

# SMart-Learning: DSLベースのUMLステートマシン図 学習支援ツール

小形 真平<sup>1,a)</sup> 岡野 浩三<sup>1,b)</sup>

**概要:** IoTシステムや組込みシステムの需要が高まり、ステートマシン図の学習の重要性は増している。ステートマシン図では、実際に生じるイベント系列に対して、適切な状態遷移を記述することが肝要であるが、そもそも、初学者がこのイベント系列を見極めることは容易ではない。本稿では、初学者が自身のステートマシン図を容易に検証できるように教師が想定するイベント系列や実際に生じたイベント系列でDSLベースのステートマシン図をシミュレートするツールであるSMart-Learningを紹介する。

**キーワード:** Unified Modeling Language, ドメイン特化言語, ステートマシン図, 学習支援, シミュレーション

## 1. はじめに

IoTシステムや組込みシステムの需要が高まり、システムの状態を捉えて仕様化するステートマシン図の学習の重要性は増している。特に外部からの入力をイベントと捉えて処理し、一定時間内に外部に応答することを繰り返すリアクティブ性をもったシステムのステートマシン図では、実際に生じるイベント系列を捉えた適切な状態遷移の記述が肝要である。

しかし、大学の授業などの時間に限りのある学習環境では、モデリング、実装、テストのサイクルを回すことは容易ではない。加えて、システムが要求を満たさない動作を行った場合、その動作のみからでは、初学者はイベント系列の想定、モデル、実機のいずれに原因があるのかを特定することが困難な場合がある。

本稿では、初学者がその原因を探りやすいように、イベント系列を初学者や教師が明示的に与えて、ステートマシン図を分かりやすくシミュレートすることをねらうツールセットSMart-Learningを紹介する。本ツールによって、イベント系列の想定またはモデルが妥当と分かれば、実機に原因があるといった推定が容易となることが期待できる。

## 2. SMart-Learning

### 2.1 ねらい

SMart-Learningは、アニメーションを含むシミュレーションツールをステートマシン図教育に導入することで、以下のように問題を改善することを趣旨とする。

- 見た目が複雑で手動では評価しにくい<sup>1</sup>が、構文的には正しい図の評価を容易にする。たとえば、多数の線が交錯する図の評価や、演算子の記述ミスの発見を容易にする。
- システムの実行以前に、実際に生じるイベント系列を把握し難い場合の図の検証を容易にする。たとえば、ライントレースを行うシステムでは、線の通り方の戦略によって、線を認識するためのセンサ（例：光センサ）から得られる値の系列に差異が生じる。その値がシステムの認識するイベントと直接紐づけられるならば、戦略によって実際に得られるイベント系列が異なることとなる。そのため、システムの実行以前に初学者がイベント系列の具体例を明確に想定することや、予め教師がテストケースとしてのイベント系列を初学者に一樣かつ適切に提供することは困難と考えられる。そこで、熟練者としての教師が実際の図を見て想定することや、システム実行時にイベントを記録することによって得られるイベント系列を基に初学者の図を検証することを容易にする。

### 2.2 構成

SMart-Learningは、UMLモデリングツールであるastah[1]のプラグインとして開発された。astahの利用を前提とするが、後述するように仕組みとしては汎用的である。

ステートマシン図は、UMLの記法を基に既定の状態やイベントを選択するように作成されたもの、つまり、DSL(Domain Specific Language)を用いて作られたものを前提とする。なお、教育上でDSLを扱う考え方は自然なものである[2]。

また、ガードやアクションは、実行可能なJavaScript

<sup>1</sup> 信州大学  
Shinshu University

a) ogata@cs.shinshu-u.ac.jp

b) okano@cs.shinshu-u.ac.jp

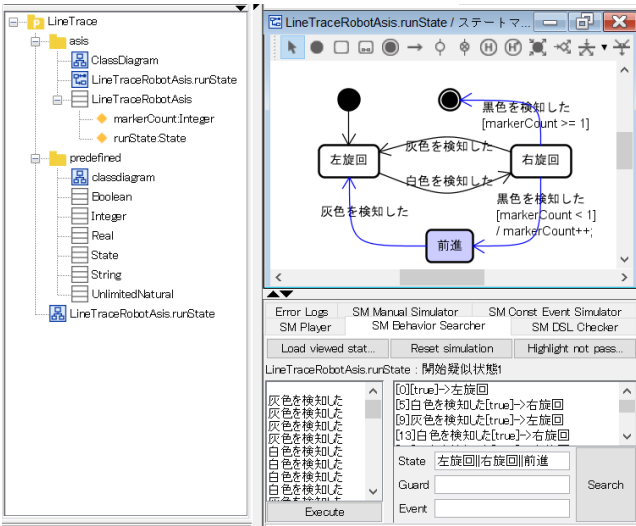


図 1 SMart-Learning の SMBS 実行例

の文法で記述されたものを前提とする。これは、astah プラグインの実装言語が Java に限定されており、Java の ScriptEngine で JavaScript が扱えることによるが、簡単な条件や振舞いの記述が多いと考えられる教育上においては、汎用的な記法として扱える見込みが高い。ガードやアクションで共通的に利用する変数は、状態マシン図のコンテキストとなるクラスに属性として定義するものとし、その型は UML の基本型である Integer, Real, UnlimitedNatural, String, Boolean のいずれかとする。なお、JavaScript の文法チェックには、パーサジェネレータである ANTLR[3] が提供する既定の JavaScript パーサを利用して行っている。

### 2.3 DSL Checker

状態マシン図が DSL に沿うことを保証する必要がある。DSL Checker は、DSL で定める状態とトリガそれぞれの語句をテキストのリストとして与え、状態マシン図がそれらのみを使用しているかを検査する。

### 2.4 State Machine Behavioral Searcher (SMBS)

SMBS は、テキストのリストとしてイベント系列を与えて状態マシンを実行する。実行結果として、どのイベントによって何の遷移が生じたかを一覧として閲覧できる。図 1 は、SMBS を表示した astah の実行画面である。図の真中下が与えたイベントのリストであり、その右上が実行結果を指す。実行結果では、たとえば、“5” 番目に生じた“白色を検知した” イベントで“右旋回” 状態に移行したことを示す。[true] はガードを表す部分であり、ガード無しの場合の内容は“true” となる。実行結果は右下のフォームで絞り込むこともできる。また、与えられたイベント系列を実行して、一度も通過しなかった状態および遷移は図上で青く強調表示できる。

### 2.5 State Machine Player (SMP)

SMP は、テキストのリストとしてイベント系列を与えて、一定の時間間隔でイベントを取り出し状態マシンに与えて実行する。そして、状態遷移に応じて、現在状態や直前に通った遷移を赤に着色して、状態遷移のアニメーションを行う。SMP はステップバイステップの実行のため、SMBS に比べて最終的な実行結果に辿り着くまでに時間がかかるが、直感的な理解の手段を提供できる。また、実行中の変数の値はオブジェクト図形式で表示でき、ステップごとに更新されるので、その変化を適時確認できる。

### 2.6 State Machine Manual Simulator (SMMS)

SMMS は、初学者が現在状態へ指定のイベントを与えて状態マシンを実行する。これは、教師などから与えられたイベント系列では行えない検証を支援する相補的な手段となる。SMMS では、現在状態の出力遷移全てのトリガからイベントを与えるボタンを生成し、表示する。なお、SMP と同様にアニメーションや変数表示も可能である。

## 3. 評価

教師視点で SMart-Learning が教育に役立つかを評価するために、実際の授業で得られた初学者 48 名のライントレスロボットの状態マシン図（図 1 に示した程度の規模）に適用した。結果の概要は以下の通りである。

- 全ての図に対して、別途に用意したイベント系列をエラーなく適用できた。これは DSL が有効に機能したと言える。
- 初学者によっては多数の線が交錯するような見込みが複雑で手動では評価しにくい図を記述していたが、その誤りを手動に比べて容易に発見できた。
- アニメーションにより、細かい誤りを手動に比べて容易に発見できた。たとえば、ガードで等価演算子とすべき箇所が代入演算子となっていた誤りを発見できた。

## 4. まとめ

本稿では、初学者が状態マシン図を与えられたイベント系列で容易に検証できるように DSL ベースの状態マシン図をシミュレートするツールである SMart-Learning を紹介した。今後、対応できる記法やセマンティクスの拡充、授業適用による初学者視点の評価を行っていきたい。

### 参考文献

- [1] Change Vision: astah, <http://astah.change-vision.com/ja/>, (accessed 2016-06-07).
- [2] Hiya, S., et al.: clooca: Web based tool for Domain Specific Modeling. In: Demos/Posters/StudentResearch@ MoDELS 2013, pp.31-35, 2013. .
- [3] Parr, T.: ANTLR, <http://www.antlr.org/>, (accessed 2016-09-09).