4. Requirement LTL:

<<Introduction for LTL requirement> >

- There is Always an exit from any screen.

$$\bigwedge_i \Diamond\, (state = screen_i)$$

- There is a screen (root), such that each screen is reached from it.

$$\bigvee_i \left[\bigwedge_j \Box \left((state = screen_i) \rightarrow \Diamond\, (state = screen_j) \right) \right]$$

- We can't move from $screen_j$ to $screen_i$ without changing or defining a parameter.

$$(\neg screen_j) \bigcup \left[\neg \left(screen_j \rightarrow (\neg ChangeParmX \; U \; Screen_i) \right) \right]$$

- Parameter cannot accept value that is not defined in the List of the possible values.

$$\bigwedge_i (P_i = ON \;||\; P_i = OFF \;||\; P_i = \;||\; P_i = defined \;||\; P_i = undefined \;||\; P_i = L1_1 \;||\; P_i = L2 \;||\; P_i = L3)$$

- There is no path to a screen that allows "Illegal parameters values".
  (Illegal i.e. value that does not defined in the list of the parameters values)

$$\bigwedge_j \left[(scree_j) \rightarrow \bigwedge_i (P_{ji} = ON \;||\; P_{ji} = OFF \;||\; P_{ji} = defined \;||\; P_{ji} = undefined \;||\; P_{ji} = L1 \;||\; P_{ji} = L2 \;||\; P_{ji} = L3) \right]$$

- Each list of parameters must be defined before entering a screen.

- Parameters values cannot change unless it was intended to do so in its path.

$\Box\, (\neg ChangeParmX \; U \; ChangeParmX)$

- If a Parameter changes in a specific state the change should be updated wherever the parameter is used.

- All parameters always must be consistent.

3.4 Requirements in LTL:

In this section, we define a set of general requirements that must be true throughout the application run. In order to translate this requirements to ltl we need to define:

States = { $screen_1$,.. $screen_m$, $ChangeParmX_1$, .., $ChangeParmX_m$}

Parameters name - X_1, X_2 .... X_k values Type=" On/Off"

Parameters name - Y_1, ... Y_k values Type=" Empty/NotEmpty"

Parameters name - Z_1,... Z_k values Type=" List"

Parameters Values={ON,OFF, $Empty$, $NotEmpty$, L1,...,Ln – parameters of specific list element }

AP a set of conditions such as $(X_1 = on)$, $(X_1 = Off)$, $(Y_i = Empty)$, $(Z_4 = L3)$ ...

1. There is always an exit from any screen.
   $\bigwedge_m \Diamond\,(state = screen_m)$
2. Reachability
   2.1. There is a screen (root), such that each screen is reachable from it:
   $$\bigvee_i \left[\bigwedge_j \Box\left((state = screen_i) \rightarrow \Diamond\,(state = screen_j)\right) \right]$$
   2.2. Each screen is reachable from $screen_i$:
   $\Box((state = screen_i) \rightarrow \wedge_j \left(\Diamond\,(state = screen_j)\right))$
3. We cannot move from $screen_j$ to $screen_i$ without changing or defining a parameter.
   $$\neg(state = screen_j) \bigcup \left[\neg\left((state = screen_j) \rightarrow \left((state = ChangeParmX)\ U\ (state = screen_i)\right)\right)\right]$$
4. Parameter cannot accept value that is not defined in the List of the possible values.
   4.1. $\Box\ [(X_k = On \vee X_k = Off\,) \rightarrow \neg\ \Diamond\ ((X_k \neq ON \wedge (X_k \neq OFF))\ ]$
   4.2. $\Box\ [(Y_i = Empty \vee Y_i = NotEmpty\,) \rightarrow \neg\ \Diamond\ ((Y_i \neq Empty \wedge (Y_i \neq NotEmpty\,))\ ]$
   4.3. $\Box\ [(Z_j = L1 \vee \ldots \vee Z_j = Ln\,) \rightarrow \neg\ \Diamond\ ((X_j \neq L1 \wedge \ldots \wedge (X_j \neq Ln\,))\ ]$
5. There is no path to a screen that allows "Illegal parameters values".
   (Illegal i.e. value that does not defined in the list of the parameters values)

$$\bigwedge_j \Big[(screen_j) \rightarrow \bigwedge_i (P_{ji} = ON \,||\, P_{ji} = OFF \,||\, P_{ji} = defined \,||\, P_{ji} = undefined \,||\, P_{ji} = L1 \,||\, P_{ji} = L2 \,||\, P_{ji} = L3)\Big]$$

6. Each list of parameters must be defined before entering a screen.
7. Parameters values cannot change unless it was intended to do so in its path.
   $\Box\ (\neg ChangeParmX\ U\ ChangeParmX\,)$

Comment [K23.069]: use ¬ instead of ≠ everywhere

For each param P_i which is changed on action of element X the following property has to be checked:

If a Parameter changes in a specific state the change should be updated wherever the parameter is used.

8. All parameters always must be consistent

$(state = screen_m \wedge X_k = On) \rightarrow \wedge_j (\text{state} = \text{screen}_j \wedge X_k = On)$

$(state = screen_m \wedge Y_i = NotEmpty) \rightarrow \wedge_j (\text{state} = \text{screen}_j \wedge Y_i = On\,NotEmpty)$

$(state = screen_m \wedge Z_j = L1) \rightarrow \wedge_j (\text{state} = \text{screen}_j \wedge Z_j = L1)$

3.5 Verification process

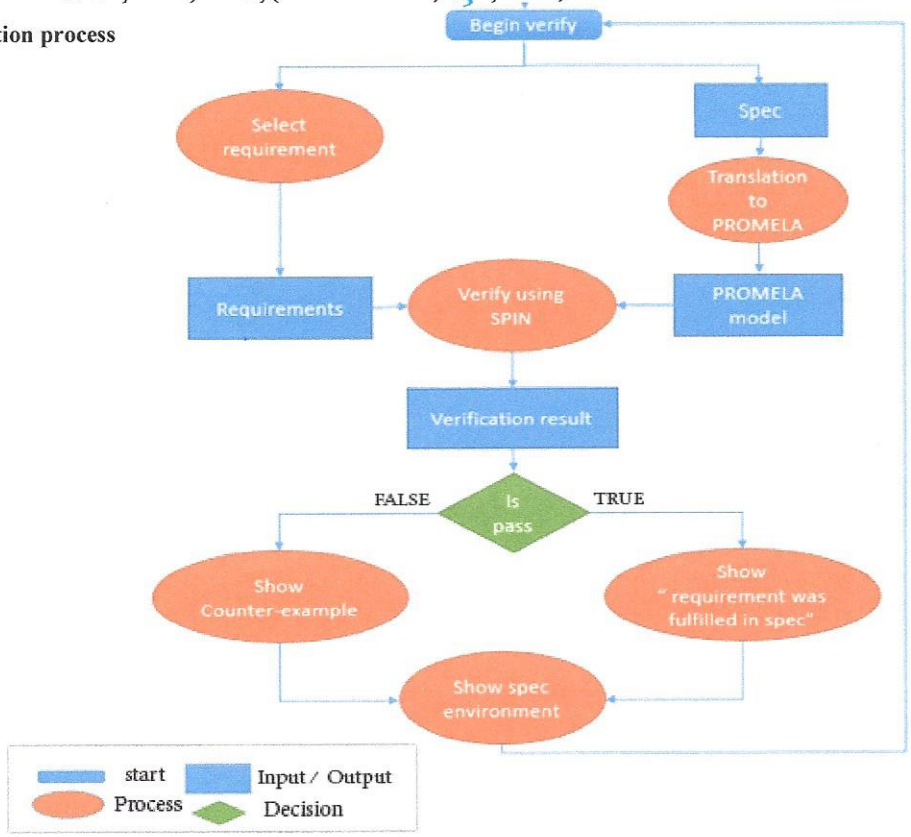

Fig 5: Flowchart: the verification process

After the user built his spec he presses "Verify Spec", a new window will pop, user chooses from this window a screen as a Root and a list of requirements that must be correct in his spec then he presses "Run" The verification process.

The verification ~~process start~~process starts with translating the spec to Program Graph that will be written in PROMELA language, and save it in a file.

~~the~~The tool ~~Create~~ creates a script file that run the PROMELA with every LTL requirement in SPIN and ~~Save~~ saves the verification results in a new file. Then, the user presses "~~show~~ Show ~~verification~~ Verification Result" to show the result.

4. EXPECTED RESULTS

$\rightarrow [\bigwedge_{Type(P_{ji})=On/Of} (P_{ji}=ON \;||\; P_{ji}=OFF) \wedge$

$\bigwedge_{Type(P_{ji})=Emp/N.} (P_{ji}=Empty \;||\; P_{ji}=NowEm) \wedge$

$\bigwedge_{Type(P_{ji})=Lis} (P_{ji}=L1 \;||\; \ldots \;||\; P_{ji}=Ln)$

For each ~~type of~~ par:

5.1 If $Type(P_{ji})=on/OF$:

$\bigwedge_j ((state=s_j) \rightarrow (P_{ji}=ON) \vee (P_{ji}=OFF))$

5.2 If Type( ) ...

(8)

For an action

$$\{P_1 = val_1, \ldots P_k = val_k\}$$

on element X the following property has to be checked:

$$\Box\left(\bigwedge_i \left((P_i = val_i) \rightarrow \bigwedge_{j \neq i} (P_j = val_j)\right)\right)$$