

多次元尺度構成法による概念活動モデルとクラス図の連動

井田 明男^{†1,a)} 金田 重郎^{†1}

概要: システム開発の現場では、各工程のモデルを連動させ、理解性と生産性を向上させることに期待が高まっている。たとえば、最上流工程でソフトシステムズ方法論を採用し、上流工程以降はオブジェクト指向アプローチを採用した場合、前者の概念活動モデルと後者のクラス図を連動できれば、直感的にわかりやすく、かつ保守性、再利用性の高いシステム構築に貢献できる。本稿では、システムの根底定義から出発し、概念活動モデルから抽出した責務を多次元尺度構成法により空間布置する手法を提案する。もともと、ソフトウェア工学は、「本質的に関連性が強い要素同士を凝集しておけば、将来の改造に対して安定である」との視点に立って責務の配分を設計してきたが、本提案はその視点を支援するものである。いくつかの例題にて提案手法を試した結果、妥当と思われる責務の空間布置を得ることができた。したがって、本手法は概念活動モデルのような最上流の成果物とクラス図に代表される中上流の成果物との橋渡しを行う一つの手法になり得ると考える。

キーワード: ソフトシステムズ方法論, 概念活動モデル, オブジェクト指向アプローチ, 知識の責務, 振る舞いの責務, 多次元尺度構成法

Applying the Multidimensional Scaling for Linking Conceptual Activity Models and Class Diagrams

AKIO IDA^{†1,a)} SHIGEO KANEDA^{†1}

Abstract: In the field of systems development, linking models are expected for each process for improving understandability and productivity. For example, when Soft Systems Methodology is applied to the uppermost process in a stream and an Object-Oriented Approach is applied to the upstream and midstream processes, if the conceptual activity model of the former and the class diagram of the latter can be linked, then we can build systems that are intuitively simple to understand, easily maintained, and reusable. This paper considers a system's root definition and proposes an approach for spatial arrangement through multidimensional scaling of responsibilities extracted from a conceptual activity model. Our proposed method based on the idea that the strong connection among the responsibilities provides robustness for class diagrams, even if the business procedures change in the application domain. We tested our proposed approach on several samples and obtained a spatial responsibility arrangement that is considerably reasonable. Therefore, we believe this approach has the potential to bridge artifacts from the uppermost stream, including conceptual activity models and midstream artifacts represented by class diagrams.

Keywords: Soft Systems Methodology, Conceptual Model, Object-Oriented Approach, Responsibility for Knowledge, Responsibility for Behavior, Multidimensional Scaling

†1 同志社大学大学院・理工学研究科, 〒610-0321, 京都府京田辺市多々羅都谷, 1-3
Graduate School of Science and Engineering, Doshisha Uni-

versity, 1-3, Miyakodani, Tatara, Kyotanabe-city, Kyoto-pref, 610-0321, Japan Doshisha University
a) ideaworks@kcn.jp

1. はじめに

今日のシステム開発の現場では、多様化、複雑化する要求を合意形成し、機敏かつ柔軟に実現していくために、モデルとコンポーネントを中心とした開発プロセスが採用されている。たとえば、最上流工程ではソフトシステムズ方法論 (Soft Systems Methodology, 以下 SSM) など適用して、関係者間で目的についての合意を行い、それら実現のための分析・設計はオブジェクト指向アプローチ (Object Oriented Approach 以下 OOA) を採用するといったプロセスがその典型例として考えられる。そのため、SSM で得られた最上流のモデルを OOA のモデルと連動させることに大きな期待が寄せられている。

しかしながら、SSM の概念活動モデルのような最上流工程の成果物と OOA のクラス図のような中上流工程の成果物を連動するための方法は確立されていないのが現状である。

一方、多次元尺度構成法 (Multidimensional Scaling, 以下 MDS) は、心理学の分野で発展し、データの潜在的構造を幾何学的に表現するために非常に有効な手法となっている。近年マーケティングや行動科学などの多く分野で利用されており、最近ではビッグデータブームで再び脚光を浴びている。過去に MDS を OOA の構造パターンの評価に利用する提案はあった[1]が、SSM のモデルと OOA のモデルを連動させるために利用したという例はないようである。

そこで、本稿では、SSM の関連システムの根底定義 (root definition) から知識の責務と振る舞いの責務を抽出し、それらの間の相互の関連度合いを簡便に数値化する基準を試案する。そして、責務間の関連度合いの行列を非計量 MDS にて解析し、責務の空間配置をオブジェクトの責務配分に援用することで、SSM のモデルと OOA のモデルを連動する手法を提案する。

以下、第2章は、提案手法の関連研究の概要について説明した後、第3章では、例題を用いて提案手法を説明する、第4章では別の例題についての提案手法の適用結果について報告する。第5章はまとめである。

2. 提案手法の関連研究の概略

2.1 ソフトシステムズ方法論

SSM は、問題が明確に定義されていない状況において、ステークホルダの世界観 (ビジネスに対する想い) のアコモデーション (異なる視点・価値観を持つ人同士が、議論を重ねて互いの視点・価値観を共存させること) を探り、問題の解決に導くプロセスである[2][3]。このような特性が近年の情報システム開発の状況に合致するため関心を集めている。

SSM は、7つのステージから成るプロセスで構成されており、その第4ステージでは、複数の候補から選択された関連システム (目的を持った人間活動システム) の根底定

義から概念活動モデル (Conceptual Model: 目的を実現するために必要な要素活動) を作成する。そして、概念活動モデルをユースケースモデルなどに代表される以降のシステム構築作業のモデルに結び付ける。

2.2 オブジェクト指向方法論と責務配分

一方、OOA の本質は、あるオブジェクトに求められる振る舞いの責務aに対して、その責務遂行に必要な知識の責務bを凝集 (カプセル化) することにより、モジュールとしてのオブジェクトの強度を高めつつ、他のオブジェクトへの依存性 (結合度) を弱めて、保守性、再利用性、拡張性の高いシステム構築に資することである。そのため、オブジェクトの識別とオブジェクトへの適切な責務の配分は重大な関心事であり、従来から多くの原則やガイドラインが提案されてきた。

たとえば、ヤコブソンの BCE[5]、ワーフスブラックのロールステレオタイプ[6]、ラーマンの GRASP[7]、マーチンの SOLID 原則[8]などがある。とくにラーマン[7]では、責務を割り当てるときのもっとも基本的な原則として「情報エキスパート」を提案している。すなわち、「責務の遂行に必要な情報 (知識) を持っているクラス、すなわち情報エキスパートに振る舞いの責務を割り当てる」ことによって、カプセル化が維持される結果、低い結合度 (low coupling) とより高い凝集度 (high cohesion) のクラスの定義が促進され、ひいては頑強で保守性の高いシステムが構築できるとしている。

しかし、これらはいくまでも責務分割のための一般的な基準やガイドラインを提供してくれるだけであり、利用者の個別の問題に即したヒントを提供してくれるわけではない。小さな例ではあるが、ビリヤードゲームのシミュレータを OOA で開発することを想像した場合、ボールと何かの衝突判定と、衝突後のエネルギーの交換を、それぞれどのクラスの責務にすべきかについて上記ガイドラインだけで判断することは容易ではない。

結果として、責務配分の妥当性が評価可能となるタイミングは、ユースケース実現の構造モデリングと振る舞いモデリングが一通り完了した後になってしまうという問題がある。できれば、もっと早い段階で、オブジェクトへの責務配分のアウトラインを把握することはできないだろうか。

2.3 多次元尺度構成法と責務空間布置

MDS は、計量 MDS と非計量 MDS に大別される。計量 MDS は、間隔尺度以上のデータについて、データの中に潜む構造をできるだけ少数次元の空間で幾何学的に表現する手法であり、非計量 MDS は計量 MDS を順序尺度のデータ

a クラスが行わなければならないこと[4]、メソッドはこの実装である。
b 振る舞いの責務を提供するために、クラスが知っていなければならないこと[4]、属性はこの実装である。

に対しても扱えるように拡張したものである[9].

MDS をイメージ的に理解するために北海道の地図を持ち出す。地図から都市間の距離を求めるのは定規で測れば容易である。しかし、逆に都市間の距離関係のみが与えられている場合、その情報だけから都市を平面上に布置して地図を復元することは容易ではない。

表 1 北海道の都市間の直線距離(km) : 文献[10]より引用

	札幌	旭川	稚内	釧路	帯広	室蘭	函館	小樽
札幌								
旭川	115							
稚内	274	202						
釧路	249	188	358					
帯広	152	118	313	98				
室蘭	88	200	360	290	195			
函館	152	263	413	330	240	64		
小樽	30	130	266	277	180	98	158	

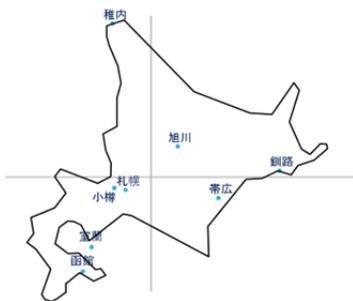


図 1 MDS による空間布置に白地図を重ね合わせたもの

MDS はこのような問題を解析的に解く手法である。表 1 は北海道の 8 都市間の直線距離を測定したものである。図 1 は表 1 のデータを MDS にかけて X-Y 平面上に都市をプロットしたものである。白地図を重ね合わせると、位置関係は実際の地図上の都市とほぼ等しく復元されていることがわかる。

MDS は本来分かっている構造を可視化するのに用いることはないが、このようなモノとモノの一対間の距離的な関係（近似性データ、非近似性データのいずれも扱える）を読み込んで、モノをできるだけ少数の次元（可読性からいえば 3 次元までが望ましい）空間に配置する手法である。

中でもクラスカルの方法[9]は、非計量 MDS の一種で、空間布置の適合度の指標としてストレスと呼ばれる値を用いる。ストレス値と空間布置適合度の評価の対応は概ね表 2 の通りであるとされる[9][10].

表 2 ストレス値と適合度の評価

ストレス値	評価
0.2	あまりよくない(poor)
0.1	まずまず(fair)
0.05	よい(good)
0.025	すばらしい(excellent)
0	完璧(perfect)

3. 提案手法の説明

以上のような状況と成果を踏まえ、本稿では、MDS を SSM と OOA のモデル連携に用いる方法を提案する。

具体的には、要求の根底定義から知識の責務と振る舞いの責務を抽出し、それらの間の相互の関連度合いを簡便に数値化する基準を試案する。そして、責務間の関連度合いの行列を非計量 MDS にて解析し、責務の空間配置を試みる。本提案の全体像を図 2 に示す。

図 2 の通り、提案手法では、SSM のシステムの根底定義から出発して、概念活動モデルから責務を抽出し、それらの間の関連度合を求めて、責務の空間布置を得る。これを後続の OOA のクラス図作成等の参考にしようというアイデアである。以下、提案手法のプロセスについて例題を用いて説明する。

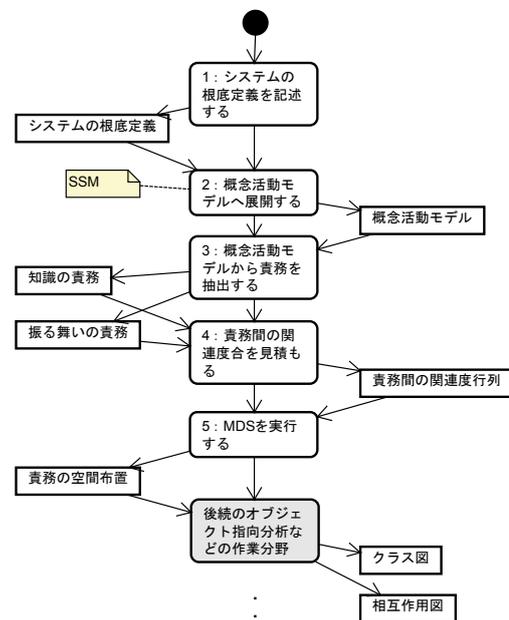


図 2 提案手法の全体像

3.1 STEP1 : システムの根底定義を記述する

システムへの要求は、突き詰めれば、「いつ」、「誰が」、「何のために」利用するシステムなのかを定義することである。たとえば、ユースケースモデルでは、「いつ」はビジネスイベント、「誰が」はアクター、「何のために」はユースケースに対応する。また、SSM においては、「(Z) ~するために、(Y) ~することによって、(X) ~するシステム」の形式で表現されたものは関連システムの根底定義と呼ばれる。根底定義を用いて「想いの活動」を表現しておくことは、SSM の学問的基準である回復可能性を担保するために重要であるとされる[2].

提案手法では、SSM の要求定義の形式として、SSM の根底定義に「誰が (A)」の情報を追加した形式を採用する。すなわち、「(A) ~が、(Z) ~するために、(Y) ~するこ

とによって、「(X) ~するシステム」という形式を用いる。行為者としての主語を明確にすることで、概念活動モデルの各ノードを行為文として切り出しやすくする狙いがある。たとえば、

【例1】「子供を入塾させようと考えている親 (A) が、塾を選ぶために (Z)、塾、教科、開講場所、月謝、特徴、講師などを検索することによって (Y)、比較検討するシステム (X)」

【例2】「ビリヤード愛好家が (A)、ゲームの腕を上げるために (Z)、コンピュータ上でゲームの動きをシミュレートすることによって (Y)、ボールの打ち方を学ぶシステム (X)」

などがこの提案手法で用いる根底定義の例である。

3.2 STEP2：概念活動モデルへ展開する

システムの根底定義は、システムへの機能的な要求を凝縮している。しかしながら、このような簡潔な定義からシステムの構成要素が持つべき責務をいきなり抽出することは情報不足のためできない。そこで、関係者（システム開発の依頼者と提供者など）間でモデリングセッションを行いつつ、根底定義から概念活動モデルに展開する。

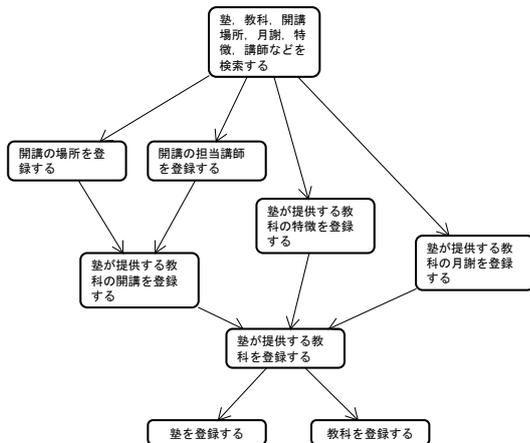


図3 例1の根底定義から展開した概念活動モデル

概念活動モデルは、根底定義で表現された手段 (Y) を稼働させるのに必要な活動を 7 ± 2 個程度の動詞 (V) + 目的語 (O) の形式で表現し、おのおのの活動を論理的依存性に照らして矢印 (目的→手段) で結合したものである[3]。図3は、例1の根底定義を概念活動モデルに展開した例である。

3.3 STEP3：概念活動モデルから責務を抽出する

概念活動モデル上の名詞を知識の責務、動詞を振る舞いの責務として抽出する。この作業は、日本語の要求記述文からオブジェクトを抽出するのに比べて格段に容易である。なぜならば、日本語の要求記述文は形式化されておらず、

また、日本語の特徴から存在文が多いのに対して、概念活動モデルへ展開後は「V (動作動詞) + O (名詞)」の形に整形されているからである[11]。

筆者らは先行研究において、英語の品詞とクラス図の要素の対応関係を明らかにしている[12]。この研究では、要求記述に出現する可算名詞がクラスまたは属性、不可算名詞が属性または属性値、動作名詞が操作、状態動詞が関連にそれぞれ対応するとしている。可算名詞については、クラスか属性かを俄かに判断することは難しい。しかし本稿の手法では後で MDS による責務の空間布置を確認するため、この段階では単純に名詞を知識の責務、動詞を振る舞いの責務とみなしてよい。そのため、オブジェクトにすべきか属性にすべきかの判断にこの段階で迷う必要はない。

3.4 STEP4：責務間の関連度合を見積もる

本手法では、後で MDS を利用するため、3.3 節で抽出した責務（それらはオブジェクトの候補、属性の候補、および操作の候補であるが、すべてひっくるめて）おのおの間の関連度合を数値化する必要がある。

関連度合の見積もりは、手作業で行うならば、できるだけ簡便な方法が望ましい。そこで、著者らは、以下のような基準で数値化することにした。基準は前述の情報エキスパートの原則[7]を満足すべく試行錯誤の末、辿りついたものである。数値は、関連度合の強さに応じて順序尺度を満たしていればよいため、0 から 4 の 5 段階の整数値を用いることにした。ただし、4 (最強) は同一の責務 (そのもの) 間の関連度合であり、対角成分の既定値となる。また 0 は関連度合が未定であることを意味する。対角成分と 0 は表示しないため以降の表にそれらの値は出現しない。

3.4.1 知識の責務同士の関連度合の数値化基準

この基準は、ある知識の責務にライフサイクルが依存する知識の責務は、ライフサイクルの依存性がないその他の知識の責務よりも近い位置に配置されるべきであるという考え方に基づいている。

具体的には、知識の責務間の関連度合の数値化では、その根拠を概念活動モデルのノードに記載された日本語の格助詞「の」に求めることにした。格助詞「の」の用法は多様であるが、概念活動モデルのノード上に記載される「の」は自ずと名詞と名詞を結び付ける「の」に限定されてくる。そしてそれらは、所属や所有を表現しているため、格助詞「の」で接続された名詞 (すなわち知識の責務) 間にはライフサイクルに関して何らかの依存性があり、それらの知識の責務間の関連度合は強いと考える。

たとえば、「開講の担当講師」であれば、講師は開講までに決定される必要があり、「塾の (による)、教科の、提供」であれば、提供を登録する前に、塾と教科の双方が登録済みである必要がある。

採用した基準を表3に示す。そして、表4はこの基準に従って知識の責務間の関連度合を求めた結果である。

表3 知識の責務同士の関連度合の数値化基準

関連度合値	意味	例
3	両者は所属、所有を表す格助詞「の」で結ばれていて、所属先あるいは所有者が一つの場合	開講の担当講師
2	両者は所属、所有を表す格助詞「の」で結ばれているが、所属先あるいは所有者が2つある場合	塾・教科の提供、(塾の提供、教科の提供に分解できる)
1	両者は所属、所有を表す格助詞「の」で結ばれているが、所属先あるいは所有者が3つある場合	商品の在庫、倉庫の在庫、期末の在庫
0	上記以外の場合	

表4 例1の知識の責務間の関連度合

知識	知識	塾	教科	塾・教科の提供	提供の月謝	提供の特徴	提供の開講	開講の担当講師	開講の場所
	塾								
	教科								
	塾・教科の提供	2	2						
	提供の月謝				3				
	提供の特徴					3			
	開講の開講						3		
	開講の担当講師							3	
	開講の場所								3

3.4.2 振る舞いと知識の責務間の関連度合の数値化基準

この基準は、ある知識の責務に対して、その状態移行も含めてライフサイクルを制御する振る舞いの責務は、ライフサイクルを制御しない他の振る舞いの責務よりも近い位置に配置されるべきであるという考え方に基づいている。

そこで、これらの責務間の関連度合の根拠を CRUD に求めることにした。CRUD とは、永続化オブジェクトを扱うソフトウェアが持つべき4つの基本機能である、Create (生成)、Read (読み取り)、Update (更新)、Delete (削除)の頭文字を並べたものであり、ライフサイクルの制御の度合は $C \geq D > U > R$ の順になる。

たとえば、「教科を登録する」という振る舞いの責務は、「教科」という新しい知識の責務を Create する。また、「開講の場所を登録する」という振る舞いの責務は、「開講の場所」という新しい知識の責務を Create するとともに、「開講」という既存の知識の責務について、その場所という知識の責務を Update すると考える。

採用した基準を表5、例題への適用結果を表6に示す。

表5 振る舞いと知識の責務間の関連度合の数値化基準

関連度合値	意味	例
C or D=3	当該振る舞いの責務は、当該知識の責務を生成(Create)または削除(Delete)する場合	教科を登録すると教科(教科を登録すると教科がCreateされる)
U=2	当該振る舞いの責務は、当該知識の責務を更新(Update)する場合	開講の場所を登録すると開講(開講の場所がCreateされるとともに開講がUpdateされる)
R=1	当該振る舞いの責務は、当該知識の責務を参照(Read)する	提供を登録すると塾および教科(塾と教科がReadされ、提供がCreateされる)
0	上記以外の場合	

表6 例1の振る舞いと知識の責務間の関連度合

振る舞い	知識	塾	教科	塾・教科の提供	提供の月謝	提供の特徴	提供の開講	開講の担当講師	開講の場所
塾を登録する		C							
教科を登録する			C						
提供を登録する		R	R	C					
月謝を登録する		R	R	U	C				
特徴を登録する		R	R	U		C			
開講を登録する		R	R	R			C		
開講の担当講師を登録する		R	R	R				U	C
開講の場所を登録する		R	R	R					U
塾情報を検索する		R	R	R	R	R	R	R	R

3.4.3 振る舞いの責務同士の関連度合の数値化基準

この基準は、ある振る舞いの責務に対して、それを直接的に利用する振る舞いの責務は、それを利用しない振る舞いの責務や間接的に利用する振る舞いの責務よりも近い位置に配置されるべきであるという考え方に基づいている。

そこで、振る舞いの責務同士の関連度合の数値化では、その根拠を概念活動モデルにおける活動ノード間の接続に求めることにした。たとえば、図2では、「塾が提供する教科を登録する」のノードが、「塾を登録する」と「教科を登録する」を直接的に利用しているため値1を与える。

採用した基準を表7、例1について求めた振る舞いの責務間の関連度合を表8、表9に示す。

表7 振る舞いの責務同士の関連度合の数値化基準

関連度合値	意味	例
1	一方の振る舞いの責務ともう一方の振る舞いの責務は概念活動モデル上で直接矢印で結ばれている場合	塾が提供する教科を登録すると塾を登録する 塾が提供する教科を登録すると教科を登録する
0	上記以外の場合	

表8 例1の振る舞いの責務間の関連度合

振る舞い	振る舞い	塾を登録する	教科を登録する	提供を登録する	月謝を登録する	特徴を登録する	開講を登録する	開講の担当講師を登録する	開講の場所を登録する	塾・教科・開講場所・月謝・特徴・講師などを検索する
塾を登録する										
教科を登録する										
提供を登録する		1	1							
月謝を登録する				1						
特徴を登録する				1						
開講を登録する				1						
開講の担当講師を登録する							1			
開講の場所を登録する							1			
塾情報を検索する					1	1	1	1	1	

3.5 STEP5 : MDS を実行する

責務間の関連度合が数値化され、表9のような正方形の下の三角行列が得られたならば、これをMDSにかける。本手法で用いる責務間の関連度合の値は、より近いと考えられる責務間ほど相対的に大きな値が与えられたデータであるため順序尺度である。したがって、MDSのなかでも非計量MDSを選択する必要がある。また、これらは責務間の近似性データであるとして処理を行う。

MDSを実行するには、SPSSのような統計パッケージを使ったり、R言語のような統計解析向けのオープンソースプロダクトを利用するなどの方法がある。しかし、今回はクラスカルの方法を文献[13]の実装をもとにLAMP環境に移植した手組のプログラムを用いることにした。手組のプログラムであればその中身がよく分かっているため、将来の研究において概念活動モデルエディタにMDSを組み込む場合に有利であると考えたためである。

表9のデータをMDSの入力として処理した結果、図4に示す責務の空間布置を得た。

表 9 例 1 に関わる責務間の関連度合 (表 4, 6, 8 の結合イメージ)

責務	塾	教科	塾・教科の提供	提供の月謝	提供の特徴	提供の開講	開講の担当講師	開講の場所	塾を登録する	教科を登録する	提供を登録する	月謝を登録する	特徴を登録する	開講を登録する	開講の担当講師を登録する	開講の場所を登録する	塾情報を検索する
塾																	
教科																	
塾・教科の提供	2	2															
提供の月謝			3														
提供の特徴			3														
提供の開講			3														
開講の担当講師						3											
開講の場所						3											
塾を登録する	3																
教科を登録する		3															
提供を登録する	1	1	3						1	1							
月謝を登録する	1	1	2	3							1						
特徴を登録する	1	1	2		3						1						
開講を登録する	1	1	1			3					1						
開講の担当講師を登録する	1	1	1			2	3							1			
開講の場所を登録する	1	1	1			2		3						1			
塾情報を検索する	1	1	1	1	1	1	1	1				1	1		1	1	

3.6 責務の空間布置の評価についての議論

図 4 を眺めると、塾と教科とが明確に分離され、それらを結んだ線のちょうど中間あたりに提供の月謝や提供の特徴といった責務が配置されているのがわかる。ストレス値については、0.188 と高いものの、同じ例題について、モデラーが手作業でクラス図を作成した場合 (図 5) とよく似た構造を得ることに成功している。

ここで、責務の空間布置を読む際の留意事項について述べる。空間布置には関連は表示されないため、読み手は関連線を補って読む必要がある。ここでの関連は、システムの根底定義に由来する。そもそも概念活動モデルは、一つの関連システムの根底定義 (目的) のために、それを実現すべく展開した目的→手段グラフであるため、おのおののノード上に記載される知識の責務 (属性) は、いずれもが目的の実現のために必要なものである。保守性、理解性、再利用性に目をつぶれば、それらは 1 つのクラスの属性として纏めたとしても動作するシステムを構築することができる。このことから、これら全ての知識の責務間には本来、直接的あるいは間接的に関連が存在していると考えられる必要がある。したがって MDS の空間布置を眺めて、近傍にまとまっている責務群をクラスとして取り出したならば、取り出したクラス間にも関連があるものと考えなければならない。もともと、ソフトウェア工学は、「本質的に関連性が強い要素同士を凝集しておけば、将来の改造に対して安定である」との視点に立って責務の分配を設計してきた[12]が、MDS の出力はその視点を補うものである。

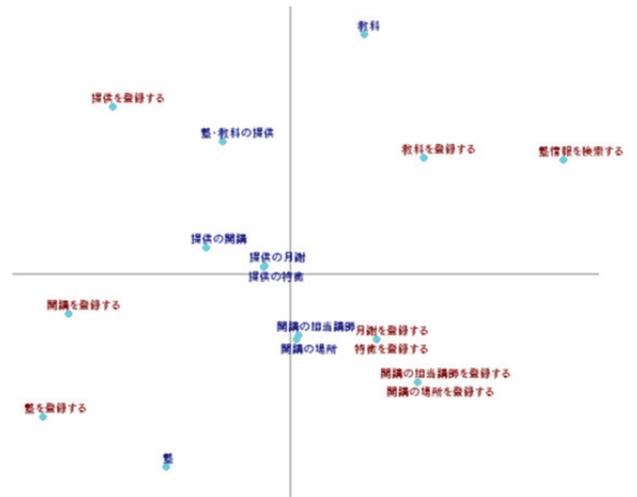


図 4 MDS で得られた例 1 の責務の空間布置
 繰り返し計算回数 44 回でストレス値は 0.188 に収束

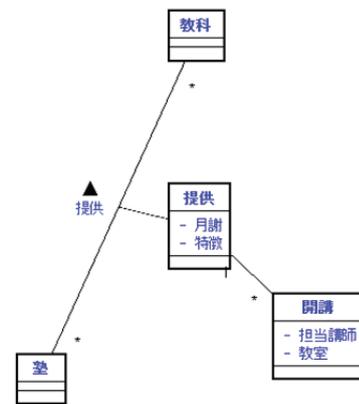


図 5 例 1 についてモデラーが手作業で作成したクラス図

4. 適用実験

本節では、3.1 で示した 2 つ目の例題に対して提案手法を適用する。すなわち、「ビリヤード愛好家が (A)、ゲームの腕を上げるために (Z)、コンピュータ上でゲームの動きをシミュレートすることによって (Y)、ボールの打ち方を学ぶシステム (X)」を根底定義とした例である。例 1 よりも若干複雑な例を用いて責務数を増やした場合のストレス値の変化を観測するとともに、空間布置を参考に作成し

た分析クラス図をもとに設計、実装までを行い提案手法の妥当性を確認する。

根底定義から、概念活動モデルを展開すると図 6 を得る。概念活動モデル上の名詞から知識の責務、動作動詞から振る舞いの責務を抽出し、おのおのの責務間の関連度合を見積もって表に纏めると表 10 を得る。そして、表 10 のデータを MDS にかけて図 7 を得る。今回のストレス値は 0.092 に改善された。これは責務数が増え責務間の情報量が増えたためと考えられる。

図7の空間布置を眺めると、ボールの転がり摩擦が、ボールよりもむしろテーブル寄りに配置されていたり、ボール同士の接触を検知する責務はテーブル寄り、ボールと何かの衝突を再現する責務はボール寄りに配置されているのが興味深い。図8のクラス図は図7の空間布置を見ながら作成したものである。やや主観もあるが、クラス図作成時には図7の情報が大いに参考になった。

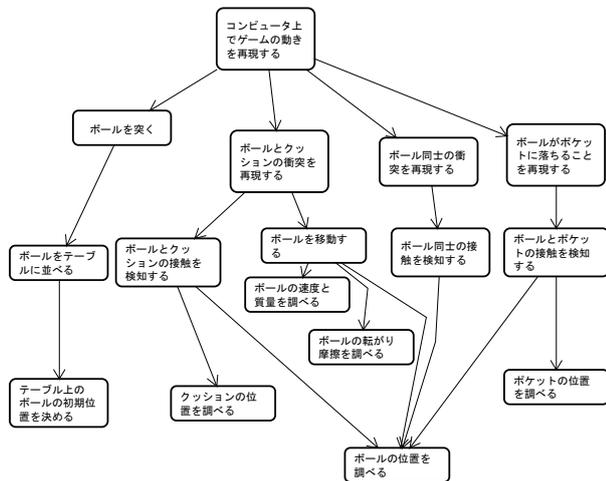


図6 例2の概念活動モデル

さらに、図8のクラス群を Model-View-Controller アーキテクチャパターン[14]の Model とし、View と Controller のクラス群を追加して設計モデルを作成した(図9)。そして、Java 言語にて実装を試みた結果も良好であった。とくに MDS の布置によって分析段階のモデルから接触の検知と衝突の再現を分離できていたことが、ボールクラスの責務を軽減し、後続の詳細設計および実装作業を容易にした。

以上の適用実験から、本提案が示した基準による責務の空間配置は、機能要求に照らして真に機能的なものか否か

は不明であるが、少なくとも高凝集と疎結合を目指して責務のカプセル化を行うオブジェクト指向の本質とは整合しており、OOA の責務分配を支援するものと考えられる。

5. まとめ

本稿では、最上流工程の方法論として SSM, 中上流工程の方法論として OOA を仮定し、システムの根底定義から概念活動モデルを経由して抽出した責務間の関連度合を数値化し、MDS を用いて空間布置したものをクラス図作成の参考にするアイデアを示した。

筆者らは、オブジェクト指向といえば、先にオブジェクトを見つけることに気を奪われがちであったが、逆に、先に責務を見つけて、空間にばら撒けば、関連度合に応じて責務のグループができ、それを眺めてクラス図を考えるアプローチも妥当であると考えられる。また、SSM の成果物を OOA の責務分配設計に繋げる方法を提示した点にオリジナリティがある。

ただし、提案手法を手作業で実施するのは非常に煩雑であるため、将来は MDS を組み込んだ概念活動モデルエディタを実装し、提案手法の責務の抽出からそれらの空間布置までを自動化することを目指したい。自動化が前提であれば、責務間の関連度合をさらにきめ細かいものにすることができる。たとえば、概念活動モデル上のあるノードから別のノードへ至るまでに経由する矢印の本数を責務間の関連度合に加味するなどが考えられる。

実装すれば、クラス図を提案してくれるインテリジェントなモデリングツールが実現できる可能性がある。引き続きこれらのテーマについて取り組んでゆきたい。

表 10 例2の概念活動モデルから求めた責務間関連度合表

責務	責務	テーブル	テーブルのクッション	テーブルのポケット	テーブルのボール	クッションの位置	ポケットの位置	ボールの位置	ボールの速度と質量	ボールの転がり摩擦	テーブル上のボールの初期位置を決める	クッションの位置を調べる	ボールの位置を調べる	ポケットの位置を調べる	ボールの速度と質量を調べる	ボールの転がり摩擦を調べる	ボールをテーブルに並べる	ボールとクッションの接触を検知する	ボールを移動する	ボール同士の接触を検知する	ボールとポケットの接触を検知する	ボールを突く	ボールとクッションの衝突を再現する	ボール同士の衝突を再現する	ボールがポケットに落ちることを再現する	コンピュータ上でゲームの動きを再現する	
テーブル																											
テーブルのクッション		3																									
テーブルのポケット		3																									
テーブルのボール		3	2	2																							
クッションの位置		2	3																								
ポケットの位置		2		3																							
ボールの位置		2			3																						
ボールの速度と質量					3																						
ボールの転がり摩擦					3																						
テーブル上のボールの初期位置を決める		1			2			2	2																		
クッションの位置を調べる		1	1			1																					
ボールの位置を調べる					1			1																			
ポケットの位置を調べる				1										1													
ボールの速度と質量を調べる					1				1																		
ボールの転がり摩擦を調べる					1				1																		
ボールをテーブルに並べる		1			1			2	2	1	1																
ボールとクッションの接触を検知する			1		1	1		1																			
ボールを移動する			1	1	1	1	1	1	2	1					1	1											
ボール同士の接触を検知する					1			1							1												
ボールとポケットの接触を検知する				1	1			1	1					1	1												
ボールを突く					2			2	2									1									
ボールとクッションの衝突を再現する			1			1		2	2				1	1													
ボール同士の衝突を再現する					2			2	2	1																	
ボールがポケットに落ちることを再現する				1	2			2	2															1			
コンピュータ上でゲームの動きを再現する		1	1	1	1	1	1	1	1	1													1	1	1	1	

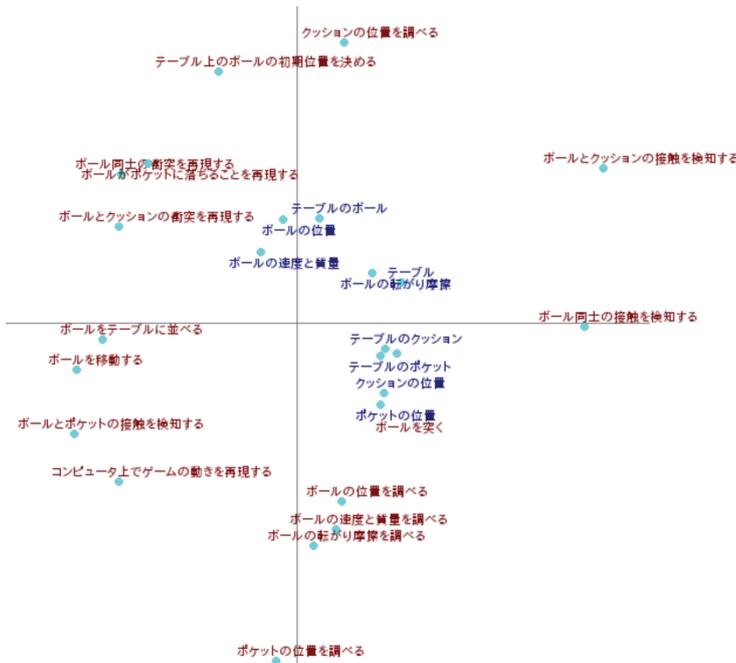


図 7 例 2 の責務空間布置

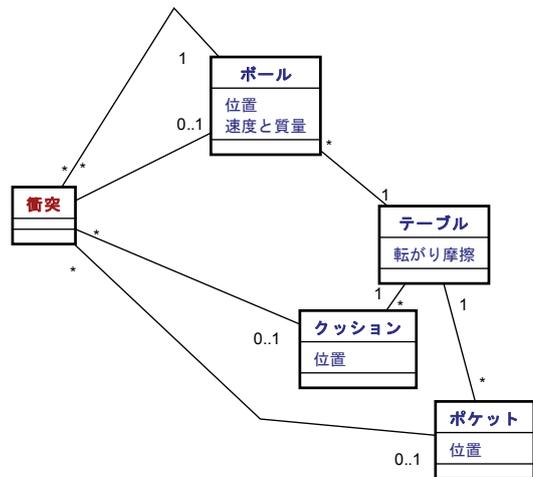


図 8 例 2 の空間布置を参考に作成したクラス図

繰り返し計算回数 73 回でストレス値は 0.092 に収束

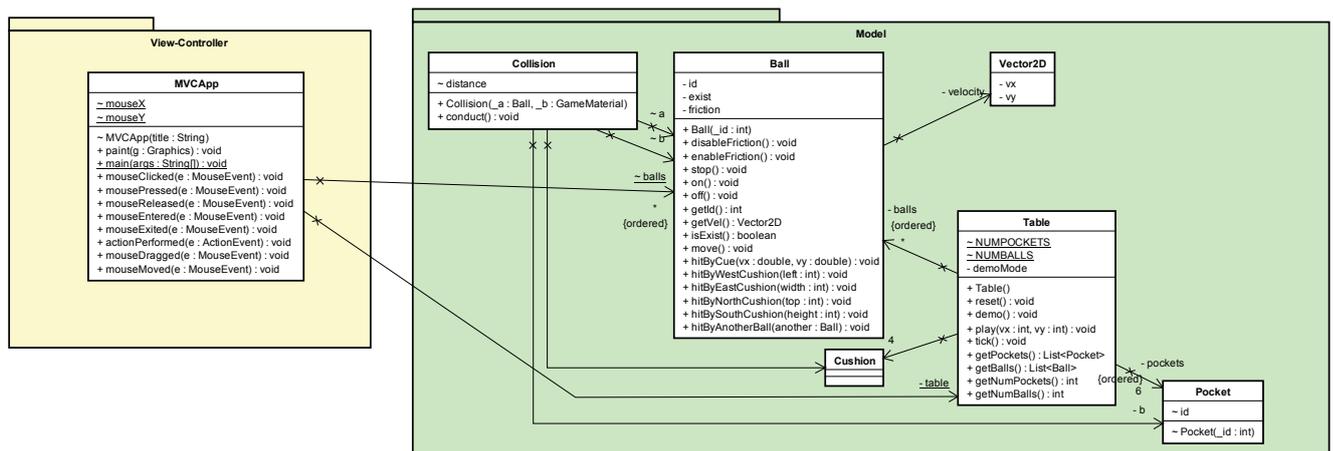


図 9 例 2 の設計のクラス図 (主要なパッケージとクラスのみ抜粋)

参考文献

- 1) 飯島正, 永田守男, 「オブジェクト指向ソフトウェアの構造パターンに対するデータ解析手法による分析評価の試み」, 電子情報通信学会技術研究報告, 知能ソフトウェア工学, pp.9-16, 1995年5月.
- 2) Peter Checkland, Jim Scholes, 「ソフト・システムズ方法論」, 有斐閣, 1994年7月.
- 3) Brian Wilson(著), 根来龍之(訳), 「システム仕様の分析学・ソフトシステム方法論」, 共立出版, 1996.
- 4) Devid Bellin, Susan Suchman Simone, 「実践 CRC カード」, ピアソン・エデュケーション, 2002年9月.
- 5) Ivar Jacobson, Patrik Jonsson, Magnus Christerson, Gunnar Overgaard, 「オブジェクト指向ソフトウェア工学 OOSE—use - case によるアプローチ」, アジソンウェスレイトッパン, 1995年9月.
- 6) Rebecca Wirfs-Brock, Alan McKean, 「オブジェクトデザイン ロール, 責務, コラボレーションによる設計技法」, 翔泳社, 2007年9月.
- 7) Craig Larman, 「実践 UML—パターンによるオブジェクト指向開発ガイド」, プレンティスホール出版, 1998年12月.

- 8) Robert C. Martin, 「アジャイルソフトウェア開発の奥義 第2版 オブジェクト指向開発の神髄と匠の技」, ソフトバンククリエイティブ, 2008年7月.
- 9) Joseph B. Kruskal, 「Nonmetric Multidimensional Scaling: A Numerical Method」, Psychometrika, Vol.29 pp.115-129, 1964.
- 10) 中村永友, 「R で学ぶデータサイエンス 2 多次元データ解析法」, 共立出版, 2009年8月.
- 11) 藤澤裕樹, 岡田裕, 一瀬邦継, 金田重郎, 「ソフトシステムズ論 方法論(SSM)と概念データモデリング(CDM)を用いた業務分析手法の提案」, 情報処理学会研究報告, 2010年.
- 12) 金田重郎, 井田明男, 酒井孝真, 「「クラス図は英語である」との観点に基づく仕様文・クラス図変換メソッドの提案」, 電子情報通信学会技術研究報告, 知能ソフトウェア工学, pp.13-18, 2013年1月.
- 13) 杉本富利, 「多次元尺度構成法」, インフォメーションサイエンス, pp125-130, 1983年8月.
- 14) Frank Buschmann, Hans Rohnert, Michael Stal, Regine Meunier, Peter Sommerlad, 金沢 典子 (他訳), 「ソフトウェアアーキテクチャソフトウェア開発のためのパターン体系」, 近代科学社, 2000年12月.