11. ケーススタディ2：入退室管理システム

11.1. 検証の概要

11.1.1. 検証対象システムの概要

本ケーススタディで取り扱うボックス管理システムは、各利用者にボックス（図中 box）が割り当てられており、IC カード認証を利用して、いくつかの部屋を経由してボックスに至るアクセスの安全性を確保するシステムである。利用者（閲覧者）は必要に応じて、自身に割り当てられているボックスにアクセスして、物品を出し入れすることができる。ボックス管理システムの全体像を 図 11-1 に示す。部屋は、待合室、閲覧室 A、閲覧室 B、ボックス格納室、および外部のスペースに分かれており、利用者はそれらの部屋を移動する。

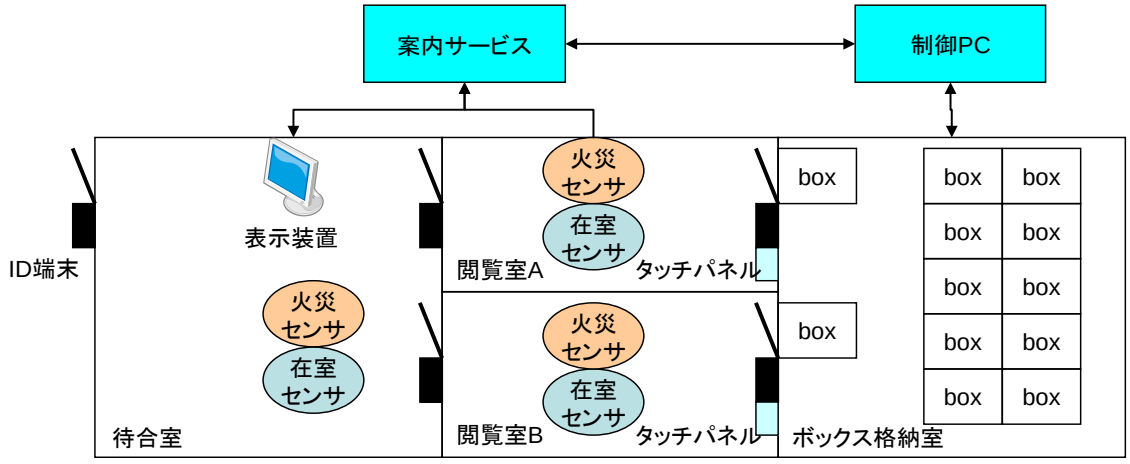


図 11-1：ボックス管理システム全体像

11.1.2. 検証の目的と検証内容の概要

閲覧者は人間であるため、システム設計時には想定しない動作を行う可能性がある。このため、システム設計時に想定しないエラーはこの部分に含まれる可能性が高い。そこで、このケーススタディでは、設計時に想定しない人間の動作を含めてシステムの検証を行い、仕様上のバグを発見することを目的とする。

モデリングでは、対象システムをなるべく忠実かつシンプルに捉えるとともに、そのモデルに合わせて網羅的な閲覧者（人間）のモデルを構築する。構築したモデルを基に、要求仕様を基に検証を行う。主な検証内容としては、以下である。

- 「閲覧室に案内されていない人が、タッチパネル脇 ID 端末で認証を行っても、常に認証をパスしない
- 閲覧室 X が空室のとき以外、常に案内は変更されない
- 待機者リストは常にオーバーフローすることはない

検証結果は、3 番目の「待機者リストは常にオーバーフローすることはない」を除き、反例は見つからなかった。

11.2. 検証対象システムの仕様

11.2.1. 検証対象システムの全体像

11.2.1.1. 検証対象システムの構成要素

検証対象システムでは、ボックス格納室、閲覧室 A および B、待合室と、案内サービスプログラム、制御 PC、PC インタフェースで構成されている。各構成要素の概要を表 11-1 に示す。

表 11-1：検証対象システムの構成要素

構成要素	概要

構成要素	概要
待合室	システムを利用する閲覧者が一番初めに入る部屋。 入室の際には、扉の脇に設置されている ID カード端末（以下、待合室扉脇 ID 端末）に、閲覧者は自身の ID カードがかざして入室する。 閲覧室の使用状況に応じて、個別の閲覧者に対して閲覧室 A または閲覧室 B への案内情報を表示する表示装置が設置されている。 また、在室センサ、火災センサが取り付けられており、人の存在の有無、火災の感知ができるようになっている。
閲覧室 A/B	各閲覧者が自身のボックスにアクセスするための部屋。 物理的なスペースの制限により、閲覧者は 1 名だけしか入れないようになっている。 閲覧室 A/B への入室の際には、各閲覧室の扉脇に設置されている ID カード端末（以下それぞれ、閲覧室 A 扉脇 ID 端末、閲覧室 B 扉脇 ID 端末）に、閲覧者は自身の ID カードがかざして入室する。 暗証番号を入力するためのタッチパネルが設置されており、その脇に正当な閲覧者であることを認証するための ID 端末（以下、タッチパネル脇 ID 端末）が設置されている。 待合室と同様に、在室センサ、火災センサが設置されている。
ボックス格納室	複数のボックスを格納し、案内サービスプログラムからの要求に応じて、該当するボックスを要求のあった閲覧室に移動したり、戻したりする。
案内サービスプログラム	新規の閲覧者の登録、待合室や閲覧室 A／B の状態管理等、システム全体の管理を行うソフトウェア・サービス。
制御 PC	ボックス格納室内のボックスの移動等を制御する機器。
PC インタフェース	ID カード端末、在室センサ、火災センサの管理を行い、カード認証とセンサ反応を案内サービスプログラムに通知するソフトウェア。

11.2.1.2. 閲覧者の基本的な動作と対象システムの動作

閲覧者は、ボックスにアクセスするために、待合室に入室し、自身が閲覧室 A または閲覧室 B のいずれかに案内されるまで待機する。待合室の入室の際には、待合室扉脇 ID 端末にて、あらかじめ配布されている ID カードをかざして、認証を行う。認証をパスすると待合室の扉のロックが解除され、待合室への入室が可能となる。閲覧室 A または閲覧室 B への案内は、待合室内に設置されている表示装置（ディスプレイ）にて表示され、閲覧者に通知される。

閲覧室 A/B のいずれかに案内された閲覧者は、該当する閲覧室に移動し、各閲覧室の扉脇に設置されている ID 端末に、ID カードをかざして認証を行う。認証をパスすると、扉のロックが解除され、閲覧室への入室が可能となる。

閲覧室に入室した閲覧者は、閲覧室内のタッチパネルの操作を行い、自身のボックスを呼び出す。ボックスを呼び出す際には、タッチパネル脇に設置された ID 端末に持っている ID カードをかざして認証を行う。認証をパスすると、ボックスの取出し口のロックが解除され、ボックスを開くことができる。

ボックスにアクセスした閲覧者は、閲覧室の扉を開いて待合室に移動し、待合室の扉から外に出て行く。閲覧室および待合室の扉は、内側からは ID 端末を用いての認証は行わない。

11.2.2. システムに求められる条件（要求仕様）

システム開発者とその発注者（ボックス管理システム所有者）が、システム開発当初に要求分析を行い、要求仕様として取り纏めた結果を表 11-2 に示す。ここで示す要求仕様は、一般の要求仕様のうち形式検証を想定したもののみ抽出したものである。この表 11-2 に示された要求仕様を、モデリングした検証対象システムに応じて修正し、検証要求とする。

表 11-2:ボックス管理システムの要求仕様

	要求仕様
「すること」	ID カードを登録している人は、待合室に入ることができる
	待合室に入った人は、必ず案内される
	閲覧室の扉認証は、閲覧室内が空室なら誰でも成功する
「しないこと」	閲覧室に人が案内されると、その人以外は、一定時間、閲覧室のボックスの認証に成功しない。 （最終的に保証すべき安全性）
	閲覧室に人が入室すると、その人が退室するまで、閲覧室のドア認証に成功しない。 （副次的な安全性。閲覧室の安全性）
	閲覧室が空室の時以外、閲覧室に人を案内しない。
	利用者は、どの部屋にも閉じ込められない。（常に、退出することが出来る。）

11.2.3. 検証対象システムの設計仕様

システム開発がボックス管理システムの要求仕様を基に作成した設計仕様について以下で説明を行う。ここで示す設計仕様から、システムのモデリングを行う。

11.2.3.1. 案内サービスプログラム

案内サービスプログラムは、PC インタフェースを通して、各部屋の ID 端末、室内センサ等から送信されてくるメッセージを基に、待合室および閲覧室A、Bの状態を管理すると共に、必要に応じて制御 PC、データベース等に動作の指示を行う。案内サービスプログラムは、待合室に入室した閲覧者のリスト（待機者リスト）を持ち、閲覧者の管理を行う。

以下では、案内サービスプログラムを実現する構成要素についてまとめる。

11.2.3.1.1. 待機者リスト

案内サービスプログラムは、システムが把握できる個々の閲覧者の位置を待機者リストにより管理する。

待機者リストは、待合室に入室した閲覧者ごとに、カード ID 番号、ボックス格納室 ID、案内閲覧室 ID、案内時刻で構成され（表 11-3）、リスト形式で管理する。

表 11-3:待機者リストのデータ型

項目	内容	備考	初期化時の値
カード ID 番号	待合室に入室した閲覧者が保有する ID カード番号。	閲覧者待合室入室時（待合室扉脇 ID 端末で認証 OK となったとき）に入力され、キーとして利用する。	待合室に入室した閲覧者の ID 番
ボックス格納室 ID	カード ID 番号に対応するボックス格納室の ID 番号。 1 つ以上のボックス格納室を想定。	1 つ以上のボックス格納室が存在する場合を想定。 閲覧者待合室入室時（待合室扉脇 ID 端末で認証 OK となったとき）に、データベースに問い合わせを行い、カード ID 番号に対応したボックスが格納されているボックス格納室 ID を記録する。	待合室に入室した閲覧者の ID 番号に対応するボックスが格納されたボックス格納室の ID
案内閲覧室 ID	ID 番号の閲覧者を案内した閲覧室の ID	閲覧室は 1 つ以上を想定。 空室となった閲覧室からの案内要求メッセージに応じて、当該 ID 番	NULL

		号の閲覧者をその閲覧室に案内する場合、閲覧室の ID 番号が記録される	
案内時刻	ID 番号の閲覧者を閲覧室 A または B に案内した時刻。	閲覧者が待合室入室時には、NULL が代入される。	NULL

11.2.3.1.2. 待合室管理プロセス

待合室管理プロセスは、PC インタフェースを介して受信するメッセージに応じて、待合室の状態を管理する。PC インタフェースを介して受信するメッセージは、待合室扉脇 ID 端末、待合室に設置されたセンサからのメッセージである。

待合室扉脇 ID 端末、待合室在室センサからのメッセージを PC インタフェースを通して取得し、待合室の状態を管理する。また、待合室扉脇 ID 端末で認証 OK となった閲覧者について、新規の閲覧者として待機者リストに登録を行う。

表 11-4:待合室管理プロセスの受信メッセージと対応するアクション

メッセージ	内容	アクション
待合室扉脇 ID 端末認証 OK	待合室扉脇 ID 端末において認証をパスしたことを示す	新規閲覧者として初期化し、待機者リストに登録する。
待合室在室センサ ON	待合室に(正当な閲覧者とは限らない)人間が存在することを示す。	ー
待合室在室センサ OFF	待合室に誰もいないことを示す。	ー
案内タイムオーバー	待機者リストに登録されているが、制限時間内に案内されなかった閲覧者が存在することを示す	待機者リストの該当項目(制限時間オーバーの閲覧者のデータ)を削除する。

11.2.3.1.3. 閲覧室A/B管理プロセス

閲覧室 X 管理プロセス(ただし、X=A or B を表す。)は、PC インタフェースと制御 PC を介して受信するメッセージに応じて、閲覧室 X の状態を管理する。PC インタフェースを介して受信するメッセージは、閲覧室 X 扉脇 ID 端末、閲覧室 X 扉、閲覧室 X 内に設置されたセンサ、閲覧室 X 内タッチパネル、閲覧室 X 内タッチパネル脇 ID 端末から送信される。制御 PC を介して受信するメッセージは、ボックス取出口の状態である。

表 11-5:閲覧室 X(X=A or B)管理プロセスが受信するメッセージと対応するアクション

メッセージ	内容	アクション
閲覧室 X 扉 ID 端末認証 OK	閲覧室 X 扉脇に附属した ID 端末に入力された ID カード番号を持って、ID 端末が認証 OK としたことを示す。	・入力されたカード ID 番号を基に、閲覧室 X の入室者リストを更新する ・閲覧者が一名以上在室する場合は「緊急事態」として、閲覧室 X の入室者リストに、新たな閲覧者を追加する。
閲覧室 X 扉閉	閲覧室 X の扉が閉まったことを示す。	・閲覧室 X の入室者リストの最後尾のデータに、メッセージを受信した時刻を記入する。 ・ボックスが格納済みであれば、閲覧室 X に閲覧者は存在しないとして、閲覧室 X の退出処理を行う。

メッセージ	内容	アクション
閲覧室 X 扉施錠	閲覧室 X の扉に鍵が施錠されたことを示す。	・緊急事態を除いて他の閲覧者が入室できないように、閲覧室 X の扉を施錠する
閲覧室 X 扉内側解錠	閲覧室 X 内の閲覧室 X の扉が解錠され、その要因が内側から開いたことを示す。	・扉の開閉が入室によるものか、退室によるものかを判断するために用いる
閲覧室 X 在室センサ OFF	閲覧室 X 室内の在室センサが OFF であり、閲覧者が 1 人もいないことを示す。	・ボックスが格納済みであれば、閲覧室 X に閲覧者は存在しないとして、閲覧室 X の退出処理を行う。 ・閲覧室 X が空室の場合、待機者リストから待合室に待機中の閲覧者を走査し、未案内の閲覧者の案内閲覧室 ID を閲覧室 X の ID にセットし、データベース（DB）に待機者リストのコピーを行う。
閲覧室 X 在室センサ ON	閲覧室 X 室内の在室センサが ON であり、閲覧者が 1 人以上いることを示す。	ー
閲覧室 X 周期監視：在室センサ OFF	在室センサの状態を周期的に観測し、その際在室センサが OFF であったことを示す。	・ボックスが格納済みであれば、閲覧室 X に閲覧者は存在しないとして、閲覧室 X の退出処理を行う。 ・閲覧室 X が空室の場合、待機者リストから待合室に待機中の閲覧者を走査し、未案内の閲覧者の案内閲覧室 ID を閲覧室 X の ID にセットし、データベース（DB）に待機者リストのコピーを行う。
閲覧室 X 周期監視：在室センサ ON	在室センサの状態を周期的に観測し、その際在室センサが ON であったことを示す。	ー
閲覧室 X 内タッチパネル脇 ID 端末認証 OK	閲覧室内のタッチパネル脇に設置された ID 端末に入力された ID カード番号が認証をパスしたことを示す	・ID 端末から入力された ID カード番号と、待機者リストに記録されている閲覧室 X に案内した ID カード番号を照合する（閲覧室内認証処理）
ボックス格納受信	ボックスが格納されたことを示す。	・閲覧者の所定の処理が終了したものとして、閲覧室 X の入室者リストから当該閲覧者のデータを削除すると共に、待機者リストを更新する（閲覧室退室処理）。
在室時間オーバー	閲覧者が定められた時間以上に閲覧室内にいることを示す。	・閲覧室 X の在室時間超過フラグを ON にする
閲覧室 X タッチパネル暗証番号照会	タッチパネルに暗証番号が入力されたことを示す。	・制御 PC に閲覧室内認証の結果を通知する。 ※制御 PC は、認証 OK を受信した場合、ボックスのロックを解除

11.2.3.1.4. PCインタフェース

PC インタフェースは、ID 端末、タッチパネル、在室センサ、火災センサなどから送信されてくる信号を案内サービスプログラムで解釈可能なメッセージに変換して、案内サービスプログラムに送信する。

以下では、PCインタフェースを構成する要素についてまとめる。

11.2.3.1.5. 待合室扉脇ID端末

待合室の入り口に附属する扉の脇に設置された ID 端末である。正規の閲覧者は、所持している ID カードを本 ID 端末にかざして認証を行い、室内に入る。

待合室扉脇 ID 端末内部には、正規の閲覧者の ID カード番号を記録しており、閲覧者が所持する ID カードから入力された ID 番号と照合を行う。ID 端末内部に記録されている ID カード番号の中の一つにマッチすると、認証成功とし、認証 OK の信号を PC インタフェースに送信すると共に、待合室の扉のロックを解除する。ID 端末内部に記録されている ID カード番号のいずれにもマッチしない場合、認証失敗とし、待合室の扉のロックは解除されない。なお認証失敗の場合には PC インタフェースを介して案内サービスプログラムに認証 NG のメッセージが送信されるが、案内サービスプログラム内部ではこのメッセージを利用しない。

11.2.3.1.6. 閲覧室X扉脇ID端末

閲覧室 A および閲覧室 B のそれぞれの入り口に附属する扉の脇に設置された ID 端末である。正規の閲覧者は、所持している ID カードを本 ID 端末にかざして認証を行い、室内に入る。

閲覧室 X 扉脇 ID 端末内部には、正規の閲覧者の ID カード番号が記録されており、閲覧者が所持する ID カードから入力された ID 番号と照合を行うと共に、閲覧室 X 内に誰もいない場合、閲覧室 X の扉が開錠され、入室が可能となる。閲覧室 X 内に誰かがいる場合には、ID カード番号の照合をパスしても、閲覧室 X の扉は開錠されず、閲覧者は入室できない（表 11-6）。

表 11-6:閲覧室 X の扉の解錠条件

		閲覧室 X 内の在室者の有無	
		無し	有り
照合	アンマッチ	解錠しない	解錠しない
	マッチ	解錠する	解錠しない

11.2.3.1.7. 閲覧室X扉

閲覧室 A および閲覧室 B の扉である。電子錠を使用しており、錠の解錠、施錠状態と扉の開閉状態の変化を通知する。通知メッセージには、発生要因が含まれている。解錠であれば、閲覧室 X 扉脇 ID 端末の認証成功が要因なのか、部屋の内部からカード認証を行わずに直接開けたことが要因なのかが区別できる。

表 11-6 の制御は閲覧室 X 扉に、施錠モードを設定することで実現する。閲覧室 X 内の在室者が無しの場合は、通常モードで動作させ、カード認証による解錠を許可する。閲覧室 X 内の在室者が有りの場合は、解錠禁止モードで動作させ、カード認証による解錠を禁止する。

11.2.3.1.8. 閲覧室X内タッチパネル及びID端末

各閲覧室 A および B 内部には、ボックスの開閉を行うための認証装置として、ID 端末とタッチパネルが設置されている。閲覧者は、閲覧室内の ID 端末に ID カードをかざすと共に、タッチパネルにて暗証番号を入力し、認証をパスするとボックスの開閉が可能となる。

閲覧室 X 内部の ID 端末から入力された ID カード番号は、案内サービスプログラムの閲覧室 X 管理プロセスにおいて、正規に案内された閲覧者の ID カード番号との照合を行う。照合結果が OK である場合には、制御 PC がボックスの制御を行い、閲覧室 X の取り出し口までボックスが移送される。照合結果が NG の場合には、ボックスの移送は行われず、認証失敗が閲覧者に伝えられ

る。

移送されたボックスの取り出しに際しては、閲覧室内のタッチパネルからの暗証番号の入力を行い、タッチパネル内部で暗証番号の照合を行い、OK であればロックが解除され、閲覧者はボックスへのアクセスが可能となる。NG であれば、ロックは解除されず、閲覧者はボックスへのアクセスができない。

11.2.3.2. センサ

待合室、閲覧室 A および閲覧室 B には、在室センサおよび火災センサが取り付けられている。

以下、在室センサおよび火災センサについて説明する。

11.2.3.2.1. 在室センサ

在室センサは、人の存在を検知し、センサ内部にその状態を記録する。PC インタフェースは、定期的に各部屋の在室センサを監視し、閲覧者の存在の有無を確認する。

在室センサは、人間が存在することを感知すると、内部状態を OFF から ON にスイッチする。人間の存在の有無だけを感知し、人数の把握はできない。また、部屋に人間が 1 人もいなくなったとき、在室センサは内部状態を ON から OFF にスイッチする。

11.2.3.2.2. 火災センサ

火災センサは、火災を検知し、センサ内部にその状態を記録する。PC インタフェースは定期的に各部屋の火災センサを監視し、火災の有無を確認する。

火災センサは、火災の有無を感知すると、内部状態を OFF から ON にスイッチする。

11.2.3.3. データベース(DB)

データベース（DB）は、登録されている閲覧者と閲覧者に割り当てたボックスが格納されているボックス格納室の対応関係を管理する静的なデータと、待機者リストを逐次複製し、待合室および閲覧室 X の在室状況を管理する動的なデータを保有する。

案内サービスプログラムの待合室管理プロセスは、PC インタフェースを通して、待合室扉脇 ID 端末から認証 OK を受信した際に、新規の閲覧者として待機者リストに登録する。この際、当該閲覧者に対応するボックスが格納されているボックス格納室の ID の問い合わせを DB に行う。DB は待合室管理プロセスの問い合わせに応じて、当該閲覧者のボックスが保管されているボックス格納室の ID を待合室管理プロセスに返す。なお、今回のケーススタディでは、ボックス格納室が 1 つであるため、どの閲覧者に対しても同じボックス格納室 ID がデータベース（DB）から待合室管理プロセスに返される。

案内サービスプログラムの閲覧室 X（X=A or B）管理プロセスは、PC インタフェースを通して、閲覧室 X 内の在室センサが OFF であることを感知すると、待機者リストにおいて未案内の閲覧者を走査し、当該閲覧者の案内閲覧室 ID を閲覧室 X にセットして、DB に待機者リストのコピーを行う。

11.2.3.3.1. 表示装置（Display）

表示装置（Display）は、一定周期ごとにデータベース（DB）が待機者リストのコピーから最新の案内者を走査し、表示内容を切り替える（）。表示内容の切り替えの際には、一旦表示内容をリフレッシュして、DB 内の待機者リストのコピーを参照して、閲覧室 A／B それぞれへの最新の案内者を探索して表示する。

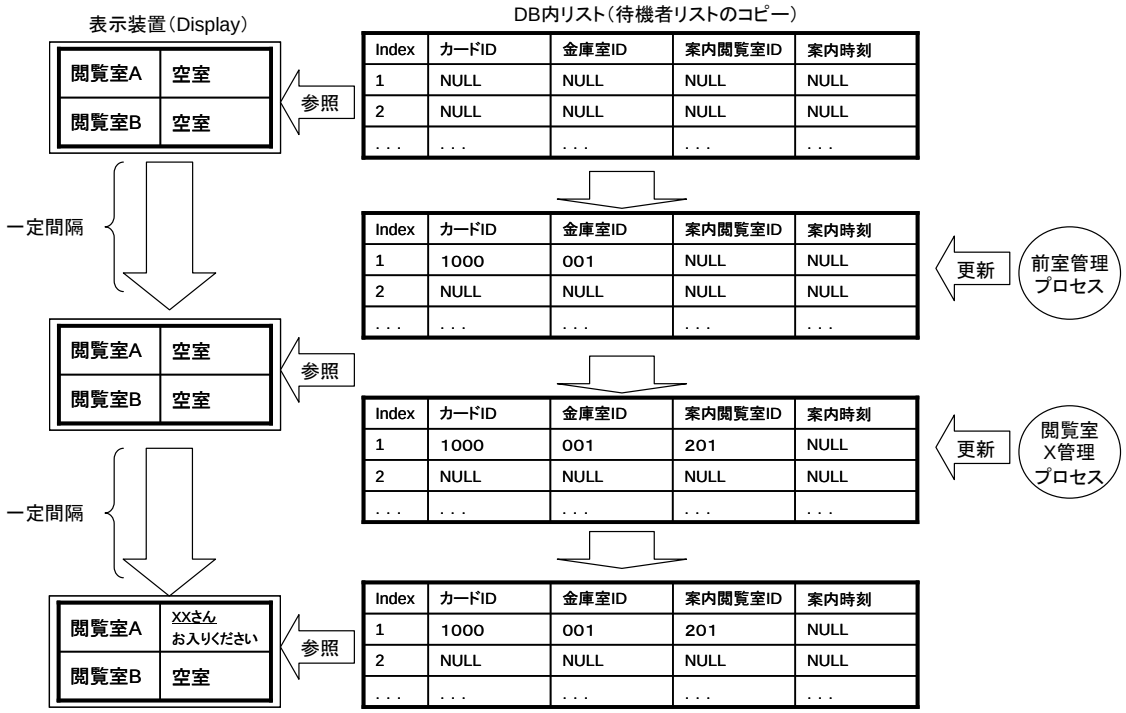


図 11-2:表示装置の更新タイミング

11.2.3.4. 閲覧者

形式検証を行うためには、対象システムだけでなく、システムの周りの環境もモデリングし、全体として閉じた系を定義しなければならない。そこで、システムにとって外部の環境に相当する閲覧者の動きを説明する。閲覧者は人間であるため想定外の動きすることが考えられる。ここではシステムとして想定する閲覧者の動作を述べる。

正当な閲覧者は、システム管理者またはサービス提供者から付与または貸与された ID カードを保有している。ID カードには、当該閲覧者に対応する ID 番号が記載されており、この ID 番号によりシステム上は一意に特定される。

閲覧者は、待合室の入り口に設置されている待合室扉脇 ID 端末に所持している ID カードをかざし、本端末で認証をパスして、待合室に入室する。

待合室では、表示装置に自身の名前が表示されるまで待機する。表示装置に自身への案内(入室する閲覧室と自身の名前)が表示されたら、その閲覧室の方向に進む。

案内された閲覧者は、閲覧室 X(A or B)の扉脇に設置された閲覧室 X 扉脇 ID 端末において、保有する ID カードをかざし、本端末で認証をパスして、閲覧室 X に入室する。

閲覧室 X に入室した閲覧者は、閲覧室内 ID 端末に所持している ID カードをかざし、タッチパネルに暗証番号を入力して、自身のボックスにアクセスする。アクセスが終了した後、ボックスを格納して、閲覧室 X の扉から待合室に移動する。

ボックスを取り出した後に閲覧室から待合室に移動した閲覧者は、待合室の扉から外部に出て行く。

11.3. 検証対象システムのモデリング

11.3.1. モデル化の前提

11.3.1.1. ケーススタディの方針

今回のケーススタディでは、対象システムの外部環境は人間(閲覧者)の動きに相当する。人間の動きに関して、システム開発時に想定していなかった動作も含めたシステムの検証を行うことを

主目的とする。これは、必ずしも人間はシステムが想定する動作だけを行うわけではないため、実際のシステムでは想定外の動作も含めた形で検証を行う必要があるためである。一方で、人間の動作は無限大であることから、検証対象システムをなるべくシンプルなものとして捉え、閲覧者の動作をそのモデルに合わせて構築する。

11.3.1.2. 検証対象とするシステムの範囲

システム構築者がシステムの構築を請け負った案内サービスプログラムを中心にモデリングを行う。つまり、ボックス格納室並びにボックスの制御を行う制御 PC はモデリングの範囲外とする。

11.3.1.3. モデリングの前提条件

検証対象システムのモデリングを行う際の前提を表 11-7 に示す。

システムの要求仕様が満たされることを示すことが目的であるため、通常時のシステム動作を検証範囲とし、非常事態等は取り扱わない。つまり、火災等の非常事態や、閲覧者の怪我、病気等の事象も取り扱わないこととする(前提 0-1、0-2)。

また、上述 6.2.2 に示したシステムの取扱範囲から、ボックス格納室は取り扱い範囲外とする(前提 0-3)。関連して、閲覧室 X 内におけるボックスの取り出しに関する処理は対象外とし(前提 1-2)、付随して、タッチパネルは取り扱わない(前提 1-2-1)、および、閲覧者や閲覧室 X でボックスに関連する状態は取り扱わない(前提 1-2-2)。ただし、検証対象システムでは、正規の閲覧者が閲覧室 X 内でボックスにアクセスすることが主要なサービスであることから、前提 1-2、1-2-1、1-2-2 を考慮し、閲覧室 X 内 ID 端末にアクセスして認証 OK となることでボックスの取り出しが終了したものとみなし簡略化する(前提 1-2-3)。

SPIN では時刻に関連する取扱が困難であるため、時刻は対象外とし、タイムオーバー関連処理は対象外とする(前提 1-1)。また、モデル化の単純化のために閲覧室 X に附属する扉はとりあつかわない(前提 1-3)、全ての閲覧者は登録済み(前提 1-5)とする。最後に、前提 0-1 から、災害等の非常事態は取り扱わないため、閲覧室 X における緊急入室、案内停止は取り扱わない(前提 1-4)。

表 11−7:検証対象システムのモデリングの前提

		前提の内容	理由
大前提	0-1	災害関連の事象・システムは取り扱わない	システム構築者は正常状態の検証をニーズとして持っているため
	0-2	閲覧者の怪我、病気等の事象とそれに対応するためのシステムは取り扱わない	システム構築者は正常状態の検証をニーズとして持っているため
	0-3	ボックス格納室については取扱い範囲外とする	ボックス格納室はシステム構築者の取り使い範囲外
モデリングの際の前提	1-1	時刻は対象外とし、タイムオーバー関連処理は対象外とする	SPIN では時刻の取扱が困難なため
	1-2	ボックスの取り出し等に関する処理は対象外とする	ボックス格納室はシステム構築者の取り使い範囲外(前提 0-3)のため
	1-2-1	閲覧室内のタッチパネル関連処理は対象外とする	ボックスの取り出しに関連する処理であるため(前提 1-2)
	1-2-2	ボックスに関連する状態は取り扱わない	ボックスの取り出しに関連する処理であるため(前提 1-2)
	1-2-3	閲覧者が閲覧室内のタッチパネル脇の ID 端末で認証 OK となった時点で、ボックスの取り出しが終了したものとみなす	閲覧室における閲覧者の動作を簡略化してモデル化するため
	1-3	閲覧室の扉は取り扱わない	モデルの単純化のため
	1-4	閲覧室内における緊急入室、案内停止は	災害関連の事象・システムは取り扱わな

		前提の内容	理由
		取り扱わない	いため（前提 0-1）
	1-5	全ての閲覧者は登録済みとする	モデルの単純化のため

11.3.2. モデリングの考え方

11.3.2.1. モデリングの要件

今回のケーススタディの要件としては、上述の通り、

- 閲覧者の動作について設計時に想定しない動作も含めて網羅的に検証できるように構築することと
- 閲覧者のモデルに合わせて、原因個所が分かるようになるべくシンプルに検証対象をモデリングすること

の2点があげられる。

また、今回検証対象としたシステムでは、多くの異なるプロセスが存在する。例えば、検証対象システムだけで、大きく分けて、待合室管理プロセス、閲覧室 A 管理プロセス、閲覧室 B 管理プロセスとDBプロセスの4つのプロセスが考えられ、この他にもセンサ等プロセスとして実装可能なものが存在する。また、これらに加えて複数個の閲覧者プロセスが考えられる。従って、

- 必要最小限の閲覧者プロセスを確保しつつ、全体のプロセス数を削減し、現実的な時間内で検証が終了するようにモデリングすること

が必要である。

11.3.2.2. モデリングの要件に関する対処方針

11.3.2.2.1. 閲覧者モデリングの方針

閲覧者のモデリングは、システム設計時に想定していた閲覧者の動作（11.2.3.4参照）をベースに有り得る状態を割り出し、現実的に有り得る状態遷移を構成することで、設計時に想定していなかった閲覧者の動作もモデリングする。

11.3.2.2.2. 検証対象システムのモデリングの方針

検証対象システムは、複数のプロセスが考えられ、それぞれのプロセスが複数の状態を有し、各プロセスが関連しあいながら状態遷移を行う。また、個々の状態で待機者リストの中からある条件に該当するレコードを検索するなど、比較的複雑な処理を行う場合がある。これらを考慮し、以下の方針でモデリングを行うこととする。

- 各プロセスが有する状態をなるべく少なくする
- 待機者リスト等を検索するなどの処理は単純化し、必要最小限にとどめる

11.3.2.2.3. プロセス数の削減の方針

検証対象システムでは、案内サービスプログラム内の待合室および閲覧室の各部屋の閲覧者の有無を管理するプロセス、待機者リストをコピーし定期的に表示装置に表示される内容を更新するDBのプロセスなど、複数のプロセスが同時並行的に動作している。これらを検証対象システムの実態に即してモデル化することは、不可能であり、また、状態爆発から検証を行うことができなくなる可能性が高い。そこで、以下の方針で検証対象システムをモデル化する。

- システムの要求仕様（表 11-2）から、主な検証対象となる案内サービスプログラムの待合室、閲覧室 A および閲覧室 B の管理を行う部分を主要プロセスとしてモデリングする
- その他のプロセスは、主要プロセスまたは閲覧者プロセスの一部に含めるか、チャネルまたはグローバル変数とする

検証対象システムでは、複数の閲覧者が想定されている。理想的にはなるべく多くの閲覧者プロセスに対して検証を行うことが望ましいと考えられるが、一方で、閲覧者プロセスが多くなるほど状態爆発により検証ができなくなる可能性が高まる。そこで、検証においては、閲覧者プロセスの

数を 2 以上で実施することとし、状態爆発が起こらない範囲で閲覧者プロセスの数を増やしながら検証を行うこととする。

11.3.3. モデリングの概要

検証対象システムをモデリングする際に、実際のシステムに比較的近いと考えられる詳細なモデルを構築した後で、上記の 11.311.3.2.2 に示したモデリングの要件に関する対処方針に従ってモデルを単純化し、状態爆発の発生を抑え検証を行い易くする。

検証対象システムの単純化したモデルを構築した上で、検証者のモデリングを行う。

11.3.3.1. 実際に近いモデルの概要

実際の検証対象システムに近いモデルのイメージ全体像を図 11-3 に示す。

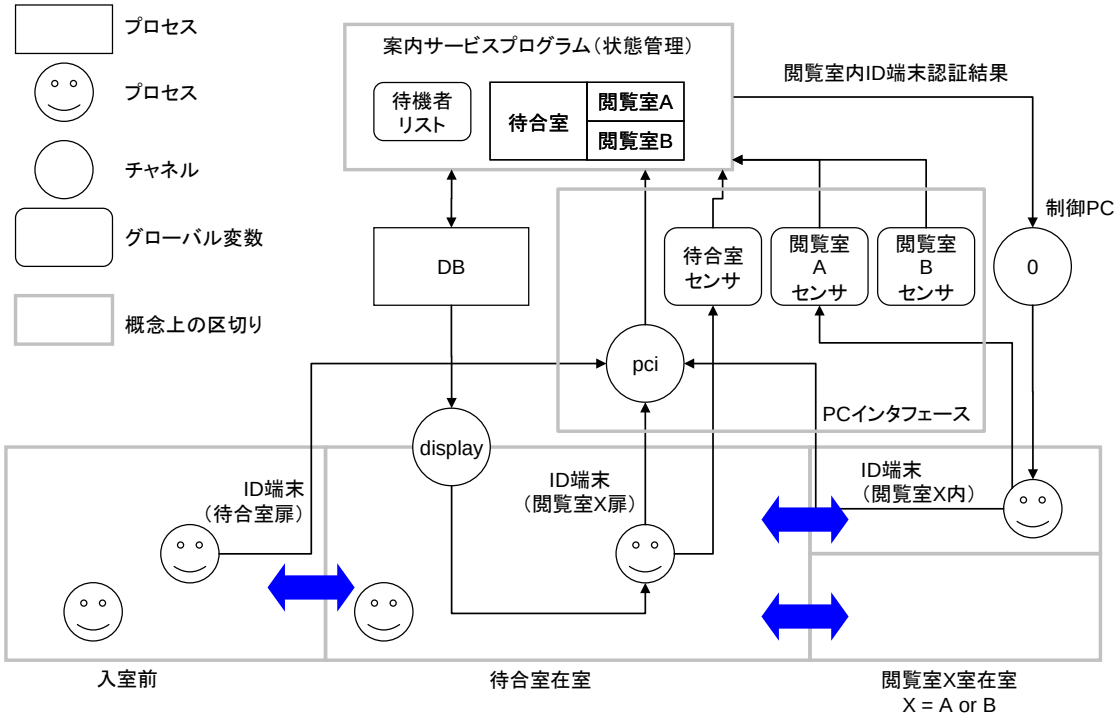


図 11-3：実際に近いモデル（全体像）

案内サービスプログラムは、待合室管理プロセス、閲覧室 X（X=A or B）管理プロセスおよび待機者リストで構成する。待機者リストは、グローバル変数として定義する。

PC インタフェースは、待合室扉脇 ID 端末や在室センサなどの信号を案内サービスプログラムで解釈可能なメッセージに変換し、案内サービスプログラムに送信する役割を担うことから、チャネル（図 11-3 pci）およびグローバル変数として表現する待合室センサ、閲覧室 A センサおよび閲覧室 B センサで構成する。なお、ここでのセンサとは在室センサである。

データベース（DB）は、プロセスとして定義する。DB プロセスは、待合室管理プロセスや閲覧室 X プロセスの要請に応じて、待機者リストのコピーや表示装置の切り替えを行う。

表示装置は、DB の要請に応じて表示内容を変更するだけであり、能動的に動作する主体ではないため、チャネルとして表現し、プロセス数の削減をはかることとした。

また動作 PC は、検証の範囲外であるが、実際は閲覧者が動作 PC の反応、つまり、ボックスの開閉を通して、閲覧室内 ID 端末の認証結果を知ることから、チャネルとして表現し、案内サービスプログラムの閲覧室 X 管理プロセスから閲覧者がメッセージを受け取るようにモデリングした。

11.3.3.2. 単純化した検証対象システムモデルの概要

単純化した検証対象システムのモデルの全体像を図 11-4 に示す。

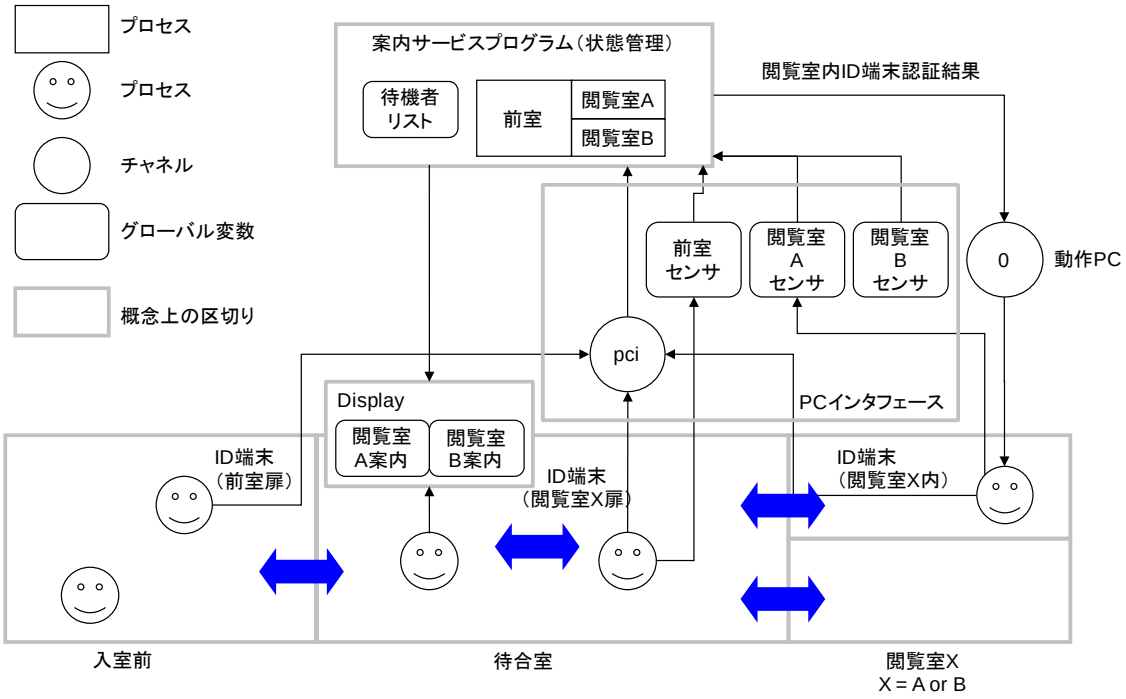


図 11-4：単純化した検証対象システムのモデル（全体像）

単純化したモデルでは、図 11-3 における DB プロセスを削減するとともに、表示装置（Display）を 2 つのグローバル変数に変更した。これは上述したプロセス数の削減を行うたと、表示装置に係る制御を単純化するためである（11.3.2.2 参照）。

ここで留意すべき点として、モデルの単純化に伴い検証結果に差異が生じないかという点である。今回のケーススタディでは、検証対象システムの動作の方が閲覧者の動作よりも十分に早いことを考慮し、案内サービスプログラムと DB のインタラクションを一体とみなし、案内サービスプログラムと DB 間のインタラクションで生じる不具合は無視することとした（11.2.3.3 参照）。

11.3.3.3. 閲覧者のモデルの概要

システム設計時に想定した、個々の閲覧者の動線を以下に示す。

閲覧者は、以下のステップで各部屋を移動する。

システム設計時に想定された閲覧者のモデル

① 閲覧者は、自分で保有する ID カードを待合室扉脇 ID 端末にかざして認証を受ける。認証をパスした場合には、待合室の扉のロックが解錠され、閲覧者は②に進む。
② 閲覧者は、待合室の扉を開けて、待合室内に移動する（③に進む）。
③ 閲覧者は、待合室内に設置された表示装置を確認する。自分の氏名（または、ID カード番号）が表示されている場合には、④に進み、表示されていない場合には、待機する（つまり、③を繰り返す）。
④ 閲覧者は、表示装置に表示されていた氏名（または、ID カード番号）とともに表示さていた閲覧室 X（A または B）に移動し、閲覧室 X 脇 ID 端末に自身の ID カードをかざして認証を受ける。閲覧室 X の扉のロックが解錠された場合、室内に移動する。閲覧室 X の扉のロックが解錠されなかった場合、③に戻る。

⑤　閲覧者は、閲覧室 X 内にあるタッチパネルに ID カード番号を入力するとともに、閲覧室 X 内 ID 端末に自身の ID カードをかざす。認証をパスした場合、閲覧室 X ない ID 端末脇にあるボックスとり出し口のロックが解錠され、ボックスへのアクセスを行う。
⑥　閲覧者は待合室に移動する。閲覧室 X でボックスにアクセスできた場合には⑦へ、閲覧室 X でボックスにアクセスできなかった場合には③へ進む。
⑦　待合室の扉を開けて、外に出る。

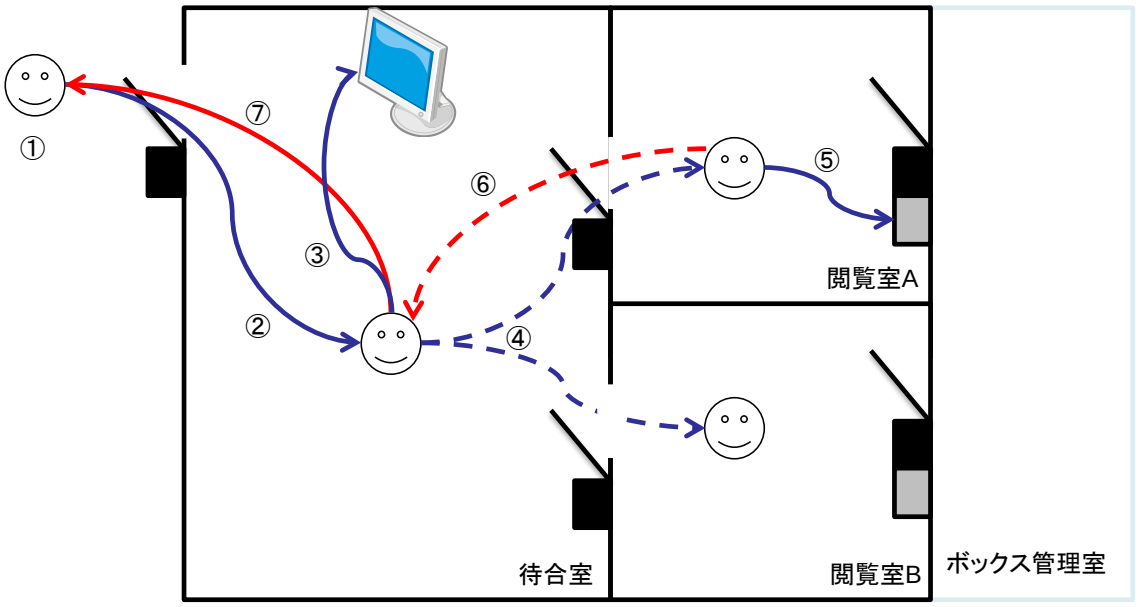


図 11-5:閲覧者の動線(システム設計時)

閲覧者のモデルには、システム設計時に想定していない動作も含めてモデリングを行う。これにより、システム設計時に想定してなかった閲覧者の動作に起因するエラーを検出する。

具体的には、上記の閲覧者の動作ステップを以下のように修正した。

システム設計時に想定していなかった動作も含めた閲覧者のモデル

①　閲覧者は、自分で保有するIDカードを待合室扉脇ID端末にかざして認証を受ける。認証をパスした場合には、待合室の扉のロックが解錠され、閲覧者は②、または、①に進む。
②　閲覧者は、待合室の扉を開けて、待合室内に移動する(③に進む)。
③　閲覧者は、待合室内に設置された表示装置を確認する。閲覧者は、表示装置の表示内容によらず、以下のいずれかに進む。
　A)　③に進む。
　B)　閲覧室Aに移動する(④に進む)。
　C)　閲覧室Bに移動する(④に進む)。
　D)　閲覧室から出る(①に進む)。
④　閲覧者は、表示装置に表示されていた氏名(または、ID カード番号)とともに表示さていた閲覧室 X(A または B)に移動し、閲覧室 X 脇 ID 端末に自身の ID カードをかざして認証を受ける。閲覧室 X の扉のロックが解錠された場合、室内に移動する。閲覧室 X の扉のロックが解錠されなかった場合、③に戻る。
⑤　閲覧者は、以下のいずれかの動作を行う。
　A)　閲覧室Xから退出する(③に進む)。
　B)　閲覧室X内タッチパネルを操作し、閲覧室X内ID端末にIDカードをかざす。認証をパスした場合、閲覧室XないID端末脇にあるボックスとり出し口のロックが解錠され、ボッ

クスへのアクセスを行い、⑥に移動する。

⑥ 閲覧者は待合室に移動する。閲覧者は以下のいずれかの動作を行う。

A) ③に進む。

B) 閲覧室Aに移動する(④に進む)。

C) 閲覧室Bに移動する(④に進む)。

D) 閲覧室から出る(⑦に進む)。

⑦ 待合室の扉を開けて、外に出る。

11.3.4. 各プロセスの詳細

11.3.4.1. 案内サービスプログラム

案内サービスプログラムは、待機者リスト、待合室管理プロセス、閲覧室 A 管理プロセスおよび閲覧室 B 管理プロセスから構成した。閲覧室 A 管理プロセスと閲覧室 B 管理プロセスは、同一の機能を有する閲覧室 A と閲覧室 B の状態を管理するプロセスであるため、Promela コード上の記述は同一とし、閲覧室 A、B それぞれに割り当てた ID 番号で識別するようにモデリングした。また、DB の一つの機能である、表示装置の更新に関して、待合室管理プロセス、閲覧室 A/B プロセスが、待機者リストを DB へ登録するタイミングで行うようにモデリングした。

11.3.4.1.1. 待機者リスト

待機者リストは、カード ID 番号、ボックス格納室 ID、案内閲覧室 ID、案内時間を 1 つのレコードとするリストである(表 11-3)。Promela のコード上では、1 つのレコードを parsonData 型として定義し、これを配列として扱うようにモデリングした(図 11-6)。なお、parsonData 型の案内時間 guide_time は Promela コード上では利用していないダミーコードである。

```
typedef parsonData  {
        int  card_id;              /*  カード ID          */
        int  box_room_id;          /**  ボックス格納室 ID        */
        int  booth_room_id;        /**  案内閲覧室 ID*/
        int  guide_time;           /**  案内時刻       **/
        };
```

図 11-6:parsonData 型の定義

11.3.4.1.2. 待合室管理プロセス

待合室管理プロセスの状態遷移表を表 11-8 に示す。

待合室は閲覧者が誰もいないとき(不在状態:ABSENCE)と、誰か 1 名以上いるとき(在室状態:PRESENCE)の 2 つの状態を定義した。この理由として、待合室に設置されている在室センサは、人の存在の有無(「誰もいない」または「1名以上人がいる」)を検知することしかできないため(11.2.3.2 章参照)、待合室管理プロセスは、待合室に閲覧者が1名以上存在するすなわち「在室」状態と、待合室に閲覧者が誰もいない「不在状態」に分けられると考えられるためである。

イベントには、チャネル pci を通して受信する「待合室扉脇 ID 端末認証 OK」と待合室のセンサ状態 ON と OFF の3つを設定した(表 11-4)。

表 11-8:待合室管理プロセスの状態遷移表

	待合室扉脇 ID 端末認証 OK id_terniminal_ok	在室センサ ON PREVIOUS_ROOM_SENSOR_STATE =SENSOR_ON	在室センサ OFF PREVIOUS_ROOM_SENSOR_STATE =SENSOR_OFF
不在	新規顧客登録	遷移:在室	

ABSENCE	ボックス格納室検索 DB 登録 遷移:在室		
在室 PRESENCE	新規顧客登録 ボックス格納室検索 DB 登録 遷移:在室		遷移:不在

待合室扉脇 ID 端末は、内部に記録されている閲覧者 ID 番号と入力された値との照合を行い、マッチするものが存在した場合にだけ PC インタフェースに認証 OK のメッセージを送信する(11.2.3.1.5 参照)。従って、認証 OK の場合のみ、待合室管理プロセスにメッセージが送信されてきてイベントが発生する。逆に、認証をパスしなかった場合、待合室管理プロセスにはメッセージは送信されない。このため、待合室管理プロセスを駆動させるイベントの一つは、「待合室扉脇 ID 端末認証 OK」のメッセージであると特定できる。「待合室扉脇 ID 端末認証 OK」のメッセージは、チャネル pci として表現される PC インタフェースに{ PREVIOUS_ROOM_DOOR_ID_TERMINAL, *card_id*, id_terminal_ok} としてバッファされるようにモデリングされるため(11.3.4.2.1 参照)、チャネル pci のバッファの中から PREVIOUS_ROOM_DOOR_ID_TERMINAL に一致するメッセージが存在するか確認した後、存在する場合はそのメッセージだけを受信する。

待合室に設置されているセンサの状態は、グローバル変数 PREVIOUS_ROOM_SENSOR_STATE として、この変数に ON(==1)、OFF(==0) がセットすることで、在室および不在のセンシングをモデリングしている(11.311.3.4 11.3.4.3 11.3.4.3.1 参照)。待機室管理プロセスは、PREVIOUS_ROOM_SENSOR_STATE を参照し、閲覧者の存在の有無を確認する。

□ 新規顧客作成処理

新規顧客作成処理は、待機者リスト waitingList[ ] に対して、待合室に新たに入室した閲覧者の parsonData 型のデータを作成して格納する。

□ ボックス格納室検索処理

ボックス格納室検索処理は、データベース(DB)プロセスに対して、新規の閲覧者のカード ID に対応するボックス格納室 ID に問い合わせを行い、戻り値を parsonData.box_room_id に代入する処理である。ただし、今回の検証対象システムではボックス格納室は 1 つであり、形式的な問い合わせとなるため、Promela コード上では、何の処理も行わないように、記述を行わなかった。

□ DB 登録処理

DB 登録処理は、DB プロセスに対して待機者リスト waiting_list[ ]を DB 内部にコピーすることを要求する処理である。

モデル上では、DB を削減しているため、DB 登録処理のタイミングで waiting_list[ ]から、次に案内すべき閲覧者(waiting_list[i]. booth_room_id == BOOTH_ROOM_A または waiting_list[i]. booth_room_id == BOOTH_ROOM_B となっているレコード)を検索し、グローバル変数でモデル化される表示装置(display_A、display_B)に代入することで、表示装置の切り替えを行う。

11.3.4.1.3. 閲覧室 X (X=A or B) 管理プロセス

閲覧室 X 管理プロセスの状態遷移表を表 11-9 に示す。

閲覧室 X も閲覧者が在室する場合(在室状態:PRESENCE)と、誰も在室していない場合(不在状態:ABSENCE)で動作が異なるため、2 つの状態を定義した。これは、特に在室センサが OFF のときに、案内者決定処理を行う部分が顕著に異なるためである。

チャネル pci を通して受信するイベントとしては、閲覧室 X 扉脇 ID 端末認証 OK (BOOTH_ROOM_DOOR_X_ID_TERMINAL, card_id, id_terminal_ok)、及び、閲覧室 X 内タッ

チパネル脇 ID 端末認証確認(INSIDE_BOOTH_ROOM_X_ID_TERMINAL, card_id, check_id_number) がある。また、グローバル変数で定義した BOOTH_ROOM_X_SENSOR_STATE を参照することで受信するイベントとして、閲覧室 X 在室センサ ON(BOOTH_ROOM_X_SENSOR_STATE_SENSOR == ON)及び閲覧室 X 在室センサ OFF (BOOTH_ROOM_X_SENSOR_STATE_SENSOR == OFF) がある。

表 11-9:閲覧室 X プロセスの状態遷移表

	閲覧室扉脇 ID 端末認証 OK id_terminal_ok	閲覧室 X 内タッチパネル脇 ID 端末認証確認	在室センサ ON PREVIOUS_ROOM_SENSOR_STATE =SENSOR_ON	在室センサ OFF PREVIOUS_ROOM_SENSOR_STATE =SENSOR_OFF
不在 ABSENCE	閲覧室扉認証処理 遷移:在室			案内者決定(待機者リスト更新) DB 登録
在室 PRESENCE		閲覧室内認証		閲覧室退出処理 遷移:不在

□ 閲覧室扉認証処理

閲覧室扉認証処理は、閲覧室 X に入室した閲覧者を管理するために、入室した閲覧者のカード ID 番号、入室時刻を記録し、閲覧室 X 入室者リストとして管理を開始する処理である。Promela のコード上では、カード ID 番号 card_id、入室時刻 time で構成される BoothRoomPasonData 型(図 11-7)を定義し、閲覧室 X 管理プロセスの内部変数として管理する形とした。なお、time は形式的に定義しているのみであり、初期化時は NULL 値を代入することとし、処理の過程では使用していない。

```
typedef BoothRoomParsonData          {
    int card_id;        /* カード ID 番号 */
    int time;           /* 入室時刻 */
};
```

図 11-7:BoothRoomParsonData 型の定義

□ 閲覧室内認証処理

閲覧室 X 内タッチパネル脇 ID 端末に入力されたカード ID 番号と、実際に案内サービスプログラムが表示装置を通して案内した閲覧室 X とカード ID 番号(つまり、待機者リストの該当するカード ID 番号と記録されている案内閲覧室 ID)が一致しているか否かを検査する。検査をパスすれば、id_terminal_ok のメッセージをチャネル pc_controller(制御 PC に相当、後述)を通して、閲覧室 X に入室した閲覧者に送信する。検査をパスしない場合、id_terminal_ng のメッセージを pc_controller を通して、閲覧室 X に入室した閲覧者に送信する。

□ 閲覧室退出処理

閲覧室 X 管理プロセスの内部変数である入室者リストに記録されている情報、つまり、閲覧室 X に入室していると考えられる閲覧者の card_id 番号を基に、待機者リスト waitingList[ ] から該当する閲覧者のレコードを削除すると共に、閲覧室 X 管理プロセスの入室者リストを初期化する。

□ 案内者決定処理

待機者リスト waitingList[ ] から未案内の閲覧者、つまり、案内閲覧室 ID (booth_room_id) に

閲覧室 ID がセットされておらず、NULL 値（Promela コード上は NL）となっているレコードを、待機者リストの先頭から探索する。該当するレコードが存在する場合には、そのレコードの booth_room_id に当該閲覧室の ID をセットする。該当するレコードが存在しない場合には、何もしない。

11.3.4.2. PCインタフェース

モデリングの上では、PC インタフェースは、各 ID 端末からのメッセージを送信するためのチャネルと、各在室センサの状態を記録するための変数によりモデリングした。

11.3.4.2.1. チャネルpci

各 ID 端末からの信号を、メッセージに変換して案内サービスプログラムに送信する PC インタフェースの機能は、単純なメッセージ変換機能であるため、Promela コード上ではモデリングではチャネル pci としてモデル化をおこなった（図 11-8）。なお、pci チャネルのサイズを 256 としたのは、待合室や閲覧室 A,B、閲覧者のプロセスからのメッセージを十分に捕捉できるようにし、チャネルのオーバーフローが起こらないようにするためである。

```
#define N_PCI            256
chan pc_interface = [N_PCI] of { int, int, mtype };
```

図 11-8：チャネル pci の定義

チャネル pci の第 1 引数は各 ID 端末に割り振った ID 番号（表 6 10）、第 2 引数は閲覧者が保有するカード ID 番号、第3引数は ID 端末から送信されるメッセージ（表 11-1）となっている。チャネル pci をこの形式としたのは、待合室管理プロセスおよび閲覧室 X プロセス（X=A or B）のそれぞれが受信するべきメッセージを自動で判別するために、端末の ID 番号を識別子として用いるためである。これは、Promela/SPIN では、チャネルの第 1 引数を条件として指定することで、該当するメッセージのみを受信する制御が行えるためである。

表 11-10：各 ID 端末の ID 番号割り振り

ID 端末	定数名称	端末 ID 番号（識別子）
待合室扉脇 ID 端末	PREVIOUS_ROOM_DOOR_ ID_TERMINL	101
閲覧室 A 扉脇 ID 端末	BOOTH_ROOM_A_DOOR_ ID_TERMINAL	201
閲覧室 B 扉脇 ID 端末	BOOTH_ROOM_B_DOOR_ ID_TERMINAL	301
閲覧室 A 内 ID 端末	INSIDE_BOOTH_ROOM_A_ ID_TERMINAL	202
閲覧室 B 内 ID 端末	INSIDE_BOOTH_ROOM_A_ ID_TERMINAL	302

各 ID 端末から送信可能なメッセージ（表 11-11）は、id_terminal_ok と check_id_number の 2 通りを定義した。待合室扉脇 ID 端末および閲覧室 X 扉脇 ID 端末では、ID 端末内部で ID カード番号の照合を行い、扉のロック／アンロックを制御し、成功の場合についてのみ、待合室管理プロセスまたは閲覧室管理プロセスで処理を行うため（11.2.3.1.5 節、11.2.3.1.6 節）、id_terminal_ok のみを定義した。check_id_number は、閲覧室 X 管理プロセスにおける閲覧室内認証処理（11.3.4.1.3□節）において利用する。閲覧室内認証処理では、閲覧室 X 内 ID 端末から入力された ID 番号が、待機者リストにおいて当該閲覧室に案内されたとされる ID 番号と同一であるか検査するため、check_id_number を定義した。

表 11-11:ID 端末から送信可能なメッセージ

メッセージ	意味	送信可能 ID 端末
id_terminal_ok	ID 端末において認証 OK であったことを示す。	待合室扉脇 ID 端末(101) 閲覧室 A 扉脇 ID 端末(201) 閲覧室 B 扉脇 ID 端末(301)
check_id_number	入力されたカード ID 番号の検証を行うことを要求する。	閲覧室 A 内 ID 端末(202) 閲覧室 B 内 ID 端末(302)

11.3.4.2.2. 待合室扉脇ID端末

案内サービスプログラムでは待合室扉脇 ID 端末で認証成功した場合のメッセージ id_terminal_ok の場合のみ処理を行うため(11.2.3.1.5 節)、プロセス数の削減を目的として、閲覧者のモデルに含めて実装することとした(図 11-9)。

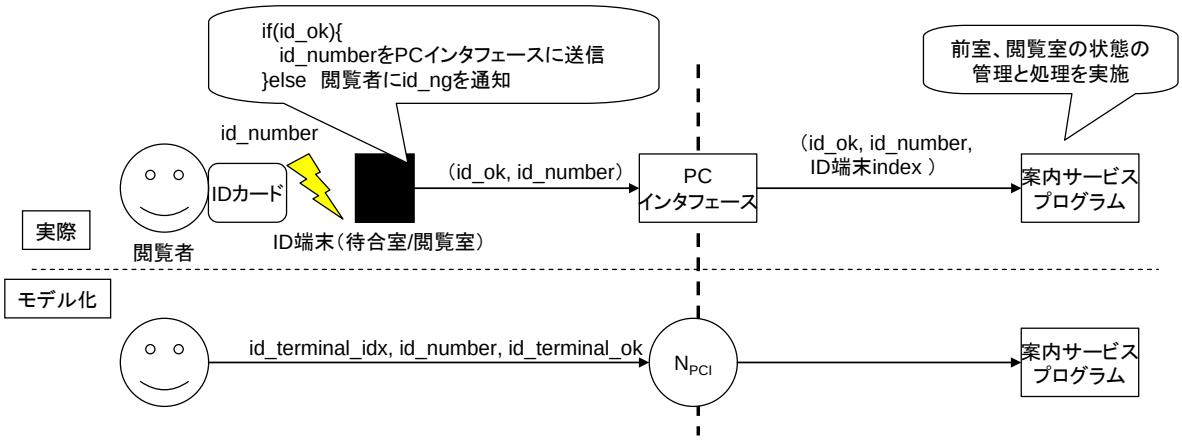


図 11-9:扉脇 ID 端末認証装置のモデル化

11.3.4.2.3. 閲覧室X扉脇ID端末

閲覧室 X 扉脇 ID 端末は、基本的には待合室扉脇と同様のモデリング(11.3.4.2.2 節)を行っている。ただし、閲覧室内 X に誰か別の閲覧者が存在する場合には、閲覧室 X の扉は開錠されない(表 11-6:閲覧室 X の扉の解錠条件表 11-6)ため、閲覧室 X の在室センサ BOOTH_ROOM_X_SENSOR_STATE==SENSOR_ON の場合は、チャネル pci に対してメッセージ id_terminal_ok は送信しない。逆に

BOOT_ROOM_X_SENSOR_STATE==SENSOR_OFF の場合は、pci に対して id_terminal_ok を送信する。

11.3.4.2.4. 閲覧室X内タッチパネル

閲覧室 X 内タッチパネルは、前提 1-2-1 よりモデリングに含めず、前提 1-2-3 から閲覧室 X 内 ID 端末の処理に含めることとした。

11.3.4.2.5. 閲覧室X内ID端末

閲覧室 X 内 ID 端末は、待合室扉脇 ID 端末および閲覧室 X 扉脇 ID 端末のモデリング(各 11.3.4.2.2 節、11.3.4.2.3 節)と同様に閲覧者のモデルに含めるが、閲覧室 X 管理プロセスにおいて閲覧室内認証処理(11.3.4.1.3 節)を行うため、チャネル pci には、{閲覧室 X 内 ID 端末 ID 番号、id_card_number、check_id_number}を送信する。

11.3.4.3. 室内センサ

11.3.4.3.1. 在室センサ

在室センサは、人感センサであり、設置された室内に閲覧者が 1 人以上存在する場合、センサ内部の状態を ON に、誰もいなくなった場合（OFF）に OFF にし、PC インタフェースに信号を送信する（11.2.3.2.1 節）。

Promela/SPIN におけるモデリングにおいては、周期的に待合室管理プロセスおよび閲覧室 X プロセスが各々に対応する在室センサの状態を監視することもあり、それぞれの在室センサの状態を PREVIOUS_ROOM_SENSOR_STATE および BOOTH_ROOM_X_SENSOR_ STATE（X=A or B）として定義し、入室時に閲覧者プロセスが SENSOR_ON とすることとした。

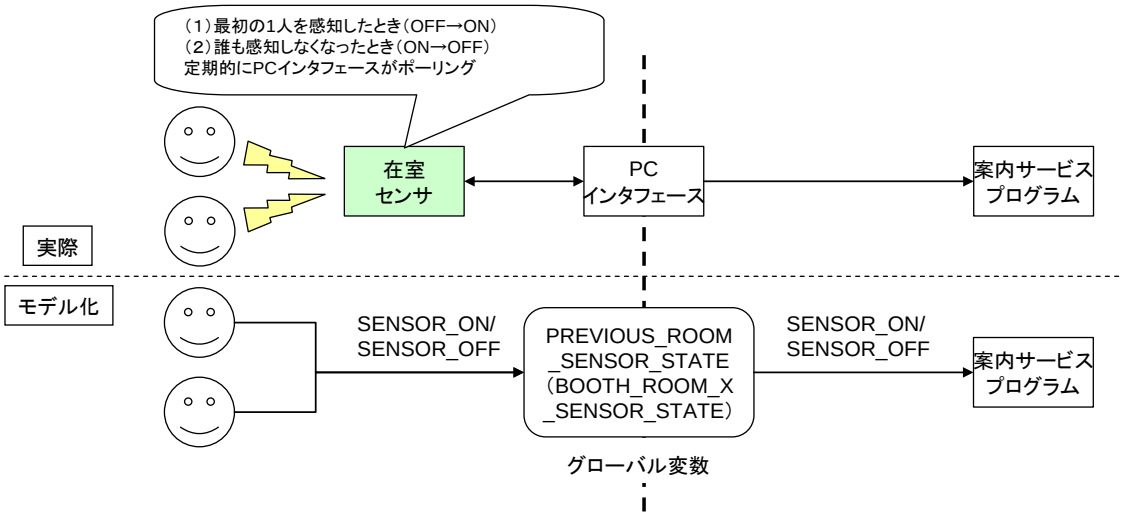


図 11-10：在室センサのモデリング

退出時には待合室と閲覧室 X で処理内容が異なる。待合室は複数人在室可能であるが、閲覧室 X は仕様上、1 名のみ在室可能である。すなわち、待合室から退出する閲覧者プロセスに含まれる在室センサ処理では、待合室に在室している閲覧者の人数が、自身が退出することで0以下となる場合には、PREVIOUS_ROOM_SENSER_STATE を SENSOR_OFF にセットし、1 以上の場合には何もしない。待合室の在室人数については、グローバル変数として HEAD_COUNT_OF_PREVIOUS_ROOM を定義している。一方で、閲覧室 X からの退出処理では、退出する場合 BOOT_ROOM_X_SENSOR_STATE を SENSOR_OFF にセットし、在室人数の確認は行わない。

11.3.4.3.2. 火災センサ

火災センサは、前提 0-1 からモデルには含めない。

11.3.4.4. データベース（DB）

データベース（DB）は、待合室管理プロセスおよび閲覧室 X（X=AorB）管理プロセスからの要求メッセージに応じて、受動的に処理を行うプロセスである。このため、プロセス数を削減するために、待合室管理プロセスおよび閲覧室 X 管理プロセスに組み込むことにした（図 11-11）。

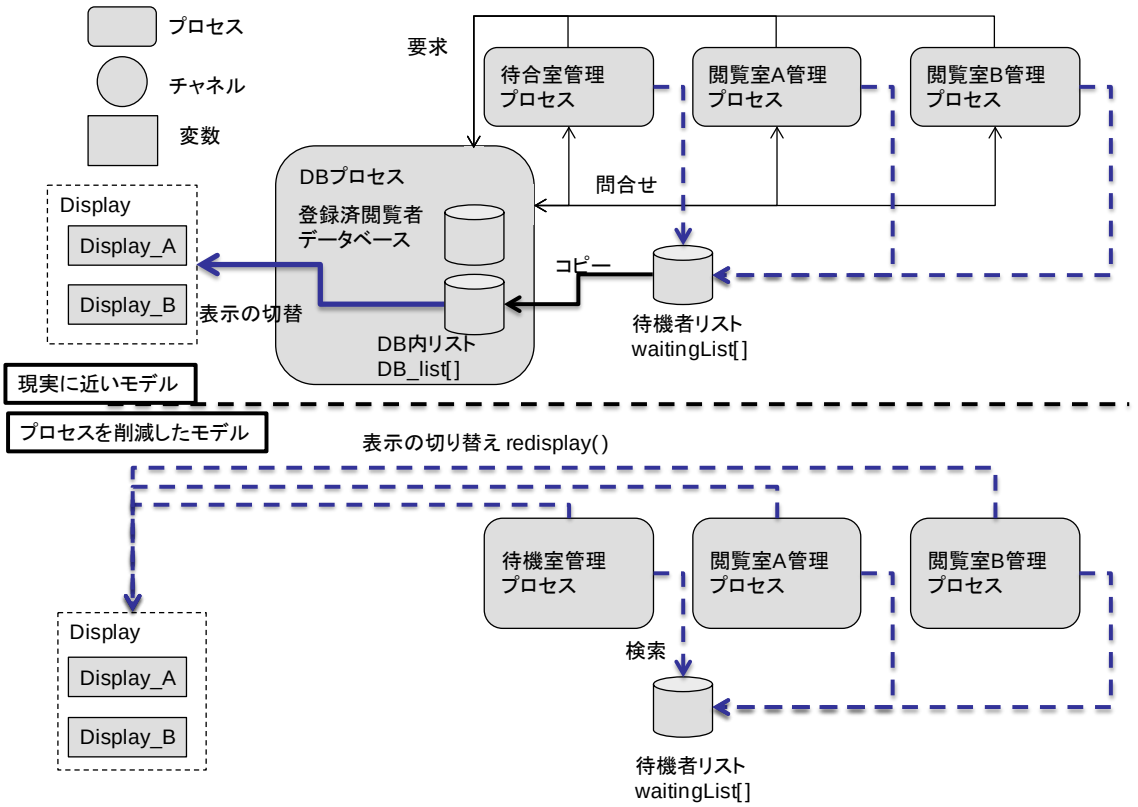


図 11-11:データベース(DB)と表示装置のモデル化

実際のシステムでは、各管理プロセス(待合室管理プロセス、閲覧室 X 管理プロセス)からの問合せや処理要求に応じて、DB は内部で管理している 2 つのデータベースそれぞれに対して、参照や操作を行う。各部屋の管理プロセスにおける処理と処理対象をまとめた表を表 11-12 に示す。

表 11-12:各管理プロセスの処理と処理対象

		対象			
処理主体	処理内容	待機者リスト		DB(内データベース)	
		参照	操作	参照	操作
待合室管理プロセス	新規顧客登録		○		
	ボックス格納室検索			○	
	DB 登録				○
閲覧室 X 管理プロセス	閲覧室扉認証				
	閲覧室内認証			○	
	案内者決定	○			
	DB 登録				○
	閲覧室退出		○		
DB	ボックス格納室検索			○	

プロセス	DB 登録				○
	表示装置表示内容更新			○	

DB プロセスを削減する際に問題となるのは、以下の点である。

- 待機者リスト waitingList[ ] の内容をコピーした DB 内リスト DB_list[ ] の更新タイミングの間にずれがあり、検証結果に影響はないか（つまり、DB プロセスを捨象したために、検証結果に違いが出ないか）

DB 内部の DB 内リスト DB_list[ ] は、待合室管理プロセスまたは閲覧室 X プロセスから DB 登録処理を要求された時点での待機者リスト waitingList[ ] の内容をコピーしたものである。また、表示装置の表示内容の更新は一定間隔で DB_list[ ]の内容を基に行われる。従って、DB プロセスを捨象するに当たって、DB_list[ ]が削減されるに当たり、モデル上では以下が上記の懸念に対する具体的な内容となる。

- DB 内リスト DB_list[ ] は待機者リスト waitingList[ ] の内容が更新された時点でその内容を正確に反映するものとみなすことができるか

DB 内部の DB 内リスト DB_list[ ] は、待機者リスト waitingList[ ] の（DB 登録が要求された時点での）内容をコピーしたリストである。待機者リスト waitingList[ ]の操作更新が行われた後、即座に DB 登録処理が行われるのであれば（つまり atomic として実装できるのであれば）、waitingList[ ] と DB_list[ ] はほぼ等価であると言えるだろう。ここで、waitingList[ ] の更新（登録、削除）は、人間である閲覧者が待合室に入室した時点（厳密には、閲覧者が待合室扉脇 ID 端末に認証をパスしたタイミングで実施される新規顧客登録の処理過程）、および、閲覧者が閲覧室 X から退出した時点（厳密には、閲覧室 X の在室センサが OFF であることを検知した時点で実施される閲覧室退出処理の処理過程）で行われる。このため、それぞれの処理（新規顧客登録および閲覧室退出処理）が行われる頻度は、閲覧者が待合室に入出するタイミングと閲覧室 X から退出するタイミングにほぼ等しく、waitingList[ ]が操作されたタイミングと DB 登録が実施されるタイミングの間に、これらの処理が行われることはほぼないだろう。従って、waitingList[ ]の操作の後、即座に DB 登録処理が行われる（つまり、atomic）であると仮定してモデリングしても今回の検証では問題ないと考えられる。

表示装置（Display）

案内の表示を行う表示装置は、DB 内に保管されている待機者リスト waitingList をコピーしたリストから、定期的に閲覧室 A および B への最新の案内者を表示する。

表示装置のモデリングでは、閲覧室 A へ案内する閲覧者の ID 番号をグローバル変数 display_A に、閲覧室 B へ案内する閲覧者の ID 番号をグローバル変数 display_B に、それぞれ代入するようにした。display_A および display_B への値の代入は、待合室管理プロセス、閲覧室 A 管理プロセスおよび閲覧室 B 管理プロセスの 3 つのプロセスが、インライン関数として実装した redisplay( ) の処理を通して行う。redisplay( ) は待機者リスト waiting_list[ ] から次に案内すべき閲覧者の ID 番号を検索し、display_A または display_B にその値を代入する。

各閲覧者は、display_A または display_B に代入されている値を参照し、自身に割り当てられている ID 番号が display_A の値と一致する場合は閲覧室 A へ、display_B の値と一致する場合には閲覧室 B へ移動する。display_A および display_B に自身に割り当てられた ID と一致する値がなかった場合、待機室にとどまる。

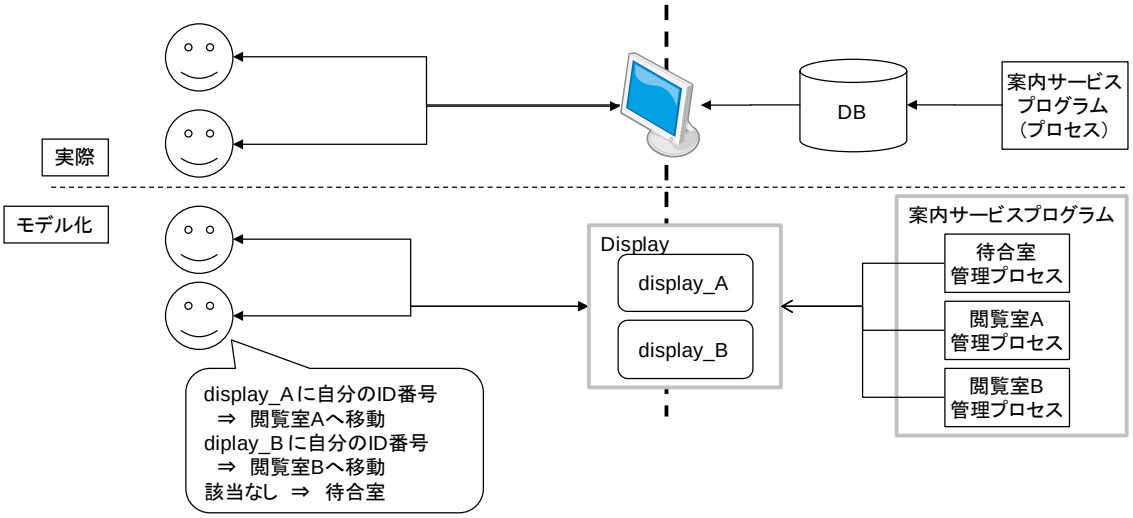
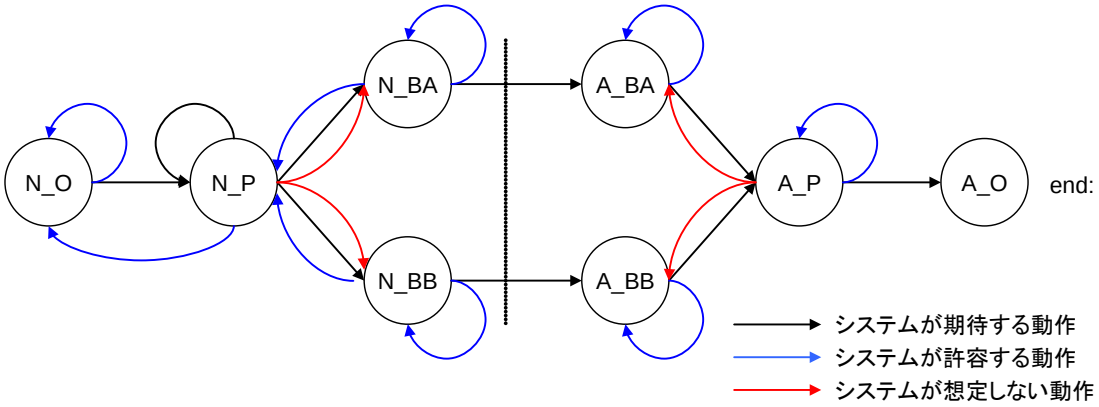


図 11-12:表示装置のモデリング

11.3.4.5. 閲覧者

閲覧者は、検証対象システムの外部環境としてモデリングする。外部環境は、物理的に起こりえるあらゆる状態を網羅した形でモデリングする。ここではシステムの想定以外の動作を閲覧者が行うことを前提として、システムで保証できる検証性質とそうでないものを区別することを目的として、閲覧者をモデリングすることとした。このため、閲覧者のモデルでは、検証対象システムのモデルに合わせて状態を細分化し、その状態間の遷移によって閲覧者の動作を Promela コード上で制御する方針とした。

実際のシステムで期待する閲覧者は、システム外部から待合室に入り、表示装置に案内される案内にそって閲覧室 X に移動し、閲覧室内 ID 端末およびタッチパネルの認証をパスして自身のボックスを取得し、閲覧室 X を退出して待合室に戻り、システムの外部に出て行き、一連の処理を終了する。ここで、例えば、表示装置の内容を無視して閲覧室 X に移動したり、ボックスを取得した後に閲覧室 X に戻るなどは、システムが想定しない閲覧者の動作である。他方、ボックスにアクセスしていないときにシステムの外に出ることや、閲覧室 X 内に留まること、ボックス取得後に待合室内に留まることなどは、システムが期待はしていないが許容されるべき閲覧者の動作である。これらの考察を基にして構成した、大枠での閲覧者のオートマトンを図 11-13 に示す。



ラベル	状態		ラベル	状態	
	ボックス	部屋		ボックス	部屋
N_O	×	システムの外	A_O	○	システムの外
N_P	×	PREVIOUS_ROOM(待合室)	A_P	○	PREVIOUS_ROOM(待合室)
N_BA	×	BOOTH_ROOM_A(閲覧室A)	A_BA	○	BOOTH_ROOM_A(閲覧室A)
N_BB	×	BOOTH_ROOM_B(閲覧室B)	A_BB	○	BOOTH_ROOM_B(閲覧室B)

図 11-13:閲覧者のオートマトン(大枠)

また、待合室における閲覧者を詳細に見ていくと、待合室入室後(INTO)、表示装置を見て(WATCH)、自分が案内されている場合はその閲覧室の扉にむかい(TO_DOOR_A または TO_DOOR_B)、閲覧室X扉脇ID端末で認証の処理を行う。案内されていない場合には、案内されるまで待機する(WAIT)。待機状態 WAIT から TO_DOOR_A または TO_DOOR_B に直接移動することや、表示装置閲覧 WATCH において表示装置 display の内容を無視して TO_DOOR_A または TO_DOOR_B に直接移動することもありえるが、この動作はシステムが想定していない動作である。他方、WAIT の際に所要を思い出すなどして外にある(LEAVE)や閲覧室Xの扉まで行って戻るなどは、システムとしては期待していないが許容できる動作である。異常の考察から、待合室における閲覧者のオートマトンを図 11-14 とした。

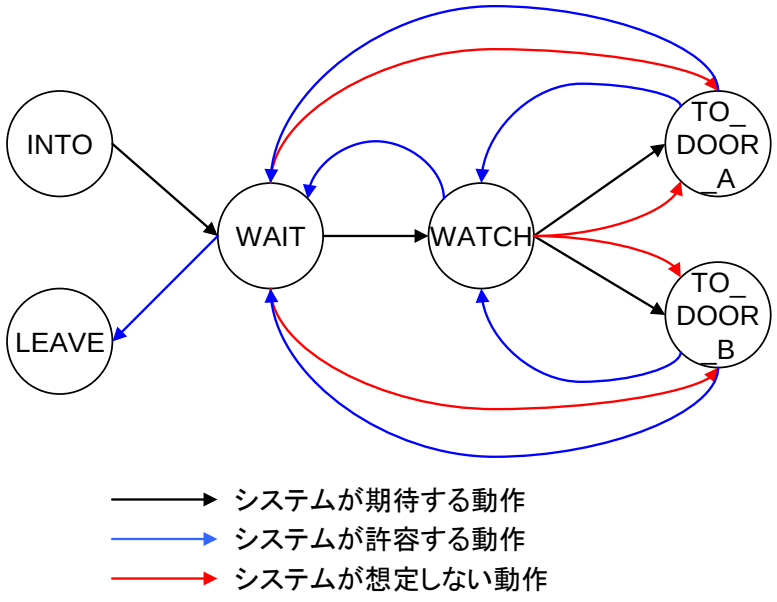


図 11-14:待合室における閲覧者のオートマトン

11.4. 検証性質の形式記述

11.4.1. 要求仕様と検証項目

上述0節で述べた検証対象システムの要求仕様と、11.3節で述べたPromela/SPINによるモデルに適用する検証性質との関係を表 11-13 に示す。検証性質については、モデルの検証に適した形に翻訳している。ここに挙げた検証性質のうち、「しないこと」に分類される検証性質を中心に検証を行った。

表 11-13:本ケーススタディにおける要求仕様と検証性質

分類	要求仕様	No.	検証性質	備考	検証対象
すること	ID カードを登録している人は、待合室に入ることができる			簡略版モデルでは、全ての閲覧者は登録済み(前提 1-5)	×
	待合室に入った人は、必ず案内される	L01	待合室に入室した人は、いつか必ず案内される	ー	○
	閲覧室の扉認証は、閲覧室内が空室なら誰でも成功する			センサを閲覧者のモデルに組み込んでいるため	×
	案内された人は、一定時間、閲覧室のボックス取り出しの認証を自分の ID カードでパスできる。	L02	閲覧室に案内された人だけが、タッチパネル脇 ID 端末で、いつか必ず認証 OK となる	簡略版モデルでは、時間を取り扱っていない(前提 1-1)	○
		L03	閲覧室から退出した閲覧者は、いつか必ず待機者リストから削除される	システム構築ベンダー殿からの要請	○
しないこと	閲覧室に人が入室すると、その人が退室するまで、閲覧室のドア認証に成功しない。(最終的に保証すべき安全性)	S01	閲覧室に案内されていない人が、タッチパネル脇 ID 端末で認証を行っても、常に認証をパスしない	簡略版モデルでは、時間を取り扱っていない(前提 1-1)	◎
	閲覧室が空室の時以外、閲覧室に人を案内しない。	S02	閲覧室 X が空室のとき以外、常に案内は変更されない	ー	◎
	利用者は、どの部屋にも閉じ込められない。(常に、退出することが出来る。)			簡略版モデルでは閲覧室の扉を取り扱っていない(前提 1-3)	×
		S03	待機者リストは常にオーバーフローすることはない	コードデバック中に index エラーを見つけたため	◎

◎:検証済み　　○:検証対象だが、未検証　　×:検証対象外

11.4.2. 検証方法

11.4.2.1. 検証内容　S01

検証内容「閲覧室に案内されていない人が、タッチパネル脇 ID 端末で認証を行っても、常に認証をパスしない」について、その検証方法を説明する。

S01 の表現を変え、「常に閲覧室 X に案内された人だけが、閲覧室 X 内タッチパネル脇 ID 端末の認証をパスする」を示すことにする。

閲覧室は閲覧室 A および閲覧室 B の 2 つがあり、閲覧室 X（A または B）に案内された閲覧者が、閲覧室 X 内タッチパネル脇 ID 端末で認証行為を行うと、いつか必ず制御 PC（チャネル pc_contoroller）を通して id_number_ok を受信する。従って、Promela コード上の閲覧者プロセスにおいて、チャネル pc_interface に check_id_number メッセージを送信する部分と、pc_controller から閲覧者プロセスが id_number_ok を受信する部分に、それぞれ SEND_CHECK_ID_X および GET_ID_OK_X のラベルを付す。

案内表示装置（チャネル display）において閲覧室 X へ案内している ID については、display に送信後 DISPLAY_ID_A にその id_number を記録しておき、閲覧者プロセスが所持している id と比較する。閲覧者プロセスが所持する id への参照は「プロセス名[_pid]:id」の形式で行うことができる。

閲覧室 X は、閲覧室 A または閲覧室 B であることから、それぞれに関して LTL を構成し、最終的には下記のようになる。

```
!□(((Human[6]:id == DISPLAYING_ID_A)&&
□(Human[6]@SEND_CHECK_ID_A→◇Human[6]@GET_ID_OK_A)) ||
((Human[6]:id == DISPLAYING_ID_B)&&
□(Human[6]@SEND_CHECK_ID_B→◇Human[6]@GET_ID_OK_B)))
```

図 11-15：検証内容 S01 の LTL 式

11.4.2.2. 検証内容　S02

検証内容「閲覧室 X が空室のとき以外、常に案内は変更されない」について説明する。

ここでは、「閲覧室 X が在室状態になったとき、いつか必ず案内が変更されることは常にない」を示すことにする。

閲覧室 X の在室については、在室センサが ON のとき、つまり、BOOTH_ROOM_X_SENSOR_STATE==SENSOR_ON の場合を考える。このとき、「いつか必ず案内が変更される」という命題を、応答性を用いて LTL の記述を行う。応答性は、命題 p、q を用いて p→◇q（p が成立したならば、いつか必ず q が成り立つ）で記述する。

次に、「案内は変更される」という命題について検討する。モデル上では、案内装置(display)は DB 登録処理のタイミングでチャネル display をリフレッシュ（チャネル display に格納されているメッセージを全て DB プロセスで破棄）して、最新の案内（閲覧室 A および閲覧室 B、それぞれに対応する待機者リストのカード ID の表示）をチャネル Display に送信することとなっており、厳密な意味では、DB 登録が行われると常に案内の内容は変更されることになる。ここでの「案内が変更される」は、「チャネル Display の内容が更新前後で変っていること」である。閲覧室 X の案内内容の変更の有無を DISPLAY_X_CHANGED として定義した上で、DB 登録処理実施前後の閲覧室 X への案内閲覧者 ID を変数 new_booth_X、old_booth_X に記録して差異を比較する。異なれば DISPLAY_X_CHANGED に true を代入、同じであれば DISPLAY_X_CHANGED に false を代入するように DB における DB 登録処理直後に記述を追加する。

```
! □((BOOTH_ROOM_X_SENSOR_STATE==SENSOR_ON) →
        ◇ (DISPLYA_X_CHANGED==true))
```

図 11-16:検証内容 S02 の基本となる LTL 式

閲覧室 X は閲覧室 A および閲覧室 B の 2 つがあるため、閲覧室 A または閲覧室 B のいずれかが在室のとき案内が変更さることはないことを示す必要がある。従って最終的には、下記のようになる。

```
!□(((BOOTH_ROOM_A_SENSOR_STATE==SENSOR_ON)
        → ◇(DISPLAY_A_CHANGED==true)) ||
    ((BOOTH_ROOM_B_SENSOR_STATE==SENSOR_ON)
        → ◇(DISPLAY_B_CHANGED==true)))
```

図 11-17:検証内容 S02 の LTL 式

11.4.2.3. 検証内容 S03

検証内容「待機者リストは常にオーバーフローすることはない」の検証方法について、説明を行う。

本検証を行う方法として、上記と同様に検証記述を行う方法と assert 文を用いて不具合を自動検出する方法が考えられる。本検証内容は、「待機者リスト waiting_list[ ] への登録を行う処理では、常に待機者リスト waiting_list[ ] の大きさ未満である」と言い換えることができる。つまり、waiting_list[ ]への登録を行う(データの登録を行う)時点で、エラーの検出ができればよいため、検証記述を用いるよりも簡単な assert 文を用いた自動検証により検証を行う方針を採用することとする。

待合室における Promela 記述を図 11-18 に示す。これに図 11-19 に示すように、待機者リストへの登録(registrate_waitingList())を実行する直前に、

assert(waiting_list_idx<MAX_WAITING_LIST_SIZE)を挿入する(図 11-19、16 行目、39 行目)。waiting_list_idx は、待機者リスト waiting_list[ ]に登録を行うために利用するインデックスを記憶する。待機者リスト waiting_list[ ]は配列で表現していることから、インデックスは 0 から始まるため、待機者リストの最大サイズ(MAX_WAITING_LIST_SIZE)－1までは、待合室に入室した閲覧者の登録が可能である。このため、waiting_list_idx が MAX_WAITING_LIST_SIZE 未満である場合は不具合ではなく、assert 文はエラーを検出しない。逆に waiting_list_idx が MAX_WAITING_LIST_SIZE 以上となる場合、オーバーフローの不具合であり、assert 文はエラーを検出し、トレイルファイルにその実行パターンを出力する。

```
1   active proctype Previous_Room(){
2       int   room_number = PREVIOUS_ROOM;
3       int   waiting_list_idx = 0;         /*   待機者リストのインデックス   */
4       int   list_idx =0;                  /*   待機者リスト検索 ( Display 表示時 ) 用 */
5       mtype   pci_m;                              /* pc_interface から受信するメッセージ */
6       int   input_id;                                     /* pc_interface から受信した card_id 番号 */
7
8       ABSENCE:          /** 不在状態 */
9               if
10              ::  (pc_interface?[PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m])->
11                      /* 前室扉脇 ID 端末からのメッセージを検索   */
12                  pc_interface??PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m;
13                      /* 前室扉脇 ID 端末からのメッセージを受信 */
14                  if
15                  ::      (pci_m==id_terminal_ok) ->          /* 認証 OK なら前室入室処理 */
16                          registrate_waitingList(waiting_list_idx, input_id);
17                          redisplay( );
18                          goto   PRESENCE;
19                  ::      else -> goto ABSENCE;
20                  fi;
21              ::  (PREVIOUS_ROOM_SENSOR_STATE == SENSOR_OFF) ->
22                          /*   周期監視在室 OFF */
23                          goto ABSENCE;
24              ::  (PREVIOUS_ROOM_SENSOR_STATE == SENSOR_ON)           ->
25                          /*              周期監視在室 ON        */
26                          goto PRESENCE;
27              ::  (HC_AO == HEAD_COUNT) -> goto TERMINATE;      /* 全員 A_O なら終了*/
28              ::  else -> goto ABSENCE;
29              fi;
30
31      PRESENCE:         /** 在室状態 */
32              if
33              ::  (pc_interface ?? [PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m]) ->
34                          pc_interface ?? PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m;
35                          /* 前室扉脇 ID 端末からのメッセージを受信 */
36                  if
37                  ::  (pci_m==id_terminal_ok) ->              /* 認証 OK なら前室入室処理 */
38                          /* 新規顧客作成 */
39                          registrate_waitingList(waiting_list_idx, input_id);pr:
40                          redisplay();
41                          goto   PRESENCE;
42                  ::      else -> goto PRESENCE;
43                  fi;
44              ::  (PREVIOUS_ROOM_SENSOR_STATE == SENSOR_OFF) ->       /*   周期監視在室 OFF */
45                  goto   ABSENCE;
46              fi;
47
48      TERMINATE: end:
49              skip;
50  }
```

図 11-18:待合室の Promela 記述

```
1   active proctype Previous_Room(){
```

```
2       int   room_number = PREVIOUS_ROOM;
3       int   waiting_list_idx = 0;         /*   待機者リストのインデックス   */
4       int   list_idx =0;                  /*   待機者リスト検索 ( Display 表示時 ) 用 */
5       mtype   pci_m;                      /* pc_interface から受信するメッセージ */
6       int   input_id;                                 /* pc_interface から受信した card_id 番号 */
7
8       ABSENCE:          /** 不在状態 */
9               if
10              ::  (pc_interface?[PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m])->
11                      /* 前室扉脇 ID 端末からのメッセージを受信 */
12                  pc_interface??PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m;
13                      /* 前室扉脇 ID 端末からのメッセージを受信 */
14                  if
15                  ::      (pci_m==id_terminal_ok) ->          /* 認証 OK なら前室入室処理 */
16                          assert(waiting_list_idx<MAX_WAITING_LIST_SIZE);
17                          registrate_waitingList(waiting_list_idx, input_id);
18                          redisplay( );
19                          goto PRESENCE;
20                  ::      else -> goto ABSENCE;
21                  fi;
22              ::  (PREVIOUS_ROOM_SENSOR_STATE == SENSOR_OFF) ->
23                      /*   周期監視在室 OFF */
24                      goto ABSENCE;
25              ::  (PREVIOUS_ROOM_SENSOR_STATE == SENSOR_ON)
26                      /*              周期監視在室 ON        */
27                      goto PRESENCE;
28              ::  (HC_AO == HEAD_COUNT) -> goto TERMINATE;      /* 全員 A_O なら終了*/
29              ::  else -> goto ABSENCE;
30              fi;
31
32      PRESENCE:       /** 在室状態 */
33              if
34              ::  (pc_interface ?? [PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m]) ->
35                      pc_interface ?? PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m;
36                      /* 前室扉脇 ID 端末からのメッセージを受信 */
37                  if
38                  ::  (pci_m==id_terminal_ok) ->              /* 認証 OK なら前室入室処理 */
39                      assert(waiting_list_idx<MAX_WAITING_LIST_SIZE);
40                      /* 新規顧客作成 */
41                      registrate_waitingList(waiting_list_idx, input_id);pr:
42                      redisplay();
43                      goto   PRESENCE;
44                  ::      else -> goto PRESENCE;
45                  fi;
46              ::  (PREVIOUS_ROOM_SENSOR_STATE == SENSOR_OFF) ->       /*   周期監視在室 OFF */
47                   goto   ABSENCE;
48              fi;
49
50      TERMINATE: end:
51              skip;
52  }
```

図 11-19：assert 文を追記した待合室の Promela 記述

11.5. モデル検査と結果

11.5.1. 検証内容 S01 の検証結果

反例は見つからなかった。

11.5.2. 検証内容 S02 の検証結果

反例は見つからなかった。

11.5.3. 検証内容 S03 の検証結果

閲覧者の人数（つまり、閲覧者プロセス数）を 1、待機者リスト waiting_list[ ]の最大サイズ（MAX_WAITING_LIST_SIZE）を 5 として、検証を行った結果、assert 文においてエラーを検出した（図 11-20）。

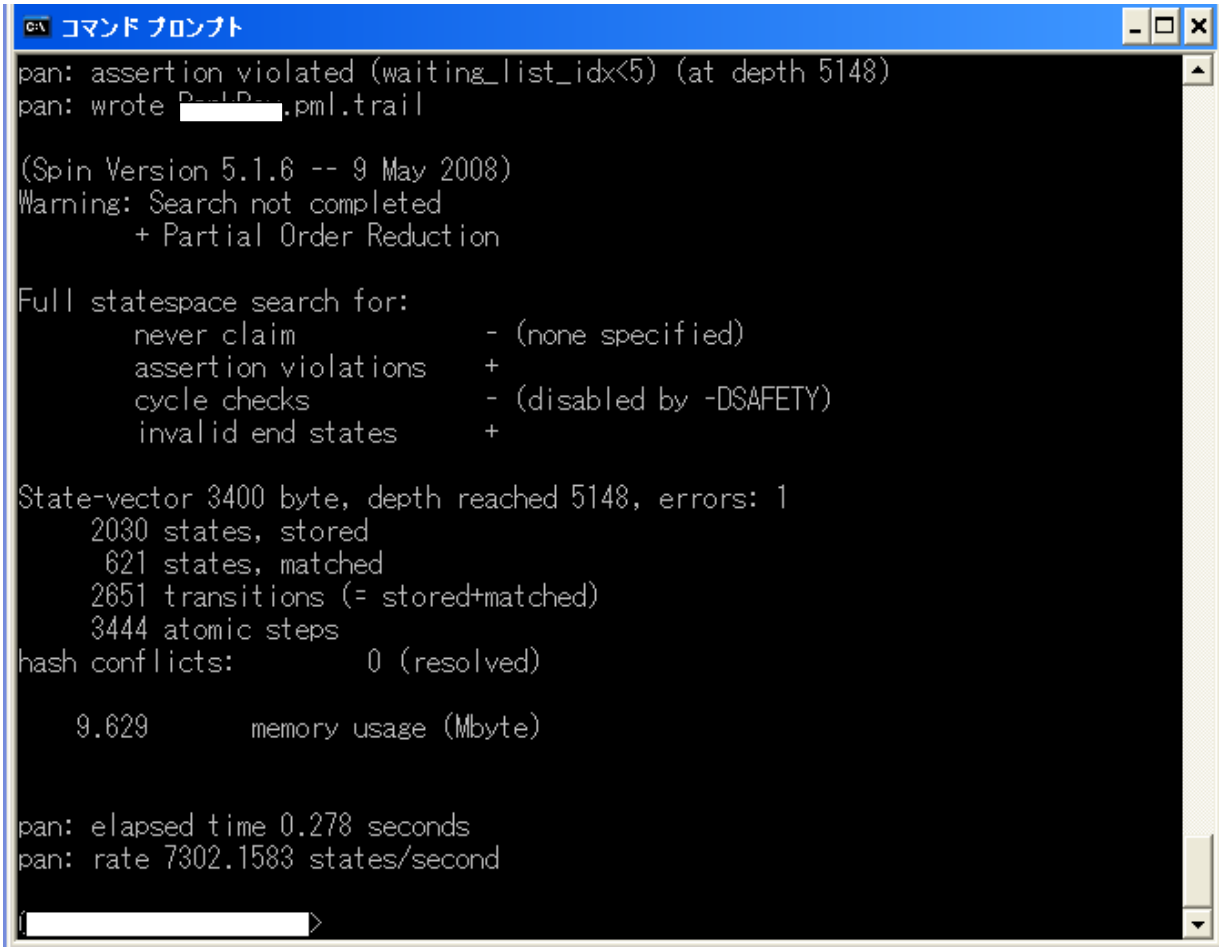


図 11-20：S03 の検証結果

図 11-20 の 11 行目に“State-Vecotor 3400byte, depth reached 5148, errors:1”とあり、エラーが検出されている。また、1 行目に“pan： assertion violated (waiting_list_idx<5) (at depth 5148)”とあることから、assert 文のところでエラーが検出され、次の 2 行目でその結果を”.trail”ファイルに書き込んでいる。

この”.trail” ファイルに基づき Spin のシミュレーション実行を行う。具体的には、「-t」オプションと共に、「-p」オプションを用いて、“spin –t –p ＜promela_code.pml＞ > <result.txt>”とプロンプト

から入力し、実行する。＜promela_code.pml＞には、検証に用いているPromelaで記載されたファイル名を代入する。<result.txt>は実行結果の出力先ファイル名を記載する。

＜result.txt＞には、実行パターンが出力されている。この結果を分析することで、エラーが起きる実行パターンを解析する。

trailファイルの分析の結果、閲覧者プロセスが外部から待合室への入室を待機者リストwaiting_list[ ] の最大サイズ MAX_WAITING_LIST 回繰り返すと、waiting_list_idx＞MAX_WAITING_LIST となり、反例となることが分かった。

このエラーは、同一の閲覧者が複数回待合室への入退出を繰り返すことにより発生する。つまり、同一の閲覧者を複数回待機者リストに登録することによって発生する。従って、待機者リストへの登録の処理（registrate_waitingList() ）を実行する直前に、登録処理の対象となる閲覧者が待機者リストに登録されているか確認し、登録されていない場合だけ registrate_wiatingList()の処理を行うようにすれば、解決する。この関数をインライン関数として、実装した待合室の Promela 記述を図11-21 に示す。図 11-21 における check_waitingList(input_id) がこの対処に相当する。この処理により、当該エラーは回避されることを、同様に検証器を作成して実行することで確認できた。

```
1   active proctype Previous_Room(){
2       int   room_number = PREVIOUS_ROOM;
3       int   waiting_list_idx = 0;          /*   待機者リストのインデックス   */
4       int   list_idx =0;                   /*   待機者リスト検索 ( Display 表示時 ) 用 */
5       mtype   pci_m;                       /* pc_interface から受信するメッセージ */
6       int   input_id;                      /* pc_interface から受信した card_id 番号 */
7
8       ABSENCE:          /** 不在状態 */
9               if
10              ::  (pc_interface?[PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m])->
11                          /* 前室扉脇 ID 端末からのメッセージを受信 */
12                      pc_interface??PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m;
13                          /* 前室扉脇 ID 端末からのメッセージを受信 */
14                  if
15                  ::      (pci_m==id_terminal_ok) ->          /* 認証 OK なら前室入室処理 */
16                          assert(waiting_list_idx<MAX_WAITING_LIST_SIZE);
17                          check_waitingList(input_id);
18                          registrate_waitingList(waiting_list_idx, input_id);
19                          redisplay( );
20                          goto PRESENCE;
21                  ::      else -> goto ABSENCE;
22                  fi;
23              ::  (PREVIOUS_ROOM_SENSOR_STATE == SENSOR_OFF) ->
24                          /*   周期監視在室 OFF */
25                          goto ABSENCE;
26              ::  (PREVIOUS_ROOM_SENSOR_STATE == SENSOR_ON)           ->
27                          /*              周期監視在室 ON        */
28                          goto PRESENCE;
29              ::  (HC_AO == HEAD_COUNT) -> goto TERMINATE;      /* 全員 A_O なら終了*/
30              ::  else -> goto ABSENCE;
31              fi;
32
33      PRESENCE:         /** 在室状態 */
34              if
35              ::  (pc_interface ?? [PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m]) ->
36                          pc_interface ?? PREVIOUS_ROOM_DOOR_ID_TERMINAL,input_id,pci_m;
37                          /* 前室扉脇 ID 端末からのメッセージを受信 */
38                  if
39                  ::  (pci_m==id_terminal_ok) ->              /* 認証 OK なら前室入室処理 */
40                          assert(waiting_list_idx<MAX_WAITING_LIST_SIZE);
41                          check_waitingList(input_id);
42                          registrate_waitingList(waiting_list_idx, input_id);pr:
43                          redisplay();
44                          goto   PRESENCE;
45                  ::      else -> goto PRESENCE;
46                  fi;
47              ::  (PREVIOUS_ROOM_SENSOR_STATE == SENSOR_OFF) ->       /*   周期監視在室 OFF */
48                  goto   ABSENCE;
49              fi;
50
51      TERMINATE: end:
52              skip;
53  }
```

図 11-21 : 対処を行った待合室の Promela 記述