# 要求分析のための合成型手法工学アプローチ

## Compositional Method Engineering for Requirements Analysis

妻 木 俊 彦

**要 約** システム開発プロジェクトは，しばしば予期せぬ障害に遭遇するが，これまでの固定的な手順に基づいた手法によって，それらの障害状況を解決することは困難であった．われわれは，一つの手法だけを使うのではなく，それぞれの障害状況に対応してそれらの手法が持っている基本的な操作を柔軟に組み合わせて使う方がより効果的であると考えた．本稿では，いくつかの著名な要求分析手法を分析し，それらの手法が共通に持っている基本的な操作を選択的に組み合わせて使用する方法を提案する．これを合成型手法工学と呼ぶことにする．

**Abstract** Although requirements projects often encounter to unexpected obstacles, it is difficult to rescue the project from the situations through an advanced method because the situation includes various problems. We thought it is suitable to compose appropriate primitive techniques by each situation rather than to use only one method with a fixed proceodure. So,we extracted common primitive techniques from some typical methods and composed them. We called the approach compositional methods engineering. In this paper,we report two examples of compositional methods.

## 1. はじめに

要求仕様の作成は，ソフトウェア開発ライフ・サイクルにおける最初の重要なプロセスである．何故なら，このプロセスを通して明確に記述された要求仕様に基づいてシステムが構築されるからである．よいシステムを構築するためには，正確で完全かつ本質的な要求仕様を定義することが重要である．

要求プロセスは，大きく要求分析と要求定義というサブプロセスに分けられる．要求分析では，要求者すなわち問題領域の専門家やシステムの利用者などから要求を聞きだし，それらの要求の正当性や完全性を評価し，システム設計に必要且つ十分な情報を収集する．一方，要求定義では，分析された要求の内容を，開発者に正確に伝えるために，厳密な表現形式で記述する．こうして，要求仕様書が作成され，顧客の要求がシステム設計者に伝えられる．

問題領域という複雑で曖昧な領域を取り扱うこと，問題領域の専門家やシステムの利用者である要求者とシステム開発者である要求分析者という立場の異なる人間同士の共同作業であることなど，他のシステム開発作業には無い不確定要素が多分に含まれていることから，要求分析作業は，特に難しい作業であるといわれている[1]．さらに，問題領域の専門家ではない分析者にとって，要求者からすべての要求を聞き出し，その問題領域の知識を正しく理解することは，この作業を一層困難にしていると言える[2]．

正しい要求を正確かつ漏れなく収集するために，要求分析作業を支援するための様々な手法が提唱されている．しかし，あるプロジェクトにとって有効な手法が，他のプロジェクトにと

っては使い難い手法であったとか，ある問題領域では効果的だった手法が，他の問題領域では不適切な手法であったということを，われわれはしばしば経験している．要求分析プロジェクトは，その性格上，様々な予期せぬ状況に遭遇する．一方，これまで提案されてきた高度な手法の多くは，それぞれの開発者の経験と思考法をもとに，彼らがその手法開発の前提として想定したプロジェクトやアプリケーションの特性に依存して作成されているため，その前提に合致した状況では極めて有効に機能するが，そこからはずれた状況では上手く機能しないということが起こる．こうして，個別のプロジェクトと手法の不一致が発生する．要求分析作業における障害は，要求分析手法とプロジェクトが置かれた状況との不適切な組み合わせにあると考えられる．ここで言うプロジェクトの状況には，プロジェクトが達成しなければならない課題とともに，システム化対象領域の大きさや複雑さ，プロジェクトのスキルや思考スタイルなどが含まれる．ある問題を抱えた状況に置かれたプロジェクトは，その問題を解決することによって新らたな状況に遷移する[3]．このように遷移する状況の中で，どのような要求プロジェクトの状況にも対応できる手法を作り上げることは難しく，それ故，要求プロセスは，他のソフトウェア開発プロセスに比べ，依然として人間の経験とスキルに依存する割合が高いという状況から脱却できないでいる．

われわれは，こうした要求分析プロジェクトの多様な状況に対応するためには，想定された特定の状況に合わせて事前に作成された固定的な手法を使用するよりも，プロジェクトの状況に合わせて，様々な基本的な操作を柔軟に組み合わて使用できるようにする方が有効であると考えた．こうした方法は，高度な手法がもつ，作成者の個人的な想定という束縛から，手法を解放してくれるからである．本稿では，多くの手法が共通に持っている基本的な操作を組み合わせて使用する合成型手法工学（Compositinal Method Engineering）を提案する．

## 2. 要求分析手法の分類

分析者が要求者から要求を引き出し，それを洗練するための様々な要求分析法が提案されている．こうした分析法は，要求者が潜在的に記憶し，理解している問題領域の知識や個別の要求項目を引き出すためのもので，要求者はこうした手法の助けを得て記憶のなかに埋もれている知識を拾い上げてくる．すなわち，要求分析法の基本的な戦略は，要求者の連想を支援することである．

これらの手法は，その特長によって，ユーザからニーズや意図を聞き出す要求抽出型，システム開発の目標の中から要求を導き出す目標指向型，問題領域のモデルに付随している要求を見つけ出す領域モデル型という 3 種類に分類することができる[4]．しかし，いずれの場合にも，問題領域の知識や個別の要求項目は，要求者からの具体的な提示を受けなければならない[4]．

### 2.1 要求抽出型手法

要求抽出型手法は，要求者から問題領域の知識や個別の要求を聞き出して，それらを記述するための手法である．要求抽出型手法には，個別の要求を直接聞き出す方法と，問題領域の知識を聞き出し，そこから要求を発見するという二つの方法がある．

システムのユーザや問題領域の専門家から，複雑なシステム化対象領域（問題領域）全体の要求を，直接聞き出すのは難しい．そこで，ユーザから要求を聞き出す多くの手法では，問題領域を，一旦，より小さな領域に分割してから要求抽出作業を行うという方法がよく用いられ

る[6, 7]．ここでは，問題領域は幾つかの小問題領域に分割される．小問題領域は，元の問題領域よりも，その構成要素の数が少なく，それ故，より単純である．単純な問題領域の方が要求を見つけ出すことが容易である．この手法のもう一つの特徴は，問題領域の全体を網羅することができるということである．元の大きな問題領域では，特徴の少ない要求は見落とされてしまう恐れがあるが，分割された小領域の集合が元の問題領域を漏れなくカバーしていれば，それぞれの小領域ごとに要求を抽出していくことは，問題領域の中を満遍なく調べたことになる．

最も一般的な分割は，銀行預金を普通預金と当座預金と定期預金及び定期積み金に分割するように，問題領域を，その概念的な構造を用いてトップダウン的に分割する方法である．しかし，この方法では問題そのものが明確にならないという指摘もある[7]．また，販売計画，生産計画，在庫計画というように，問題領域を作業工程によって分割する方法もある．この方法は，作業に焦点を当てることが多いので，ビジネス環境全体の機能要求を抽出するのに向いているため，要求分析の前段階のビジネスモデリングなどでよく用いられる[8]．システム化対象領域を直接分割するのではなく，他の対象領域を分割し，それをシステム化対象領域に写像する場合もある．ユースケース法[9]では，問題領域そのものを対象とするのではなく，システムの利用者であるアクターを通して要求を抽出しようとする．この場合にも，アクターをヒューマンアクターとシステムアクターに分けたり，ユーザとオペレータに分けたりして，要求抽出を容易にしようとしている．分割される問題領域はシステム化対象領域とは限らない．FURPS モデル[10]や ISO/IEC 9126[11]では，要求そのものを問題領域とし，要求の概念的な性質によって要求をカテゴリー分割している．FURPS モデルでは要求を，機能性，使用性，信頼性，性能及び支援性という五つのカテゴリーに分類している．要求は，それぞれのカテゴリー毎に抽出される．

このように，要求を直接抽出する型の手法には，次のような操作がある

・業務領域分割：対象問題領域を，業務の特性によって分割する
・作業工程分割：対象問題領域を，作業工程によって分割する
・要求元分割　：要求を，要求の発生源によって分割する
・要求範疇分割：要求を，その特性によって分割する

要求抽出型手法のもう一つの方法は，問題領域内部の構造や，そこで発生する変化を知識として聞き出し，そこから要求を抽出する手法であり，その典型がシナリオである．シナリオは，問題領域の変化を物語として記述するもので，登場人物の行動や問題領域内の変化をテキストとして記述する．その記述は冗長性を含み，厳密性に欠ける恐れがあるが，それだけ多様な情報と想像力を読み手に提供してくれる．要求者から聞き出した豊富な知識を元に，問題領域の将来予測などのさまざまな分析が可能である[12]．しかし，生の知識の中には，複数の要求者同士の間で，互いに対立する知識や要求が提示されることがあり，その解決には，供給者間の調整など情報分析以外の手法が必要になる場合が多い．ユースケース法では，それぞれのユースケースごとにシナリオを作成し，要求を具体化している．ここでは，シナリオは，システムの外部仕様を定義するとともに，システム境界を定義するためにも使われている．i*法[13]では，問題領域内のエージェント同士の依存関係が，物語風の図式で表現される．明示的ではないにしても，この図式の背後にはシナリオがあると考えてよい．このモデルにコンピュータ・システムを介在させることによって，システム機能を明らかにしている．ユースケース法が，ぞれぞれのユースケースをシナリオによって具体化しているのに対し，i*法では，はじめにシナリ

オがあり，そこから個別の要求が抽出されていると考えられる．

### 2.2　目標指向型手法

目標指向型分析法は，要求の意味を扱う手法である．システムには何らかの目標があるはずで，個別の要求はその目標を実現するための手段であるという考えに基づいている．この手法は，古くから，経営科学やオペレーションズ・リサーチでも用いられてきた手法であり，提示された要求を系統的に分析することによって，抽象的な要求を具体化したり，その実現方法を導出するために使用される．また，提示された要求の正当性や厳密性を検証したり，新たな要求を発見するのにも有効である．しかし，要求の分析が恣意的になりすぎたり，要求が目標に特化し過ぎたりすることによって，柔軟性に欠ける場合がある．この種の手法では，分析結果を AND/OR 関係を持つ木構造モデルで表現する．i*法では，問題領域内のエージェント間の依存関係から，タスクを抽出し，タスク分解によって具体的な機能への展開を図っている．

最も一般的な分解法は，ゴール分解とも呼ばれ，提示された要求の実現手段を分析するものである[14]．手段分解では提示された要求に対し「how」という質問を発し，もとの要求（ゴール）を，それを実現するための手段である下位要求（サブゴール）に分解する．手段分解の結果として，サブゴールによる AND/OR グラフが作成される．手段分解を繰り返し，代替関係にある多くの下位要求が求められたとき，どの要求を採用すべきかの指針を，分解木の弧に効果とリスク，あるいは作業量の指標を設定することによって求めることができる[15]．

もう一つの分解は，その要求が提示された理由を分析しようというものである．目的分割によって得られた上位目的は，もとの要求より，より本質的な要求であり，これによって要求の正当性が評価できる．目的分解では，もとの要求に対し「why」という質問を使用する．矛盾した複数の要求が提示された場合，それぞれの要求の上位要求を調べ，より上位レベルでの要求の比較によって，矛盾を解消できる場合がある．

要求は，「why」や「how」のほかに，「who」，「what」，「when」，「where」などの質問を使用して分解することによって，要求の文脈を明確にすることができる．KAOS 法[16]では，格文法[17]に類似したメタクラスのもと，「how」によって Action を，「who」にとって Agent を，「what」によって Entity を，「when」によって Event を，「on what」によって Object と Relation を導出し，それらの結果を形式的に記述する．また，Zachman らは，これらの疑問詞を使用して業務情報アーキテクチャを構築している[18]．

目標指向型手法には，次のような操作がある

・目的分解：提示された要求の上位目標を求める
・手段分解：提示された要求やシステム目標から，その実現手段を導出する
・文脈分解：提示された要求の背景を定義する

### 2.3　領域モデル型手法

そもそも要求というのは問題領域の中に存在しているものであるから，まず，問題領域をモデル化する必要があるというのが領域モデル型手法の考え方である．問題領域には様々な要素が含まれ，そのままでは要求を発見することが困難な複雑な系である．複雑さの陰に隠れている要求項目を浮かび上がらせるために，問題領域を抽象化し，モデル化するのが有効である．問題領域を抽象化していくと，その問題領域を構成している実体やその実体の振る舞いやなど

が見えてくる．それらの構成要素をモデルとして表現し，それらのモデルの中から要求を導出しようとするのが領域モデル型手法である．

問題領域のモデルには，単なる要求項目だけでなく，問題領域内のさまざまな情報が包含されているため，全体的な視点から要求を捉えることができる．すなわち，モデルは問題領域の俯瞰図であり，問題領域の概念的な理解や，その構造的な変更を行うのに適している．その一方で，冗長な情報は捨象されており，問題の本質を捉えやすい．モデルの構成要素は，問題領域内の実体や概念であったり，機能や振る舞いであったりする．モデル化によって，問題領域の基本構造が見えてくるが，モデルから個別の要求を発見するのが難しい場合がある．

システム開発において問題領域をモデル化することの必要性は，構造化手法の時代から唱えられていた．JSD 法[19]の実体構造図は，問題領域内の実体と，その実体の動作または事象によって構成される．その後，オブジェクト指向の登場とともに，システム開発のために作成されるモデルは，一般に，静的モデルと動的モデルに分けられるようになった．静的モデルは，問題領域の実体や概念を抽象化したものであり，オブジェクト指向のクラスは，その属性と操作及び他のクラスとの関係によって定義される[20]．

一方，人間の脳の知覚中枢は連続処理装置であり，流れに沿ってよどみなく読み進めるように記述された方針は分かりやすい[21]．すなわち，対象の連続的な変化を知覚し，理解することに優れている．こうした人の持っている特長を生かし，問題領域内の変化に注目し，そこから要求を抽出しようする手法は古くから用いられてきた．構造化手法[21]では，データ処理作業を行っている組織と，それらの組織同士の間でのデータの流れを物理的データフロー・ダイアグラムとして記述し，それを詳細化する中で，データ処理機能に変換していく．一方，問題領域の変化を，変化を引き起こす主体の操作として捉えた構造化手法に対し，対象の変化を直接捉えようとする手法に状態遷移図[22]がある．ここでは，変化は対象の状態の変化として捉えられる．状態は対象の属性の値の集合によって定義できるので，多くのオブジェクト指向方法論[9, 20]では，状態遷移法によって対象の動的側面をモデル化しようとしている．

このようにモデルには静的モデルと動的モデルがあり，さらに静的モデルには，状態遷移モデルや操作モデルなどがある．それぞれのモデルは，その表記方法が決められている．領域モデル型手法の操作には，次のようなものがある．

- 実体構造モデル：クラス図などにより，問題領域の中の実体を表現する静的モデル
- 状態遷移モデル：対象の変化を状態遷移図として表現する動的モデル
- 制御構造モデル：フローチャートなどにより，処理操作の制御構造を描く動的モデル
- 因果律モデル　：データフロー図などにより，入出力データによって処理操作の因果関係を描く動的モデル

## 3.　合成型手法工学

このように，高度な手法は，その中に，幾つかの基本的な技法や操作をもち，それらを固定的に組み合わせて，想定する課題に対応しようとしている．したがって，要求分析のような多様で複雑な作業を支援するには，完成された手法よりも，提示された課題ごとに，適切な技法を選択し，適用する方が有効である場合が多い．しかし，単独の技法を選択的に使用するには，それなりのガイドラインが必要である．われわれは，むしろ，上記 3 種類の手法ごとに，そこに含まれる幾つかの技法を統合して，汎用的な技法を作り上げる方が有効であると考えた．

合成型手法工学では，上で述べた要求分析手法のグループごとの基本的な操作を，プロジェクトの状況に合わせて，柔軟に組み合わせて使用する．したがって，新たに組み合わされて作られた手法は，もとのグループの目的や特徴をそのまま引き継いでいる．このことは，新たな手法の合成を容易にしている．

それぞれのグループの中で，どの操作を組み合わせればよいかが，問題となる．一般に，組み合わせる操作の数は二つまでとするのが良い．組み合わせる操作の数が多ければ，その手法はそれだけ汎用性を高めるが，われわれの経験では，三つ以上の組み合わせは，かえって，その手法の使い難さを増徴させるだけになる．手法の汎用性を高めるには，組み合わせる操作の特徴に注目するのが良い．すなわち，目的の重複部分が少ない操作を組み合わせれば，それだけ，新しい手法がカバーする範囲が広がることになるからである．

以下に，われわれが提案する要求マトリックスと，目的手段ネットワークについて述べる．要求マトリックスは要求抽出型手法の一つである問題領域分割のための合成型手法であり，目的手段分解ネットワークは目標指向型手法のための合成型手法である．領域モデル型については，実体構造と制御構造の組み合わせや，制御構造と因果律の組み合わせなどが有効である．しかし，既にさまざまな合成型手法が提案されているので，本稿では取り上げない．たとえば，実体構造と制御構造の組み合わせに関しては，UML のコラボレーション図とシーケンス図，またアークテクチャ[23]は異なるが JSD 法[20]の実体構造図などでも提案されているし，制御構造と因果律の組み合わせに関しても UML のアクティビティ図として既に提案されている．

### 3.1 要求マトリックス

領域抽出型技法のなかの領域分割法は，複雑な問題領域の中から個別の要求を探してくるための重要な技法であるが，これまでの手法は，問題領域の単一の側面にのみ注目していた．しかし，問題領域が複雑であれば，それを複数の側面から同時に分割する方がより効果的である．われわれは問題領域を二つの直交する軸を使って分析することにした．これを要求マトリックスと呼ぶ．軸の選択に当たっては，経験上，効果的と思われる軸を優先させること，採用する二つの軸は，軸の直交の効果を上げるために，できるだけ離れた軸を採用することにした．

要求元分割は，ユースケース法や i＊法などで採用され，要求項目への連想が容易であることが認められている．この要求元分割に直交するもう一つの軸として作業工程分割と要求範疇分割を組み合わせる．こうして，二つのマトリックスが出来上がる．ひとつは，作業工程分割軸と要求元分割軸を組み合わせたビジネス要求マトリックスで，もうひとつは，範疇分割と要求元分割を組み合わせたシステム要求マトリックスである．

ビジネス要求マトリックスは，ビジネス機能に焦点を当てたマトリックスである．表 1 は，製品の製造販売を行う会社の簡単なビジネス要求マトリックスであり，その横軸には，ビジネス活動の工程が配置されている．受注と発注という工程を，一つの工程と見るか，別個の工程と見るかとか，現在は別の工程である意匠設計と製造設計を将来は統合すると言うように，ビジネス工程をどのように分割するかは，個々の企業ごとに異なっているので明確な基準を設定することはできない．ビジネス分析などでよく用いられている価値連鎖[24]などを用いるのも有効な手段である．一方，縦軸の要求元は，ビジネスにおける責任構造によって分割するのがよい．ここでは，ビジネス活動における計画，管理，実行という責任機能を分割の基準として用いている．実行系は，通常，幾つかの異なった役割に分割できるので，役割によってさらなる

分割を行ってもよい．ビジネス要求マトリックスによって抽出された要求を，人間系とシステム系に分けることによって，システム境界が明らかになり，システム要求が選別される．表1に示した例には，基本的な機能だけが載せられているが，問題領域の専門家から話を聞きながら，それぞれの機能を詳細化し，リストアップしていく．マトリックスが大きくなり過ぎたときは，それぞれの枠を独立させて運用する．

**表1　ビジネス要求マトリックスの例**

| | 企画 | 設計 | 製造 | 販売 |
|---|---|---|---|---|
| 計画 | 事業計画<br>新製品企画<br>全体生産計画 | 設計計画 | 個別生産計画<br>部品調達計画 | 市場戦略<br>販売計画 |
| 管理 | 市場調査 | 設計評価<br>原価計算 | 調達管理<br>工程管理 | 販売管理<br>顧客管理 |
| 実行 | クレーム処理 | 意匠設計 | 製造設計<br>部品展開<br>製品組み立て | 市場開拓<br>個別販売 |

一方，システム要求マトリックスは，これから作成しようとするソフトウェアシステムに対する要求に焦点を当てたマトリックスである．表2は，シンポジウムの事務局の業務を支援するシステムに関するマトリックスの一部で，横軸には，機能要求，効率要求，品質要求といった要求カテゴリーを配置し，縦軸にはビジネス・ニーズ，ユーザ・ニーズ，システム・ニーズという具合に，要求の出所を配置した．こうして問題領域は九つの小領域に分割され，それぞれの領域ごとにシステムに対する要求が抽出される．システム要求マトリックスの各軸の分割法は，プロジェクトの状況に応じて，この基本的な構造を拡張して使用することができる．例えば，製品開発では，FURPS モデルのような五つの要求カテゴリーを採用するのもよい．また，機能要求とビジネス・ニーズの領域だけを取り上げ，ビジネス・ニーズに，実際のニーズの所有者を配置すれば，それはユースケース図とほぼ同等の意味を持つことになる．

### 3.2　目的手段ネットワーク

目標分割型手法には，目的分割，手段分割，文脈分割の三つの操作があり，単独の目的分解や手段分解は，上述したように要求分析でも良く使われている．しかし，目的分解と手段分解を一つのグラフで表現すると，木構造グラフは網構造グラフになるが，本来の目的である，上位レベルの要求や実現手段の発見以外に，代替案の導出や，非機能要求の発見など，幾つかの新しい利用法が生まれてくる．

目的手段ネットワークの中で，同一の上位目標を持った複数の実現手段のうち，OR 関係にあるものが代替手段となる．図1は，表2と同じ，シンポジウム支援システムの要求の中から，セキュリティに関する目的手段ネットワークの一部を抜きだしてきたもので，半円で繋いだ実現手段同士が代替案の関係にある．投稿論文や著者情報を保護するための手段として「個人認証」によって該当ファイルへのアクセスを制限することができる．しかし，「個人認証」を行うには，それなりの準備が必要であり，早急な対応が難しい．その代替として，個人認証システムが導入されるまでは，取りあえず，簡易な「パスワード」によるアクセス管理によって対応することになる．このように，代替案は，ある要求に対し，まず，その上位目標を求め，次

**表 2　システム要求マトリックスの例**

| | 機能要求 | 性能要求 | 品質要求 |
|---|---|---|---|
| ビジネス・ニーズ | ・募集要項の送付<br>・投稿論文の収集・保管<br>・査読委員の依頼<br>・査読委員への送付<br>・査読結果の入手<br>・応募者への合否通知<br>・応募論文の検索<br>： | ・応募論文総数：300件 | ・応募者の機密保護<br>・応募論文の機密保護<br>・査読結果の機密保護<br>・障害時の代替ルート |
| ユーザ・ニーズ | ・論文の投稿<br>・投稿論文の入手<br>・応募要綱の入手<br>・査読結果の送付<br>： | ・操作時間：30秒以内 | ・簡単な操作 |
| システム・ニーズ | ・登録論文の更新<br>： | ・障害復旧：半日以内 | ・論文ファイルの二重化<br>・論文のファイル形式統一 |

に，その上位目標の実現手段を展開し，元の要求と OR 関係にある要求を選択することによって導出できる．その代替案の妥当性の検証は，別途，行わなければならない．

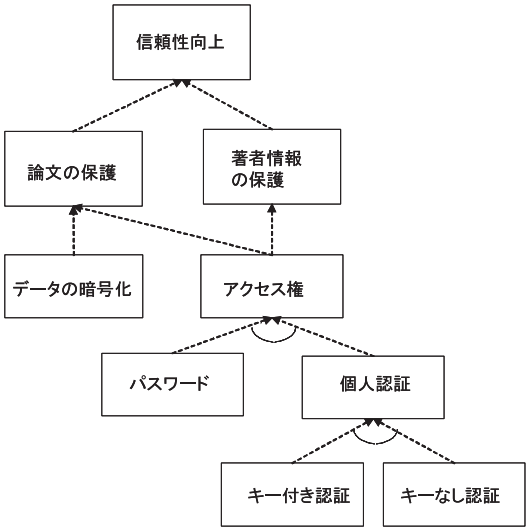


**図 1　代替案を示す目的手段ネットワークの例**

代替案の検討だけではなく，手段分解したサブゴールに，目的分解で得られた幾つかのスーパーゴールを連携させることによって，新たな付加価値を持ったゴールを得ることができる．図 2 は，同じシステムの目的手段ネットワークである．「シンポジウム支援システム」の開発目的の一つである「ノウハウ喪失の防止」という上位目標と，「計画の立案・実行」という要求から「出状レターのテンプレート化」と「年間計画表のテンプレート化」という実現案が導出されている．

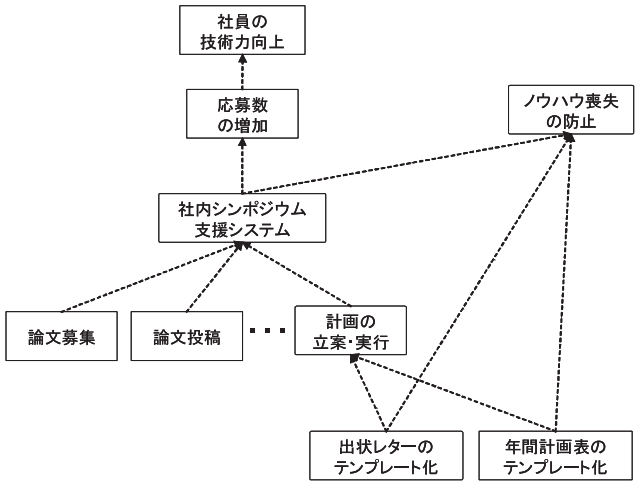


**図 2　非機能要求を伴う目的手段ネットワークの例**

## 4.　議論とまとめ

人間が考えうる限りの基本的な手法は，既に考えつくされているとも言われている．こうした考えに立てば，残された手は，既存の手法の分解か組み合わせ以外ないことになる．既存の手法を，それを構成している操作に分解し，プロジェクトの状況にあわせて，自由に変形できるようにしようとしたり[25,26]，複数の異なった手法を組み合わせて使おうという試みも行われている[27]．われわれが提案する合成型手法工学は，そうした新たな試みの一つである．

本稿では，要求抽出型手法の中の領域分割技法と，目標指向型手法について，二つの合成型手法を提案したが，当然，その他の技法の組み合わせも考えられる．さらに，要求分析プロセスで使用される技法には，個々であげた情報分析のための手法の他にも，類似のアプリケーションの事例を挙げて，要求者の想像力を喚起する誘導型の手法や，ブレーンストーミング法のような人間の振る舞いに関する手法もある．複雑な要求分析プロジェクトでは，こうした異なった型の手法の組み合わせも重要であるが，それらの組み合わせの有効性の検証は，われわれの今後の課題である．

しかし，どのような便利な手法や技法が提案されても，現場の分析者に，それを使いこなすだけの技術力が求められることに変わりはない．

## 5.　謝　　辞

状況遷移マップから本稿に至る研究は，情報処理学会のソフトウェア工学研究会（SIGSE）の要求工学 WG における活動の一環であり，研究の場を与えていただいた該 WG のメンバー各位に謝意を表したい．

---

**参考文献**　［1］ Loucopoulos P. and Karakostas V. *System Requirements Engineering*, McGraw-Hill, 1995.
［2］ Christel M.G. and Kang K.C. “ Issues in Requirements Elicitation ”, Technical Report CMU/SEI-92-TR-12, 1992.
［3］Tsumaki T.“ Requirements Engineering Pattern Stucture ”, *Proc. of the 11 th Asia Pacific Software Engineering Conference*, 2004. pp 502-509.

［4］ 玉井哲夫ソフトウェア工学の基礎，岩波書店，2004.
［5］ IEEE, Recommended Practice for Software Requirements Specifications, Std 830-1998, 1998.
［6］ Booch, G. *Object-Oriented Analysis and Design with Application*, 2 nd edition, The Benjamin/Cummings Publishing. 1994.
［7］ Jacson, M. *Problem Frame*, Addson-Wesley, 2001.
［8］ Eriksson, H.E. and Penker, M. *Business Modeling with UML*, John Wiley & sons, Inc., 2000.
［9］ Jacobson I., Christerson M., Jonsson P. and Overgaard G. *Object-Oriented Software Engineering*, Addison Wesley, 1992.
［10］ Grady R. *Practical Software Metrics for Project Management and Process Improvement*. Prentice Hall, 1992
［11］ ISO/IEC 9126-Information Technology, Software Product Evaluation, Quality, Characteristics and Guidelines for their Use.
［12］ Carroll J.M. *Making Use : Scenario-Based Design of Human-Computer Interction*, MIT, 2000.
［13］ Yu, E.“ Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering ”, *3 rd IEEE International Symposium on Requirements Engineering* (*RE ’97*), 1997.
［14］ Anton A.I.“ Goal-Based Requirements Analysis ”, *Proc. 2 nd International Conference on Requirements Engineering*, pp.136-144, IEEE, 1996.
［15］ Kaiya H., Horai H. and Saeki M.“ AGORA : Attributed Goal-Oriented Requirements Analysis Method, ” *Proc. of Requirements Engineering* (*RE ’02*), IEEE, 2002.
［16］ Dardenne A., Fickas S. and van Lamsweerde A.“ Goal-Directed Concept Acquisition in Requirements Elicitation ”, *Proc. 6 th International Workshop on Software Specification and Design*. IEEE Computer Society Press, 1991.
［17］ Fillmore C. The case for case, in E. Bach and R. T. Harms (eds), *Universals in Linguistic Theory*, Holt, Rinehart and Winston, London, pp. 0-88.
［18］ Sowa J. F. and Zachman J. A.“ Extending and formalizing the framework for information systems architecture ”, *IBM System Journal*. Vol 31, No 3, 1992.
［19］ Jacson, M.A. *System Development*, Prentice Hall, 1983.
［20］ Rumbaugh J. et al. *Object-oriented modeling and design*, Prentice-Hall, 1991.
［21］ Demarco T. *Structured Analysis and System Specification*, Youdon Inc., 1979.
［22］ Harel D.“ Statecharts : a Visual Formalism for Complex Systems ”, *Science of Computer Programming* 8, 1987.
［23］ Show M. and Garlan D. *Software Architecture*, Pretice Hall, 1996.
［24］ Porter M.E. *The Competiyive Advantage of Nations*, Free Press, 1990.
［25］ Brinkkemper, S., Saeki, M., Harmsen, F.“ Meta-Modelling Based Assembly Techniques for Situational Method Engineering ”, *Information Systems*, Vol 24, No.3, 1999.
［26］ Jolita Ralyte, Rebecca Deneckere, Colette Rolland“ Towards a Generic Model for Situational Method Engineering ”, *CAiSE 2003* : 95-110.
［27］ http：//www.sei.cmu.edu/str/descriptions/deda.html

**執筆者紹介**　妻　木　俊　彦（Toshihiko Tsumaki）

1970 年日本ユニシス(株)入社．以後，リアルタイムシステム，人工知能システム，オブジェクト指向モデリングなどの開発・研究に携わる．情報処理学会ソフトウェア工学研究会運営委員，同要求工学 WG メンバー，情報処理学会標準化委員会 SC 7/WG 7 委員，日本情報処理開発協会情報処理技術者試験委員，芝浦工業大学情報システム学部非常勤講師．

著書：「オブジェクト指向モデリング」,日刊工業新聞社，1999（共著），「ソフトウェアの要求獲得法と仕様書の書き方」，トリケップス，2003（共著）．