

モデリング教育支援環境における コンパイルサーバの機能と問題点の整理

大宅 剛生^{*1}, 香山 瑞恵^{*1}, 永井 孝^{*2}
信州大学大学院^{*1}, ものづくり大学^{*2}

Functions and Problems of the Compile Server in Modeling Education Support Environment

Kouki Ootaku^{*1}, Mizue Kayama^{*1}, Takashi Nagai^{*2}

^{*1} Graduate School of Science & Technology, Shinshu University

^{*2} Institute of Technologists

あらまし:本研究の目的は,モデリング教育支援環境の改良である. このモデリング教育支援環境はモデルエディタ, モデルコンパイラ, ソースコンパイラの 3 種のツールからなる. ここではソースコンパイラであるコンパイルサーバを対象とした改良を検討する. 中学校技術科正規授業での運用を通して, コンパイルサーバの問題点を整理した. 本稿では本支援環境の概要を述べ, コンパイルサーバの機能を示す. そのうえで, 授業運用を通じたコンパイルサーバの問題点をまとめ, 改良方針を示す.

キーワード: コンパイルサーバ, モデル駆動開発, モデリング

1. はじめに

近年, 問題を効果的に解決するための方法として, モデル化の能力の育成が重要視されている. モデル化の能力を習得することで「考える力」や「捉える力」, 「表現する力」を養成できるといわれている⁽¹⁾. そのため, 大学などの高等教育機関の情報専門学科のカリキュラム標準⁽²⁾にモデル化の能力を育成するための学習項目が組み込まれている. この傾向は, 義務教育においても同様である. 中学校技術科の学習指導要領解説⁽³⁾では, 「D 情報の技術」における計測・制御のプログラミングによる問題を解決する学習活動として, 栽培ロボットや生活サポートロボットのモデルを開発することが例示されている.

モデル化の能力の育成に当たり, モデル駆動開発 (Model Driven Development. 以下, MDD) ツールを用いた授業が実践されている⁽³⁻⁵⁾. 一般に, MDD ツールは商用のソフトウェアが多く, 技術者向けの機能が充実している. それに応じて, モデリング教育向けの MDD ツールの機能拡張も必要となる. そこで, 我々

は MDD 機能を有するモデリング教育支援環境を構築し, 運用してきた⁽⁵⁻¹¹⁾.

本研究では, モデリング教育支援環境のうちコンパイルサーバに焦点を当て, 教育向け機能の改良を行ってきた. 今回, これまでの運用を通して, コンパイルサーバの問題点を整理した. 本稿では本支援環境の概要を述べ, コンパイルサーバの機能を示す. そのうえで, 授業運用を通じたコンパイルサーバの問題点をまとめ, 改良方針を示す.

2. MDD とモデリング教育支援環境

2.1 MDD

MDD は, 「システムのモデルを作成し, それを実際のものに変換することが可能である (we can construct a model of a system that we can then transform into the real thing.)」という考えに基づくソフトウェア開発方法論である⁽¹²⁾. 既に, 産業界において, MDD はモデルからプログラムを自動生成する技術として実用化されている⁽¹³⁾.

一般的な MDD ツールは、モデルエディタとモデルコンパイラから構成される。モデルエディタは利用者がモデルを記述するツールである。モデルコンパイラは、利用者が記述したモデルからソースコードを生成するツールである。

2.2 モデリング教育支援環境

我々は MDD 機能を有するモデリング教育支援環境を運用してきた⁽⁵⁻¹¹⁾。具体的には、2012 年からは大学教育に適用し、2018 年からは中学校技術科の正規授業に適用している。この環境は、クラウド型 MDD ツールである clooca⁽¹⁴⁾と、原らの研究⁽¹¹⁾により開発されたコンパイルサーバを統合したものである。

本支援環境の概要を図 1 に示す。この環境は、MDD

ツールとコンパイルサーバから成る。このうち、MDD ツールには astah⁽¹⁵⁾, clooca, SRPS⁽¹⁶⁾を用いている。本研究ではこれらの MDD ツールに対して、モデル図の記法誤りを自動チェックする機能を具備した^(7,9,17)。コンパイルサーバは本研究で独自開発したものであり、3つのアプリケーション：front, chief, builder からなる。コンパイルサーバは、モデルコンパイラで生成されたソースコードをコンパイルし、対象デバイスに応じた実行コードに変換する。対象デバイスごとのコンパイル環境は Docker コンテナ化されている。コンパイルサーバから実行コードが学習者の端末にダウンロードされる。学習者は実行コードを対象デバイスに転送・実行することで、モデルの妥当性を判断する。コンパイルサーバについては 3 章で詳述する。

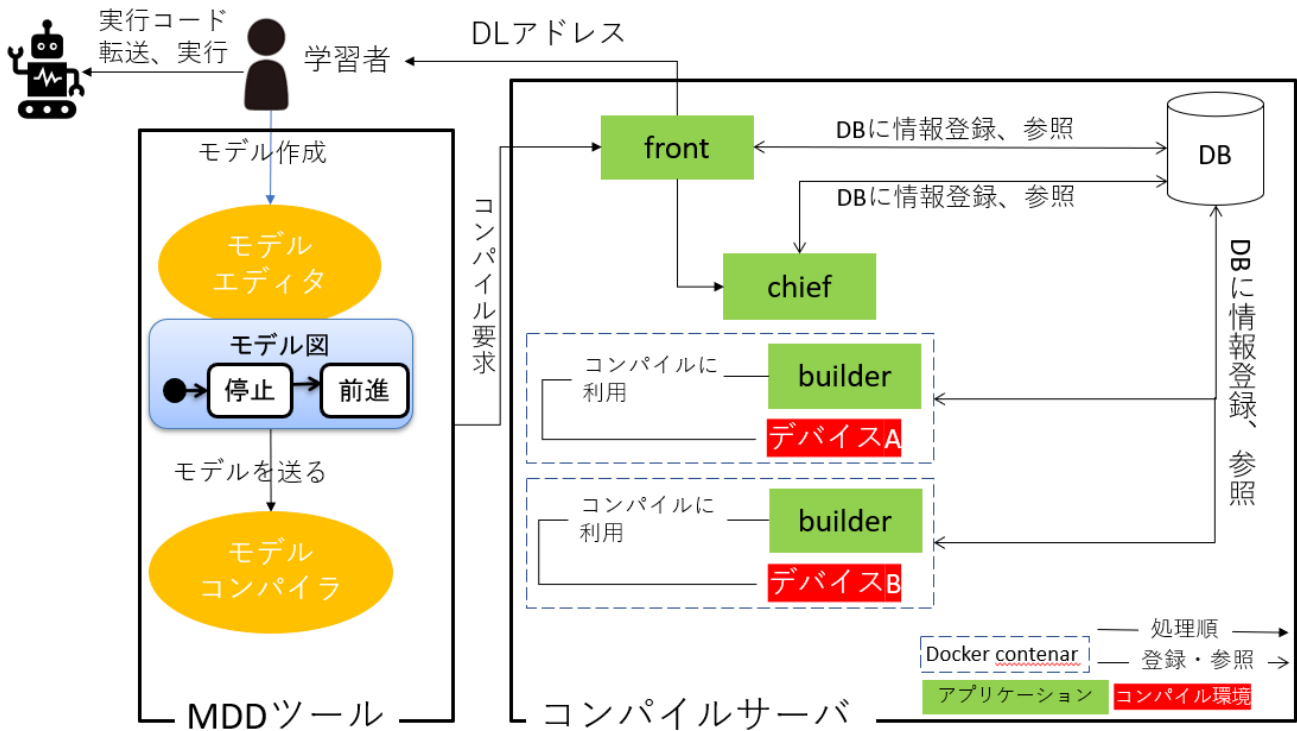


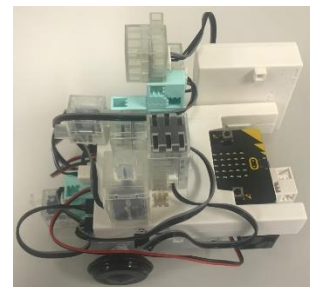
図 1 モデリング環境概要図



(a) Lego Mindstorm NXT



(b) Studuino



(c) micro:bit

図 2 対象デバイス

2.3 対象デバイス

本支援環境での対象デバイスは Lego Mindstorm NXT (図 2(a), 32bit ARM7 デバイス), Studuino (図 2(b), AVR デバイス), micro:bit (図 2(c), ARM coretex M0 デバイス)の 3 種類である。これらのデバイスで使用可能なセンサは、赤外線センサ、タッチセンサ、光センサ、音センサ等である。アクチュエータは、電子ブザー、各色 LED, DC モータ等が利用できる。

3. コンパイルサーバの機能と動作

本章では、コンパイルサーバの機能と動作を述べる。MDD ツールとコンパイルサーバ(各アプリケーションと DB)の動作をシーケンス図として図 3 に示す。

3.1 コンパイルサーバの概要

本コンパイルサーバは 3 つのアプリケーション : front, chief, builder からなる。front は MDD ツールからのコンパイル要求を受け付ける。chief はコンパイル処理を負荷分散する。builder はコンパイルを実行する。3 つのアプリケーションの情報は専用 DB で管理される。これらのアプリケーションが独立に機能することでコンパイル処理を効率化する。

3.2 front

このアプリケーションの機能は主として 2 つである。1 つ目は、MDD ツールからコンパイル要求を受け付けて、ソースコードなどの情報を DB へ登録する機能である。2 つ目は、builder の動作が終わった後に実行コードを学習者にダウンロードさせる機能である。

front が機能するのは 2 つのタイミングである。1 つは、MDD ツールから POST メソッドを受信したタイミング(図 3 中 1)である。もう 1 つは、MDD ツールからコンパイル要求 ID の GET メソッドを受信したタイミング(図 3 中 17)である。以下、動作の番号は図 3 中の番号に対応させる。

[POST メソッドの場合]

2. zip ファイル内の対象デバイス名の取得。
3. DB に、ユーザ名、コンパイル要求 ID、モデルから生成したソースコードを含む zip ファイル、コンパイル対象デバイス名を登録。
4. chief のアドレスに GET メッセージを送信。

5. DownloadURL を含めたレスポンスを MDD ツールに送信。

[GET メソッドの場合]

18. コンパイル要求 ID をキーとしたレコードを DB から取得。
19. 実行コードを取得。
20. 実行コードを取得できた場合、実行コードを送信。
21. 実行コードを取得できない場合、該当するエラーとしてレスポンスを送信。

3.3 chief

このアプリケーションの機能は、コンパイル要求を振り分け、負荷分散を行う。各コンパイル要求を振り分ける際には、コンパイルを担当する Builder のアドレスを DB に登録する。chief が機能するのは front から GET メッセージを受け取ったタイミングである。

6. 対象デバイスに応じた builder のアドレスを DB に登録。

3.4 builder

このアプリケーションの機能は、対象デバイス毎のコンパイルを行い、生成された実行コードを DB に登録する。builder が機能するのは chief が DB に情報を登録したタイミングからコンパイル要求のタスクが完了するまでである。

7. DB からコンパイル要求 ID をキーに該当するレコードを検索・取得。
8. 該当するレコードの状態を「コンパイル中」に変更。
9. 該当レコードからソースコードが圧縮された zip ファイルを取得。
10. 該当デバイスのツールチェーンを用いてソースコードをコンパイル。

コンパイルが成功した場合:

12. 実行コードを該当レコードに登録。
13. 該当レコードの状態を「成功」として登録。

コンパイルが失敗した場合:

14. 該当レコードにコンパイル時のトレースバックを登録。
15. 該当レコードに状態を「失敗」として登録。

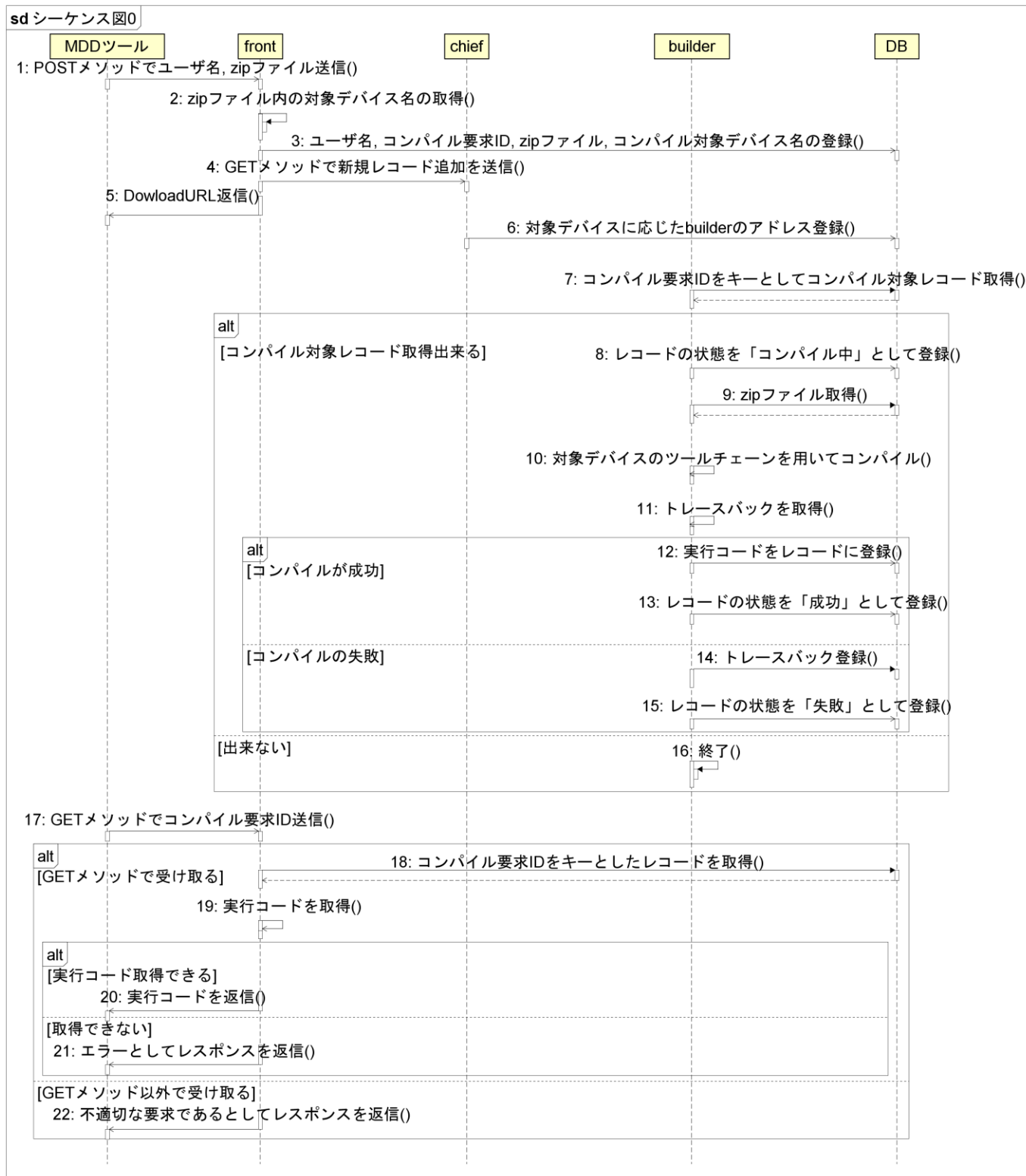


図 3 コンパイラサーバの動作概要図

4. コンパイラサーバの問題点と改良方針

4.1 中学校技術科での運用概要

本モデリング教育支援環境は 2018 年度より公立中

学校技術科の正規授業で運用している。利用者数は毎年約 280 名である。2018・2019 年度は Studuino を対象デバイスとした。2020・2021 年度は Studuino と micro:bit の 2 種の対象デバイスを用いている。これらの授業では、信号機をモデリング対象としている。

2018 から 2020 年度の 3 か年で合計 6,619 回のコンパイルリクエストがあった。学習者 1 人あたりでは、7.9 回のリクエストとなる。

4.2 問題点

図 4 にコンパイルリクエストの成否の割合を年度毎に示す。この中学校の運用では 3 か年平均で 4.6%(6,619 回中 306 回)のコンパイルが失敗となっていた。しかし、2020 年度のある授業では、52.0%(25 回中 13 回)が失敗となり、別の授業では、50.0%(80 回中 40 回)が失敗となることもあった。コンパイルが失敗すると、学習者のみでの対応は難しい。それは、コンパイルが失敗した際に表示されるメッセージが学習者にとっては難解、あるいは情報不足だからである。これは、円滑な授業の妨げになりえる。解決するためには、モデル図の作成に問題があるのか、支援環境の不具合なのか等、コンパイルが失敗した問題箇所を特定する必要がある。更に、学習者自身や、学習者を指導する教師が即座に対策をとれるようなメッセージをコンパイルサーバが提示することが考えられる。

コンパイルが失敗する主な理由は 2 つある。図 3 のシーケンス図における 2 と 10 である。これらは、builder が関連する処理である。

- 2. zip ファイル内の対象デバイス名の取得。
- 10. 該当デバイスのツールチェーンを用いてソースコードをコンパイル。

2 は、MDD ツールで生成された zip ファイルの中にある対象デバイスの情報と、DB に登録されたコンパイル対象名を比較する動作である。この比較が原因と

なるコンパイルの失敗は、①zip ファイルが破損している場合、②zip ファイルに対象デバイスの情報が含まれていない場合等が考えられる。

10 でコンパイルが失敗する要因は、③コンパイルに時間がかかる場合、④生成されたソースコードに不備がある場合である。

一方、現状のコンパイルサーバから提示されるコンパイルの失敗に関する情報は 2 種類である。1 つ目の情報はコンパイルのエラー文である。その表示が出るのは④の場合である。この場合、提示される情報が難解であり、学習者自ら対策をとることは難しい。2 つ目の情報は「none」である。この表示が出るのは①～③の場合である。この場合は、提示される情報が不足しており、学習者自らが対策をとることは難しい。それはコンパイルサーバの管理者や学習者を指導する教師も同様である。

4.3 コンパイルサーバの改良方針

学習者にとって難解あるいは、情報不足なコンパイルの失敗に関する情報提供の内容を改良する。

学習者にとって難解なコンパイルのエラー文が表示される場合(④)、現状はコンパイルのツールチェーンのトレースバックが示される。これを、実行コードを生成する言語コンパイラのトレースバックに変更する。これにより、不具合のあるソースコードの行数と不具合の内容が提示できるようになると期待する。また、不具合の内容は、学習者や教師にも理解できるように表現しなおす必要があろう。

学習者にとって情報不足な「none」が表示される場合は、①～③の要因に応じた情報提示が必要となろう。①と②は主として MDD ツールに起因する。これらの場合には、コンパイル対象となる zip ファイルに不具合がある旨の情報提供が考えられる。③は Docker コンテナ化された builder 内でコンパイル待ちが生じることに起因する。これは同時に多数のコンパイル要求が与えられた場合に生じる可能性がある。この場合には、builder の数を増加させる対策をしたうえで、コンパイル結果を得るまでに待機が必要である旨や待機時間等の情報提供が考えられる。

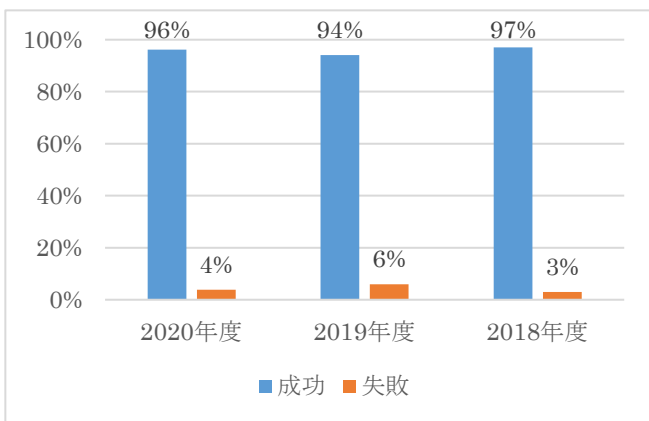


図 4 年度毎のコンパイルリクエストの成否の割合

5. おわりに

本研究の目的はモデリング教育支援環境のコンパイルサーバを対象とした改良である。中学校技術科正規授業での運用を通して、コンパイルサーバの問題点を整理した。コンパイルが失敗した際に表示されるメッセージは、学習者にとって難解、あるいは情報不足である。シーケンス図にまとめたコンパイルサーバの動作を概観し、コンパイルが失敗する要因を整理した。そのうえで、コンパイルが失敗した際に学習者に提示されるメッセージの改良方針をまとめた。

今後改良方針に従い、コンパイルサーバを改良していく。また、授業運用を通し、改良したコンパイルサーバを評価していく。

謝辞：本研究は科研費基盤研究 B:16H03074 の支援を受けた。

参 考 文 献

- (1) 情報処理学会, “カリキュラム標準 J17 カリキュラム標準ソフトウェアエンジニアリング領域(SE) J17-SE-報告書-20180319,” 2018.
- (2) 文部科学省, “中学校学習指導要領(平成 29 年告示) 解説技術・家庭編,” https://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2019/03/18/1387018_009.pdf (accessed 2021/12/1).
- (3) 赤山聖子, 久住憲嗣, 部谷修平他, “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”, 情報処理学会論文誌, 55(1), pp.72-84, 2014.
- (4) Starrett C., “Teaching UML Modeling Before Programming at the High School Level”, Proc of the 7th IEEE International Conference on Advanced Learning Technologies, pp.713-714, 2007.
- (5) 香山瑞恵, 小形真平, 永井孝, “モデル駆動開発方法論に基づく UML プログラミング教育環境”, 教育システム情報学会誌, 36(2), pp.118-130, 2019.
- (6) 小形真平, 香山瑞恵, “モデリング教育支援のためのステートマシンモデル評価手法の提案” 組込みシステムシンポジウム 2019 論文集, pp.45-52, 2019.
- (7) 但馬将貴, 香山瑞恵, 小形真平, 橋本昌巳, “UML に基づく概念モデリングにおける状態遷移図に対するモデル記法チェック機能の効果”, 信学技報知能ソフトウェア

工学, 116(418), pp.7-12, 2017.

- (8) 宮島和音, 小形真平, 香山瑞恵, 岡野浩三, “UML モデリング教育を支援するルールベースのクラス図採点支援ツール”, 情処研報ソフトウェア工学, 2015-SE-189(23), pp.1-6, 2015.
- (9) 増元健人, 香山瑞恵, 小形真平, 橋本昌巳, “クラス図を用いた基礎的概念モデリングにおける誤り分析に基づく初学者向け誤り自動検出機能の開発”, 情処研報ソフトウェア工学, 2015-SE-187(15), pp.1-7, 2015.
- (10) 横田寛明, 香山瑞恵, 小形真平, 橋本昌巳, 大谷真, “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”, 情処研報コンピュータと教育, 2014-CE-126(9), pp.1-6, 2014.
- (11) 原行人, 西尾圭太, 増田壮志, 香山瑞恵, 久住憲嗣, 伊東一典, 橋本昌巳, 大谷真, “MDD のための基礎教育環境の構築”, 組込みシステムシンポジウム 2011 論文集, pp.16-1-16-5, 2011.
- (12) Mellor, S. J., Clark, A. N., Futagami, T., “Model-Driven Development”, IEEE Software, 20 (5), pp.14-18, 2003.
- (13) 経済産業省, 自動車産業におけるモデル利用のあり方に関する研究会, https://www.meti.go.jp/policy/mono_info_service/mono/automobile/mbd/mbd.html (accessed 2021/12/1).
- (14) Hiya, S., Hisazumi, K., Fukuda, A. et al., “clooca: Web based tool for Domain Specific Modeling,” Proceedings of 16th ACM/IEEE MoDELS Demonstrations, pp.31-35, 2013.
- (15) Change Vison, Inc., “Premier Diagramming, Modeling Software & Tools | Astah”, <https://astah.net/> (accessed 2021/12/1).
- (16) 丸山凌凱, 香山瑞恵, 永井孝, “課題管理機能を有する UML プログラミング環境の設計と実装”, 教育システム情報学会研究会, 2021(5) (印刷中).
- (17) 中野敬久, 香山瑞恵, 永井孝, 大宅剛生, “コンテナ型仮想化によるモデリング教育用コンパイルサーバ利用の考察”, 信学技報, 120(335) ET2020-49, pp. 17-22, 2021.
- (18) 大宅剛生, 香山瑞恵, 永井孝, “コンテナ型仮想化によるモデリング向けコンパイルサーバでの micro:bit 対応機能の評価”, 第 46 回教育システム情報学会全国大会講演論文集, pp.107-108, 2021.