ソフトウェア工学 135－12
(２００１. １１. ２２)

プロセス径路合成場の概念による開発プロセスのモデル化

大木 幹雄†

†日本工業大学　工学部情報工学科

E-mail: ohki@nit.ac.jp

あらまし
ソフトウェアの開発プロセスは本質的に不確定性を伴う．本稿ではこのような不確定性をプロセスモデルに組み込む基本的な方法として，場の量子論的に類比を求め議論を展開する．対象とする開発プロセスは，概念モデリング工程に限定する．その上で，モデリングで用いる判断基準を場の量子論的な観点から定式化した結果を示す．次いでデザインパターンに対しても同様な発想でパターン導出の過程をモデル化できる可能性を示す．さらに分析過程の時間経過に伴う変化をモデル化するために，場に対して種々の力を導入する必要性について簡単に述べ，それらにもとづいて分析過程管理モデルを構築する指針を考察する．

キーワード
分析判断基準，量子力学，デザインパターン，CK メトリックス

A modeling of the design analysis based on the process-path synthesizing concept

Mikio Ohki

Nippon Institute of Technology

Abstract
Every software development processes essentially have uncertainties. This Paper describes the way of how to introduce the uncertainties to an process model on the analogy of Quatum field theory. For the purpose of modeling the conceptual modeling process, I introduced few kind of forces for explaining the dynamics of analysis process. Furthermore, I applied Quantum Theory approach to the modeling of management of analysis process. The result of consideration about these management modeling are also described in this paper.

Keyword
Analysis Criteria, Quantum Mechanics, Design Pattern, CK Metrics

1. はじめに

ソフトウェア開発プロセスの定式化に関する議論の多くは，類似するシステムを一定の品質・工期で開発することを目的に，開発ソフトウェアの特性に応じた作業手順のあり方や管理・統制のあり方，あるいは保存すべき成果物等に焦点を当てたものになっている．これらの議論の共通基盤として，開発プロセスは標準化できるもの，あるいは標準化することが望ましいとの基本理念が横たわっているように思える．しかしながら，開発プロセスは本当に標準化できるものであろうか．さらに言えば標準化は現実に効果を上げ続けるのであろうか．開発するシステムに対する要求仕様は，刻々変化するし，実装上の技術も開発の都度変化する．標準作業工程なるものが策定されたとしても，開発要員に浸透するころには陳腐化し始めている．さらに詳細な部分は実施する作業毎に変更されることも多い．開発能力の成熟度が高いプロジェクトでは，詳細な開発プロセスや決定事項はデータベースに記録保存される．しかし，現実にはプロジェクトに固有の雑多な制約条件から，それらが実際に再利用されることは少ない．

このような状況に柔軟に対応するための一方策として，開発プロセスをパターン化し，その組合せにより開発プロセスを動的に生成して，適用性の向上を確保する方法が考えられる．しかし，デザインパターン以上に，開発プロセスはプロジェクト固有の制約条件や社会環境に影響される部分が多いことから，現象と解決策といった経験重視の利用にとどまる可能性が高い．開発プロセスを科学的に管理するには，現象面の背後にある人間作業の原理原則を計数化が可能なモデルによって表現し，予測できなければならない．

このような素朴な願望に端を発して行った幾つかの考察を本稿では展開する．具体的には，開発プロセスのうち，特に概念分析設計工程に限定して，分析設計プロセスで現れる不確定性をどのように捉えるべきかについて考察したのち，分析設計プロセスのモデル化について提案を行いたい．なお，本稿ではいくつかの仮説を導入しているが，これらの計数的な検証とはなお継続中であることをお断りしておきたい．

2. 開発プロセスの不確定性に対する考え方

ソフトウェア開発の大部分は，頭脳作業によって占められている．すなわち，要求分析からソフトウェア分析設計，プログラミングまで，ソフトウェア開発の大部分は直接見ることができない頭脳活動によって占められている．頭脳活動は，多かれ少なかれコンピュータアルゴリズムとは異なる“決定論的ではあるが計算不可能な不確定要素”が多く含まれる．標準化が目指すアプローチは，このような不確定要素を詳細に分類して一定の枠内に押し込めるか，意図的に無視するアプローチと言えよう．しかしながら，不確定要素は，頭脳活動の本質であると見る考え方もあり，それがアルゴリズムに従うコンピュータと人間の意識活動の差であるとしている（代表的な例としてペンローズの脳量子理論[1]などがある）．この考え方を拡張解釈すると，頭脳労働を主とするソフトウェア開発プロセスには，計算不可能な要素が本質的に含まれることになり，開発プロセスがプログラミング可能とする“プロセスプログラミング”の発想とは相容れないことになる．

一方，筆者はかねてから，概念モデリングの判断基準に関して，概念モデリングの作業は，作成すべきドキュメントや手順で規定するよりも「分析の視点とリンクした判断基準の連鎖で規定すべきである」と考え，いくつかの判断基準とその適用手順を示してきた[2]．しかし，判断基準について検討を深めれば深めるほど，判断基準を導き出す過程にも何らかの判断基準（メタ判断基準）が必要であることを実感するにいたった．このようなメタ判断基準の対象となるものは，もはやモデリングの構成要素ではなく，事象を整理する人間の思考過程そのものになる．そこでは人間の思考作業がもつ本質的な不確定性をいかにモデル化して扱うかの問題に直面せざるをえなくなる．

このような不確定性を本質的に伴うような対象物を扱った理論として，（多少，唐突ではあるが）場の量子論[3]がある．場の量子論的は，電磁場の構成要素である光子の状態が確率的にしか捉えられないこと，光子にいくつかの特性を持ち込むと数々の実験現象が説明可能なこと，および電磁場が光子の生成消滅を担う媒体であることなどのいくつかの特徴をもつ．これらの特徴の中で場から光子が生成される点，および場の状態が確率的にしか予測できない点等は，人間の

思考状態のモデル化に適した側面をもっている．そこで，場の量子論的な発想を概念モデリング過程に対比させてみる．するとモデリング要素（例えば，属性やメソッド，イベント等）の抽出とは，人間の思考状態に対応した場を想定してそこから量子化によって光子を生成することに対応させうることがわかる．構成要素を抽出する判断基準は，光子を量子化する条件に対応することになる．デザインパターン[4]は，場に対するいろいろな制約条件によって，場がいくつかの安定固有状態に陥ったとき観測される現象として解釈できる．判断基準を導くメタ判断基準とは，場の時間的な振舞いを規定する方程式を定めることにほかならないことになる．

以上のように思考過程のもつ不確実性を扱うためのモデルとして場の量子論的な発想は幾つかの有望な側面をもつ．以下では，このような発想をもとにして，分析設計過程に範囲を絞り，対応付けを試みる．

3．構成場の基本的な特徴と分析プロセスとの対比

3．1　構成場とは

場の量子論的な発想で分析設計過程をモデル化する出発点として，まず分析設計開始時の内容が確定していない初期状態を，場の構成要素の不確定な状態の重ね合わせ状態に対応付ける（以後このように量子論的な場の状態に対応付けられたものを「構成場Ψ」と呼ぶことにする）．すると分析設計過程とは“あいまいな初期状態”から出発して，時間経過にしたがって，あいまい性が拡散してゆく過程に対応つけられる．拡散状態が無限に続くのを避けるため，分析設計過程の途中にマイルストーンを設けてレビューを繰返し，機能や属性，制約条件等のあいまい性を取り除く努力が行われる．これは量子論的に考えると，いろいろ構成要素の重ね合わせ状態を観測し確定させる，いわゆる“観測によって確率波の波束を収縮させること”に対応するになる．観測によって状態の“その観測時点・個所”の特性が確定しても，次の時点から構成場Ψの拡散がはじまる．分析設計過程も同様に，レビュー後に発生するあいまい性を押さえ込むために，レビューが繰り返される．一見当たり前のような行われているレビューを量子論的に解釈すると以上のようになる．

3．2　量子化演算としての分析の判断基準

場の量子論的な発想で分析設計プロセスを捉えることの最大の利点は，構成場Ψから分析設計の構成要素を抽出する操作を数学的にモデル化することが可能な点にある．例えば，クラス分析を行い，クラスがもつ属性やメソッドを抽出するときの判断基準は，次のような構成場Ψを想定することによって定式化ができる．ただし，構成場Ψはイベントの種類に対応する集合 ε とその発生タイミングに対応する離散的な位相時間集合 τ からなる抽象部分空間と実時間 t の関数で記述されると仮定する．ε と τ は独立しており，τ は実時間 t とは異なる“分析システムが稼働するときの時間”を表している．このような構成場から属性やメソッド，およびクラスを抽出する操作に対応する量子化演算は以下のように定義される．

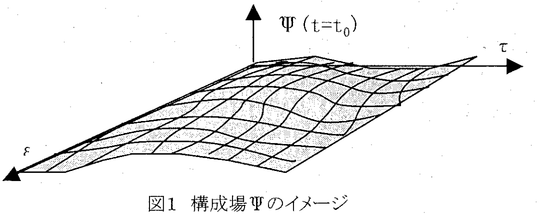

図1　構成場Ψのイメージ

(1) 属性の抽出　$::= \partial \Psi(\tau, \varepsilon, t) / \partial \tau = 0$
によって構成場Ψを量子化

属性とは，分析設計で明らかにする構成要素の不確定な状態に対応した構成場Ψ（図1を参照）の位相時間偏微分の成分が 0，すなわち，位相時間の変化に対して安定しているΨの点に対応するものと考える．属性そのものは，イベント発生のタイミングに対応する位相時間が変化しても変化しない（ただし，属性値は変化する）．そこで，構成場の状態が位相時間の変化に対しても変化しない点を属性と考える（属性はイベントの種類に依存して分析過程で追加削除されるが，ここではこのような点は考慮に入れない）．

属性は，分析プロセスで扱う“システム動作上の時間（すなわち，イベントの発生タイミングの連鎖）”内で変化することが少なく安定的であることから，このような定式化は妥当なものと考える．なお，$\partial \Psi / \partial \tau = 0$ の点には，通常“属性名”とゆう名称が付けられ，実時間の経過と共に位相空間内を移動してゆく．

(2) メソッド::＝　$\partial \Psi (\tau, \varepsilon, t) / \partial \varepsilon = 0$ によって構成場Ψを量子化

メソッドとは，構成場Ψのイベント種類集合εの偏微分成分が0，すなわち，イベント種類集合の変化に対して安定しているΨの点に対応するものと考える．一般にメソッドそのものは，イベントの種類が変化しても変化しない．ただし，イベントの発生によるメッセージ送信を介して起動する個々のメソッド（すなわち，メソッドの実現値）はイベント種類の変化に応じて変化する．分析において抽出の対象になるメソッドは，イベントの種類が増減しても変化はせず，むしろ，イベントの発生によって送信されてくるメッセージの種類に依存する．この点からして，εとτで張られる構成場Ψは，メッセージ交換の媒体に対応するものと考えてよい．なお，構成場は分析過程の進捗に合わせて変化することから，以上の量子化は実時刻t_0に時間を限定したものであることに留意しておく必要がある．

(3) 概念クラス::＝　$\{\ \alpha_i \mid \varepsilon = \varepsilon_i\ \&\ \partial \Psi (\tau, \varepsilon, t) / \partial \tau = 0\ \}$ すなわち，イベント種類集合εに対応しイベント発生タイミングτで量子化された属性の集合．

属性のみでメソッドをもたない概念クラス原型は，同一のイベントの発生タイミングに量子化によって抽出された属性をまとめたものである．なぜなら，これらの属性集合は，同一の発生タイミングで初期化されることを意味し，したがって，同じ概念クラスに属すべき属性であることを意味しているからである．

3. 3 量子化された構成子の特徴

前節に引き続き，量子化によって抽出された個々の属性やメソッドはいろいろな特性をもつものとする．以下では，抽出された属性やメソッドを“構成子”と呼ぶことにする．個々の構成子には，属性名やメソッド名のように名称が付けられる．構成子にもたせるべき特性として次を考える．

① ν：状況レベル

構成子同士を比較するとき便宜的に導入したレベルである．構成子は図2に示すように抽出されたとき，割当てられた位置（＝割当て番号）毎に状況レベルの低い順から埋まってゆく．

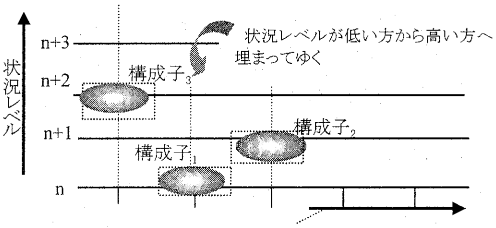

図2 構成子のもつ状況レベル

② π：多重度

図3で示しように構成子集合内の同じ位置（＝割当て番号）で構成子が存在しうる個数を示し，1かそれ以外の正整数値をとる．同じ状況レベルでは，属性は1以外の多重度をとるものは存在できない．メソッドは，同じ状況レベルに1以上の多重度をとることができる．属性とメソッドによって異なる継承規則（属性は下位クラスでは再定義できないが，メソッドはできる）を表すのに用いる．

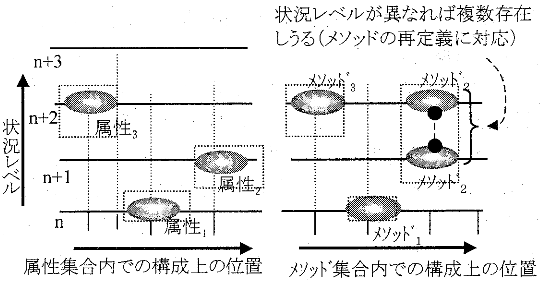

図3 構成子のもつ多重度

③ λ：存在寿命

属性あるいはメソッドが，構成場の中で存在する時間を示すものである．属性をもつ関連や多対多関連で用いる．

3. 3 量子化と構成子の特性によるデザインパターン導出のモデル化

以上に述べた構成場の量子化と構成子の特性を用いると，概念クラスの3つの判断基準がこれらによって再解釈できる．同時に構造に関するデザインパターンの導出過程の一部がモデル化できる[5]．例えば，Adapterパターンは，2つの概念クラスに対応する状態ϕ_1，ϕ_2の合成状態の相互関連$|\phi_1 + \phi_2|^2$を展開

したとき現れる相互作用項ϕ_1・ϕ_2に対応付けることができる．また，構成子がもつ多重度の制約から，Bridgeパターンに対応する状態を導出する過程が図4に示すように再現できる．同様にしてCompositeパターンやDecoratorパターンの導出が可能であるが，ここでは割愛する．

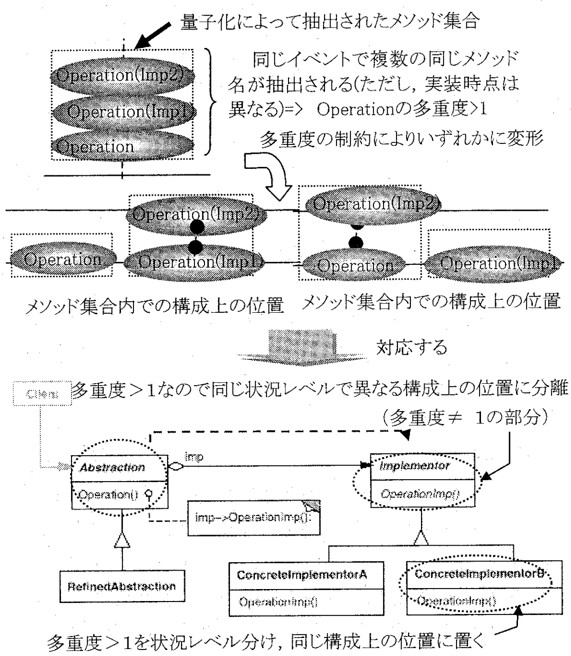

図4 構成場と構成子の特性にもとづくBridgeパターン導出

4．構成場のダイナミクスと分析過程との対比

前項までは分析過程を通して流れる実時間の一時点に時間を固定した構成場について述べてきたが，構成場は分析過程の進捗に対応して，変化して行く．このような変化を表す拡散方程式を直ちに導き出すことは難しい．構成場から量子化によって抽出された構成子間にいろいろな力を想定しながら，場の変化の履歴に対応するクラスライブラリ等と比較して仮説が一致するか確認しながら行う必要がある．

ここでは，幾つかの力の概念を導入し，それらの相互作用によって，構成場のダイナミクスが生じるものとして検討を進める．以下では前述のクラス分析を再度事例とし取り上げ，その力の分類と定式化について述べる．本稿の主題である開発プロセスのモデル化は，構成場の時間的拡散を最小限に抑える方法として，後に議論する．

4.1 クラス内の属性とメソッドの結合力

構成子間で働くいろいろな力を定式化するために，量子化した属性集合を成分としてもつベクトル$|\alpha\rangle$，およびメソッド集合を成分としてもベクトルを$|\mu\rangle$とする．このとき，クラスのi番目の属性$\alpha(i)$がj番目のメソッド$\mu(j)$で用いられるかの関係を行列形式で示したものを次式のように定義する．Eは，実際にクラス内での属性とメソッドの関係を観測する操作を意味する．

$$\phi^{\alpha}_{\mu}(i,\ j)=E(\alpha(i)\mu(j)) \quad \cdots\cdots \text{(式1)}$$

属性数をm，メソッド数をnとすると，$\phi^{\alpha}_{\mu}(i,\ j)$はi番目の属性がj番目のメソッドで用いられていたときに1，用いていないときに0を要素値してとるm×nの行列要素を表している．これを属性集合ベクトル$|\ \alpha\rangle$，メソッド集合ベクトル$|\ \mu\rangle$を用いて表すと式2のようになる．

$$\phi^{\mu}_{\alpha} = |\ \mu\rangle\ E\ \langle\alpha\ | \quad \cdots\cdots \text{(式2)}$$

この行列のノルムを式3のように定義する．

$$\|\ \phi^{\alpha}_{\mu}\ \| = \phi^{\alpha}_{\mu}(i,\ j)^{T} \times \phi^{\alpha}_{\mu}(i,\ j) \quad \cdots\cdots \text{(式3)}$$

ノルム$\|\phi^{\alpha}_{\mu}\|$はn×nの行列で，要素 i,j は，Chidamber,Kemererの定義したメソッドm_iとm_jのCohesion係数[6]に対応している．また，クラス内の結合力fを，ノルムを用いて式4のように定義する．

$$f = \mathrm{tr}\ \|\ \phi^{\alpha}_{\mu}\ \|\ /\ 2 \quad \cdots\cdots \text{(式4)}$$

ここで，式3のノルムが実際にCohesion係数に対応していること具体例を用いて示そう．

以下のような$|\ \alpha\rangle$，$|\mu\rangle$を考える．

$$|\ \alpha\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \qquad |\mu\rangle = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix}$$

$$\Phi^{\alpha}_{\mu} = |\alpha\rangle\ E\ \langle\mu\ | = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}\ E\ \begin{pmatrix} m_1 & m_2 & m_3 \end{pmatrix}$$

$$= \begin{pmatrix} E(a_1\ m_1) & E(a_1\ m_2) & E(a_1\ m_3) \\ E(a_2\ m_1) & E(a_2\ m_2) & E(a_2\ m_3) \end{pmatrix}$$

$E(a_i\ m_j)$をe_{ij}と置き，次式のように表現する．

$$\Phi^{\alpha}_{\mu} = \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \end{pmatrix}$$

すると$\|\ \phi^{\alpha}_{\mu}\ \| = \Phi^{\alpha\ T}_{\mu} \times \Phi^{\alpha}_{\mu}$は，次のように表せる．