

Ambient 計算に基づく動的な海上物流の監視システム

辻村 俊伸^{†1} 樋口 昌宏^{†1} 加藤 暢^{†1}

海上物流の世界では、貨物の流通量の増加に伴い取り扱いの誤りの増加が問題となっている。この問題に対し我々は、Ambient Calculus による物流システムの記述と、実際の物流がその記述どおりに行われているかを監視するシステムを開発している。物流システムは、荷物、コンテナ、コンテナ船というように、複数のより小さなパッケージを収容したパッケージがより大きなパッケージに収容されるという階層構造を持っている。一方 Ambient Calculus は、階層構造を持ち動的に構造が変化するシステムを形式的に記述するための言語であり、この特徴から物流システムの持つ階層構造を簡潔に表現することができる。本稿では、実際のコンテナ輸送で使われる数種類の書類を元に、各コンテナの搬送経路を表現する Ambient Calculus 式を自動生成するシステム、及び RFID を用いて検知した貨物の動きが Ambient Calculus 式の遷移に沿ったものであるかを監視するシステムについて述べる。近年現実の海上物流において、コンテナがいくつかの中継港を経由して輸送されるハブ & スポークシステムが普及している。本システムはこのようなハブ & スポークシステムに対して、Ambient Calculus 式の生成やコンテナ監視ができるよう構築されている。

A Management System for Freight Systems with Dynamic Routing based on the Ambient Calculus

TOSHINOBU TSUJIMURA,^{†1} MASAHIRO HIGUCHI^{†1}
and TORU KATO^{†1}

We are investigating freight management systems ensuring the correctness of container handling during shipping. Such systems determine the correctness by comparing container handling, which is sensed by RFID, with transitions of formal models (formulae) written in the Ambient Calculus. Freight systems have the nested structures, i.e., packages (e.g. containers) containing smaller packages (e.g. luggage) that are accommodated by a larger entity such as a container ship. The Ambient Calculus is a formal description language that is suitable for modeling systems with the nested structures that dynamically change. The management system consists of two components: a system that automatically generates formulae of the Ambient Calculus and another system

that manages the actual freight. The former system generates formulae that express the transportation routes of containers based on several trading documents such as shipping invoices, B/L instructions and routing plan tables of ships. The latter system senses the movement of containers by using RFID and checks their consistency with the transitions of corresponding formulae. The management system can treat hub and spoke type freight systems where containers leave a spoke port for another spoke port via more than one hub port that are recently becoming common.

1. はじめに

近年、物流の世界では、貨物の流通量の増加に伴いコンテナ管理の重要性が高まっており、コンテナをどのように管理していくかについてさまざまな方法が模索されている^{5),6)}。我々は現在、物流書類を用いて物流システム全体をモデル化し、物の動きを自動的に監視するシステムに関する研究を行っている。物流書類そのままでは形式的な取り扱いが困難であるため、本研究では物流書類の記載内容を Ambient Calculus²⁾ のプロセス式で表現し、実際の物流がそのプロセス式の記述通りに行われているかを監視する方法を考案した。すでに送り状⁸⁾ という実際に貿易で使われる書類をプロセス式に変換し、バーコードと Linux ボードを用いて検知した物の移動と、プロセス式の遷移とを関連付けることで記述通りの貨物輸送が行われていることを半自動で監視するシステムを構築している⁷⁾。

しかし文献 7) のシステムは、貨物の輸送経路や船の航路が予め定まっており、中継港での積み替えが無い単純な物流システムを対象としている。またバーコードを用いて監視を行うため、監視方法が半自動的である。そこで本論文では、中継港での貨物の積み替えが発生するような物流システムにも対応でき、またコンテナの取り扱いを自動的に監視できるシステムを提案する。前者に対しては、中継港でコンテナが乗り換える船を指定する輸送経路指定表や船の航路表に記載された情報も利用して、プロセス式を動的に生成することで対応する。後者に関しては、バーコードを用いて物の移動を検知する半自動的な方法は、実際の物流ではコンテナ船 1 隻で数千のコンテナが取り扱われることから現実的でないため、物の移動の検知を全自動的に行えるよう RFID を用いたシステムとする。

^{†1} 近畿大学理工学部情報学科

Kinki University School of Science and Engineering Department of Informatics

2. FCL 海上物流システム

本論文で対象とする物流システムは文献 7) と同じ一つのコンテナを一人の荷主が利用する FCL(Full Container Load) 海上輸送とする。

2.1 FCL 海上物流とハブ・アンド・スポーク・システム

海上物流の分野では近年ハブ・アンド・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の下で、各海運会社の拠点港(あるいは中継港)をハブ港、ハブ港の周りのローカルな港をスポーク港とし、ハブ港を中心にした輸送を行うことにより、効率的な輸送を行う輸送システムである。このシステムの普及により、コンテナを船から中継港でコンテナヤードに降ろし、別の船へ積み替える輸送が増加している³⁾。ハブ・アンド・スポーク・システムにおいて、船の運航計画は天候の影響などにより刻々と変化することにより乗り換えの船は中継港に貨物が届いてから決定されるのが普通である。

そこで、本論文では i) コンテナは一般に複数の中継港を経由して荷出し港から最終仕向港へ輸送する、ii) コンテナの輸送経路はあらかじめ決定されているのでは無く、中継港に到着後乗り換えの船を決定する、ような海上物流を適切にモデル化し、その検査システムの構築について議論することにする。

2.2 物流定義書類とそれに基づく物流の実行

通常、物流は書類により定義され、その書類に基づいて貨物の取扱いあるいは船の運航といった作業が実行される。以下物流に用いられる書類と、それに基づく作業の実施について説明する。

荷主(輸出者)はコンテナの荷出し港と最終仕向港を指定した送り状を作成し、これに基づいて輸送業者がコンテナ輸送を実行する。コンテナの船への荷積み、港への荷降ろし作業は輸送業者が各港で作成する B/L Instructions に従って行われる。B/L Instructions には、輸送に使用されるコンテナの本数と各コンテナのコンテナ番号の他、荷積み港、荷降ろし港、コンテナが積み込まれる船(船名と航海番号)といった情報が記載されている。また、輸送業者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航路として記載されている。

また、船が出港する際、その港における全ての荷降ろし、荷積み作業が完了している必要がある。さらに、荷降ろし、荷積み作業は対象となる船が港に到着してからしか実行してはいけない。このように海上物流の実行では船の運航と貨物取扱い作業の間の同期が必要で

ある。

以上の書類のうち送り状と航路表は、それぞれ荷主、輸送業者によって任意に作成されるものである。一方、B/L Instruction は輸送業者が送り状の記載に従って、コンテナが最終仕向港に届くように各港で作成される。B/L Instruction の作成時には、そのコンテナを次にどの船にのせてどこまで運ぶか選択が輸送業者によって行われる。そのような選択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。そこで本論文では上記の書類に加えて、各港において、貨物の最終仕向港ごとにもっとも適切な船と荷降ろし港を指定した輸送経路指定表が作成されているものとする。また、輸送経路指定表は、適切な船と荷降ろし港の選択は船便の運行状況によって時々刻々と変動するものであることより随時変更可能なものとする。

3. 物流システムの Ambient Calculus による表現

Ambient Calculus²⁾ 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記述することができる言語である。この特徴から物流システムの持つ階層構造を簡潔に表現することが出来る。さらに、様相論理の一種である Ambient Logic¹⁾ の公理系を用いて、プロセス式そのものが意図した性質を持っているかどうかを検証することが可能である⁴⁾。ここでは、物流システムにおいて 1) その時点での物流に関わる実体間の階層構造、2) 今後の物流計画(その時点で発行されている B/L Instruction, 航路表, 輸送経路指定表の記載内容、3) 船の運航と貨物取扱いの間で必要な同期、を ambient 式で表現することを考える。そのような式は物流システムの現在の状態と、その後許されている動作の系列を表現したものと考えることができる。

3.1 階層構造とその変化の表現

物流システムは港の中にコンテナヤードがあり、コンテナヤードの中にコンテナがある、といった階層構造を持っている。本研究では港、船、コンテナヤード、コンテナといった実体をそれぞれ ambient として記述することにより、物流システムの持つ階層構造を表現する。例えば東京港の中に船とコンテナヤードがあり、コンテナヤードの中にコンテナがあるといった階層構造を ambient 式で記述すると、式 (1) のようになる。

$$\text{東京} [\text{船} [] \mid \text{コンテナヤード} [\text{コンテナ} []]] \quad (1)$$

東京港を表す ambient である $\text{東京} []$ の括弧の中に、船を表す ambient である 船 とコンテナヤードを表す ambient である $\text{コンテナヤード} []$ を記述し、コンテナヤードを表す

ambient である コンテナヤード [] の括弧の中にコンテナを表す ambient である コンテナ [] を記述する．このように ambient の中に ambient を記述することにより，物流に関わる実空間の階層構造を直截的に表現することが出来る．

また物流システムが持つ階層構造は，コンテナや船の移動が発生するたびに变化する．コンテナや船の移動による階層構造の変化は，遷移規則²⁾を用いることにより表現出来る．船が移動する例として，東京から神戸へ移動する船を ambient 式で記述する．式 (2) がこのような運航の初期状態を表している．

$$\text{東京 [船 [out 東京.in 神戸]] | 神戸 []} \quad (2)$$

$$\xrightarrow{\text{out 東京}} \text{船 [in 神戸] | 東京 [] | 神戸 []} \quad (3)$$

$$\xrightarrow{\text{in 神戸}} \text{東京 [] | 神戸 [船 []]} \quad (4)$$

船の移動は *in* や *out capability*²⁾ を用いてを表現する．例えば式 (3) では，船を表す ambient の中に東京から出港する動作を表現する *out 東京* という capability を記述することにより，船が港から出港する動作を表現している．式 (4) では，船を表す ambient に神戸へ入港する動作を表現する *in 神戸* という capability を記述することにより，船が港へ入港する動作を表現している．

3.2 同期機構の記述

物流システムをモデル化するためには，2.2 節で述べたような船とコンテナの移動の同期を表現する必要がある．このような同期を表現するためにも，本研究では ambient を利用する．本稿ではこのような ambient を制御用 ambient と記述する．

制御用 ambient の例としてコンテナの荷降ろしを例に説明する．荷降ろしに関する制御用 ambient である *unload* は，式 (5) のように港 ambinet の中に記述されており，*in capability* を実行することにより船 ambinet 内に移動する (式 (6))．これにより船に積み込まれたコンテナ ambient は荷降ろしに関する遷移が可能になる (式 (7))．

$$\text{港 [unload[in 船] | 船 [コンテナ [check_out(unload) 船]]]} \quad (5)$$

$$\xrightarrow{\text{in 船}} \text{港 [船 [unload[] | コンテナ [check_out(unload) 船]]]} \quad (6)$$

$$\xrightarrow{\text{check_out(unload) 船}} \text{港 [コンテナ [] | 船 [unload[]]]} \quad (7)$$

ただし，“*check_out(unload) 船*” は，*in unload.out unload.out 船* の略記である．

3.3 動的な輸送経路指定表の表現

本論文で対象としているハブアンドスポークシステムでは，コンテナは中継港に到着する

度に，送り状に記載された最終仕向港と輸送経路決定表に基づいて新たな B/L Instructions が作成される．ここでは，そのような仕組みを Ambient Calculus の枠組で記述する方法を述べる．

コンテナの中継港での乗り換えを表現するために，本研究では *router* と名付けた制御用 ambient を導入した．*router ambient* はその内部に輸送経路指定表の内容にしたがってコンテナの経路を指定する制御用 ambient *toPORTNAME* を持つ．また *toPORTNAME ambient* の内容は，輸送経路指定表を更新することにより，更新後の輸送経路指定表に沿った ambint 式に変更することが出来る．*router ambient* が持つコンテナの経路指定に関する制御用 ambient のうち，主要なものを以下に示す．

router

コンテナの経路指定を行うための *toPORTNAME* を内部に持っている制御用 ambient である．

toPORTNAME

PORTNAME に行くためには，次にどの船に乗ればいいのかを ambient 式の形式で持っている制御用 ambient である．

query

コンテナ ambient が *router ambient* に問い合わせるための制御用 ambient であり，送り状の記載内容に対応するものである．*query ambient* が *toPORTNAME ambient* 内に移動することにより，*toPORTNAME ambient* 内の *ans ambient* が移動可能になる．

ans

コンテナ ambient からの問い合わせに対し，次にどの船に乗ればいいのかを回答するための制御用 ambient である．*toPORTNAME ambinet* からコンテナ ambient へ移動し，コンテナ ambient 内で *ans ambient* が *open* されることにより，コンテナ ambient は次の輸送のための B/L Instructions に対応する情報を取得する．また *ans* の中には経路情報に加えて，コンテナ ambient が次の港で，*router ambient* に次の経路を問い合わせるための *query* も含まれている．

lcomp

router ambient がコンテナ ambient からの問い合わせを受けた際に，*ans* と共に生成されるコンテナの荷積みに関する制御用 ambient である．*lcomp* が *router ambient* から船 ambient に移動し船 ambient にコンテナ ambient が移動すると，制御用 ambient *lcomp* は *open* される．*lcomp* が *open* されることにより，コンテナの荷積みが行われたことを確認し，船 ambient は移動 (出航) 可能となる．

する事を確かめる

- 正しくない移動を行った場合に知らせる
 - 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以下の実装を行った。
- 物流書類からプロセス式を自動生成するシステム
 - 物を表現する ambient ごとの ambient 式の分散処理
 - 物の動きとプロセス式遷移との照合
 - 上記関連付けに伴う動作エラーの警告

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 ambient 式の遷移に沿ったものであるかを調べるものである。その為に本システムは、RFID を用いてコンテナの移動を検知することとする。

4.1 機器の配置と分散処理

コンテナには IC タグを貼りつけ、船には PC を設置する。港とコンテナヤードには IC タグに対する読み書きを行うためのリーダ/ライタを接続した PC を設置する。各 PC や IC タグは、親 ambient へのポインタ、自身を表す ambient とその capability、子 ambient の情報 (制御用 ambient の場合は式そのもの、モノをあらわす ambient の場合はその名前とポインタ) を部分式という形で保持している。これらの部分式を組み合わせることで、物流システム全体を表す式が構成される。各 PC は親、子 ambient へのポインタを介して、必要に応じてお互いの保持している式を参照することができる。また、各 PC には ambient 式を解釈し、可能な遷移を実行する処理系が搭載されている。

4.2 プロセス式の生成

本システムでは、港、コンテナヤードといった動的な変動を行わない ambient に関する式はあらかじめ生成しておき、各 PC に収容しておく。そして、新たに送り状や航路表が発行されたり、輸送経路決定方針が変更された際に、それらを表すプロセス式を生成し、それまでのプロセス式に付加することにより物流システムの現在の状態を式として保持する。

送り状からコンテナの識別子、荷出し港と最終仕向港の情報を取得する。また航路表から船の運航経路を取得する。これらの情報は Excel 形式で記述された書類から POI を用いて取得している。POI とは jakarta プロジェクトの一つで Java で Excel を操作するライブラリである。物流書類から取得した情報を用いて、コンテナ ambient、船 ambient を生成する。輸送経路指定表が変更された場合には、オペレータとの対話処理に基づいてプロセス式を生成する。

新たな送り状が発行された場合は、コンテナ ambient が生成され、そのコンテナが荷出し港のコンテナヤードに到着した際に、荷出し港のコンテナヤード ambient の子として既存の ambient 式に追加される。新たな航路表が送り状が発生した場合についても、その船が発発港に到着した際に、生成された ambient がその港の ambient の子として既存の ambient 式に追加される。

また輸送経路指定表の内容を更新する場合は、新規の輸送経路指定表の内容を記述した制御用 ambient と、既存の輸送経路指定表を無効化する制御用 ambient が生成され、その港のコンテナヤード内の router ambient の子として追加される。その制御用 ambient が動作することにより輸送経路指定表の内容が書き換えられる。

このように物流システムに新たな輸送計画が発生した場合は、その輸送計画の内容を表現した ambient 式を既存の式に追加する形式で、新たな輸送計画を含めた物流システムを表現している。

4.3 コンテナ移動監視方法

コンテナヤードから船への積み込みなどコンテナの移動が発生した場合、各 PC に接続されたリーダ/ライタが、コンテナに貼り付けられた IC タグを感知し、その物の ambient 式を読み取る。PC はどのようなモノの移動が行われているのかを判断するとともに、自身の保持している部分式と IC タグから読み取った式を基に、行われているモノ移動に対応する ambient 式の遷移が可能であるかを判定する。可能であれば遷移後の ambient 式を IC タグに書き戻し、不可能であれば警告を行う。遷移可能性の判定、ならびに式の遷移は PC 中に搭載されている ambient 式の処理系により行われる。

5. 実 験

5 個のコンテナがコンテナ船千島丸に積み込まれ、仙台から東京に輸送され、東京で降ろされたコンテナは日本丸に積み込まれ上海に輸送される、といった物流システムを想定した屋内実験を行った。

この実験の中で経路指定が行えているか、コンテナの誤った移動を検出できているかを確かめた。PC は港、船、コンテナヤードに一台ずつ用意し、各 PC の OS は windows XP である。RFID 機器は次の三菱製 UHF 帯機器である。リーダライタ装置: RFIDRF-RW003, アンテナ: RF-ATCP002, タグ: RF-TGM005(64Kbit)。以上の機器構成でアンテナから 5m まで離れた場所でタグへの読み書きが可能であった。

5.1 誤った移動の検出実験

仙台港のコンテナヤード (CY) にあるコンテナの荷積み時に仙台港の PC でエラー検査を行った。積み間違いを検出すると、エラーが PC のモニタに出力された。同様にコンテナの降ろし間違い、コンテナの積み忘れ、降ろし忘れを検出すると、エラーが PC のモニタに出力する。

5.2 動的な経路指定実験

千島丸が東京港に入港するとコンテナの荷降ろしが行われ、その際コンテナの経路指定も行われる。この経路指定が正しく行われるかを確認する実験を行った。コンテナが東京港に降ろされるときに、このコンテナに貼り付けられた IC タグには式 (10) のような ambient 式が記入されている。

$$CO[query[out\ CO.out\ CY.in\ router.in\ to\ 上海] | !open\ ans] \quad (10)$$

このコンテナが船から下ろされる際、港の PC に接続された RFID リーダがタグを読み取り、式 (10) がその PC の処理系で処理される。この時 *query ambient* が、*router ambient* 内に移動し、次に乗る船や降りる港などの情報を問い合わせる。*router ambient* が持つ *ambient* 式のうち、今回の経路指定に関わる *ambient* 式を単純化した式を以下に示す。

```
router[
  to 上海 [
    open query.ans[
      out to 上海.out router.out CY.in 千島丸.in CO.
      out CY.in 日本丸.out 日本丸.in 上海.query[]
    ] ] ]
```

式 (10) の *queryambient* は *router ambient* 内の *to 上海 ambient* 内に移動する。*to 上海* は *query* が移動してくると、コンテナ *ambient* に上海へ行くための経路が記述された制御用 *ambient ans* を返す。コンテナ *ambient* 内に移動した *ans ambient* は式 (10) が持つ *open ans* capability によって *open* される。そしてコンテナ *ambient* は上海への経路を取得する。経路取得後のコンテナ *ambient* の *ambient* 式を以下に示す。

$$CO[out\ CY.in\ 日本丸.out\ 日本丸.in\ 上海.query[]]$$

東京港でこのコンテナを下ろした際、以上の動作が起こり上海への経路情報がタグに書き込めた。このような動作を 5 個のコンテナで連続して行い、いずれも正しく経路情報をコンテナのタグに伝えることができていたことを確認した。この実験の結果から、輸送経路指定表の記述にしたがって中継港でのコンテナ船の乗り換えを *ambient* 式で記述し、移動を監

視することが出来ていることが確かめられた。

6. 結 論

本論文では、ハブアンドスポークシステムのような現実的な海上物流に対応した物流監視システムを構築について議論した。本システムはコンテナの積み替えを *ambient* 式で記述するための書類として、輸送経路指定表を用意し、輸送経路指定表の内容を *ambient* 式で記述した制御用 *ambient router ambient* を生成する。コンテナは港に入ると、*router ambient* に次に乗る船を問い合わせ、次に乗る船とその船から降りる港を決定する。現実の海上物流では、コンテナが乗り換える船は、中継港がその都度指定しているために、コンテナが次に乗る船を中継港で決定する本システムの動作は、現実の物流に則した動作である。

またコンテナの移動の検知に RFID を利用することにより、コンテナの移動が物流書類の内容を記述した *ambient* 式の遷移に沿った移動かの照合が自動的に行えるようになった。照合した結果、コンテナの移動が *ambient* 式の遷移に沿わない移動だった場合、その間違った移動がコンテナの積み間違い、積み忘れなのかと言う間違った移動に対してどのような間違いであるかを判別できるようになった。その際 *ambient* 式から正しい移動を出力することが出来る。今後屋外でのより大規模な実験を予定している。

参 考 文 献

- 1) Cardelli, L. and Gordon, A.D.: Anytime Anywhere Modal Logics for Mobile Ambients, *Proceedings of the 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) 今井昭夫: 国際海上コンテナ輸送概論, 東海大学出版会 (2009).
- 4) 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, Vol.3, No.1 (2010).
- 5) 国土交通省: 「メコン地域陸路実用化走行試験」～インドシナ半島物流を変える陸路物流の実用化へのチャレンジ～. http://www.mlit.go.jp/kisha/kisha07/15/151018_.html.
- 6) 国土交通省: 米国国土安全保障省及び国土交通省による海上貨物追跡タグシステム (MATTS) の通信能力実証実験. http://www.mlit.go.jp/kisha/kisha07/11/110427_.html.
- 7) 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) (2007).
- 8) 山口範高: 貿易書類の見方・書き方, 同文館出版 (2007).