

場面の概念を用いた離散・連続混合システム のシミュレーションモデル* 〔場面遷移ネット(STN)の提案〕

川田 誠一^{*1}, 川田 尚吾^{*2}, 渡辺 敦^{*1}

New Simulation Model for Discrete-Continuous Hybrid Systems Using the Concept of Scene 〔Proposal for the Scene Transition Net (STN)〕

Seiichi KAWATA, Shogo KAWADA and Atsushi WATANABE

In order to describe the discrete-continuous hybrid system, the concept of the scene transition net (STN) is proposed. The STN consists of two principle elements, events and scenes, and integrates discrete event systems and continuous dynamical systems into a single model. The representation of STN can be easily converted to an object-oriented program for simulation. An example clearly demonstrates the effectiveness of the method.

Key Words: Modeling, Production System, Discrete-Continuous Hybrid System, Scene, Object Oriented Program

1. 緒 言

ニュートン以来、我々は様々な自然界の現象を微分方程式を用いて表現し、理解してきた。しかし微分方程式によって表現された CVDS (Continuous Variable Dynamic System) モデルは自然現象などの連続状態システムのモデル化には有効であるが、生産システム、通信システムなど人工システムのモデル化には不向きである。たとえば航空券の予約システムの記述を考えた場合、これは本質的に離散事象であり、微分方程式で記述することはできない。このような離散的な視点からのモデル化を行うためのモデルとして、ペトリネットや有限状態機械等の様々な DEDES (Discrete Event Dynamic System) モデルが開発されている⁽¹⁾。

しかし、実際に何らかの人工システムのシミュレーションを行う場合、連続と離散の両方の視点が必要となることがある。このようなシステムは離散・連続混合システムと呼ばれており、生産システムの設計においても、このようなシステムのシミュレーションの実

現が課題となっている。Fishwick⁽²⁾ は異種モデルの統合を用いた離散・連続混合システムのモデリングアプローチを示した。しかし、このアプローチでは複数のシステムが互いに影響しながら並列的に挙動する場合を表現することは難しい。

著者らは、複数のシステムが相互に影響しながら並列的に挙動する連続・離散混合システムを表現する方法として、場面の概念を導入した離散・連続混合システムのモデルを提案する。さらに、オブジェクト指向アプローチを用いてこのモデルを簡潔にコンピュータ上に表記できることを示す。

2. システムの捉え方

複数のサブシステムが互いに影響を及ぼしながら挙動しているシステムを捉えるとき、我々はよく、サブシステム個々の挙動の集合体として全体のシステムの挙動を表現する。たとえば、ある野球チーム(システム) T1 を考えてみる。T1 のある日の試合の内容を表現する場合、我々はよく T1 に所属する選手(サブシステム) P1, P2, … のその日の活躍を表現することによって、チーム T1 全体の活躍を表現しようとする。このように、何らかのシステムの挙動をそのシステムのサブシステムの挙動の集合として表現しようとする

* 原稿受付 平成4年12月18日。

^{*1} 正員、東京都立大学工学部 (〒192-03 八王子市南大沢1-1)。

^{*2} 学生員、東京都立大学大学院。

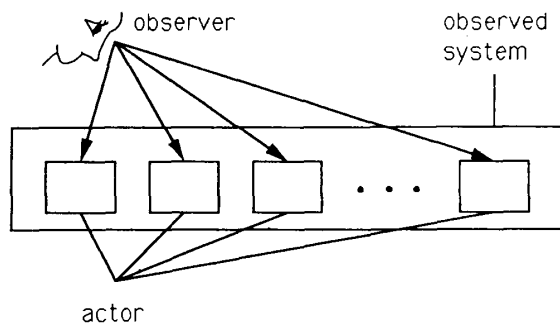


図1 アクタと被観測システム

場合に、このシステムを被観測システム(Observed System)、サブシステムをアクタ (Actor) と呼ぶ。図1は被観測システムとアクタのイメージを表している。

定義1 被観測システム(Observed System)

観測者がなんらかの観測 Ob を行うとき、その観測対象となるシステム S を観測 Ob における被観測システムと呼ぶ。

定義2 アクタ (Actor) 定義1の観測 Ob が、S のサブシステム $S_1, S_2 \dots S_n$ を対象とした観測 $Ob_1, Ob_2, \dots Ob_n$ を組合わせたものであるとき、 $S_1, S_2, \dots S_n$ を観測 Ob におけるアクタと呼ぶ。

アクタはシステムであるため、これをさらに幾つかのサブシステムに分割することができる。たとえば、ある野球マニアがこの試合におけるあるプレーヤ P_i のバッティングやピッチングのフォームに注目し、腕や膝の挙動にまで注意を払った場合を考えよう。この場合、観測者はまず、チームのプレーヤの個々の活躍に注目し、さらに個々のプレーヤの腕や膝の動きに注目する。この時、観測者は始めに個々にチームを被観測システム、プレーヤをアクタとした観測を行い、次にプレーヤを被観測システム、腕や膝をアクタとした観測を行い、これらの観測を統合したものであるとしてこの試合におけるチーム T1 の活躍を捉えている。このように、アクタをさらにいくつかのサブアクタ (Subactor) に分割して、より詳細な観測をすることが可能である。このような入れ子構造となっているシステムにおいて被観測システム、アクタをどのように決めるかという問題は、観測者がどのような観測を行うのかに依存している。オブジェクト指向シミュレーションにおいては、一般にアクタをオブジェクトとして定義し、シミュレーションを行う。よって、オブジェクト指向シミュレーションにおいてどのようなオブジェクトを定義するかという問題は、どのような観測を行うかに依存する。



図2 コインを拾う人間の挙動

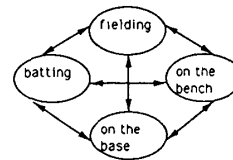


図3 野球のプレーヤの挙動

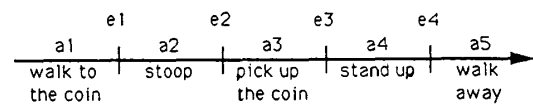


図4 活動と事象

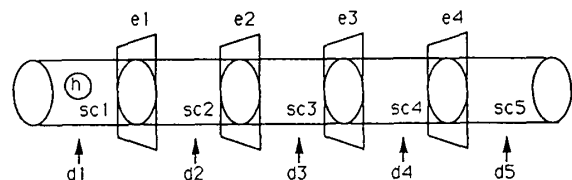


図5 場面のイメージ

3. 場面という概念

複雑な挙動を示すシステムを解析するとき、我々はよく、そのシステムの挙動を幾つかに場合に分類して、これらの組合わせとして全体の挙動を表現しようとする。たとえば人間が落ちているコインを拾うという動作をモデル化する場合、人間の動作を「コインに歩いて近づく」、「かがむ」、「コインをつかむ」、「立ち上がる」、「コインを持って歩く」の5つの動作に分類して考えたり (図2)、あるいは野球のプレーヤの動作を「守備」、「バッティング」、「出塁」、「ベンチ入り」の4つに分けて考えたり (図3) する。

DEDS においてはこのように分割された挙動を活動 (Activity) と呼び、また、ある活動から他の活動へ遷移する際の瞬間的な出来事を事象 (Event) と呼んでいる。これをコインを拾う人間を例にして図示すると図4のようになる。

ここに連続的な変数が入ってきた場合を考える。たとえば、このコインを拾う人間の頭部の重心座標 $h = (x, y, z)$ を考えてみる。図4の a_1, a_5 では h はほぼ水平に移動するであろう。 a_2, a_4 では上下方向に移動し、 a_3 ではほぼ静止した状態にあるであろう。このように h は $a_1 \sim a_5$ において、それぞれに固有の拘束条件にしたがって動的に挙動する。これを概念的に表したものが図5である。

図5において各場面は事象によって分割されている。そしてシステムの時間的挙動は、場面、事象、場面、事象のシーケンスで表現される。通常の離散事象システムでは、この場面は活動と定義されており、次の事象が生起するまでその状態は変化しない。したがってある事象とある事象との間における活動の何らかの動的变化を表現するには、新たに事象を定義して、その活動を分解することになる。しかし、活動を単なる事象間の状態を静的に表現するものとせず、連続的な動的挙動をも表現するものと定義しなせば、離散事象システムと連続動的システムを統合できる。そこで次のように場面を定義する。

定義3 場面(Scene) あるシステムSの活動 a_1, a_2, \dots, a_n が定義されたとする。活動 a_i におけるSのダイナミクスを d_i としたとき ($i=1, 2, \dots, n$)、活動 a_i とダイナミクス d_i を組合わせたものをシステムSの場面 sci と呼ぶ。

定義4 パフォーマ(Performer) 定義3-1において、システムSを場面 sci のパフォーマと呼ぶ。

定義5 現行場面(Current Scene), 現行パフォーマ(Current Performer) あるシステムSについて場面 sci ($i=1, 2, \dots, n$) を定義する。ある時刻 t において、システムSが場面 sck [$k \in (1, 2, \dots, n)$] にあったとする。このとき場面 sck を時刻 t におけるシステムSの現行場面と呼ぶ。逆にシステムSを時刻 t における場面 sck の現行パフォーマと呼ぶ。

前述した例では人間の動作を5つの場面に分けていたが場合によってはさらに大まかな、またはさらに細かい分割を行うことも可能である。たとえば「コインを持っている」、「コインを持っていない」の2場面だけに分割することも可能であろう。あるいは「かがむ」という場面を、「顔を下に向ける」、「腰を曲げる」、「膝を曲げる」など、細かく分割することもできる。このように、場面もアクタと同様に入れ子構造になっている。よって一つの場면을複数のサブ場面 (Subscene) に分割することができる。アクタの場合と同様、どのような場面を設定するかという問題は、どのような観測をおこなうかという問題に依存する。

DEDS では、ある瞬間的な事象を境に活動が瞬時に変化する。よってあるシステムの現行場面も、なんらかの事象を境に瞬時に変化する。このような場面の变化を場面遷移 (Scene Transition) と呼ぶ。また、場面の变化をひき起こす事象のことを場面遷移境界 (Scene Transition Boundary) と呼ぶ。

定義6 場面遷移, 場面遷移境界 あるシステムSについて、場面 sci ($i=1, 2, \dots, n$) が定義されたと

する。ここで $j \neq k$, $j, k \in (1, 2, \dots, n)$ として, $t=t_0$ においてある事象 $E_{j,k}$ が起こったならば,

$$CS(S, t_0-0) = scj \Rightarrow CS(S, t_0+0) = sck$$

となるような事象 $E_{j,k}$ をシステムSの場面 scj から場面 sck への場面遷移境界と呼ぶ。またこのときのSの現行場面の变化を場面遷移と呼ぶ。ここで $CS(S, t)$ はシステムSの時刻 t における現行場面を表す。

次に複数のアクタによって構成されるシステムについて考えてみる。ここでは二人のボクサー A, B による試合を例に考えてみる。A, B に関して、それぞれ試合中、休憩中の二つの場面を設定する (図6)。

この時 A における試合中という場面と、B における試合中という場面をばらばらに考えるのは不自然である。よってこれを図7のように一つの場面にまとめる。これはペトリネットにおける融合の考え方である⁽³⁾。

場面 RestA にあったアクタ, BoxerA と、場面 RestB にあったアクタ, BoxerB は、開始ゴングによって場面 Fight に入ってくる。場面 Fight において二つのアクタ BoxerA と BoxerB は互いに影響を及ぼしながら場面 Fight におけるダイナミクスの影響を受け、自らの持つ変数 (たとえば、疲労度、損傷度、攻撃力など) を変化させる。やがて終了ゴングが鳴ると、BoxerA は RestA に、BoxerB は RestB に戻る。そして RestA, RestB において二つのアクタはそれぞれの場面におけるダイナミクスに応じて自らの変数を変化させる (たとえば疲労の回復など)。再び開始ゴングが鳴ると、以上の動作を繰り返す。

この場面 Fight のように、アクタ個々について場面を設定するよりも、複数のアクタの取りうる場面を一つにまとめた方が概念的に分かりやすい場合がよくあ

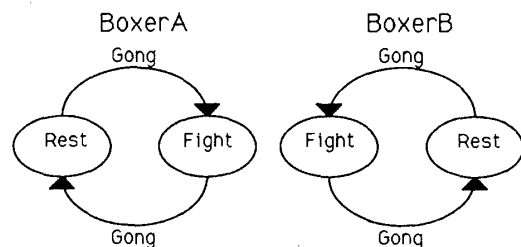


図6 ボクサーの場面

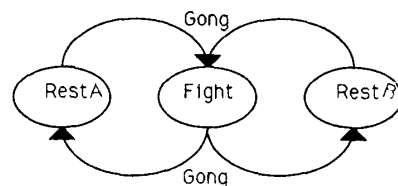


図7 ボクサーの場面

る。このようにして、複数のアクタについて設定された場面を一つにまとめたものを一般化場面と呼ぶ。

定義 7 一般化場面 (Generalized Scene) システム S_1, S_2, \dots, S_n を考える。システム S_i に対して、場面 $sci_1, sci_2, \dots, sci_{m_i}$ が定義されたとする ($i=1, 2, \dots, n$)。このとき p 個のシステム $S_{q1}, S_{q2}, \dots, S_{qp}$ [$n \geq p, q_1 \neq q_2 \neq \dots \neq qp, q_1, q_2, \dots \in (1, 2, \dots, n)$] について定義された場面, $scq1_{Aq1}, scq2_{Aq2}, \dots, scqp_{Aqp}$ [$Aqw \in (1, 2, \dots, m_{qw}), w=1, 2, \dots, p$] をひとつにまとめて、次のような場面, $scGen$ を定める。

$\forall t, cp(Sqw, t) = scqw_{Aqw} \Leftrightarrow cp(Sqw, t) = ScGen$ このとき、場面 $ScGen$ を、場面 $scqw_{Aqw}$ に対して、一般化場面と呼ぶ。

定義 8 パフォーマ (Performer) 定義 6 において、システム $S_{q1}, S_{q2}, \dots, S_{qp}$ を場面 $scGen$ のパフォーマと呼ぶ。

なお、一般化場面における現行場面、現行パフォーマは定義 5 と同様に定義される。

4. 場面遷移ネット

場面と事象を用いた離散・連続混合システムの動的挙動を図式的に表現するために場面遷移ネットを提案する。場面遷移ネットはペトリネットのコンセプトを基にシステムの場面遷移の規則を表現するとともに、複数のアクタがどのようにかわりあってゆくかを示している。場面遷移ネットの特徴をまとめると以下のようになる。

- ・ペトリネットでは失われてしまう、各場面のパフォーマが表記されている。

- ・場面遷移ネットの形で表現されたシステムは、ダイレクトにオブジェクト指向プログラムによってシミュレートできる。

- ・場面遷移ネットを書くことにより、シミュレーションに適切なクラス分けが可能

- ・場面遷移ネットを書くことにより、シミュレーションに適切な場面分割が可能

図 7 のボクサーの例を場面遷移ネットに書き直すと図 8 のようになる。基本的に、場面遷移ネットは図 9 の場面遷移のチャートにパフォーマを書き加えたものである。場面遷移ネットの主要素は以下の通りである。

- ・場面ボックス (Scene Box)：ペトリネットのプレイスに相当する。場面を表すもので、パフォーマが書き記されている。

- ・トランジション (Transition)：場面遷移境界を表す。すなわち、ペトリネットのトランジションと同じ

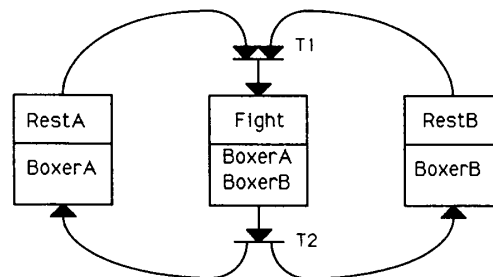


図 8 場面遷移ネット

く事象を表す。

- ・アーク (Arc)：アクタの場面遷移を表す。

場面遷移ネットの発火規則 (すなわち、場面遷移境界) は、各トランジションごとに明記しなくてはならない。これは場面遷移ネットの発火の条件は各アクタの持つ連続変数に依存する場合が多いため、ペトリネットのように離散変数のみによる発火規則 (あるプレイスにおけるトークンの有無) だけでは表現しきれないからである。

5. 場面遷移ネットを利用した オブジェクト指向シミュレーション

場面遷移ネットによって表現されたシステムは、オブジェクト指向プログラミングによって簡単にシミュレーションが可能である。ここでは、オブジェクト指向言語 Smalltalk 80 を用いたシミュレーションの方法を述べる。

プログラムのクラス構造は図 9 のようになっている。以下、各クラスの役割を示す。

- ・Net：場面遷移ネットを表す。クラス変数として各場面の現行システムを保持する。また、クラスメソッドとして各トランジションの発火条件を持つ。

- ・SimClock：時間を扱うクラス。クラスメソッド now で、あらゆるオブジェクトのメソッドから時間にアクセスすることが可能。

- ・Model 1, Model 2, ...：これらのクラスのインスタンスが、アクタを表す。インスタンス変数として現行場面を保持している。インスタンスメソッド continue を送ると、現行場面を参照して適切なダイナミクスにより自らの持つ連続変数を変化させる。

- ・NetModel：Model 1, Model 2, ... に共通の変数、メソッドを定義したスーパークラス。

これらのクラスを設定したのち、以下の要領でシミュレーションを進める。シミュレーション結果は各アクタに記録し、必要に応じて出力する。

1. SimClock の時刻を初期化する。

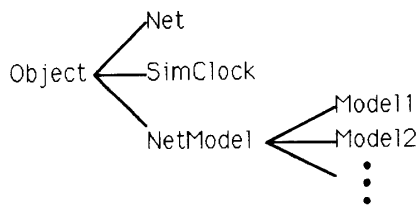


図 9 クラス構造

2. Model 1, Model 2, …のインスタンスをクリエイトし, Net のクラス変数に収める。
3. ネット内のすべてのアクタ(Model 1, Model 2, …のインスタンス)にインスタンスメソッドを送り, 連続変数の計算を行う。
4. 発火可能なネット内のトランジションをすべて発火させる。
5. SimClock の時刻を刻み時間分だけ進める。
6. 3に戻る。

6. シミュレーション例

実際に場面遷移ネットを利用したシミュレーションの例を紹介しよう。連続鋳造片の加熱炉への搬送ラインのシミュレーションを行った。

図 10 は今回シミュレーションを行った搬送ラインを示している。A, B, E, D, F はバッファ, ①～⑧はローラテーブルである。シミュレーションの目的はこのラインが対応できる搬送量, および搬送中のスラブの温度変化の検証である。この場合, 搬送規則が離散系に, 放射によるスラブの温度変化が連続系になっている。

搬送規則は次のようになっている。

- ・1St と 2St は交互に動作し, スラブは 1St から A へ, 2St から B へそれぞれ払い出される。
- ・①搬入可でありかつ, E, D, F にスラブが無い場合に A または B のスラブを①へ搬入する。
- ・①搬入可であり, E, D, F にスラブが無い場合, F, D, E の順でスラブを①へ搬入する。
- ・①搬入不可の場合, F, D, E の順にスラブを詰める。
- ・①搬入されたスラブは, 同一テーブル上にスラブが重複しないように②～⑧を経て加熱炉へ搬入される。

また, スラブ内部の温度分布は無視でき, スラブの温度変化は放射による移動熱量 $Q[\text{J/sec}]$ の影響によってのみ変化すると仮定する。ここで Q は次式で表される⁽⁴⁾。

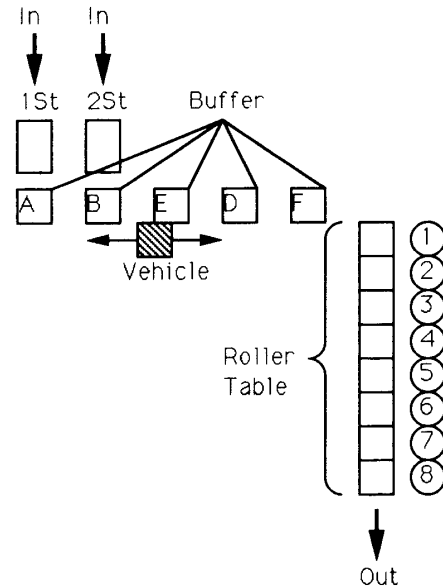


図 10 搬送ライン

$$Q_r = \sigma A_1 (T_1^4 - T_2^4) \left\{ \frac{1}{\frac{1}{\epsilon_1} + \frac{A_1}{A_2} \left(\frac{1}{\epsilon_2} - 1 \right)} \right\}$$

ここで,

$\sigma[\text{W/m}^2\text{K}^4]$: ステファン・ボルツマン係数

$T_1[\text{K}]$: スラブの平均温度

$T_2[\text{K}]$: 工場内の室温

$A_1[\text{m}^2]$: スラブの表面積

$A_2[\text{m}^2]$: スラブを囲む面の面積

ϵ_1 : スラブの平均放射率

ϵ_2 : スラブを囲む面の平均放射率

である。

$\sigma, T_2, A_1, \epsilon_1$ は搬送中一定であるが, A_2, ϵ_2 は次のように変化する。

- a) バッファ E, D, F 上にある時

$$A_2 = 7.00$$

$$\epsilon_2 = 0.200$$

- b) a 以外

$$A_2 \rightarrow \infty, \text{すなわち,}$$

$$Q_r = \epsilon_1 \sigma A_1 (T_1^4 - T_2^4)$$

この搬送システムを場面遷移ネットで表現すると, 場面 SC1～SC17, および場面遷移境界 T1～T21 が定義される。本論文におけるネットの全貌の表記は省略するが, これらの場面が持つ意味は次のようになる。

SC1: 1st にスラブ有り

SC2: 2st にスラブ有り

SC3: a が空き

SC4: b が空き

SC5: 1st から a へスラブが移動中

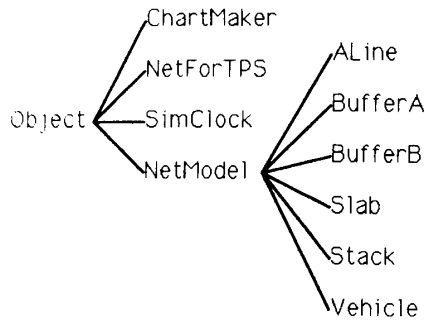


図 11 クラス構造

- SC6: 2st から *b* ヘスラブが移動中
 SC7: *a* にスラブ有り
 SC8: *b* にスラブ有り
 SC9: 台車空き
 SC10: スタック (*d*~*f*) にスラブ無し
 SC11: スタックに搬入可能
 SC12: スタックに搬入不可
 SC13: 1 に搬入可能
 SC14: 1 に搬入不可
 SC15: スタック内でスラブを移動中
 SC16: 1 ヘスラブを移動中
 SC17: アウトプット

シミュレーションは Smalltalk 80 を用いて 5 章の要領で行った。今回のシミュレーションにおけるクラス構造は図 11 のようになった (cf. 図 9)。主なクラスの概要は次の通り。

- CharMaker: グラフ作成ツール
- NetForTPS: ネットを表すクラス、クラス変数として SC1~SC17 を持ち、ここに各場面をとるシステム (すなわち, ALine, BufferA, BufferB, Slab, Stack, Vehicle のインスタンス) を収納。

NetForTPS の主なクラスメソッド

T1: 面遷移条件 T1 を確認し、満たされていれば場面遷移を実行。

T2: 面遷移条件 T2 を確認し、満たされていれば場面遷移を実行。

•
•
•

T21: 面遷移条件 T21 を確認し、満たされていれば場面遷移を実行。

continue: ネット上のすべてのシステムのダイナミクスを計算。

• SimClock: 時間を司るクラス。クラスメソッド now を送ることにより、どこからでもシミュレーショ

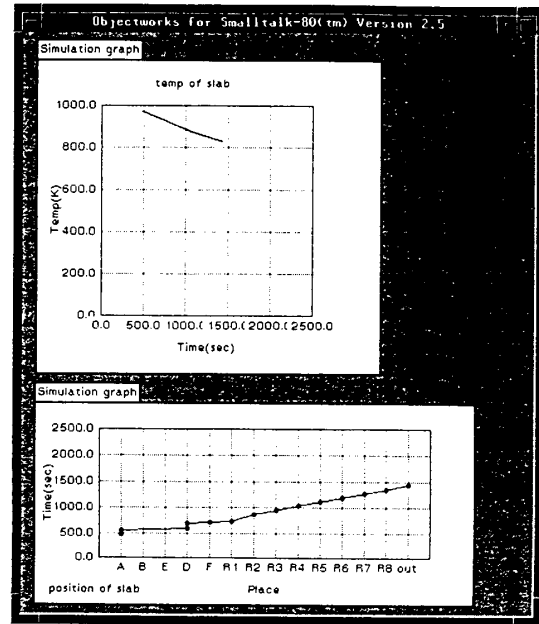


図 12 スラブの挙動

ン時の現在時刻が分かる。

• Netmodel: システムを表すクラスのスーパークラス。インスタンス変数として現行場面を保持している。

- ALine: ローラテーブルを表す。
- BufferA: バッファ A を表す。
- BufferB: バッファ A を表す。
- Slab: スラブを表す。
- Stack: バッファ D, E, F を表す。
- Vehicle: 台車を表す。

図 12 にシミュレーション結果として、あるスラブの位置と温度変化を表すグラフを示す。グラフより、このスラブはバッファ A に入り、その後バッファ D, F を経由してローラテーブルに運ばれた後、出力されていることが分かる。またその間、約 140[K] の温度変化があったことが分かる。このシミュレーションにおける各定数は以下の通り。

$$\sigma = 5.67 \times 10^{-8}$$

$$T_2 = 293$$

$$A_1 = 1.3$$

$$\varepsilon_1 = 0.8$$

また、スラブの比熱 C は次のようにおいた。

$$C = 670 [\text{J/kgK}]$$

7. 結 論

場面のコンセプトを用いた離散・連続混合システムの捉え方、および場面遷移ネットを用いたオブジェクト指向シミュレーションについて述べた。また、実例

として連続鑄造片の加熱炉への搬送ラインのシミュレーションを行った。

本論文で提案したシミュレーション方式は、様々な生産システムのシミュレーションに利用可能であるとともに、新たなプログラミングパラダイムとして、幅広い分野での応用を期待できる。

最後に、資料その他御協力いただきました川合成治氏（富士電機情報システム研究所）に感謝します。

文 献

- (1) Yu-Chi, H., Scanning the Issue, *Proc. IEEE*, (1989), 3-6.
- (2) Paul, A. F., Hetrogeneous decomposition and inter-level coupling for combined modeling, *Winter Simulation Conference*, (1991), 1120-1128.
- (3) 岩澤・三浦・河合・川田, ペトリネットを応用した生産システムの応用方法, 第36回システム制御情報学会研究発表講演会論文集, (1992), 117-118.
- (4) 小林, 移動論, (1973), 202. 朝倉書店.