・時間的な広がりを持った、現実世界の部品（piece）である。例えば野球や、数学の授業、生産システムといったものである（状況は、実数範囲のような非具体的な系（system）でも確認されてはいるが）。

状況理論においては、対象となる状況を他の状況と区分けするものは、その状況にあることを支援するか、その状況にあるように維持している情報の部品（piece of information）である。状況理論では、こうした情報の部品それぞれを「インフォン（infon）」と呼んでいる。対象となる分野のインフォンは、それ自身がその分野に存在しているオブジェクトや特性、そして関係から構成されている。（ここでのオブジェクトとは、物理的オブジェクトだけでなく、数字や期間といった抽象的オブジェクトも、広義的に含んでいると解釈されている）

より明確に説明するならば、対象となる状況 s の基本インフォン（basic infon）は「情報の基本ユニット（fundamental units of information）」であり、良くも悪くも、状況 s の中のそうした関係と、その関係に対応する引数（appropriate argument）の組み合わせで「生

成されている（generated）」。つまり、状況 s の基本インフォンは、以下の書式で表される、文法的に正当なユニット（legitimate unit）として発生し得る全ての物によって構成されているのである：

オブジェクト　$a_1, \ldots a_n$ は関係 r の関係にある

と

オブジェクト $a_1, \ldots a_n$ は関係 r の関係にない

ここで、r と a_i は、s の構成物の全てである。（こうした個別のオブジェクト間で保持されている関係は、一階関係（first-oeder relations）として知られている）。

以上の 2 つのインフォンは、言語で表現すると、それぞれ「(r a_1 ..., a_n +)」と「(r a_1 ..., a_n -)」となる。状況 s は基本「正」インフォン（basic "positive" infon）「(r a_1 ..., a_n +)」を、オブジェクト $a_1, \ldots, a_n$ が状況 s に存在し（少なくとも状況 s である期間のある時点で）、また状況 s で関係 r が成立している時のみ「支援」する。そして状況 s は、基本「負」インフォン（basic "negative" infon）「(r a_1 ..., a_n -)」を、オブジェクト $a_1, \ldots, a_n$ が状況 s に存在するものの、関係 r が成立していない時のみ「支援」する。

その為、例えば、インフォン「(mother-of Hillary Chelsea +)」と「(mother-of Chelsea Hillary -)」は、1995 年の典型的なホワイトハウスの状況 s 下で、支援されるか、もしくは保持されているの。それとは対照的に、そうした状況は「(mother-of Chelsea Hillary +)」と「(mother-of Hillary Chelsea -)」を否定する（deny）する、もしくは、こうしたインフォンはそうした状況では失敗する（fail）、と言う事もできる。

こうした事実は、s が適切なホワイトハウス状況であれば、言語によって

(supports s (mother-of Hillary Chelsea +))
(supports s ((mother-of Chelsea Hillary -))
(denies s ((mother-of Chelsea Hillary +))
(denies s (mother-of Hillary Chelsea -))

とそれぞれ表現されることになる。

注意すべき事は、状況は（一般的に）世界の「限定された」部品であることから、ある状況 s に存在しているオブジェクト b は、別の状況 s¢には存在していない、ということである。つまり、状況 s¢は、b を「言及できない（silent）」となるのである。より正確には、

「状況 s¢は b についての情報を何も支援しない」となる。つまり状況は「情報への顧慮が偏っている（partial with respect to information）」のである。その為、状況は物事の全ての個体や全ての状態（state）についての全ての疑問には答えられないのである。

例えばヒューストンにあるアストロドーム球場での典型的な野球の試合は、シアトルにある Pike Place 市場の銀鮭の価格についての情報を、全く伝えることがないのである。

状況 s が対象となる基本インフォンを支援しているということは「(r a_1 ... a_n +)」、個体 a_1, ... , a_n が、状況 s を通して関係 r にある、ということである。しかし、例えば典型的な「朝」という状況で、人が「睡眠」から「歩行」へと変化するように、状況 s の中では物が変化し得るのである。そこで状況理論では、これに該当する場合には、時間的間隔（期間）を個体と見なし、一階関係の引数の中に時間というパラメータを明確に含める事によって、こうした時間的変化を反映させている。

その為、例えば特性「寝ている（asleep)」は、個体と時間的間隔（期間）の間で保持された二位関係であると考えられている。もしも状況 s が、個体 b の家での午前 6 時から午前 8 時までの間の典型的な「朝」状況であるとすれば、

```
(supports s (asleep b 0600 +))
(supports s (asleep b 0800 -))
```

の両方になる。(もしも関係する時間的パラメータが判明しているならば、もちろん、都合次第で隠ぺいすることは可能である（？))。

IDEF3 で使用されているバージョンの状況理論の意味論では、状況の発生する期間に挟まれた全ての期間もまた、状況に現れていることが、前提条件になっている。その為、午前6時から午前8時までの間に発生している状況は､例えば午前6時から 15 分まで期間は、午前 6 時半から 45 分までの期間の前にある、というような、全ての関係のある時間的情報を支援していることになる。

A.4.2　種類、UOB、プロセス（Types, UOBs, and Processes）

殆どの物理系（sysytem）では、ある面で似通った状況が複数発生していることが観察されている。そうした場合、その似通った状況は、同一の「種類（type)」であると言える。例えば、「Bill Clinton が火曜日にランニングしている」という状況と、「水曜日にランニングしている」という別の状況は、多くの面で異なっているが、「Clinton がランニングしている」という点については似ているのであり、その為、どちらも同じ種類の状況のインスタンスになる。

その為、一般に状況種類（situation type）は、多くの異なった具体的な状況による表現が可能な、「繰返しのできる（repeatable)」パターンになる。しかしこれは、厳密には IDEF3 における UOB の特徴（character）であり、その為 UOB は状況種類によって識別されている。状況理論では、状況種類は類似の状況から抽出された演算子（operator）と、適切な抽象化変数（abstruction variable）※17 によって明確にされている。そしてここでは演算子「type-of」を用いることにする。

これらを用いて、上で説明した「Clinton がランニングしている」という活動を表現してみると、次のようになる。

```
(type-of ?sit (supports ?sit (running Clinton)))
```

同様に、個々のオブジェクトもまた、ある明確な面から見る事で同一物に成り得ることから、同一の「オブジェクト種類（object type)」のインスタンスとして考える事も可能である。その為、Bill Clinton と Jimmy Cater は、どちらも男性の政治家ということにおいて似ていることから、彼らはどちらも「男性政治家」という種類になる。そして「男性政治家」は、Clinton と Carter によって共有された特性としても考える事が可能であり、詳細説明言語によって、次のように表現する事ができる。

```
(type-of ?x (and (politician ?x) (male ?x)))
```

IDEF3 では、状況種類とオブジェクト種類の両方が、特性によって（状況種類は状況の特性、オブジェクト種類は個体の特性によって）単純に識別できるようになっている。

演算子「type-of」は、特性抽象化演算子（property abstruction operator)「kappa」の観念変数（notational variant）として単純化する事も可能である。（A.1.3 を参照）

本文注 17：

正式な状況理論では、抽象化変数は、実世界にある実際の「変数オブジェクト」と意味論的に一致する変数であり、「パラメーター」や「不定元（indeterminates）」として知られているものである。それらを必要とする明確な表現的必要性が存在していたとしても、この論文の中では、こうした実体を無効化することが可能である。（訳注：意味わからん）

ほとんどすべてのプロセス記述の意味論的内容が、「種類（type）」に関係したものであることから、プロセス獲得とプロセスモデリング（更にはモデリング一般）において、種類は非常に重要である。

より正確に言うならば、典型的なプロセスは、プロセスの中の対象となる活動でのプロセスフローを反映するというやり方で構造化された、相互に関係し合った UOB の集合体、つまりは、対象となる活動における UOB 種類のインスタンス間の時間的関係、として考えるのが、一番である。

例えば、図 A-1 のような塗装プロセスについて考えてみる。このダイヤグラムは、（前に何もない、「部品塗装」箱によって表現されている）UOB「部品塗装」のインスタンスから開始され、その後「塗膜検査（Test Coverage）」へと続く、普遍的なプロセスについて説明している。このプロセスフローでは、プロセスが別の「部品塗装」インスタンスへとループバックするか、もしくは部品乾燥へと続行されるかは、検査結果に依存している。その為、原則として、この 1 個のプロセスが、ある明確なイベントの過程（courses of events）によってインスタンス化（be instantiated）される方法が無数に存在しており、それは「部品塗装」活動へとループバックが何回繰り返されるかに依存している。

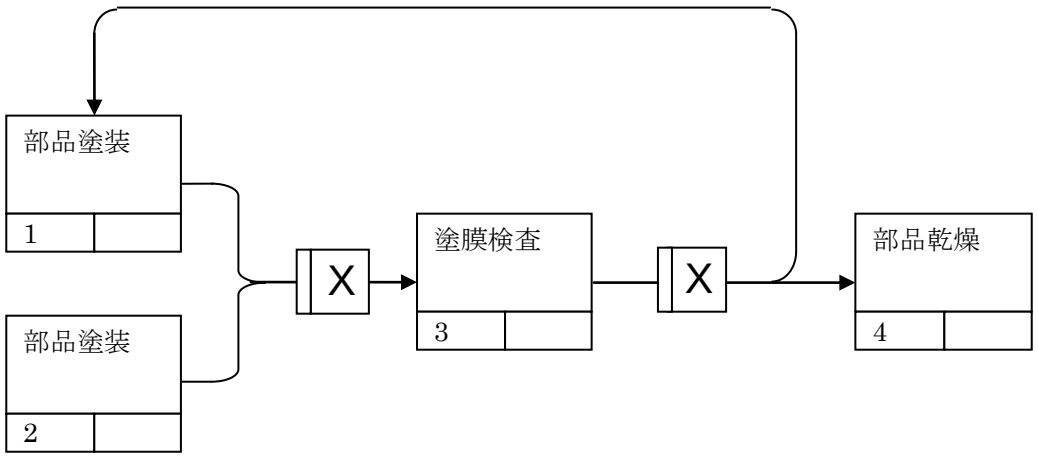


図 A-1　塗装／検査／乾燥　シナリオ

より一般化するならば、UOB のような状況種類は、任意の状況で使用される変数「?sit」と、その種類の状況に共通な状態（conditions）を明確にする式（formula）j によって、明確にされていると言える。特に、活動は「j となるような状況種類」と読む「(type-of ?sit j)」という式によって言及されている。その為、上に挙げた例の「(type-of ?sit (supports ?sit (running Clinton)))」は、「Clinton がランニングしているということを支援する状況種類」もしくは、もっと自然に「Clinton がランニングしているという状況種類」と読むことが出来るのである。

「?sit」が s を参照した時に j が真である場合にのみ、状況 s は「(type-of ?sit j)」という状況種類の状況になる。もし j が式「(supports ?sit i)」であれば（i はインフォン項）活動は「内部的に（internally)」明確化され、そうでなければ活動は「外部的に（externally)」明確化される。前者では、活動の発生が支援するインフォンを用いて、内部的仕様がその活動を記述しているが、後者では、外部的仕様が活動の発生が支援するインフォンの代わりに、活動の特性を参照している。(外部的明確化の例：活動の発生の原因、コスト管理に関連するコスト、等)

A.4.3　基本的な状況理論での関係（Basic Situation Theoretic Relations）

状況理論は、その対象世界が幾つもの異なった「意味論分類（semantic categories)」へと分割されているという意味において、高度に「種類分け（typed)」されている。この意味論分類の種類として最も顕著なものとして、オブジェクトや一階特性、一階関係、インフォン、状況、イベントの過程（courses-of events)、オブジェクト種類、状況種類、プロセス、期間（temporal interval）といったものがある。

こうした区別（distinction）を獲得する為に、状況「理論」は、詳細説明言語が定義する、こうした分類（category）を表現した項の範囲内で、組立てられている。それに加え、オブジェクトの分類の何が特別な関係に立つことが可能であるかということについて正確に表現した公理（axiom）に沿い、そうした特別な関係のクラスを意味するように、項の一揃い（variety）が定義されている。

「彼」や「彼女」といった言葉を排除して（? at his or her disposal)、こうした項を用いる事により、ユーザーは、IDEF3 スキーム図言語の項では表現できなかった、どのよう

な追加情報や制約についても、明確に表現する事が可能となっている。

具体的な関係については、以下のようになっている（訳注：本文は文章の羅列で説明しているが、分かり難いので表にまとめ直した）：

関係	対象	成立条件
supports denies	状況とインフォン	A.4.1 で説明済
occurrence-of	状況 s と UOBU	s が U のインスタンスである場合のみ
activation-of	イベントの過程 c と プロセス P	c が P の発生である場合のみ
occurs-in	状況 s と イベントの過程 c	s が c 上で発生する場合のみ
activity-in	UOBU と プロセス P	P を構成する状況種類の中に U がある場合のみ （occurs-in の種類レベルでの鏡像）
of-type	状況 s と UOBU	s が U のインスタンスである場合のみ
object-in	オブジェクト b と 状況 s か UOBU	b が、s の中、もしくは U のインスタンスの中で発生する場合のみ

時間的関係の一揃い（variety）は、複雑なプロセスの時間的構成（ある時間における構成）を記述する為に必要である。

この中で最も重要な関係は「meets」である。ある期間 i は別の期間 j に「meets」するのは、i の終了点が j の開始点となる場合のみとなる。

それ以外の時間的関係には、「precedes」「starts」「finishes」「overlap」「during」といったものがあるが、「meets」と同じやり方で定義可能である（A.5.4 参照）。注意すべきは、期間は一階オブジェクトとして取り扱われていること、そしてその為に、時間的関係は全て一階関係になるということである。時間的関係は、状況の間で一致している時間的関係の一揃い（variety）を定義する為にも使用されている（？）。

A.5　IDEF3 詳細説明の正式な言語（Formal Language）

これまで詳細説明言語の基本部分（core）について説明してきたが、この章では、適切な公理の定義を通して、状況理論の基本的な意味論分類の導入と定義を行ってゆく。幅広い範囲を対象とする為に、公理をまず普通の文章（訳注：原文では英語の文章）で示し、それから詳細言語で記載するが、詳細言語部分は読みやすくする為にイタリック体（訳注：ここでは Arial）で記述する。

ここに挙げた詳細言語の基本部分の拡張は、本格的な状況理論の書式になっていないということに注意しなければならない。それよりも、この論文の第 3 章「IDEF3　プロセス記述言語」等にある例として書かれた IDEF3 詳細説明を、状況を中心に表現することを優先している（訳注：かなり意訳）。

また、ここでは公的なモデル理論についても説明を省いている。この論文の目的は、あくまでも経営モデリングと知識エンジニアリングにおいて、効率的に IDEF3 を「使用」することであり、非公式な形で詳細言語の意味論の直観的特徴について説明をしているだけである。

A.5.1　詳細説明言語の基本部分の拡張（Extending the Core Elaboraiton Language）

詳細説明言語の基本部分を、完全な IDEF3 詳細説明言語へと拡張する為に、まず必要となる作業の一つは、インフォン項というクラスを新しく定義する事である。

<infornterm> ::=　**(** <term> <term>$^{+}$ **)** |
(**and** <infonterm> <infonterm> $^{+}$ **)** |
(**or** <infornterm> <infonterm> $^{+}$ **)** |
(<quant> <var> <infonterm>)

「**(** <term> <term>$^{+}$ **)** 」という式で表されているインフォン項は、「原子」インフォン項と呼ばれている。

これに加えて、分類「<compterm>」 は、分類「<infonterm>」に含まれるように変更される。より堅固なものにする為に、KIF に含まれている明確なセット理論機構をここに

追加することが可能ではあるが、今回の目的からは外れてしまうので割愛する。KIF のセット理論の詳細については、「Genesereth & Fikes, 1992」を参照のこと。

A.5.2　基本状況理論意味論分類（Basic Situation Theoretic Semantic Categories）

ここでは、状況理論の基本的な意味論分類の導入と、その特徴の説明を行う。

```
(define-relation infon (?x))
(define-relation situation (?x))
(define-relation UOB (?x))
(define-relation COE (?x))
(define-relation process (?x))
(define-relation interval (?x))
(define-relation polarity)
```

それぞれの単体の公理は、「基本的な意味論分類は全て互いに素である（disjoint)」ということを表わしている。つまり、インフォンや関数、関係である個体は存在しないということである。

```
(forall (?x ?y)
  (=> (and   (or   (= ?x individual) (= ?x infon) (= ?x relation) (= ?x situation)
                   (= ?x UOB) (= ?x COE) (= ?x process) (= ?x polarity))
             (or   (= ?y individual) (= ?y infon) (= ?y relation) (= ?y situation)
                   (= ?y UOB) (= ?y COE) (= ?y process) (= ?y polarity))
             (=/ ?x ?y))
      (forall ?z (not (and (?x ?z) (?y ?z))))))
```

A.5.2.1　一階関係（First-order Relation）

この論文で取り扱っているバージョンの状況理論では、インフォンは一階関係と個体だけで構成されている。その為、明確な定義と適切なスキーマによる公理化を行うには、一階関係についての観念が必要になる。

(define-relation FO-relation (?rel)
　　:=> (relation ?rel))

「一階関係は、個体が n タプル（n-tuples）の時のみに真となる関係である」

(forall (?rel : (FO-relation ?rel) (=(arity ?rel) n)))
　(forall (*var*$_1$ ... *var*$_n$: (?rel *var*$_1$ ... *var*$_n$))
　　(and (individual *var*$_1$) ... (individual *var*$_n$))))

典型的な一階関係の時間による拡張は、その引数の中に期間というパラメータを含める事によって達成されるということを思い出して欲しい。例えば、関係「walks」は、期間 t を「通して」b が歩いている場合にのみ、個体 b と期間 t の間で維持されている 2 位関係として扱われている。そして期間そのものが個体であることから、上の定義に適応している。

A.5.2.2　インフォン項の公理（Axioms for Infon Terms）

一階関係の観念が与えられると、（null オブジェクトにはならない）正当な（legitimate）インフォンを表現する為に、インフォン項に対して何が必要となるのかということを表現する公理を与えることが可能となる。

「基本インフォン項は、個体を表現した n 個の項を持つ n 位一階関係を表現した 1 個の項と、極性（polarity）を表わす 1 個の項で構成されていた場合にのみ、インフォンを表現する」

(forall (?rel var_1 ... var_n ?pol)
 (<=> (infon (?r var_1 ... var_n ?pol))
  (and (FO-relation ?r)
   (individual var_1)
   ...
   (individual var_n)
   (polarity ?p))))

「合流（conjunctive）インフォン項（分離（disjunctive）インフォン項）は、個々の合流（conjunct）（分離（disjunct））がインフォンを表現している場合にのみ、インフォンを表現する」

(forall (?inf1 ?inf2)
 (<=> (and (infon (and ?inf1 ?inf2)))
  (and (infon ?inf1) (infon ?inf2))))

(forall (?inf1 ?inf2)
 (<=> (and (infon (or ?inf1 ?inf2)))
  (and (infon ?inf1) (infon ?inf2))))

量化インフォン（quantified infon）の相似公理（analogous axiom）は、スキーム（schema）として宣言されなければならない。Q を何かしらの量化記号、var を何かしらの変数、そして infterm を何かしらのインフォン項であるとする。もしも var が、infterm の原子インフォンの中で、第 1 項として自由に発生する場合、次のようになる：

(<=> (infon (*Q var infterm*))
 (*Q* (*var* : (FO-relation *var*)) (infon *infterm*))))

同様に、infterm の原子インフォン項で、var が第 1 項として自由に発生しない場合（発生範囲が制限されている場合、自由に発生するが infterm の原子インフォン項に第 1 項としてではない場合、infterm に全く発生しない場合、等）、次のようになる：

(<=> (infon (*Q var infterm*))
(*Q* (*var* : (individual *var*)) (infon *infterm*))))

A.5.3　基本状況理論関係（Basic Situation Theoretic Relation）

ここでは、基本的な状況理論関係の導入を行い、そして公理を定義する事により、その正しい引数の種類（legitimate argument types）を言明する（delcare）。

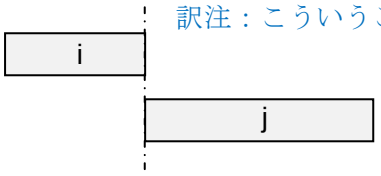
注意すべきは、詳細説明言語の基本部分（core）の基本的な分類（basic category）と共に、ここでは「正統さ（legitimate）」を意味論的に理解し、原理に基づいて（axiomatically）補強していることである。例えそれがどのような項であろうとも、以下のように関係項を導入する事により、文法的に正しい（correct）「文（sentences）」を構成する為に使用可能である。しかし、そうした文は、公理化された制約が満足された場合にのみ、真と成り得る。

```
(define-relation supports (?x ?y ?z)
    :=> (and (situation ?x) (infon ?y) (interval ?z)))
(define-relation denies (?x ?y)
    :=> (and (situation ?x) (infon ?y) (interval ?z)))
(define-relation occurrence-of (?x ?y)
    :=> (and (situation ?x) (UOB ?y)))
(define-relation activation-of (?x ?y)
    :=> (and (COE ?x) (process ?y)))
(define-relation occurs-in (?x ?y)
    :=> (and (situation ?x) (COE ?y)))
(define-relation activity-in (?x ?y)
    :=> (and (UOB ?x) (process ?y)))
(define-relation of-type (?x ?y)
    :=> (and (situation ?x) (UOB ?y)))
(define-relatoin object-in (?x ?y)
    :=> (and (individual ?x) (or (situation ?y) (UOB ?y))))
```

A.5.4　基本時間関係（Basic Temporal Relations）

ここでは、期間と期間の間にある（between intervals）幾つかの基本的な時間的関係について紹介する。IDEF3 で使用されている期間を特徴付けする為に必要な最も重要な関係は、「meets」関係である。これは期間である場合のみ真となり得るものであり、次のような部分定義によって指示される。

訳注：こういうこと？

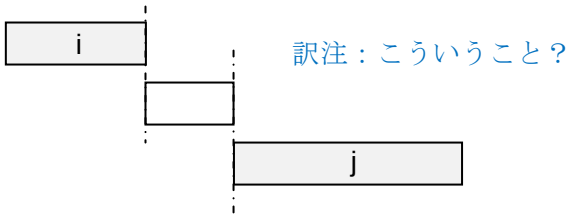



```
(define-relation meets (?x ?y)
    :=> (and (interval ?x) (interval ?y)))
```

直感的に、注意すべき点は、ある期間が他の期間と合う（meets）のは、1 つ目の期間の終点が、2 つ目の期間の始点と一致する場合のみであることだ。この[Allen & Hayes, 1987]に書かれている「meets」関係のロジックは、仮のものである（assumed）（[van Bentham, 1983]も参照のこと）。

meets を用いて、使いやすい時間的関係の一揃い（variety）を定義をすることが可能である。その 1 つが strongly-precedes であり、1 つ目の期間と 2 つ目の期間の間に期間が存在する場合のみ、これはある期間が別の期間よりも強力に先導（strongly precedes）している。

訳注：こういうこと？



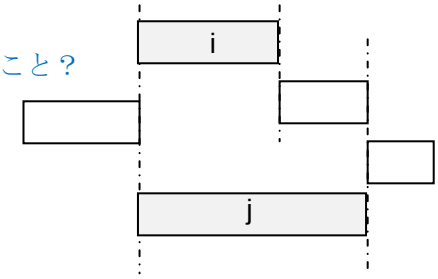

```
(define-relation strongly-precedes (?x ?y)
    := (exists (?z ?w : (/= ?z ?w)) (and (meets ?x ?z) (meets ?z ?w) (meets ?w ?y))))
```

注意すべきなのは、点（point）は期間（interval）であることから、?x と?y の間に 2 つの別の期間を入れ込まなければならない事である。実際に、ある点は、それ自身に meets する間隔として定義する事が可能である（これについては、Allen と Hayes の間で意見が分かれている）。

(define-relation point (?x)
:= (meets ?x ?x))

ある時間的期間 i が、別の時間的期間 j を start するのは、i と j が共にある期間によって meets されており、かつ i が meets した期間が meets した期間に、j が meets している場合のみである：

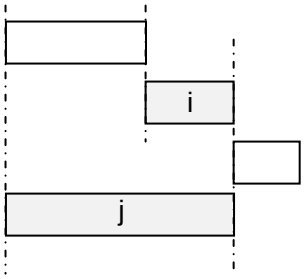
訳注：こういうこと？



(define-relation starts (?x ?y)
:= (exists ?z (and (meets ?z ?x) (meets ?z ?y)
(exists ?w (and (meets ?y ?w) (strongly-precedes ?x ?w)))))

同様に、時間的期間 i が別の期間 j に finish するのは、i と j が共に対象となる期間に meet するも、j を start した期間が i に meet している場合のみである。

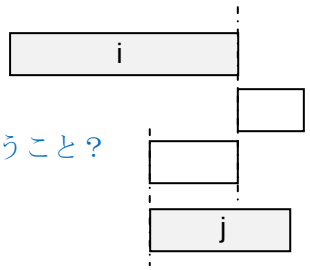
訳注：こういうこと？



(define-relation finishes (?x ?y)
:= (exists (?z ?w : (starts ?w ?y)) (and (meets ?x ?z) (meets ?y ?z) (meets ?w ?x))))

そして i が j を overlap するのは、i を finish するある期間が、j を start する場合のみである。

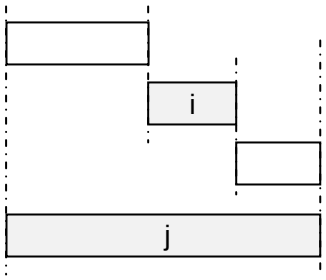
訳注：こういうこと？



(define-relation overlaps (?x ?y)
:= (exists ?z (and (finishes ?z ?x) (starts ?z ?y))))

更に、i が j を during しているのは、j を start するある期間が i に meet し、そして j を finish するある期間に i が meet している場合のみである：

訳注：こういうこと？



(define-relation during (?x ?y)
:= (exists (?z ?w) (and (starts ?z ?y) (meets ?z ?x) (finishes ?w ?y) (meets ?x ?w)))

他の便利な関係もまた、同様なやり方で定義して行くことが可能である。

A.5.5　状況発生期間（The Interval Over Which a Situation Occurs）

プロセスとその活動を記述する際に、対象となる状況が発生している期間について話をできるようにしておくことは、非常に重要である。そこで、対象となる状況に適用する事で、その状況の発生期間を正確に返す関数を定義しておく：

```
(define-function interval-of
    :=> (forall (?sit ?t)
                (=> (= (interval-of ?sit) ?t)
                    (and (situation ?sit) (interval ?t)))))
```

時間的関係が与えられたら、状況の発生している期間を定義する事により、状況間で発生する便利な時間的関係の取り揃え（variety）を、状況が発生している複数の期間の間の時間的関係を一致させると言う手法を用いて定義する事が可能になる（？）。詳細については[Menzel & Mayer, 近日公開]を参照して欲しい。

A.5.6　分類変数の使用（？Using Sorted Variables）

IDEF3 詳細説明言語の正しい書式というものは完成しておらず、特に変数は、種類が 1 つしか無い。しかし、A.4 章と 3 章で説明された非公式な例では、幅広い種類の変数を、その値が限定されてはいるものの、UOB や状況、インフォンといった様々な意味論分類として使用している（？）。

制約された変数の多くの異なった分類を持つ、複数分類言語（？many-sorted language）と呼ばれているものの中の多くの文が、詳細説明言語のような単一分類言語（？single sorted language）の一文へと直接的に翻訳可能であるように、こうした実践（practice）は単に代替記法（？alternative notation）を便利に使っただけのものであると解釈することが可能である。

そのトリック（trick、秘訣、ごまかし）は、単一分類言語の中で、本来は複数分類言語のみに制限されている分類変数（sorted variables）を使用する意味論分類を表現した項を、

使用するだけである。

例えば、「?sit」の提案（support）は、状況に対してのみ割り当てられている分類変数（？sorted variable）であり、また「?ind」は、個体に対してのみ割り当てられている分類変数である。そして、文「(forall (?sit ?ind：(FOO ?sit)) (BAR ?ind ?sit)))」は、どのような状況 s と個体 b に対しても、もしも状況 s が FOO であれば、b は s に対して BAR を生成するということを言っている。しかし明確にする為に、単一分類詳細言語を用いた「(forall (?x ?y：(situation ?x) (individual ?y) (FOO ?x)) (BAR ?y ?x))」という厳密な表現にすることも可能である。

分類変数（？sorted variable）を使用する事は、従って悪気のない（innocuous）事であり、実際に、そうした使い方は典型的に、単一分類表記法（single-sorted notation）で記述された一文の長さを劇的に短くすることになることから、むしろ推奨されている。複数分類（many-sorted）言語と単一分類（single-sorted）言語の間の関係についての厳密な考慮（rigorous account）については、第 4 章の 4.3[Enderton, 1972]を参照して欲しい。

多くの場合、複数分類言語を使っているのか、それとも単一分類言語を使っているのかに関係無く、一般的にその意図された意味論分類を反映させた、「?ind」、「?rel」、「?f」、「?sit」、「?inf」、「?uob」、「?coe」等の、変数を選択するのが実践的である。

訳注：最後の最後ですら、良くわからない。

「sorted variable」や「single-sorted language」、「many-sorted language」という単語からして、よくわからない。

「interpretation(logic)」についての Wiki では：

> **2 つ以上の異なった「sort of variables」を持つと言う事は、異なった種類のオブジェクトを表現するものであり、それぞれの「sort of variables」毎に異なった「domain」を持つ。そして「many-sorted logic」の例として「平面ユークリッド幾何学」を挙げ、2 つの「sort」として点と線を示している。点は点、線は線でそれぞれ関係シンボルを持ちつつ、点と線の間にも二元投射関係 E を持っている。**

としている

そうした辺りの雰囲気を感じつつ、もう一度上の半端な訳文を読むと、**「?ind」、「?rel」、「?f」、「?sit」、「?inf」、「?uob」、「?coe」等の、IDEF3 に関係する変数を厳密な定義なく使っても構わないんじゃね？**くらいの意味合いの文章なのでは、と思われる。

付録 B　IDEF3 用語集

用語（英語／訳語）	説明
Activation 活動	スキーム図によって表現されているプロセスの中の、全てもしくは幾つかの UOB のインスタンスの内、その時間的・ロジック的特性が、スキーム図に明記された時間的・ロジック的条件を満足しているものの集合
Conditions, Entry 進入条件	リンクの変移条件が満たされることで、リンクの始点の状態からリンクの終点の状態へとオブジェクトが変移中である際に、あるオブジェクト（変移中のオブジェクトとは別物である事もある）がリンクの終点の状態へと進入する為の必要十分条件（訳注：相当に意訳）。進入条件は、オブジェクト状態と変移リンクの間にあるインターフェースと本質的に関係している。（訳注：本文 P.83、P.84 を参照。以下同）
Conditions, Exit 退出条件	あるオブジェクトが、対象となる状態から離脱する為に必要となる、十分条件。退出条件は、オブジェクト状態と本質的に関係している。
Conditions, State 状態条件	対象となる状態である為の、個別な必要条件。状態条件は、オブジェクト状態と本質的に関係している。
Conditions, Transition 変移条件	始点の状態から終点の状態への変移が試みられる為に、個別に必要で、合わせて十分な条件（？）。 変移条件は、オブジェクト状態と変移リンクの間にあるインターフェースと本質的に関係している。
Constraints 制約	一般的には、あるシステム（system、系）で維持されなければならない（もしくは維持してはならない）事の宣言。殆どの場合、制約は、そのシステムが意図された機能を果たす為に維持されなければならない、領域上のオブジェクトか、もしくはオブジェクト間の結合のロジカル的な特性を表現している。 制約は、あるプロセス上のオブジェクト間もしくはプロセス間で維持されていることが知られている、識別された条件である。
Context 背景、文脈	研究中の系の中の、記述作成活動が発生する境界・周囲を確立する境界として識別されている部分。もしくは、分野専門家の知識のドキュメント化と翻訳に使用される設定。
Context statement 背景（文脈）宣言	IDEF3 プロセス記述作成活動の境界（通常はシステムのどの部分が含まれ、どの部分が除外されるかと言うやり方で表現される）と必要な詳細レベルを識別する、書かれた宣言。背景宣言は IDEF3 記述概要フォームに書かれる。

用語（英語／訳語）	説明
Decomposition 分解	1 個の UOB を多くの文脈によって解釈し、それを他の複数の UOB を用いて表現した記述の中の 1 つ。 スキーム図は、1 個のプロセスについてのより詳細な視界（view）もしくは、異なった物の見方（perspective）を、明確に定義された視点と共に、提供してくれる。
Description 記述	分野専門家の知識もしくは経験の領域（realm）の範囲内の物事についての、事実もしくは信念を記録したもの
Domain 分野	半導体分野とか抽象代数学の分野といった、興味（interest）の範囲。分野は、分野内に典型的に存在しているオブジェクトやプロセスの特徴的な種類について語る為の独特な語彙を持っている。
Domain expert 分野専門家	ある分野に関して、ある明確な面から識別された特徴について、多くの知識を持ち、また深く精通していると考えられている個人。興味対象分野の知識の、主要な情報源の役割を果たす。
Elaboration 詳細説明	詳細説明は、スキーム図上にある IDEF3 要素（UOB やオブジェクト状態、交差点、リンク等）の詳細な特徴を提供する。
Elaboration Language 詳細説明言語	プロセス関連情報の表現に特化された、構造化された文字言語。IDEF3 詳細説明言語は、一階様相論理（first-order modal logic）と集合論（set theory）の記述に全力を発揮する。
Facts 事実	現実世界で保持されている関係性。事実はオブジェクトについて成されている主張（assertion）である。
Form, IDEF3 Description Summary IDEF3 記述概要フォーム	開発中か、もしくは完成したプロセス記述の概要についてまとめられた、構造化されたドキュメント（structured document）。プロジェクトの目的と背景（context）を記録し、プロセス記述を記録する為に使用されている全ての意味論とドキュメントについての概要を提供している。
Form, Elaboration 詳細説明フォーム	スキーム図上にある IDEF3 要素（UOB やオブジェクト状態、リンク、交差点等）の詳細な特徴を提供する為に使用されている、構造化されたドキュメント。詳細説明ドキュメントには、一般的に以下の物が含まれている： ①要素の名前、ラベル、番号。 ②要素に関連している、オブジェクト種類、インスタンス、事実、制約のリスト。 ③要素についての文字による記述

用語（英語／訳語）	説明
Form, IDEF3 Schematic IDEF3 スキーム図フォーム	全ての IDEF3 フォームの基本構成。IDEF3 スキーム図フォームは 3 つの主要な区画に分割される： ①作業情報（上部）：キット批評作業の支援に使用される ②情報伝達欄（中央部）：伝達すべき主要な情報が記載される この欄には通常、スキーム図に使用されるが、それ以外の目的（用語集、チェックリスト、注記、スケッチ等）にも使用可能である ③識別欄（下部）：フォーム上の情報の背景と目的を確立する
Form, Object Schematic Summary オブジェクトスキーム図概要フォーム	IDEF3 オブジェクトスキーム図の内容の概要を説明した、構造化されたドキュメント
Form, Pool プールフォーム	記述の作成中に識別されたシナリオ、オブジェクト、UOB、オブジェクト状態をリストにする為に使用される構造化ドキュメントであり、そうした要素の素材資料への追跡可能性を提供する。
Form, Process Schematic Summary プロセススキーム図概要フォーム	IDEF3 プロセススキーム図の内容の概要を説明した、構造化されたドキュメント
Form, Source Material Description 素材資料記述フォーム	素材資料として追跡されている個々の項目（item）についての、より詳細な情報を記録する為に、素材資料記録（Source Material Log）と共に使用されている、構造化ドキュメント。特に、このフォームは、素材資料の中で議論されている主要な概念の簡潔な概要を獲得と、素材資料から IDEF3 記述要素への追跡可能性の提供の目的で使用されている。
Form, Source Material Log 素材資料記録フォーム	プロジェクトを通して収集された全てのデータの識別と追跡に使用されている構造化されたドキュメント。素材資料記録は、IDEF3 プロジェクトで収集、使用されている全ての素材資料への主要な索引（index）として用いられている。
IDEF	統合定義（Integration Definition）の頭文字。特に IDEF0、IDEF1、IDEF1X、IDEF3、IDEF4、IDEF5 を含んだ、経営統合の為の相互補完手法（mutually-supportive methods）のファミリーのことを指し示す為にも使用されている。

用語（英語／訳語）	説明
IDEF0	機能モデリングの為の IDEF 手法
IDEF1	情報モデリングの為の IDEF 手法
IDEF1X	意味論データモデリングの為の IDEF 手法
IDEF2	シミュレーションモデリングの為の IDEF 手法
IDEF3	プロセス記述獲得の為の IDEF 手法
IDEF4	オブジェクト指向設計の為の IDEF 手法
IDEF4/C++	C++オブジェクト指向プログラム言語の実装に特化された､オブジェクト指向設計の為の IDEF 手法
IDEF5	存在論（オントロジー、Ontology）記述獲得の為の IDEF 手法
Individual 個体	現実世界オブジェクトの、最も論理的な基本種類（most logocally basic kind)。判り易い例を挙げると、人間や、具体的物理オブジェクト、プログラムのような明確な抽象オブジェクト等である。より高位の普遍的な論理階位（logical order）にある特性や関係といったオブジェクトとは異なり、個体は本質的に複数のインスタンス化ができない（are not multiply instantiable)。また、個体は一階（first-order）オブジェクトとしても知られている。
Instance インスタンス	ある活動（avtivation）に関する、発生し得る活動パターンの中の 1 つが表現されている、特別な場合
Interview インタビュー	調査のある方向性を追求する為に、分野専門家と顔を突き合わせて行う会合（meeting）
Junction 交差点	論理分岐（logic branching）を図示する為のメカニズムを提供する IDEF3 スキーム図言語の要素
Kit キット	ダイヤグラムやテキスト、用語集、決定概要（decision summary)、背景情報（background information）等の IDEF3 記述の構成物を、批評やコメントを行えるようにパッケージ化してまとめたもの。IDEF3 には、オブジェクトキット、シナリオキット、記述キットの 3 種類のキットがある。
Kit, Description 記述キット	IDEF3 記述とその関連ドキュメントに、対象となるプロジェクトの全てのシナリオを含めることにより完成したシナリオキットとオブジェクトキットを、編集したもの。承認を受けた記述キットは、開発作業の中の最終配布物の 1 つとなる。

用語（英語／訳語）	説明
Kit, Object オブジェクト キット	1 個以上のオブジェクトと、それに関連する全てもしくは部分的なドキュメンテーションについて言及したキットである。オブジェクトキットには、オブジェクトスキーム図や詳細説明が含まれており、批評とコメントを行えるようにパッケージ化されている。
Kit, Senario シナリオキット	1 個のシナリオと、それに関連するドキュメンテーションについて言及したキットである。シナリオキットには、プロセススキーム図とそれに関連した UOB 分解物、UOB 詳細説明、リンク詳細説明等が含まれており、批評とコメントを行えるようにパッケージ化されている。
Kit Contents Sheet キット用内容物 シート	キットの構成物リストにより多くの欄が必要な場合に使用される、キット用表紙シートを拡張したもの。
Kit Cover Sheet キット用 表紙シート	IDEF3 キットとしてまとめられた資料と、批評要求物、そしてキットの内容の索引を、識別する構造化されたドキュメント
Kit Review キット批評	IDEF3 プロセス記述の検証に用いられている批評と承認のプロセス
Link リンク	IDEF3 スキーム図言語の構文要素であり、他の IDEF3 構文要素を連結する為に使用される。リンクは、UOB やオブジェクト状態、オブジェクトの間の重要な関係性を表現する。IDEF3 リンクとして重要となる関係の種類には、時間的、論理的、因果的、本質的、慣習的な関係がある。
Link, Constrained Precedence 制約順序リンク	順序リンクの特殊形であり、単純な順序という活動の意味論に、追加の制約を課したもの。順序リンクを参照。
Link, Dashed 点線リンク	IDEF3 スキーム図言語の構文要素で、プロセススキーム図上の 2 つの UOB 間の（恐らくは制約）関係の存在を強調する為に使用される。点線リンクは、事前に定義された意味論を持っておらず、その為、「s」リンクもしくは「ユーザー定義」リンクと呼ばれている事が多い。
Link, First-order 一階リンク	一階オブジェクト間で維持されている関係

用語（英語／訳語）	説明
Link, Precedence 順序リンク	IDEF3 スキーム図言語の構文要素で、ある UOB のインスタンスと別の UOB のインスタンスとの間にある、時間的順序関係を表現する為に使用される。
Link, Relation 関係リンク	IDEF3 スキーム図言語の構文要素で、オブジェクトスキーム図上の、オブジェクト間、オブジェクトとオブジェクト状態の間、オブジェクト状態間、等で維持されている、追加関係を表現する為に使用される。
Link, Second-order 二階リンク	一階オブジェクトと二階オブジェクトの間や、特性や関係の間で維持されている関係。二階オブジェクトや特性、関係同士の間で維持されている関係。
Link, Strong transiton 強変移リンク	「同一の」オブジェクトが開始状態から終了状態へと変移しているという追加情報を持つ、変移リンクの特殊形
Link, Transition 変移リンク	IDEF3 スキーム図言語の構文要素で、IDEF3 オブジェクトスキーム上で、ある状態から出発し、別の状態へと至る、「変移」関係を表現する為に使用される。
Method 手法	ある作業のセットを完遂する為に構成された、単一目的の規律もしくは実践（discipline or practice）。IDEF 手法は、プロセスの習得を加速し、また素人実践者でも、特定の分析活動や設計作業に従事する経験豊富な個人に近い効率を発揮できるよう補助できるように、特化されている。IDEF 手法は、規律化されたアプローチを通して、優秀な実践経験と調和させつつ、高い効率（や品質、生産性）を矛盾なく達成できるように、ユーザーを導いてくれる。
Model モデル	対象となる現実世界のシステム（system、系）の特徴を、明確な適切な曲面から模倣する為に作られた、オブジェクトと特性、関係で構成された理想的なシステム。 モデルは、ある分野内の予め定義された興味範囲についての信頼ある予測因子（predictor）に「十分に近い」と仮定された、理想化されたシステムである。（訳注：P.12 を参照）
Needs statement 需要宣言	要求元（人もしくはプロジェクト）を記録し、プロジェクトの目的を意訳（paraphrase）する宣言文（statement）

用語（英語／訳語）	説明
Note box 注記箱	IDEF3 スキーム図言語の構文要素であり、以下の目的で使用される： ・特定のオブジェクトや IDEF3 要素と関連する関係が注記箱を取付けた部分に関与していることを強調 ・参照データや参照オブジェクト（スクリーン配置等？）といった明確な例と結び付ける ・対象となる詳細説明と関連する特別な制約セットを強調する
Object オブジェクト	プロセスに関与している個体、もしくは個体のクラス。 個体（individual）を参照
Object, First-order 一階オブジェクト	個体（individual）を参照
Object, Second-order 二階オブジェクト	個体のクラスと一階関係は、二階オブジェクトである。 個体（individual）と一階リンク（First-order link）を参照
Object State オブジェクト状態	明確な特性もしくは条件を表現する個体、もしくは個体のクラス（一般的に、普通名詞よりも形容詞によって指示されることになる）。例えば、武器システム開発プログラムは、日程節点（milestone）の決定によって開始される状態変移として見ることが出来る、幾つもの異なった段階を経由している。（訳注：何が例なのか判らない）
Occurrence 発生	あるシナリオ活動の中での UOB のインスタンス。「発生（Occurrence）」は、IDEF3 プロセス記述のある部分で、IDEF3 プール項目（シナリオや UOB、オブジェクト、オブジェクト状態）を使用することを示す為に使用されることもある（？）。例えば、同一の UOB プール項目は、同一のプロセススキーム図に 2 か所以上で使用される。（訳注：何が例なのか判らない）
Process プロセス	1 個以上の個体が関連する、ある時間的期間（瞬間的期間でも可）の、現実世界のイベントもしくは物事の状態。典型的に、あるプロセスは、プロセス上の 1 個以上の個体の特性において、ある種の変化に関係している。プロセスインスタンスとして参照されることもある。
Property 特性	複数並列にインスタンス化可能な、抽象的で、一般的な特徴（feature or characteristic）。同一の特性が、別個の個体によって共有されることもある。

用語（英語／訳語）	説明
Purpose 目的	IDEF3 記述開発活動に従事することにより、達成される目標もしくは終点。スキーム図と詳細説明を作成する事そのものが目的である場合には、単純にプロセスをドキュメント化することが目的になる。しかし多くの場合、IDEF3 記述の作成は、何かしらの発見や経営決定活動を補助しつつ、継続されるものである。
Purpose statement 目的宣言文	以下のいずれかのものを明確にする書式化された宣言： ①作業の主要な目標 ②記述が満足すべき需要 ③顧客が回答を望んでいる質問や発見 また目的宣言文は、2 つの部分へと分割する事が可能である： （1）需要宣言文（Needs Statement）の定義 （2）プロセス記述がどのようにして使用されるのかという形での知識目標の定義 このような目的宣言文は、IDEF3 記述概要フォームによってドキュメント化される。
Referent 参照	IDEF3 スキーム図言語の構文要素であり、UOB シナリオもしくは変移スキーム図へと参照する為に使用される。
Relation 関係	2 個以上のオブジェクトの間で保持されている、抽象的で普遍的な関連、もしくは結合。特性と同様に、関係もまた、複数並行的にインスタンス化することが可能である（別々のオブジェクトで共有可能、等）。あるインスタンスで、オブジェクト間で関係が保持されている場合、このオブジェクトは引数（argument）と呼ばれている。
Relation, Temporal 時間的関係	時間的位置や期間の間の関係。before, during, overlap 等
Role, Analyst 分析者	IDEF3 記述の主要な開発者。
Role, Commentor コメンター	IDEF3 プロジェクトチームのメンバーであり、仮図（draft）の IDEF3 記述の批評を行い、批判を書類にまとめる。
Role, Librarian 司書	ドキュメントのファイルの維持管理や、コピーの作成、IDEF3 キットの配布、記録の保持を担当する。

用語（英語／訳語）	説明
Role, Project leader プロジェクト リーダー	IDEF3 記述開発作業の監督と指導を担当する、管理者の役割を果たす。プロジェクトリーダーは、特に、記述開発作業の成果やチーム組織、リーダーシップ、そして日程管理と予算管理に対して究極的な責任を負っている。
Role, Reader 読者	IDEF3 プロジェクトチームのメンバーで、仮図の IDEF3 の批評を行うが、コメント書類の作成は行わない。
Role, Reviewer 批評者	対象となる分野と（もしくは）IDEF3 手法の知識を持ち、仮図の記述とドキュメントの批評と（もしくは）コメントを行う、IDEF3 プロジェクトチームメンバー。チームメンバーと分野専門家が、批評者に成り得る。読者とコメンターを参照。
Role, Team member チームメンバー	IDEF5 存在論（オントロジー）記述プロジェクトに関係する人 （訳注：なぜ、突然に IDEF5 の事が出てくるのか不明）
Schematic スキーム図	IDEF3 スキーム図言語の語彙を用いて、言語の文法ガイドラインに従って構築される、連結されたダイヤグラム。IDEF3 スキーム図言語によって図形化されることにより、プロセス記述の構築が促進される。
Schematic, Process プロセススキーム図	あるシナリオをプロセス中心視界（process-centered view）によって獲得、表示することを支援する IDEF3 スキーム図
Schematic, Object オブジェクトスキーム図	興味対象にある 1 個以上のシナリオをオブジェクト中心視界によって獲得、表示することを支援する IDEF3 スキーム図
Schematic, Transition 変移スキーム図	オブジェクトスキーム図の一種で、シナリオ（もしくはプロセス種類）のインスタンスの中に関わっているオブジェクトによって追跡された状態変移を特徴付けている。
Schematic, Enhanced Transition 拡張変移スキーム図	シナリオに関係するオブジェクトと関係についての背景設定情報（context-setting information）を含むが、直接の興味対象の状態変化挙動は表示していない、変移スキーム図。
Senario シナリオ	それを分解できるという意味合いにおいては、どのような UOB もシナリオに成り得る。

用語（英語／訳語）	説明
Source material 素材資料	テキストブック、研究論文、経営マニュアルや手続マニュアルといった経営仕様ドキュメント、インタビューメモのセット、観察記録、等の、プロセス記述開発プロジェクトに関係した情報を持つもの
Unit of Behavior（UOB） 行動ユニット（UOB）	IDEF3 で、「出来事（happenings）」の種類を記述する為に使用されている用語。機能やプロセス、シナリオ、活動、運用、決定、行動、イベント、手続、といったものの概念は、ある制限された挙動に関係する「出来事」を表現している。UOB という用語は、こうした概念をカプセル化している。
UOB Box UOB 箱	IDEF3 スキーム図言語の構文要素で、現実世界のプロセスを表現する為に使用されている。
UOB, Child 子 UOB	UOB を分解してできる UOB
UOB, Parent 親 UOB	あるシナリオの役割を演じている UOB で、プロセス記述の背景（context）を確立している。
Validation 検証	構築された IDEF3 プロセス記述が文法的に、意味論的に正しい事をチェックし、保証するプロセス。IDEF3 プロセス記述の検証の主な手法は、キットの検証と承認を通して行われる。
Validation, Syntactic 文法の検証	構築された IDEF3 スキーム図が、IDEF3 言語の文法ルールを満足していることをチェックし、保証するプロセス。
Validation, Sematic 意味論の検証	IDEF3 記述にある宣言文（statement）が、実際に分野専門家の主張を獲得しているかどうかをチェックし、保証するプロセス。
Viewpoint 視点	システムもしくはプロセスを調査、記述する際に取られる物の見方（perspective）。役割に特有で客観的な視点は、IDEF3 の UOB 分解機構を使用する事で獲得される。