情報処理学会論文誌　数理モデル化と応用　Vol.11　No.3　84–99 (Dec. 2018)

多重Ambient Calculusを用いた動的な海上物流計画に対するモデル検査

加藤 暢[1,a)]　高岡 久裕[2]　樋口 昌宏[1]　大山 博史[3]

受付日 2018年7月5日，再受付日 2018年9月4日，
採録日 2018年9月17日

概要：海上コンテナ輸送では，気象や貨物量の急激な変化など様々な要因を考慮し，使用する船や中継港などは輸送の途中で動的に決定される．我々は，このような動的な物流計画を対象とし，コンテナ取扱の妥当性を自動的に判定する監視システムを開発している．これは，物流計画をプロセス代数の一種である多重 Ambient Calculus でモデル化し，RFID 機器を用いてとらえたコンテナの挙動とモデルを比較することでコンテナ取扱の妥当性を判定するものである．本稿ではまず本監視システムの実装について述べる．ところで，自動的に生成されるモデルが物流計画の内容を正確に反映していない場合，この監視活動は無意味なものとなってしまう．そこで本稿では，このモデルが，動的に変化する物流システムの所期の性質を満たすことを自動的に確認するモデル検査システムを提案する．

キーワード：プロセス代数，Ambient Calculus，海上物流システム，モデル検査

Model Checking for Dynamic Freight Schedule with Multiple Ambient Calculus

Toru Kato[1,a)]　Hisahiro Takaoka[2]　Masahiro Higuchi[1]　Hiroshi Ohyama[3]

Received: July 5, 2018, Revised: September 4, 2018,
Accepted: September 17, 2018

Abstract: Vessels and hub ports used in maritime logistics of container are dynamically determined depending on various factors such as weather condition or the sudden increase of amount of containers. Aiming supervising such dynamically changing freight plans, we are developing a freight management system that can not only monitor the handling of containers with RFID devices but also confirms the correctness of it by modeling whole the freight plans with Multiple Ambient Calculus that is a kind of process algebra. This paper shows the implementation of the system. If the model includes wrong information, however, the system would be useless. Thus, this paper also proposes a model checking system that confirms the model satisfies the properties expected to the freight plan that dynamically changes.

Keywords: Process Algebra, Ambient Calculus, Maritime Logistics, Model Checking

1. 序論

1.1 コンテナ輸送と管理について

近年のコンテナ貨物増加にともない，政府機関や多数の物流関連団体がRFIDタグなどの電子機器を用いた自動的なコンテナ管理に関する研究を行っている．特に2006年からフェーズ1として始まった「国際物流における電子タグ実証実験」は，コンテナのトレーサビリティ，セキュリティ，サプライチェーンマネジメントの効率化を目的としたものであり，経済産業省と日本物流団体連合会に所属する多くの大手海運会社により実施されてきた[13]．2008年に行われたフェーズ3では，日本とオランダ間の倉庫までの貨物の追跡実験が行われている．また，コンテナ貨物のグローバルコンテナトラッキングシステム技術

[1] 近畿大学大学院総合理工学研究科
Kindai University, Higashiosaka, Osaka 577–8502, Japan
[2] 株式会社日立システムズ
Hitachi Systems, Ltd., Shinagawa, Tokyo 141–8672, Japan
[3] 国立高等専門学校機構広島商船高等専門学校
NIT Hiroshima College, Toyoda, Hiroshima 725–0231, Japan
[a)] kato@info.kindai.ac.jp

の普及のための標準化が提案され，その中で韓国や中国で進んでいるグローバルトラッキングシステムの調査結果が報告されている [25]．さらに，2014 年に，東芝ロジスティックスによる RFID タグを用いたコンテナの追跡実験が中国大連港と横浜港の間で行われたことが報告されている [23]．これら以外にも様々な研究・実験が報告されている [18], [19], [20], [21]．

このように，コンテナの輸送を RFID タグにより管理する仕組みは近年高い関心を集めており，一部は実際に運用が開始されている．これらに共通する考え方は，タグの ID とコンテナ情報をデータベースなどで紐付けしておき，リーダで読み取った ID をネットワークを介してデータベースを持つサーバに送り，個々のコンテナの動きをサーバ側で監視するというものである．したがってこれらは，無線 LAN やインターネットなどネットワーク環境が常に活用できることを前提とした仕組みになっている．

これに対し我々は，コンテナ情報の管理を中央のサーバに頼るのではなく，タグ（研究初期にはバーコード）そのものに書き込んだ情報をもとにコンテナの取扱いを監視する方法を研究してきた [5], [6], [8], [9], [12]．

1.2　物流システムのモデル化と分散型管理

我々が提案しているこれらの方法は，物流システム全体を Ambient Calculus（AC）[2] またはその拡張である多重 Ambient Calculus（MAC）[7] のプロセス式でモデル化し，プロセス式の遷移としてコンテナの動きを表現するものである．タグに書き込む情報はコンテナの動作を表すプロセス式であり，RFID リーダで読み取ったコンテナの式を船や港などを表すプロセス式と並行合成し，遷移可能かどうかを確認することでコンテナの取扱いの正しさを確認する．これにより，コンテナターミナルで行われている，年間数十万個にも及ぶ目視によるコンテナの積み込み・積み下ろしの確認作業を自動化することを目指している．

AC は，動的な階層構造を代数式で表現することができ，ネットワーク間を移動するようなモバイルプロセスの記述に特化したプロセス代数である．一方海上物流にも，大きな枠組みであるコンテナ船が小さな枠組みであるコンテナを積んでいるという階層構造が存在している．そしてその階層構造は，船がある港から別の港へと移動したり，コンテナの積込みや積降ろしをするとき動的に変化する．

我々は AC と物流システムの双方が持つ動的な階層構造という類似点に着目し，文献 [12] において，物流書類の内容を AC のプロセス式を用いて表現し，実際の物流がそのプロセス式の記述どおりに行われているかを監視するシステムを提案した．さらに，現実の物流で頻繁に生じるコンテナの追加，キャンセルなど動的な変更にも柔軟に対応できるモデル化を行うため，我々は文献 [7] において MAC を提案し，文献 [5] では MAC を用いた物流監視システムを提案した．このシステムは，エクセル形式で記述された物流書類から MAC のプロセス式を生成する機能，MAC の式を遷移させることのできる処理系，および RFID 機器を用いて検知した実際のコンテナの取扱いをプロセス式の遷移と対比させ，その取扱いが正しいものであるかどうかを監視する機能からなる．

1.3　MAC によるモデル化の利点

物流システムのモデルに基づく監視は，しかし自動的に生成されるモデルが物流計画の内容を正確に反映していない場合無意味なものとなってしまう．たとえば，本来載せられるべきでない船にコンテナが載せられようとしている際にその取扱いを承認するようなプロセス式では，本来の役割を果たすことができない．このような誤りが生じる原因は主に 2 つあり，1 つはプロセス式生成機能に内在する誤り，もう 1 つは物流書類に内在する人為的な誤りである．特に後者は，我々の提案する以外の方法を用いた監視システムでも発生することが考えられる．

このような問題を輸送が始まる前に防ぐ方法として，我々は時相論理の一種である Ambient Logic（AL）[1] を用いたモデル検査手法を提案した [6], [8]．AC や MAC では，時間経過に沿って生じるプロセス式の変化は，モノが他のモノに入ったり出たりといった空間の変化を表している．そのような性質を表現するために AL には，いつか，ずっと，といった時間に関する様相記号に加え，どこかで，どこでも，といった場所に関する様相を表現するための記号が用意されている．これらの様相記号を使った論理式によって，たとえば「いつか必ずこのコンテナは目的地に輸送される」あるいは，「特定の場所以外でのコンテナの積み下ろしは行われない」といった，物流システムに求められる正確性，安全性に関する性質が簡潔に記述できる．文献 [6], [8] では，物流システムに求められるこのような性質を AL によって記述し，AC または MAC でモデル化された物流システムがその性質を満たすかどうかを確認する方法を提案した．

1.4　動的な物流計画に対応した監視機能の提案（目的 1）

船の乗換えを含むコンテナの海上輸送では，コンテナが次に載せ換えられる船は運送途中で動的に決定される．文献 [5] で提案した物流監視システムでは，コンテナが港に到着した時点で次に乗り換えるべき船を，港を表すプロセスから動的に取得する仕組みは実装していた．ただしそのプロセス式はあらかじめ用意してあるものだけであり，以下に述べる実際の物流システムの持つ性質に充分対応できるものではなかった．

コンテナターミナル運営会社，フォワーダー会社，大手船会社にインタビューによる調査を行った結果，単に天候の影響だけではなく，繁忙期など貨物の量が急激に増加す

る場合には，本来載せ換えられるはずであった船が利用できなくなり，別の船が割り当てられる場合も少なくないことが分かった．その際にどの船を選択すれば良いかなどの意思決定を支援するソフトウェアが利用されているが，最終的にどのコンテナをどの船に優先して載せるか，後回しにするかなどの決定は様々な要因を考慮し人間が行っている．このようにコンテナの輸送経路は，様々な要因の元で動的に決定され，コンテナターミナルには，コンテナが到着する 1 日前までにその情報が，2.1 節で述べる船積み依頼リストという書類によって船会社から伝えられる．コンテナターミナルでは船積み依頼リストに基づいて，コンテナの積み込み，積み下ろしのリストが生成され，そのリストを基に積み込み積み下ろし作業や確認作業が行われている．

このようなコンテナ輸送をモデル化し，動的に設定される複雑な経路をたどるコンテナの取扱いを正しく監視するためには，モデルの自動生成機能も動的な変更に対応させる必要がある．本稿の目的の 1 つは，上記のようなコンテナの乗換えに関する問題に対応できる仕組みを文献 [5] で提案した物流監視システムに取り入れ，より現実に即した監視ができるシステムを提案することである．

1.5　動的な計画に対応したモデル検査の提案（目的 2）

文献 [6], [8] で提案したモデル検査手法は，物流システムに求められる性質をあらかじめ生成済みのプロセス式が満たすかどうかを確認するものであった．これらの手法では，プロセス式を遷移させすべての非決定的な選択を表す遷移グラフを生成し，そのグラフを深さ優先で探査し，各ノードの全体式に対して非時相式の検査を木の親子関係を見ながら行い，sometime modality や everytime modality のような時間的な検査を経路を見ながら行っている．特に文献 [6] では，MAC に対して定義される弱双模倣等価性 [7] を利用したグラフの枝刈りを行うことで状態空間爆発問題を回避する方法を提案している．本稿のもう 1 つの目的は，文献 [6] で提案したモデル検査手法を拡張し，動的な物流計画に対応できるモデル検査手法を提案することである．

1.6　本稿の構成

2 章では，本稿で対象としている物流システムについて述べる．3 章では本研究で基礎としている AC と MAC について，その構文規則，遷移規則および MAC による物流システムの記述例について述べる．4 章では，MAC と RFID 機器を用いて本研究で構築してきた物流監視システムについて説明した後，本稿で提案する機能拡張について述べる．5 章では，本稿で提案する動的な経路設定機能について述べる．6 章では，物流システムに期待される性質を記述するための論理である AL について説明する．7 章では，本稿で提案するモデル検査システムについて説明する．8 章では，本稿で提案する物流監視システムおよびモデル検査システムの実現性を示すために行った実験について述べる．9 章では，AC を用いたモデル検査について，関連研究と本稿との比較を行う．2，3，4（前半），6 章が本研究の背景や各種定義にかかわる部分，4（後半），5，7，8 章が本稿で新たに提案する機能にかかわる部分である．

2.　海上物流システム

本章では，本研究で対象とするコンテナを用いた海上物流システムについて説明する．コンテナを用いた海上物流システムは，大きく LCL（Less than Container Load）と FCL（Full Container Load）に分類される．前者は 1 個のコンテナを複数の荷主が利用する形態，後者は 1 個のコンテナを 1 人の荷主が利用する形態である [22]．本研究ではその構造の単純さから後者を対象としている．

2.1　ハブ&スポークシステムと物流定義書類

海上物流の分野では近年ハブ&スポークシステムと呼ばれる輸送システムが普及している．これは海運会社間の提携関係（アライアンス）の下で，各海運会社の拠点港をハブ港，ハブ港の周りのローカルな港をフィーダ港とし，ハブ港を中心とした輸送を行うことにより，効率的な輸送を行うシステムである [22]．本稿ではハブ&スポークシステム上で複雑な経路をたどるコンテナ輸送を対象とする．

物流システムの動作は書類を使って規定され，その書類に基づいてコンテナの取扱いあるいは船の運航などの作業が行われる．本節では，物流に用いられる書類と，それに基づく作業について説明する．

荷主はコンテナの仕出港と仕向港[*1]を指定した送り状を作成し，それに基づいて船会社は B/L Instructions と呼ばれる書類を作成する．B/L Instructions には，輸送に使用されるコンテナの個数と各コンテナのコンテナ番号のほか，仕出港，仕向港，コンテナが積み込まれる船といった情報が記載されている．また，船会社は各船舶の航路表を作成し，それに従って船の運航が行われる．航路表には船を識別するための船名と航海番号，その船が寄港する港が寄港順に記述された一覧が記載されている．これらの書類を集約し，最終的には船積み依頼リストと呼ばれる指示書が船会社で作成され，コンテナターミナルに送られる．**図 1** に示す船積み依頼リストの例は，コンテナ船（SHIP OTSUKAMARU，航海番号（VOY NO）v2 で識別）で運ばれてきたコンテナの積み替え情報が示されている．この中では，香港港を仕向港（for HONGKONG）とするコンテナ（container # KATU1875606 などで識別）が，中継

[*1] コンテナの出発する港と目的地の港をそれぞれ仕出港と仕向港，それ以外の港を中継港と呼ぶ．フィーダ港が仕出港や仕向港となる場合が多いが，ハブ港が仕出港や仕向港となることも少なくない．

for	HONGKONG						
at	KOBE		SHIP	OTSUKAMARU		VOY NO.	v2
Booking#	Container#		NEXT SHIP	Last POD		B/L#	
7180076560	KATU1875606		SANADAMARU	HONGKONG		1345	
7180076560	KATU1875607		SANADAMARU	HONGKONG		1595	
7180076560	KATU1875608		SANADAMARU	HONGKONG		2354	

図 1　船積み依頼リストの例
Fig. 1　Sample of shipping list.

港である神戸港（at KOBE）で，次に SANADAMARU に載せ換えられることが示されている．

船積み依頼リストの作成時には，そのコンテナを次にどの船に載せてどの港まで運ぶかを船会社が決定する．この決定が適切に行われることによって，効率的な物流が可能となる．文献 [5] では，船積み依頼リストがあらかじめ与えられているものとして監視システムを構築した．

しかしハブ&スポークシステムでは 1.4 節で述べたように，中継港においてあるコンテナが載せ換えられると予定されていた船が，天候の影響で到着しない，あるいは，季節による他社の臨時のコンテナにより満杯になってしまうなどの影響で，当初予定の船とは別の船に載せなければならないといった事態も希ではない．したがって船積み依頼リストは輸送の始まる前に作成されるのではなく，コンテナが中継港に到着する前日までに最新の運行状況を基に船会社により作成され，コンテナターミナルに送られる．

ハブ&スポークシステムにおいてこのように動的に決定される輸送計画に対応できる監視システムの実現が，1.5 節で述べた目的 1 である．

3.　多重 Ambient Calculus（MAC）

文献 [9], [12] では，Ambient Calculus（AC）を用いて物流システムのモデル化を行っていた．AC は，Microsoft Research の Luca Cardelli と Andrew D. Gordon によって開発されたプロセス代数であり，動的な階層構造を持つシステムを形式的に表現することができる．この特徴を活かして，ネットワーク上で動的に情報をやり取りするための様々な実装が提案されている [14], [16]．またこの特徴により，物流システムの持つ階層構造を簡潔に表現できる．

しかし実際の物流システムには，指定されたコンテナが積み込まれた後に船が出航するなどの同期的な性質があり，それらを AC だけで表現するとプロセス式の構造が複雑になってしまう．また，コンテナの追加やキャンセルといった動的な貨物量の変化に対応するには，プロセス式の書き換えが必要になるなど，AC 単独で物流システムをモデル化することが困難であったため，我々は文献 [7] において MAC を提案した．

本章では MAC の基礎となっている AC の構文規則と遷移規則について述べた後，MAC について説明する．

3.1　AC に関する諸定義

構文規則および遷移規則は文献 [2] で以下のように定義されている．

定義 3.1（AC の構文規則）

$P, Q ::=$	*processes*	$M ::=$	*capabilities*
$(\nu n)P$	restriction	x	variable
0	inactivity	n	name
$P \mid Q$	composition	*in* M	can enter into M
$!P$	replication	*out* M	can exit out of M
$n[P]$	ambient	*open* M	can open M
$M.P$	action	ϵ	null
$(x).P$	input action	$M.M'$	path
$\langle M \rangle$	output action		

□

定義 3.2（AC の遷移規則）

$n[in\ m.P \mid Q] \mid m[R] \to m[n[P \mid Q] \mid R]$	(In)
$m[n[out\ m.P \mid Q] \mid R] \to n[P \mid Q] \mid m[R]$	(Out)
$open\ n.P \mid n[Q] \to P \mid Q$	(Open)
$(x).P \mid \langle M \rangle \to P\{x \leftarrow M\}$	(Comm)
$P \to Q \Rightarrow (\nu n)P \to (\nu n)Q$	(Res)
$P \to Q \Rightarrow n[P] \to n[Q]$	(Amb)
$P \to Q \Rightarrow P \mid R \to Q \mid R$	(Par)
$P' \equiv P, P \to Q, Q \equiv Q' \Rightarrow P' \to Q'$	(Struct)

□

これらに加え，AC にはプロセス式の記述を簡略するための書法として definition がある．definition とは，一連の動作を行うプロセスを手続として定義するものであり，プロセス式の先頭に以下のような形で宣言する．

$$def\ f(x_1, \ldots, x_n)\ =\ P;$$

文献 [9] で示した AC による物流システムのモデル化の例を簡略化した式を用いて，定義 3.1，3.2 を直観的に説明する．式 (1) は，1 つのコンテナを東京港で船に積み込み大阪港へ移送するという物流システムを表したプロセス式である[*2]．

$$TK[co1[\ in\ SHIP.\ lcomp1[out\ co1]\ ]] \mid OSK[\ ] \mid SHIP[in\ TK.open\ lcomp1.\ out\ TK.\ in\ OSK] \quad (1)$$

式 (1) は，5 つの ambient：TK，$co1$，$lcomp1$，OSK，$SHIP$ からなっており，$lcomp1$ 以外の 4 つはそれぞれ東京港，コンテナ，大阪港，船という現実のモノを表している．そして，TK，OSK，$SHIP$ が composition 演算子 | で結ばれていることから，各港と船が隣り合った位置に存在し，また，$co1$ ambient が TK ambient の [括弧] の内側に記載されていることから，コンテナが東京港の中に存在しているという物流システム全体の構造を表している．

$SHIP$ ambient の持つ *in* TK capabilitiy に対し定義 3.2

[*2] 実際に使用している式では $TK[CY[co1[...]...]$ のようにコンテナはコンテナヤード内に存在するという物流システムの構成を忠実に表現しているが，ここでは説明を簡単化するため省略している．

の (In) を適用し，式 (1) は式 (2) に遷移する．この遷移は船が東京港へ入港する動作を表している．

$$TK[co1[\ in\ SHIP.\ lcomp1[out\ co1]\ ] \\ |\ SHIP[open\ lcomp1.\ out\ TK.\ in\ OSK]] \mid OSK[\ ] \tag{2}$$

式 (2) の *SHIP* ambient には，東京港から出航するための *out TK* capability が存在するが，その前に *open lcomp*1 が存在することにより *out TK* は実行できない．これは，コンテナの積み込みが完了するまでは船が出航できない，という物流システムの制約を表している．その後コンテナの積み込み動作を表す *co*1 の *in SHIP* が実行され *co*1 が *SHIP* ambient の中に入るという遷移を行い，さらに，*co*1 の中の *lcomp*1 ambient が *co*1 の外に移動するという遷移を経て式 (3) が得られる．

$$TK[SHIP[co1[\ ] \mid lcomp1[\ ] \\ |\ open\ lcomp1.\ out\ TK.in\ OSK]] \mid OSK[\ ] \tag{3}$$

定義 3.2 の (Open) により *open lcomp*1 が実行され，式 (3) は式 (4) に遷移する．これは，検査員がコンテナの積み込みを確認し，船に出航許可を与えたことを表す遷移である．

$$TK[SHIP[co1[\ ] \mid\ out\ TK.\ in\ OSK]] \mid OSK[\ ] \tag{4}$$

式 (4) では *SHIP* ambient の *out TK* が実行可能であるため，*SHIP* は *TK* の外に移動可能となる．以上見てきたように，AC ではプロセス式の構造で物流システムの構造を表し，capablity の実行や制約によって，物流システム内の対象物の動作や対象物間の同期的制約を表すことができる．

式 (1)〜(4) に現れる *lcomp*1 は，実際のモノを表す ambient ではなく，モノとモノの間に課せられる同期的制約を表すために使用される ambient である．このような ambient を**制御用 ambient** と呼ぶ．式 (1) の *lcomp*1 ambient は *in SHIP* に先行されているため，定義 3.2 の (Out) 規則の適用外となり，*out co*1 は実行されない．このように何らかの capability に先行されているもの，およびその子孫の ambient は遷移に関与しない．そのような ambient を**非活性 ambient** といい，そうでないものを**活性 ambient** と呼ぶ．

3.2　MAC の構文規則と遷移規則

ここまで見てきたように，AC により物流システムの構造や動作を表現することは可能であるが，コンテナ数が増えるとその構造が複雑になる．たとえば式 (1) でコンテナをもう 1 つ増やすためには，1 行目に *co*2 ambient を追加し，2 行目の *out TK* の前に *open lcomp*2 を追加するのだが，それら以外にも積み下ろしの制御も含め様々な箇所に制御用 ambient や関連する capability が分散して記述されるため，コンテナの追加やキャンセルといった現実の物流システムで頻繁に起こる変化に柔軟に対応することが困難となる．この問題を MAC では，複数のプロセス式の組で物流システムをモデル化することで回避している [6], [7]．MAC の構文規則である全体式と個別式は文献 [2] で以下のように定義されている．

定義 3.3（全体式と個別式（MAC の構文規則））
$P_1, P_2, \ldots, P_n$ をそれぞれ AC のプロセスとする．このとき $\overline{P} = (P_1, P_2, \ldots, P_n)$ を全体式，各 $P_i (1 \le i \le n)$ を $\overline{P}$ の個別式，あるいは第 i 成分と呼ぶ．　□

簡略化した例を用いて定義 3.3 を説明する．式 (5) は，2 つのコンテナ *co*1 と *co*2 を東京港 *TK* で船 *SHIP* に積み込むという物流システムを表した全体式である[*3]．

$$\begin{pmatrix} TK[co1[\ in\ SHIP.\ lcomp[out\ co1]\ ]] \mid OSK[\ ] \\ \mid SHIP[in\ TK.open\ lcomp.\ out\ TK.\ in\ OSK]\ , \\ TK[co2[\ in\ SHIP.\ lcomp[out\ co2]\ ]] \mid OSK[\ ] \\ \mid SHIP[in\ TK.open\ lcomp.\ out\ TK.\ in\ OSK]\ , \\ TK[\ ] \mid OSK[\ ] \mid SHIP[in\ TK.\ out\ TK.\ in\ OSK] \end{pmatrix} \tag{5}$$

式 (5) の 2, 3 行目がコンテナ *co*1 の，4, 5 行目が *co*2 の，それぞれ東京港での取扱を表す個別式，6 行目が東京港から大阪港へ向かう船の航路を表す個別式である．このように MAC では，各コンテナおよび船ごとに個別式という独立した AC のプロセス式でそれぞれの性質を記述する．これにより，コンテナの追加やキャンセルを個別式の挿入や削除で表現することが可能となる．

MAC では，ただ 1 つの個別式にのみ現れる ambient および制御用 ambient を**個別 ambient** と呼び，複数の個別式に共通して現れる，制御用 ambient 以外の ambient を**大域 ambient** と呼ぶ．式 (5) において，*co*1, *co*2 ambient はそれぞれ第 1 および 2 成分である個別式にのみ現れる個別 ambient である．また *TK*，*OSK*，*SHIP* は複数の個別式に共通に現れる大域 ambient である．以降，大域 ambient 名は先頭に大文字を用いて区別する．

MAC の遷移規則は，文献 [7] で遷移ラベルなどの補助定義を用いて定義されている．

[*3] 第 1，第 2 成分のコンテナがどの船に積まれるかを表す部分は，5.1 節に示すように船積み依頼リストを基に生成される．船の航路にかかわる部分，および船そのものの航路を表す第 3 成分は航路表から生成される．

定義 3.4（遷移ラベル l [7]）

$n[in\ m.P \mid Q] \mid m[R] \rightarrow_l m[n[P \mid Q] \mid R]$,
$l =$ “n enter m” (In)

$m[n[out\ m.P \mid Q] \mid R] \rightarrow_l n[P \mid Q] \mid m[R]$,
$l =$ “n exit m” (Out) □

定義 3.5（遷移ラベル集合 $\mathcal{L}$ [7]）

全体式 $\overline{P}$ に対して，$\mathcal{L}_G(\overline{P})$ を $\overline{P}$ のすべての個別式の遷移ラベルのうち，大域 ambient 名のみからなる遷移ラベルの集合とする．また $\mathcal{L}_I(\overline{P})$ を，$\overline{P}$ 中のすべてての個別式の遷移ラベルのうち，個別 ambient 名を含む遷移ラベルの集合とする．さらに $\mathcal{L}_I^i(\overline{P})$ を，$\mathcal{L}_I(\overline{P})$ の部分集合で $\overline{P}$ の第 i 成分に関する遷移ラベルの集合とする． □

定義 3.6（**MAC** の遷移規則）

（個別遷移規則）

$P_i \rightarrow_l Q_i \wedge l \in \mathcal{L}_I^i(\overline{P})$
$\wedge \mathcal{H}_G(\overline{P}) \neq \emptyset \wedge \mathcal{G}_{A_G}(\mathcal{H}(P_i)) = \mathcal{G}_{A_G}(\mathcal{H}(Q_i))$
$\Rightarrow \overline{P} = (P_1, \ldots, P_i, \ldots, P_n) \rightarrow_l \overline{Q} = (P_1, \ldots, Q_i, \ldots, P_n)$

（大域遷移規則）

$\forall i(P_i \rightarrow_l Q_i) \wedge l \in \mathcal{L}_G(\overline{P})$
$\wedge \mathcal{H}_G(\overline{P}) \neq \emptyset \wedge \mathcal{H}_G(\overline{Q}) \neq \emptyset$
$\Rightarrow \overline{P} = (P_1, \ldots, P_n) \rightarrow_l \overline{Q} = (Q_1, \ldots, Q_n)$ □

定義 3.6 の両規則の 1 行目と 3 行目についてのみ直観的に説明する[*4]．個別遷移規則は，3 行目に示すように全体式の中でただ 1 つの個別式（P_i）のみが変化するような遷移のための規則であり，主にコンテナの移動や移動制限の解除を表すために使用される．個別遷移規則の 1 行目の条件にあるように，遷移ラベルはその個別式の個別 ambient にかかわるものに限定される．

一方大域遷移規則は，3 行目に示すように全体式内の複数の個別式がいっせいに変化するような遷移のための規則であり，主に船の移動を表すために使用される．大域遷移規則の 1 行目の条件にあるように，遷移ラベルは大域 ambient にかかわるものに限定される．式 (5) に示した全体式では，すべての個別式で $l = SHIP$ enter TK という遷移が可能であるため大域遷移が可能であり，その結果式 (6) に遷移する．

$$
\begin{aligned}
&(\\
&\quad TK[co1[\ in\ SHIP.\ lcomp[out\ co1]] \\
&\qquad \mid SHIP[open\ lcomp.\ out\ TK.\ in\ OSK]] \mid OSK[\,]\,, \\
&\quad TK[co2[\ in\ SHIP.\ lcomp[out\ co2]] \\
&\qquad \mid SHIP[open\ lcomp.\ out\ TK.\ in\ OSK]] \mid OSK[\,]\,, \\
&\quad TK[SHIP[\ out\ TK.\ in\ OSK]] \mid OSK[\,] \\
&)
\end{aligned}
\tag{6}
$$

[*4] 両規則の 2 行目に現れる $\mathcal{G}_X$ と $\mathcal{H}_G(\overline{P})$ に関する正確な定義と説明は文献 [7] に譲るが，これらの行の条件は，遷移前および遷移後の全体式で構造に矛盾がないことを表している．

式 (6) では第 3 成分で $l = SHIP$ exit TK という遷移が可能であるが，他の成分では不可能であるため大域遷移は起こらない．その代わり，第 1，2 成分で式 (2)〜(4) に相当する個別遷移が起こり式 (7) に遷移する．

$$
\begin{aligned}
&(\\
&\quad TK[SHIP[co1[\,] \mid out\ TK.\ in\ OSK]] \mid OSK[\,]\,, \\
&\quad TK[SHIP[co2[\,] \mid out\ TK.\ in\ OSK]] \mid OSK[\,]\,, \\
&\quad TK[SHIP[\ out\ TK.\ in\ OSK]] \mid OSK[\,] \\
&)
\end{aligned}
\tag{7}
$$

ここで全個別式で $l = SHIP$ exit TK という遷移が可能となるため大域遷移が可能となり，式 (8) に遷移する．

$$
\begin{aligned}
&(\\
&\quad TK[\,] \mid SHIP[co1[\,] \mid\ in\ OSK] \mid OSK[\,]\,, \\
&\quad TK[\,] \mid SHIP[co2[\,] \mid\ in\ OSK] \mid OSK[\,]\,, \\
&\quad TK[\,] \mid SHIP[\ in\ OSK] \mid OSK[\,] \\
&)
\end{aligned}
\tag{8}
$$

このように MAC では，遷移規則によってすべてのコンテナが積まれた後に船が出航するといった同期的な動作を表現するため，AC のような複雑な記述は不要になる．

4. 多重 Ambient Calculus を用いた物流監視システムとその拡張について

本システムは，コンテナの移動が物流書類の内容を記述した MAC の全体式の遷移に沿ったものであるかを調べるものである．そのために本システムでは，数メートル離れたタグとの通信が可能な UHF 帯 RFID を用いてコンテナの移動を検知する．その目的は以下のとおりである．

- コンテナが輸送計画どおりに移動することを確かめる．
- 計画に反した移動を行った場合に通知する．
- 現在のコンテナやコンテナ船の位置を把握する．

以上の目的を達成するために本システムは以下の機能を備える．

- 物流書類から全体式を自動生成するシステム
- 物を表現する ambient ごとの全体式の分散処理
- 物の動きと全体式の遷移との照合
- 上記関連付けにともなう動作エラーの表示

本章では，これらの機能を実現するために文献 [5] で提案した機器配置について説明した後，本稿で提案する機能拡張について述べる．

4.1 機器の配置と分散処理について

本システムの機器の配置を**図 2** に示す．それぞれの機器の役割は以下のとおりである．

- ゲートをコンテナが通りコンテナヤードへ入ることを

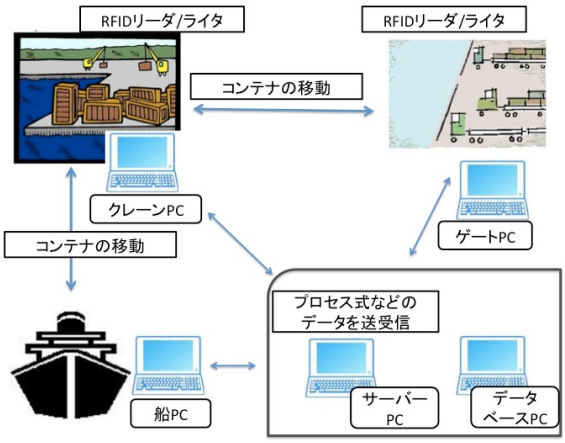

図 2　機器の配置
Fig. 2　System configuration.

監視するためのゲート PC

- コンテナヤードから船へのコンテナの積み込み，および船からコンテナヤードへの積み降ろしを監視するためにクレーンに取り付けるクレーン PC
- 積み込むべきコンテナが積まれてから，あるいは積み降ろすべきコンテナが積み降ろされてから船が出航するといった同期的な動作を監視するための船 PC
- コンテナ式と RFID タグの ID を管理するためのデータベース PC[*5]
- これらの PC を管理するサーバの役割を持つ PC

本システムは，他の機関で研究されている既存の監視システム [13], [23], [25] のように中央のサーバでコンテナ情報を管理する集中型ではなく，上記のような複数の PC を用いる分散型となっている．多くのコンテナ移動に関する通信が 1 つのサーバに集中することを避けるという一般的な分散システムの利点ももちろんあるが，RFID 機器で検知したコンテナ移動の情報をタグで管理し，タグから読み取ったコンテナ式を含む個別式が現実の移動と同じように遷移可能であるかどうかの検査を，中央のサーバには頼らず現場の PC で行うことが分散型にする理由である．本システムの対象は，屋外でのコンテナ積み込み積み降しであり，特に国外のローカル港の PC と継続的に通信可能であるとは限らない．そのため本システムでは，1 つのコンテナを積み込みあるいは積み下ろしするごとに，リーダ/ライタを制御する PC と中央サーバが通信しその妥当性を判断するのではなく，その妥当性を，タグに記述されたコンテナ式を含む個別式に照らし合わせて判断する方法を採用している．

ただしまったく通信を行わないわけではなく，4.2 節や 5.1 節で述べるサーバ PC と各 PC 間の通信は必要である．1 隻に積み込まれる数千から 2 万個におよぶコンテナ 1 つ 1 つの監視のための数千から 2 万回の通信を削減することが主眼である．

本システムを用いたコンテナ取扱いの監視は，以下の順番で実施される．

手順 1) 港への搬入

仕出港の場合：トレーラに積まれたコンテナがターミナルゲートを通過する際，コンテナに貼られたタグの ID を読み取り搬入の妥当性を確認する．

中継港の場合：船からコンテナを積み下ろすの際，コンテナに貼られたタグからプロセス式を読み出し，積み下ろしの妥当性を確認する．

手順 2) 港からの搬出

仕出港の場合：コンテナに貼られたタグの ID をキーとしてデータベースからプロセス式を読み出し，積み込みの妥当性を確認する．

中継港の場合：コンテナに貼られたタグからプロセス式を読み出し，積み込みの妥当性を確認する．

手順 3) 出航の妥当性の確認

積載べきすべてのコンテナが積まれたかどうか，積み下ろすべきコンテナが下ろされたかどうかを確認する．

手順 1)，2) については文献 [5] にて詳述している．手順 3) が本稿で新たに提案する動的な経路決定にかかわる部分であり，文献 [5] に対する拡張である．

4.2　動的な物流計画下での積載確認機能

本節では手順 3) について説明する．実際の海上物流では中継港でコンテナの積み替えを行う場合，船会社から中継港に遅くとも 1 日前までにどの船にどのコンテナを載せるのかが船積み依頼リストによって伝えられる．実際に積み込みが行われる際，船積み依頼リストを基に作られた作業リストを見ながら積み込み確認を人の手で行うため，積み込むべきコンテナがすべて積み込まれてから出港を許可するという同期制御が行われている．

文献 [5] では，乗り換える船があらかじめ決まっていることを想定した実装を行っていた．本稿では，実際の物流システムと同等に，船積み依頼リストによる動的な船の決定に対応できるよう，文献 [5] の監視システムを拡張する．

船 1 で中継港に運ばれてきたコンテナをクレーン 1 で積み下ろし（クレーン PC1 で監視），クレーン 2（クレーン PC2 で監視）で船 2 に載せて他の港へ送り出す場合を想定して説明する．5.1 節で述べるように，本システムでも実際の物流と同じようにその船が中継港に着く前に船積み依頼リストに相当する書類を中継港のサーバ PC へ送る．サーバ PC では，その船積み依頼リストを基に仮のコンテナ式を含む個別式を生成し，この個別式を含めた全体式がサーバ PC からクレーン PC2 に送られる．

[*5] 数千個に及ぶタグにプロセス式を人の手で書き込むことは現実的ではない．そこで，そのタグを貼り付けるコンテナのコンテナ式を含む個別式とそのタグの ID をデータベースに登録しておき，コンテナが最初に船に積まれるときにのみ，タグ ID をキーにしてデータベースからコンテナ式を読み取り，積み込み確認後の式をタグに書き込む．

この仮の式は式 (5) の $co1$ や $co2$ ambient の内部が空になった形をしており，そのため決して積み込みに相当する遷移は起こらず，$out\ TK$ をガードしている $open\ lcomp$ が消費されることもない．したがってこの仮の式がクレーン PC2 にある間は船 2 の出航は許可されない．

コンテナが船 1 によって港に到着しクレーン 1 によって積み下ろされた際，正式な個別式がタグに書き込まれる．正式な個別式は 5.1 節で述べる definition によって与える．

このコンテナが船 2 に積み込まれようとしたときに，クレーン PC2 上で仮の式がタグから読み取った正式な式に置き換わる．その後は式 (6)〜(8) で述べた手順で船 2 が出航可能かどうかが確認される．以上の手順により，動的に決定される船 2 に対してもコンテナの積み忘れを防止することが可能となる．

5. プロセス式自動生成機能の拡張

文献 [5] で提案したプロセス式自動生成機能は，物流監視システムにおいて使用される MAC のプロセス式を，物流書類から読み取った情報を基に自動的に生成するものである．これにより，船の航路をプロセス式で表現した個別式，各コンテナの積み込み，積み降ろしなどの情報を表現したコンテナの個別式が生成され，これらの式からなる全体式によって物流計画全体をモデル化するとができる．船の個別式は，船会社が管理する船の運航表（**図 3**）と，船積み依頼リストに記載されている船の名前から生成される．通常これらの表はエクセル形式で与えられているため，ファイルからの読取りには Apache POI [10] を用いる．

本章では，動的な物流計画下でのコンテナの積み換えをモデル化するために提案する機能拡張について述べる．

5.1 船積み依頼リストの動的な投入と definition

本稿で拡張するプロセス式自動生成機能では，船が中継港に入る前にサーバ PC に船積み依頼リストを送る．サーバ PC では積み替え対象となる各コンテナが次にどの船に載せられるのかという情報を船積み依頼リストから取り出し，4.2 節で述べた仮の式を生成してクレーン PC2 に送る．それとは別に，船積み依頼リストから取り出した情報を基に definition の形をした部分式を生成しクレーン PC1 に送る．そしてコンテナが積み降ろされる際に definition を基に具体的なプロセス式が展開される．definition の例を以下に示す．

$$def\ loadtoHK(co)\ = \\ open\ startLoad.out\ CY.in\ Ship.lcomp[out\ co] \tag{9}$$

式 (9) は，次にどの船に乗るかをコンテナ co に伝えるための definition を簡略化したものである．この definition を利用する個別式の例を以下に示す．

PORT HONGKONG

船名	船種	入港予定時刻	出港予定時刻	全長	総トン数	前港	次港
SHIP_A	一般貨物船	2017/11/26/15:00	–	75m	500トン	TOKYO	–
SHIP_B	一般貨物船	2017/11/26/17:00	–	75m	500トン	TOKYO	–
SHIP_C	一般貨物船	–	2017/11/27/12:00	75m	500トン	–	DUBAI
SHIP_D	一般貨物船	–	2017/12/4/12:00	75m	500トン	–	DUBAI

図 3　コンテナ船運航表の例
Fig. 3　Sample of diagram.

$$OSK[SHIP[\cdots \mid co1[\cdots loadtoHK(co1)]\ ]\ |CY[\ ]] \tag{10}$$

式 (10) は，大阪港で船から下ろされるコンテナの取扱を監視するための個別式である．$co1[\cdots loadtoHK(co1)]$ の部分がタグから読み取られる式であり，それ以外の部分はあらかじめクレーン PC1 に持たせておき，それらを基に式 (10) がクレーン PC1 で生成される．コンテナの積み下ろし確認をした後，$loadtoHK(co1)$ が式 (9) に基づいて展開され，展開後の式がタグに書き戻される．このようにして，次に載る船をコンテナ ambient に伝えることで，各コンテナが積み降ろされた港で動的に輸送経路が決定される物流をモデル化し監視することができる．

5.2 物流監視システムの規模

以上の機能を組み込んだシステムを Java 言語により実装し，全体でクラス数 127，行数約 16,900 となった．個々の規模は以下のとおりである．

- プロセス式自動生成部：クラス数 14，行数約 1,200
- 輸送監視部：クラス数 51，行数約 6,600
- MAC 処理系：クラス数 55，行数約 8,800
- その他：クラス数 7，行数約 300

動的な経路選択機能を持たない先行研究 [5] と比較して，3,000 行程の増加である．

6. 物流システムのための Ambient Logic

文献 [6], [17] において我々は，AL を用いて物流システムが満たすべき性質を記述し，物流システムを表す AC または MAC の式がその性質を満たすことを確認できるモデル検査システムを提案した．ただし文献 [1] で定義されているすべての様相記号に対応した検証系の構築は容易ではないため，それらの中で物流システムに求められる性質を記述するために必要となる様相記号だけを持つ，限定的な AL を導入した．本稿でも同じ論理を用いる．

6.1 使用する構文

本研究では，時間に関係する性質を満たすことを全体式に対してのみ検査するため，限定的な AL を以下のように非時相論理式と時相論理式に分けて定義している．

定義 6.1（論理式）
（非時相論理式）

η　names

$\mathcal{A}\ \mathcal{B}$　::=

T	*true*
$\neg\mathcal{A}$	*negation*
F	*false* $(=\neg T)$
$\mathcal{A}\vee\mathcal{B}$	*disjunction*
$\mathcal{A}\wedge\mathcal{B}$	*conjunction*
$\mathcal{A}\Rightarrow\mathcal{B}$	*implication*$(=\neg\mathcal{A}\vee\mathcal{B})$
$\mathcal{A}\mid\mathcal{B}$	*composition*
$\eta[\mathcal{A}]$	*location*
$\blacklozenge\mathcal{A}$	*somewhere modality* [*6] □

（時相論理式）

$\mathcal{T}\ \mathcal{U}$　::=

$\mathcal{A}$	
$\neg\mathcal{T}$	
$\mathcal{T}\vee\mathcal{U}$	
$\mathcal{T}\wedge\mathcal{U}$	
$\Diamond\mathcal{T}$	*sometime modality*
$\Box\mathcal{T}$	*everytime modality* □

定義 6.1 において，*composition*，*location*，*somewhere modality* は，文献 [1] で導入されている AL 独自の様相記号の一部である．定義 6.1 のこれら以外の論理記号は通常の時相論理で用いられる記号である．

6.2　全体式に対する充足関係

MAC の全体式 $\overline{P}$ が AL の論理式 $\mathcal{T}$ を満たすことを示す充足関係 $\overline{P}\models\mathcal{T}$ の正確な定義は文献 [6], [17] に譲り，本稿では直観的な説明に留める．

個別式 P と非時相論理式 $\mathcal{A}$ に対し，$P\vDash\blacklozenge\mathcal{A}$ は，P が持つ階層構造のどこかに $\mathcal{A}$ という論理式を満たすプロセスが存在する場合に真となる．

全体式 $\overline{P}$ から到達可能なすべての全体式の集合を $\mathcal{R}(\overline{P})$ とする．$\overline{P}\models\Diamond\mathcal{T}$ は，$\mathcal{T}$ を満たすような全体式 $\overline{P'}$ が，$\mathcal{R}(\overline{P})$ の中に存在するとき真になる．$\overline{P}\models\Box\mathcal{T}$ は，$\mathcal{R}(\overline{P})$ のすべての要素 $\overline{P'}$ が $\mathcal{T}$ を満たす場合に真となる．

6.3　検査する性質と論理式

本稿では，物流システムが持つべき性質として次の 2 点を定め，AL の論理式で表現する．

定義 6.2（目標とする検査項目）

p1：いつか必ず貨物（コンテナ）は目的地に輸送される．

$$\Box\Diamond(\blacklozenge co[T]\Rightarrow\blacklozenge PORT\_B[CY[co[T]\mid T]\mid T]). \tag{11}$$

p2：港以外での貨物（コンテナ）の積み下ろしは行われない（不正な貨物の移動の禁止）．

$$\Box(\blacklozenge co[T]\Rightarrow\neg(co[T]|T)). \tag{12}$$ □

性質 p1 が式 (11) で表現できることは文献 [6] でを示している．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 “コンテナがトップレベル（ほかのすべての ambient の外）に存在することはない” といい換えることができ式 (12) で表すことができる．

文献 [6] においては，定義 6.2 に加え以下の性質も検査していた．

p2’：決められた港以外での貨物（コンテナ）の積み下ろしは行われない（不正な貨物の移動の禁止）．

p3：コンテナ船には決められた貨物（コンテナ）が積み込まれる．

しかし本稿で提案した監視システムの拡張では，使用する船や港が動的に定まることを前提としており，それらすべてを論理式で表現することが困難であったため，定義 6.2 に示す性質に限定した．

7.　モデル検査システム

本章では，本稿で提案するモデル検査システムについて説明する．本検査システムでは，与えられた MAC の全体式に対して初めに状態遷移グラフ（7.2 節で詳述）を作成する．遷移グラフの各ノードは，遷移途中の全体式の構造を表す構文木を持っている．この構文木の capability のリストと木の親子関係を見ることで可能な遷移を見つけて全体式を遷移させ，遷移後の構文木をもつ新たなノードをグラフに追加する．可能な遷移が複数ある場合（Ship1 が先に Port に入る場合と Ship2 が先に Port に入る場合など）に枝分かれが生じる．

このようにして生成した遷移グラフを深さ優先で探査し，各ノードの全体式に対して非時相式の検査を木の親子関係を見ながら行い，*sometime modality* や *everytime modality* のような時間的な検査を経路を見ながら行い，その全体式が定義 6.2 に示した性質 p1 と p2 を満たすかどうかを確認する．

7.1　動的な輸送計画とモデル化

2.1 節で述べたように，中継港でコンテナが載せ変えられる船は船積み依頼リストによって定義される．この船積み依頼リストは中継港にコンテナが到着する 1 日前までに送られるため，コンテナ輸送が始まる前には定まっていない．通常はコンテナが目的地に到着する最適な経路が船会社で設定され，途中で使用される船や港が選ばれる．文献 [6] では，そのような最適な経路をとる物流計画を想定してモデル検査を行う方法を提案した．しかし実際の物流ではこの経路は途中で動的に定まり，しかも必ずしも最適

[*6] ◇ との識別を容易にするため，本稿では *somewhere modality* の記号に ◆ を用いる．

な経路をとるとは限らない．たとえばコンテナ co1 を東京から香港を経由してドバイに輸送する場合，通常は図 3 に示した運行表にある SHIP_C が選ばれ船積み依頼リストが作られる．しかし，SHIP_C が他の荷主のコンテナで満載になった場合は，次週に出航する SHIP_D が選ばれ船積み依頼リストが作成される．より複雑な経路を辿るコンテナ輸送の場合，他の中継港を辿る船が選ばれることもまれではない．

そこで本稿では，どのような経路が選ばれたとしてもそのコンテナが目的地にたどり着くことを示すことのできる検査，あるいは，ある経路を選んだ場合にたどり着けない可能性のあることを示すことのできるモデル検査システムを提案する．

5.1 節で示したように，本稿で提案する物流監視システムではこのような動的に定まる経路をモデル化できるようになっている．具体的には，船積み依頼リストによって指定される経路を definition の形で定義し，コンテナ式の中で definition を呼び出すことで，コンテナ式の中に次に載せ変えられる船の情報が伝えられる．モデル検査システムではこの仕組みを流用し，各中継港に与えられる可能性のある船積み依頼リストに対応する definition をそれぞれ複数生成する．どの definition を用いるかという選択は，どの経路を選ぶのかという選択を表すことになり，この definition の数だけ遷移グラフのノードから枝分かれを発生させる．これにより，考えうるすべての経路を表す状態遷移グラフを得ることができる．

例として，TK 港から OSK 港を中継して HK 港へ 1 つのコンテナを輸送する物流計画を想定して処理の流れを説明する．式 (9)，(10) で示したように，コンテナが中継港で積み下ろされる際に，definition 呼び出し（$loadtoHk(co1)$）が実際の definition 定義に置き換えられる．definition の定義式は**図 4** に示すように港ごとにファイルとして生成され，さらに各港からの複数の経路がそれぞれ番号づけされたファイルで区別して保管される．*loadtoHK* の呼び出しに対し，OS2.amb，OS3.amb または OS4.amb を与えた場合の展開と遷移があるため，それぞれに対応する枝分かれが発生する．なお，図 4 の中の TK2 や OS3 などの数字は時間的な前後関係を表しており，TK2.amb で指定されている船が OSK 港に到着後に利用可能な船は OS3.amb，OS4.amb で指定されている船である．これにより時間的にありえない definition の組合せは排除している．たとえば，東京と大阪を同時刻に出航する船の組合せからなる経路を表すノードは作られないようにしている．

7.2　状態遷移グラフ

遷移グラフの構造を**図 5** に示す．MAC のプロセス式が表す非決定的な性質（たとえばどのコンテナを先に積み下ろしたり積み込むか）による枝分かれは文献 [6] のモデル検査システムでも同じように発生する．それに加え，7.1 節で説明した中継港での輸送経路を考慮した分岐も発生させる．

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（**図 6**）．

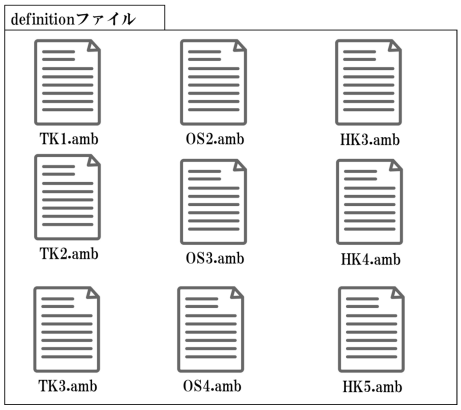

図 4　検査開始時に生成される def ファイル
Fig. 4　Definition files generated at the start of model checking.

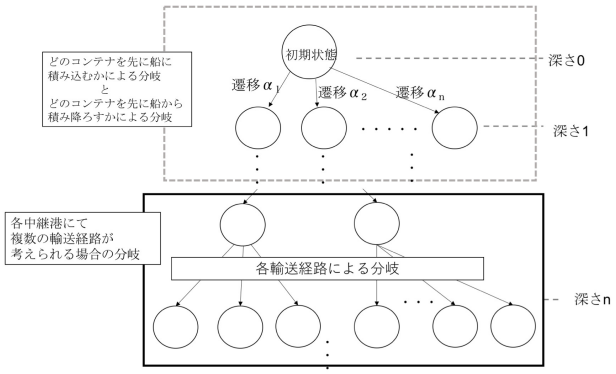

図 5　状態遷移グラフ
Fig. 5　State transiton graph.

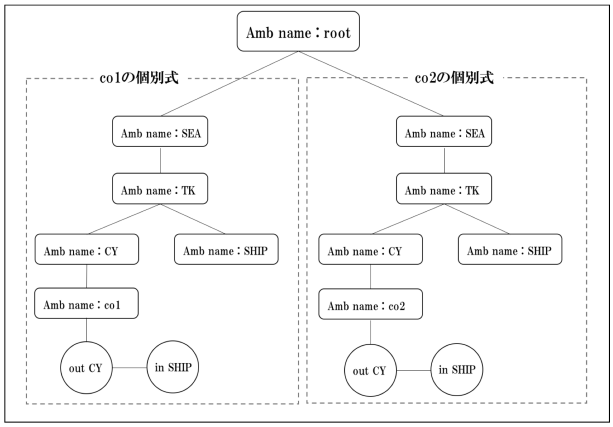

図 6　各ノードが持つ構文木
Fig. 6　Syntax tree in each node of a state transition graph.

図 6 に示した構文木内の個別式内の各ノードは，ambient 名，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 6 は，式 (13) を構文木で表した例である．

$$\begin{aligned} &TK[CY[co1[out\ CY.in\ SHIP]] \\ &\quad |SHIP[out\ TK]], \\ &TK[CY[co2[out\ CY.in\ SHIP]] \\ &\quad |SHIP[out\ TK]] \end{aligned} \tag{13}$$

式 (13) は，コンテナ 1（$co1$ ambient）とコンテナ 2（$co2$ ambient）が東京港（TK ambient）のコンテナヤード（CY ambient）に搬入されている状態を表した全体式の一部である．ここでは，説明のために船の航路を表す個別式や制御用 ambient などの詳細は省略している．

ここで，$co1$ と $co2$ 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，単純に考えると図 6 の構文木を持つノードからは 2 つの子ノードが発生する．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，たとえば 1,000 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，中継港から次の港への輸送経路が複数存在する場合の枝分かれ（図 5 の各輸送経路による分岐）も考慮しなければならないため，ノード数が膨大となり状態空間爆発が発生してしまう．本研究では，前者の枝分かれに対しては文献 [6] で提案された方法を用い，状態空間爆発の抑制を図る．

7.3　弱双模倣等価

我々は文献 [7] において MAC の全体式間の関係である弱双模倣等価関係を定義し，文献 [6] ではこの関係を利用し，大規模な物流計画のモデル検査をより小規模な物流計画のモデル検査に帰着する方法を提案した．本稿でもこの方法を利用する．

定義 7.1（弱双模倣等価 [7]）

$\overline{P}$，$\overline{Q}$ と観測可能イベント（遷移ラベル）の集合 $\mathcal{O}$ に対して，B が以下の性質を満たすとき，関係 B は $\mathcal{O}$ に対して弱双模倣であるという．また，$\overline{P}$ と $\overline{Q}$ は観測可能イベント集合 $\mathcal{O}$ に対して弱双模倣等価であるといい，$\overline{P} \simeq_{\mathcal{O}} \overline{Q}$ と書く．

$(\overline{R}, \overline{S}) \in B$ であるとき以下が成り立つ．ただし $\mathcal{O}^c$ は $\mathcal{O}$ の補集合を表すものとする．

1) $\overline{R} \rightarrow_l \overline{T}$（$l \in \mathcal{O}$）ならば $\overline{S} \stackrel{*}{\rightarrow}_{l'} \overline{U}$（$l' \in (\mathcal{O}^c)^* l (\mathcal{O}^c)^*$）かつ $(\overline{T}, \overline{U}) \in B$ なる $\overline{U}$ が存在する．
2) $\overline{R} \rightarrow_l \overline{T}$（$l \in \mathcal{O}^c$）ならば $\overline{S} \stackrel{*}{\rightarrow}_{l'} \overline{U}$（$l' \in (\mathcal{O}^c)^*$）かつ $(\overline{T}, \overline{U}) \in B$ なる $\overline{U}$ が存在する．
3) $\overline{S} \rightarrow_l \overline{U}$（$l \in \mathcal{O}$）ならば $\overline{R} \stackrel{*}{\rightarrow}_{l'} \overline{T}$（$l' \in (\mathcal{O}^c)^* l (\mathcal{O}^c)^*$）かつ $(\overline{T}, \overline{U}) \in B$ なる $\overline{T}$ が存在する．
4) $\overline{S} \rightarrow_l \overline{U}$（$l \in \mathcal{O}^c$）ならば $\overline{R} \stackrel{*}{\rightarrow}_{l'} \overline{T}$（$l' \in (\mathcal{O}^c)^*$）かつ $(\overline{T}, \overline{U}) \in B$ なる $\overline{T}$ が存在する．

□

$\overline{P} \simeq_{\mathcal{O}} \overline{Q}$ 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，定義 7.1 の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．

7.3.1　不変式を用いたプロセス式のモデル検査

文献 [7] では，文献 [3] で定義されている Invariant を MAC に対して以下のように定義している．

定義 7.2（不変式（Invariant）[7]）

MAC 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 $\overline{P}_{(i)}$ 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 $\mathcal{T}$ に対して $\simeq_{\mathcal{L}(\mathcal{I})}$ が $\mathcal{T}$ の真偽を保存する関係ならば，すなわち $\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} \overline{P}_{(i)}$ ならば $\mathcal{I} \models \mathcal{T}$ と $\overline{P}_{(i)} \models \mathcal{T}$ 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 $\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} \overline{P}_{(i)}$ と $\mathcal{I} \models f$ の検査に帰着できる．一般に，$\simeq_{\mathcal{L}(\mathcal{I})}$ は任意の AL の論理式 $\mathcal{T}$ の真偽を保存するわけではないが，$\mathcal{T}$ から時相記号を削除した式（極大非時相式）が $\blacklozenge co[T] \Rightarrow \mathcal{A}$ の形をしている場合は保存されることを文献 [6] で示した．本稿で使用する論理式（式 (11) と式 (12)）はこの形をしているため，検査対象の全体式 $\overline{P}$ のモデル検査を，その不変式である $\mathcal{I}$ に対するモデル検査に帰着させることができる．以上の考え方に沿って，本稿で提案するモデル検査の手順を以下のように定めた．

手順 1) 検査対象の全体式 $\overline{P}$ に対する不変式 $\mathcal{I}$ を求める．
手順 2) $\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} \overline{P}$ を確認する．
手順 3) $\mathcal{I}$ に対し，7.1 節で示した状態遷移グラフを生成する．
手順 4) 状態遷移グラフを深さ優先探査し，式 (11) と (12) の真偽を確認する．

手順 1) に対しては，対象となる全体式に含まれるコンテナを，同じ仕出港と仕向港を持つコンテナごとにグループにまとめ，各グループのコンテナを 1 つずつ含むような全体式を不変式する．それが本当に不変式かどうかは手順 2) によって機械的に確認できる．

手順 2) に対しては，$\mathcal{I}_2 = (\mathcal{I}, P_1, \ldots, P_l)$ という全体式を作成し，$\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} \mathcal{I}_2$ を確認するすだけでよいことを文献 [6] で示している．ただし l はコンテナのグループ数，各 P_i は，各グループに属するコンテナの個別式で $\mathcal{I}$ 内に存在しないものである．つまり，各グループからもう 1 つずつ個別式

を選び$\mathcal{I}$に付け加えた全体式が$\mathcal{I}_2$である．$\mathcal{I} \simeq_{\mathcal{L}(\mathcal{I})} \mathcal{I}_2$が成り立つことは，このようにして加えた個別式の遷移が$\mathcal{I}$の遷移を妨げないことを意味する．このことを示すことができれば，$\simeq_{\mathcal{L}(\mathcal{I})}$の推移性により手順 2) を示したことになる．

7.4　モデル検査システムの規模

以上の機能を組み込んだモデル検査システムを Java 言語を用いて実装し，全体でクラス数は 114，行数は約 8,800 行となった．個々のプログラムの規模は以下のとおりである．

- 状態遷移グラフ生成プログラム：クラス数 48，行数約 4,300
- 弱双模倣等価性の検証プログラム：クラス数 25，行数約 2,500
- プロセス式に対する様相論理式の充足確認プログラム：クラス数 29，行数約 1,200
- その他：クラス数 12，行数約 800

動的な経路選択機能を持たない先行研究 [6] と比較して，5,000 行程の増加である．

8.　実験

本章では，監視システムの実現性を示すために行った屋内実験，およびモデル検査システムの性能を確認するために行った実験について述べる．前者の実験は，4.1 節で説明した機器を**図 7** に示すように各港ごとに設置し，ダンボール箱に RFID タグを貼りつけてコンテナと見立てて行った．使用した RFID リーダ・ライタは三菱電機社製の RF-RW312，RFID アンテナは三菱電機社製 RF-ATCP012 である．RF タグは金属対応タグであり，容量が 500 バイトである．文献 [5] で行った屋外実験では，同機器を用いて 2.5 m の距離から時速 40 km で走行する自動車に貼り付けたタグ ID の読み取りが問題なくできていることを確認している．また，静止状態のタグに対し 2.5 m の距離でプロセス式の内容の読み取り書き込みが問題がなく行えることを確認している．

8.1　監視システムの屋内実験（目的 1 の確認）

「コンテナ 5 個を船 A で東京港から大阪港へ運んだ後，大阪港で別の船に積み換えて香港港へ運ぶ」という想定で行った屋内実験について示す．これは，本監視システムが動的な経路設定に対応できることを確認するための実験である．

まずコンテナ 5 個が船 A に積み込まれる．その際，タグから取得したコンテナ ID に対応した個別式が期待どおりに遷移したことから，その積込みに誤りがないことを確認し，遷移後のコンテナ式がタグに書き込まれた．コンテナの積込みが完了した後，船 A を東京港から出航させた．

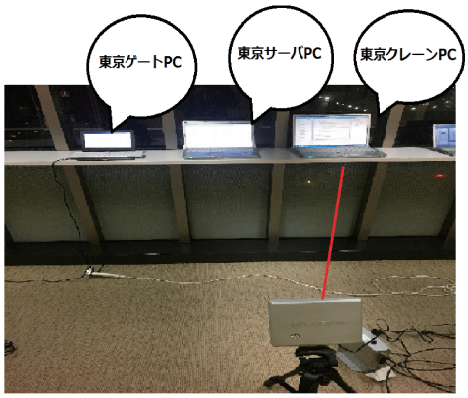

図 7　屋内実験の様子
Fig. 7　Indoor experiment.

表 1　実験に使用した計算機の仕様
Table 1　Spec of the PC for the experiments.

OS	Linux 4.4.92 (64 bit)
CPU	Intel Core i7-7700@3.60 GHz
メモリ	62.8 GByte

船 A が東京港から大阪港へ移動している最中に，船積み依頼リストのエクセルファイルをトップサーバから大阪港のサーバ PC に送り，大阪港のクレーン PC にそれらのコンテナを船 C に積み込むという definition が送られたことを確認した．

船 A を大阪港に入航させ，東京港で積み込んだコンテナを積み降ろした．その際，次に船 C に載ることが definition の形でクレーン PC からコンテナに伝えられたことを確認した．その後船 C を大阪港に入航させ，コンテナを船 C に積み込んだ．その際，コンテナは経路情報更新後のプロセス式を持っていたため，船 C への積込みに誤りがないと判定されたことを確認した．以上により，載せる船を動的に指定した場合においてもコンテナの移動を正しく監視できることを確認した．また，コンテナが大阪港に到着する前に船 C を出航させようとし，積み込み忘れがあるとの警告が出ることを確認した．積み込み・積み下ろし確認においては，リーダ/ライタによる式の読み込み，遷移確認，式の書き込みが，1 つのコンテナに対し 15〜20 秒で終了することを確認した．

8.2　モデル検査実験（目的 2 の確認）

物流システムを表現したいくつかの全体式を対象に，所期の性質を表現した AL の論理式が満たされるかどうかの検査実験を行った．実験に使用した計算機の仕様を**表 1** に示す．

検査実験で確認する物流システムの所期の性質として，定義 6.2 に示した p1 (式 (11)) と p2 (式 (12)) を使用した．

8.2.1　一般的な輸送計画に対する検査（実験 1）

海上物流では一般に，1 つの船に目的地の異なる複数の

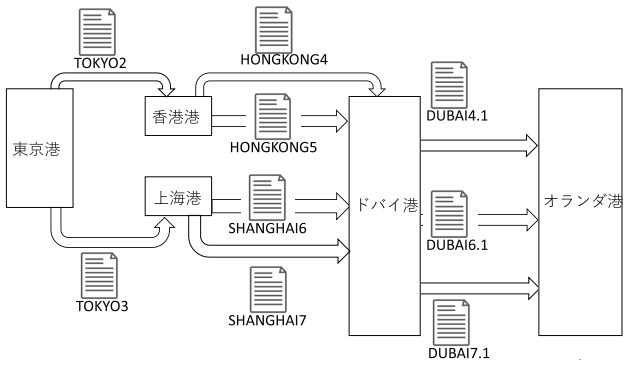

図 8　実験 1 の輸送経路
Fig. 8　Shipping routes for experiment 1.

コンテナが混載される．そのような状況を想定し，4 つの仕向港にそれぞれ 2,000 個のコンテナを輸送する物流計画を表現した全体式に対する検査実験を行った．

この輸送計画において，各港でコンテナを積み込むことが可能な船，およびそこから列挙できるすべての輸送経路は**図 8** のようなものとした．図 8 中の二重矢印（⟹）上にある TOKYO2 などと書かれた書類図が図 4 に示した def ファイルであり，それらのファイルに記載されている definition により次の港へ向かう船が指定される．東京港では，香港港と上海港に向かう船が選択でき（`TOKYO2, 3`），香港港と上海港ではドバイ港へ向かう船がそれぞれ 2 隻ずつ選択できる（`HONGKONG4, 5, SHANGHAI6, 7`）．

さらにドバイ港ではオランダ港へ向かう船が 3 隻選択できる（`DUBAI4.1, 6.1, 7.1`）．書類名の末尾の数字は時間に関する制約を表しており，コンテナ積み替え時には，積み下ろす船の数字より大きい数字を持つ船のみに積み込み可能であることを意味する．たとえばドバイ港では，SHANGHAI6 で運ばれてきたコンテナは DUBAI6.1 または DUBAI7.1 にのみ積み替え可能であり，DUBAI4.1 は出航した後なのでそれらのコンテナが DUBAI4.1 に載せ替えられることはない．

このような輸送計画に対し，すべてのコンテナが物流システムの所期の性質を満たしているかを検査した．検査では，7.1 節に示したように船積み依頼リストからこれらの definition を生成し，7.2 節に示したように definition 呼び出しの際にどの definition を選択するかによって生じる組合せを反映した状態遷移グラフが生成される（7.3.1 項で述べた手順 3)）．このグラフを探査することで，輸送中に動的に決定される可能性のあるあらゆる経路について，所期の性質の検査が行われる（7.3.1 項で述べた手順 4)）．

その結果，すべてのコンテナがすべての輸送経路に対し，それぞれの仕向港へ到着するといった結果が得られた．検査に要した時間は約 327 秒であった．内訳は，不変式の生成（7.3.1 項で述べた手順 1)）に 20 秒，不変式の確認（手順 2)）に 123 秒，グラフの生成に 167 秒，p1 の検査に 17 秒，p2 の検査に 38 秒である．

8.2.2　問題のある輸送計画に対する検査（実験 2）

この実験では，コンテナ輸送を表した全体式が，何らかの不備で間違えた輸送計画となっていた場合，その輸送計画が誤ったものであることを検出できるかを検査した．

輸送経路および各コンテナ数は実験 1 と同じである．ただし，図 8 の DUBAI7.1 を DUBAI6.2 に変更し，オランダ港向けのコンテナが SHANGHAI7 でドバイ港に到着した場合，このコンテナがオランダ港に到着できないようにしてある．

この検査実験の結果，「すべてのコンテナがすべての輸送経路に対し，仕向港へ到達できる」が偽であるという結果が得られた．検査に要した時間は約 324 秒であった．内訳は実験 1 とほぼ同じで，p1 の検査のみ 14 秒と僅かに短くなっている．またその際に得られる全体式情報から，目的港へ到達できなかった原因も特定できる．

8.3　考察

8.1 節に示した監視実験は，目的 1 に示したように，輸送途中に決定される経路を中継港で動的にプロセス式に反映させ，RFID タグを用いてコンテナの取扱を監視する機能を確認するものである．

使用できる機器数の制限により 8.2 節に示したモデル検査実験ほど複雑・大規模な物流システムを対象としたものにはなっていないが，目的 1 の基本的な各機能が問題なく動作していることを示すことができた．

本稿で示した実験は単純な構成の物流システムを想定し屋内で行ったものであるが，使用している機器が屋外でも有効であることは文献 [5] で示している．これにより，我々の提案する方法で実際の海上物流システムで生じる様々な動的変更に追随できる監視が可能になると考えられる．また検査時間は 10 数秒であることを確認したが，クレーンによる実際のコンテナ移動には数分掛かることから，実用上問題ないといえる．また，1 つ 1 つの移動の確認に中央のサーバとの通信を必要としない本手法は，通信障害による遅延が発生せず必ずコンテナ移動中に検査が終了するため，実用的な方法だといえる．なお，この実験では設備の都合から 5 個のコンテナ移動の確認が限界であったが，1 時間に数十個の移動が行われる実際の物流システムを想定した実証実験，より複雑な航路を想定した実証実験も今後必要だと考えている．

8.2 節に示したモデル検査実験は，実際に行われている海上物流と同規模（コンテナ数千個）の輸送計画を想定して行った．この実験により，輸送途中で動的に決定される経路すべてに対し，物流システムに求められる所期の性質を満たすかどうかの検査が，実用的な時間で可能であることを示すことができた．これにより，実際の物流システムにおいて，人為的に経路が変更された場合でもその妥当性を短時間で示すことができると考えられる．

日本周辺におけるコンテナ輸送の基幹航路は，北米・アジア航路と欧州・アジア航路であり [22]，文献 [22] では後者の実例として，神戸 → シンガポール → ロッテルダム → ハンブルクという航路が示されている．その例では中継港として，シンガポール（アジア拠点）とロッテルダム（欧州拠点）の 2 港が使用されている．それら以外にも多数の航路が文献 [22] に「代表的な国際複合一貫輸送ルート」として示されているが，ハブ&スポークシステムでは多くの場合航路はこのように 2 つの地域のハブ港と，それらからつながる両端のフィーダ港（あるいはそれよりも小規模）からなっている．図 8 に示す経路例では，アジア地域での中継港の選択（香港 or 上海），また，中継港から次の中継港への複数の船の選択といったように様々な選択を含んでいるため，実際のコンテナ輸送で起こりうる多くの航路選択をカバーしている．もちこんこれで十分というわけではなく，図 8 に示す経路例よりも複雑な航路も存在するため，たとえば東京港の前に日本のローカル港，オランダ港の後に欧州のローカル港などを付け加えた航路を設定した実験も行っており，同様の結果を確認している．

9. 関連研究

本章では，AC を用いたモデル検査についていくつかの関連研究を取り上げ，本稿の手法と比較する．

文献 [15] では，AC と AL を用いたセキュリティポリシに関するモデル検査手法が提案されている．これは，ネットワーク設計においてネットワーク自体を AC でモデル化し，ユーザがそのネットワーク内のどのドメイン，ホストに移動しても，ユーザに許されたアクセス権限を逸脱しないようセキュリティポリシが確保されているかどうかを確認するものである．ここで示されている手法はオーソドックスなものであり，AC のプロセス式の状態遷移をクリプキ構造のグラフで表し，そのグラフをたどることで AL 式の表す時相的性質および空間的性質が満たされているかどうかを確認するものである．実装実験では 16 個の ambient を含むプロセス式に対するモデル検査が 6 分で終了すると示されているが，状態空間爆発への対処は行っていないため，数百，数千個の ambient で対象物をモデル化するモデル検査は不可能であり，partial order reduction [3] も今後の課題とされている．我々はすでに同様の手法を用いたモデル検査手法を物流システムに対して提案しているが [8]，partial order reduction や物流システムの持つ対称性を利用した効率化を行っても，150 個のコンテナ ambient を含む数百規模のプロセス式に対するモデル検査が実用上の限界（4 時間半）であった．そのため本稿では，すべての対象物の動作を網羅的に検査する直接的な方法ではなく，弱双模倣性を利用したモデル検査手法を提案し，その効果を確認している．

文献 [4] では，リプリケーションにより実行中に無限に規模が増大するような AC のプロセスに対し，そのプロセス式を HABA [11] と呼ばれるオートマトンに変換し，検査したい性質のみを含む有限状態のオートマトンに対してモデル検査を行う方法を提案している．特にサーバ・クライアントモデルに着目し，クライアントの無限のリクエストにサーバがリプリケーションで答えるようなシステムを想定し，オーバフローが起こらないこと，およびリクエストには必ず答えが返ってくるなどの性質が有限時間で検査できることが示されている．ただし数千個におよぶ対象物の非決定的な動作すべてに対し実用的な時間で検査が終了するか否かは示されていない．本稿で提案しているモデル検査手法も有限の状態のみを持つ MAC のプロセス式のみを対象としたものであるが，我々の手法では MAC のプロセス式を他のモデルに置き換えることなく検査可能であり，数千個のコンテナ移動に関する検査を弱双模倣性を利用して実用的な時間で終了することを示している．

10. 結論・今後の課題

本稿では，コンテナ海上物流を多重 Ambient Calculus を用いてモデル化し RFID 機器により自動的に監視するシステムにおいて，輸送経路が動的に変化する場合でも監視を継続できる機能を提案した．この機能により，多数のコンテナが乗換えを含む複雑な経路を辿るような輸送に対しても柔軟なモデル化が可能となり，より現実に即した監視システムとすることができた．さらに，モデル化された物流システムに対し，所期の性質を満たすかどうかを確認することのできるモデル検査システムを提案した．

本研究ではその単純さ故に FCL のみを対象としているが，LCL もコンテナ単位では輸送にかかわる船積み依頼リストは FCL と同じであるため，本稿で示す手法で対応できると考えている．ただし LCL では 1 つのコンテナに複数の荷主の荷物が混載されるため物流書類が複雑になり，プロセス式自動生成システムの拡張が必要になるため，今後実現性の検証を行う必要があると考えている．

ところで，RFID を用いた物流監視については，1 章で述べたように様々な機関が 10 年以上にわたり研究を続けているが，いまだに商用としては普及していない．主な理由は 2 つあり，1 つは RFID タグを付けられたコンテナとそうでないコンテナが混在した場合監視活動が煩雑になってしまうことである．もう 1 つは，輸送の途中でタグの情報が損なわれたときに回復が困難であるという，タグの信頼性に関する問題である．

後者の理由に対し我々の方法では，途中のタグの読み書きで使用した PC にタグのバックアップをとっておけるため，ある港で読み込みエラーが起きたとしても，その前の港で積み込んだ際のデータをこの港の PC に送りタグに書き込み直すことでエラーの回復が可能である．他の機関の仕組みでは，監視途中のタグの書き込みは想定していない

ため，このような回復は不可能である．

前者の理由に対しては完全な解決策は今のところ存在しないが，我々は現在日栄インテック社の協力を得て読み書きの精度が高いタグ [24] を用いた実験を行っている．さらに，リーダ/ライタもクレーンに取り付ける固定式のものから，より利便性の高いハンディタイプの機器に移行する予定である．これらの機器を利用することにより，煩雑さを最小限に抑えた確認作業ができると期待している．

今後これらの機器を使えるようミドルウェアを開発し，本システムの実用性を高める必要がある．

謝辞　本研究を進めるにあたって，コンテナの取扱いの確認作業や動的な船の決定など実際の物流に関する様々な事項をご教示いただきました商船港運，後藤回漕店，日本郵船，商船三井各社に感謝します．また，RFID タグに関しご協力いただきました日栄インテック社に感謝します．本研究は科研費（15K00040，25330095）の助成を受けたものである．

参考文献

[1] Cardelli, L. and Gordon, A.D.: Any time Anywhere Modal logics for Mobile Ambients, *POPL'00*, *Proc. 2000 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp.365–377 (2000).
[2] Cardelli, L. and Gordon, A.D.: Mobile Ambients, *Theoretical Computer Science*, Vol.240, pp.177–213 (2000).
[3] Clarke, E., Grumberg, O. and Peled, D.: *Model Checking*, MIT Press (2000).
[4] Dino, D.: A Parametric Model for the Analysis of Mobile Ambients, *LNCS*, Vol.3780, pp.401–417 (2005).
[5] 橋本隆弘，加藤　暢，樋口昌宏：多重 Ambient Calculus と UHF 帯 RFID 機器を用いた海上物流監視システム，情報処理学会論文誌：プログラミング，Vol.6, No.2, pp.1–12 (2011).
[6] 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：プログラミング，Vol.5, No.3, pp.50–60 (2012).
[7] 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，Vol.5, No.2, pp.79–87 (2012).
[8] 加藤　暢，樋口昌宏，植田直人：物流システムに対する Ambient Logic モデル検査システム，情報処理学会論文誌：数理モデル化と応用，Vol.3, No.1, pp.73–86 (2010).
[9] Kato, T. and Higuchi, M.: A Handling Management System for Freight with the Ambient Calculus, *International Journal of Mathematical Models and Methods in Applied Sciences*, Vol.3, pp.179–188 (2009).
[10] The Apache POI Project: Apache POI - the Java API for Microsoft Documents, poi.apache.org (2002).
[11] Montanari, U. and Pistore, M.: *History-Dependent Automata: An Introduction*, pp.1–28, Springer Berlin Heidelberg (online), DOI: 10.1007/11419822_1 (2005).
[12] 森本大輔，加藤　暢，樋口昌宏：Ambient Calculus を用いた物流検査システム，情報処理学会論文誌，Vol.48, No.SIG 10 (PRO33), pp.151–164 (2007).
[13] MTI 株式会社（日本郵船グループ），野村総合研究所：平成 18 年度電子タグ実証実験データキャリアの国際標準化事業 (2007)，入手先 ⟨www.meti.go.jp/report/downloadfiles/g71120a01j.pdf⟩.
[14] 岡田翔太，馬谷誠二，林　奉公，八杉昌宏，湯淺太一：Safe Ambients のための Java フレームワーク，情報処理学会論文誌：プログラミング，Vol.4, No.3, pp.26–41 (2011).
[15] Unal, D., Akar, O. and Caglayan, M.U.: Model Checking of Location and Mobility Related Security Policy Specifications in Ambient Calculus, *LNCS*, Vol.6258, pp.155–168 (2010).
[16] 吉岡信和，田原康之，本位田真一：モバイルエージェントによる柔軟なコンテンツ流通を実現するアクティブコンテンツ，情報処理学会論文誌：データベース（TOD），Vol.44, No.SIG18, pp.45–57 (2003).
[17] 加藤　暢，樋口昌宏，植田直人：物流システムに対する Ambient Logic モデル検査システム，情報処理学会論文誌：数理モデル化と応用，Vol.3, No.1, pp.73–86 (2010).
[18] 経済産業省：物流業界における電子タグ等の活用実証実験国際コンテナ輸送 (2005)，入手先 ⟨www.mlit.go.jp/kisha/kisha05/10/100329_2/02.pdf⟩.
[19] 国土交通省：メコン地域陸路実用化実証走行試験 (2007)，入手先 ⟨www.meti.go.jp/press/20071018006/press_mMM.pdf⟩.
[20] 国土交通省：サプライチェーンにおける海上貨物追跡タグシステム（MATTS）の実証実験の実施について (2008)，入手先 ⟨www.mlit.go.jp/report/press/port02_hh_000006.html⟩.
[21] 国土交通省：コンテナ物流情報サービスシステム「Colins」の中国との連携について (2011)，入手先 ⟨www.mlit.go.jp/report/press/port02_hh_000053.html⟩.
[22] 今井昭夫：国際海上コンテナ輸送概論，東海大学出版会 (2009).
[23] 東芝ロジスティクス：東芝ロジが異業種向けに次世代グローバル混載サービス拡大，カーゴニュース，10 月 13 日，p.1 (2015).
[24] 日栄インテック：RF タグ『金属対応 UHF パッシブタグ』(2015)，入手先 ⟨http://www.ipros.jp/product/detail/2000229559/⟩.
[25] 野村総合研究所：戦略的国際標準化推進事業（国際標準共同研究開発事業：海上コンテナトラッキングシステムに関する標準化に係るフィージビリティスタディ）成果報告書 (2012)，入手先 ⟨www.meti.go.jp/meti_lib/report/2012fy/E002582.pdf⟩.

加藤 暢　（正会員）

1997 年岡山大学大学院自然科学研究科博士課程修了．博士（工学）．1998 年日本学術振興会特別研究員（PD）．2000 年より近畿大学理工学部講師．現在，准教授．並行論理型言語の意味論，プロセス代数に関する研究に従事．

高岡 久裕　（正会員）

2018 年近畿大学大学院博士前期課程修了．修士（工学）．現在（株）日立システムズ勤務．在学中，プロセス代数に関する研究に従事．

樋口 昌宏　（正会員）

1983 年大阪大学基礎工学部情報工学科卒業．1985 年同大学院博士前期課程修了．（株）富士通研究所勤務，大阪大学基礎工学部助手・講師等を経て，現在，近畿大学理工学部情報学科教授．博士（工学）．分散システムの記述，検証，試験に関する研究に従事．

大山 博史

1995 年広島大学大学院理学研究科博士課程後期物理学専攻修了，博士（理学）．1995 年より国立広島商船高等専門学校．現在，国立高等専門学校機構広島商船高等専門学校教授．船舶による輸送，放射線計測に関する研究に従事．