

ロボット動作設計を対象にした 状態遷移図による概念モデリング教育への モデル駆動開発方法論導入の効果

横田寛明^{†1} 香山瑞恵^{†2} 小形真平^{†2} 橋本昌巳^{†2} 大谷真^{†2}

本研究では、モデル駆動開発方法論を初学者向け概念モデリング教育に導入し、学習者自身が対象物の振舞いを観察する事によるモデルを評価させている。これまでに、2種類の方法でモデル駆動開発方法論による設計手法を導入してきた。一つは机上モデリングとモデル駆動開発方法論によるモデリングを分けて行う方法であり、もう一つは机上モデリングとモデル駆動開発方法論によるモデリングを合わせて行う方法である。本稿では、これら2種類の方法におけるモデル駆動開発方法論導入前後のモデル品質の差、およびモデルに含まれる誤りパタンの分析から、モデル駆動開発方法論導入の効果を検討し、状態遷移図による概念モデリング教育の展開方法を検討する。

Effect of Model-Driven Development Method in Behavioral Modeling Education using Simple State Machine Diagrams for Robot Behavior Design

HIROAKI YOKOTA^{†1} MIZUE KAYAMA^{†2} SHINPEI OGATA^{†2}
MASAMI HASHIMOTO^{†2} MAKOTO OTANI^{†2}

The purpose of this study is to explore the educational methodology about behavioral modeling for beginners. We have been introducing the model driven development (MDD) method in the modeling education for university freshmen. By using the MDD method, we expect our students to be able to evaluate their own model. In this study, we introduce two types of educational procedures with the MDD. In this paper, we analyze the difference quality of models and the error patterns included in students models. We also compare the models that are made before and after the MDD with these two procedures. Then the differences of two MDD procedures are discussed.

1. はじめに

問題解決すべき対象をモデリングすることは、情報化社会を生きる全ての者に必要なリテラシでもある[1,2,3]。対象の動的モデルとしての振舞いを記述する状態遷移図は、ACM や IPSJ 等が示す情報専門教育の主要5分野 (CS・IS・SE・CE・IT) はもとより、一般情報教育 (GE) および高等学校での情報科の学習内容とも関連する。しかし、大学における状態遷移図の教育は、論理回路やソフトウェアの設計手法として実施されている。これらの先行事例では、主として専門教育の一環として取り上げられることが多い。近年では、工学系専門学校生を対象としたモデリング教育の手法として、状態遷移図を用いる提案[4]がされており、学習対象者の低年齢化がみられる。

本研究では、高等教育における専門教育に入る前段階での、一般情報教育の枠組みにおける概念モデリングでの動的な対象を表現する手法として状態遷移図を取り上げる。この活動では、モデル駆動開発手法およびツールを用い、学習者自身にモデル図を評価させる試みを行っている[5]。2012年の報告によれば、モデル駆動開発手法をオブジェク

ト指向モデリング教育に導入している例は国内外に存在しないという[6]。本研究では、必ずしもオブジェクト指向を意識するものではない。本稿では、状態遷移図を用いた概念モデリングに対するモデル駆動開発論導入の効果について、モデル品質の変化に基づき考察する。

2. 本研究のアプローチ

本研究では、本来は学習者自らが成すべき作業を、モデル駆動開発環境の導入とドメイン特化言語を用いた初学者に対するモデリング学習を2010年度より試行している[5]。

2.1 モデル駆動開発について

モデル駆動開発 (Model-Driven Development : MDD) とは対象に関する知識を「モデル」に変換することによって行われるソフトウェア開発手法の総称である[7]。MDD で用いられるモデルとは、UML 等の図形言語で表現されたモデル図を指すことが多い。モデルエディタによりモデル図を作成することでシステムを設計し、さらにモデルコンパイラによりモデル図からプログラムコードを自動生成する。今回使用したモデルエディタ[8]aの例を図1左に示す。プログラムコードをコンパイルすることで、対象デバイス上での実行コードが生成できる。すなわち、MDDにより、

^{†1} 信州大学大学院理工学系研究科
Graduate School of Science & Technology, Shinshu University.

^{†2} 信州大学工学部
Faculty of Engineering, Shinshu University.

a 株式会社 Technical Rockstars の clooca を信州大学様にカスタマイズした。

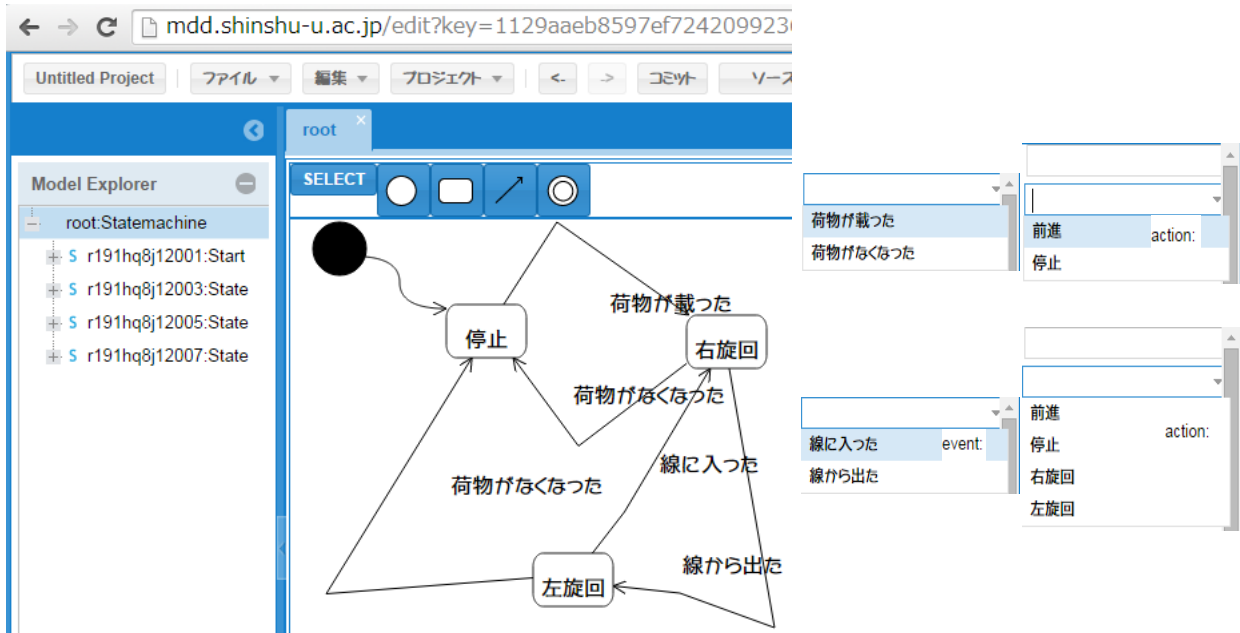


図 1. MDD モデルエディタ (左)と、本稿で示した課題モデル記述用 DSL (右) の例

モデル図からの実行コード生成がほぼ自動でなされる環境を実現できる。この手法の導入は、設計の抽象度を高めることと同義である。設計の抽象度が高まると、“どのように実装(記述)するのか”ではなく、“どのように解決(設計)するのか”といった側面により一層注力できるようになる。

複数の課題を想定した際の MDD の導入方法として、以下の 2 種を考えた。一つ目は机上で全ての課題を考え、その後 MDD を用いる方法 (以下、一括 MDD と称す)。二つ目は、1 課題毎に、机上で考え、MDD を用いる事を繰り返す方法 (以下、逐次 MDD と称す) である。

2.2 ドメイン特化言語について

MDD におけるドメイン特化言語 (Domain-Specific Language : DSL)とは、特定な問題領域における課題解決用に特化した語彙集合である。DSL を用いる事で、学習者に対して解決すべき課題を整理するために語彙を制限することができる。これは思考空間を狭くし、難易度を調節する事が可能となる。本研究で用いた DSL の例を図 1 右に示す。

2.3 本研究におけるリサーチクエスチョン

本研究では、以下のリサーチクエスチョンを設けた。

1. MDD を導入することで、初学者が適切なモデルを記述できるようになるか？
2. MDD を導入することで、自分で誤りを修正できるか？
3. 一括 MDD と逐次 MDD において、結果に差があるか？
これらについて、考察する。

3. 実験方法

3.1 被験者

本稿では 2 つのグループの被験者を取り上げる。被験者 2013 年群は 2013 年度における大学情報工学科 1 年生 78 名とした。実施時期は 2013 年 4 月下旬から 5 月中旬にかけて

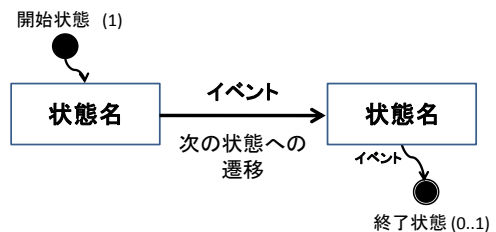


図 2. 本研究で用いた状態遷移図の記法

である。被験者 2014 年群は 2014 年度における同学科 1 年生 85 名とした。実施時期は 2014 年 5 月上旬から 5 月中旬にかけてである。被験者は両群ともに情報工学における知識は持っていない。

3.2 状態遷移図の記法

本研究で導入する状態遷移図の記法を図 2 に示す。UML2.x で規定されているステートマシン図に対して、記述要素を状態および状態名、遷移およびイベント名だけに制限し、以下の制約を与えた。

- 開始状態は、必ず 1 つ設ける
- 終了状態は、存在する場合は 1 つ
- 同じ状態は、1 つの図に複数存在しない
- 1 つの状態から異なる 2 つ以上の遷移は出ない
- 1 つの状態から、同じ名前の遷移は複数出ない

このモデル図を記述できるようモデルエディタのメタモデルを定義した上で、後述するように課題毎の DSL を設計し、被験者が解答する環境とした。

3.3 実験課題

実験課題は、LEGO Mindstorms NXTV2 で動作するサービスの設計とした。課題は 3 問で構成される。課題 1 は荷物搬送を行う課題である。この課題では、ロボットの荷台に荷物が乗っている時には前進し、乗っていない時には停止

する振舞いを満たす事が要求される。課題2はライントレースを行う課題である。この課題では、黒い線をトレースしながら前進していく振舞いを満たすことが要求される。この課題で用いられるコースは台形型とそら豆型である。課題3は課題1と課題2の複合課題である。課題1の要求と、課題2の要求の双方を同時に満たすことが要求される。

荷物搬送に関しては、LEGOのタッチセンサを1つ制御する。タッチセンサがONとなったイベントを「荷物が載った」、OFFとなったイベントを「荷物がなくなった」と定義し、LEGOの状態として「前進」と「停止」の2種類の動作を利用させる。図1の右上が当該モデル記述用DSLである。このDSLは2つの状態と2つのイベントからなる。

ライントレースに関しては、LEGOの光センサを1つ制御する。光センサが黒を感知したイベントを「線に入った」、黒ではない色を感知したイベントを「線から出た」と定義し、LEGOの状態として「前進」・「停止」・「右旋回」・「左旋回」の4種類の動作を利用させる。図1の右下が当該モデル記述用DSLである。このDSLは4つの状態と2つのイベントからなる。

複合課題に関しては、4つの状態と4つのイベントからなるDSLを用いることとした。

3.4 実験手順

2013年群には一括MDDを、2014年群には逐次MDDをそれぞれ導入した。両群に対してMDD導入前後の図を対象に、結果の比較を行う。2013年群は、MDD導入前の図を「宿題図」、導入後の図を「修正図」、2014年群では、MDD導入前の図を「修正前」、導入後の図を「修正後」とする。なお、両群ともに対象物となるロボットは、3人で1台を共有させた。

3.5 モデル図の評価方法

リサーチクエスチョンに合わせて2つの評価観点を設ける。一つ目は課題の達成段階であり、二つ目は謝りパターンである。達成段階については、まず課題毎に正答とみなされた解答の割合を示す。その上で、課題毎に複数の達成段階を想定し、そのそれぞれの段階に相当する解答の割合、MDD導入前後での達成段階の変化、MDD導入後の図における課題2での達成段階と課題3での達成段階の関係について考察する。

誤りパターンについては、まず出現した誤りパタンの種類を示す。その上で、MDD導入前後での誤りパタンの変化、MDD導入後の図における課題2での誤りパターンと課題3での誤りパタンの関係について考察する。

4. 実験結果

ここでは、3.5に示した評価方法に基づき、MDD導入前後の図を比較することで、モデル図の品質の変化およびモデル図に内包されていた誤りパタンの種類を示す。

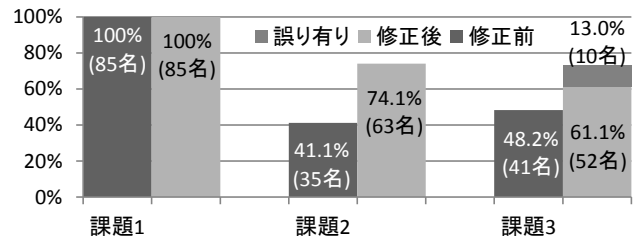


図3. 2013年群のMDD導入前後での課題毎の正答率

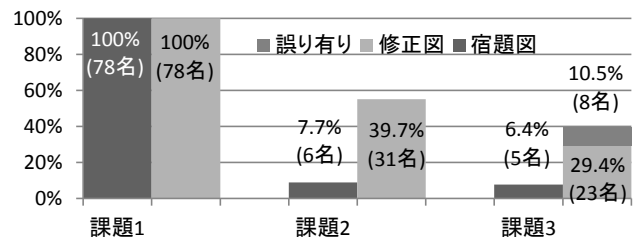


図4. 2014年群のMDD導入前後での課題毎の正答率

表1. 確認された誤りパターン

カテゴリ	パタン	誤りの詳細
記法ミス	P-a	一つの状態から同名の遷移が複数出ている
	P-b	一つの遷移図に同じ名前状態が複数ある
冗長な表現	P-c	自己遷移をしている
	P-d	同名の遷移が同じ方向に複数出ている
要求解釈ミス	P-e	状態を通過する
	P-f	荷物が落ちても止まらない
	P-g	荷物が載っていても動作する

4.1 モデル図の品質の変化

2013年群と2014年群について、各課題の要求を満たした解答の割合を正答率として図3、図4に示す。これらの図では課題毎に左側にMDD導入前、右側にMDD導入後の結果を示している。また、ここでいう正答とは、課題1では荷物搬送が動作した解答、課題2ではそら豆型コースが動作した解答、課題3では荷物搬送ができ、そら豆型コースが動作した解答である。特に、モデル図の品質という観点から、正答ではあるが誤りパターン(4.2で詳述)を含む解答の割合を「誤り有り」として示す。

これらの図から、両群共に課題1は正答率がMDD導入前後で100%であった。課題2、3ではMDD導入前よりも導入後の正答率が向上していた。2014年群の方がMDD導入前の課題2、3の正答率が高かった。「誤り有り」は、課題3のみで確認され、両群とも10%強含んでいた。

4.2 誤りパタンの分析

両群の解答における誤りパターンについて分析した結果、3カテゴリ(記法ミス、冗長な表現、要求解釈ミス)7パタンの誤りを確認した。確認された誤りパターンを表1に示す。これらは先行研究で示された結果[7]と同様であった。

5. 考察

5.1 達成段階

課題毎に、不動作から正答にいたる達成段階を設定した。課題2では「不動作」,「台形(台形型コースは動作したが,そら豆型コースは動作しなかった解答)」,「そら豆(正答)」の3パターン,課題3では「不動作」,「荷物×台形(荷物搬送が動作せず,台形型コースのみ動作)」,「荷物×そら豆(荷物搬送が動作せず,そら豆型コースが動作)」,「荷物○台形(荷物搬送が動作し,台形型コースのみ動作)」,「荷物○そら豆(正答)」の5パターンを設けた。

5.1.1 達成段階の変化

課題2では,2013年群の結果として,MDD導入前後で「不動作」が17.9%から0%,「台形」が74.4%から44.9%,「そら豆(正答)」が7.7%から55.1%と変化した。2014年群の結果として,MDD導入前後で「不動作」が14.1%から12.9%,「台形」が44.7%から12.9%,「そら豆(正答)」が41.2%から74.1%と変化した。両群共にMDD導入後で,不動作から正答に至る中間段階である「台形」の割合が減り,正答の割合が増えた。

しかし,2014年群の「不動作」は,MDD導入後も確認された。この「不動作」12.9%の内,導入前で「不動作」3.5%,「台形」8.2%,「そら豆」1.2%であった。すなわち,導入前には台形型コースが動作した解答が,導入後にはどのコースも動作しない解答となっていた。この結果の要因は,動作の確認不足ではないかと考える。これに該当する解答では,正答モデルと同じ状態に対して,誤りを含む遷移が記述されていた。このモデルでそら豆型コースを動作させることは不可能である。対象者らはそら豆型コースでの動作を確認しないまま提出したのではないだろうか。

課題3の達成段階毎の解答割合を,2013年群は図5に,2014年群は図6に示す。これらの図において,正答に近い解答として「荷物×そら豆」に着目すると,MDD導入前後において,2013年群では0%から9.0%に,2014年群では17.4%から11.8%に変化した。台形コースが動作した解答(「荷物×台形」と「荷物○台形」を合わせた解答)に着目すると,MDD導入前後において,2013年群では74.3%から51.3%に,2014年群では9.4%から5.9%に変化した。すなわち,MDD導入後には,不動作から正答に至る中間段階にあたる「荷物×台形」,「荷物○台形」,「荷物×そら豆」の割合が両群で減少していた。

しかし,2014年群の「不動作」は,課題2と同様にMDD導入後も確認された。これらの「不動作」15.9%の内,導入前で「不動作」が13.5%,「荷物×そら豆」が1.2%,「荷物○そら豆」が1.2%だった。すなわち,導入後の「不動作」の多くが,導入前にもどのコースも動作しなかった解答であった。これらの解答では,導入前後で図の形が変わっていないものが多かった。振舞いを確認させても,導入前に

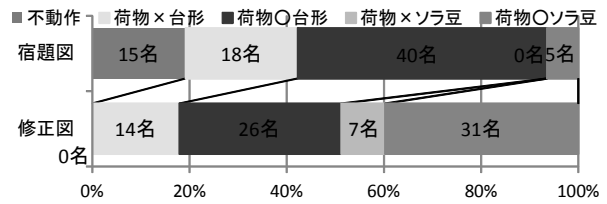


図5. 2013年群の課題3における達成段階ごとの解答割合

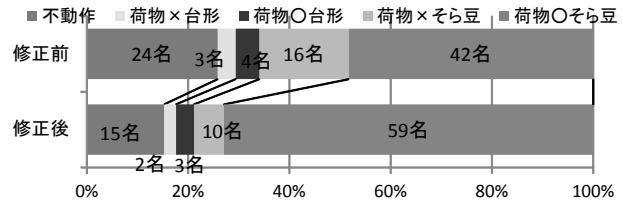


図6. 2014年群の課題3における達成段階ごとの解答割合

作成した図からどのように変更すれば正答に至るか分からなかったのではないかと考える。

5.1.2 リサーチクエスト1における考察

「MDDを導入することで,初学者が適切なモデルを記述できるようになるか?」というリサーチクエスト1に対して,MDD導入前後での,1)モデル品質の変化は,両群で課題2と3においてMDD導入後に正答率が上昇した事(4.1),2)達成段階は,導入後に中間段階の解答割合が減少していた事(5.1.1)から,MDD導入によって,初学者が適切なモデルを記述できるようになると考えられる。

5.2 誤りパターン

MDD導入前後の誤りパターンの変化について考察する。

5.2.1 誤りパターンの変化

課題3におけるMDD導入前後の7種の誤りパターン毎の発生割合を整理した。2013年群のクロス集計表を表2に2014年群を表3に示す。この表においては,横軸は導入前,縦軸は導入後の結果である。両軸には,表1に示した誤りパターン(3カテゴリ7種)に加え,これらの誤りを含まないモデル図(ミスなし)として,5.1.1で示した「台形」と「そら豆」の計9項目がある。この表は各解答における誤りの件数を集計したものである。一つの解答が複数の誤りを含む場合もあった。最右列と最下列は合計として各項目に該当する解答割合を示している。

これらの表の,MDD導入前後において,ミスなしの合計は,2013年群は50%から62.8%,2014年群は45.9%から62.4%に変化していた。MDD導入前後で誤りパターンを含まない解答割合がMDD導入後に増加していた。MDD導入後の誤りを含む解答割合については,3カテゴリ7種の内,5種で減っていた。

1解答あたりの平均誤り件数は,MDD導入前後において2013年群では2.5件から2.1件となった。2014年群では1.3件から1.2件となった。この結果から,2013年群に比べ,

表 2. 2013 年群の課題 3 の MDD 導入前後での誤りパターン変化

		修正図										
		記法ミス		冗長な表現		要求解釈ミス			ミスなし		合計	
		P-a	P-b	P-c	P-d	P-e	P-f	P-g	台	豆		
宿題図	記法ミス	P-a	28.2	-	1.3	-	-	1.3	5.1	3.8	6.4	48.7
		P-b	-	-	-	-	-	-	-	2.6	0.0	2.6
	冗長な表現	P-c	-	-	1.3	-	-	-	1.3	2.6	5.1	9.0
		P-d	1.3	-	-	2.6	-	-	-	2.6	1.3	3.8
	要求解釈ミス	P-e	3.8	-	-	1.3	7.7	3.8	1.3	0.0	5.1	16.7
		P-f	12.8	1.3	-	1.3	5.1	24.4	2.6	1.3	5.1	33.3
		P-g	5.1	-	1.3	1.3	-	-	3.8	2.6	1.3	10.3
	ミスなし	台	-	1.3	-	-	-	1.3	1.3	26.9	14.1	43.6
		豆	-	-	-	-	-	-	-	-	6.4	6.4
	合計		28.2	2.6	1.3	6.4	7.7	25.6	5.1	33.3	29.5	-

表 3. 2014 年群の課題 3 の MDD 導入前後での誤りパターン変化

		修正後										
		記法ミス		冗長な表現		要求解釈ミス			ミスなし		合計	
		P-a	P-b	P-c	P-d	P-e	P-f	P-g	台	豆		
修正前	記法ミス	P-a	4.7	1.2	-	-	2.4	1.2	-	-	4.7	11.8
		P-b	1.2	3.5	-	-	-	-	1.2	-	-	3.5
	冗長な表現	P-c	-	-	-	-	-	-	-	-	-	-
		P-d	-	-	-	-	-	-	-	-	-	-
	要求解釈ミス	P-e	1.2	-	-	-	20.0	2.4	1.2	-	2.4	22.4
		P-f	2.4	-	-	1.2	5.9	2.4	1.2	-	15.3	27.1
		P-g	-	1.2	-	-	1.2	-	1.2	-	1.2	3.5
	ミスなし	台	-	-	-	-	-	-	-	1.2	-	1.2
		豆	-	-	-	-	1.2	1.2	-	1.2	42.4	44.7
	合計		5.9	3.5	-	1.2	25.9	5.9	3.5	2.4	60.0	-

表 4. 2013 年群の課題 2 と 3 での MDD 導入後の達成段階

		課題3修正図					合計
		不動作	台形		そら豆		
			荷物×	荷物○	荷物×	荷物○	
課題2修正図	不動作	-	-	-	-	-	0.0
	台形	-	16.7	28.2	-	-	44.9
	そら豆	-	1.3	5.1	9.0	39.7	55.1
	合計	0.0	17.9	33.3	9.0	39.7	100.0

表 5. 2014 年群の課題 2 と 3 での MDD 導入後の達成段階

		課題3修正後					合計
		不動作	台形		そら豆		
			荷物×	荷物○	荷物×	荷物○	
課題2修正後	不動作	9.4	1.2	-	-	2.4	12.9
	台形	2.4	1.2	2.4	-	5.9	12.9
	そら豆	3.5	-	1.2	5.9	64.7	74.1
	合計	15.3	2.4	3.5	5.9	72.9	100.0

2014 年群の方が複合誤りを起こしているケースが少なかった。また、表中灰色部分は、MDD 導入前後で各誤りを修正できなかった割合を示している。導入前の発生割合に対して、最も修正されなかった誤りパターンは 2013 年群では P-f で、2014 年群では P-e であった。

確認された 7 種の誤りパターンはすべて、発生しても LEGO が動作する誤りである。特に、P-a から P-e までの 5 種は、達成段階では正答となっても、存在する可能性がある。誤りを含んで正答となった解答の比率は、2013 年群では全正答 42.3% 中 10% であり、2014 年群では 72.5% 中 13% であった。すなわち、2013 年群に比べ 2014 年群の方が正答となった解答におけるモデル品質が向上していた。

5.2.2 リサーチクエスト 2 における考察

「MDD を導入することで、自分で誤りを修正できるか？」というリサーチクエストに対して、MDD 導入前後で、1) 導入後で誤りパターンを含まない解答が両群で増加している事、2) 導入後に 7 種の誤りパターンの内 5 種で発生割合が減少している事、3) 導入後に平均誤り件数が減少している事が指摘できる。また、同時に、4) 導入後においても修正されなかった誤り（2013 年群では P-a と P-f、2014 年群では P-e）がある事、5) 導入後に発生割合が増えた誤りパターン（2013 年群では P-b と P-d、2014 年群では P-d と P-e）がある事から、MDD を導入する事で、誤りを含む解答割合

合と平均誤り件数は減少するが、自分で修正できない誤りパターンは依然として存在すると考えられる。

これまでに確認されている誤りパターンは、モデル図の品質に関する誤りであり、LEGO の動作を不動作にさせる誤りではない。そのため、対象の動作を確認させてモデル図を評価させるために MDD を導入しても、振舞いだけを見て誤りに気づくことは難しいのではないだろうか。課題の内容と要求を把握したうえで、振舞いを観察するように学習プロセスを設計する必要がある。

5.3 MDD 導入方法の違い

MDD の導入について、3.4 に示した通り、2013 年群は一括 MDD、2014 年群は逐次 MDD をそれぞれ用いた。手法を変えた要因は、2013 年群の MDD 導入後の課題 2 と 3 において、ライントレースの動作コース毎の達成段階で整理した結果、課題 2 から 3 にかけて解答の達成段階が良好化した解答がなかったからである。

5.3.1 課題 2 と課題 3 の達成段階の相関関係

MDD 導入後の課題 2 と 3 の達成段階の相関について、2013 年群の結果を表 4 に、2014 年群の結果を表 5 に示す。横軸は課題 2 の、縦軸は課題 3 の達成段階を示す。いずれも、「不動作」、「台形」、「そら豆」の 3 種の達成段階を示している。課題 3 についてはさらに、荷物搬送の達成段階も示してある。表中の各セルには、発生件数割合を示す。最

右列と最下列には、各行列の合計値を示す。表中網掛けした6つのセルには、課題2から3にかけて解答の達成段階が良好した解答割合を示す。

両表から、課題2と3両方で正答となっている割合は、2013年群39.7%に比べて2014年群64.7%と多くなった。また、中間段階の割合が2013年群44.9%に比べて2014年群3.6%と大幅に減少した。この結果から、2014年群では正答率と達成段階が2013年群に比べて向上している。表中網掛け部分に注目すると、2013年群では該当する解答がないのに対して、2014年群では3つのセルで解答が存在し、約10%の解答が良好された。

5.3.2 課題2と課題3の誤りの相関関係

課題2と3の導入後の図において、誤りパターンがどのように推移したかを考察する。この結果を分析する意味としては、課題2で修正できなかった誤りは、課題3でも修正できないのかどうかを検討することにある。

今回の結果として、課題2のMDD導入後に発生した誤りは、2013年群ではP-a、P-c、P-eであり、2014年群ではP-a、P-eであった。課題3では、2013年群では7種全て、2014年群ではP-c以外の全てであった。課題3の誤りの発生率を両群で比較すると、2013年群に比べ、2014年群は全ての項目で発生率が少なかった。すなわち、2014年群においては発生した誤りパタンの数も発生率も減少していた。

また、課題2と3の両群で共通に発生した謝りパターンはP-aとP-eであった。さらに、課題2あるいは課題3の発生割合が20%を越える誤りパターンは、2013年群でP-aとP-e、2014年群はP-eであった。ここで、P-aを調べると、2013年群では発生割合が多い上に修正ができないのに対して、2014年群では発生率が少ない。P-eを調べると、2013年群では課題3のみで発生率が多いのに対し、2014年群では、課題2と3で発生率が多い上に修正ができない。

5.3.3 リサーチクエスト3について

「一括MDDと逐次MDDにおいて、結果に差があるか？」というリサーチクエスト3に対して、達成段階(3-1)と誤りパターン(3-2)の2つの観点から述べる。

3-1については、1) 2014年群の方が正答率と達成段階は高い事(5.3.1)、2) 課題2と3の達成段階が良好した解答が存在した事(5.3.1)から、一括MDDよりも逐次MDDの方が正答率と達成段階は向上した。一方で、3) 2014年群では課題2と3において、導入後に「不動作」が存在する事(5.1.1)も明らかになった。

3-2については、4) 2014年群の正答となった解答の方がモデル品質は高い事(5.2.1)、5) 2014年群においては発生した誤りパタンの数も発生率も減少している事(5.3.2)、6) 2014年群においては発生率が多く修正できない誤りパタンの数が減少した事(5.3.2)から、一括MDDよりも逐次MDDの方が誤りパタンの数も発生率も減少する傾向が確認できた。一方、7) 2013年群と2014年群では、発生率

が多く修正できない誤りが異なる事(5.3.2)から、逐次MDDでも修正できない誤りが存在することが分かった。

2013年群と2014年群で、発生率が多く修正できない誤りパターンが異なる要因の一つに、MDDの導入方法の違いがあるのではないかと考える。2013年群は一括MDDであったため、MDD中に課題内容を熟考する機会が少なく、モデル図とLEGOの動作の対応を図ることが学習活動の中心であったのではないかと考える。一方、2014年群は逐次MDDであり、課題毎に机上モデリングとMDDを行うことで、課題内容を熟考する機会がより多く得られていたのではないかと考える。逐次MDDの方法は、机上で作成した後、LEGOの動作確認に入ったが、机上で作成する時間に差があり、実際に他人が動かしていたLEGOの動きを見て、それを参考にした可能性がある。

この改善策として、5.2.2と同様に、学習者に誤りを気付かせるよう、モデルエディタに誤りを自動検知する機能を実装する事があげられる。

6. おわりに

本稿では、大学1年生を対象とした概念モデリング教育におけるMDD導入前後のモデル品質の差と、誤りパタンの差について考察した。その結果、MDDを導入する事で、モデル図の品質は向上し、学習者自身が誤りを修正できるという結果が得られ、また、一括MDDよりも逐次MDDの方が、正答率と達成段階が向上し、誤りパタンの発生数と発生率を抑えることを確認した。しかし、同時に発生率が多く修正ができないパターンも存在していた。今後は、学習者が状態遷移図に関してより理解を深めることを目標として、モデルエディタに学習支援機能を付加していく。

参考文献

- 1) European Commission: "Emerging Skills and Competences- A transatlantic study: Final Report", http://ec.europa.eu/education/more-information/doc/2011/skills_en.pdf (accessed 2014/09/16).
- 2) Assessment and teaching of 21st century skills: "ACT21 home page", <http://atc21s.org/> (accessed 2014/09/16).
- 3) J. Kramer: "Is Abstraction The Key To Computing?", Vol.50, No.4, pp.37-42, CACM (2007).
- 4) 赤山聖子他: "オブジェクト指向モデリング教育におけるモデル駆動開発ツールの活用方法の検討", 情報処理学会論文集, vol.55, pp.72-84(2014)
- 5) 横田寛明他: "初学者によるロボット動作設計のための状態遷移図記述におけるモデル駆動開発方法論導入の効果", 情報システム教育学会第4回研究会, pp.7-12 (2013)
- 6) 情報処理推進機構: "モデルベース設計検証技術者スキル体系化調査報告書", https://www.ipa.go.jp/sec/softwareengineering/reports/20120229_2.html (accessed 2014/9/7)
- 7) J,Bezivintal et. al.,: "TeachingModeling: Why, When, What?", pp.55-62, MODELS 2009 (2009).
- 8) 信州大学: "信州大学 clooca サイト", <http://mdd.shinshu-u.ac.jp/> (accessed2014/9/16)