

# 属性付きトークンを導入した拡張プレース/トランジションネット によるソフトウェアモデリングの提案

黒住 遼† 高木 智彦‡

香川大学工学部† 香川大学創造工学部‡

## 1. はじめに

ソフトウェアモデリングのための表記法の一つである PN (Place/transition Net) は、並行動作を含むソフトウェアの状態遷移を表現するのに適している。PN に対して VDM++によるアクションやガードなどの形式的定義を加え、より複雑な表現ができるように拡張した表記法が EPN (Extended PN) であり、これに基づいて VDM (Vienna Development Method) 仕様を作成する手法も提案されている [1]。

本研究ではさらに、属性付きトークン、すなわち、VDM++によって定義される属性（オブジェクトを特徴づける変数）をもつトークンを EPN に導入することを提案する。この EPN を EPNAT (EPN with Attributed Tokens) と呼ぶ。EPNAT によって、ソフトウェアにおける複数のオブジェクトの動作を表現したり、複数の処理の優先順位を表現したりするための効果的な方法を考察する。また、EPNAT に基づいて VDM 仕様を作成する方法も提案する。本稿は、2 節で EPNAT とその VDM 仕様の作成について、3 節で EPNAT の有効性に関する考察を示す。

## 2. 提案手法

### 2.1 EPNAT

PN は、ソフトウェアの各オブジェクトの状態を示すプレース、処理を示すトランジション、処理の実行による状態の変化を示すアーク、各オブジェクトの現状態を示すトークンで構成されるモデルである。PN では、ソフトウェアの基本的な状態遷移しか表現することができない。そこで EPN では、トランジションに対して、処理の開始に際し外部から与えられる情報（引数）や、その処理の詳細な内容（アクション）、処理の開始が可能になるための条件（ガード）を、VDM++によって定義できるようになっている。しかしながら、EPN では、多数のオブジェクトを含むソフトウェアをモデリングする場合、たと

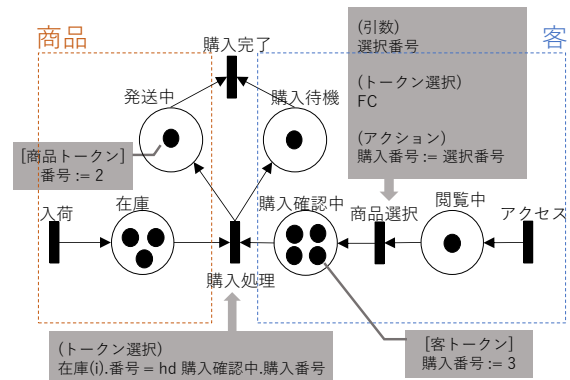


図 1. ネットショップの振舞いを示す EPNAT

え同じ属性や動作をもつオブジェクト（同型オブジェクト）であっても、オブジェクト毎にプレースやトランジションなどを記述する必要があり、モデルが大規模化したり限られた数のオブジェクトしか扱えなかったりする問題があった。そこで EPNAT では、各トークンをそれぞれ一つのオブジェクトとして表現して必要な属性をもたせ、同型オブジェクトの共通状態をプレース、共通処理をトランジションとして表現する。各プレースは、どの型のオブジェクトのトークンを保持可能かが定められている。また、トランジションには、発火時に優先的に使用するトークンを適切に選択するための方法を定めることができる。選択方法の記述が特でない場合は FC (First Come) 方式が採用され、プレースに到達した時間の早いトークンから順に使用することとなる。

図 1 は、簡略化されたネットショップの振舞いを示す EPNAT である。オブジェクトの型として客と商品があり、トークンはどちらかに対応する。客トークンを保持し得るプレース群、商品トークンを保持し得るプレース群から構成されており、各トークンがプレース間を移動する様子から、ソフトウェアや各オブジェクトの状態遷移を表現することができる。

たとえば「商品選択」トランジションは FC 方式であり、「閲覧中」プレースへの到達時間の早いトークンから使用されることが示されてい

る。また、「購入処理」トランジションでは、FC 方式で「購入確認中」プレースから客トークンを取得する。そしてその客トークンのもつ「購入番号」と一致する、「在庫」プレース内の商品トークンを取得し、それぞれ遷移先プレースに移動させることが示されている。この例から、EPNAT は任意の数のオブジェクトを扱えるだけでなく、各オブジェクトを区別した上で、処理の優先度をふまえたオブジェクトの動作を簡素に表現できることが分かる。

### 2.2 VDM 仕様の作成

EPNAT に基づいて VDM 仕様を作成することができる。図 2 は、図 1 に基づいて作成した VDM 仕様の一部である。(a)部では、属性付きトークンをレコード型として定義する。(b)部で、保持し得る属性付きトークンに対応するレコード型変数の列型変数として各プレースを定義する。その直下は、アクションやトークン選択処理で使用する変数の定義である。そして(c)部において、各トランジションを操作として定義する。(d)部は、「購入処理」トランジションにおけるアクションやトークンの移動処理である。FC 方式は、当該トランジションの遷移元プレースに対応する列型変数の先頭要素を取り出して、遷移先プレースに対応する列型変数の末尾に挿入することで実現している。トランジションのガードは(e)部に記述されている。

### 3. 考察

属性付きトークンの導入によって、ソフトウェアにおける複数のオブジェクトの動作を表現したり、複数の処理の優先順位を表現したりすることができることが 2.1 節において分かった。EPNAT によってソフトウェアの振舞いを確認することもできるし、あるいは個々の属性付きトークンに着目することによって、各オブジェクトの振舞いを確認することもできる。ただし、有効性に関しては、より現実的な規模の例題を通して今後評価する必要がある。また、2.2 節では、EPNAT に基づく VDM 仕様の作成を示すことができた。VDM 仕様はツール上で実行可能であるため、ソフトウェアを実装する前にその基本動作を確認し、早期に問題点を明らかにできる可能性がある。ただし、VDM 仕様の作成にかかる労力を削減するために、EPNAT から VDM 仕様を生成するツールの開発が必要である。

### 4. おわりに

ソフトウェアにおける複数のオブジェクトの

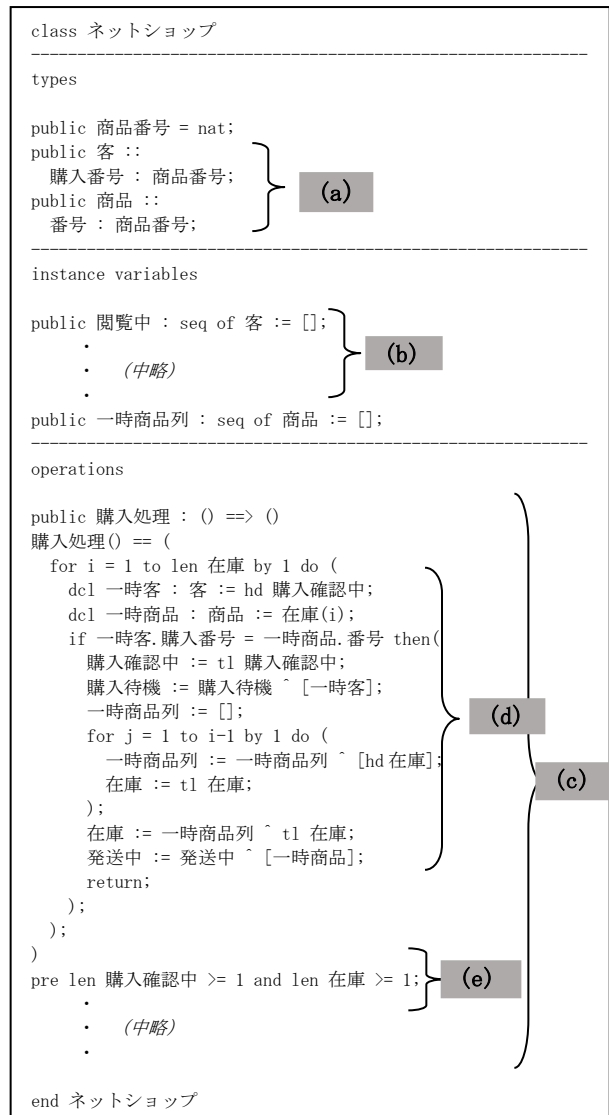


図 2. ネットショップの振舞いを示す VDM 仕様の一部

動作を表現したり、複数の処理の優先順位を表現したりするために EPN を拡張した表記法である EPNAT と、それに基づいた VDM 仕様の作成方法を提案した。今後の研究では、EPNAT と VDM 仕様を連携させ、多数のオブジェクトから構成されるソフトウェアの振舞いを視覚的に確認できるツールを開発する。そして、ソフトウェアのモデリングだけでなくテストを効果的に支援する手法を検討する予定である。

### 参考文献

[1] 高木智彦, 赤木章紀, “拡張プレース/トランジションネットに基づく VDM 仕様の構築手法の提案”, 情報処理学会第 79 回全国大会, pp.195-196, 2017.