

2K-9

Ambient Calculus による物流システムの記述と検証

森本 大輔[†] 加藤 暢[‡] 樋口 昌宏[†]

近畿大学大学院総合理工学研究科エレクトロニクス系工学専攻[†]

近畿大学理工学部情報学科[‡]

1. はじめに

近年、国際物流の取扱量は飛躍的に増大している。貿易実務の世界は書類中心の社会であるが、IT化の流れにより電算処理が取り入れられ始めている。例えば食品の生産履歴などを参照可能とするためにICタグなどを取り付けトレーサビリティを確保しようという動きなどである。しかし、実際のモノの流れは適切に管理されてるとは言えず、貨物の取り違えなどが発生している現状がある。そこで、我々は解決策の一つとして Ambient Calculus 1) による物流システムの形式的記述に基づく、物流管理システムの構築について研究を進めている。本報告では具体的な物流システムを Ambient Calculus で記述し、検証を行った結果について述べる。

2. 物流管理システム

物流システムは、荷物、コンテナ、コンテナ船といった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。この構造を図1に例示する。

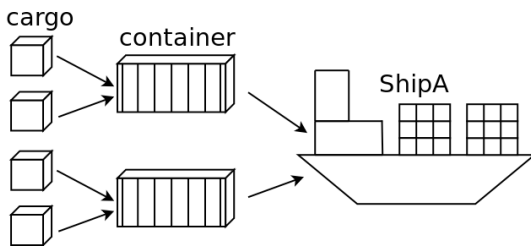


図1: 物流システムの持つ階層構造の例

本報告で提案するシステムは、図1で示すような階層構造を形式的に記述し、現実の物流システムをモデル化するものである。これにより、貨物が意図した場所に正しく運ばれることなどの性質を捕らえることが可能となる。

3. Ambient Calculus

Ambient Calculus は、Luca Cardelli ら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ための言語である。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できる。さらに様相論理の一種である Ambient Logic 2) の公理系を用いて記述の正当性を形式的に検証することが可能である。Ambient Calculus の構文規則は文献 1) により以下のよう

$$\begin{aligned}
 \text{name} &::= n \\
 \text{variable} &::= x \\
 P &::= 0 \mid (v \text{name})P \mid P \mid Q \mid !P \mid M \mid P \mid M.P \\
 &\quad \mid (\text{variable}).P \mid \langle M \rangle \\
 M &::= \text{variable} \mid \text{name} \mid \text{in } M \mid \text{out } M \mid \text{open } M \mid \varepsilon \mid M.M
 \end{aligned} \tag{1}$$

Ambient Calculus では式間の遷移関係を定義する遷移規則が与えられている。単純な遷移の例として、コンテナ船の中に存在しているコンテナが船の中から出て行く動作について考える。

$$\text{ship}[\text{container}[\text{out container}]] \rightarrow \text{ship}[\] \mid \text{container}[\] \tag{2}$$

out は Ambient を外へ移動させる能力を表現している。式(2)では、元々 ship の中にあった container が、ship の外へと移動する遷移を表している。他の動作として、Ambient を他の Ambient の中に移動させる in、Ambient の境界を消滅させる open などがある。我々はすでに Ambient Calculus を実際に動作させる処理系を開発している 3)。これは Ambient Calculus で書かれた式を実際に遷移させることにより、現実の物の動作をシミュレートする能力を持つ。

4. 物流システムの Ambient Calculus による記述

4.1 コンテナ移動の Ambient 表現

二つの港 PORTA と PORTB が存在し、PORTA にあるコンテナヤード (CY) に荷物 (container) が入っているとい

A specification of a physical distribution system in Ambient Calculus and its verification
 speaker : Daisuke Morimoto
[†]Kinki University Interdisciplinary Graduate School of Science and Engineering
[‡]Kinki University School of Science and Engineering Department of Informatics

う状態は式 (3) で表現される。

$$\begin{aligned}
& PORTA[CY[container[Q]]|ShipA[R]]|PORTB[CY[]] \\
Q \triangleq & out\ CY.in\ ShipA.dep[out\ container \\
& .open\ departure]|open\ arr.out\ ShipA.in\ CY \quad (3) \\
R \triangleq & open\ dep.departure[out\ PORTA.in\ PORTB \\
& .open\ arrival]|arrival[arr[in\ container]]
\end{aligned}$$

ここで Q と R は Ambient の移動を制御するための式である。

式 (3) の中で、最初に遷移するのは *container* の中に含まれている *out CY.in ShipA* しかない。これが遷移することにより、*container* は *CY* を出て *ShipA* の中に移動する。このときの状態は式 (4) で表される。

$$\begin{aligned}
& PORTA[CY[]] \\
& |ShipA[open\ dep.departure[out\ PORTA \\
& .in\ PORTB.open\ arrival] \\
& |arrival[arr[in\ container]] \quad (4) \\
& |container[dep[out\ container.open\ departure] \\
& |open\ arr.out\ ShipA.in\ CY]] \\
& |PORTB[CY[]]
\end{aligned}$$

次に遷移可能なのは、*dep* の中に含まれている *out container* のみである。これにより *dep* が *container* を出て *ShipA* の中に移動。その後唯一、遷移可能である *ShipA* の中の *open dep* により *dep* は境界を失い、*open departure* が実行され、*out PORTA.in PORTB.open arrival* まで遷移可能となる。この遷移により、*ShipA* 自体が *PORTA* を出て *PORTB* に入るという遷移が行われる。*open arrival* が実行される直前の式は式 (5) で表される。

$$\begin{aligned}
& PORTA[CY[]] \\
& |PORTB[CY[]]|ShipA[open\ arrival \\
& |arrival[arr[in\ container]] \quad (5) \\
& |container[open\ arr.out\ ShipA.in\ CY]]
\end{aligned}$$

次に *open arrival* が実行可能となり、*arrival* の境界が消滅することにより *arr* が露出し、その中に含まれている *in container* が遷移する。*container* に含まれている *open arr* が遷移することにより *out ShipA.in CY* が遷移可能となり *container* が *PORTB* に存在する *CY* に移動する。このときの状態は式 (6) で表される。

$$\begin{aligned}
& PORTA[CY[]] \\
& |PORTB[CY[container[]]]|ShipA[]] \quad (6)
\end{aligned}$$

以上で、コンテナを模した Ambient がある港にあるコンテナヤードから、別の港にあるコンテナヤードに必ず移動することが形式的に記述できた。

4.2 Ambient Logic の適用

Ambient Calculus のプロセス式が望まれる性質を満たしているかどうかを Ambient Logic を用いて示すことが可能である。例えば式 (3) の *container* が最終的に *PORTB* のコンテナヤード (*CY*) に移動することが望まれる。

荷物 (*container*) が *PORTB* のコンテナヤード (*CY*) の中にあるという性質は、Ambient Logic で記述すると以下のようになる。

$$\begin{aligned}
& PORTA[T@CY] \\
& |PORTB[CY[T@container]]|T@ShipA \quad (7)
\end{aligned}$$

式 (3) に望まれる性質、すなわちいつか必ず論理式 (7) を満たすという性質は以下のようになる。

$$\begin{aligned}
& \diamond PORTA[T@CY] \\
& |PORTB[CY[T@container]]|T@ShipA \quad (8)
\end{aligned}$$

式 (6) が論理式 (7) を満たすこと及び、式 (3) が式 (6) に必ず遷移することを確認することで、式 (3) が論理式 (8) を満たすこと、つまり貨物運送の初期状態が正当なものであったことを検証した。

5. 今後の方針

本報告では、物流管理システムの一部として Ambient Calculus を用いて貨物の流れを形式的に記述した。今後、物流管理システムを構築していく上で、実際の貨物と Ambient Calculus 式の遷移をどのように対応付けるかを考える必要がある。例えばバーコードリーダーなどで貨物が移動したことを把握し、その移動が正当なものであるかを Ambient プロセス式の遷移で判別する仕組みなどが考えられる。

参考文献

- 1) Cardelli, L. and Gordon, A. D.: Mobile Ambients, *LNCS*, Vol. 1872, pp. 333–347 (2000).
- 2) Cardelli, L. and Gordon, A. D.: Ambient Logic, *Mathematical Structures in Computer Science* (2003).
- 3) 西本規善: JavaCC による Ambient Calculus 処理系の実装に関する研究～Typed Ambient Calculus への拡張～, 2003 年度近畿大学卒業論文 (2004).