A Freight Management System with the Ambient Calculus

TORU KATO
Kinki University
Department of Informatics
Kowakae 3 4 1, Higashi Osaka
Japan
kato@info.kindai.ac.jp

MASAHIRO HIGUCHI
Kinki University
Department of Informatics
Kowakae 3 4 1, Higashi Osaka
Japan
higuchi@info.kindai.ac.jp

Abstract: We investigate a management system that ensures the correctness of cargo distribution in freight systems. The system determines the correctness by comparing the movement of containers, which is sensed by IC tags with formal models. These models are written in the ambient calculus and generated automatically based on several documents used in real freight systems. The ambient calculus is a formal description language that is suitable for representing freight systems with nested structures that dynamically change. An implementation of the system and the results of a simple experiment using it are presented.

Key–Words: Freight System, RFID, Ambient Calculus, Formal Model

1 Introduction

In the field of distribution, the increase in cargo handling errors is a serious problem as the amount of freight circulation increases. To deal with this problem, several systems using IC tags have been developed for the management of containers. For example, Ref. [4] shows an experiment by the Japanese government that used GPS to trace the land routes of trucks carrying luggage with IC tags. Ref. [5] shows a cooperative experiment by the Japanese and US governments that tried to trace the location of 100 containers with the Marine Asset Tag Tracking System (MATTs) during navigation from Yokohama Port to the Port of Los Angeles.

The information that those systems try to track is, however, restricted to the location of cargo. We paid attention on the nested structure of freight systems, that is, packages (e.g. containers) containing smaller packages (e.g. luggage) that are accommodated by a larger entity such as a container ship (Fig.1). Using the information of such nested structure, we are able not only to trace the route of containers but also to check their movement more precisely, including problems such as incorrect loading (or unloading) to (or from) ships.

We are researching a distribution management system based on descriptions of freight systems in the ambient calculus. The ambient calculus [1] (AC) is a formal description language originally developed for describing the nested structure of the Internet or the behavior of mobile agents whose nested structure dynamically changes. The ambient calculus is suitable for modeling freight systems, which also have a dynamically changing nested structure.

We propose an automatic generating system of AC formulae, and a management system for the actual freights. The former system generates formulae that express the transportation way of containers based on several documents such as shipping invoices, B/L instructions, container packing lists, and routing plan tables of ships. The latter system checks the movement of containers that have IC tags by comparing them with the reduction of the formulae. We also present the results of a simple experiment to show the effectiveness of these systems.

This paper is organized as follows. Section 2 shows the freight systems we concentrated on and the documents used in actual trade. In Section 3, we review the syntax and the operational semantics of AC. Sections 4 and 5 explain our freight management system. In Section 6, we show the results of an experiment on the system.
2 Freight Systems and Documents

In freight systems called full container load (FCL), each container is occupied by the luggage of one owner. The FCL procedure is as follows. Containers that have already been packed by each sender wait in the container yard of a port to be loaded. After the arrival of the container ship, those containers are brought from the container yard to the ship. The ship leaves the port for its destination port after all its containers are loaded. Several hundred to 8,000 containers can be loaded on one ship.

Those containers are unloaded and carried to a container yard after the ship arrives at their destination port. The ship leaves that port after those containers are unloaded there.

The movement of all the containers and the shipping vessel for a shipment are specified by several documents such as shipping invoices, B/L instructions, container packing lists, and routing plan tables of the ship. Some examples of these are shown in Figs.2 and 3.

Example of shipping invoice

<table>
<thead>
<tr>
<th>Means of Transport &amp; Route</th>
<th>On or About</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shipped on Ship A</td>
<td>From TOKYO, JAPAN</td>
</tr>
</tbody>
</table>

Figure 2: Example of shipping invoice

Example of B/L Instruction

<table>
<thead>
<tr>
<th>Ship Name</th>
<th>Container No.</th>
<th>Port of Loading</th>
<th>Port of Discharge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ocean Vessel v1</td>
<td>TOKYO, JAPAN</td>
<td>KOBE, JAPAN</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Example of B/L Instruction

The purpose of the management system presented is to automatically check whether containers are carried accurately in accordance with these documents during transport from a departure container yard to a destination container yard. To do so, the system generates a formal model of the freight system from those documents so that machine checking is possible.

3 Ambient Calculus

This section reviews the syntax and the semantics of AC originally defined in Ref. [1].

We assume there are infinite sets of names ranged over by m, n. Capabilities and processes are ranged over by M, N and P, Q, respectively.

Definition 3.1 (Syntax)

\[ n \mid \text{names} \quad P, Q \quad \text{:= processes} \]

\[ M, N \quad \text{:= capabilities} \quad 0 \quad \text{inactivity} \]

\[ \text{in } n \quad \text{can enter } n \quad P[Q] \quad \text{composition} \]

\[ \text{out } n \quad \text{can exit } n \quad n[P] \quad \text{ambient} \]

\[ \text{open } n \quad \text{can open } n \quad M.P \quad \text{action} \]

\[ \epsilon \quad \text{null} \quad (n \in P) \quad \text{restriction} \]

\[ M, N \quad \text{path} \]

The capability “\text{in } n” means the ambient that encloses it can go into the ambient \text{n}[\]. The capability “\text{out } n” means the reverse of “\text{in } n”. The capability “\text{open } n” dissolves the ambient \text{n}. These actions are prescribed in the reduction rules in Definition 3.2.

Definition 3.2 (Reduction: \( \rightarrow \))

\[ n[m \cdot P | Q] \quad m[R] \rightarrow m[n[P | Q] | R] \]

\[ n[m[out] m \cdot P | Q] | R | n[P | Q] | m[R] \]

\[ \text{open } n \cdot P \quad n[Q] \rightarrow P | Q \]

\[ P \rightarrow Q \Rightarrow P | R \rightarrow Q | R \]

\[ P \rightarrow Q \Rightarrow (m)P \rightarrow (m)Q \]

\[ P \rightarrow Q \Rightarrow (n)P \rightarrow (n)Q \]

\[ P \rightarrow Q \Rightarrow n[P] \rightarrow n[Q] \]

\[ P'' \equiv P, P \rightarrow Q, Q \equiv Q' \Rightarrow P'' \rightarrow Q' \]

In the first rule of Definition 3.2, the capability “\text{in } m” leads the ambient \text{m}[\] into the ambient \text{m}[\]. Then the capability is consumed and the process on the left hand side is reduced to that on the right hand side.

The syntax of AC originally defined in [1] has \( x \) variables, (\( x \)) (input), (\( M \)) (output) and !P (replication) that are omitted in Definition 3.1 and 3.2 because we do not use them for representing the behavior of freight systems in this paper.

4 Modeling Freight Systems by AC

Formulae of AC that represent the behavior of freight systems are automatically generated by the system from Excel files of the types of trade documents shown in Section 2.

These formulae represent the current state of the freight system and the possible future behavior of objects (ships, ports, container yards, and containers). The formulae are composed of two kinds of ambients: physical ambients and control ambients. The former represent real objects such as ships, ports, container yards, or containers, the latter are used for restricting the movement of physical ambients. This section shows how freight systems are modeled by those ambients and explains the role of control ambients.
4.1 Example of Ambients

We give an example of formulae generated by the system below that represent a shipment of 500 containers from Tokyo Port to Kobe Port.

\[
SHIP_{v1} \quad \text{in } TOKYO . \text{open cnt.} . \cdots \cdot \text{open cnt.out } TOKYO \\
\quad \text{in } KOBE . \text{open cnt.} . \cdots \cdot \text{open cnt.out } KOBE \\
\quad \open lcomp_{a1}.cnt[ ] \cdots \open lcomp_{a500}.cnt[ ] \\
\quad \open ulcomp_{a1}.ucnt[ ] \\
\quad \cdots \cdots \open ulcomp_{a500}.ucnt[ ] 
\]

(1)

\[
TOKYO[ \\
load_{v1} . \text{in } SHIP_{v1}.out \ SHIP_{v1}.in \ CY ] \\
\]

(2)

\[
CY[ \\
CO_{a1} . \text{in } load_{v1}.out \ load_{v1}.out \ CY \\
\quad \text{in } SHIP_{v1}.lcomp_{a1}.out \ CO_{a1} \\
\quad \text{in } uoload_{v1}.KOBE . \text{out } uoload_{v1}.KOBE \\
\quad \text{out } SHIP_{v1}.in \ CY \\
\quad \ulcomp_{a1}.out \ CO_{a1}.out \ CY \\
\quad \text{in } SHIP_{v1}[ ] \\
\quad \cdots \cdots \ CO_{a500}[ \cdots \cdots ] 
\]

(3)

\[
KOBE[uoload_{v1}.KOBE . \text{in } SHIP_{v1} ] \\
\]

(4)

\[
CY[ upreload_{a1} . \text{in } CO_{a1}.out \ CO_{a1} \\
\quad \text{out } CY . \text{in } SHIP_{v1}[ ] \\
\quad \cdots \cdots \ ulcomp_{a500}[ \cdots \cdots ] 
\]

The formula (1) represents a ship that will enter Tokyo Port, load containers there, and carry them to Kobe Port. The formula (2) represents Tokyo Port and its container yard where 500 containers to be loaded have been loaded. The formula (3) represents Kobe Port. All of the formulae are in the ambient $SEA[ ]$.

4.2 Control Ambients

The ambients with capitalized names are physical ambients and the other ambients are control ambients. The control ambients restrict the movement of physical ambients. For example, in the formula (1), the ambient $SHIP_{v1}$ has a capability $out \ TOKYO$ so that the ship can leave Tokyo Port. The ambient, however, does not get out from the $TOKYO$ ambient because that capability is blocked by the preceding capabilities $open \ cnt$. Those capabilities will be consumed after all the ambients $CO_{a\ast}$ enter the ambient $SHIP_{v1}$ and all the control ambients $lcomp_{a\ast}[ ]$ appear in $SHIP_{v1}$. This represents a restriction on the real container ship, that is, it can leave the port only after all of the containers that must be loaded have been loaded.

Another example of a control ambient is the ambient $load_{v1}$ in the formula (2). It waits for the ambient $SHIP_{v1}$ to enter the ambient $TOKYO$ and it enters and leaves $TOKYO$ after $SHIP_{v1}$ enters $TOKYO$. Then, it goes into the ambient $CY$ in $TOKYO$ to enable all the ambients $CO_{a\ast}$ to exit the ambient $CY$. This means containers are brought out from the container yard only after the container ship arrives.

5 Freight Management System

The freight management system maintains the formulae representing the current state of the freight system. The system determines whether the container handling is valid with sensing the movement of container by RFID and checking the reduction representing the movement is possible in the current formulae.

5.1 AC processing system

The AC processing system [3] reduces process (formula) according to Definition 3.2 on distributed environments. Each processing system contains several ambients (a part of the entire formula) and refers remote ambients if necessary.

In the processing system, physical ambients and control ambient are reduced in different way:

**physical ambients** When the movement of a real object (ship or container) is notified to the processing system, the system checks whether the corresponding reduction is possible for the formula representing the object. If possible, the system ensures the movement is valid and reduces the formula, else the system issues a warning.

**control ambients** The reductions related to control ambients are made in the processing system as soon as they become possible.

5.2 System Configuration

The devices used in our system are shown in Fig.4. We will set up a PC with an AC processing system in the office of each port, in each container yard, and in the cargo room of each ship. RFID reader/writers are connected to the PCs in the container yards and cargo rooms of the ships. An IC tag is put on each container.

We put the formulae generated from the trade documents to the appropriate PCs and IC tags. Some of the formulae are simplified and decomposed for distributed management. When, for example, containers are shipped from Tokyo Port to Kobe Port, the formula (2) is simplified and decomposed into the formula (4), (5) and (6), and distributed to the PC in the
5.3 Checking the Container Handling

Whenever a container exits or enters a container yard or a ship, the RFID reader/writer there reads the IC tag on the container and the management system on the PC determines if that movement is valid.

Container Exiting

In Figure 5 the system determines handling of a container is not valid because the container CO_a1 is brought out from the container yard before the ship arrives at TOKYO port.

Figure 4: System configuration

Tokyo Port, the PC in container yard of Tokyo Port and IC tags of the containers respectively.

\[ \text{TOKYO}[ \text{load}_v1[\text{in SHIP}_v1.\text{out SHIP}_v1.\text{in CY}]] \]

\[ \text{CO}_a1[\text{in load}_v1.\text{out load}_v1.\text{out CY}} \]

\[ \text{CY}\text{[CO}_a1\text{]} \]

\[ \text{CO}_a1[\text{in load}_v1.\text{out load}_v1.\text{out CY}} \]

\[ \text{CY}\text{[CO}_a1\text{]} \]

\[ \text{CY}\text{[CO}_a1\text{]} \]

\[ \text{CY}\text{[CO}_a1\text{]} \]

This means each of the processing system in each PC maintains a physical ambient i.e. maintains all the control ambients in it and the name of all the physical ambients in it. The distributed management like this enables us to suppress the traffic of communication between PCs for managing freight systems. When we need to check the locations or the states of all objects, all the PCs communicate and we can make the entire formula representing the state of the freight system.

Figure 5: Checking movement (exiting container yard)

When the container CO_a1 passes the gate of the container yard, the RFID reader/writer reads the IC tag and the processing system gets the formula (6). The processing system finds the corresponding reduction is impossible because of the capabilities in load_v1.out load_v1 in front of the out CY. So the management system shows a warning. This means the ship SHIP_v1 has not yet arrived and the handling is invalid.

After the ship arrives, PCs in the ship, port, and container yard communicate each other, and the ambient load_v1[\text{in SHIP}_v1] comes into the PC at the container yard as follows.

Figure 6: Communication between PCs

When the ship SHIP_v1 is about to enter Tokyo Port, the PC on the ship transmits the formula SHIP_v1[\text{in TOKYO}]. That is the communication (a) illustrated in Fig.6. As the formula TOKYO[\text{in \ldots}] SHIP_v1[\text{in TOKYO}] can be reduced to
TOKYO[SHIP,v1[· · · ]], the PC at Tokyo Port has the formula in Fig. 6. After several reductions, the formula is reduced to (9).

However, the PC that maintains the ambient CY is in the container yard, so the PC at the port transmits the ambient load,v1[· · · ] to the PC in the container yard. That is the communication (b) illustrated in Fig. 6.

As a result, when the container with the ambient CO,a1 passes the gate and the RFID reader/writer reads the IC tag, the formula on the PC of the container yard is reduced to (10).

\[
CO,a1[in load,v1.out load,v1.out CY · · · ] \\
[ · · · | load,v1[· · · ]]
\]

The container with the ambient CO,a1 can exit the container yard because the capabilities in load,v1.out load,v1 are enabled and consumed immediately. When the container exits the container yard, the RFID reader/writer writes back the formula (11) to the IC tag of the container, which is the remains of the reduction of (6).

\[
CO,a1[in SHIP,v1.lcomp,a1[out CO,a1] \\
|in unload,v1.KOBE.out unload,v1.KOBE .out SHIP,v1.in CY \\
ulcomp,a1[out CO,a1.out CY .in SHIP,v1]]
\]

6 Experiment
This section shows a simple experiment assuming that we simulate the transport of five containers from Tokyo Port to Kobe Port on one container ship.

First, we confirmed that the formula generating system generated the formulae (1), (3), (4), (5) and 5 formulae like (6) for containers from the documents related to the shipping of five containers from Tokyo Port to Kobe Port. (We also confirmed the system could generate formulae for more than 1,000 containers). We wrote each formula of container (6) on an IC tag that was put on a box representing a container (Fig. 8), and deployed the formulae for the ship (1), on the container and the processing system gets the formula (11). The formula on the PC on the ship is reduced to (12).

\[
SHIP,v1[· · · |CO,a1[in SHIP,v1 · · · ]]
\]

Since the formula (12) can be reduced to (13), the management system confirms the loading of the container.

\[
SHIP,v1[· · · |CO,a1[· · · ]]
\]

Then the reader/writer write back the ambient CO,a1[· · · ] in the formula (13) to the IC tag.

When a container is about to be loaded onto the wrong ship, for example the ship with the ambient SHIP,v2, then the PC on the ship has the formula (14) that cannot be reduced any more, and the management system rejects the loading by showing a warning on the PC monitor.

\[
SHIP,v2[· · · |CO,a1[in SHIP,v1 · · · ]]
\]
Tokyo Port (4), its container yards (5), Kobe Port (7) and its container yards (8) on their respective PCs.

Then we started checking. As shown in Fig.8, we moved boxes (containers) by hand and made sure the RFID reader/writer could read the formulae, the processing system reduced those formulae, the PCs transmitted the formulae, the management system determined the validity of the movement, and the RFID reader/writers wrote back the formulae on the IC tags after the reductions. In Fig.9, the windows of the GUI system show the contents of all formulae just after the experiment started. Each rectangle in the window represents a real object such as a port or a ship. Clicking a rectangle shows more precise information.

When we moved a container correctly, it took 7.8 seconds for the system to read the formula (6) from the IC tag, confirm the movement, reduce the formula to the formula (11), and write back the formula (11) onto the IC tag.

A warning dialog box appeared when we made the ship leave the port before the completion of container loading (Fig.10). The Japanese messages in the dialog read “you left something behind”, “List of the things left behind [CO_a4, CO_a5]” and “OK?”.

7 Conclusions

The freight management system presented in this paper confirms whether containers are handled correctly during a shipping by comparing the handling to formal AC formulae. The formulae are generated based on documents used in actual trading. Thus, when the system confirms the correctness of the treatment of the containers, it means the containers have been handled in accordance with the instructions in the documents.

We confirmed that the system determined the validity of the container handling in a practical amount of time, so we think will be able to test the system at a practical scale by using more efficient RFID devices.

By modeling freight systems with AC, the model checking is possible [6]. Such a model checking enables us to make sure that containers will be delivered to their destinations correctly in any case before the transport starts. To check freight systems that transport more the 1,000 containers, however, we will need a partial order reduction [2] and more sophisticated formulae than are presented in this paper.

References: