

モデル駆動開発方法論を用いた状態遷移図記述による ロボット動作設計の評価 — 大学1年生と2年生の比較より —

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

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を観察することによるモデル評価を導入している。本稿では、モデル駆動開発方法論導入前後のモデル品質の差と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大学1年生と2年生の結果比較に基づき考察する。

Evaluation of Simple State Machine Diagrams in Behavioral Modeling Education with Model-Driven Development Method - comparison the novice group with the programming-known group -

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

The purpose of this study is to explore the educational methodology about behavioral modeling for beginners. We have been introducing the model driven development method in our modeling education for university freshmen and sophomore. An model editor and a model compiler based on a meta model defined by teacher have used in these courses. In this paper, we evaluate the educational effectiveness of the MDD method for modeling education with state machine diagram.

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IPJS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、これらの能力育成に関する初学者向けの教育方法論は未だに確立されておらず[4,5]、国内外において長年研究が進められている[6-11]。しかし、現在までに状態遷移図の作成における初学者の誤り分析の例はない。

机上で作成された状態遷移図を評価するには、学習者自身による確認や検証を行うこととなる。これまでに机上で状態遷移図を評価するための方法論がいくつか提案されている[12,13]。しかし、それらを初学者に利活用させることは一般には難しい。

そこで、本研究では、モデル駆動開発の方法論を初学者教育に導入し、学習者自身が対象物の振舞を観察することによりモデルを評価する活動を行わせることとした。本稿では、MDD導入前後のモデル図品質の差、およびモデルに含まれる誤りパタンの分析に基づくモデル駆動開発方法論導入の効果について、プログラミングに関する既習知識

が異なる2つの被験者群の差異に基づき考察する。

2. 本研究のアプローチ

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[4]。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の設計し、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。さらに、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、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、すなわち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。

また、MDDにおけるDSL(Domain-Specific Language)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[14]。DSLを用いることで、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、思考対象となる空間を狭めることと同義であり、課題の難易度を制御する一要素となりえる。

^{†1} 信州大学工学部
Shinshu University, Faculty of Engineering

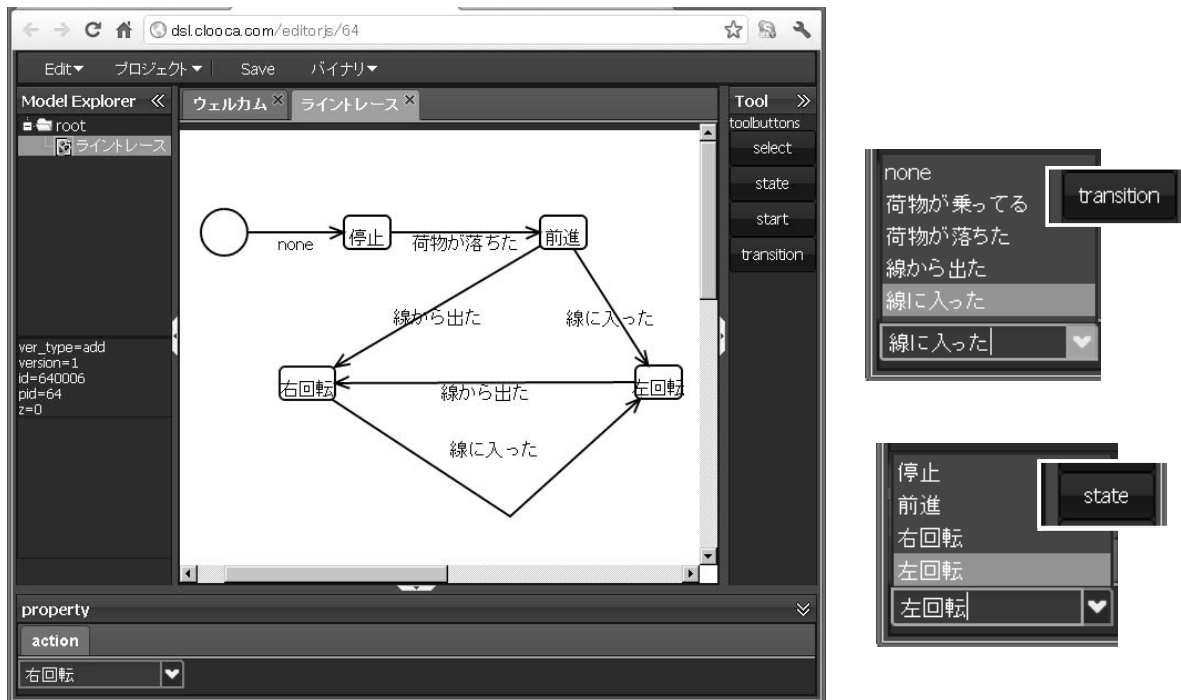


図 1. MDD モデルエディタ・コンパイラ(左)と、当該モデル記述用 DSL の例(右上：遷移の語彙, 右下：状態の語彙)

Fig.1 An example of the model edit & compiler for MDD(left),
 and examples of DSL defined for the meta model (left-upper : transition, left-lower : state).

本研究では、本来は学習者自らが成すべき作業を、MDD 環境の導入と DSL の利用に置き換え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[14]。図 1 に本研究で用いているモデルエディタ・コンパイラの画面例と、当該モデル記述用の DSL の語彙例を示す。a. モデル図として記述する要素、および要素に与える語彙集合としての DSL は、モデルエディタ上でメタモデルを定義することで、自由に設定できる。

3. 実験方法

3.1 被験者

被験者 13T 群は大学の情報工学科 1 年生 78 名とした。実施時期は 2013 年 4 月下旬から 5 月上旬にかけてである。11T 群の被験者は大学入学直後であり、情報工学に関するいかなる専門的な授業を受けていない。

一方、被験者 12T 群は大学同学科 2 年生 55 名とした。実施時期は 2013 年 11 月上旬から 11 月中旬にかけてである。12T 群の被験者は大学 2 年生の後期課程であり、プログラミングを 2 学期(1 年後期と 2 年前期, 各 3 単位)の講義・演習を受講済みである。また、1 年前期に本実験の課題を一度経験している。

3.2 状態遷移図の記法

本研究で導入する状態遷移図の書式を図 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

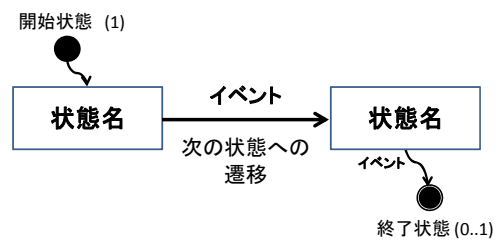


図 2. 本研究で用いた状態遷移図の記法
 Fig.2 Notation of a simple state machine diagram

述要素を状態および状態名、遷移およびイベント名のみに制限し、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 NXT V2 で動作するサービスの設計とした。本稿では、両群の被験者が共に解答した「荷物搬送・ライトレース課題」を取り上げる。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(荷物搬送)と、黒い線を

a 遷移に対する語彙のうち「none」は初期状態から生じる遷移に対して与える特殊な語彙である。当該モデルエディタの仕様。

トレースしながら前進していく振舞(ライントレース)と同時に満たすことが求められる。全体として、DSL としては、4種のイベントと4種の状態(図1右部)を組み合わせ、サービスを設計することとなる。

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「荷物が乗っている」、OFFとなったイベントを「荷物が落ちた」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。

ライントレースに際しては、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。

3.4 実験手順

13T群の被験者には、本実験の前に3.2に示した書式での状態遷移図の読み方および書き方に関して十分慣れさせた。その上で、3.3の課題を提示し、モデルエディタ・コンパイラ(Group Clooca[16,17])上でモデル図記述のみを行うように指示した。解答には各自にとって十分な時間を割り当てさせるために、特に時間制限を設けなかった。このモデル図は、実験前に静止画ファイルとして指定場所に提出させた(以下、「宿題図」と称する)。本実験ではまず、本研究で開発したコンパイルサーバにより、モデルエディタ上の宿題図からLEGO上での実行コードに変換させた。その結果をLEGOに転送させた上で、モデル図に基づくLEGOの振舞を観察させた。振舞に不備がある場合には、被験者が課題を解決したと判断するまでモデル図の修正・デバイスの観察を繰り返させた。実験終了時には、課題に対する最終成果となるモデル図を提出させた(以下「修正図」と称する)。

12T群の被験者には、13T群と同様の手順を提示した上で、そら豆型コースを踏破するモデルを記述するという課題達成度を満たすような修正を行わせて上でモデル図を提出させた(これが12T群における「修正図」となる)。授業内で正しくモデルが修正できなかった者は、授業時間外にロボットを貸し出し、修正の機会を与えた。

3.5 モデル図の評価方法

課題に対する評価観点は荷物搬送とライントレースの2つの側面から次のように定めた。荷物搬送としては、提示した搬送要求を満たしたか否かで評価した(「荷物○」、「荷物×」)。ライントレースとしては、踏破したコースの形状で評価した。ここで対象とするコースは台形型とそら豆型の2種類である(図3参照)。台形型は右旋回あるいは左旋回といった片側旋廻のみで踏破可能である。一方、そら豆型は右旋回と左旋回の両方を併用しないと踏破できない。そら豆型が踏破できると、台形型踏破可能なため、コースの動作の評価結果は「不動作」、「台形まで動作(以下、台形



図3. ライントレース用のコース(左:台形型, 右:そら豆型)
 Fig.3 Outline of courses for the line trace & messenger problem.
 (left : trapezoid-type, right : bean-type)

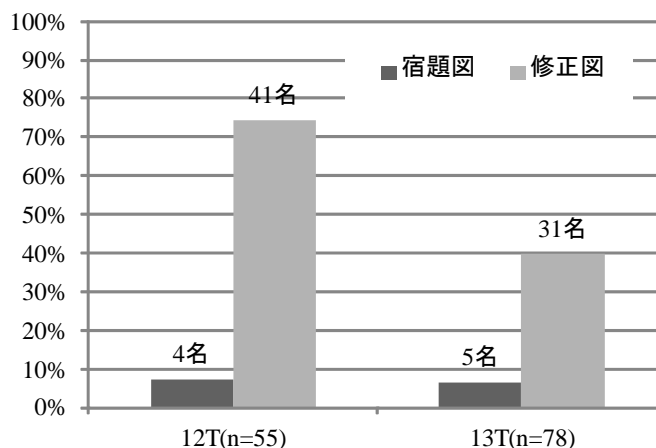


図4. 宿題図と修正図の正答率

Fig.4 Percentages of numbers answered correctly for the problem (dark : pre, pale : after).

と略す)」、「そら豆が動作(以下、そら豆と略す)」の3パターンに分類した。

4. 実験結果

ここでは、3.5に示した評価規準に基づき、宿題図と修正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エラーを示す。

4.1 モデル図の品質の変化

まず、課題における宿題図と修正図における正答率の差を概観し、さらに評価規準に基づく詳細分析の結果を示す。

4.1.1 正答率の比較

ここでの正答とは、「そら豆型コース上でのライントレースを行いつつ、荷物搬送を行うことができるモデル図(「荷物○」かつ「そら豆」)が作成されていた」回答とする。図4に課題における宿題図と修正図の正答率を示す。

12T群における正答は、宿題図で7%、修正図で74%と大幅に改善した。13T群でも宿題図で6%、修正図で38%と改善がみられる。いずれの群もMDD導入によるモデル図の品質の向上が確認できた。特に、12T群に対してのみ、そら豆型コースを踏破するという要求を満たすよう指示した上でモデル図を記述させていた。MDD導入によるモデル図品質の良化が13T群と比較して、顕著であった。

表 1. 宿題図と修正図の達成度合の比較

Table 1 Level of achievement for answers of the 12T group(a) and the 13T group(b)

(a) 12T 群

(b) 13T 群

		修正図					合計	
		不動作	台形		そら豆			
			荷物×	荷物○	荷物×	荷物○		
宿題図	不動作	1	-	-	-	5	6	
	台形	荷物×	2	-	1	3	16	22
		荷物○	4	-	1	2	17	24
	そら豆	荷物×	-	-	-	-	-	0
		荷物○	-	-	-	-	4	4
合計		7	0	2	5	41	55	

		問3修正図					合計	
		不動作	台形		そら豆			
			荷物×	荷物○	荷物×	荷物○		
問3宿題図	不動作	-	2	4	1	8	15	
	台形	荷物×	-	11	-	4	3	18
		荷物○	-	1	22	2	15	40
	そら豆	荷物×	-	-	-	-	-	0
		荷物○	-	-	-	-	5	5
合計		0	14	26	7	31	78	

4.1.2 要求に対する達成度合の比較

回答を 3.5 に示した評価規準で分類した結果を表 1 に示す。(a)が 12T 群, (b)が 13T 群である。いずれの表においても、縦方向に宿題図の達成度合を、横軸方向に修正図の達成度合を取り、宿題図と修正図における達成度合の変化を該当人数として示した。

表 1(a)の 12T 群においては、荷物搬送をしながら、台形型あるいはそら豆型のいずれかのコースを踏破できた被験者の割合が、全体の 51%から 76%まで増加している。また、踏破コースの観点からは、宿題図でそら豆型が踏破できない(「不動作」あるいは「台形」)ものが 51 回答(全回答の 93%)に対して、その 72%にあたる 37 回答が修正図では「そら豆」に修正されていた。修正図における不動作の回答が、修正図よりも増加した。しかし、荷物搬送の要求が満たせなく、「台形」に相当する回答は、22 から 0 となったことなど、他の達成基準を満たす回答の改善状況を考慮すると、モデル修正時間が十分に取れていなかった可能性が示唆される。

一方、表 1(b)の 13T 群においては、「不動作」が宿題図で 19%であったが、修正図では 0%となった。すなわち、全被験者が台形型かそら豆型のいずれかのコースを踏破できるよう修正できていた。さらに、「荷物○」かつ「台形」あるいは「そら豆」の割合が、全体の 58%から 73%に増加している。また、踏破コースの観点からは、宿題図でそら豆型が踏破できない(「不動作」あるいは「台形まで動作」)ものが 73 回答(全体の 94%)に対して、その 45%にあたる 33 回答が修正図では「そら豆」に修正されていた。

両群において、荷物搬送の観点からも、踏破コースの観点からも、修正図のモデル品質の改善が確認できた。

4.2 誤りパタンの分析

4.2.1 誤りのカテゴリと誤りパターン

これまでの分析から、被験者のモデル図に含まれる誤り

パターンは、課題に非依存なパターンと、課題に依存するパターンに分類できることがわかっている。

課題に依存しないパターンは、具体的な誤り記述のパタンの抽象化が可能である。確認されている典型的なモデルパターンを図 5 に示す。このパターンはさらに「記法ミス」と「冗長な表現」の 2つのカテゴリに分かれる。

「記法ミス」カテゴリには 3.2 に示した記法および制約に沿っていない回答が相当する。このカテゴリはさらに、「同じ遷移名」(図 2 中 P-a)、「同じ状態」(P-b)、「イベント名未記入」(P-h)に分類される。

「冗長な表現」は、誤りではないがモデル図をさらに簡略化できる回答が相当する。このカテゴリはさらに、「自己遷移」(P-c)と「同遷移複数」(P-d)と分類される。

図 6 に課題に依存する誤りパターンを示す。このパターンは、「要求解釈ミス」カテゴリのみである。これは課題に示された要求を満たしていない回答が相当する。誤りの種類として、「状態通過」(図 3 中 P-e)、「止まらない」(P-f)、「勝手に動作する」(P-g)の 3つが確認されている。

図 5 および図 6 に示した誤りを含むモデル図は、今回用いたモデルエディタ・コンパイラ環境では、要求に沿うかのような振舞として観察されるが、モデル図の品質を低下させる要因となることから、本研究では誤りとして取り上げている。

4.2.2 誤りパターン毎の発生率

図 7 に誤りパターン毎の発生率を示す。(a)が 12T 群, (b)が 13T 群の結果である。

図 7(a)の 12T 群における「ミスなし」の回答は、宿題図で 29%、修正図で 58%と増加した。その他、3種の誤りカテゴリ全てにおいて、宿題図と比較して修正図での発生率が低下した。発生率が最も多かったのは「要求解釈のミス」、次は「記法ミス」であった。共に宿題図での発生率は約 50%であった。また、「冗長な表現」発生率が少なかった。

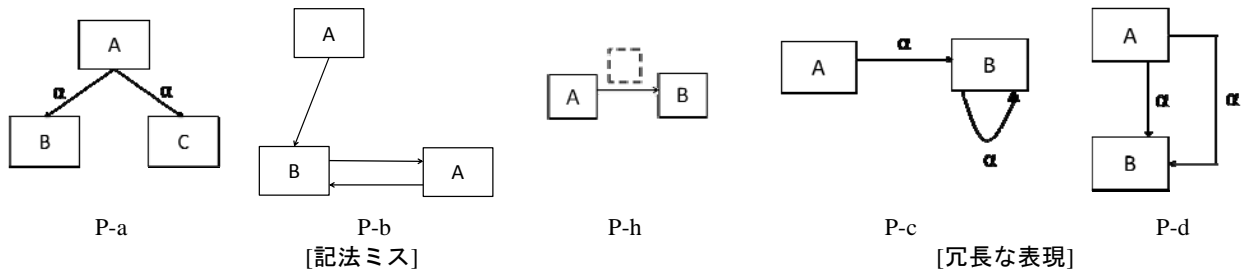


図 5. 課題に依存しない誤りパターン

Fig.5 Typical error patterns which are independent for the problem (or requirement) .

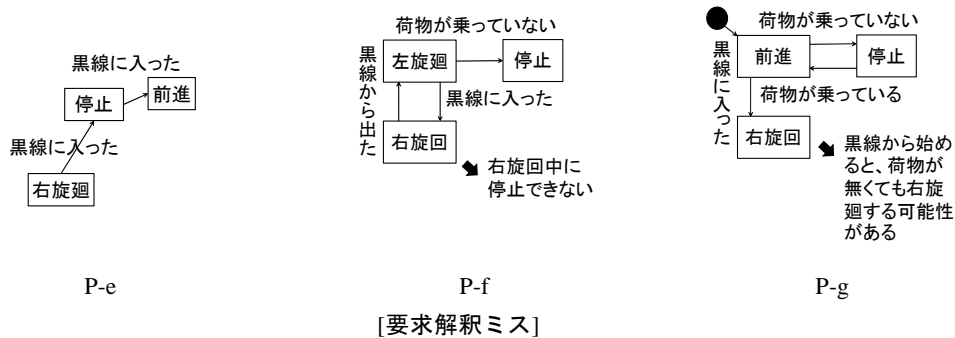


図 6. 課題に依存する誤りパターン

Fig.6 Typical error patterns which are dependent for the problem (or requirement) .

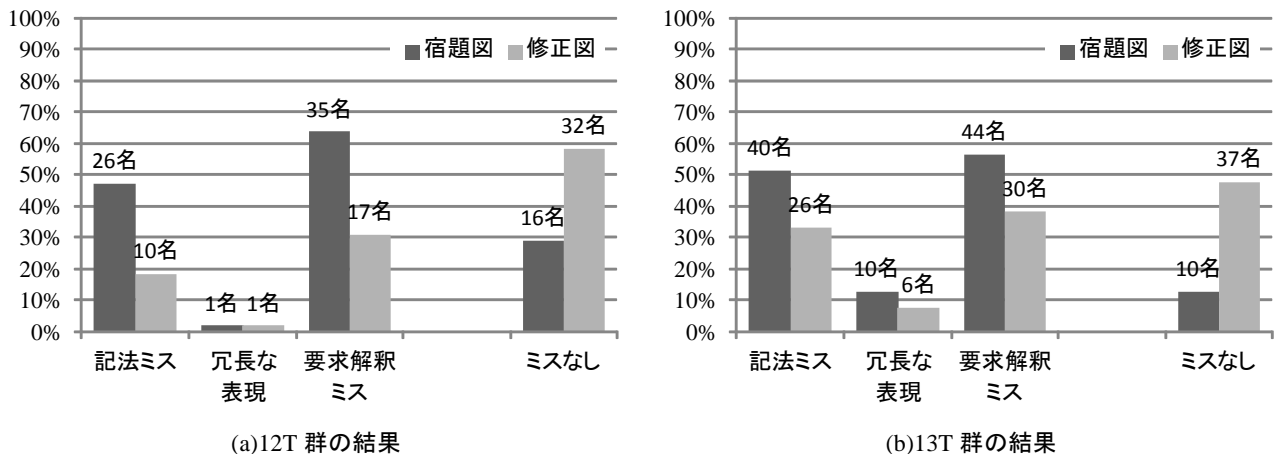


図 7. 誤りパターン毎の発生率

Fig.7 Rate of occurrence of three error patterns and the answers which have no error pattern.

13T 群の結果は、12T 群と同様の傾向を示した。宿題図に比べて修正図での「ミスなし」は顕著に増加している。「要求解釈ミス」の発生率が高く、「記法ミス」がそれに続く。「冗長な表現」は3種の誤りカテゴリ内では最も低い発生率であるが、12T 群と比較すると、宿題図においては約6倍の割合で生じていた。

5. 考察

4章の結果を、以下の2種のリサーチクエスチョンに基づき考察する。

1: MDD 手法を導入することで、被験者が記述するモデルの品質は向上するのか

2: MDD 手法を導入することで、被験者はモデル図の誤りを自ら修正できるか。もし全ての誤りを修正できないのであれば、修正しやすい誤りは何で、修正が難しい誤りは何か。

5.1 リサーチクエスチョン1について

宿題図と修正図の正答率および達成度合の比較結果から、前提知識の異なる12T群と13T群の両群において、MDD手法を導入することで、被験者が記述するモデルの品質は向上したといえる。

今回得られた結果を総括すると、宿題図に含まれていた誤りを、モデル図を動作させ、その振舞を観察することにより、客観的な評価が行えるようになった。さらに、その

結果に基づき、モデル図を修正させることで、要求に対するモデル図品質を向上させることにつながったと考える。

5.2 リサーチクエスト2について

誤りパタンの分析結果から、MDD 手法を導入しても、修正しやすい誤りと、修正しにくい誤りがあるといえる。

12T 群・13T 群に共通して、比較的修正しやすい誤りは「冗長な表現」、特に「自己遷移」(P-c)であった。一方、修正されにくい誤りは「記法誤り」と「要求解釈のミス」であった。

修正されにくい誤りの原因は、1)モデルエディタ・コンパイラの性能と、2)動作のテストケースの不足の2点だと考えられる。

1)の原因：教授者側が不適なモデル図だと判断する回答であっても、モデル駆動開発のために導入するモデルエディタやモデルコンパイラの性能により、デバイスの振舞としては要求を満たしているように観察できる場合がある。デバイスの振舞を観察することだけでモデル図の妥当性を判断しようとする場合、明らかな「記法ミス」は被験者に気付かれない場合が多いことが分かった。

2)の原因：デバイスの振舞を観察する際、一部の要求部分だけをテストしており、全体の要求を満たせなくても動作したと勘違いするケースがある。これにより、正しく動作したと錯覚し、「要求解釈ミス」の誘発につながったと考える。

6. おわりに

本稿では、大学1年生(13T群)と2年生(12T群)を対象としてロボット動作設計における状態遷移図のモデル品質について考察した。ここでは、モデリング教育においてMDDの方法論としてモデルエディタ・コンパイラの導入とDSLによる語彙制御を導入し、それら導入前後のモデル図品質の差、およびモデルに含まれる誤りパタンの分析を試みた。

その結果、初学者である13T群に導入した際と同様に、プログラミング既習者である12T群においても、MDD手法を導入することで、被験者が記述するモデルの品質が向上することが定量的に確認された。

また、MDD手法を導入することでも、被験者はモデル図の誤りを自ら全て修正することができないこと、モデル図に含まれる誤りには主として3つのカテゴリがあり、それぞれ複数の誤りパターンが存在すること、そして誤りパターンには対象物の振舞を観察することで学習者が自ら修正しやすいものと、修正しにくいものがあることを示した。

今後は、モデリング教育に特化したMDDツールの設計および実現と、DSLの導入により段階的にモデル図の難易度を制御する教育方法論の具体化を図るとともに、他の情報系の知識が、この学習にどういった影響を与えるのか考察をし、より良い教育手法の開発を目指す。

謝辞 本研究はJSPS 科研費 22300286 の助成を受けた。

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 2013/09/20).
- 2) Assessment and teaching of 21st century skills : ACT21 home page, <http://atc21s.org/> (accessed 2013/09/20).
- 3) J. Kramer : “Is Abstraction The Key To Computing ? ” , Vol.50, No.4, pp.37-42, CACM (2007).
- 4) J.Bezivintal et. al., : “Teaching Modeling : Why, When, What?” , pp.55-62, MODELS 2009 (2009).
- 5) IPA : “モデルベース設計検証技術者スキル体系化調査 調査報告書” (2012).
- 6) 中尾信明 : “オブジェクト指向, UMLに関する教育の視点と分析” , 情処研報, 2004-CE-74(2), pp.9-16 (2004).
- 7) Institutionen för datavetenskap - Computer Science Education Group : Workshops on OO Education, <http://www8.cs.umu.se/research/education/ooEduWS.html>
- 8) J. Niere et. al., : “Thinking in Object Structures: Teaching Modeling in Secondary Schools” , The 6th ECOOP Workshop on Pedagogies and Tools for Learning Object-Oriented Concepts (2002), <http://www.uni-paderborn.de/cs/ag-schaefer/Veroeffentlichungen/Quellen/Papers/2002/PTLOOC2002.pdf>
- 9) H.C. Cham et. al., : “An evaluation of Novice End-User Computing Performance : Data Modeling, Query Writing and Comprehension” , J. of the American Society for Information Science and Technology, Vol.56, No.8, pp.843-853 (2005).
- 10) 長尾祐樹他 : “初心者用 UML の提案とその評価” , 情処研報, 2008-CE-97(7), pp.45-52 (2008).
- 11) J. Niere et. al., : “Avoiding anecdotal evidence : An experience report about evaluating an object-oriented modeling course” , MoDELS/UML 2005 Educator's Symposium, pp.63-70 (2005).
- 12) 森広芳文他 : “シミュレーションを利用した形式的検証システム” , 信学会論文誌 D-1, J84-D-1(4), pp.367-377 (2001).
- 13) 辻孝吉他 : “ペトリネットによる通信プロトコル状態遷移表のモデル化と動作検証” , 電気学会論文誌 C, 115(5), pp.713-718 (1995).
- 14) 三ツ井欽一他 : “モデリングとツールを駆使したこれからのソフトウェア開発技法-モデル駆動開発手法を中心として-” , 情報処理, 45(1), pp.1-33 (2004).
- 15) 香山瑞恵他 : “状態遷移図作成に際する初学者の誤り分析とそれに基づく教育方法の検討” , 情処研報, 2012-CE-117(7), pp.1-9 (2012).
- 16) 香山瑞恵他 : “ロボットでの動作シミュレーションが可能なシステムモデリング用 Web 教材” , 第5回日本情報科教育学会全国大会講演論文集, p.54 (2012).
- 17) Clooca : <http://www.clooca.com/>, 株式会社 Technical Rockstars.