LTL で記述されるマニュアルに対するオートメーション サプライズの検出*

高橋 怜士[†]・潮 俊光[†]・足立 正和[†]

Detection of Automation Surprises for a Manual Modeled by Linear-Time Temporal Logic*

Satoshi TAKAHASHI[†], Toshimitsu USHIO[†] and Masakazu ADACHI[†]

In a human-machine system, an interface displays an abstracted state of the machine. The abstracted state may differ from that anticipated by a user. Such a phenomenon is called an automation surprise and may cause a serious human error. In this paper, we assume that the user operates the machine according to a manual which is modeled by LTL (Linear-Time Temporal Logic). We propose a formal method for detection of three automation surprises and livelocks due to the manual. First, we transform the LTL representation of the manual into a transition system. Next, we propose a composite model derived from the transition system and a machine model, which describes concurrent behavior of the machine and the user. Finally, solving a reachability problem and searching strongly-connected components for the composite model, both the automation surprises and live lock can be detected formally.

1. はじめに

人間-機械系において、ユーザは、インタフェース表示を見ながら、マニュアルに書かれた指示に従ってシステムを操作する。ここで、ユーザに提示されたマニュアルに不備があったり、インタフェース表示を抽象化しすぎて必要な情報を提示できなかつたりすると、ユーザが意図した振舞いと実際のシステムの振舞いとの間で不整合が生じ、システムからの応答がユーザの予期していないものとなることがある。このような望ましくない状況をオートメーションサプライズとよび[1]、実際にオートメーションサプライズを原因とするヒューマンエラーが重大な事故の原因となった例が報告されている[2]。

そこで、オートメーションサプライズを生じることなくユーザが適切な操作ができるために十分な情報をインタフェースが表示しているかを数理モデルを用いて検証することが重要な研究課題となっている[3-5]。たとえば、Degani と Heymann は、有限状態機械で表現されたマシンモデルとユーザモデルの並行的な動作を合成

モデルにより表し、オートメーションサプライズが起こるような状態に可到達かどうかを調べることににより、インタフェースとユーザモデルの正しさを検証する手法を提案している[6]。宇川らは Mode Confusion, Refusal State, Blocking State という3種類のオートメーションサプライズに着目し、それらを形式的に検出する方法を提案し[7]、検出ツールを開発している[8]。Rushby は Mur ϕ [9] とよばれる検証システムを用い、モデル検査法を応用してオートメーションサプライズを検出する手法を提案している[10]。また、足立らは双模倣性に基づくオートメーションサプライズが発生しないインタフェースの設計法を提案している[11]。しかし、ユーザはある目的をもってシステムを操作し、その操作手順をマニュアルとして記述することが一般的である。したがって、ユーザがマニュアルの記述どおりに操作したときにオートメーションサプライズが発生するかどうかを調べることが重要となる。

一方、LTL (Linear-Time Temporal Logic) は時間を扱うことのできる時相論理であり、離散事象システムの検証などに用いられる。たとえば、Kumar は、LTL を用いて書かれた仕様を基に、離散事象システムの故障診断を行っている[12]。LTL は論理的に人間の動作を記述する方法としても適しており、マニュアルを LTL で形式的に記述できる。また、LTL からオートマトン表現へ

* 原稿受付 2005 年 11 月 9 日

[†] 大阪大学 基礎工学部 Faculty of Engineering Science, Osaka University; 1-3 Machikaneyamacho, Toyonaka city, Osaka 560-8531, JAPAN

Key Words: automation surprise, discrete event system, linear-time temporal logic.

の変換法が提案されている [14].

本論文では、ユーザがマニュアルどおりに操作している場合を考える。すなわち、LTL で表現されたマニュアルモデルをユーザモデルと考える。マニュアルを LTL で記述するための原子命題の表現方法を提案し、この原子命題と LTL のオートマトン記述からマニュアルのトランジションシステムモデルを求める。このモデルとシステムのモデルを合成することでオートメーションサプライズを検出する方法を提案する。

本論文の構成は以下の通りである。まず 2. で議論に必要な LTL の定義を示す。3. でマシンモデルとインタフェースの表現を述べる。4. でマニュアルを LTL でモデル化し、その LTL モデルからマニュアルをトランジションシステムに変換する手法について述べる。つぎに、5. でオートメーションサプライズの検出法を提案し、例題を用いて提案手法によってオートメーションサプライズが検出できることを示す。最後に 6. で本論文のまとめを示す。

2. LTL

本論文ではマニュアルが LTL で表現できる場合を考える。本節では、そのために必要な LTL のオペレータとそのセマンティクスを説明する [13].

本論文でマニュアルは論理演算子 $\neg, \wedge, \vee$ に加えて、以下のパス限量子 **A** と線形時間論理演算子 **X**, **U** を用いて記述できるとする。

A: すべてのパス上で性質が満たされる。

X: パスの次の状態で性質が満たされる。

U: 二つの要素を持ち、片方の性質が満たされるまですう片方の性質が満たされる。

LTL には 2 種類の論理式がある。一つは特定の状態で真となる状態論理式であり、もう一つは特定のパスで真となるパス論理式である。LTL では以下の文法に基づき、状態論理式とパス論理式が定まる。

- もし $p \in AP$ ならば p はパス論理式である。
- f と g がパス論理式ならば $\neg f$, $f \vee g$, $f \wedge g$, **X** f , $f \mathbf{U} g$ はパス論理式である。
- **A** f は状態論理式である。ここで f はパス論理式である。

ここで AP は原子命題の集合である。また、True を常に真となる原子命題、False を常に偽である原子命題とする。以上のオペレータのセマンティクスを Kripke 構造 $M = (S, R, L)$ を用いて表す。ここで、 S は有限の状態集合であり、 $R \subseteq S \times S$ は状態間の遷移関係である。ここで、すべての状態 $s \in S$ について、 $(s, s') \in R$ である $s' \in S$ があると仮定する。また、 $L: S \rightarrow 2^{AP}$ は状態から AP の集合への写像であり、 $L(s)$ は $s \in S$ で真となる原子命題の集合である。また、パス π に対してその i 番目の接尾辞を π^i とおく。すなわち、 $\pi = \sigma_1, \sigma_2, \sigma_3, \dots$ としたとき、 $\pi^i = \sigma_i, \sigma_{i+1}, \dots$ である。

f を状態論理式として、 f が M の状態 s で満たされるとき $M, s \models f$ と書く。同様に、 g をパス論理式として g が M のパス π 上で満たされるとき $M, \pi \models g$ と書く。 M が文脈から明らかなき場合は、 M を省略する。また、関係 $\models$ は g_1 と g_2 をパス論理式としたとき以下の通り再帰的に定義される。

- $M, s \models \mathbf{A}g_1 \Leftrightarrow s$ から始まるすべての π で $M, \pi \models g_1$
- $\forall p \in AP, M, \pi \models p \Leftrightarrow s$ を π の最初の状態として $p \in L(s)$
- $M, \pi \models \neg g_1 \Leftrightarrow M, \pi \not\models g_1$
- $M, \pi \models g_1 \vee g_2 \Leftrightarrow M, \pi \models g_1$ または $M, \pi \models g_2$
- $M, \pi \models g_1 \wedge g_2 \Leftrightarrow M, \pi \models g_1$ かつ $M, \pi \models g_2$
- $M, \pi \models \mathbf{X}g_1 \Leftrightarrow M, \pi^1 \models g_1$
- $M, \pi \models g_1 \mathbf{U} g_2 \Leftrightarrow M, \pi^k \models g_2$ のような $k \geq 0$ が存在し、すべての $0 \leq j < k$ で $M, \pi^j \models g_1$

3. 機械とインタフェースのモデル

人間-機械系の相互作用によるオートメーションサプライズを検出するために、機械のモデルであるマシンモデルと、人間のモデルであるユーザモデル、マシンモデルの状態とユーザモデルの状態との対応関係を表すインタフェースを定義し、それらの合成モデルを用いるアプローチがある [6,7]. 本節では、論文中で用いられるマシンモデルとインタフェースを表す数理モデルを説明する。

3.1 マシンモデル

本論文ではマシンモデルは、決定的トランジションシステム G_M でモデル化されるとする。

$$G_M = (X_M, \Sigma_M, \Delta_M, x_{M0})$$

ただし、 X_M はマシンモデルの状態の集合、 Σ_M は事象の集合、 $\Delta_M \subseteq X_M \times \Sigma_M \times X_M$ は状態遷移関係、 $x_{M0} \in X_M$ はマシンモデルの初期状態である。 $(x_1, \sigma, x_2) \in \Delta_M$ は状態 $x_1 \in X_M$ から事象 $\sigma \in \Sigma_M$ の生起によって $x_2 \in X_M$ に遷移することを表す。マシンモデルは決定的であるので、 $(x_1, \sigma, x_2) \in \Delta_M, (x_1, \sigma, x'_2) \in \Delta_M$ ならば $x_2 = x'_2$ である。事象集合 Σ_M は以下のように分割できるとする。

$$\Sigma_M = \Sigma_M^o \cup \Sigma_M^u, \Sigma_M^o \cap \Sigma_M^u = \emptyset$$

ただし、 Σ_M^o はユーザによって引き起こされる事象、すなわちユーザコマンドの集合であり、 Σ_M^u はユーザが引き起こすことのできない事象、すなわち内部事象の集合である。また、内部事象はユーザが直接観測できないとするが、インタフェース表示の変化によっていずれかの内部事象が生じたことをユーザが知ることはできる。 $x \in X_M$ と $\sigma \in \Sigma_M$ に対して、 $(x, \sigma, x') \in \Delta_M$ が定義されているとき $\delta_M(x, \sigma)!$ と表記する。定義されていないときは $\neg \delta_M(x, \sigma)$ と書く。以下、他のトランジションシステムに対しても同様の表記を用いる。

3.2 インタフェース

インタフェースとは、マシンモデルの状態を抽象化し、ユーザに提示するためのものである。よって、マシンモデルの状態 $x_M \in X_M$ とインタフェース表示 $x_I \in X_I$ を結びつける関数 $I_M: X_M \rightarrow X_I$ としてインタフェースを定義する。ただし X_I はインタフェース表示の集合を表す。

4. マニュアルモデルの数理

一般にマニュアルはマシンへのコマンドと、コマンドによるインタフェース表示の変化に基づいた論理式でモデル化できる。そのため本論文では、まずマニュアルの LTL 表現を示し、次にその LTL 表現をトランジションシステムへ変換する手法を提案する。そしてマニュアルの例と変換後のトランジションシステムを示す。

4.1 マニュアルの LTL 表現

マニュアルは初期インタフェース表示 $\mathbf{x}_{man}^{init} \subseteq X_I$ と LTL の論理式 l_{man} の二つでモデル化できる。 l_{man} は $e_U \in \Sigma_M^o \cup \{emp\}$, $\mathbf{x}_{man} \subseteq X_I$ に対する二項組 $(e_U, \mathbf{x}_{man})$ を原子命題とする LTL の論理式である。ここで、“ e_U の生起後のインタフェース表示は $\mathbf{x}_{man}$ の要素である”ことを $(e_U, \mathbf{x}_{man})$ は意味する。

$e_U = emp$ とは、ユーザがコマンドを入力しないことを意味する。たとえば、 $(emp, \{x', x''\})$ は“ある内部事象 $\sigma \in \Sigma_M^u$ によって x' か x'' というインタフェース表示になる”ことを意味する。

マニュアルは操作者へ適切な操作を指示するためのものなので、以下に示す表記を禁止する。

禁止される論理積

二つのコマンドを同時に入力するという状況はない。したがって、 α, β をコマンドとしたとき、 $(\alpha, \mathbf{x}_{man}) \wedge (\beta, \mathbf{x}'_{man})$ のような原子命題同士の論理積は禁止する。ただし、 $\mathbf{x}_{man}, \mathbf{x}'_{man} \subseteq X_I$ である。

禁止される論理和

以下のようなマニュアルを基に操作を行うとする。

$$((d, \{x1, x2\}) \wedge \mathbf{X}(a, \{x4\})) \vee ((d, \{x1, x3\}) \wedge \mathbf{X}(b, \{x4\}))$$

このとき、“コマンド d を入力してインタフェース表示が $x1$ になった”という状況でユーザは状態 $x4$ に移るためにコマンド a を入力すべきか、 b を入力すべきか判断がつかない。したがって、 $((e_U, \mathbf{x}_{man}) \wedge \mathbf{X}f_1) \vee ((e_U, \mathbf{x}'_{man}) \wedge \mathbf{X}f_2)$ のとき、 $\mathbf{x}_{man} \cap \mathbf{x}'_{man} = \emptyset$ でなければならない。ただし、 f_1 と f_2 は論理式であり $f_1 \neq f_2$ である。

禁止される Until 演算子

論理式 $(e_U, \mathbf{x}_{man}) \mathbf{U}(e'_U, \mathbf{x}'_{man})$ に対して $e_U = e'_U$ である必要がある。これは $e_U \neq e'_U$ のときは、ユーザがどこで e'_U というコマンドを入力するか判断できないからである。また、上述の制限と同様に、 $\mathbf{x}_{man} \cap \mathbf{x}'_{man} = \emptyset$ で

なければならない。

4.2 マニュアルの LTL 表現からマニュアルモデルへの変換

本節ではマニュアルの LTL 表現をマニュアルモデルを表すトランジションシステムに変換する方法を説明する。

マニュアルモデルを求めるために、まず Gerth らのアルゴリズム [14] を用いて、マニュアルの LTL 表現を Büchi オートマトン $\mathcal{B} = (\Sigma, Q, \Delta, Q^0, F)$ に変換する。ここで、 Σ は原子命題と命題 True および False からなる命題の集合で $\mathcal{B}$ の事象集合である。 Q は状態の集合であり、 $\Delta \subseteq Q \times \Sigma \times Q$ は遷移関係である。 Q^0 は初期状態の集合であり、 F は受理状態の集合である。 $\mathcal{B}$ から得られるパスは事象を命題と考えたとき、与えられた LTL 式を満足する。

$\mathcal{B}$ はマニュアルが満たす命題の列を生成する。オートメーションサプライズを検出するために、マニュアルと整合した $\Sigma_M^o \cup \{emp\}$ 上のコマンド列を生成するとトランジションシステム $G_U = (X_U, \Sigma_U, \Delta_U, x_{U0})$ 、ユーザの目標状態の集合 $X_{UF} \subseteq X_U$ 、および G_U の各状態 $x_U \in X_U$ でのユーザインタフェース表示 $I_U(x_U) \subseteq X_I$ を表す対応関係 $I_U: X_U \rightarrow 2^{X_I}$ を、 $\mathcal{B}$ から求める。ここで、原子命題 (e_U, x_{man}) の定義からコマンド入力によってインタフェース表示の変化が一意に決まらないために $I_U(x_U)$ はインタフェース表示の集合となる。一般にマニュアルに基づいて有限回のコマンドを入力してユーザの目的が達成され、その目的が達成された状態が X_{UF} である。また、 $(q_1, \sigma, q_2) \in \Delta$ で σ に論理積や True, False が含まれていた場合ユーザがその操作を行うことはできないため、そのような遷移関係は G_U への変換時に削除される。

- $X_U = Q$ とする。すなわち、状態に関しては Büchi オートマトン $\mathcal{B}$ の状態をそのまま利用する。
- $\Sigma_U = \{e_U | \Sigma \text{ に含まれる原子命題 } (e_U, \mathbf{x}_{man})\} \subseteq \Sigma_M^o \cup \{emp\}$ である。 Σ_U はユーザマニュアルに従って操作するときに入力する可能性のあるコマンドの集合であり、システムの状態変化によるユーザインタフェース表示の変化がある場合には事象 emp を含む。
- x_U を $q_1 \in Q$ と対応する状態、 x'_U を $q_2 \in Q$ と対応する状態として、 $(q_1, \sigma, q_2) \in \Delta$ であり、 σ に含まれる原子命題 $(e_U, \mathbf{x}_{man})$ がただ一つであるときに、 $(x_U, e_U, x'_U) \in \Delta_U$ とする。
- x_U を $q_2 \in Q$ と対応する状態として、ある $q_1 \in Q$ に対し、 $(q_1, \sigma, q_2) \in \Delta$ であり、 σ に含まれる原子命題 $(e_U, \mathbf{x}_{man})$ がただ一つであるときに、 $I_U(x_U) = \mathbf{x}_{man}$ とする。ただし $I_U(x_{U0}) = \mathbf{x}_{man}^{init}$ とする。Gerth らのアルゴリズムの性質から、ある状態 $q \in Q$ に遷移する際の事象 σ は遷移元が複数存在しても同一である。そのため、事象 σ によって q へ遷移する枝が

複数存在しても q に対応する x_U での $I_U(x_U)$ は一意に定まる.

- $x_{U0} = Q^0$ とする.
- X_{UF} は $q \in F$ のうち, True による自己ループのみしか遷移を持たない状態を q' としたとき, q' の前の状態と対応する X_U の状態である.

例題

“インタフェース表示 x のとき, コマンド a を入力すると, インタフェース表示 x' になり, その後しばらく待つとインタフェース表示 x'' の状態に変化する.” というマニュアルは以下のように記述できる.

$$\begin{aligned} \mathbf{x}_{man}^{init} &= \{x\} \\ l_{man} &= (a, \{x'\}) \wedge \\ &\quad (\mathbf{X}((emp, X_I - \{x''\}) \mathbf{U}(emp, \{x''\}))) \end{aligned}$$

以上の式をマニュアルモデルへ変換すると Fig. 1 となる. ここで, 状態のラベルは, I_U によって求められたインタフェース表示を表している. このとき, $X_{UF} = \{x''$ のラベルを持つ状態} である.

Fig. 1 Example of manual model

4.3 インタフェース表示の集合を用いたマニュアルの簡略化

本論文では, マニュアルの原子命題 ($e_U, \mathbf{x}_{man}$) の $\mathbf{x}_{man}$ が集合である. そのため, 論理和を用いるかわりにインタフェース表示の集合を用いてマニュアルを記述することができる.

以下のようなマニュアルを考える.

- (1) インタフェース表示 $init$ のとき, コマンド d を入力するとインタフェース表示 $x1, x2, x3, x4$ のどれかになる. ここで, コマンド d を入力し続けるとやがてインタフェース表示 $x4$ となる
- (2) 次にコマンド a を入力するとインタフェース表示 $x5$ になる
- (3) 次にコマンド c を入力するとインタフェース表示 $x6$ になる

というマニュアルは以下のように記述できる.

$$\begin{aligned} \mathbf{x}_{man}^{init} &= \{init\} \\ l_{man} &= ((d, \{x1\}) \vee (d, \{x2\}) \vee \\ &\quad (d, \{x3\})) \mathbf{U}((d, \{x4\}) \wedge \\ &\quad \mathbf{X}((a, \{x5\}) \wedge \mathbf{X}(c, \{x6\}))) \end{aligned}$$

以上の式をマニュアルモデルへ変換すると Fig.2 となる. このマニュアルはインタフェース表示の集合を用いて以下のように記述することができる.

$$\begin{aligned} \mathbf{x}_{man}^{init} &= \{init\} \\ l_{man} &= ((d, \{x1, x2, x3\}) \mathbf{U}((d, \{x4\}) \wedge \\ &\quad \mathbf{X}((a, \{x5\}) \wedge \mathbf{X}(c, \{x6\})))) \end{aligned}$$

以上の式をマニュアルモデルへ変換すると Fig.3 となる.

インタフェース表示の集合を用いてマニュアルを書いた場合, 集合のすべての要素が合成時に用いられるとは限らず, 冗長な表現となることがある. しかし, マニュアルモデルの状態数および遷移関係の数が減るので効率的に計算ができる.

Fig. 2 Manual model from formula with logical additions

Fig. 3 Manual model from formula with set of the displays

5. オートメーションサプライズの検出

本節では, オートメーションサプライズを定義する. つぎに, マシンモデルとマニュアルのトランジションシステム表現を合成し, オートメーションサプライズを検出するための手法を提案する. 最後に, 例題を用いて提案手法が有効であることを示す.

5.1 オートメーションサプライズの定義

本論文では, Mode Confusion, Refusal State, Blocking State という 3 種類のオートメーションサプライズ [7] に着目する. これらの定義を以下に示す.

【定義 1】 (Mode Confusion) 次式を満たす $\beta \in \Sigma_U$ が存在するとき, $(x_M, x_U) \in X_M \times X_U$ の次の状態, $(\delta_M(x_M, \beta), \delta_U(x_U, \beta))$ を Mode Confusion という.

$$I_M(\delta_M(x_M, \beta)) \notin I_U(\delta_U(x_U, \beta)) \quad (1)$$

Mode Confusion は, ユーザがコマンドを入力した後, マニュアルと違うインタフェース表示になるオートメーションサプライズである. たとえば, “インタフェース表示 A のとき, コマンド b を入力するとインタフェース表示 B になる” というマニュアルが与えられ, ユーザが

コマンド b を入力したとき、インタフェース表示が B 以外になるという状況を示す。

【定義 2】 (Refusal State) 次式を満たす $\beta \in \Sigma_U$ が存在するとき、 $(x_M, x_U) \in X_M \times X_U$ を Refusal State という。

$$((\neg \delta_M(x_M, \beta)!) \wedge (\delta_U(x_U, \beta)!)) \wedge (\beta \neq emp) \quad (2)$$

Refusal State はユーザが入力したコマンドにマシンが反応しないというオートメーションサプライズであり、マニュアルではコマンドを入力可能であると書かれているにもかかわらず、マシンモデル上ではそのコマンドに対応する遷移がない場合に発生する。本論文ではマニュアルにユーザが何もしないことを表す emp というコマンドを用いることができるため、本定義ではそれが考慮されている。

【定義 3】 (Blocking State) 次式を満たす $\beta \in \Sigma_M^u$ と、 $x'_M \in X_M$ が存在するとき、 $(x_M, x_U) \in X_M \times X_U$ を Blocking State という。

$$((x_M, \beta, x'_M) \in \Delta_M) \wedge (\neg \delta_U(x_U, emp)!) \wedge (I_M(x'_M) \notin I_U(x_U)) \quad (3)$$

Blocking State はマシンモデルの内部事象によってインタフェース表示がマニュアルで書かれていない形で勝手に変化することを表すオートメーションサプライズである。マシンモデル上には内部事象による遷移でインタフェース表示が変わることがあるにもかかわらず、マニュアルにそれが書かれていない場合に発生する。

5.2 マシンモデルとマニュアルモデルの合成

オートメーションサプライズを検出するために合成モデル G_{com} を定義する。

$$G_{com} = (X_{com}, \Sigma_{com}, \Delta_{com}, X_{com0}) \quad (4)$$

- X_{com} は合成モデルの状態の集合であり以下のように表される。

$$X_{com} = \{MC, BS, RS\} \cup \{(x_M, x_U) \mid x_M \in X_M, x_U \in X_U, I_M(x_M) \in I_U(x_U)\}$$

ここで MC は Mode Confusion を表す状態、 BS は Blocking State を表す状態、 RS は Refusal State を表す状態である。

- $\Sigma_{com} = \Sigma_M^o \cup \{emp, \epsilon\}$ である。ここで、 ϵ は Σ_M^u の遷移に対応する事象である。
- $\Delta_{com} \subseteq X_{com} \times \Sigma_{com} \times X_{com}$ は状態遷移関係であり、マシンモデル上で $x_M \in X_M$ 、マニュアルモデル上で $x_U \in X_U$ にいるとして、 $\alpha \in \Sigma_M \cup \{emp\}$ に対して以下の 4 通りに場合分けを行って求められる。ここで、 $(x_M, \alpha, x'_M) \in \Delta_M$ 、 $(x_U, \alpha, x'_U) \in \Delta_U$ である。

- $(\delta_M(x_M, \alpha)!) \wedge (\delta_U(x_U, \alpha)!)$ の場合

$$\begin{aligned} ((x_M, x_U), \alpha, (x'_M, x'_U)) &\in \Delta_{com} \\ &\text{if } I_M(x'_M) \in I_U(x'_U) \\ ((x_M, x_U), \alpha, MC) &\in \Delta_{com} \\ &\text{otherwise} \end{aligned}$$

- $(\neg \delta_M(x_M, \alpha)!) \wedge (\delta_U(x_U, \alpha)!)$ の場合

$$((x_M, x_U), \alpha, RS) \in \Delta_{com} \quad \text{if } \alpha \neq emp$$

- $(\delta_M(x_M, \alpha)!) \wedge (\neg \delta_U(x_U, emp)!) の場合、 $\alpha \in \Sigma_M^o$ ならば遷移は定義されない。それ以外の場合、$

$$\begin{aligned} ((x_M, x_U), \epsilon, (x'_M, x'_U)) &\in \Delta_{com} \\ &\text{if } I_M(x'_M) \in I_U(x_U) \\ ((x_M, x_U), \alpha, BS) &\in \Delta_{com} \\ &\text{otherwise} \end{aligned}$$

- $(\delta_M(x_M, \alpha)!) \wedge (\delta_U(x_U, emp)!) の場合$

$$\begin{aligned} ((x_M, x_U), emp, (x'_M, x'_U)) &\in \Delta_{com} \\ &\text{if } I_M(x'_M) \in I_U(x'_U) \\ ((x_M, x_U), emp, MC) &\in \Delta_{com} \\ &\text{otherwise} \end{aligned}$$

- $X_{com0} \subseteq X_{com}$ は合成モデルの初期状態であり、 $X_{com0} = \{(x_M, x_{U0}) \mid x_M \in X_M, I_M(x_M) \in I_U(x_{U0})\}$ である。

論理和や Until によって合成モデル中に非決定な遷移が生じることがある。このような非決定な遷移は f_1 と f_2 を $f_1 \neq f_2$ である論理式としたとき、マニュアルに $((e_U, \mathbf{x}_{man}) \wedge \mathbf{X}f_1) \vee ((e_U, \mathbf{x}'_{man}) \wedge \mathbf{X}f_2)$ であるような部分論理式が含まれている場合に発生する。このとき、マシンモデルは決定的でありかつ論理和の条件より $\mathbf{x}_{man} \cap \mathbf{x}'_{man} = \emptyset$ であるため、少なくともどちらかでも MC への遷移が定義される。ここで、非決定な遷移のうちどちらかの遷移によって MC 以外の状態に遷移できるならば、ユーザは操作後のインタフェース表示を見ることによってそれに対応するマニュアル上の状態を一意に決定することができるため、オートメーションサプライズとはならない。そのため、合成モデルの各状態で、遷移元の状態から MC へ遷移するコマンドと同一のコマンドによって MC 以外の状態へ遷移できるとき、 MC への遷移を消去する。この操作によって得られたトランジションシステムを新たに G'_{com} とおく。

ところで、マシンモデルとマニュアルによっては G'_{com} 中に X_{UF} の要素を含まない強連結成分が生じることがある。合成モデル中にそのような強連結成分が生じている場合、Mode Confusion, Blocking State, Refusal State というオートメーションサプライズが起きることがない状況でも、ユーザが自分の意図する状態にいつ

まで経っても到達できないことがある。このことは、マニュアルに従って操作できないことに対応し、操作者は不安となるので、避けるべき状況である。このような状況を以下、ライブロックとよぶことにする。

5.3 オートメーションサプライズの検出

オートメーションサプライズの検出のためには、合成モデル G'_{com} に対して MC , BS , RS のいずれかの状態に可到達かどうかを調べればよい。すなわち、 G'_{com} の accessible な部分オートマトンを $Ac(G'_{com})$ とおくと、 $Ac(G'_{com})$ の状態に MC , BS , RS のいずれかが含まれていれば、オートメーションサプライズが発生する。また、ライブロックの検出には $Ac(G'_{com})$ 中の強連結成分のうち、極小のものにマニュアルモデル上の X_{UF} の要素が存在するかどうかを調べればよい。

例題 1

マシンモデルが Fig.4 で与えられているとする。ここで、 $\Sigma_M^o = \{b, r, a\}$, $\Sigma_M^u = \{d, g\}$ である。また、マシンモデルとインタフェースの対応が Fig.5 で与えられているとする。ここで、図の上段が X_M 、下段が X_I である。

- (1) インタフェース表示が HIGH1 であるとき、コマンド r を入力すると、インタフェース表示が MEDIUM となる
- (2) つぎに、コマンド r を入力するとインタフェース表示が LOW2 となる
- (3) さらにコマンド a を入力するとインタフェース表示が HIGH1 となる

というマニュアルの LTL 表現は以下のようになる。

$$\mathbf{x}_{man}^{init} = \{\text{HIGH1}\} \quad (5)$$

$$l_{man} = (r, \{\text{MEDIUM}\}) \wedge \mathbf{X}(r, \{\text{LOW2}\}) \wedge \mathbf{XX}(a, \{\text{HIGH1}\}) \quad (6)$$

このときのマニュアルモデルは Fig.6 である。

マニュアルモデルとマシンモデルを合成した合成モデルは Fig.7 となり、オートメーションサプライズを表す状態に可到達であることがわかる。ここで破線は ϵ による遷移を表している。

この合成モデルから

- (1) HIGH1 と対応している H1 でコマンド r を入力し、LOW2 と対応している L3 に遷移する
- という操作で Mode Confusion が、
- (1) HIGH1 と対応している H2 でコマンド r を入力し、MEDIUM と対応している M2 に遷移する
 - (2) MEDIUM と対応している M2 から MEDIUM と対応している M1 に内部事象 g により遷移する
 - (3) MEDIUM と対応している M1 から LOW2 と対応している L2 にコマンド r により遷移する

と操作し、その後内部事象 g が起こったとき、Blocking State が、コマンド a を入力しようとしたとき Refusal State が起こることがわかる。

Fig. 4 Machine model

X_M	L1	L2	L3	M1	M2	M3
X_I	LOW1	LOW2	MEDIUM			
X_M	H1	H2	H3			
X_I	HIGH1		HIGH2			

Fig. 5 Interface

Fig. 6 Manual model of example 1

Fig. 7 Composite model of example 1

例題 2

マシンモデル、マシンモデルとインタフェースの対応が例題 1 と同じである場合、

- (1) インタフェース表示が HIGH2 であるとき、コマンド r を入力すると、インタフェース表示が MEDIUM となる
 - (2) その後、しばらく待つとインタフェース表示が LOW2 となる
- というマニュアルの LTL 表現は以下のようになる。

$$\mathbf{x}_{man}^{init} = \{\text{HIGH2}\} \quad (7)$$

$$l_{man} = (r, \{\text{MEDIUM}\}) \wedge (\mathbf{X}((emp, \{\text{MEDIUM}\}) \mathbf{U}(emp, \{\text{LOW2}\}))) \quad (8)$$

このときのマニュアルモデルはFig.8であり、合成モデルはFig.9である。この合成モデルでは*emp*による強連結成分が生じライブロックになっていることがわかる。

Fig. 8 Manual model of example 2

Fig. 9 Composite model of example 2

6. おわりに

本論文では、マシンモデルを離散事象システムとし、LTLによって表現されたマニュアルと等価なトランジションシステムによってマニュアルモデルを表現した。そして、マシンモデル、マニュアルモデルの両者をユーザインタフェースを介して合成することにより、オートメーションサプライズが検出できることを示した。さらに、例題を用いて提案手法が有効であることを示した。LTLを用いてマニュアルを記述することにより、時間的なユーザの振舞いに対する記述が可能になるため、ライブロックというマニュアル通りの操作が途中でできなくなる状況の検出もできるようになった。

今後の課題としては、LTLの記述力に関する考察、さらに提案モデルのCTLやCTL*[14]への拡張等が挙げられる。さらに、マニュアルをLTLによって記述した場合、冗長な表現が含まれる可能性があるため、与えられたマニュアルが適切な操作をするために必要かつ十分なものであるかどうかを検証する方法についても検討したい。また実用化に向けて、内部事象を含んだ形でマシンモデルを定式化した場合の状態数や、その検証コストに関する議論も今後必要である。

謝 辞

本研究の一部は大川情報通信基金からの支援により行われた。ここに謝意を表す。

参 考 文 献

- [1] N.Sarter, D.Woods and C.Billings: Automation surprises; *Handbook of Human Factors and Ergonomics* (G.Salvendy), Wiley-Interscience, pp. 1926–1943 (1997)
- [2] E. Palmer: Oops, it didn't arm. -a case study of two

- automation surprise; *Proceeding of the 8th International Symposium on Aviation Psychology*, pp. 227–232 (1995)
- [3] A. Degani: Modeling Human-Machine Systems: On Modes, Error, and Patterns of Interaction; *Ph. D. Thesis, Georgia Institute of Technology* (1996)
- [4] A. Degani and M. Heymann: Pilot-autopilot interaction: A formal perspective; *International Conference on Human-Computer Interaction: HCL-Aero 2000*, pp. 157–168 (2000)
- [5] J. Brederke and A. Lankenau: A rigorous view of mode confusion; *Proceedings of SafeComp 2002* (2002)
- [6] A. Degani and M. Heymann: Formal verification of human automation interaction; *Human Factors*, Vol. 44, No. 1, pp. 28–43 (2002)
- [7] Y. Ukawa, T. Ushio, M. Adachi and S. Takai: Formal detection of three autonomous surprises in human-machine interaction; *IEICE Trans. Fundamentals*, Vol. E87-A, No. 11, pp. 2878–2884 (2004)
- [8] 宇川, 潮, 足立: 離散事象システムにおけるオートメーションサプライズの検証支援ツールの開発; 第49回システム制御情報学会研究発表講演会講演論文集, pp. 337–338 (2005)
- [9] D. L. Dill: The Murphi verification system; *Proceedings of the Eighth International Conference on Computer Aided Verification CAV* (R. Alur and T. A. Henzinger), Springer Verlag, pp. 390–393 (1996)
- [10] J. Rushby: Using model checking to help discover mode confusions and other automation surprises; *Reliability Engineering and System Safety*, Vol. 75, pp. 167–177 (2002)
- [11] M. Adachi, T. Ushio and Y. Ukawa: Design of user-interface without automation surprises for discrete event systems; *Control Engineering Practice* (2006) (to appear in)
- [12] S. Jiang and R. Kumar: Failure diagnosis of discrete-event systems with linear-time temporal logic specifications; *IEEE Transactions on Automatic Control*, Vol. 49, No. 6, pp. 934–945 (2004)
- [13] E. A. Emerson: Temporal and modal logic; *Handbook of Theoretical Computer Science* (J. van Leuwen ed.), Elsevier Science Publishers B. V. (1990)
- [14] E. M. Clarke, O. Grumberg and D. Peled: *Model Checking*, Chapter 9, MIT Press, pp. 121–139 (2000)