safety architect

Propagation tree report

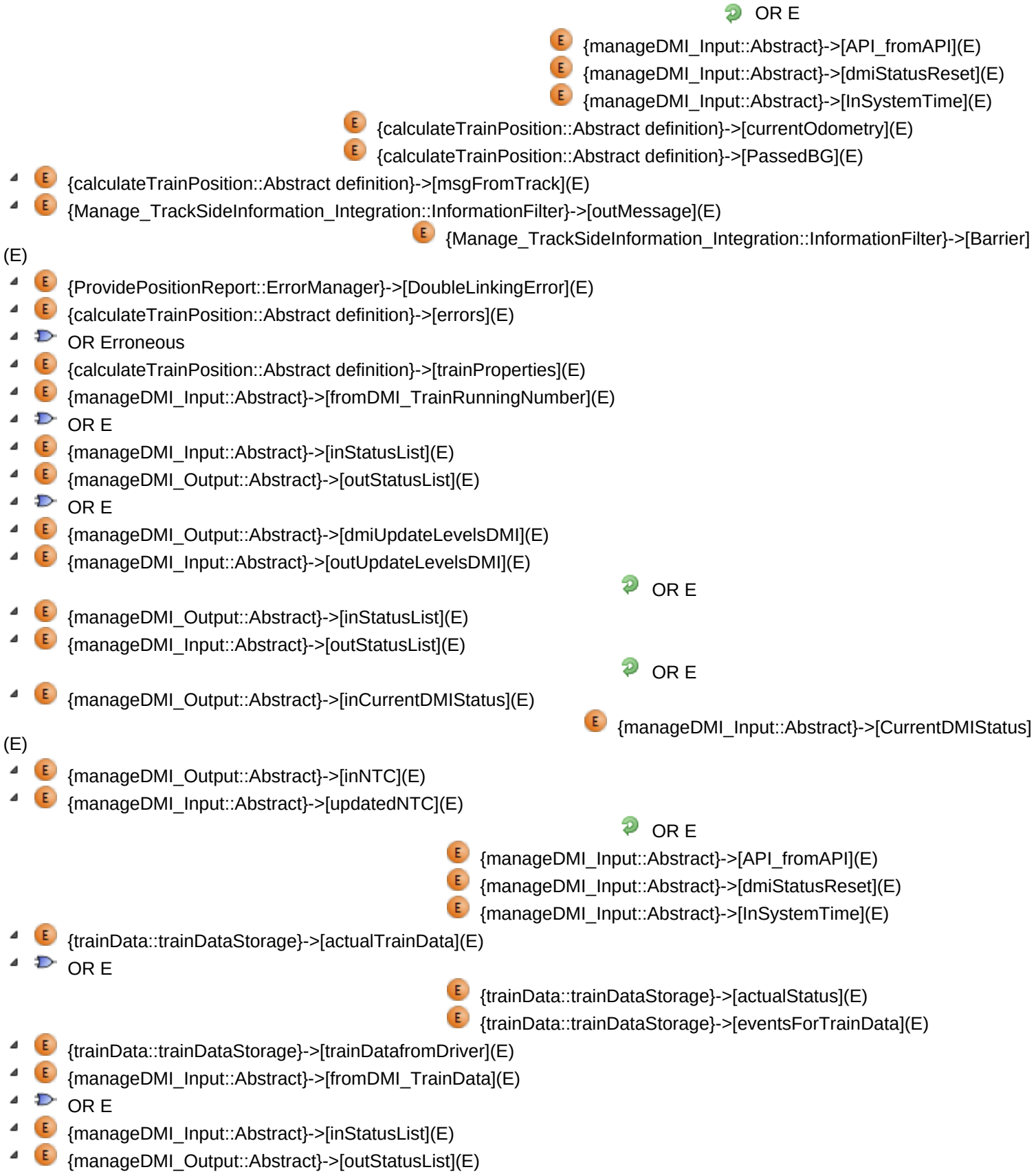
1. Feared event

Name : Erroneous Message
Date : Dec 9, 2015, 3:08 PM

2. Propagation tree

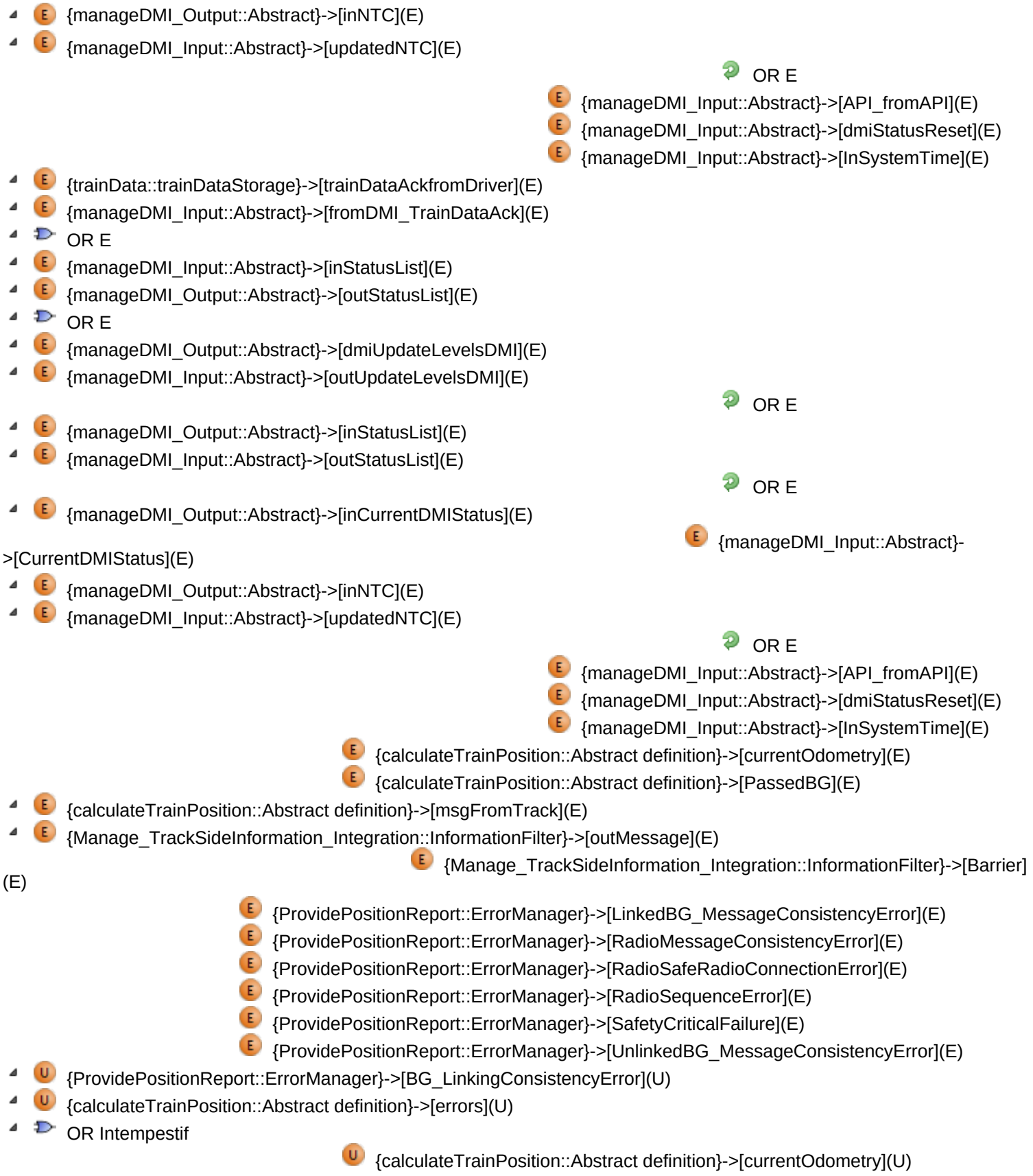
- Erroneous Message
- {ProvidePositionReport::Build_SinglePosReport}->[posRepHeader](E)
- OR Gate
- {ProvidePositionReport::Build_SinglePosReport}->[errorMsg](E)
- {ProvidePositionReport::ErrorManager}->[errorVector](E)
- OR Gate
- {ProvidePositionReport::ErrorManager}->[BG_LinkingConsistencyError](E)
- {calculateTrainPosition::Abstract definition}->[errors](E)
- OR Erroneous
- {calculateTrainPosition::Abstract definition}->[trainProperties](E)
- {manageDMI_Input::Abstract}->[fromDMI_TrainRunningNumber](E)
- OR E
- {manageDMI_Input::Abstract}->[inStatusList](E)
- {manageDMI_Output::Abstract}->[outStatusList](E)
- OR E
- {manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](E)
- {manageDMI_Input::Abstract}->[outUpdateLevelsDMI](E)
  - OR E
    - {manageDMI_Output::Abstract}->[inStatusList](E)
    - {manageDMI_Input::Abstract}->[outStatusList](E)
      - OR E
        - {manageDMI_Output::Abstract}->[inCurrentDMIStatus](E)
          - {manageDMI_Input::Abstract}->[CurrentDMIStatus](E)
        - {manageDMI_Output::Abstract}->[inNTC](E)
        - {manageDMI_Input::Abstract}->[updatedNTC](E)
          - OR E
            - {manageDMI_Input::Abstract}->[API_fromAPI](E)
            - {manageDMI_Input::Abstract}->[dmiStatusReset](E)

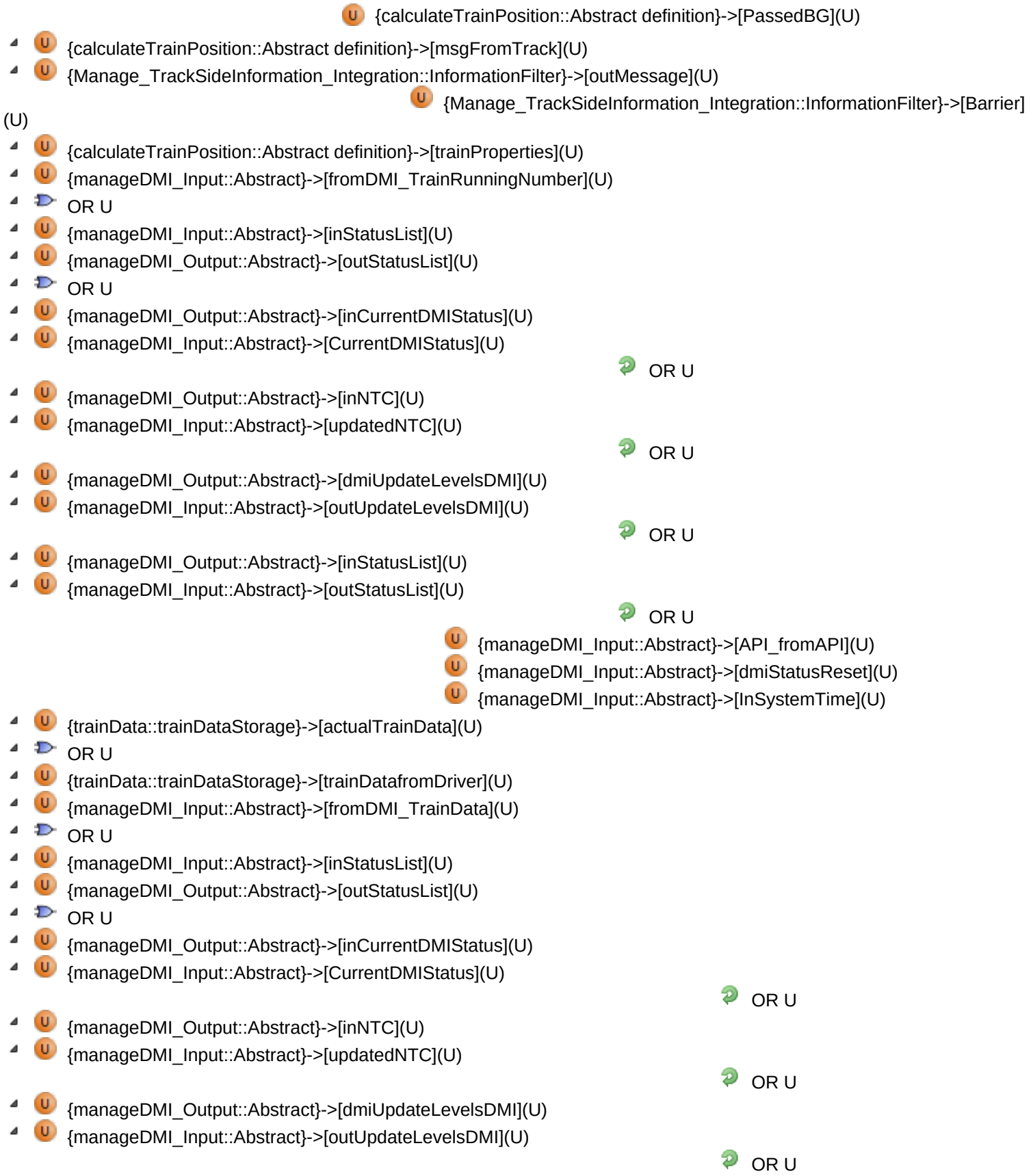
{manageDMI_Input::Abstract}->[InSystemTime](E)
{trainData::trainDataStorage}->[actualTrainData](E)
OR E
{trainData::trainDataStorage}->[actualStatus](E)
{trainData::trainDataStorage}->[eventsForTrainData](E)
{trainData::trainDataStorage}->[trainDatafromDriver](E)
{manageDMI_Input::Abstract}->[fromDMI_TrainData](E)
OR E
{manageDMI_Input::Abstract}->[inStatusList](E)
{manageDMI_Output::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](E)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](E)
OR E
{manageDMI_Output::Abstract}->[inStatusList](E)
{manageDMI_Input::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](E)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](E)
{manageDMI_Output::Abstract}->[inNTC](E)
{manageDMI_Input::Abstract}->[updatedNTC](E)
OR E
{manageDMI_Input::Abstract}->[API_fromAPI](E)
{manageDMI_Input::Abstract}->[dmiStatusReset](E)
{manageDMI_Input::Abstract}->[InSystemTime](E)
{trainData::trainDataStorage}->[trainDataAckfromDriver](E)
{manageDMI_Input::Abstract}->[fromDMI_TrainDataAck](E)
OR E
{manageDMI_Input::Abstract}->[inStatusList](E)
{manageDMI_Output::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](E)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](E)
OR E
{manageDMI_Output::Abstract}->[inStatusList](E)
{manageDMI_Input::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](E)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](E)
{manageDMI_Output::Abstract}->[inNTC](E)
{manageDMI_Input::Abstract}->[updatedNTC](E)

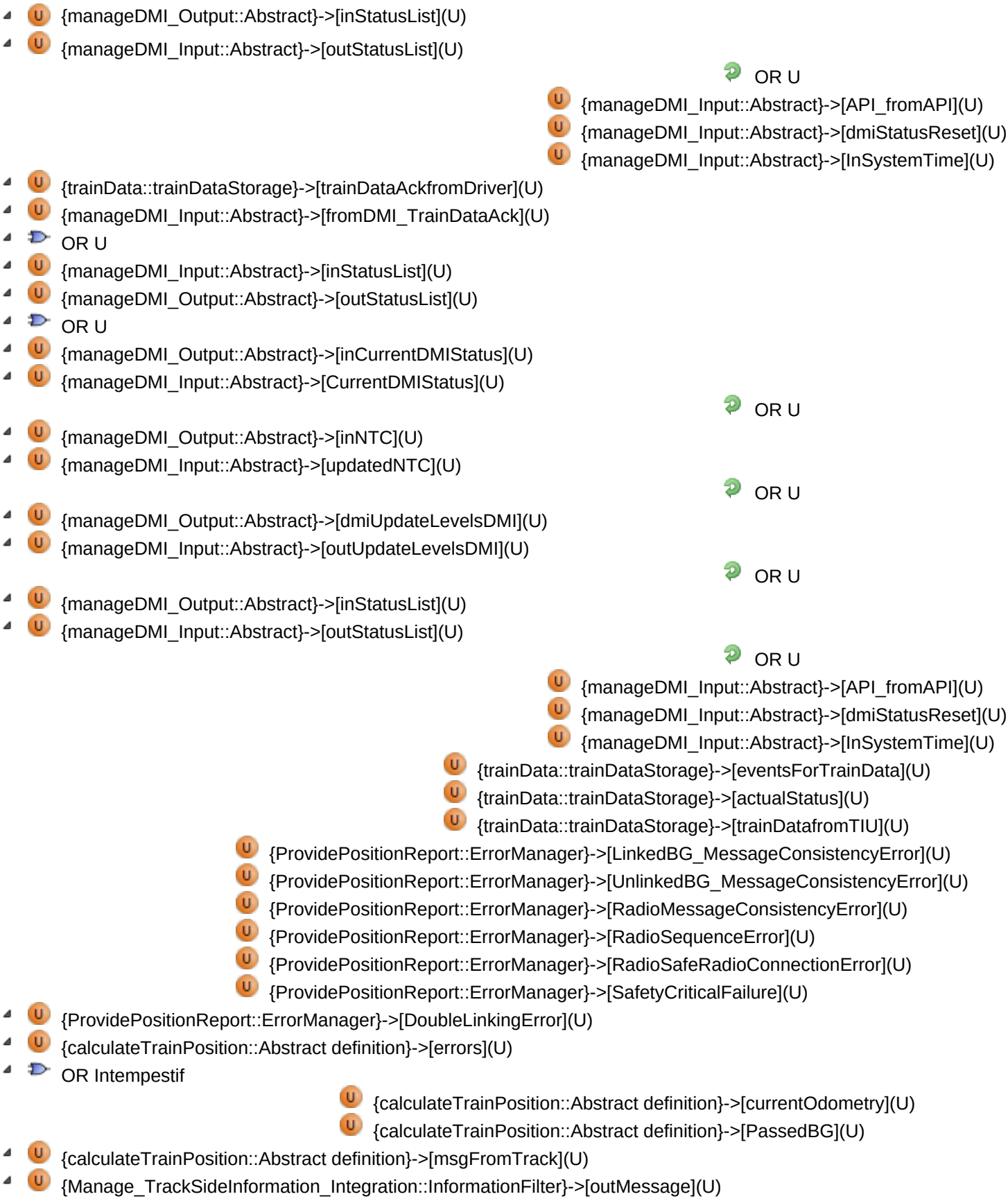

OR E
{manageDMI_Input::Abstract}->[API_fromAPI](E)
{manageDMI_Input::Abstract}->[dmiStatusReset](E)
{manageDMI_Input::Abstract}->[InSystemTime](E)
{calculateTrainPosition::Abstract definition}->[currentOdometry](E)
{calculateTrainPosition::Abstract definition}->[PassedBG](E)
{calculateTrainPosition::Abstract definition}->[msgFromTrack](E)
{Manage_TrackSideInformation_Integration::InformationFilter}->[outMessage](E)
{Manage_TrackSideInformation_Integration::InformationFilter}->[Barrier](E)
{ProvidePositionReport::ErrorManager}->[DoubleLinkingError](E)
{calculateTrainPosition::Abstract definition}->[errors](E)
OR Erroneous
{calculateTrainPosition::Abstract definition}->[trainProperties](E)
{manageDMI_Input::Abstract}->[fromDMI_TrainRunningNumber](E)
OR E
{manageDMI_Input::Abstract}->[inStatusList](E)
{manageDMI_Output::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](E)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](E)
OR E
{manageDMI_Output::Abstract}->[inStatusList](E)
{manageDMI_Input::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](E)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](E)
{manageDMI_Output::Abstract}->[inNTC](E)
{manageDMI_Input::Abstract}->[updatedNTC](E)
OR E
{manageDMI_Input::Abstract}->[API_fromAPI](E)
{manageDMI_Input::Abstract}->[dmiStatusReset](E)
{manageDMI_Input::Abstract}->[InSystemTime](E)
{trainData::trainDataStorage}->[actualTrainData](E)
OR E
{trainData::trainDataStorage}->[actualStatus](E)
{trainData::trainDataStorage}->[eventsForTrainData](E)
{trainData::trainDataStorage}->[trainDatafromDriver](E)
{manageDMI_Input::Abstract}->[fromDMI_TrainData](E)
OR E
{manageDMI_Input::Abstract}->[inStatusList](E)
{manageDMI_Output::Abstract}->[outStatusList](E)

OR E
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](E)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](E)
OR E
{manageDMI_Output::Abstract}->[inStatusList](E)
{manageDMI_Input::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](E)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](E)
{manageDMI_Output::Abstract}->[inNTC](E)
{manageDMI_Input::Abstract}->[updatedNTC](E)
OR E
{manageDMI_Input::Abstract}->[API_fromAPI](E)
{manageDMI_Input::Abstract}->[dmiStatusReset](E)
{manageDMI_Input::Abstract}->[InSystemTime](E)
{trainData::trainDataStorage}->[trainDataAckfromDriver](E)
{manageDMI_Input::Abstract}->[fromDMI_TrainDataAck](E)
OR E
{manageDMI_Input::Abstract}->[inStatusList](E)
{manageDMI_Output::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](E)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](E)
OR E
{manageDMI_Output::Abstract}->[inStatusList](E)
{manageDMI_Input::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](E)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](E)
{manageDMI_Output::Abstract}->[inNTC](E)
{manageDMI_Input::Abstract}->[updatedNTC](E)
OR E
{manageDMI_Input::Abstract}->[API_fromAPI](E)
{manageDMI_Input::Abstract}->[dmiStatusReset](E)
{manageDMI_Input::Abstract}->[InSystemTime](E)
{calculateTrainPosition::Abstract definition}->[currentOdometry](E)
{calculateTrainPosition::Abstract definition}->[PassedBG](E)
{calculateTrainPosition::Abstract definition}->[msgFromTrack](E)
{Manage_TrackSideInformation_Integration::InformationFilter}->[outMessage](E)
{Manage_TrackSideInformation_Integration::InformationFilter}->[Barrier](E)

{ProvidePositionReport::ErrorManager}->[DoubleRepositioningError](E)
{calculateTrainPosition::Abstract definition}->[errors](E)
OR Erroneous
{calculateTrainPosition::Abstract definition}->[trainProperties](E)
{manageDMI_Input::Abstract}->[fromDMI_TrainRunningNumber](E)
OR E
{manageDMI_Input::Abstract}->[inStatusList](E)
{manageDMI_Output::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](E)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](E)
OR E
{manageDMI_Output::Abstract}->[inStatusList](E)
{manageDMI_Input::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](E)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](E)
{manageDMI_Output::Abstract}->[inNTC](E)
{manageDMI_Input::Abstract}->[updatedNTC](E)
OR E
{manageDMI_Input::Abstract}->[API_fromAPI](E)
{manageDMI_Input::Abstract}->[dmiStatusReset](E)
{manageDMI_Input::Abstract}->[InSystemTime](E)
{trainData::trainDataStorage}->[actualTrainData](E)
OR E
{trainData::trainDataStorage}->[actualStatus](E)
{trainData::trainDataStorage}->[eventsForTrainData](E)
{trainData::trainDataStorage}->[trainDatafromDriver](E)
{manageDMI_Input::Abstract}->[fromDMI_TrainData](E)
OR E
{manageDMI_Input::Abstract}->[inStatusList](E)
{manageDMI_Output::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](E)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](E)
OR E
{manageDMI_Output::Abstract}->[inStatusList](E)
{manageDMI_Input::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](E)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](E)


{manageDMI_Output::Abstract}->[inNTC](E)
{manageDMI_Input::Abstract}->[updatedNTC](E)
OR E
{manageDMI_Input::Abstract}->[API_fromAPI](E)
{manageDMI_Input::Abstract}->[dmiStatusReset](E)
{manageDMI_Input::Abstract}->[InSystemTime](E)
{trainData::trainDataStorage}->[trainDataAckfromDriver](E)
{manageDMI_Input::Abstract}->[fromDMI_TrainDataAck](E)
OR E
{manageDMI_Input::Abstract}->[inStatusList](E)
{manageDMI_Output::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](E)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](E)
OR E
{manageDMI_Output::Abstract}->[inStatusList](E)
{manageDMI_Input::Abstract}->[outStatusList](E)
OR E
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](E)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](E)
{manageDMI_Output::Abstract}->[inNTC](E)
{manageDMI_Input::Abstract}->[updatedNTC](E)
OR E
{manageDMI_Input::Abstract}->[API_fromAPI](E)
{manageDMI_Input::Abstract}->[dmiStatusReset](E)
{manageDMI_Input::Abstract}->[InSystemTime](E)
{calculateTrainPosition::Abstract definition}->[currentOdometry](E)
{calculateTrainPosition::Abstract definition}->[PassedBG](E)
{calculateTrainPosition::Abstract definition}->[msgFromTrack](E)
{Manage_TrackSideInformation_Integration::InformationFilter}->[outMessage](E)
{Manage_TrackSideInformation_Integration::InformationFilter}->[Barrier](E)
{ProvidePositionReport::ErrorManager}->[LinkedBG_MessageConsistencyError](E)
{ProvidePositionReport::ErrorManager}->[RadioMessageConsistencyError](E)
{ProvidePositionReport::ErrorManager}->[RadioSafeRadioConnectionError](E)
{ProvidePositionReport::ErrorManager}->[RadioSequenceError](E)
{ProvidePositionReport::ErrorManager}->[SafetyCriticalFailure](E)
{ProvidePositionReport::ErrorManager}->[UnlinkedBG_MessageConsistencyError](E)
{ProvidePositionReport::ErrorManager}->[BG_LinkingConsistencyError](U)
{calculateTrainPosition::Abstract definition}->[errors](U)
OR Intempestif
{calculateTrainPosition::Abstract definition}->[currentOdometry](U)


{calculateTrainPosition::Abstract definition}->[PassedBG](U)
{calculateTrainPosition::Abstract definition}->[msgFromTrack](U)
{Manage_TrackSideInformation_Integration::InformationFilter}->[outMessage](U)
{Manage_TrackSideInformation_Integration::InformationFilter}->[Barrier](U)
{calculateTrainPosition::Abstract definition}->[trainProperties](U)
{manageDMI_Input::Abstract}->[fromDMI_TrainRunningNumber](U)
OR U
{manageDMI_Input::Abstract}->[inStatusList](U)
{manageDMI_Output::Abstract}->[outStatusList](U)
OR U
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](U)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](U)
OR U
{manageDMI_Output::Abstract}->[inNTC](U)
{manageDMI_Input::Abstract}->[updatedNTC](U)
OR U
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](U)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](U)
OR U
{manageDMI_Output::Abstract}->[inStatusList](U)
{manageDMI_Input::Abstract}->[outStatusList](U)
OR U
{manageDMI_Input::Abstract}->[API_fromAPI](U)
{manageDMI_Input::Abstract}->[dmiStatusReset](U)
{manageDMI_Input::Abstract}->[InSystemTime](U)
{trainData::trainDataStorage}->[actualTrainData](U)
OR U
{trainData::trainDataStorage}->[trainDatafromDriver](U)
{manageDMI_Input::Abstract}->[fromDMI_TrainData](U)
OR U
{manageDMI_Input::Abstract}->[inStatusList](U)
{manageDMI_Output::Abstract}->[outStatusList](U)
OR U
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](U)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](U)
OR U
{manageDMI_Output::Abstract}->[inNTC](U)
{manageDMI_Input::Abstract}->[updatedNTC](U)
OR U
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](U)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](U)
OR U


{manageDMI_Output::Abstract}->[inStatusList](U)
{manageDMI_Input::Abstract}->[outStatusList](U)
OR U
{manageDMI_Input::Abstract}->[API_fromAPI](U)
{manageDMI_Input::Abstract}->[dmiStatusReset](U)
{manageDMI_Input::Abstract}->[InSystemTime](U)
{trainData::trainDataStorage}->[trainDataAckfromDriver](U)
{manageDMI_Input::Abstract}->[fromDMI_TrainDataAck](U)
OR U
{manageDMI_Input::Abstract}->[inStatusList](U)
{manageDMI_Output::Abstract}->[outStatusList](U)
OR U
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](U)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](U)
OR U
{manageDMI_Output::Abstract}->[inNTC](U)
{manageDMI_Input::Abstract}->[updatedNTC](U)
OR U
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](U)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](U)
OR U
{manageDMI_Output::Abstract}->[inStatusList](U)
{manageDMI_Input::Abstract}->[outStatusList](U)
OR U
{manageDMI_Input::Abstract}->[API_fromAPI](U)
{manageDMI_Input::Abstract}->[dmiStatusReset](U)
{manageDMI_Input::Abstract}->[InSystemTime](U)
{trainData::trainDataStorage}->[eventsForTrainData](U)
{trainData::trainDataStorage}->[actualStatus](U)
{trainData::trainDataStorage}->[trainDatafromTIU](U)
{ProvidePositionReport::ErrorManager}->[LinkedBG_MessageConsistencyError](U)
{ProvidePositionReport::ErrorManager}->[UnlinkedBG_MessageConsistencyError](U)
{ProvidePositionReport::ErrorManager}->[RadioMessageConsistencyError](U)
{ProvidePositionReport::ErrorManager}->[RadioSequenceError](U)
{ProvidePositionReport::ErrorManager}->[RadioSafeRadioConnectionError](U)
{ProvidePositionReport::ErrorManager}->[SafetyCriticalFailure](U)
{ProvidePositionReport::ErrorManager}->[DoubleLinkingError](U)
{calculateTrainPosition::Abstract definition}->[errors](U)
OR Intempestif
{calculateTrainPosition::Abstract definition}->[currentOdometry](U)
{calculateTrainPosition::Abstract definition}->[PassedBG](U)
{calculateTrainPosition::Abstract definition}->[msgFromTrack](U)
{Manage_TrackSideInformation_Integration::InformationFilter}->[outMessage](U)

{Manage_TrackSideInformation_Integration::InformationFilter}->[Barrier](U)

- {calculateTrainPosition::Abstract definition}->[trainProperties](U)
- {manageDMI_Input::Abstract}->[fromDMI_TrainRunningNumber](U)
- OR U
- {manageDMI_Input::Abstract}->[inStatusList](U)
- {manageDMI_Output::Abstract}->[outStatusList](U)
- OR U
- {manageDMI_Output::Abstract}->[inCurrentDMIStatus](U)
- {manageDMI_Input::Abstract}->[CurrentDMIStatus](U)
  - OR U
- {manageDMI_Output::Abstract}->[inNTC](U)
- {manageDMI_Input::Abstract}->[updatedNTC](U)
  - OR U
- {manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](U)
- {manageDMI_Input::Abstract}->[outUpdateLevelsDMI](U)
  - OR U
- {manageDMI_Output::Abstract}->[inStatusList](U)
- {manageDMI_Input::Abstract}->[outStatusList](U)
  - OR U
    - {manageDMI_Input::Abstract}->[API_fromAPI](U)
    - {manageDMI_Input::Abstract}->[dmiStatusReset](U)
    - {manageDMI_Input::Abstract}->[InSystemTime](U)
- {trainData::trainDataStorage}->[actualTrainData](U)
- OR U
- {trainData::trainDataStorage}->[trainDatafromDriver](U)
- {manageDMI_Input::Abstract}->[fromDMI_TrainData](U)
- OR U
- {manageDMI_Input::Abstract}->[inStatusList](U)
- {manageDMI_Output::Abstract}->[outStatusList](U)
- OR U
- {manageDMI_Output::Abstract}->[inCurrentDMIStatus](U)
- {manageDMI_Input::Abstract}->[CurrentDMIStatus](U)
  - OR U
- {manageDMI_Output::Abstract}->[inNTC](U)
- {manageDMI_Input::Abstract}->[updatedNTC](U)
  - OR U
- {manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](U)
- {manageDMI_Input::Abstract}->[outUpdateLevelsDMI](U)
  - OR U
- {manageDMI_Output::Abstract}->[inStatusList](U)
- {manageDMI_Input::Abstract}->[outStatusList](U)
  - OR U

{manageDMI_Input::Abstract}->[API_fromAPI](U)
{manageDMI_Input::Abstract}->[dmiStatusReset](U)
{manageDMI_Input::Abstract}->[InSystemTime](U)
{trainData::trainDataStorage}->[trainDataAckfromDriver](U)
{manageDMI_Input::Abstract}->[fromDMI_TrainDataAck](U)
OR U
{manageDMI_Input::Abstract}->[inStatusList](U)
{manageDMI_Output::Abstract}->[outStatusList](U)
OR U
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](U)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](U)
OR U
{manageDMI_Output::Abstract}->[inNTC](U)
{manageDMI_Input::Abstract}->[updatedNTC](U)
OR U
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](U)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](U)
OR U
{manageDMI_Output::Abstract}->[inStatusList](U)
{manageDMI_Input::Abstract}->[outStatusList](U)
OR U
{manageDMI_Input::Abstract}->[API_fromAPI](U)
{manageDMI_Input::Abstract}->[dmiStatusReset](U)
{manageDMI_Input::Abstract}->[InSystemTime](U)
{trainData::trainDataStorage}->[eventsForTrainData](U)
{trainData::trainDataStorage}->[actualStatus](U)
{trainData::trainDataStorage}->[trainDatafromTIU](U)
{ProvidePositionReport::ErrorManager}->[DoubleRepositioningError](U)
{calculateTrainPosition::Abstract definition}->[errors](U)
OR Intempestif
{calculateTrainPosition::Abstract definition}->[currentOdometry](U)
{calculateTrainPosition::Abstract definition}->[PassedBG](U)
{calculateTrainPosition::Abstract definition}->[msgFromTrack](U)
{Manage_TrackSideInformation_Integration::InformationFilter}->[outMessage](U)
{Manage_TrackSideInformation_Integration::InformationFilter}->[Barrier](U)
{calculateTrainPosition::Abstract definition}->[trainProperties](U)
{manageDMI_Input::Abstract}->[fromDMI_TrainRunningNumber](U)
OR U
{manageDMI_Input::Abstract}->[inStatusList](U)
{manageDMI_Output::Abstract}->[outStatusList](U)
OR U
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](U)

{manageDMI_Input::Abstract}->[CurrentDMIStatus](U)
OR U
{manageDMI_Output::Abstract}->[inNTC](U)
{manageDMI_Input::Abstract}->[updatedNTC](U)
OR U
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](U)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](U)
OR U
{manageDMI_Output::Abstract}->[inStatusList](U)
{manageDMI_Input::Abstract}->[outStatusList](U)
OR U
{manageDMI_Input::Abstract}->[API_fromAPI](U)
{manageDMI_Input::Abstract}->[dmiStatusReset](U)
{manageDMI_Input::Abstract}->[InSystemTime](U)
{trainData::trainDataStorage}->[actualTrainData](U)
OR U
{trainData::trainDataStorage}->[trainDatafromDriver](U)
{manageDMI_Input::Abstract}->[fromDMI_TrainData](U)
OR U
{manageDMI_Input::Abstract}->[inStatusList](U)
{manageDMI_Output::Abstract}->[outStatusList](U)
OR U
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](U)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](U)
OR U
{manageDMI_Output::Abstract}->[inNTC](U)
{manageDMI_Input::Abstract}->[updatedNTC](U)
OR U
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](U)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](U)
OR U
{manageDMI_Output::Abstract}->[inStatusList](U)
{manageDMI_Input::Abstract}->[outStatusList](U)
OR U
{manageDMI_Input::Abstract}->[API_fromAPI](U)
{manageDMI_Input::Abstract}->[dmiStatusReset](U)
{manageDMI_Input::Abstract}->[InSystemTime](U)
{trainData::trainDataStorage}->[trainDataAckfromDriver](U)
{manageDMI_Input::Abstract}->[fromDMI_TrainDataAck](U)
OR U
{manageDMI_Input::Abstract}->[inStatusList](U)
{manageDMI_Output::Abstract}->[outStatusList](U)
OR U

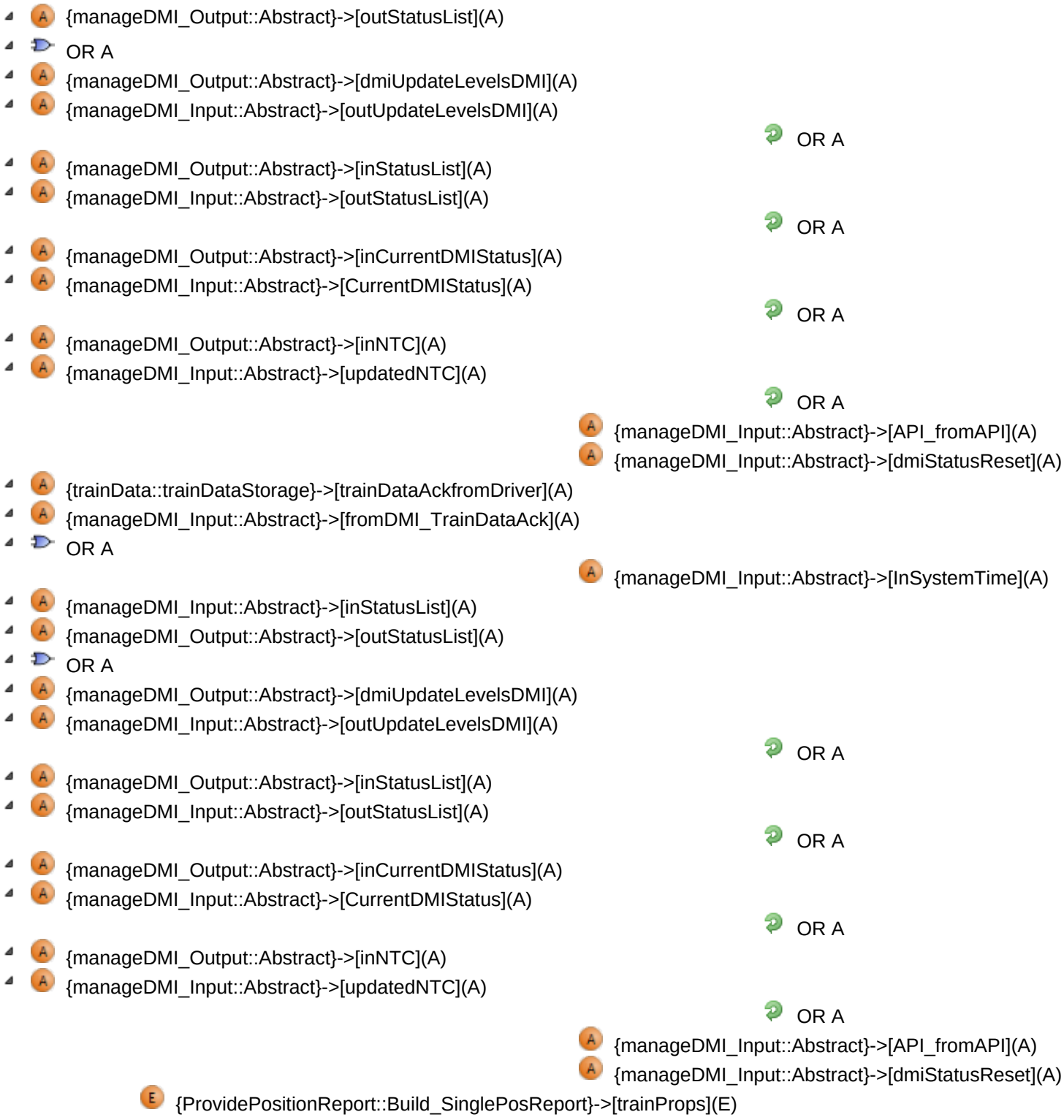
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](U)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](U)
OR U
{manageDMI_Output::Abstract}->[inNTC](U)
{manageDMI_Input::Abstract}->[updatedNTC](U)
OR U
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](U)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](U)
OR U
{manageDMI_Output::Abstract}->[inStatusList](U)
{manageDMI_Input::Abstract}->[outStatusList](U)
OR U
{manageDMI_Input::Abstract}->[API_fromAPI](U)
{manageDMI_Input::Abstract}->[dmiStatusReset](U)
{manageDMI_Input::Abstract}->[InSystemTime](U)
{trainData::trainDataStorage}->[eventsForTrainData](U)
{trainData::trainDataStorage}->[actualStatus](U)
{trainData::trainDataStorage}->[trainDatafromTIU](U)
{ProvidePositionReport::ErrorManager}->[BG_LinkingConsistencyError](A)
{calculateTrainPosition::Abstract definition}->[errors](A)
OR Abscent
{calculateTrainPosition::Abstract definition}->[currentOdometry](A)
{calculateTrainPosition::Abstract definition}->[msgFromTrack](A)
{Manage_TrackSideInformation_Integration::InformationFilter}->[outMessage](A)
{Manage_TrackSideInformation_Integration::InformationFilter}->[Barrier](A)
{calculateTrainPosition::Abstract definition}->[trainProperties](A)
{manageDMI_Input::Abstract}->[fromDMI_TrainRunningNumber](A)
OR A
{manageDMI_Input::Abstract}->[InSystemTime](A)
{manageDMI_Input::Abstract}->[inStatusList](A)
{manageDMI_Output::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](A)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](A)
OR A
{manageDMI_Output::Abstract}->[inStatusList](A)
{manageDMI_Input::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](A)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](A)
OR A
{manageDMI_Output::Abstract}->[inNTC](A)

{manageDMI_Input::Abstract}->[updatedNTC](A)
OR A
{manageDMI_Input::Abstract}->[API_fromAPI](A)
{manageDMI_Input::Abstract}->[dmiStatusReset](A)
{trainData::trainDataStorage}->[actualTrainData](A)
OR A
{trainData::trainDataStorage}->[actualStatus](A)
{trainData::trainDataStorage}->[eventsForTrainData](A)
{trainData::trainDataStorage}->[trainDatafromTIU](A)
{trainData::trainDataStorage}->[trainDatafromDriver](A)
{manageDMI_Input::Abstract}->[fromDMI_TrainData](A)
OR A
{manageDMI_Input::Abstract}->[InSystemTime](A)
{manageDMI_Input::Abstract}->[inStatusList](A)
{manageDMI_Output::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](A)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](A)
OR A
{manageDMI_Output::Abstract}->[inStatusList](A)
{manageDMI_Input::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](A)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](A)
OR A
{manageDMI_Output::Abstract}->[inNTC](A)
{manageDMI_Input::Abstract}->[updatedNTC](A)
OR A
{manageDMI_Input::Abstract}->[API_fromAPI](A)
{manageDMI_Input::Abstract}->[dmiStatusReset](A)
{trainData::trainDataStorage}->[trainDataAckfromDriver](A)
{manageDMI_Input::Abstract}->[fromDMI_TrainDataAck](A)
OR A
{manageDMI_Input::Abstract}->[InSystemTime](A)
{manageDMI_Input::Abstract}->[inStatusList](A)
{manageDMI_Output::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](A)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](A)
OR A
{manageDMI_Output::Abstract}->[inStatusList](A)
{manageDMI_Input::Abstract}->[outStatusList](A)
OR A

{manageDMI_Output::Abstract}->[inCurrentDMIStatus](A)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](A)
OR A
{manageDMI_Output::Abstract}->[inNTC](A)
{manageDMI_Input::Abstract}->[updatedNTC](A)
OR A
{manageDMI_Input::Abstract}->[API_fromAPI](A)
{manageDMI_Input::Abstract}->[dmiStatusReset](A)
{ProvidePositionReport::ErrorManager}->[LinkedBG_MessageConsistencyError](A)
{ProvidePositionReport::ErrorManager}->[UnlinkedBG_MessageConsistencyError](A)
{ProvidePositionReport::ErrorManager}->[RadioMessageConsistencyError](A)
{ProvidePositionReport::ErrorManager}->[RadioSequenceError](A)
{ProvidePositionReport::ErrorManager}->[RadioSafeRadioConnectionError](A)
{ProvidePositionReport::ErrorManager}->[SafetyCriticalFailure](A)
{ProvidePositionReport::ErrorManager}->[DoubleLinkingError](A)
{calculateTrainPosition::Abstract definition}->[errors](A)
OR Abscent
{calculateTrainPosition::Abstract definition}->[currentOdometry](A)
{calculateTrainPosition::Abstract definition}->[msgFromTrack](A)
{Manage_TrackSideInformation_Integration::InformationFilter}->[outMessage](A)
{Manage_TrackSideInformation_Integration::InformationFilter}->[Barrier](A)
{calculateTrainPosition::Abstract definition}->[trainProperties](A)
{manageDMI_Input::Abstract}->[fromDMI_TrainRunningNumber](A)
OR A
{manageDMI_Input::Abstract}->[InSystemTime](A)
{manageDMI_Input::Abstract}->[inStatusList](A)
{manageDMI_Output::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](A)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](A)
OR A
{manageDMI_Output::Abstract}->[inStatusList](A)
{manageDMI_Input::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](A)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](A)
OR A
{manageDMI_Output::Abstract}->[inNTC](A)
{manageDMI_Input::Abstract}->[updatedNTC](A)
OR A
{manageDMI_Input::Abstract}->[API_fromAPI](A)
{manageDMI_Input::Abstract}->[dmiStatusReset](A)

{trainData::trainDataStorage}->[actualTrainData](A)
OR A
{trainData::trainDataStorage}->[actualStatus](A)
{trainData::trainDataStorage}->[eventsForTrainData](A)
{trainData::trainDataStorage}->[trainDatafromTIU](A)
{trainData::trainDataStorage}->[trainDatafromDriver](A)
{manageDMI_Input::Abstract}->[fromDMI_TrainData](A)
OR A
{manageDMI_Input::Abstract}->[InSystemTime](A)
{manageDMI_Input::Abstract}->[inStatusList](A)
{manageDMI_Output::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](A)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](A)
OR A
{manageDMI_Output::Abstract}->[inStatusList](A)
{manageDMI_Input::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](A)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](A)
OR A
{manageDMI_Output::Abstract}->[inNTC](A)
{manageDMI_Input::Abstract}->[updatedNTC](A)
OR A
{manageDMI_Input::Abstract}->[API_fromAPI](A)
{manageDMI_Input::Abstract}->[dmiStatusReset](A)
{trainData::trainDataStorage}->[trainDataAckfromDriver](A)
{manageDMI_Input::Abstract}->[fromDMI_TrainDataAck](A)
OR A
{manageDMI_Input::Abstract}->[InSystemTime](A)
{manageDMI_Input::Abstract}->[inStatusList](A)
{manageDMI_Output::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](A)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](A)
OR A
{manageDMI_Output::Abstract}->[inStatusList](A)
{manageDMI_Input::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](A)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](A)
OR A
{manageDMI_Output::Abstract}->[inNTC](A)

{manageDMI_Input::Abstract}->[updatedNTC](A)
OR A
{manageDMI_Input::Abstract}->[API_fromAPI](A)
{manageDMI_Input::Abstract}->[dmiStatusReset](A)
{ProvidePositionReport::ErrorManager}->[DoubleRepositioningError](A)
{calculateTrainPosition::Abstract definition}->[errors](A)
OR Abscent
{calculateTrainPosition::Abstract definition}->[currentOdometry](A)
{calculateTrainPosition::Abstract definition}->[msgFromTrack](A)
{Manage_TrackSideInformation_Integration::InformationFilter}->[outMessage](A)
{Manage_TrackSideInformation_Integration::InformationFilter}->[Barrier](A)
{calculateTrainPosition::Abstract definition}->[trainProperties](A)
{manageDMI_Input::Abstract}->[fromDMI_TrainRunningNumber](A)
OR A
{manageDMI_Input::Abstract}->[InSystemTime](A)
{manageDMI_Input::Abstract}->[inStatusList](A)
{manageDMI_Output::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](A)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](A)
OR A
{manageDMI_Output::Abstract}->[inStatusList](A)
{manageDMI_Input::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](A)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](A)
OR A
{manageDMI_Output::Abstract}->[inNTC](A)
{manageDMI_Input::Abstract}->[updatedNTC](A)
OR A
{manageDMI_Input::Abstract}->[API_fromAPI](A)
{manageDMI_Input::Abstract}->[dmiStatusReset](A)
{trainData::trainDataStorage}->[actualTrainData](A)
OR A
{trainData::trainDataStorage}->[actualStatus](A)
{trainData::trainDataStorage}->[eventsForTrainData](A)
{trainData::trainDataStorage}->[trainDatafromTIU](A)
{trainData::trainDataStorage}->[trainDatafromDriver](A)
{manageDMI_Input::Abstract}->[fromDMI_TrainData](A)
OR A
{manageDMI_Input::Abstract}->[InSystemTime](A)
{manageDMI_Input::Abstract}->[inStatusList](A)


{manageDMI_Output::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](A)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](A)
OR A
{manageDMI_Output::Abstract}->[inStatusList](A)
{manageDMI_Input::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](A)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](A)
OR A
{manageDMI_Output::Abstract}->[inNTC](A)
{manageDMI_Input::Abstract}->[updatedNTC](A)
OR A
{manageDMI_Input::Abstract}->[API_fromAPI](A)
{manageDMI_Input::Abstract}->[dmiStatusReset](A)
{trainData::trainDataStorage}->[trainDataAckfromDriver](A)
{manageDMI_Input::Abstract}->[fromDMI_TrainDataAck](A)
OR A
{manageDMI_Input::Abstract}->[InSystemTime](A)
{manageDMI_Input::Abstract}->[inStatusList](A)
{manageDMI_Output::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[dmiUpdateLevelsDMI](A)
{manageDMI_Input::Abstract}->[outUpdateLevelsDMI](A)
OR A
{manageDMI_Output::Abstract}->[inStatusList](A)
{manageDMI_Input::Abstract}->[outStatusList](A)
OR A
{manageDMI_Output::Abstract}->[inCurrentDMIStatus](A)
{manageDMI_Input::Abstract}->[CurrentDMIStatus](A)
OR A
{manageDMI_Output::Abstract}->[inNTC](A)
{manageDMI_Input::Abstract}->[updatedNTC](A)
OR A
{manageDMI_Input::Abstract}->[API_fromAPI](A)
{manageDMI_Input::Abstract}->[dmiStatusReset](A)
{ProvidePositionReport::Build_SinglePosReport}->[trainProps](E)