Vol.2020-MPS-127 No.1
2020/3/2

動的な海上物流計画にコンテナ群再編成で対応する多重 Ambient Calculus を用いたモデル検査

寺尾 侑[†1]　加藤 暢[†1,a)]　樋口 昌宏[†1,b)]

概要：海上コンテナ輸送では，気象状況など様々な要因を想定し，輸送航路や輸送に用いる船舶は中継港に到着時点で動的に決定される．我々は動的な物流計画のモデル化に特化した形式言語である多重 Ambient Calculus(MAC) を用いたモデル検査に関する研究を進めている．本発表では，個々のコンテナに動的に設定される輸送計画を MAC でモデル化する手法を提案する．これは，動的に与えられる輸送計画を元に，多数のコンテナを次に載せる船舶が同一のコンテナ群にグループ化するものである．これにより，モデル検査で一般的に問題となる状態空間爆発を回避しつつ，より正確な検査が可能となる．

キーワード：プロセス代数, モデル検査, Ambient Calculus, 海上コンテナ輸送

A Model Checking with Multiple Ambient Calculus for Dynamic Freight Schedule by Regrouping Containers

Yu Terao[†1]　Toru Kato[†1,a)]　Masahiro Higuchi[†1,b)]

1. はじめに

世界の貿易全体における海上コンテナ輸送の割合 (トン数ベース) は 99%以上であり，全世界で扱われるコンテナの数は年間 1 億 TEU にも及ぶ [1] が，コンテナの積み込み・積み下ろしの最終確認は，コンテナヤード内の係員が主体となって目視で行われている．これに対し，文献 [2] では Ambient Calculus(AC)[3] の拡張である多重 Ambient Calculus(MAC)[4] と RFID 機器を用いた物流監視システムが提案されている．

海上物流には，海の中に港や船が存在し，また，コンテナが船に積み込まれるといった動的な階層構造が存在している．MAC は，このような動的な構造を代数式で表現するために提案されたプロセス代数である．文献 [2] の監視システムは，物流計画全体を MAC のプロセス式でモデル化し，プロセス式の遷移と RFID 機器を用いて検知した実際のコンテナの動きを対比し，コンテナの取り扱いが妥当なものかを監視するものである．

しかし，このプロセス式が実際の物流を正確に反映したものでない場合，それに基づく監視活動も無意味なものになってしまう．これに対し文献 [2] では，時相論理の一種である Ambient Logic(AL)[6], [7] を用いたモデル検査手法が提案されている．一般に，1 つ 1 つのコンテナに対してモデル検査を行うと，あらゆる場合を考慮した場合，容易に状態空間爆発が起こり検査が困難になってしまう．文献 [2] では，弱双模倣等価性 [4] を用いて，同じ荷受港と目的港を持つコンテナをグループ化することにより，多数のコンテナに対するモデル検査を少数のコンテナのモデル検査に帰着させ，この問題を回避している．

[†1] 現在，近畿大学
Presently with Kindai University
a) kato@info.kindai.ac.jp
b) higuchi@info.kindai.jp

しかし，現実の物流では，荷受港と目的港が同一のコンテナ全てが必ずしも同じ経路を辿るとは限らない．例えば，繁忙期等の理由で本来載せるべき船に全てのコンテナを載せることが出来なかった場合，コンテナにつけられた重要度を参照し，重要度の高い順に元の船に載せられる．そのため，荷受港で計画した通りにコンテナ輸送が行われるとは限らない．

そこで本研究では，各中継港で動的に与えられる輸送計画を基にコンテナグループを更に動的に再編成し，実際の物流システムの性質をより正確に検査可能なモデル検査システムを提案する．

2. 多重 Ambient Calculus による物流記述

文献 [4] において提案されている MAC では，n 個のプロセス式の組 $\overline{P} = (P_1, \ldots, P_n)$ で一つの物流計画を表す．以下ではプロセス式 P_i を個別式と予備，$\overline{P}$ を全体式と呼ぶ．

物流計画の記述例

物流計画の表現として，Ambient Calculus を用いて，通常の海上物流に用いられる送り状 (invoice) の記載内容の，1 つの貨物 x を港 y から港 z への輸送を表現した式は以下のように記述できる．

定義 2.1 (Invoice 式).

```
Invoice(x, y, z) ≜
SHIP[ in y.( load[out SHIP.in CY. in x]
           | open lcomp. out y. in z.out z)]
| y[ CY[ x[open load. out CY. in SHIP
. lcomp[out x] ] ] ]
| z[ CY[] ]
```

□

同様に，船の航路のみを表現した式を例えば以下のように記述できる．

定義 2.2 (SHIProute 式).

```
SHIProute≜
SHIP[ in TK.out TK.in KB.out KB.in MJ.out MJ]
| TK[] | KB[] | MJ[]
```

□

定義 2.1 と定義 2.2 を組み合わせた物流計画を表す式を以下のように定義する．

定義 2.3 (物流計画を表すプロセス式).

$\overline{P} = (P_1, P_2, \ldots P_n)$

$P_i = SHIProute_i (0 < i \leq N_0)$

$P_i = Invoice(c_i, S_k, D_k) (N_{k-1} < i \leq N_k)$

$(1 \leq k \leq R)$ □

ここで，N_0 は輸送に用いる船の数，R は積荷港，荷降ろし港で分類した貨物の種類，$N_k - N_{k-1}$ は k 番目の種別の貨物の数を表す．この式は現在の物流計画を表し，式を遷移させることで将来の物流計画の状態を表現することができる．本論文では定義 2.3 のような式を検査の対象とする．

3. 物流システムのための Ambient Logic

モデル検査とは，システムの振る舞いを表現したモデル (プロセス式) が，調べたい性質を表す様相論理の論理式を満たすかどうかを検査することである [8].

3.1 状態遷移グラフ

本節では，与えられた MAC のプロセス式を遷移させて作成される状態遷移グラフについて述べる．このグラフを網羅的に探査することで，以下の項目が満たされているかを検査することが可能となる．

- 到達可能性
  特定の状態に到達する可能性
- 安全性
  意図しない状態が最終遷移状態とならずに動作する可能性
- 不変性
  特定の性質があらゆる状態において常に成り立つ可能性

本検査システムでは，これらの性質をモデルが満たすことを確認するため，図 1 のような状態遷移グラフを生成する．状態遷移グラフの各ノードは，遷移経路を表す実行可能な遷移の情報と，それぞれの遷移経路から到達する子ノード情報，複数のプロセス定義のリストである definition リスト，そして各ノードにおける全体式を構文木として持っている．個別式内の各ノードは，ambient，子 ambient の情報，capability 情報を持つ．この capability 情報と，構文木からなる各 ambient の関係を見ることで，次に実行可能な遷移を見つけ，遷移させることで状態遷移グラフの生成を行う．

図 1 のコンテナの積み込みや積み下ろし順による枝分かれは，文献 [5] のモデル検査システムでも同様に発生するが，図 1 の中継港にて輸送経路を考慮した分岐は文献 [2] で提案されている仕組みである．この状態遷移グラフは，どのコンテナを先に積み込むかといったことや，輸送経路が複数存在する場合において，枝分かれが生じる．

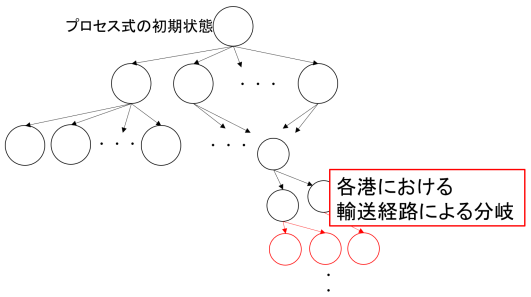

図 1 状態遷移グラフ

$$
\begin{array}{l}
1\ TK[\\
2 \quad CY[\\
3 \quad\quad co1[out\ CY.in\ SHIP] \\
4 \quad] \\
5 \quad |SHIP[out\ TK] \\
6\ ] \\
7\ , \\
8\ TK[\\
9 \quad CY[\\
10 \quad\quad co2[out\ CY.in\ SHIP] \\
11 \quad] \\
12 \quad |SHIP[out\ TK] \\
13\ ]
\end{array}
\tag{1}
$$

式 (1) は，コンテナ 1($co1$ アンビエント) とコンテナ 2($co2$ アンビエント) の 2 つの個別式からなる全体式である．ここでは，説明のために簡略化したプロセス式を用いるが，本来は制御アンビエントなどを用いて船の到着通知を港やコンテナに行うことで，コンテナや船が移動するような同期制御を取り入れた式に対する検査を行う．

式 (1) は，コンテナ 1 とコンテナ 2 が東京港 (TK アンビエント) のコンテナヤード (CY アンビエント) に搬入されている状態を表している．そして，コンテナ 1 とコンテナ 2 を船に載せることで船は東京港を出発する．

ここで，$co1$，及び $co2$ は，$out\ CY.in\ SHIP$ が実行可能な状態である．そのため，$co1$ が先に船に $SHIP$ に入る場合と，$co2$ が先に船に入る場合の 2 つの場合分けが存在するため，子ノードは 2 つの分岐が存在する．このようなコンテナが多く存在するほど，コンテナの積み込み順は爆発的に増加し，例えば 1000 個のコンテナの積み込み順は 1000! 通り存在してしまう．また，次の輸送経路が複数考えられる def の展開時も各輸送経路に対して網羅的に検査する必要がある．そのため，使用する船や次の中継港の違いによる枝分かれが生じる (図 1 の各輸送経路による分岐)．この枝分かれは，各輸送経路に適した definition リストをそれぞれの子ノードに与えることによって表現される．

ここでいう definition リストとは，それぞれの港において同じ定義名で定義内容が異なる definition が複数定義されたもののことを意味する．これらは同じ定義名ではあるが，それぞれの異なる航路を表した式になっており，これらを使用することによってコンテナの輸送経路を動的に変化させることができる．

このような枝分かれによって，コンテナが n 個ある場合，深さが 1 深くなると $n!$ 個ほどのノードが生成され，深さにつれてノード数が膨大となり状態空間爆発が発生してしまう．そこで，本研究では文献 [5] のように弱双模倣等価性を用い，状態空間爆発の抑制を図る．

弱双模倣等価性 $\simeq$ 義は文献 [4] に譲る．

直感的に $\overline{P} \simeq_{\mathcal{O}} \overline{Q}$ ならば，$\mathcal{O}$ のみが観測可能であるとしたときの $\overline{P}$ から実行可能な観測可能イベント系列の集合と，$\overline{Q}$ から実行可能な観測可能イベント系列の集合が一致する．

本検査プログラムでは，弱双模倣等価の観測可能なイベント集合 $\mathcal{O}$ は，大域遷移によるイベントの集合とし，それ以外を観測不能なイベントの集合とする．

3.2 不変式を用いたプロセス式のモデル検査

多重 Ambient Calculus の全体式の無限または有限集合 $\mathcal{F} = \{\overline{P}_{(1)}, \overline{P}_{(2)}, \ldots\}$ に対して，ある全体式 $\mathcal{I}$ が $\mathcal{F}$ 中の任意のプロセス $\overline{P}_{(i)}$ に対して $\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} \overline{P}_{(i)}$ であるとき，$\mathcal{I}$ を $\mathcal{F}$ の不変式という．論理式 f に対して, $\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} \overline{P}_{(i)}$ ならば $\mathcal{I} \models f$ と $\overline{P}_{(i)} \models f$ が同値であることが保証されれば, $\mathcal{F}$ に含まれる任意の全体式 $\overline{P}_{(i)}$ のモデル検査 $\overline{P}_{(i)} \models f$ を $\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} (\mathcal{I}, P_{(1)})$ と $\mathcal{I} \models f$ の検査に帰着できる．不変式 $\mathcal{I}$ に対して $\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} (\mathcal{I}, P(i))$ が成り立つならば，$\simeq_{\mathcal{L}(\mathcal{I})}$ の推移性と数学的帰納法を用いて，任意の n について $\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} (\mathcal{I}, P(i), \ldots, P(n))$ が成り立つ．すまわち $\mathcal{I}$ は $\mathcal{F} = \{(\mathcal{I}, P(i), \ldots, P(k)) \mid k \in \mathcal{N}\}$ の不変式である [5].

本検査では，次に載せる船が同一のコンテナを 1 つのグループとして編成し，代表元を 1 つ取り出してその代表元がそのグループに所属するコンテナ全てと弱双模倣等価関係であるかを確認することで代表元を不変式とする．各グループの不変式を求めることで，多数のコンテナの検査を少数のコンテナの検査に帰結することができ，モデル検査時の状態空間爆発の抑制を図る．

3.3 Ambient Logic

本研究で提案するモデル検査システムでは，3.1 節で説明した状態遷移グラフを用いて，プロセス式生成システムにより生成された式が物流システムの所期の性質を満たしているか検証するために，様相論理の一種である Ambient Logic[6] を用いる．

文献 [5] では，物流計画を表現した MAC のプロセス式に対して，以下の性質を満たしているか確認している．

定義 3.1 (対象とする性質).

p1 いつか必ず貨物 *(コンテナ)* は目的地に輸送される．

$$\Box\Diamond(\blacklozenge co[T] \Rightarrow \blacklozenge PORT\_B[CY[co[T] \mid T] \mid T]). \tag{2}$$

p2 特定の場所以外での貨物 *(コンテナ)* の積み下ろしは行われない．

$$\Box(\blacklozenge co[T] \Rightarrow \neg co[T]|T) \quad (3)$$

.

本研究では，コンテナを載せることが可能な船が多数存在する中，どれかを選択してコンテナ輸送を行う物流計画を対象としている．本研究では p1 と p2 の性質を満たしているか確認する．

これらの性質を満たすことを確認するため，以下の Ambient Logic を用いた様相論理式をモデル検査システムに与える．

(2) 式の $CY[co[T] \mid T]$ の部分はコンテナヤード CY の中にコンテナ co が存在しており，また $co[T] \mid T$ でコンテナヤード CY の中に，さらにコンテナ co 以外が存在してもよいことを示している．$\blacklozenge PORT\_B[CY[co[T] \mid T] \mid T]$ は，プロセス式のどこかで $PORT\_B[CY[co[T] \mid T] \mid T]$ という階層構造が存在することを示している．この式の前に $\blacklozenge co[T] \Rightarrow$ をつけることにより，全体式 $\{P_1, \ldots, P_n\}$ の中で co アンビエントを持つ個別式 P_i に対してのみ上記の条件の成立を検査することになる．さらにこの式に F をつけることで，全てのパス上のどこかでその状態が必ず成り立つことを示している．

p2 の性質は，コンテナを積み込む直前または積み下した直後は $(PORT\_A[\blacklozenge co[T] \mid T])$ が成り立ち，積み込み，積み下ろしができるのは，それぞれ港の中でしか行われないなので“コンテナが船や港より外に存在することはない”と言い換えることができ (3) 式で表すことができる．

4. コンテナ群再編成によるモデル検査

文献 [2] のモデル検査システムでは，荷受港で渡される Shipping Order List(SOL) を参照し，荷受港の時点で目的港が同一のコンテナを 1 つのグループとし，各グループの代表元をそのグループの不変式としてモデル検査を行っている (図 2)．

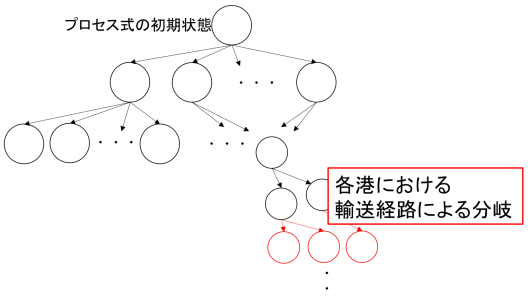

図 2　文献 [2] のモデル検査システム

検査の際には，definition を用いてその物流計画で考えられる全てのコンテナの移動パターンを網羅的に検査している．

本モデル検査システムでは，各中継港で図 3 の様に同じ目的港別にグループ化したコンテナを更に次に載せる船が同一のコンテナに細分化して再編成する．

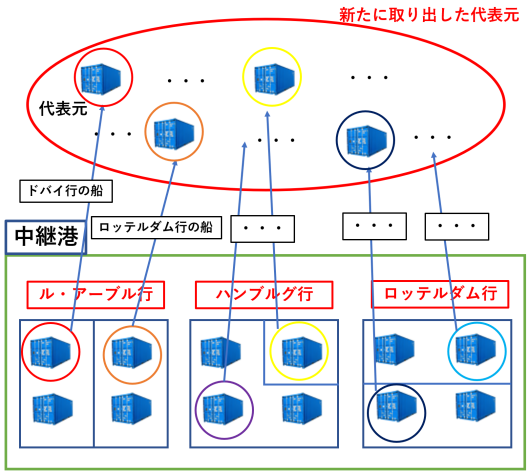

図 3　中継港で行うコンテナの再編成

各中継港で行う再編成には船が出港して次の港に着く少なくとも 3 日前までに発行される SOL を参照して，次にどのコンテナがどの船に載せるかを確認する．MAC のプロセス式では，$out\ PORT$ という capability を消費した際に，definition を展開したプロセス式を確認し，どのコンテナがをどの船に載せるかを判断する．

文献 [2] では，荷受港で物流計画を検査する際に使用する船とコンテナが移動する様々な可能性を考慮した検査を行っているが，これは最初に definition という形でどの船をどのコンテナに載せるかといった計画を定義していた．本検査システムでは，より現実の海上物流に即するため，最初に与える definition には荷受港で計画されている船のみを定義し，船が出港した際に次の港でどの船を使用するか，またはどの船にどのコンテナを使用するかといったことを定義する definition を生成する．物流計画を実行する際に，非決定的にどの船を使用するか決めることが可能ではあるが，MAC のプロセス式で記述する場合にそれぞれの船に対して唯一の名前を付けてその船を識別するため，予め使用する船は名前を決めておかなくてはならない．本モデル検査システムでは物流計画において，どのような航路用いても最終的には監視対象のコンテナは目的港に辿り着くことを検査するために，図 4 のように，使用する船の名前とその航路を与えておく．

本モデル検査システムは各中継港で以下の工程を踏まえて検査を行う．

(1) 前の港もしくは荷受港で編成したコンテナ群から，次に載せる船が同一のコンテナで再編成する
(2) 再編成したグループから代表元を 1 つ取り出し，その代表元が属するグループのあるコンテナと弱双模倣等価関係であることを確認し，それを不変式とする
(3) 生成した不変式から状態遷移グラフを作成する

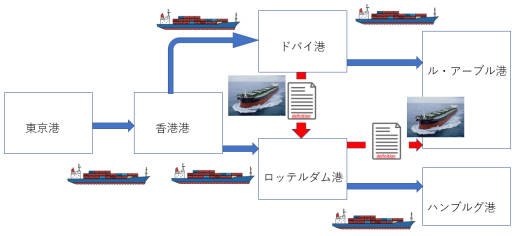

図 4　予め与えておく計画外の船

(4) その状態遷移グラフが物流システムが満たすべき性質を満たしているかどうかを検査する

1，2 では，各中継港では再編成して物流計画の修正に対応した後，新たにグループの代表元を選出し，それらの代表元が所属するグループのあるコンテナと弱双模倣等価関係であるか確認し，確認が出来たらその代表元をそのグループの不変式として状態遷移グラフを作成する．この状態遷移グラフを生成する際には，コンテナの航路も変更されるため，計画内で使用される予定であった definition を書き換える．

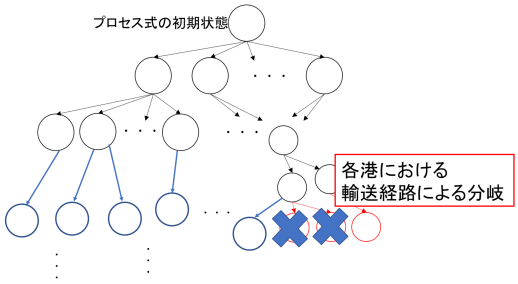

図 5　本モデル検査システムの状態遷移グラフ

3 において生成する状態遷移グラフは図 5 のようになる．definition を書き換えるため，前の港で生成した状態遷移グラフで遷移する可能性があった状態が消失することになり，代わりに新たな状態が生まれる．

ただし 1， 2 は，実際の現場ではその必要がある場合のみ行う．何らかの要因が発生しなかったり，荷受港で計画していた通りに輸送が成されている場合は，検査時間の短縮のために，コンテナの再編成や不変式の生成及びそれに伴う状態遷移グラフの生成を行わない．これは前の港や荷受港でモデル検査を行い，船やコンテナの取り扱い及び MAC のプロセス式の遷移が妥当であると判断されているからである．

4 では，生成した状態遷移グラフに対して物流システムが満たすべき性質を満たしているかを検査する．物流計画を表現した MAC のプロセス式の状態遷移グラフと物流システムの性質を AL で表現した論理式は，*p1*，*p2* を満たすかどうかで検査を行う．*p1* は荷受港で監視対象とした全てのコンテナが最終的に目的港内のコンテナヤードに到達するという性質を表し，*p2* はコンテナ必ず港内で積み込み・積み下ろしの動作が行われるという性質を表している．そのため，書き換えた definition がこの工程に影響を及ぼすことはない．4 も 1，2 と同様に，コンテナの再編成が必要ない場合には行わない．

5.　実験

本モデル検査システムを用いて図 6 の航路を用意し，その正当性を検証した．

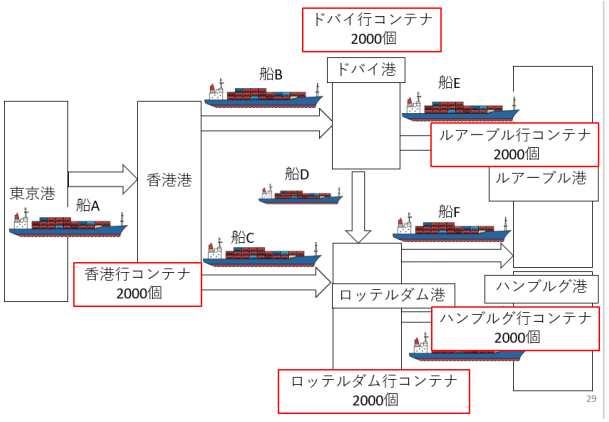

図 6　実験に使用した航路

実験として，荷受港時点の計画では東京から香港へは全てのコンテナが船 A に載せて運ぶ．また，ルアーブルに向かうコンテナは香港とドバイを経由してルアーブルに辿り着く．

ここで，香港にて船 C にロッテルダムやハンブルグに向かうコンテナ全てを載せられないという事象が発生し，計画とは異なる船に載せられるというように，動的に計画を変更するようにした．船 C にはロッテルダム行のコンテナが半分，ハンブルグ行のコンテナが半分積み込まれ，反対に船 B にはロッテルダム行，ハンブルグ行のコンテナが同数積み込まれる．また，船 B には本来載せる予定のなかったコンテナが載せられたことにより，ルアーブル行のコンテナが全て載せられなくなり，船 C に 800 個載せるようにした．次のドバイやロッテルダムでも同様に目的港にたどり着くように船の載せ替えを行った．最終的にルアーブルには，東京-香港-ドバイ-ルアーブル，東京-ドバイ-ロッテルダム-ルアーブル，東京-ロッテルダム-ルアーブルという 3 通りの航路を使って到達するように実験のための計画を設計した．

これらの動的な計画の変更は各港で definition を用いて動的に経路を与えて実験を行った．

検査に掛かった時間は，コンテナの再編成，不変式の生成，状態遷移グラフの生成，モデル検査全て合わせて約 72 秒であった．

また，計画が物流システムを満たさないという計画であった場合，満たさないという旨を表示することが出来る

かという図 7 実験も行った．

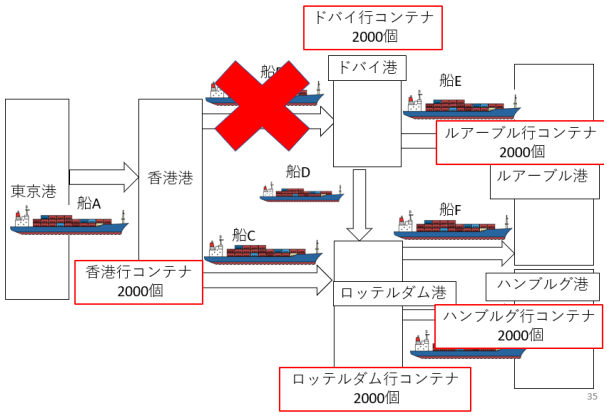

図 7 航路を断って目的港に到達しないコンテナが存在する計画

香港からドバイに向かう船を記述せずにプロセス式を実行させ，モデル検査を行った．最終状態では，ドバイ以外の港では全てのコンテナを運ぶことが出来たが．しかしドバイ行のコンテナは唯一ドバイに向かう船が断たれたことにより到達できなくなり，ドバイに到達できない旨を表示してモデル検査は途中で終了した．これは物流システムが満たすべき，全てのコンテナが目的港に到達するという性質を満たしていないと判定したからである．この検査の掛かった時間は，すべての検査を合わせて約 54 秒であった．

6. まとめ

本研究では，文献 [2] のモデル検査システムでは検査出来なかった，より現実の物流に即した物流計画を検査できるモデル検査システムを構築した．このモデル検査システムでは，目的港以外の寄港する全ての港でモデル検査を行ったため，文献 [2] のモデル検査システムを大幅に改良することになった．

また，より複雑な計画に対するモデル検査を行う際に，検査時間が増大するため，モデル検査アルゴリズムを改良して検査時間の短縮を図る必要があると思われる．

参考文献

[1] SHIPPING NOW 2019-2020,
公益社団法人 日本海事センター,
http://www.jpmac.or.jp/img/relation/pdf/2019pdf-full.pdf(参照 2019-12-23).

[2] 加藤暢，高岡久裕，樋口昌宏，大山博史: 多重 Ambient Calculus を用いた動的な海上物流に対するモデル検査, 情報処理学会論文誌, Vol. 117, No. 12, pp. 1-6, 2018.

[3] Cardelli,L. and Gordon,A.D.: Mobile Ambients, Theoretical Computer Science, Vol. 240, pp. 177-213, 2000.

[4] 樋口昌宏, 加藤暢: 物流システム記述のための多重 Ambient Calculus, 情報処理学会論文誌プログラミング (PRO): Vol.5, No.2, pp.79-87(2012)

[5] 樋口昌宏, 森田哲平, 加藤暢: 多重 ambient calculus による物流記述に対する弱双模倣等価性を用いたモデル検査, 情報処理学会論文誌, Vol. 5, No. 3, pp. 50-60, 2012.

[6] L. Cardelli, A.D. Gordon. Any time anywhere: Modal logics for mobile ambients. POPL 2000 Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, pp. 365-377, 2000.

[7] Model checking mobile ambients. Witold Charatonik, Silvano Dal Zilio, Andrew D Gordon, Supratik Mukhopadhyay, and Jean-Marc Talbot. Model checking mobile ambients. Theoretical Computer Science, Vol. 308, No. 1-3, pp.277-331, 2003.

[8] 吉岡信和, 田辺良則, 田原康之, 長谷川哲夫, 磯部祥尚. モデル検査による設計検証. 日本ソフトウェア科学会学会誌：コンピュータソフトウェア, Vol. 31, No. 4, pp. 40-65, 11 2014.