１．本研究の目的

原価計算に用いられる計算の手続きや仮定に関する記述・表現方法には，例えば原価計算基準中の言語的な描写や，一般的なテキストに記述されている計算式といった，ユーザ次第で記述された内容に関する解釈の仕方が異なることや，読みづらさを与えるといった，情報量を制限させてしまう記述・表現方法となっている．この様なことは，専門外のユーザだけに起こることではなく，原価計算の専門家でも，場合によっては様々な表現方法を用いないと，計算に用いられた仮定や手続きを理解することは出来ない．

そして，この様な曖昧な記述・表現方法で記述されている原価計算という計算の技法をコンピュータに実装すると，曖昧な部分があるが故にシステムへ実装する際に仕様が定まらず，システムを通して作られる情報に対する追試性を失い，その情報が作られる根拠を説明できなくなる．

この専門性に左右される原価計算の説明に用いられる記述・表現方法による課題を無くすために，様々な研究者が情報技術を応用し解決を図ってきた．この様な解決の過程を考察するためには，次の 2 つのロジックを考える必要がある．それは，情報技術を原価計算に適用するためのロジックと，情報技術が適用された原価計算システムをユーザが使うためのロジックである．本稿では，この 2 つの汎用的なロジックを実現する技術として，特に過去に企業に導入され運用された実績もある「構造マトリクス」に着目し，この技術が，情報技術と原価計算システムとユーザを結ぶインタフェイスの役割を果たしているのか考察する．さらに，各情報技術基盤が実現してきた原価計算に対する解釈の差異や誤解を取り除く基盤を概観し，各情報技術基盤では未だ解決できていなかった点と情報技術基盤間の繋がりを明らかにする．そして，原価計算に用いられた計算の手続きや仮定から生じる認識の差異や誤解を解消するための共通言語を開発することにより，原価計算に対する誤解や解釈の差異を排除出来る基盤を更に強固なものとすることが本研究の目的である．

2．本論文の構成

第 1 章　問題提起

第 2 章　原価計算システム上での課題

　　第 1 節　データベース基盤上の課題

　　第 2 節　計算構造基盤上の課題

　　第 3 節　ユーザインタフェイス上の課題

　　第 4 節　その他の課題

第 3 章　情報技術基盤を原価計算の視点からアプローチした研究における原価計算システム基盤の整備

　　第 1 節　事象アプローチと会計データモデルによる DB 基盤の整備

　　第 2 節　オブジェクト指向原価計算とその応用例による基盤整備

第 4 章　構造マトリクスを適用したユーザインタフェイスによるデータベース・計算構造

基盤の操作・表現
第 1 節　構造マトリクスの概要
第 2 節　構造マトリクスの歴史的背景と評価の動向
第 3 節　構造マトリクスによる DB・計算構造基盤の UI 的表現
第 4 節　構造マトリクスによる新たな拡張 可能性と構造マトリクスの限界
第 5 章　原価計算の計算機能に対する記述・操作言語的な考察
第 1 節　原価計算の計算機能に対する記述・操作言語化に関する考察
第 2 節　構造マトリクスの DSL 的評価
第 3 節　本章のまとめと新たに生じた課題
第 6 章　その他の発展可能性の追求
第 1 節　マルチエージェント概念の概略
第 2 節　本章で提示するテストシナリオ
第 3 節　マルチエージェント概念が他の領域で使われている例
第 4 節　テストシナリオを実現するためのマルチエージェント要素
第 5 節　テストモデルの構築と本章の目的との適合性
第 7 章　本稿のまとめと残された課題

3. 情報技術基盤の関連性と明らかになった課題

情報技術基盤を考察するため，原価計算システムの基盤に，DB・計算構造・UI 基盤という 3 つの情報技術基盤に着目した．DB 基盤には会計データモデル論，計算構造基盤にはオブジェクト指向による計算構造基盤，UI 基盤には構造マトリクスを代表的な技法として焦点を当てた．

これら情報技術基盤が原価計算システムに貢献した機能と研究の関連性を述べていく．会計データモデル論では多次元ベクトル的なスキーマに関する構築の方法論の提示と，構築されたスキーマが特定の処理に影響を受けてスキーマの構造が変化しないようなスキーマと処理の独立性を確保する仕組みの提示に成功した．このことによって，計算に用いたレコードと分散する DB のレコードの互換性と，計算結果と計算に用いたレコードの結びつきを確保することが実現される．しかし，処理とデータ属性の結び付けの煩雑さと，DB 基盤であるが故に処理の重複を把握できないという限界があった．

そこで DB 限界を解消する技法として，オブジェクト指向による計算構造基盤が用いられるようになった．オブジェクト指向では，オブジェクトやオブジェクトの協調関係により，処理とデータ属性の結び付けを容易にした排除の実現も果たした．また，オブジェクトの協調関係により，処理過程を入れ子構造で表現できることから，DB 基盤上の課題を克服しながら，計算構造を構築出来るという強みもある．さらに，オブジェクトを用いることで計算結果から計算に用いた処理とレコードに可逆可能となり，レコード内のデータ属性を切り替えることで様々な情報を生成できるようになり，その事が処理の重複を排除す

ることに繋がるというメリットを有する．このオブジェクト指向を原価計算に適用したものが，オブジェクト指向原価・収益計算であり，その具体的な応用方法にスナップショットコスティングや，リアルタイム原価管理があり，原価計算のシミュレーション基盤を整える応用法となりうる可能性を持つことになった．しかし，適用した情報技術をどの様な形でユーザに記述・操作・表示するのかといった UI 的課題に関しては述べられていいという課題が残った．

そこで，この 2 つの基盤を記述･表現する UI として，構造マトリクスが妥当であるという仮説を立てた．結果，構造マトリクス中にオブジェクト指向的な要素が確認出来たため，2 つの基盤の反映が限定的に可能であることが分かった．なぜ限定的であるかというと，構造マトリクスの表現の制限上，DB 基盤上からのデータのインポートと時系列データのシート上への反映といった点の実現が困難であるためである．

この様に，3 つの基盤は欠点を補い，相互に作用することで次に挙げる原価計算上に起こる認識の差異や誤解を解消する仕組みを持つことになる．

3 つの基盤の情報技術基盤を原価計算の視点からアプローチした研究に共通する仕組みとは，計算に用いたデータや，処理の仮定や手続きに，可逆可能な環境を整えることによって，元データと処理に可逆可能な結果情報を生成し，その情報が作られる根拠を確保しようとする基盤になった．

上述したように，各情報技術基盤の原価計算システムに対する貢献と情報技術基盤が関連しあうことによる原価計算上に起こる認識の差異や誤解を解消する仕組みを述べた．

しかし，先に述べた情報技術基盤の中には今までの研究で述べられていない課題があることを発見した．会計データモデル論では，データ項目の不一致が生じる可能性を排除し切れていないことや，データ属性の時系列性を DB 結合により得られる標準 VIEW への反映といった，精緻な原価計算を実現するための仕組みに欠ける事が新たな課題として出てきた．

また，DB・計算構造基盤には所与とする前提条件に不備があることがわかった．その不備とは，データモデルや計算処理過程を構築する際，ユーザを問わず，経営活動内で生じているデータ構造や情報を作成するシステムの振る舞いを理解しているという不備である．この前提条件の不備を排除する仕組みをどの様に実現するかが課題として残った．

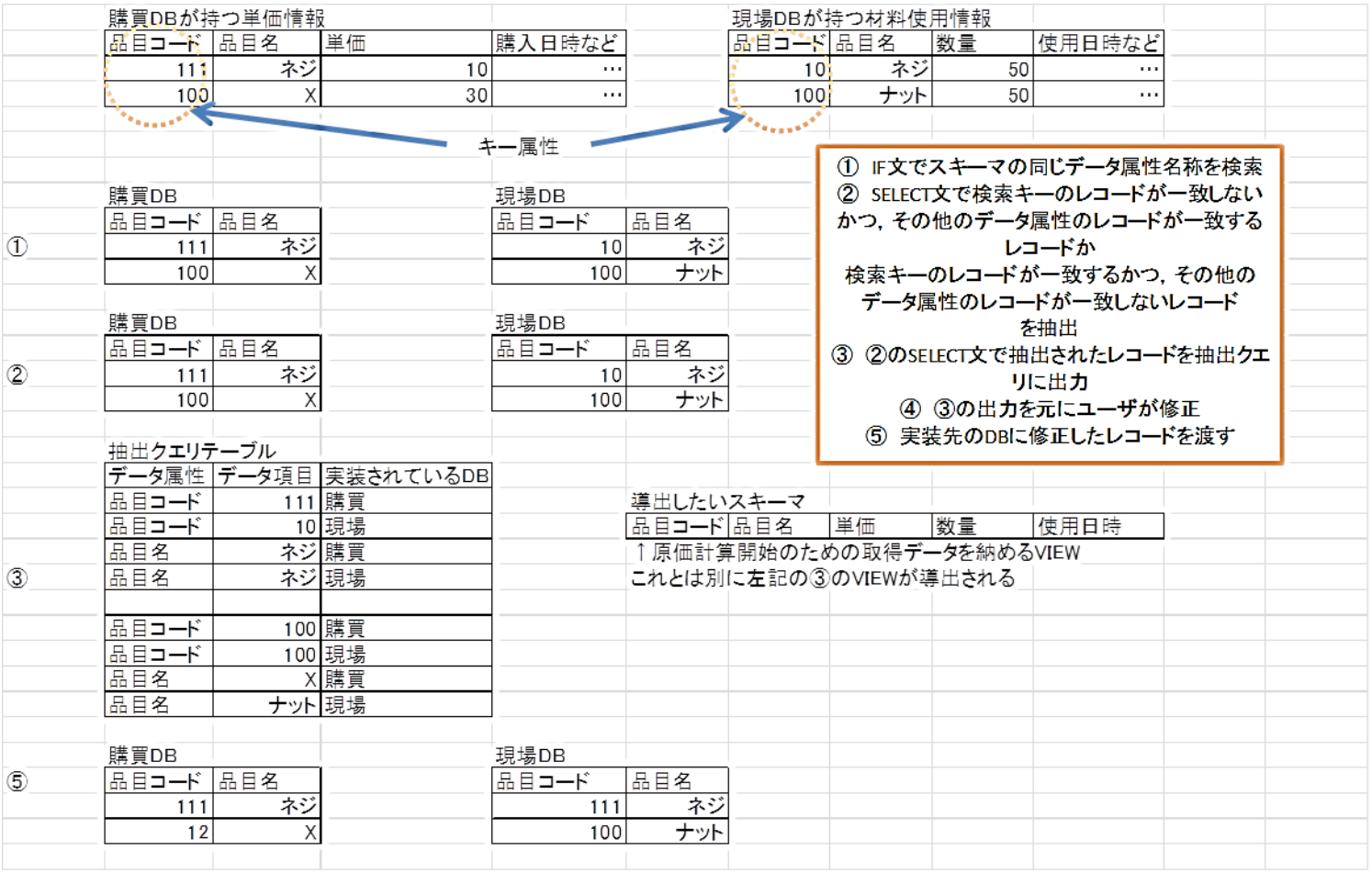
更に，取り上げた情報技術基盤には原価計算の計算手続きを全ユーザが理解できる形で記述・操作するための共通言語を準備し活用させるといった視点が欠けており，本研究の目的である，原価計算に用いられた計算の手続きや仮定に対する誤解や解釈の差異を除く仕組みの提供が実現されていない．

そこで本稿では，原価計算の計算手続きの中でも配賦計算に着目し，配賦モデルを構築し，配賦モデル内にある計算の仮定を記述出来る仕組みを考察することも新たな課題として取り上げる．

4. 本研究による新たな課題に対する対応

まず，データ項目の不一致ガ生じる可能性を排除し切れていない点や，時系列データの DB を結合して得られる標準 VIEW への反映といった，精緻な原価計算を実現するための仕組みに欠けるといった課題に対する対応を述べる．

図表 1　標準 VIEW に導入したデータ項目の標準化支援機能の具体例



図表 1 にあるように，原価計算開始のための取得データのための標準 VIEW 作成時にデータ項目が標準化されていないレコードを抽出し，それを記録する VIEW を用意することでデータ項目の標準化を促進する仕組みを開発した．

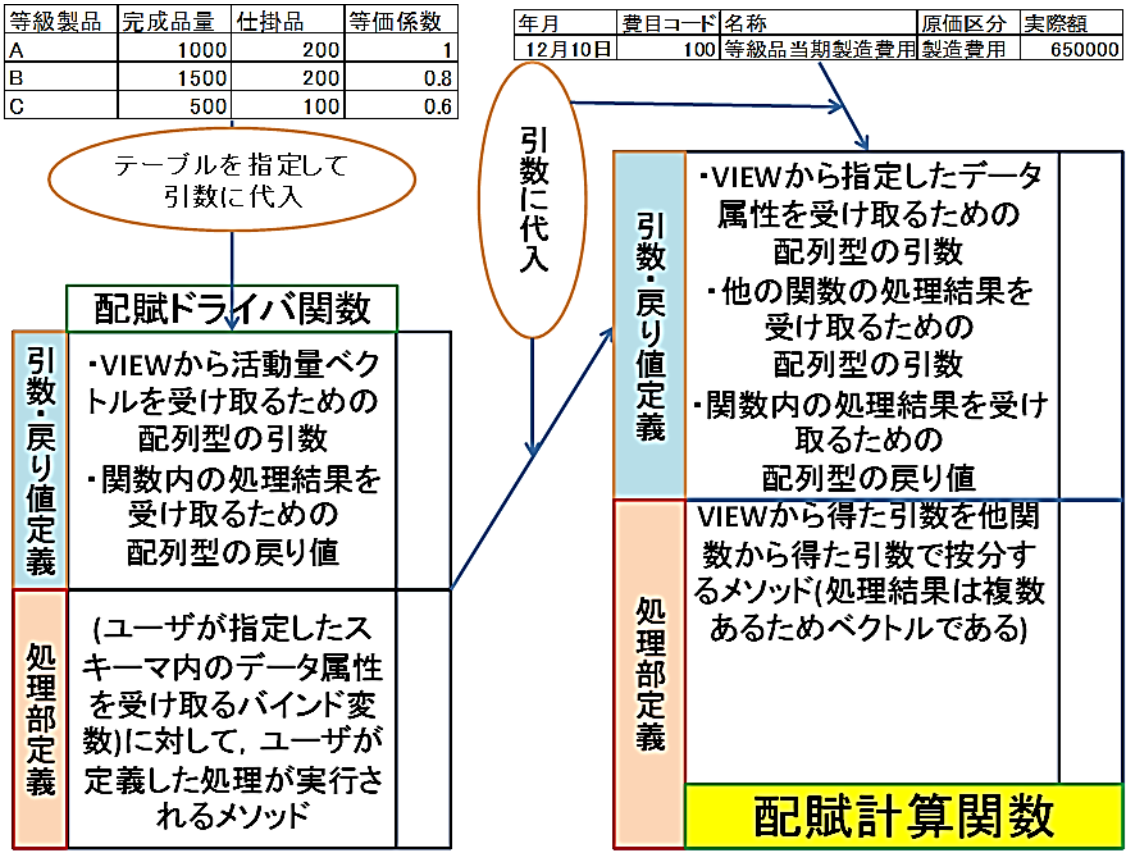
次に，時系列データのユーザに対する認識支援に関して，標準 VIEW 作成時に次の機能を提供することでその実現を図った．

図表 2　時系列データを保持した元データテーブルの検索と表示

データ属性の時系列性を計算処理へ反映することが精緻な原価計算の計算機能実現において重要な機能となるが，図表 2 にあるように原価計算開始のための取得データを得るためのデータモデリング時にデータ属性の時系列性をユーザに認識させる事によって，正しい標準 VIEW 作成を支援可能となろう．

次に，本研究の主たる目的でもある，原価計算の計算機能を記述・操作・表現する事に特化した共通言語の開発を行った．結論から言うと，配賦計算を対象に，関数型の記述・操作言語を開発することで，計算過程作成者が用いた計算の手続きや仮定を表現する事に成功した．具体的には次の図表を参照したい．

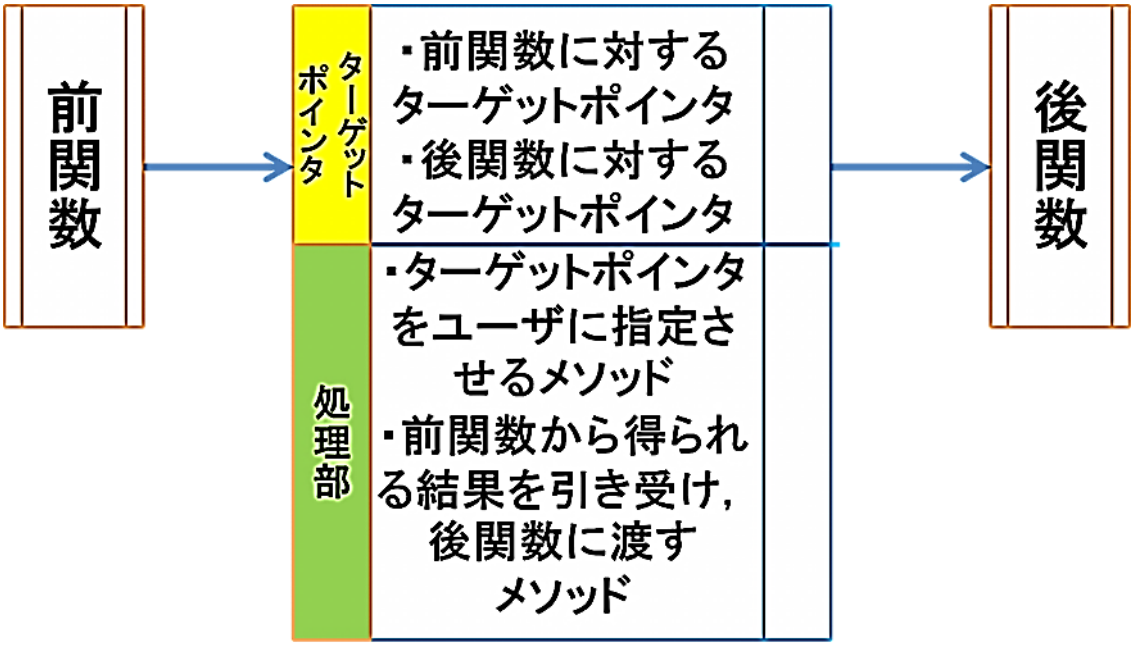
図表 3　関数の階層化による計算ステップ構築



図表 3 で定義した関数のように，配賦計算には配賦する際のドライバにユーザの意図が入るため，配賦計算を行う前にドライバを計算するための関数を設け，そこで計算された結果を，配賦計算関数が呼び出し計算のステップを構築出来るようにした．

次に，処理の流れの記述・操作化に関する記述・操作言語化を考察した．図表 3 では，配賦処理を分割した上で関数化し，計算過程を関数から関数を呼び出すという入れ子による階層化により表現しようとしてきた．しかし，この関数の呼出による計算過程の表現という形態を取ってしまうと，引数に関数を入れ子して使うため，関数の繋がりを辿ることが困難になり，このモデルを用いても全てのユーザに計算構造や計算の仮定を表現・伝搬することが困難となる．そこで，関数を分割してつなぎ合わせて使用できる環境を整えることによりこの問題点の解決を図った．

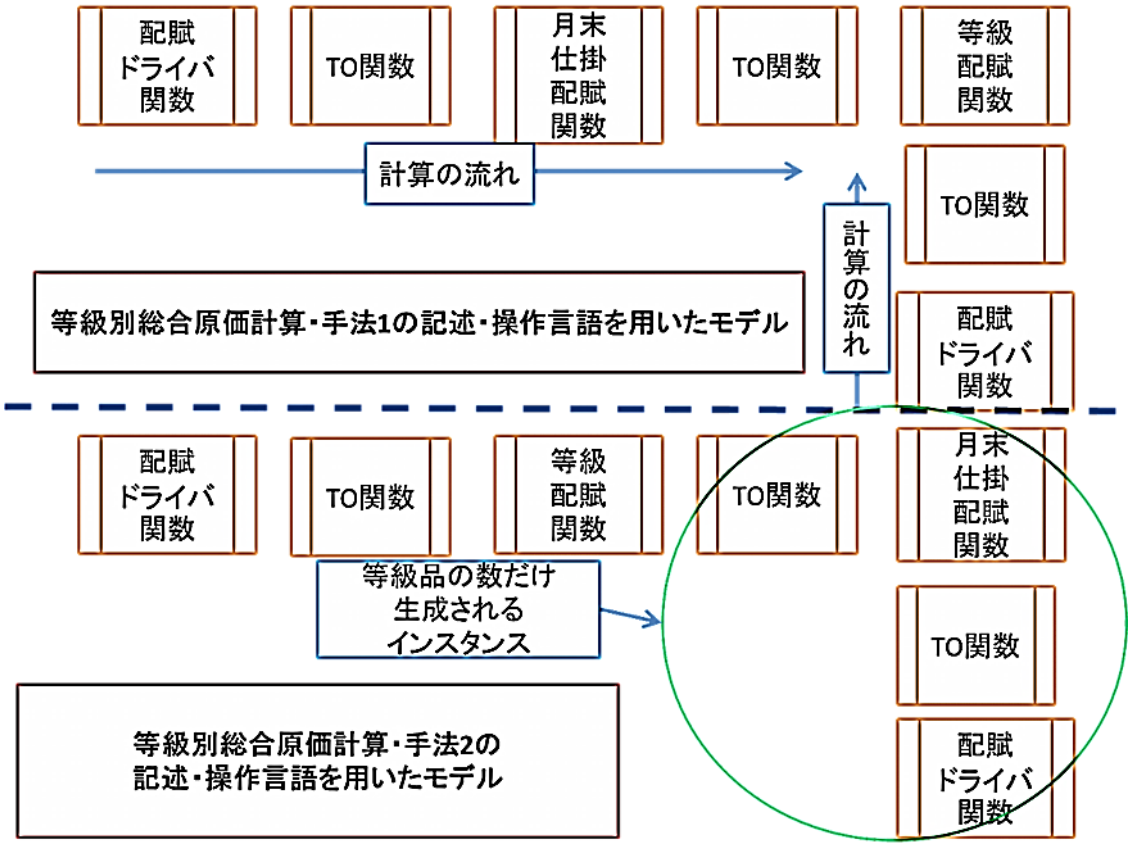
図表 4　TO 関数イメージ



図表 4 に示した TO 関数を用いることで，関数の入れ子構造を関数の連結構造へ変換することが可能となる．この事により，作成した関数を用いて計算構造のセマンティックモデルを構築する基盤を整えることが出来るという副次的な効果も生まれるのである．

では，この作成した記述・操作言語群を用いて，等級別総合原価計算の計算仮定の違いを表現する(図表 5).

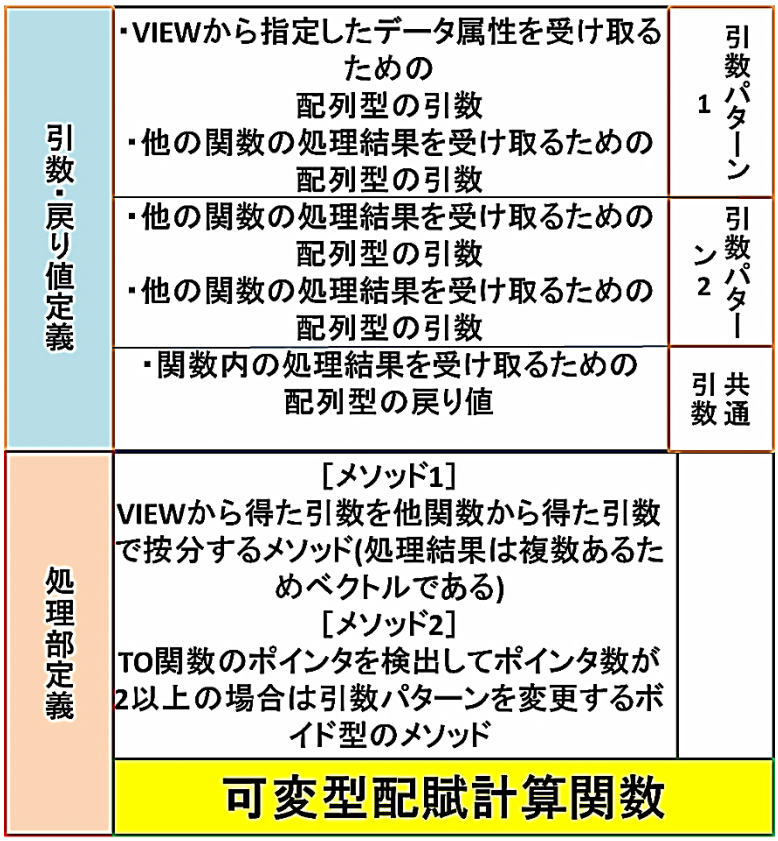
図表 5　等級別総合原価計算の記述・操作言語を用いたモデル図



図表 5 の様に，呼出可能な関数と，関数間の関係性を記述するための TO 関数を用いることで，原価計算の計算目的のためのセマンティックなモデリングをしつつ，モデリングの素材は関数であるため，そのまま処理を実行出来る環境を提示することが可能である．また，原価計算のためのライブラリを用いて計算構造をモデリングすることを続けることで，図表 5 の様に，似たモデル図が作成出来てもその違いを一目見ただけでユーザが認識でき，計算構造のモデリングパターンの構築が実現できる可能性があることにも意義があると言えよう．

ただし，図表 6 から分かるように，計算順序の二つ目の配賦関数が 2 つの関数の処理結果を受け取ることになり，図表 3 で開発した配賦関数の引数では受け皿となり得ない．そのため，配賦関数を次の可変型関数に再開発した．(図表 6 参照)

図表 6　配賦計算関数の可変化



この可変型関数によって，実際にモデルから計算を実行しても引数エラーが生じることが無くなり，配賦モデルであると同時に計算実行可能なシステムとしての DSL 開発に成功した．

この様に，本稿で考察した DB 基盤と計算構造基盤を踏襲しつつ，各基盤が実現できなかったユーザに対する記述・操作言語を活用するためのライブラリ環境を提供することにより，ある目的のための原価計算をモデリングしながら計算を実行することが可能となり，また，そのモデル図を用いることで，その計算構造を他のユーザに伝搬することが可能となるのである．この事により，本稿を通しての目的である，原価計算における誤解や解釈の差異をシステム的に解決することの実現につながると十分に言えるだろう．

最後に，各情報技術基盤では所与条件とされていた専門外の領域であっても全てのユーザがそのモデルを構築出来るという条件の不備に関しては，専門外の領域のモデル化はその専門のユーザに行ってもらう方が構築されるモデルに関する信頼性も確保出来る．その為の基盤にマルチエージェント概念を用いる事で，新たな基盤構築の実現が出来る可能性がある事を提示したが，この点に関してはまだ起案の段階であり，本項ではテストモデルを構築したが実用にはまだ実験が必要であるため，今後の課題として残った．

5. 原価計算の計算機能専用の記述・操作言語を構築することにより生まれた構

造マトリクスの新たな評価

原価計算の計算機能に関する(特化した)記述・操作言語という観点から構造マトリクスの評価を行うことで，構造マトリクスの原価計算のための DSL といった新たな評価の視点を見る事ができる．この評価の視点を新たに創出することには意義がある．第 4 章で行った評価は，技法自体の評価や，汎用的な技法として生み出された構造マトリクスであればどの様な計算の技法に当てはめても上手く稼働するといった仮定の下，原価計算システムとして運用に用いられ上手く稼働したことに対する評価が主たるものであった．

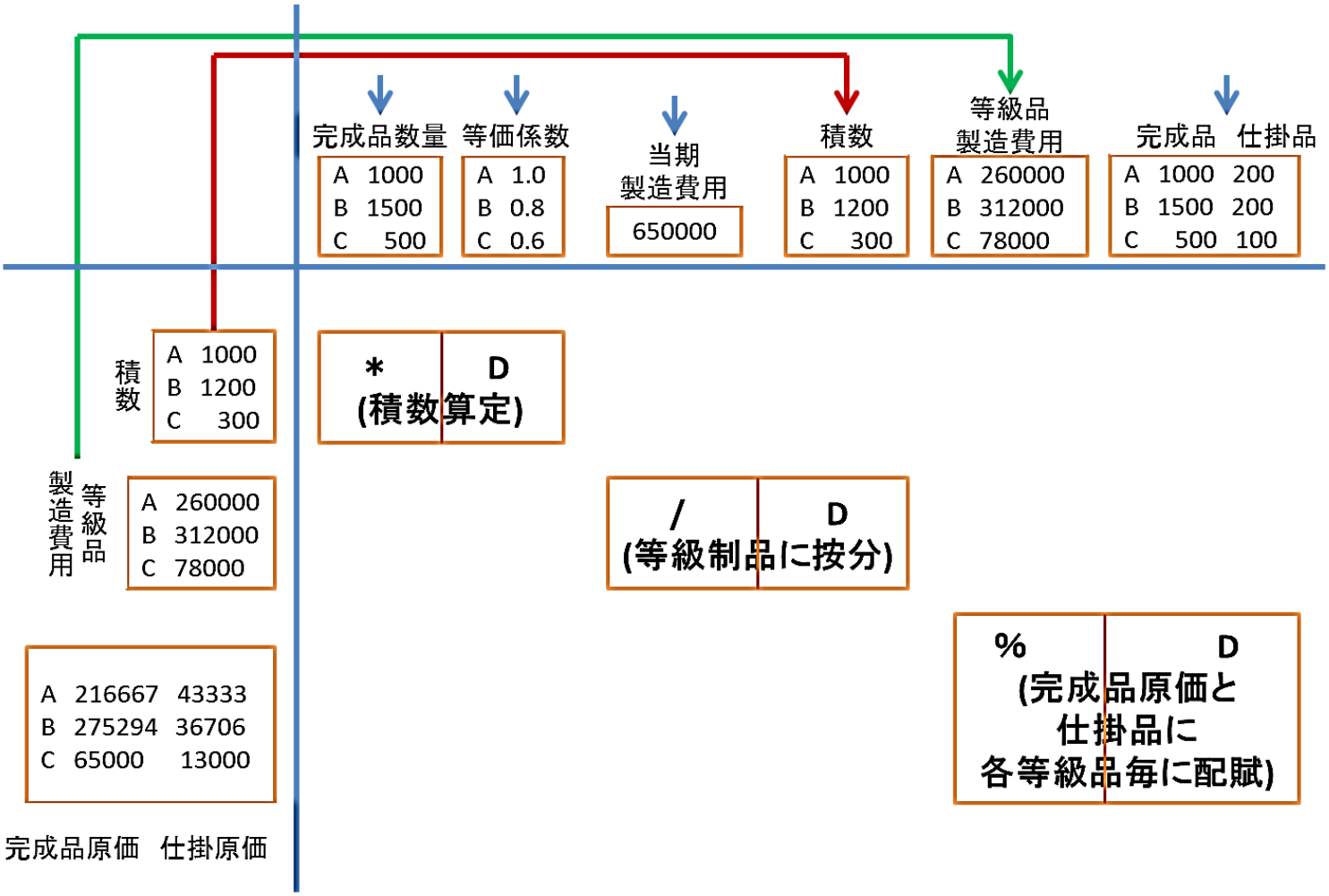
しかし，ここで行う原価計算に用いられた計算の手続きや仮定ための DSL であるといった視点でこの技法を再評価することで，本来の原価計算のための技法としての有用性を知ることができる．

まず，構造マトリクスが DSL 的な要素を持ち合わせるのかといった評価を行った結果，DSL の開発アプローチでも特殊な形態である，言語ワークベンチに該当することが分かった．言語ワークベンチは DSL 開発環境を，DSL を利用するための環境と同じドメイン(ないしツール)に持っているといった点で，他の DSL 開発アプローチとは異なる．つまり，他の DSL は言語としての意味合いが強いものであったが，言語ワークベンチは言語の開発と利用を同じ環境で使用できる点で，統合的な開発利用のための一つのツールとして成り立つといったところに特殊性があるのだ．

次に，DSL といった視点で構造マトリクスを見た場合，原価計算の計算機能のための記述・操作言語としての役割を充足しているのかといった評価を行う．

具体的に，図表 7 の構造マトリクスを用いた等級別製品配賦モデルを用いて構造マトリクスが上述した役割を充足しているのか評価を行う(図表 7 参照)．

図表 7　構造マトリクスを用いた等級別製品配賦モデル



まず，図表 7 中に出てくる CAL 記号について次の図表 8 に説明をまとめた(下図表 8).

図表 8　図表 7 中のモデル内に出てくる CAL 記号の説明　参考：外山(2000)を筆者編集

D	Dummy の意味；何も計算しない。複数の入力ブロックを参照する計算 CAL 記号 P,C,＊,/,%,X と同一行に使用し計算の順の同期化(Multiple Wait)、およびマクロ･テーブル上での同時使用の明示記号として使用する。
＊	Inter Multiply 上辺ブロック相互の掛け算の意味；記号＊に続いて直上の上辺ブロック以外の固有名を指定し直上の上辺ブロックと乗算する。通常、記号 D と併用する。
/	Inter Divide 上辺ブロック相互の割り算の意味；記号/に続いて直上の上辺ブロック以外の固有名を指定し直上の上辺ブロックと割り算する。通常、記号 D と併用する。
%	Prorate 割り掛けの意味；記号%に続いて直上の上辺ブロック以外の固有名を指定し、その数値の比率を算出し直上の上辺ブロックと乗算する。通常、記号 D と併用する。

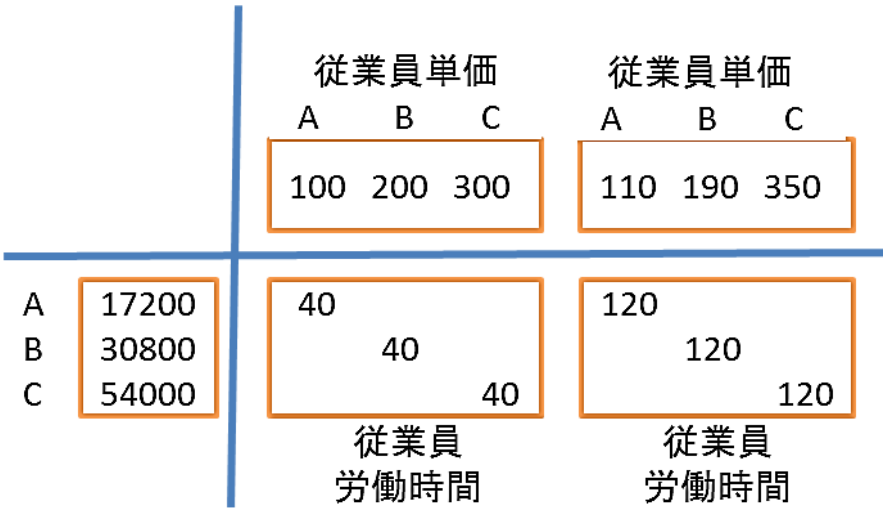
図表 7・8 と DSL 要素の確認で行った評価を元に，原価計算の計算機能に対する記述・操作言語としての役割を評価する.

図表 7 中にあるように，構造マトリクスでは原価計算の計算構造を，シート全体を用い

て表現しており，計算の開始地点は上･左辺部のセルが関係していないセルが先決変数となる．また，原価計算の計算構造の中にある計算仮定を，CAL 記号を用いて表現している．更に，上･左辺部の数値セルに対して階層表現を用いることで数値情報にデータ属性などの意味づけを行えるため，計算過程で数値情報がどの様な処理を受けるとデータ属性が変移するのかが目視で確認出来る点も注目すべき点である．この図表 7 の配賦モデル例から，構造マトリクスという汎用的な DSL 開発環境を用いても原価計算の計算機能に対する記述・操作言語基盤として活用出来ることが確認出来た．

しかし，構造マトリクスは増分原価モデルの実際原価モデルの表現といった点で使用に制限がかかる(図表 9)．

図表 9　同一のデータ属性項目が同一シート上に複数生じるケース



図表 9 は，ある期間の従業員単価に変動があった場合の従業員労務費計算の例であるが，この場合，同一シート上に同じデータ属性の項目が複数出力されることとなり，記述・表現共に煩雑で見にくいものとなってしまう．この様な場合，変動が起きた時点でシートを変更して新たなシートに記述するか，単価の変動を平準化するなどして多少の精緻な計算を実現できない事を承知の上で計算を行う事になるなどの障害が発生する[1]．

この様に，増分原価計算時に，単価の変動といった時系列データを同一シート上に取り込むためには同じ単価属性の項目を複数個登場させなければならないため，意図を伝搬しやすい環境を提示している当技術の良さが損なわれる可能性もあるという点ではデメリットとなる可能性もある．

しかし，構造マトリクスでは当期売上げ台数といった収益となる変数を先決変数として活用することで，その収益を生み出す原価構造をシート上に出力するといった，一般的な積上げ計算方式の増分原価計算とは表現や計算の流れが異なるものの，収益･原価計算と言

[1] このことは，データ入力時に，スキーマから引っ張って来た様々なデータ属性情報を持つ多次元ベクトルが適用されているものを，構造マトリクスの構成要素に対応したデータ属性にあてはめると自動的に多次元ベクトルが射影により次元削除され，特定のデータ属性しか活用されない為である．

う, 収益との結びつきが確認出来る原価を計算する原価計算の本質を, 構造マトリクス DSL 環境に出力し，更にシート上で様々な諸条件を操作出来るため，増分原価計算においても有効なツールであることは確かである.

6. 残された課題

本研究に直接関係する課題として，開発した原価計算に用いられた計算の手続きや仮定に関する記述・操作言語のバリュエーションに関する考察と，記述・操作言語を用いて構築した原価計算の計算体系のモデリングパターンのバリュエーションに関する開発の少なさである． 特に，増分原価計算専用の DSL とモデリングパターンの開発は原価計算に最も必要な部分であるため，開発を急ぐべきであろう.

また，原価計算の計算機能である増分原価といった点に関して課題をいうなら，特に生産管理システムとの連動をいかに実現するのかといった課題がある．この事は，本稿第 1 章でも述べた様に，現在は極めて変化の激しい収益を産む構造の変化が起こっており，それに伴い，原価発生の構造の変化も大なり小なり起こっている現状で，この原価発生の構造の変化の影響を直に受ける原価情報は，増分原価であるため，原価発生の構造の大部分を占める生産管理システムとのスムースな連動が必要となる.

そういった意味では, 西岡(2011)で提唱されたリアルタイム原価管理といった手法はこの課題に対するクリティカルな解決策であると言えよう．しかし，リアルタイム原価管理は文献を読む限りではスケジューラの知識をかなり深く理解していなければ，専門領域外のユーザがこの技法から生じる情報の価値を理解することが出来ないといった課題もあるのではないかと考える.

この様な原価計算と直結した関係性を持つシステムの仕組みを理解するためのパーサー(構文解析器)のような機能を，原価計算システムを研究する立場としては研究する必要性もあろう.

また，本研究の枠を超える課題ではあるが，次の課題も原価情報を作る上では考慮に入れなければならない点だと考えられる．それは，企業間を超えた原価計算システムを構築するといった場合，見せてもいい原価情報とそうでない原価情報をシステム上でどの様に表現するのかといった課題であり，今後出てくる可能性がある課題であると言える．このような課題に対して文献としての記録はないが，尾畑が提唱するオブジェクト指向原価計算では，オブジェクトの隠蔽化といった特質を利用してその対策が可能であるということを伺ったことがある．確かに，オブジェクト指向の特質に，オブジェクトの仕様を見せなくする仕組みがあるが，それをユーザにどの様に表現するのかといった部分で課題があるのだと考える.

この様に残された課題を述べてきたが，この研究を通して原価情報という定量的な情報

が，企業活動の管理体制の強化のための全ユーザの共通言語となれば幸いだと考えている．