AS5506/3 Error Model V2 Annex Syntax Card (Dec 2013)

E.3 Error Model Library

```
error_model_library ::=
 annex EMV2 none;
 |  annex EMV2 {**
   [ error_type_library_definition ]
   ( error_behavior_state_machine_definition )*
   ( type_mapping_set_definition )*
   ( type_transformation_set_definition )*
 **};
```

E.4 Error Model Subclause

```
error_model_subclause ::=
 annex EMV2 none;
 |
 annex EMV2 {**
    [ use types error_type_library ( , error_type_library )* ; ]
    [ use type equivalence type_mapping_set_reference ; ]
    [ use behavior error_behavior_state_machine_reference ; ]
    [ error_propagation_section ]
    [ component_error_behavior_section ]
    [ composite_error_behavior_section ]
    [ connection_error_behavior_section ]
    [ user_defined_point_path_section ]
    [ properties ( contained_property_association )+ ]
**} [ in_modes ];
```

E.5 Error Type Library, Error Types, Type Sets, Aliases

```
error_type_library_definition ::=
 error types
   [ extends error_type_library ( , error_type_library )* with ]
   ( error_type_definition | error_type_alias_definition
    type_set_definition | type_set_alias_definition )+
   [ properties ( contained_property_association )+ ]
end types;

error_type_library ::= package_name -- reference

error_type_definition ::=
  defining_identifier : type [ extends error_type ] ;

error_type_alias_definition ::=
  defining_identifier renames type error_type;

error_type ::=
  [error_type_library::]error_type_identifier |
  [error_type_library::]error_type_alias_identifier

type_set_definition ::=
  defining_identifier : type set { type_set_element ( , type_set_element )* } ;

error_type_set_alias_definition ::=
  defining_identifier renames type set type_set ;

type_set ::=
  [error_type_library::]type_set_identifier |
  [error_type_library::]type_set_alias_identifier

type_set_element ::= error_type | type_set | type_product

type_product ::= error_type_reference ( * error_type_reference )+

type_set_constructor ::= { type_set_element ( , type_set_element )* }

type_set_constraint ::= type_set_constructor | {NoError}

type_instance ::= { error_type | type_product }
```

E.8 Error Propagation Section

```
error_propagation_section ::=
    error propagations
     [ error_propagation | error_containment ]+
     [ flows (error_source | error_sink | error_path )+ ]
    end propagations;

error_propagation ::=
    error_propagation_point : ( in | out ) propagation type_set ;

error_containment ::=
    error_propagation_point : not ( in | out ) propagation type_set ;

error_propagation_point ::=
  feature_reference | binding_reference | propagation_point_identifier

feature_reference ::=
    ( feature_group_identifier . )*  feature_identifier
    | access

binding_reference ::= processor | memory | connection | bindings

error_source ::=
  defining_identifier : error source
    ( outgoing_error_propagation_point | all ) [ effect_type_set ]
    [ when fault_source ] ;

error_sink ::=
  defining_identifier : error sink
    ( incoming_error_propagation_point | all ) [ type_set ] ;

error_path ::=
  defining_identifier : error path
      ( incoming_error_propagation_point | all  ) [ type_set ] ->
      ( outgoing_error_propagation_point | all )
       [ target_type_instance | use mappings type_mapping_set ] ;

fault_source ::= error_behavior_state [type_set ] | type_set | “description”
```

E.8.3 User Defined Propagation Points and Paths

```
user_defined_point_path_section ::=
propagation paths
   ( propagation_point )*
( connections
   ( propagation_point_connection )* )?
end paths ;
```

```
propagation_point ::=
  defining_identifier : propagation point ;

propagation_point_connection ::=
    defining_identifier :
     source_user_defined_error_propagation_point ->
       target_user_defined_error_propagation_point ;

user_defined_propagation_point ::=
    ( subcomponent_identifier . )? propagation_point_identifier
```

E.9 Error Behavior State Machine

```
error_behavior_state_machine_definition ::=
  error behavior defining_state_machine_identifier
  [ use types error_type_library ( , error_type_library )* ; ]
  [ use transformations type_transformation_set ; ]
  [ events (error_event | recover_event | repair_event )+ ]
  [ states ( error_state )+ ]
  [ transitions (transition | branching_transition )+ ]
  [ properties ( contained_property_association )+ ]
  end behavior ;

error_behavior_state_machine ::=
    [package_name::]error_behavior_state_machine_identifier

error_event ::=
  defining_identifier : error event [ type_set ]
    [ when “error_event_condition” ] ;

recover_event ::=
  defining_identifier : recover event
     [ when initiator_reference ( , initiator_reference )* ] ;

initiator_reference ::=
  mode_transition_reference | event_port_reference | self_event_reference

repair_event ::=
  defining_identifier : repair event
     [ when initiator_reference ( , initiator_reference )* ] ;

error_state ::= defining_identifier : [ initial ] state [ type_set ] ;

transition ::=
    [ defining_identifier : ]
    transition_source -[ error_condition ]-> transition_target ;

branching_transition ::=
    [ defining_identifier : ]
    source_state -[ error_condition ]-> (transition_branches);

source_state ::= all | (error_state_identifier  [ type_set ]

transition_target ::=
    error_state_identifier [ type_instance ]
    | same state

transition_branches ::=
    transition_target with branch_probability
     ( , transition_target with branch_probability )*

branch_probability ::= fixed_probability_value | others

fixed_probability_value ::=
    real_literal | [ package_identifier::]real_property_constant_identifier

error_condition ::=
  condition_trigger | ( error_condition )
  | error_condition and error_condition | error_condition or error_condition
  | numeric_literal ormore ( condition_trigger ( , condition_trigger )+ )
  | numeric_literal orless ( condition_trigger ( , condition_trigger )+ )

condition_trigger ::=
    error_behavior_event_identifier [ type_set ]
    | incoming_error_propagation_point [ type_set_constraint ]
    | subcomponent_identifier . outgoing_error_propagation_point
          [ type_set_constraint ]
```

E.10 Component Error Behavior Section

```
component_error_behavior_section ::=
    component error behavior
      [ use transformations type_transformation_set ; ]
      [ events (error_event | recover_event | repair_event )+ ]
      [ transitions (transition | branching_transition )+ ]
      [ propagations ( outgoing_propagation )+ ]
      [ detections ( error_detection )+ ]
      [ mode mappings ( error_state_to_mode_mapping )+ ]
    end component;

outgoing_propagation ::=
  [ defining_identifier : ]
   (source_state | all) -[ [ error_condition ] ]-> propagation_target ;

propagation_target ::=
    (error_propagation_point | all) [ target_type_instance | {noerror} ]

error_detection ::=
  [ defining_identifier : ]
  (source_state | all ) -[ [ error_condition ] ]-> error_detection_effect ;

error_detection_effect ::=
    ( port_identifier | own_event_reference ) ! [ ( error_code_value ) ]

own_event_reference ::= self.event_or_event_data_source_identifier

error_code_value ::=
    integer_literal | enumeration_identifier | property_constant_term

error_state_to_mode_mapping ::=
  error_state_identifier [ type_instance ]
     in modes (mode_name ( , mode_name )*);
```

E.11 Composite Error Behavior Section

```
composite_error_behavior ::=
  composite error behavior
```

```
      states { composite_error_state }+
  end composite;

composite_error_state ::= [ defining_identifier : ]
  [ ( subcomponent_state_expression | others ) ]-> composite_state_identifier
      [ target_type_instance ] ;

composite_state_expression ::=
  state_element | ( composite_state_expression )
  | composite_state_expression and composite_state_expression
  | composite_state_expression or composite_state_expression
  | numeric_literal ormore ( state_element ( , state_element )+ )
  | numeric_literal orless ( state_element ( , state_element )+ )

state_element ::=
    subcomponent_error_state [ type_set ]
    | in error_propagation_point [type_set_constraint ]

subcomponent_error_state ::=
  ( subcomponent_identifier . )+ error_state_identifier
```

E.12 Connection Error Behavior Section

```
connections_error_behavior_section ::=
connection error
  [ use transformations type_transformation_set ; ]
  ( connection_error_source )*
end connection;

connection_error_source ::=
  defining_identifier :
    error source ( connection_identifier | all )
   [effect_type_set [ when ( fault_source_type_set | "description" ) ] ] ;
```

E.13 Type Mapping Set and Type Transformation Set

```
type_mapping_set_definition ::=
  type mappings defining_identifier
  [ use types error_type_library ( , error_type_library )* ; ]
   (type_mapping_rule )+
  end mappings;

type_mapping_rule ::= source_type_set -> target_type_instance ;

type_transformation_set_definition ::=
  type transformations defining_identifier
  [ use types error_type_library ( , error_type_library )* ; ]
   (type_transformation_rule )+
  end transformations;

type_transformation_rule ::=
(source_type_set_constraint | all) -[ [contributor_type_set_constraint]]->
target_type_instance;

type_mapping_set ::= [package_name::]type_mapping_set_identifier

type_transformation_set ::= [package_name::]type_transformation_set_identifier
```

E.7 Error Model Properties

```
StateKind: enumeration (Working, NonWorking)

DetectionMechanism : aadlstring

FaultKind : enumeration (Design, Operational)

Persistence : enumeration (Permanent, Transient, Singleton)

OccurrenceDistribution: record (
        ProbabilityValue : aadlreal;
        OccurrenceRate : aadlreal;
        MeanValue : aadlreal;
        StandardDeviation : aadlreal;
        ShapeParameter : aadlreal;
        ScaleParameter : aadlreal;
        SuccessCount : aadlreal;
        SampleCount : aadlreal;
        Probability : aadlreal;
        Distribution : EMV2::DistributionFunction;)

DistributionFunction : type enumeration (Fixed, Poisson, Exponential, Normal,
Gauss, Weibull, Binominal)

DurationDistribution : record (
        Duration : Time_Range;
        Distribution : EMV2::DistributionFunction;)

PropagationTimeDelay: record (
        Duration : Time_Range;
        Distribution : EMV2::DistributionFunction;)

Hazards: list of record
  (crossreference: aadlstring; -- cross reference to an external document
   hazardtitle: aadlstring; -- short descriptive phrase for hazard
   description: aadlstring; -- description of the hazard
   failure: aadlstring; -- system deviation, giving rise to failure effect
   failureeffect: aadlstring; -- consequences of a failure on system operation
   phases: list of aadlstring; -- operational phases in which hazard is active
   environment: aadlstring; -- operational environment
   mishap: aadlstring; -- event (series) resulting in unintentional death, 882
   failurecondition: aadlstring; -- event (series) resulting in death ARP4761
   risk: aadlstring; -- description of risk.
   severity: aadlinteger; --
   likelihood: EMV2::LikelihoodLabels;
   targetseverity: aadlinteger; -- acceptable risk in terms of severity
   targetlikelihood: EMV2::LikelihoodLabels; -- acceptable risk likelihood
   developmentassurancelevel : EMV2::DALLabels;
   verificationmethod : aadlstring; -- verification method to address the hazard
   safetyreport       : aadlstring; -- summary
   comment            : aadlstring; -- additional information about the hazard
   )
-- there are tailored ARP4761 and MILSTD882 variants of the Hazards property
```

E.6 Predeclared Error Types

```
CommonErrors: type set { ServiceError, TimingRelatedError, ValueRelatedError,
ReplicationError, ConcurrencyError};
--service related errors
ServiceError: type;
ItemOmission: type extends ServiceError;
ServiceOmission: type extends ServiceError;
SequenceOmission: type extends ServiceError;
TransientServiceOmission: type extends SequenceOmission;
LateServiceStart: type extends SequenceOmission;
EarlyServiceTermination: type extends SequenceOmission;
BoundedOmissionInterval: type extends SequenceOmission;
ItemComission: type extends ServiceError;
ServiceCommission: type extends ServiceError;
SequenceCommission: type extends ServiceError;
EarlyServiceStart: type extends SequenceCommission;
LateServiceTermination: type extends SequenceCommission;

--timing related errors
TimingRelatedError: type set {ItemTimingError, SequenceTimingError,
ServiceTimingError};
-- Item timing errors
ItemTimingError: type;
EarlyDelivery: type extends ItemTimingError;
LateDelivery: type extends ItemTimingError;
--Rate/sequence timing errors
SequenceTimingError: type;
HighRate: type extends SequenceTimingError;
LowRate: type extends SequenceTimingError;
RateJitter: type extends SequenceTimingError;
-- Service timing error
ServiceTimingError: type;
DelayedService: type extends ServiceTimingError;
EarlyService: type extends ServiceTimingError;

-- aliases for timing errors
TimingError renames type ItemTimingError; -- legacy
RateError renames type SequenceTimingError;
EarlyData renames type HighRate;
LateData renames type LowRate;
ServiceTimeShift renames type ServiceTimingError;

--value related errors
ValueRelatedError: type set {ItemValueError, SequenceValueError,
ServiceValueError};
-- item value errors
ItemValueError: type;
UndetectableValueError: type extends ItemValueError;
DetectableValueError: type extends ItemValueError;
OutOfRange: type extends DetectableValueError;
BelowRange: type extends OutOfRange;
AboveRange: type extends OutOfRange;
OutOfBounds: type extends DetectableValueError;
-- sequence errors
SequenceValueError: type;
BoundedValueChange: type extends SequenceError;
StuckValue: type extends SequenceError;
OutOfOrder: type extends SequenceError;

ServiceValueError: type;
OutOfCalibration: type extends ServiceValueError;

-- Common aliases for value related errors
ValueError renames type ItemValueError;
IncorrectValue renames type ItemValueError;
ValueCorruption renames type ItemValueError;
BadValue renames type ItemValueError;
SequenceError renames type SequenceValueError;

SubtleValueError renames type UndetectableValueError;
BenignValueError renames type DetectableValueError;
SubtleValueCorruption renames type DetectableValueError;
-- Detectability (Benign/Subtle) represent a characteristic of error types

--replication errors
ReplicationError: type;
AsymmetricReplicatesError: type extends ReplicationError;
AsymmetricValue: type extends AsymmetricReplicatesError;
AsymmetricApproximateValue: type extends AsymmetricValue;
AsymmetricExactValue: type extends AsymmetricValue;
AsymmetricTiming: type extends AsymmetricReplicatesError;
AsymmetricOmission: type extends AsymmetricReplicatesError;
AsymmetricItemOmission: type extends AsymmetricOmission;
AsymmetricServiceOmission: type extends AsymmetricOmission;

SymmetricReplicatesError: type extends ReplicationError;
SymmetricValue: type extends SymmetricReplicatesError;
SymmetricApproximateValue: type extends SymmetricValue;
SymmetricExactValue: type extends SymmetricValue;
SymmetricTiming: type extends SymmetricReplicatesError;
SymmetricOmission: type extends SymmetricReplicatesError;
SymmetricItemOmission: type extends SymmetricOmission;
SymmetricServiceOmission: type extends SymmetricOmission;

-- aliases for replication
InconsistentValue renames type AsymmetricValue;
InconsistentTiming renames type AsymmetricTiming;
InconsistentOmission renames type AsymmetricOmission;
InconsistentItemOmission renames type AsymmetricItemOmission;
InconsistentServiceOmission renames type AsymmetricServiceOmission;
AsymmetricTransmissive renames type AsymmetricValue;

--concurrency errors
ConcurrencyError: type;
RaceCondition: type extends ConcurrencyError;
ReadWriteRace: type extends RaceCondition;
WriteWriteRace: type extends RaceCondition;
MutExError: type extends ConcurrencyError;
Deadlock: type extends MutExError;
Starvation: type extends MutExError;
```