自動検針システム

検査項目と検査の結果

付属書３

目次

本書は、自動検針システムのモデル検査の検査項目と検査の結果について述べたものである。

１．検査項目の選定

「付属書１　自動検針システムのシステム仕様書」に記述された各設備の動作仕様の全てと、それぞれの動作仕様を組み合わせることによって得られるシステム全体の動作仕様を検査項目とした。

（１）設備の動作仕様

「付属書１　自動検針システムのシステム仕様書」の「GWの動作仕様」と「WHMの動作仕様」の中の各項目をそのまま検査項目とした。ただし今回はWHM内でのデータ設定と、センタ～GW～WHM間の通信に重点を置いている。従って、センタからの要求の出力は、GWでの要求の入力処理を検査するための手段として考え、センタ自体の動作仕様は検査項目に含めない。

（２）システム全体の動作仕様

システム全体についての動作仕様として次の（ａ)、（ｂ）２つを検査項目とした。

（ａ）パターン設定完了の確認

センタ～GW～WHM間の通信が正常に動作し、WHM内でのデータ設定も正常に動作すれば、自動検針システムの重要な動作の１つであるパターンデータの設定が完了する。従って、次の検査項目を設けた。

「センタから要求が出力されれば、最終的にWHM内でのパターン設定が完了すること」

（ｂ）初期状態への再帰可能性

自動検針システムはどんな状態からでも、デッドロックにおちいらず、いつかは初期状態に戻ることが期待される。従って次の検査項目を設けた。

「システムが初期状態から動作して、いかなる状態からでも再び初期状態へ戻ることができること」

２．検査項目とその結果

ここでは、検査項目毎に、検査の際の付加条件、条件付けの方法、ＣＴＬ論理式、検査結果を示す。また、検査の過程で留意すべき点がある場合は「メモ」として記録した。以下、２．１でGWの動作仕様の検査、２．２でWHMの動作仕様の検査、２．３でシステム全体の動作の検査について述べる。

２．１　GWの動作仕様の検査

２．１．１　GWの動作仕様（１）－（ａ）－仕様①の検査

（１）要求の受付及び指令データの作成・出力
（ａ）検針要求の受付
　①センタから検針要求があった場合、センタとGW間の通信が正常であれば要求を受け付け検針指令データを作成しWHMへ出力する。

センタとGW間の通信状態は、GWの電源の状態と同じである（注）ので、GWの電源が正常でセンタから検針要求があれば、検針指令データ（正常データ：Ｍ　ｏｒ　異常データ：ＮＧ）を作成することを検査する。

注)「付属書２　仕様のモデル化とコード化　－　３－（２）－（ａ)」参照

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Req = M & GW_P = ON ) -> AX( Ord_Dt = M | Ord_Dt = NG ) )
```

（４）検査結果　：　ｔｒｕｅ

２．１．２　GWの動作仕様（１）－（ａ）－仕様②の検査

（１）要求の受付及び指令データの作成・出力
（ａ）検針要求の受付
②センタから検針要求があった場合、センタとGW間の通信が異常であれば要求を受け付けることは出来ず、検針指令データは作成することは出来ないこととととする。

GWの電源が異常であれば、次にGWの電源が正常で検針要求を受け付けるまでは、検針指令データを作成しないことを検査する。

（１）付加条件

いつか必ず、GWの電源が正常で検針要求を受け付けることとする。

（２）条件付けの方法

SMV ソースコード内で、「FAIRNESS　GW_P = ON & Req = M」を付加する。

（３）ＣＴＬ論理式

```
AG ( ( Req = M & GW_P = OFF ) -> A[ Ord_Dt != M U ( GW_P = ON & Req = M ) ] )
```

＜メモ＞　上記ＣＴＬ論理式について

(1) 条件「FAIRNESS GW_P = ON & Req = M」を付加しない場合
　→　前条件が成立してから、永久に GW_P = ON とならないパスが存在するため、False となる。
　　（ A[φUψ]　は　ψが永久に１とならないパスが存在すれば False となる ）

(2) 条件「FAIRNESS GW_P = ON」を付加した場合
　→　Req = Wt の時だけ GW_P = ON となり（Req = M の時には GW_P = OFF となり）、GWがいつまでも要求を受け付けないパスが存在するため、False となる。
　　（ A[φUψ]　は　ψが永久に１とならないパスが存在すれば False となる ）

（４）検査結果　：　ｔｒｕｅ

２．１．３　GWの動作仕様（１）－（ｂ）－仕様①の検査

（１）要求の受付及び指令データの作成・出力
（ｂ）パターン設定要求の受付
①センタからパターン設定要求１があった場合、センタとGW間の通信が正常であれば要求を受け付け、パターン設定指令１データを作成しWHMへ出力する。
また、GWからWHMへ出力する指令データは、内容が正しい正常データである場合と内容が間違っている異常データである場合があるものとする。

本項目は、２．１．１で、検針要求をパターン設定要求に置き換えることにより検査することができる。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Req = P1 & GW_P = ON ) -> AX( Ord_Dt = P1 | Ord_Dt = NG ) )
AG ( ( Req = P2 & GW_P = ON ) -> AX( Ord_Dt = P2 | Ord_Dt = NG ) )
AG ( ( Req = P3 & GW_P = ON ) -> AX( Ord_Dt = P3 | Ord_Dt = NG ) )
```

（４）検査結果　：　ｔｒｕｅ

２．１．４　GWの動作仕様（１）－（ｂ）－仕様②の検査

（１）要求の受付及び指令データの作成・出力
（ｂ）パターン設定要求の受付
②センタからパターン設定要求１があった場合、センタとGW間の通信が異常であれば要求を受け付けることは出来ず、パターン設定指令１データは作成することは出来ないこととする。パターン設定指令２データ及びパターン設定指令３データについも同様とする。
また、GWからWHMへ出力する指令データは、内容が正しい正常データである場合と内容が間違っている異常データである場合があるものとする。

本項目は、２．１．２で、検針要求をパターン設定要求に置き換えることにより検査することができる。

（１）付加条件

いつか必ず、GWの電源が正常でパターン設定要求１、２、３を受け付けることとする。

（２）条件付けの方法

●パターン設定指令１データの作成

SMV ソースコード内で、

「FAIRNESS　GW_P = ON & Req = P1」を付加する。

●パターン設定指令２データの作成

SMV ソースコード内で、

「FAIRNESS　GW_P = ON & Req = P2」を付加する。

●パターン設定指令３データの作成

SMV ソースコード内で、

「FAIRNESS　GW_P = ON & Req = P3」を付加する。

（３）ＣＴＬ論理式

```
AG ( ( Req = M & GW_P = OFF ) -> A[ Ord_Dt != P1 U ( GW_P = ON & Req = P1 ) ] )
AG ( ( Req = M & GW_P = OFF ) -> A[ Ord_Dt != P2 U ( GW_P = ON & Req = P2 ) ] )
AG ( ( Req = M & GW_P = OFF ) -> A[ Ord_Dt != P3 U ( GW_P = ON & Req = P3 ) ] )
```

（４）検査結果　：　ｔｒｕｅ

２．１．５　GWの動作仕様（２）の検査

（２）WHMからの返送入力
　　WHMから返送が出力された場合は、GWとWHM間の通信が正常であれば返送を入力する。
　　（GWとWHM間の通信が異常の場合は（５）の条件に従う）

WHMからの返送（Ret）を入力した場合、GWはセンタへの報告（Rpt）を出力する。従ってここでは、「センタへの報告を出力する処理　＝　WHMからの返送入力完了」とみなして検査を行う。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Ret = OK & Com = 1 ) -> AX( Rpt = OK ) )
```

（４）検査結果　：　ｔｒｕｅ

２．１．６　GWの動作仕様（３）の検査

（３）センタへの報告（正常時）
　　WHMからの返送を入力した場合は処理が正常に完了したとみなして、その旨（ＯＫ）の報告をセンタに出力する。

GWとWHM間の通信が正常で、返送を入力した場合は、センタへの報告がＯＫとなることを検査する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Ret = OK & Com = 1 ) -> AX( Rpt = OK ) )
```

（４）検査結果　：　ｔｒｕｅ

２．１．７　GWの動作仕様（４）－仕様①の検査

（４）センタへの報告（異常時１）
GWからWHMへ指令データを出力する際に、GWとWHM間の通信に異常が発生し指令データを出力出来なかった場合は、処理が正常に完了しなかったとみなして、以下の条件に従う。
①GWの電源が正常であればその旨（ＮＧ）の報告をセンタに出力する。

「指令データを出力する」は、指令データに何らかのデータが入っている状況であり、「Ord_Dt != EMP」と表す。GWとWHM間の通信が異常でGWの電源が正常の場合に、指令データが「指令無し」でなければ、センタへの報告が NG となることを検査する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Com = 0 & GW_P = ON & Ord_Dt != EMP ) -> AX( Rpt = NG ) )
```

（４）検査結果　：　ｔｒｕｅ

２．１．８　GWの動作仕様（４）－仕様②の検査

（４）センタへの報告（異常時１）
GWからWHMへ指令データを出力する際に、GWとWHM間の通信に異常が発生し指令データを出力出来なかった場合は、処理が正常に完了しなかったとみなして、以下の条件に従う。
②GWの電源が異常であれば、センタへの報告は出力出来ないこととする。

GWの電源が異常であれば、次にGWの電源が正常でセンタから何らかの要求を受け付けるまで（一旦、初期状態に戻り、再度センタから要求を受け付けるまで）は、センタへの報告は「報告無し」のままであることを検査する。

（１）付加条件

いつか必ず、GWの電源が正常でセンタから何らかの要求を受け付けることとする。

（２）条件付けの方法

ＳＭＶソースコード内で、「FAIRNESS　GW_P = ON & Req != OFF」を付加する。

（３）ＣＴＬ論理式

```
AG ( ( Com = 0 & GW_P = OFF & Ord_Dt != EMP ) ->
                    A[ Rpt = EMP U ( GW_P = ON & Req != OFF ) ] )
```

（４）検査結果　：　ｔｒｕｅ

＜メモ＞

本検査項目では、センタの待機状態のリセット処理を追加するまでは、GWの電源が異常の場合はセンタへの報告ができず、センタは待機状態のままフリーズしてしまい、システムとしてそれ以上、状態遷移できない状態になっていた。リセット処理の追加により、前条件成立後に「GW_P = ON & Req != OFF」となるパスが存在することとなった。

２．１．９　GWの動作仕様（５）－仕様①の検査

（５）センタへの報告（異常時２）
WHMから返送が出力されたにも関わらず、GWとWHM間の通信に異常が発生し返送を入力することが出来なかった場合は、処理が正常に完了しなかったとみなして、以下の条件に従う。
①GWの電源が正常であればその旨（ＮＧ）の報告をセンタに出力する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Com = O & GW_P = ON & Ret = OK ) -> AX ( Rpt = NG ) )
```

（４）検査結果　：　ｔｒｕｅ

２．１．１０　GWの動作仕様（５）－仕様②の検査

（５）センタへの報告（異常時２）
WHMから返送が出力されたにも関わらず、GWとWHM間の通信に異常が発生し返送を入力することが出来なかった場合は、処理が正常に完了しなかったとみなして、以下の条件に従う。
②GWの電源が異常であれば、センタへの報告は出力出来ないこととする。

GWの電源が異常であれば、次にGWの電源が正常でセンタから何らかの要求を受け付けるまで（一旦、初期状態に戻り、再度センタから要求を受け付けるまで）は、センタへの報告は「報告無し」のままであることを検査する。

（１）付加条件

いつか必ず、GWの電源が正常でセンタから何らかの要求を受け付けることとする。

（２）条件付けの方法

ＳＭＶソースコード内で、「FAIRNESS　GW_P = ON & Req != OFF」を付加する。

（３）ＣＴＬ論理式

```
AG ( ( Com = 0 & GW_P = OFF & Ret = OK ) ->
                A[ Rpt = EMP U ( GW_P = ON & Req != OFF ) ] )
```

（４）検査結果　：　ｔｒｕｅ

＜メモ＞

本検査項目では、センタの待機状態のリセット処理を追加するまでは、ＧＷの電源が異常の場合はセンタへの報告ができず、センタは待機状態のままフリーズしてしまい、システムとしてそれ以上、状態遷移できない状態になっていた。リセット処理の追加により、前条件成立後に「GW_P = ON & Req != OFF」となるパスが存在することとなった。

２．１．１１　GWの動作仕様（６）の検査

（６）センタへの報告（異常時３）
ＧＷとWHM間の通信が正常であるにも関わらずWHMから返送を入力することが出来なかった場合（WHMが無応答の場合）は、その旨（ＮＧ）の報告をセンタに出力する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Com = 1 & Ret = NR ) -> AX ( Rpt = NG ) )
```

（４）検査結果　：　ｔｒｕｅ

２．１．１２　GWの動作仕様（７）の検査

（７）異常時の区別
　　GWは上記（４）～（６）に示した異常時１～３の状態を区別できないこととする。従って異常時１～３の場合のGWからセンタへの報告は全て同じ報告（「ＮＧ」）とする。

２．１．７、２．１．９、２．１．１１の検査で、センタへの報告（Rpt）が全て同じ報告（「NG」）となっていることを確認する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

２．１．７　：

```
AG ( ( Com = 0 & GW_P = ON & Ord_Dt != EMP ) -> AX( Rpt = NG ) )
```

２．１．９　：

```
AG ( ( Com = 0 & GW_P = ON & Ret = OK ) -> AX ( Rpt = NG ) )
```

２．１．１１：

```
AG ( ( Com = 1 & Ret = NR ) -> AX ( Rpt = NG ) )
```

（４）検査結果　：　ｔｒｕｅ

２．２　ＷＨＭの動作仕様の検査

２．２．１　ＷＨＭの動作仕様（１）－（ａ）－仕様①の検査

（１）指令データの受付及び返送の作成・出力
（ａ）検針指令データの受付
①ＧＷから検針指令データが出力された場合、ＧＷとＷＨＭ間の通信が正常であれば指令データを受け付け、返送を作成しＧＷへ出力する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Ord_Dt = M & Com = 1 ) -> AX ( Ret = OK ) )
```

（４）検査結果　：　ｔｒｕｅ

２．２．２　WHMの動作仕様（１）－（ａ）－仕様②の検査

（１）指令データの受付及び返送の作成・出力
（ａ）検針指令データの受付
②GWから検針指令データが出力された場合、GWとWHM間の通信が異常であれば指令データを受け付けることは出来ず、返送は作成出来ないこととする。

GWとWHM間の通信が異常であれば、次に通信が正常で検針指令データを受け付けるまでは返送を作成しないことを検査する。

（１）付加条件

GWとWHM間の通信はいつか必ず正常となり、その時、GWからWHMに対しては指令データが出力されることとし、WHMはその指令データを受け付けることとする。（メモ参照）

（２）条件付けの方法

SMV ソースコード内で、「FAIRNESS　Com = 1 & Ord_Dt != EMP」を付加する。

（３）ＣＴＬ論理式

```
AG ( ( Ord_Dt = M & Com = 0 ) -> A[ ( Ret = EMP ) U ( Com = 1 & Ord_Dt != EMP ) ] )
```

＜メモ＞　上記ＣＴＬ論理式について

(1) 条件「FAIRNESS Com = 1 & Ord_Dt != EMP」を付加しない場合
　→　前条件が成立してから、永久に Com=1 とならない（GW及びWHMの電源が両方 OFF となる場合等の）パスが存在するため、False となる。
　（ A[φUψ] は　ψが永久に１とならないパスが存在すれば False となる ）

(2) 条件「FAIRNESS Com = 1」を付加した場合
　→　Ord_Dt != EMP の時には Com = 0 となり（それ以外の時に Com = 1 となり）、WHMがいつまでも指令データを受け付けないパスが存在するため、False となる。
　（ A[φUψ] は　ψが永久に１とならないパスが存在すれば False となる ）

（４）検査結果　：　ｔｒｕｅ

２．２．３　WHMの動作仕様（１）－（ｂ）－仕様①の検査

（１）指令データの受付及び返送の作成・出力
（ｂ）パターン設定指令データの受付
①GWからパターン設定指令１データが出力された場合、GWとWHM間の通信が正常であれば指令データを受け付け、返送を作成しGWへ出力する。

本項目は、２．２．１で、検針指令データをパターン設定指令データに置き換えることにより検査することができる。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Ord_Dt = P1 & Com = 1 ) -> AX ( Ret = OK ) )
AG ( ( Ord_Dt = P2 & Com = 1 ) -> AX ( Ret = OK ) )
AG ( ( Ord_Dt = P3 & Com = 1 ) -> AX ( Ret = OK ) )
```

（４）検査結果　：　ｔｒｕｅ

２．２．４　WHMの動作仕様（１）－（ｂ）－仕様②の検査

（１）指令データの受付及び返送の作成・出力
（ｂ）パターン設定指令データの受付
②GWからパターン設定指令１データが出力された場合、GWとWHM間の通信が異常であれば指令データを受け付けることは出来ず、返送を作成することは出来ないこととする。

パターン設定指令１データ及びパターン設定指令３データについも同様とする。

本項目は、２．２．２で、検針指令データをパターン設定指令データに置き換えることにより検査することができる。

GWとWHM間の通信が異常であれば、次に通信が正常で検針指令データを受け付けるまでは返送を作成しないことを検査する。

（１）付加条件

GWとWHM間の通信はいつか必ず正常となり、その時、GWからWHMに対しては指令データが出力されることとし、WHMはその指令データを受け付けることとする。（メモ参照）

（２）条件付けの方法

SMV ソースコード内で、「FAIRNESS　Com = 1 & Ord_Dt != EMP」を付加する。

（３）ＣＴＬ論理式

```
AG ( ( Ord_Dt = P1 & Com = 0 ) -> A[ ( Ret = EMP ) U ( Com = 1 & Ord_Dt != EMP ) ] )
AG ( ( Ord_Dt = P2 & Com = 0 ) -> A[ ( Ret = EMP ) U ( Com = 1 & Ord_Dt != EMP ) ] )
AG ( ( Ord_Dt = P3 & Com = 0 ) -> A[ ( Ret = EMP ) U ( Com = 1 & Ord_Dt != EMP ) ] )
```

（４）検査結果　：　ｔｒｕｅ

２．２．５　WHMの動作仕様（２）の検査

（２）無応答
　　GWから入力した指令データが異常の場合は、何も処理をせず
　　無応答とする。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Ord_Dt = NG & Com = 1 ) -> AX ( Ret = NR ) )
```

（４）検査結果　：　ｔｒｕｅ

２．２．６　WHMの動作仕様（３）－（ａ）－仕様①の検査

（３）パターンデータの蓄積
（ａ）GWとWHM間の通信が正常の場合
GWからパターン設定指令１データ、パターン設定指令２データ、パターン設定指令３データが正常データとして出力された場合、GWとWHM間の通信が正常であれば指令データを受け付け、以下の条件に従ってパターンのデータをメモリ上に蓄積する。

①パターンデータの蓄積は必ずパターン１から開始しなければならない。従って、パターン１のデータを蓄積していない状態で、パターン設定指令２データあるいはパターン設定指令３データを入力した場合は、それらのデータの蓄積は行わない。

パターン１のデータを蓄積していない状態はメモリ上のデータが空データの場合だけである。従ってメモリ上のデータが空データの場合は、パターン１のデータが蓄積されるまでは、空データのままであることを検査する。また本システムでは、全パターンデータのメモリ蓄積が完了した時点でパターン設定指令１データを入力すると、メモリはパターン１のみのデータで上書きされる（３．３－（３）－（ａ）－⑤参照）が、パターン設定指令２データあるいはパターン設定指令３データを入力した場合は、上書きされないことも検査する。

（１）付加条件

ＧＷとＷＨＭ間の通信はいつか必ず正常となり、その時、ＧＷからＷＨＭに対してはパターン設定指令１データが出力されることとし、ＷＨＭはその指令データを受け付けることとする。

（２）条件付けの方法

SMV ソースコード内で、「FAIRNESS　Com = 1 & Ord_Dt = P1」を付加する。

（３）ＣＴＬ論理式

```
AG ( P_Cnt = EMP -> A[ P_Cnt = EMP U ( P_Cnt = P100 ) ] )
AG ( P_Cnt = ALL -> !E[ ( P_Cnt = P120 | P_Cnt = P103 ) U ( P_Cnt = P100 ) ] )
```

（４）検査結果　：　ｔｒｕｅ

２．２．７　WHMの動作仕様（３）－（ａ）－仕様②～④の検査

（３）パターンデータの蓄積

（ａ）GWとWHM間の通信が正常の場合

②パターン１のデータを蓄積後に、パターン設定指令２データあるいはパターン設定指令３データを入力した場合は入力したパターンのデータをパターン１のデータと合わせて、メモリ上に蓄積する。（GWからはパターン設定指令２データ及びパターン設定指令３データは順不同で出力されるためWHM内でも２データと３データの順序は問わない）

③パターン１及びパターン２のデータを蓄積後に、パターン設定指令３データを入力した場合は、パターン３のデータをパターン１及びパターン２のデータと合わせてメモリ上に蓄積し、この時点でメモリ上に全てのパターンデータが揃いメモリ蓄積完了とする。

④パターン１及びパターン３のデータを蓄積後に、パターン設定指令２データを入力した場合は、パターン２のデータをパターン１及びパターン３のデータと合わせてメモリ上に蓄積し、この時点でメモリ上に全てのパターンデータが揃いメモリ蓄積完了とする。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

仕様②：

```
AG ( ( P_Cnt = P100 & Ord_Dt = P2 & Com = 1 ) -> A[ P_Cnt = P100 U P_Cnt = P120 ] )
```

```
AG ( ( P_Cnt = P100 & Ord_Dt = P3 & Com = 1 ) -> A[ P_Cnt = P100 U P_Cnt = P103 ] )
```

仕様③：

```
AG ( ( P_Cnt = P120 & Ord_Dt = P3 & Com = 1 ) -> A[ P_Cnt = P120 U P_Cnt = ALL ] )
```

仕様④：

```
AG ( ( P_Cnt = P103 & Ord_Dt = P2 & Com = 1 ) -> A[ P_Cnt = P103 U P_Cnt = ALL ] )
```

＜メモ＞

A[φUψ]はψが永久に１とならないパスが存在すれば False となる。逆に A[φUψ]が true になった場合は、いつか必ずψが成立し、それまではφであることが証明される。

例）仕様②の場合

A[P_Cnt = P100 U P_Cnt = P120]は、いつか必ず P_Cnt = P120 となり、それまでは P_Cnt = P100 であることを示す。

（４）検査結果　：　ｔｒｕｅ

２．２．８　WHMの動作仕様（３）－（ａ）－仕様⑤の検査

（３）パターンデータの蓄積
（ａ）GWとWHM間の通信が正常の場合
⑤メモリが「空」の状態、あるいは上記②、③、④の後にパターン設定指令１データを入力すると、メモリはパターン１のみのデータで上書きされ、パターン１のデータから蓄積を開始する。

その時のメモリの状態に関わらず、GWとWHM間の通信が正常でパターン１のデータを入力すれば、メモリがパターン１のみのデータで上書きされることを検査する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( Ord_Dt = P1 & Com = 1 ) -> AX ( P_Cnt = P100 ) )
```

（４）検査結果　：　ｔｒｕｅ

２．２．９　WHMの動作仕様（３）－（ａ）－仕様⑥の検査

（３）パターンデータの蓄積
（ａ）GWとWHM間の通信が正常の場合
⑥パターンデータを蓄積中、途中でGWから検針指令データを入力してもパターンデータの蓄積は継続することとする。この場合もGWに対しては検針指令データに対する返送を出力する。

「パターンデータを蓄積中」とは、メモリ上のデータが空データ以外の時である。従って以下のように場合分けして検査を行う。

●全パターンのメモリ蓄積が完了している場合
メモリ上のデータがそのままであるか、あるいは本体へのパターン設定が完了しメモリが空データで上書きされた状態で、GWに対して検針指令データに対する返送を出力することを検査する。

●上記以外（メモリ上のデータが空データの場合は除く）
メモリ上のデータはそのままで、GWに対して検針指令データに対する返送を出力することを検査する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

●全パターンのメモリ蓄積が完了している場合

```
AG ( ( P_Cnt = ALL & Ord_Dt = M & Com = 1 ) ->
    AX ( ( P_Cnt = ALL | ( P_Cnt = EMP & P_Set = ON ) ) & Ret = OK ) )
```

●上記以外（メモリ上のデータが空データの場合は除く）

```
AG ( ( P_Cnt = P100 & Ord_Dt = M & Com = 1 ) -> AX ( P_Cnt = P100 & Ret = OK ) )
AG ( ( P_Cnt = P120 & Ord_Dt = M & Com = 1 ) -> AX ( P_Cnt = P120 & Ret = OK ) )
AG ( ( P_Cnt = P103 & Ord_Dt = M & Com = 1 ) -> AX ( P_Cnt = P103 & Ret = OK ) )
```

（４）検査結果　：　ｔｒｕｅ

２．２．１０　WHMの動作仕様（３）－（ａ）－仕様⑦の検査

（３）パターンデータの蓄積
（ａ）GWとWHM間の通信が正常の場合
⑦メモリ上へのパターンデータ蓄積中にWHMの電源に異常が発生した場合でも、それまで蓄積したパターンデータは保持することができるものとする。

WHMの電源が異常となってもメモリ上のデータの状態が保持されることを検査する。

●全パターンのメモリ蓄積が完了している場合
次に通信が正常となりパターン設定指令データを入力してメモリを上書きするか、あるいは本体への設定が完了するまではメモリ上のデータの状態が保持されることを検査する。

●上記以外
次に通信が正常となりパターン設定指令データを入力してメモリを上書きするまではメモリ上のデータの状態が保持されることを検査する。

（１）付加条件

GWとWHM間の通信はいつか必ず正常となり、その時、GWからWHMに対してはパターン設定指令データが出力されることとし、WHMはその指令データを受け付けることとする。

（２）条件付けの方法

DEFINE 文にて以下のように定義し、

```
DEFINE  Ord_Dt_P := Ord_Dt = P1 | Ord_Dt = P2 | Ord_Dt = P3
```

SMV ソースコード内で、「FAIRNESS　Com = 1 & Ord_Dt_P = 1」を付加する。

（３）ＣＴＬ論理式

●全パターンのメモリ蓄積が完了している場合

```
AG ( ( WHM_P = OFF & P_Cnt = ALL ) ->
        A[ P_Cnt = ALL U ( P_Set = ON | ( Com = 1 & Ord_Dt_P = 1 ) ) ] )
```

●上記以外

```
AG ( ( WHM_P = OFF & P_Cnt = EMP ) ->
        A[ P_Cnt = EMP U ( Com = 1 & Ord_Dt_P = 1 ) ] )
AG ( ( WHM_P = OFF & P_Cnt = P100 ) ->
        A[ P_Cnt = P100 U ( Com = 1 & Ord_Dt_P = 1 ) ] )
AG ( ( WHM_P = OFF & P_Cnt = P120 ) ->
        A[ P_Cnt = P120 U ( Com = 1 & Ord_Dt_P = 1 ) ] )
AG ( ( WHM_P = OFF & P_Cnt = P103 ) ->
        A[ P_Cnt = P103 U ( Com = 1 & Ord_Dt_P = 1 ) ] )
```

（４）検査結果　：　ｔｒｕｅ

２．２．１１　WHMの動作仕様（３）－（ｂ）の検査

（３）パターンデータの蓄積
（ｂ）GWとWHM間の通信が異常の場合
　　GWとWHM間の通信が異常であれば、指令データを受け付けることができずメモリ上への蓄積も出来ないこととする。

GWとWHM間の通信が異常となった場合、メモリ上のデータの状態が変化しないことを検査する。

●全パターンのメモリ蓄積が完了している場合
　次に通信が正常となりパターン設定指令データを入力してメモリを上書きするか、あるいは本体への設定が完了するまではメモリ上のデータの状態が変化しないことを検査する。

●上記以外
　次に通信が正常となりパターン設定指令データを入力してメモリを上書きするまではメモリ上のデータの状態が変化しないことを検査する。

（１）付加条件

GWとWHM間の通信はいつか必ず正常となり、その時、GWからWHMに対してはパターン設定指令データが出力されることとし、WHMはその指令データを受け付けることとする。

（２）条件付けの方法

DEFINE 文にて以下のように定義し、

```
DEFINE  Ord_Dt_P := Ord_Dt = P1 | Ord_Dt = P2 | Ord_Dt = P3
```

SMV ソースコード内で、「FAIRNESS　Com = 1 & Ord_Dt_P = 1」を付加する。

（３）ＣＴＬ論理式

●全パターンのメモリ蓄積が完了している場合

```
AG ( ( Com = 0 & P_Cnt = ALL ) ->
        A[ P_Cnt = ALL U ( P_Set = ON | ( Com = 1 & Ord_Dt_P = 1 ) ) ] )
```

●上記以外

```
AG ( ( Com = 0 & P_Cnt = EMP ) -> A[ P_Cnt = EMP U ( Com = 1 & Ord_Dt_P = 1 ) ] )
AG ( ( Com = 0 & P_Cnt = P100 ) -> A[ P_Cnt = P100 U ( Com = 1 & Ord_Dt_P = 1 ) ] )
AG ( ( Com = 0 & P_Cnt = P120 ) -> A[ P_Cnt = P120 U ( Com = 1 & Ord_Dt_P = 1 ) ] )
AG ( ( Com = 0 & P_Cnt = P103 ) -> A[ P_Cnt = P103 U ( Com = 1 & Ord_Dt_P = 1 ) ] )
```

（４）検査結果　：　ｔｒｕｅ

２．２．１２　WHMの動作仕様（４）－（ａ）－仕様①の検査

（４）パターンデータの設定
（ａ）WHMの電源が正常の場合
　全てのパターンデータのメモリ蓄積が完了した場合、WHMの電源が正常の場合は、以下の条件に従って処理を行う。
　　①全パターンデータのメモリ蓄積が完了した時点で、本体にパターンデータを設定する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( P_Cnt = ALL & WHM_P = ON ) -> AX ( P_Set = ON ) )
```

（４）検査結果　：　ｔｒｕｅ

２．２．１３　WHMの動作仕様（４）－（ａ）－仕様②の検査

（４）パターンデータの設定
（ａ）WHMの電源が正常の場合
全てのパターンデータのメモリ蓄積が完了した場合、WHMの電源が正常の場合は、以下の条件に従って処理を行う。
②本体へのパターンデータの設定が完了した時点で、メモリは「空データ」で上書きされる。（ただし空データの上書きと同時に、GWからパターン設定指令１データを入力した場合は、パターン１の処理を選択し、メモリはパターン１のみのデータで上書きすることとする。）

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

```
AG ( ( !( Com = 1 & Ord_Dt = P1 ) & P_Cnt = ALL & WHM_P = ON ) ->
                                                    AX ( P_Cnt = EMP ) )
```

注）P_Cnt = EMP　は P_Set = ON と同時に起こるので前条件に
P_Set = ON は不要

（４）検査結果　：　ｔｒｕｅ

（４）－（ａ）－仕様②は検査によって不具合が発見されたため、（ただし～）の部分を追記（仕様変更）し、上記のような検査を行うことにより「ｔｒｕｅ」を出力することができた。以下に、その経緯を示す。

＜仕様変更の経緯＞

仕様変更をする以前は、（ただし～）の部分が存在しないため、以下のようなＣＴＬ論理式とした。

```
AG ( ( P_Cnt = ALL & WHM_P = ON ) -> AX ( P_Cnt = EMP ) )
```

上記ＣＴＬ論理式での検査結果はｆａｌｓｅとなった。反例として以下のようなパスが出力された。

反例：全てのパターンデータのメモリ蓄積が完了し、かつ本体へのパターンデータの設定が完了し、かつWHMの電源が正常の状態で、WHMがGWからパターン設定指令１データを入力したため、メモリは空データで上書きされず、パターン１のみのデータで上書きされた。

仕様書中の次の２つの仕様がこの反例に関わる。

（３）－⑤
：メモリが「空データ」の状態、あるいは上記②、③、④の後にパターン設定指令１データを入力すると、メモリはパターン１のみのデータで上書きされ、パターン１のデータから蓄積を開始する。
（４）－（ａ）－②
：本体へのパターンデータの設定が完了した時点で、メモリは「空データ」で上書きされる。

反例の理由は次のように分析できる。上の 2 つの仕様は互いに独立したものであり、仕様書の上では同時に起こることが可能である。しかし、実際には同時に起こってはならないことがらであり、仕様書作成時に見落とされて独立した記述になっていた。状態遷移図作成時には、これに気付か

ず、2 つの仕様が独立した関係ではなく、仕様（４）－（ａ）－②よりも（３）－⑤が必ず優先されるモデルにしてしまった。その結果、パターン設定が完了したにも関わらずメモリが空データで上書きされず一方の仕様が満たされない反例が生じた。これは 2 つの仕様が両立せず矛盾する不具合の例である.

この仕様の矛盾を解消するために、２つの仕様が同時に動作するような状況では、（３）－⑤の動作仕様が優先されることとした。解消策は、以下のように太字の部分を追記して、システム仕様書に反映した。

（３）－⑤

：メモリが「空データ」の状態、あるいは上記②、③、④の後にパターン設定指令１データを入力すると、メモリはパターン１のみのデータで上書きされ、パターン１のデータから蓄積を開始する。**（ただし空データの上書きと同時に、GWからパターン設定指令１データを入力した場合は、パターン１の処理を選択し、メモリはパターン１のみのデータで上書きすることとする。）**

仕様変更後の本検査項目のＣＴＬ論理式は次のように修正した。

```
AG ( ( I( Com = 1 & Ord_Dt = P1 ) & P_Cnt = ALL & WHM_P = ON) ->
                                        AX ( P_Cnt = EMP ) )
```

２．２．１４　WHMの動作仕様（４）－（ｂ）の検査

（４）パターンデータの設定
（ｂ）WHMの電源が異常の場合
　　全パターンデータのメモリ蓄積が完了した場合でも、WHMの電源が異常の場合は本体への設定は出来ないこととする。

全パターンデータのメモリ蓄積が完了した場合でも、次にWHMの電源が正常となるまでは、本体への設定が完了しないことを検査する。

（１）付加条件

いつか必ずWHMの電源が正常となることが必要である。

（２）条件付けの方法

SMV ソースコード内で、「FAIRNESS WHM_P = ON」を付加する。

（３）ＣＴＬ論理式

```
AG ( ( P_Cnt = ALL & P_Set = OFF & WHM_P = OFF ) ->
                        A[ ( P_Set = OFF ) U ( WHM_P = ON ) ] )
```

（４）検査結果　：　ｔｒｕｅ

２．３　システム全体の検査

２．３．１　検査項目１（WHM内でのパターン設定完了の確認１）

（ａ）パターン設定完了の確認
センタ～GW～WHM間の通信が正常に動作し、WHM内でのデータ設定も正常に動作すれば、自動検針システムの重要な動作の１つであるパターンデータの設定が完了する。従って、「センタから要求が出力されれば、最終的にWHM内でのパターン設定が完了する」ことを検証する。

以下に示す付加条件のもとでは、WHM内で将来のある時点で必ずパターン設定が完了することを検査する。

（１）付加条件

（ａ）付加条件１
WHMの電源及びGWの電源が常に正常である。

（ｂ）付加条件２
GWから出力するパターン設定データは常に正常である。

（ｃ）付加条件３
センタからのパターン設定要求が、少なくとも１回は以下に示す順序のどちらかの順序で出力される。
●パターン設定要求１　→　２　→　３
●パターン設定要求１　→　３　→　２

（２）条件付けの方法

（ａ）付加条件１

SMV ソースコード内で、常に WHM_P=ON、GW_P=ON とする。

（ｂ）付加条件２

SMV ソースコード内で、「FAIRNESS　Req_Hty=ALL」を付加する。

（ｃ）付加条件３

SMV ソースコード内で、WHM へ出力する指令データ（Ord_Dt）を常に以下のように設定する。

●パターン設定指令１データ：Ord_Dt = P1
●パターン設定指令２データ：Ord_Dt = P2
●パターン設定指令３データ：Ord_Dt = P3

上記記述は、常に指令データが異常（Ord_Dt=NG）とならないことを示す。

（３）ＣＴＬ論理式

将来のある時点で必ず、パターン設定完了フラグ（P_Set）が ON となる。

```
AG ( AF ( P_Set = ON ) )
```

（４）検査結果　：　ｔｒｕｅ

２．３．２　検査項目２（WHM内でのパターン設定完了の確認２）

検査項目２では、上記検査項目１から付加条件「(b)パターン設定指令データは常に正常である」と付加条件「(c)センタからの要求が、少なくとも１回は以下に示す順序のどちらかの順序で出力される。」を削除して検査を行う。ただし、本検査を行うために新条件(b)を追加する。（本検査でも付加条件(c)は必要であるが、この条件は新条件(b)に含まれるため）

すなわち、以下に示す付加条件のもとでは、WHM内で将来のある時点で必ずパターン設定が完了することを検査する。

（１）付加条件

（ａ）付加条件１

WHMの電源及びGWの電源が常に正常である。

（ｂ）付加条件２

GWから出力するパターン設定指令データは正常の場合と異常の場合があるが、少なくとも１回は以下に示す順序のどちらかの順序で正常な指令データが出力される。

●パターン設定指令１データ　→　２データ　→　３データ

●パターン設定指令１データ　→　３データ　→　２データ

ここで、上記順序で３つ連続して正常な指令データとなる必要はなく、途中で異常データが混在してもよい。

例１）

１データ正常→２データ異常→**３データ正常**→**２データ正常**

例２）

１データ正常→**２データ正常**→１データ異常→３データ異常→**３データ正常**

例３）

３データ正常→２データ異常→**１データ正常**→**２データ正常**→１データ異常→**３データ正常**

（２）条件付けの方法

（ａ）付加条件１

SMV ソースコード内で、常に WHM_P=ON、GW_P=ON とする。

（ｂ）付加条件２

SMV ソースコード内で、「FAIRNESS　Dt_Hty=ALL」を付加する。

（３）ＣＴＬ論理式

将来のある時点で必ず、パターン設定完了フラグ（P_Set）が ON となる。

```
AG ( AF ( P_Set = ON ) )
```

（４）検査結果　：　ｔｒｕｅ

２．３．３　検査項目３（初期状態への再帰可能性の確認）

（ｂ）初期状態への再帰可能性の確認
「システムが初期状態から動作して、いかなる状態からでも再び初期状態へ戻ることができる」ことを検査する。

システムがいかなる状態であっても、初期状態へ戻ることができることを検査する。

（１）付加条件　：　なし

（２）条件付けの方法　：　同上

（３）ＣＴＬ論理式

初期状態とは、全ての変数の値が初期値に戻った状態である。ただし、Req = OFF で Wt_Cnt = 0 となることはないので、Wt_Cnt は除く。

```
AG ( EF ( GW_P = ON & WHM_P = ON & Req = OFF & Ord_Dt = EMP & Ret = EMP &
        Rpt = EMP & P_Cnt = EMP & P_Set = OFF & Req_Hty = EMP & Dt_Hty = EMP ) )
```

（４）検査結果　：　ｔｒｕｅ

本検査では、センタの動作仕様に不具合が発見されたため、センタの待機状態のリセット処理を追加（仕様変更）し、上記のような検査を行うことにより「ｔｒｕｅ」を出力することができた。以下に、その経緯を示す。

＜仕様変更の経緯＞

仕様変更をする以前は、検査結果はｆａｌｓｅとなった。反例として以下のようなパスが出力された。

反例：センタから要求が出力されたが、GWの電源が異常でありその要求を入力出来ない。またGWは電源異常のため、要求を入力できなかった旨を報告としてセンタに出力することができない。そのためセンタはGWからの報告を待ち続ける「待機状態」を維持し続け初期状態である「要求無し」の状態に戻れず、システム全体としても初期状態に戻ることができない。

センタの動作仕様は検査の対象外であったが、この反例によって図らずも、「センタの動作仕様」の次の仕様が記述不足であることがわかった。

（４）　：待機中にGWから報告が出力された場合は、センタとGW間の通信が正常であれば報告を受け付け、待機状態を抜け次の要求をGWに対して出力する。

反例のパスを経ると初期状態に戻ることができないことが発見できた。これは、センタが待機状態のまま“フリーズ”してしまい、システム全体がフリーズするパスである。原因は、センタが待機中にGWから報告が出力されなかった場合の動作仕様を記述していないためである。すなわち仕様書の記述不足が原因であった。これを解消するために、センタが待機状態となった場合は、GWから報告が出力されなくても、タイムアウトにより待機状態を抜けることとした。解消策は、以下のように太字の部分を追記して、システム仕様書に反映した。

（４）　：　待機中にGWから報告が出力された場合は、センタとGW間の通信が正常であれば報告を受け付け、待機状態を抜け

次の要求をＧＷに対して出力する。**また、センタは待機状態となった時点でカウントを開始し、ＧＷから報告を入力できずカウンタが「７」となった場合は待機状態を抜けることとする。（これは、待機状態のままフリーズする状態を回避するためである）**

３．まとめ

検査の結果、自動検針システムのシステム仕様書に２つの不具合があることがわかったが、それらを改修し、再度検査を行うことで、全ての検査項目で「ｔｒｕｅ」を出力することができた。