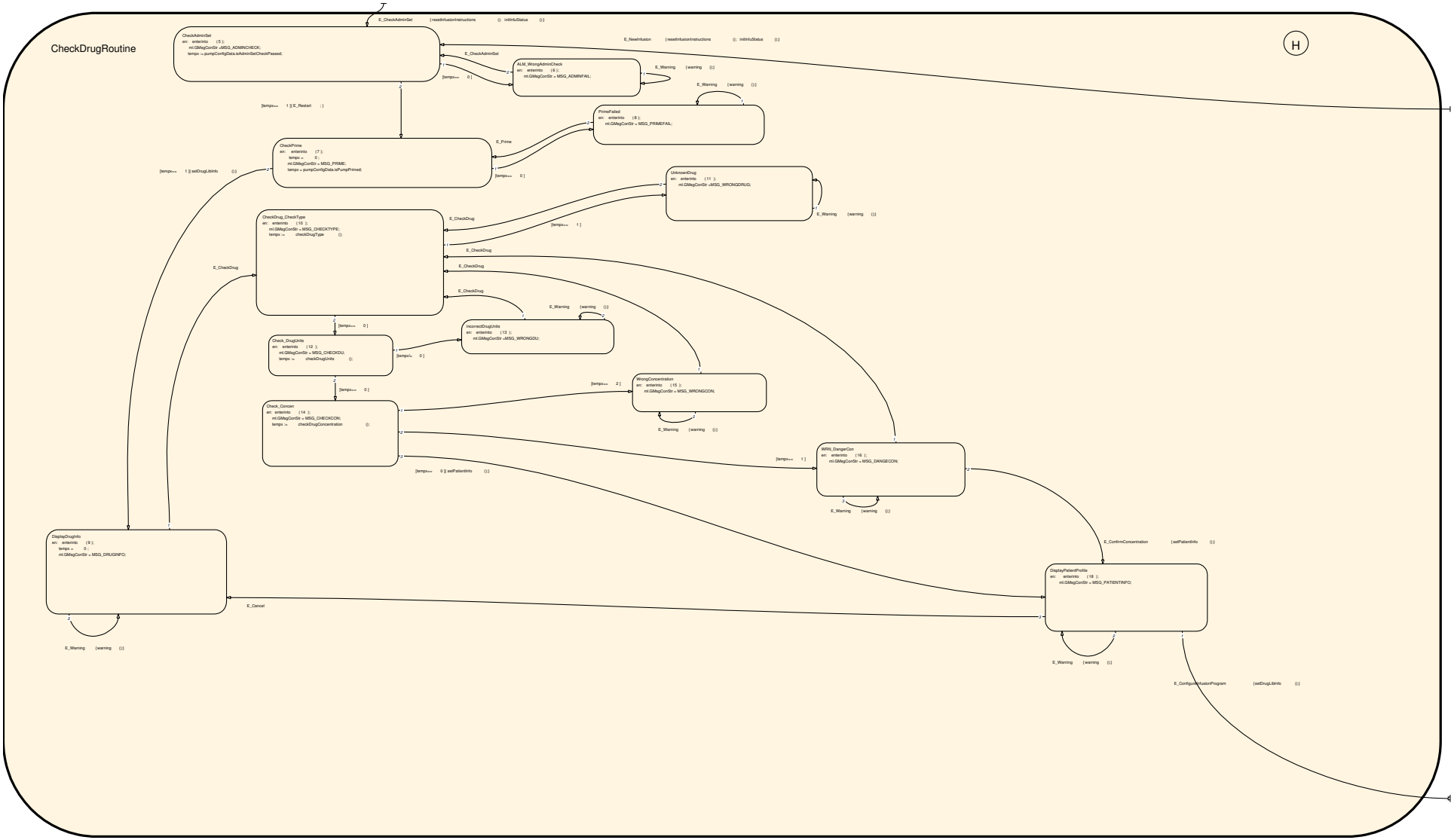
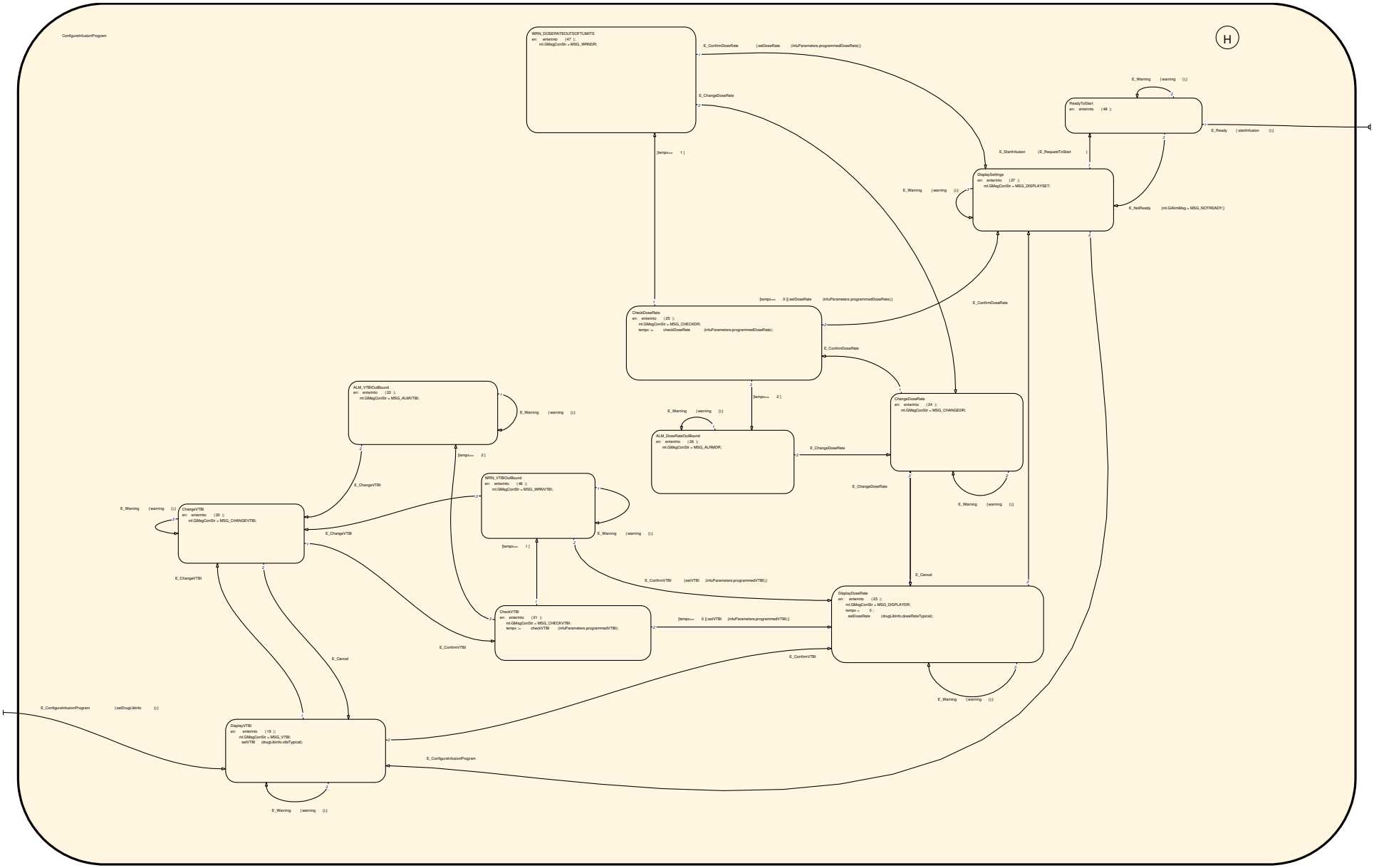
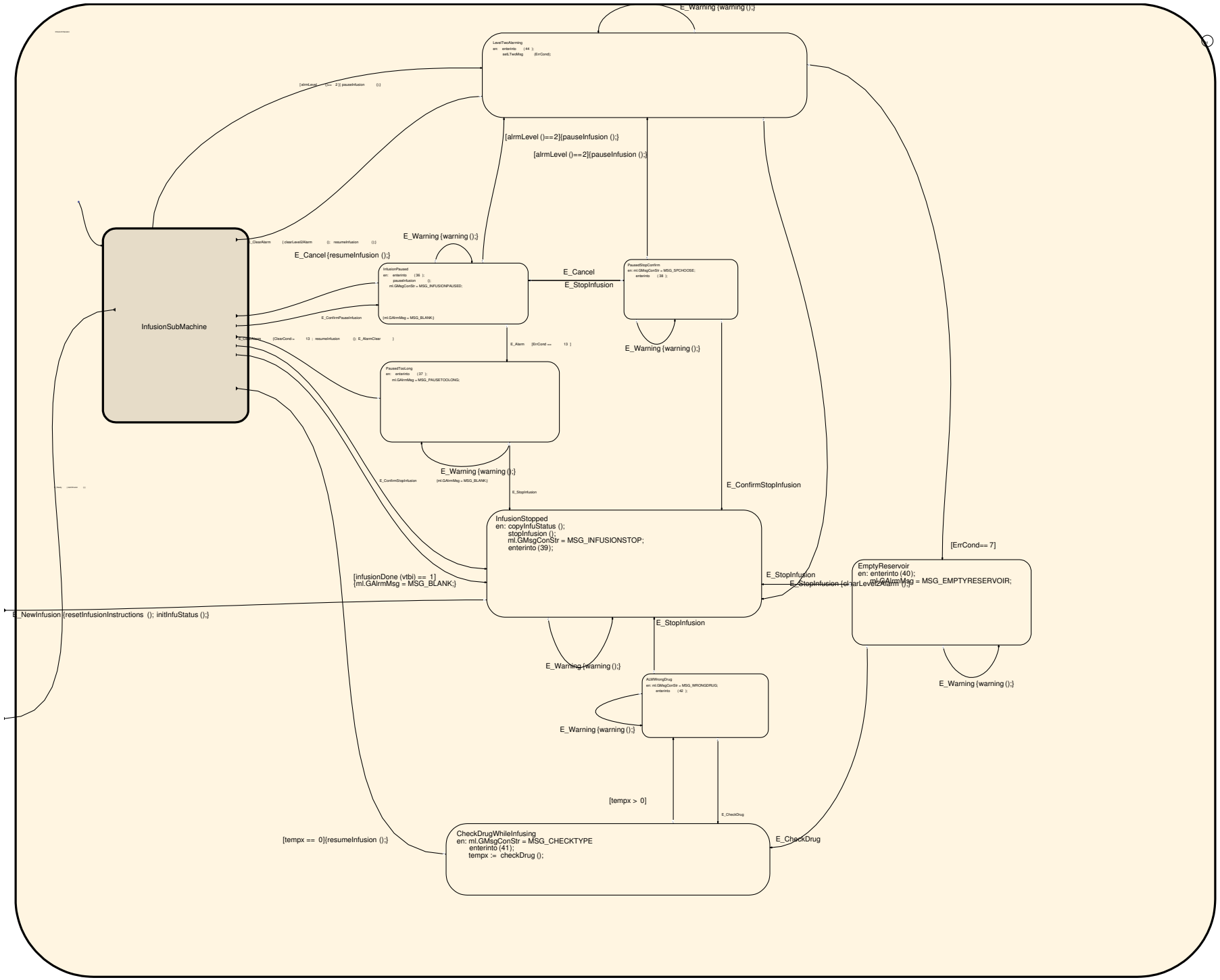
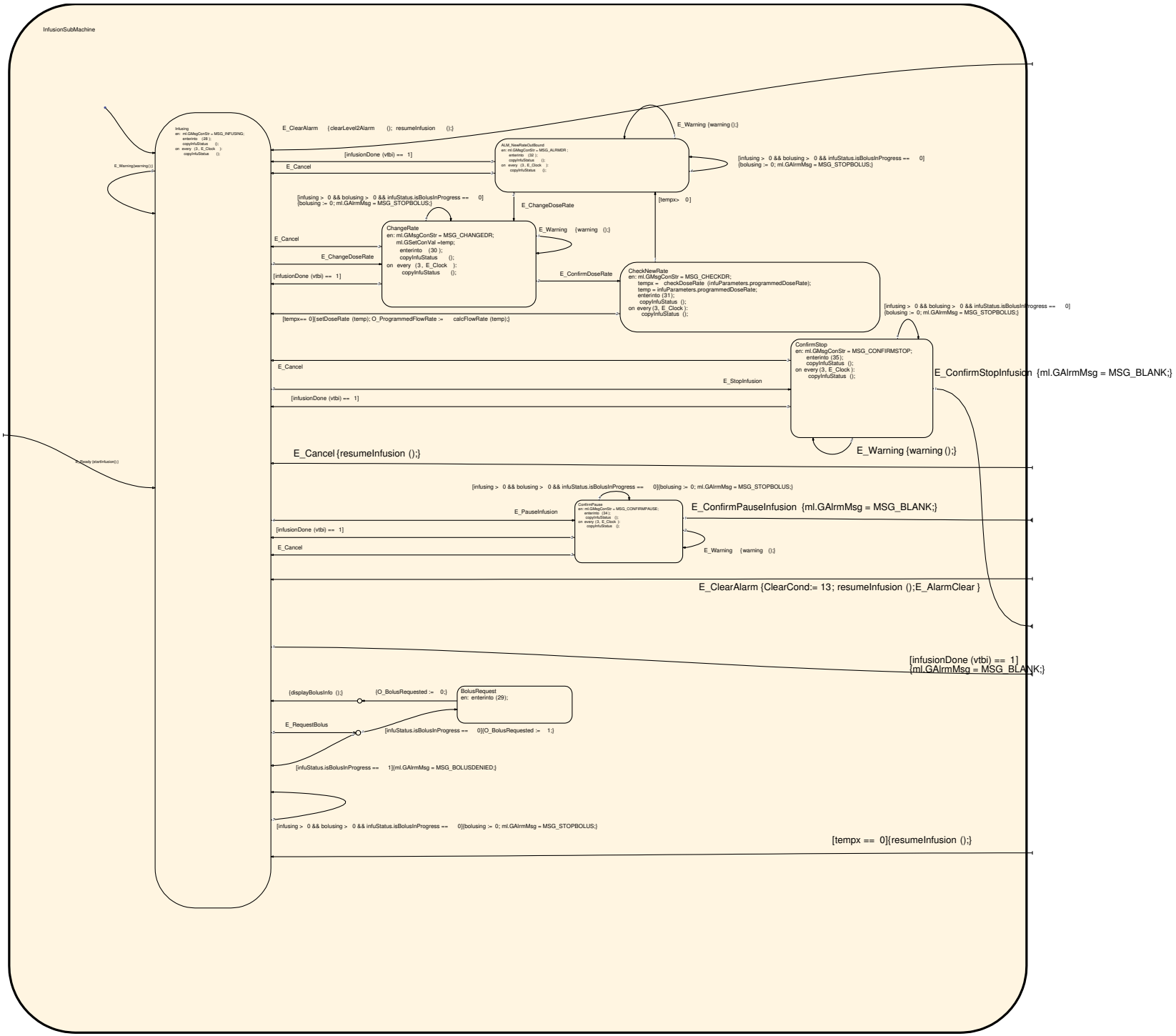
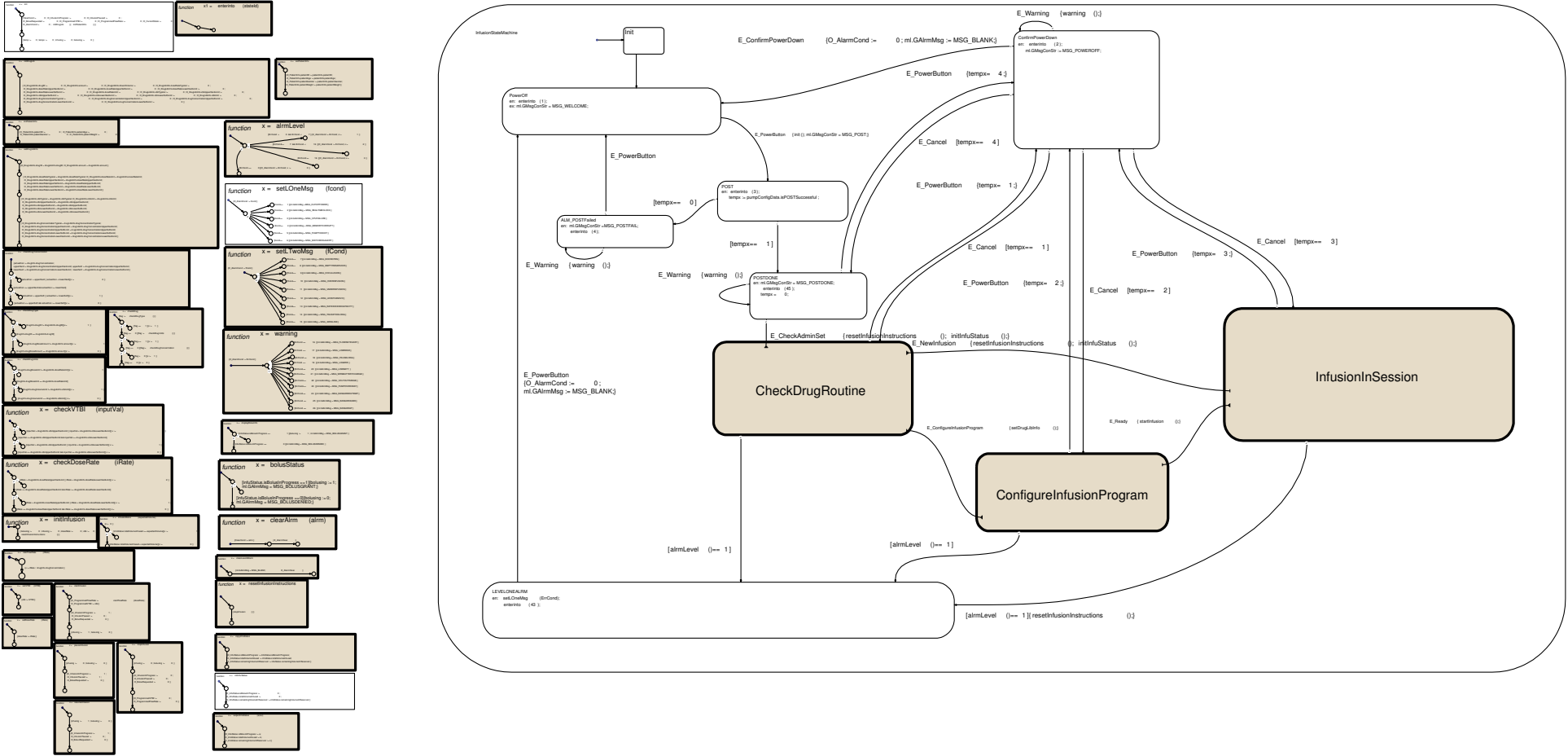

CheckDrugRoutine
H
E_CheckAdminSet {resetInfusionInstructions (); initInfuStatus ()}
CheckAdminSet
en: enterInto (5);
ml.GMagConStr =MSG_ADMINCHECK;
tempx := pumpConfigData.isAdminSetCheckPassed;
E_NewInfusion {resetInfusionInstructions (); initInfuStatus ()}
E_CheckAdminSet
[tempx== 0]
ALM_WrongAdminCheck
en: enterInto (6);
ml.GMagConStr = MSG_ADMINFAIL;
E_Warning {warning ()}
E_Warning {warning ()}
[tempx== 1]|| E_Restart ;]
PrimeFailed
en: enterInto (8);
ml.GMagConStr = MSG_PRIMEFAIL;
E_Prime
CheckPrime
en: enterInto (7);
tempx := 0 ;
ml.GMagConStr = MSG_PRIME;
tempx = pumpConfigData.isPumpPrimed;
[tempx== 0]
[tempx== 1]|| setDrugLibInfo ()]
UnknownDrug
en: enterInto (11);
ml.GMagConStr =MSG_WRONGDRUG;
E_Warning {warning ()}
CheckDrug_CheckType
en: enterInto (10);
ml.GMagConStr = MSG_CHECKTYPE;
tempx := checkDrugType ();
E_CheckDrug
[tempx== 1]
E_CheckDrug
E_CheckDrug
E_CheckDrug
E_CheckDrug
E_CheckDrug
E_Warning {warning ()}
[tempx== 0]
Check_DrugUnits
en: enterInto (12);
ml.GMagConStr = MSG_CHECKDU;
tempx := checkDrugUnits ();
[tempx!= 0]
IncorrectDrugUnits
en: enterInto (13);
ml.GMagConStr =MSG_WRONGDU;
[tempx== 0]
Check_Concen
en: enterInto (14);
ml.GMagConStr = MSG_CHECKCON;
tempx := checkDrugConcentration ();
[tempx== 2]
WrongConcentration
en: enterInto (15);
ml.GMagConStr = MSG_WRONGCON;
E_Warning {warning ()}
[tempx== 0]|| setPatientInfo ()]
[tempx== 1]
WRN_DangerCon
en: enterInto (16);
ml.GMagConStr = MSG_DANGECON;
E_Warning {warning ()}
E_ConfirmConcentration {setPatientInfo ()}
DisplayDrugInfo
en: enterInto (9);
tempx := 0 ;
ml.GMagConStr = MSG_DRUGINFO;
E_Cancel
E_Warning {warning ()}
DisplayPatientProfile
en: enterInto (18);
ml.GMagConStr = MSG_PATIENTINFO;
E_Warning {warning ()}
E_ConfigureInfusionProgram {setDrugLibInfo ()}


ConfigureInfusionProgram
H
WRN_DOSERATEOUTSOFTLIMITS
en: enterinto (47);
ml.GMagConSir = MSG_WRNDR;
E_ConfirmDoseRate {setDoseRate (infuParameters.programmedDoseRate);}
E_ChangeDoseRate
E_Warning {warning ();}
ReadyToStart
en: enterinto (48);
E_Ready {startInfusion ();}
E_StartInfusion {E_RequestToStart }
DisplaySettings
en: enterinto (27);
ml.GMagConSir = MSG_DISPLAYSET;
E_Warning {warning ();}
E_NotReady {ml.GAlrmMag = MSG_NOTREADY;}
[tempo== 1]
[tempo== 0]{ setDoseRate (infuParameters.programmedDoseRate);}
E_ConfirmDoseRate
CheckDoseRate
en: enterinto (25);
ml.GMagConSir = MSG_CHECKDR;
tempo := checkDoseRate (infuParameters.programmedDoseRate);
E_ConfirmDoseRate
[tempo== 2]
ALM_VTBIOutBound
en: enterinto (22);
ml.GMagConSir = MSG_ALMVTBI;
E_Warning {warning ();}
E_Warning {warning ();}
ALM_DoseRateOutBound
en: enterinto (26);
ml.GMagConSir = MSG_ALRMDR;
E_ChangeDoseRate
ChangeDoseRate
en: enterinto (24);
ml.GMagConSir = MSG_CHANGEDR;
E_ChangeDoseRate
E_Warning {warning ();}
[tempo== 2]
E_ChangeVTBI
WRN_VTBIOutBound
en: enterinto (46);
ml.GMagConSir = MSG_WRNVTBI;
E_Warning {warning ();}
E_Warning {warning ();}
ChangeVTBI
en: enterinto (20);
ml.GMagConSir = MSG_CHANGEVTBI;
E_ChangeVTBI
E_ChangeVTBI
[tempo== 1]
E_ConfirmVTBI {setVTBI (infuParameters.programmedVTBI);}
E_Cancel
DisplayDoseRate
en: enterinto (23);
ml.GMagConSir = MSG_DISPLAYDR;
tempo := 0 ;
setDoseRate (drugLibInfo.doseRateTypical);
CheckVTBI
en: enterinto (21);
ml.GMagConSir = MSG_CHECKVTBI;
tempo := checkVTBI (infuParameters.programmedVTBI);
[tempo== 0]{ setVTBI (infuParameters.programmedVTBI);}
E_ConfirmVTBI
E_Cancel
E_ConfirmVTBI
E_Warning {warning ();}
E_ConfigureInfusionProgram {setDrugLibInfo ();}
DisplayVTBI
en: enterinto (19);
ml.GMagConSir = MSG_VTBI;
setVTBI (drugLibInfo.vtbiTypical);
E_ConfigureInfusionProgram
E_Warning {warning ();}


E_Warning {warning ();}
LevelTwoAlarming
en: enterinto (44);
setLTwoMsg (ErrCond);
{ alrmLevel () == 2 }{ pauseInfusion ();}
[alrmLevel ()==2]{pauseInfusion ();}
[alrmLevel ()==2]{pauseInfusion ();}
InfusionSubMachine
E_ClearAlarm { clearLevel2Alarm (); resumeInfusion ();}
E_Warning {warning ();}
E_Cancel {resumeInfusion ();}
InfusionPaused
en: enterinto (36);
pauseInfusion ();
ml.GMsgConStr = MSG_INFUSIONPAUSED;
ml.GAlrmMsg = MSG_BLANK;
E_Cancel
E_StopInfusion
PausedStopConfirm
en: ml.GMsgConStr = MSG_SPCHOOSE;
enterinto (38);
E_ConfirmPauseInfusion
E_ClearAlarm {ClearCond == 13 ; resumeInfusion (); E_AlarmClear }
E_Alarm [ErrCond == 13]
E_Warning {warning ();}
PausedTooLong
en: enterinto (37);
ml.GAlrmMsg = MSG_PAUSETOOLONG;
E_Warning {warning ();}
E_ConfirmStopInfusion
{ml.GAlrmMsg = MSG_BLANK;}
E_StopInfusion
E_ConfirmStopInfusion
InfusionStopped
en: copyInfuStatus ();
stopInfusion ();
ml.GMsgConStr = MSG_INFUSIONSTOP;
enterinto (39);
[infusionDone (vtbi) == 1]
{ml.GAlrmMsg = MSG_BLANK;}
E_StopInfusion
E_StopInfusion {clearLevel2Alarm ();}
[ErrCond== 7]
EmptyReservoir
en: enterinto (40);
ml.GAlrmMsg = MSG_EMPTYRESERVOIR;
E_Warning {warning ();}
E_NewInfusion {resetInfusionInstructions (); initInfuStatus ();}
E_StopInfusion
E_Warning {warning ();}
ALMWrongDrug
en: ml.GMsgConStr = MSG_WRONGDRUG;
enterinto (42);
E_Warning {warning ();}
[tempx > 0]
E_CheckDrug
[tempx == 0]{resumeInfusion ();}
CheckDrugWhileInfusing
en: ml.GMsgConStr = MSG_CHECKTYPE
enterinto (41);
tempx := checkDrug ();
E_CheckDrug


InfusionSubMachine
Infusing
en: ml.GMsgConStr = MSG_INFUSING;
enterinto (28);
copyInfuStatus ();
on every (3, E_Clock);
copyInfuStatus ();
E_Warning{warning();}
E_ClearAlarm {clearLevel2Alarm ();; resumeInfusion ();}
[infusionDone (vtbi) == 1]
E_Cancel
ALM_NewRateOutBound
en: ml.GMsgConStr = MSG_ALRMDR ;
enterinto (32);
copyInfuStatus ();
on every (3, E_Clock);
copyInfuStatus ();
E_Warning {warning ();}
[infusing > 0 && bolusing > 0 && infuStatus.isBolusInProgress == 0]
{bolusing := 0; ml.GAlrmMsg = MSG_STOPBOLUS;}
[infusing > 0 && bolusing > 0 && infuStatus.isBolusInProgress == 0]
{bolusing := 0; ml.GAlrmMsg = MSG_STOPBOLUS;}
E_ChangeDoseRate
[tempx> 0]
ChangeRate
en: ml.GMsgConStr = MSG_CHANGEDR;
ml.GSetConVal =temp;
enterinto (30);
copyInfuStatus ();
on every (3, E_Clock);
copyInfuStatus ();
E_Warning {warning ();}
E_Cancel
E_ChangeDoseRate
[infusionDone (vtbi) == 1]
E_ConfirmDoseRate
CheckNewRate
en: ml.GMsgConStr = MSG_CHECKDR;
tempx = checkDoseRate (infuParameters.programmedDoseRate);
temp = infuParameters.programmedDoseRate;
enterinto (31);
copyInfuStatus ();
on every (3, E_Clock);
copyInfuStatus ();
[infusing > 0 && bolusing > 0 && infuStatus.isBolusInProgress == 0]
{bolusing := 0; ml.GAlrmMsg = MSG_STOPBOLUS;}
[tempx== 0]{setDoseRate (temp); O_ProgrammedFlowRate := calcFlowRate (temp);}
ConfirmStop
en: ml.GMsgConStr = MSG_CONFIRMSTOP;
enterinto (35);
copyInfuStatus ();
on every (3, E_Clock);
copyInfuStatus ();
E_ConfirmStopInfusion {ml.GAlrmMsg = MSG_BLANK;}
E_Cancel
E_StopInfusion
[infusionDone (vtbi) == 1]
E_Warning {warning ();}
E_Cancel {resumeInfusion ();}
[infusing > 0 && bolusing > 0 && infuStatus.isBolusInProgress == 0][bolusing := 0; ml.GAlrmMsg = MSG_STOPBOLUS;}
E_PauseInfusion
ConfirmPause
en: ml.GMsgConStr = MSG_CONFIRMPAUSE;
enterinto (34);
copyInfuStatus ();
on every (3, E_Clock);
copyInfuStatus ();
E_ConfirmPauseInfusion {ml.GAlrmMsg = MSG_BLANK;}
[infusionDone (vtbi) == 1]
E_Cancel
E_Warning {warning ();}
E_ClearAlarm {ClearCond:= 13; resumeInfusion ();E_AlarmClear }
[infusionDone (vtbi) == 1]
{ml.GAlrmMsg = MSG_BLANK;}
{displayBolusInfo ();}
{O_BolusRequested := 0;}
BolusRequest
en: enterinto (29);
E_RequestBolus
[infuStatus.isBolusInProgress == 0]{O_BolusRequested := 1;}
[infuStatus.isBolusInProgress == 1]{ml.GAlrmMsg = MSG_BOLUSDENIED;}
[infusing > 0 && bolusing > 0 && infuStatus.isBolusInProgress == 0][bolusing := 0; ml.GAlrmMsg = MSG_STOPBOLUS;}
[tempx == 0]{resumeInfusion ();}


InfusionStateMachine
Init
E_ConfirmPowerDown
{O_AlarmCond := 0 ; ml.GAlrmMsg := MSG_BLANK;}
E_Warning {warning ();}
ConfirmPowerDown
en: enterinto (2;);
ml.GMsgConSir := MSG_POWEROFF;
E_PowerButton {tempx= 4 ;}
PowerOff
en: enterinto (1;);
ex: ml.GMsgConSir = MSG_WELCOME;
E_PowerButton {init (); ml.GMsgConSir = MSG_POST;}
E_PowerButton
E_Cancel [tempx== 4]
E_PowerButton {tempx= 1 ;}
POST
en: enterinto (3;);
tempx := pumpConfigData.isPOSTSuccessful ;
[tempx== 0]
NUM_POSTFailed
en: ml.GMsgConSir =MSG_POSTFAIL;
enterinto (4;);
[tempx== 1]
E_Warning {warning ();}
E_Warning {warning ();}
POSTDONE
en: ml.GMsgConSir = MSG_POSTDONE;
enterinto (45 ;);
tempx := 0;
E_Cancel [tempx== 1]
E_PowerButton {tempx= 2 ;}
E_PowerButton {tempx= 3 ;}
E_Cancel [tempx== 3]
E_Cancel [tempx== 2]
E_CheckAdminSet {resetInfusionInstructions (); initInfuSta tus ();}
E_NewInfusion {resetInfusionInstructions (); initInfuSta tus ();}
CheckDrugRoutine
InfusionInSession
E_PowerButton
{O_AlarmCond := 0 ;
ml.GAlrmMsg := MSG_BLANK;}
E_ConfigureInfusionProgram {setDrugLibInfo ();}
E_Ready { startInfusion ();}
ConfigureInfusionProgram
[alrmLevel ()== 1]
[alrmLevel ()== 1]
LEVELONEALRM
en: setLOneMsg (ErrCond);
enterinto (43 ;);
[alrmLevel ()== 1]{ resetInfusionInstructions ();}