IBM Software Group

Safety Analysis Profile for the UML

Bruce Powel Douglass, Ph.D.
Chief Evangelist, IBM Rational
Bruce.Douglass@us.ibm.com
Twitter: @BruceDouglass
http://tech.groups.yahoo.com/group/RT-UML/

Go to IBM

Rational software

Innovation for a smarter planet

What is Safety?

- Safety is freedom from accidents or losses.
- Safety is not reliability!
  - Reliability is the probability that a system will perform its intended function satisfactorily.
- Safety is not security!
  - Security is protection or defense against attack, interference, or espionage.

Safety-Related Concepts

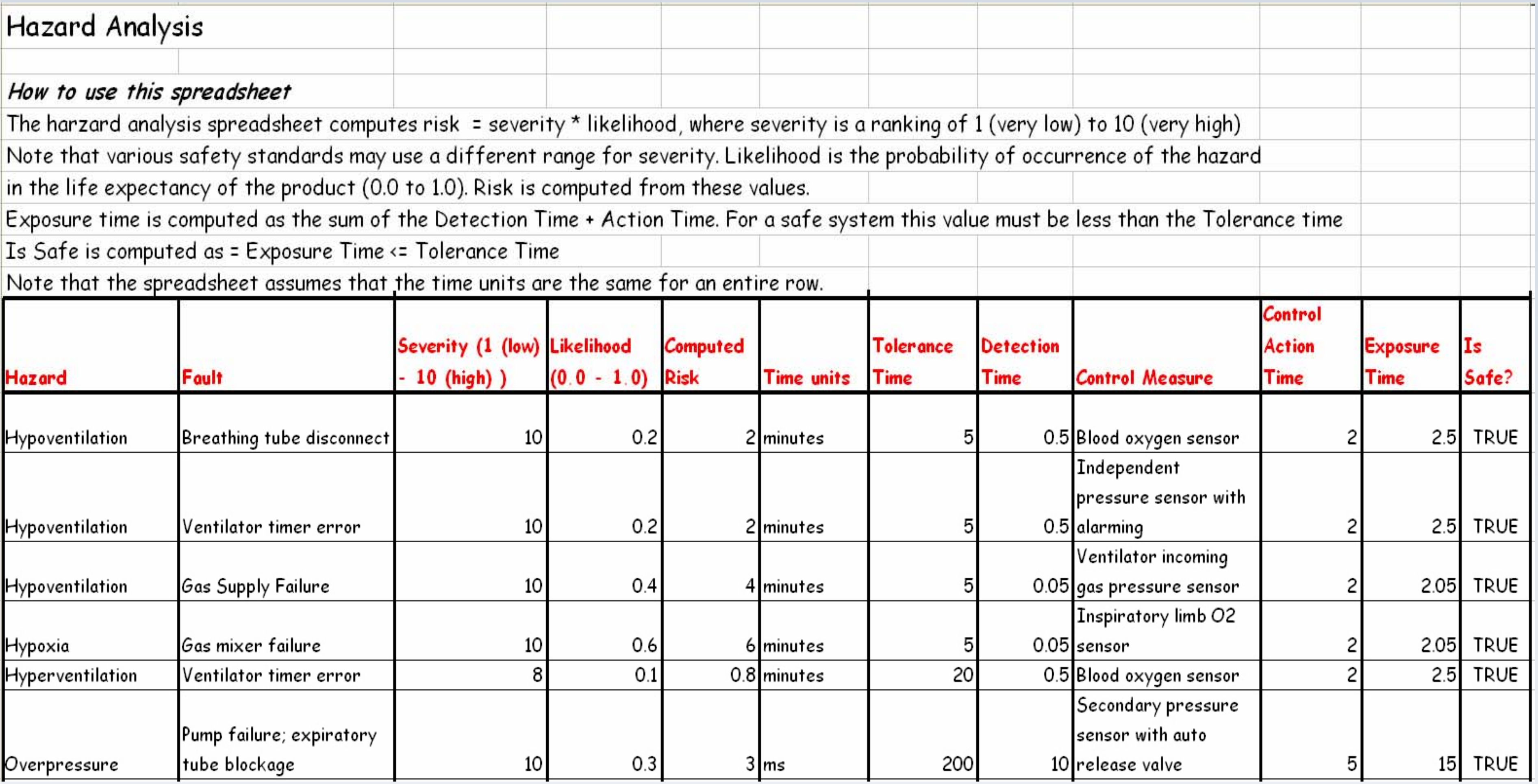
- *Accident* is a loss of some kind, such as injury, death, or equipment damage
- *Risk* is a combination of the likelihood of an accident and its severity:

  *risk = p(a) * s(a)*
- *Hazard* is a set of conditions and/or events that leads to an accident.

Safety-Related Concepts

- A failure is the nonperformance of a system or component, a random fault
  - A random failure is one that can be estimated from a pdf,
  - Failures are events
  - e.g., a component failure
- An error is a systematic fault
  - A systematic fault is an design error
  - Errors are states or conditions
  - e.g., a software bug
- A fault is either a failure or an error

Hazard Analysis



Hazard Analysis

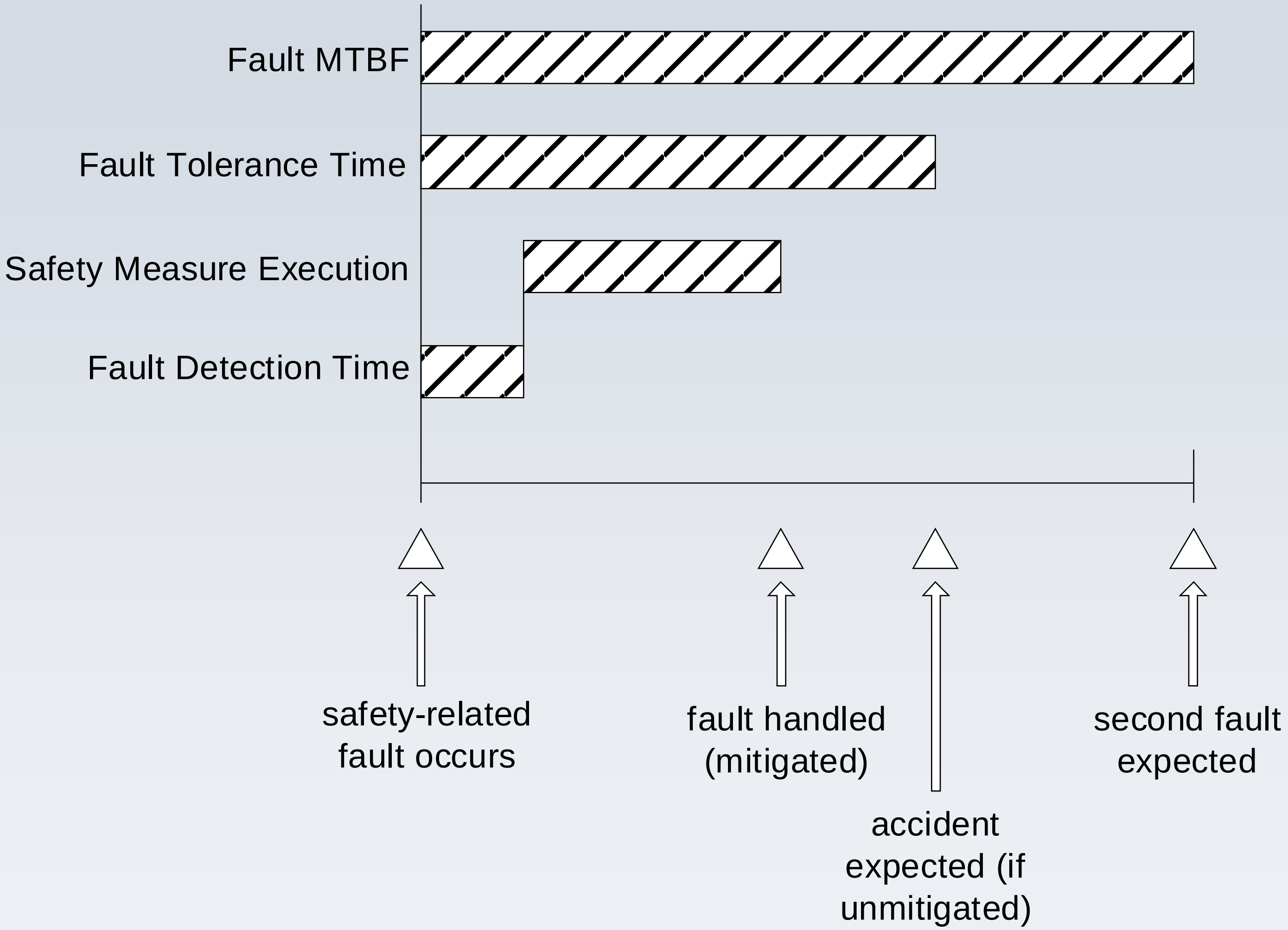
How to use this spreadsheet

The harzard analysis spreadsheet computes risk = severity * likelihood, where severity is a ranking of 1 (very low) to 10 (very high)
Note that various safety standards may use a different range for severity. Likelihood is the probability of occurrence of the hazard
in the life expectancy of the product (0.0 to 1.0). Risk is computed from these values.
Exposure time is computed as the sum of the Detection Time + Action Time. For a safe system this value must be less than the Tolerance time
Is Safe is computed as = Exposure Time <= Tolerance Time
Note that the spreadsheet assumes that the time units are the same for an entire row.

Hazard	Fault	Severity (1 (low) - 10 (high))	Likelihood (0.0 - 1.0)	Computed Risk	Time units	Tolerance Time	Detection Time	Control Measure	Control Action Time	Exposure Time	Is Safe?
Hypoventilation	Breathing tube disconnect	10	0.2	2	minutes	5	0.5	Blood oxygen sensor	2	2.5	TRUE
Hypoventilation	Ventilator timer error	10	0.2	2	minutes	5	0.5	Independent pressure sensor with alarming	2	2.5	TRUE
Hypoventilation	Gas Supply Failure	10	0.4	4	minutes	5	0.05	Ventilator incoming gas pressure sensor	2	2.05	TRUE
Hypoxia	Gas mixer failure	10	0.6	6	minutes	5	0.05	Inspiratory limb O2 sensor	2	2.05	TRUE
Hyperventilation	Ventilator timer error	8	0.1	0.8	minutes	20	0.5	Blood oxygen sensor	2	2.5	TRUE
Overpressure	Pump failure; expiratory tube blockage	10	0.3	3	ms	200	10	Secondary pressure sensor with auto release valve	5	15	TRUE

Safety Fault Timeline

$t_{\text{fault dectection}} < t_{\text{safety measure execution}} < t_{\text{fault tolerance time}} << t_{\text{MTBF}}$



Fault MTBF

Fault Tolerance Time

Safety Measure Execution

Fault Detection Time

safety-related fault occurs

fault handled (mitigated)

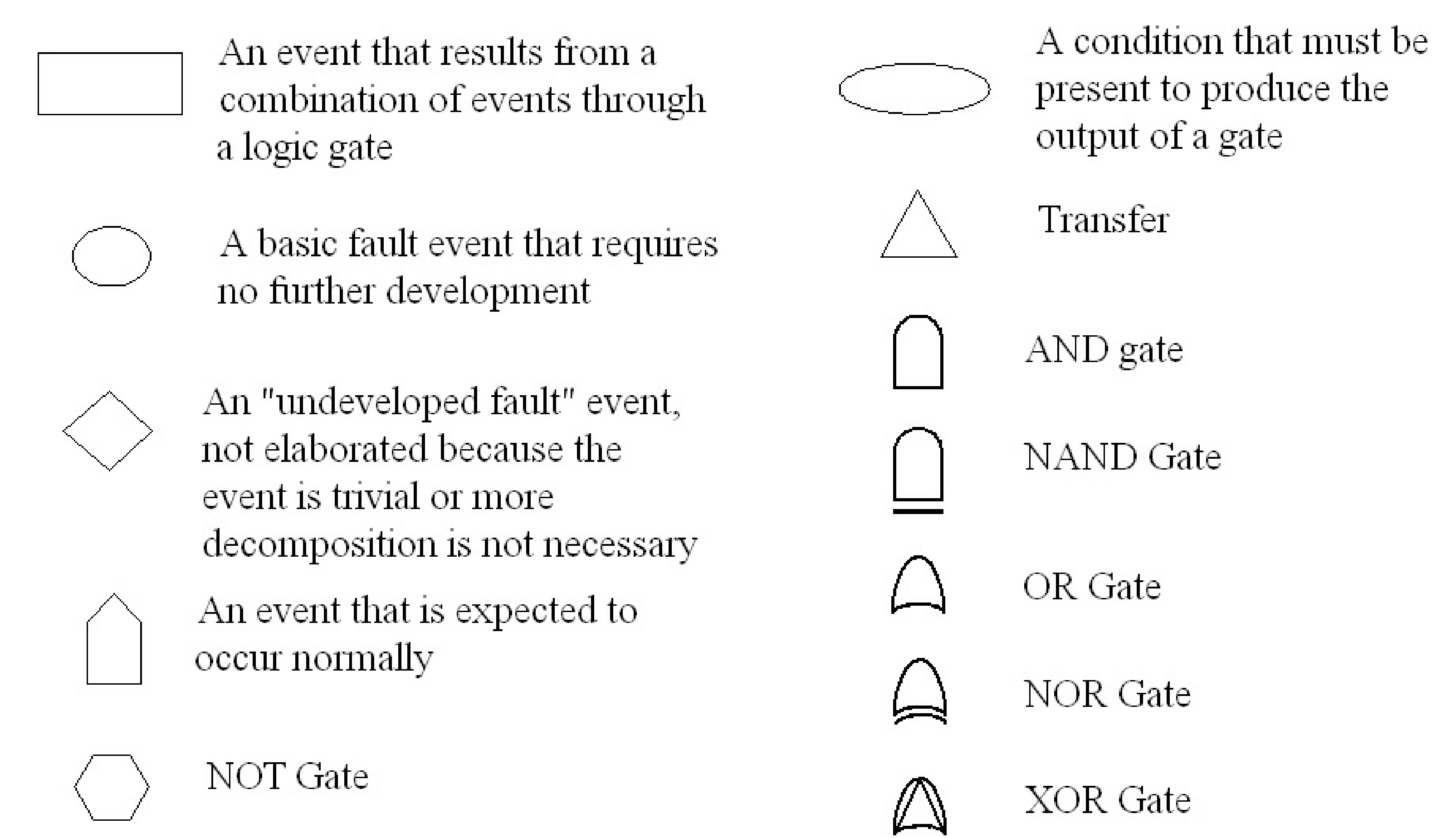
accident expected (if unmitigated)

second fault expected

Safety Measures

- Safety measures do one of the following:
  - Remove the hazard
  - Reduce the risk, either by
    - Reducing the likelihood of the accident
    - Reducing the severity of the accident
  - Identify the hazard to supervisory personnel so that they can handle it within the fault tolerance time
- The purpose of the safety measure is to avoid accident or loss

Fault Tree Analysis (FTA)



An event that results from a combination of events through a logic gate

A basic fault event that requires no further development

An "undeveloped fault" event, not elaborated because the event is trivial or more decomposition is not necessary

An event that is expected to occur normally

NOT Gate

A condition that must be present to produce the output of a gate

Transfer

AND gate

NAND Gate

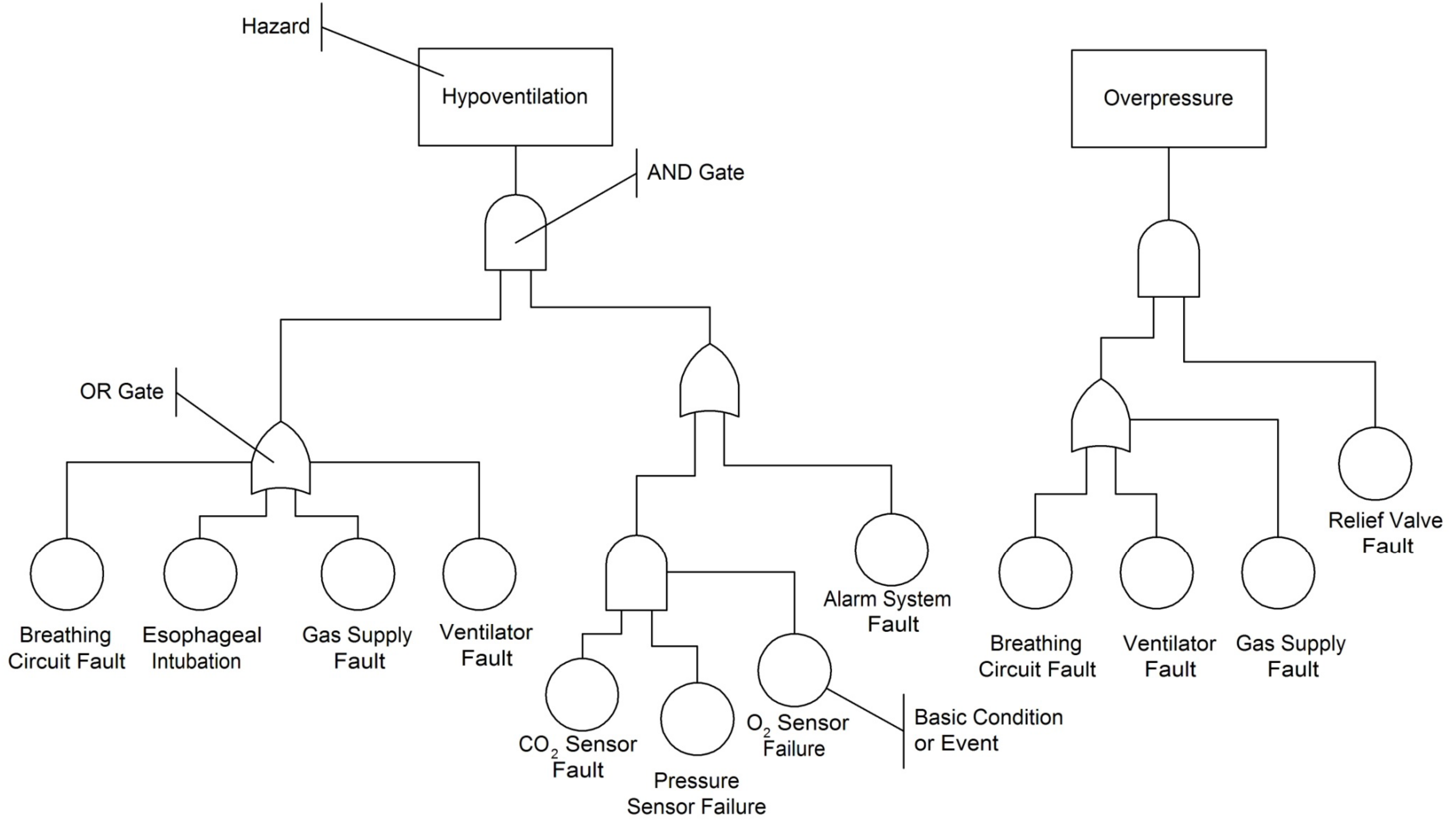
OR Gate

NOR Gate

XOR Gate

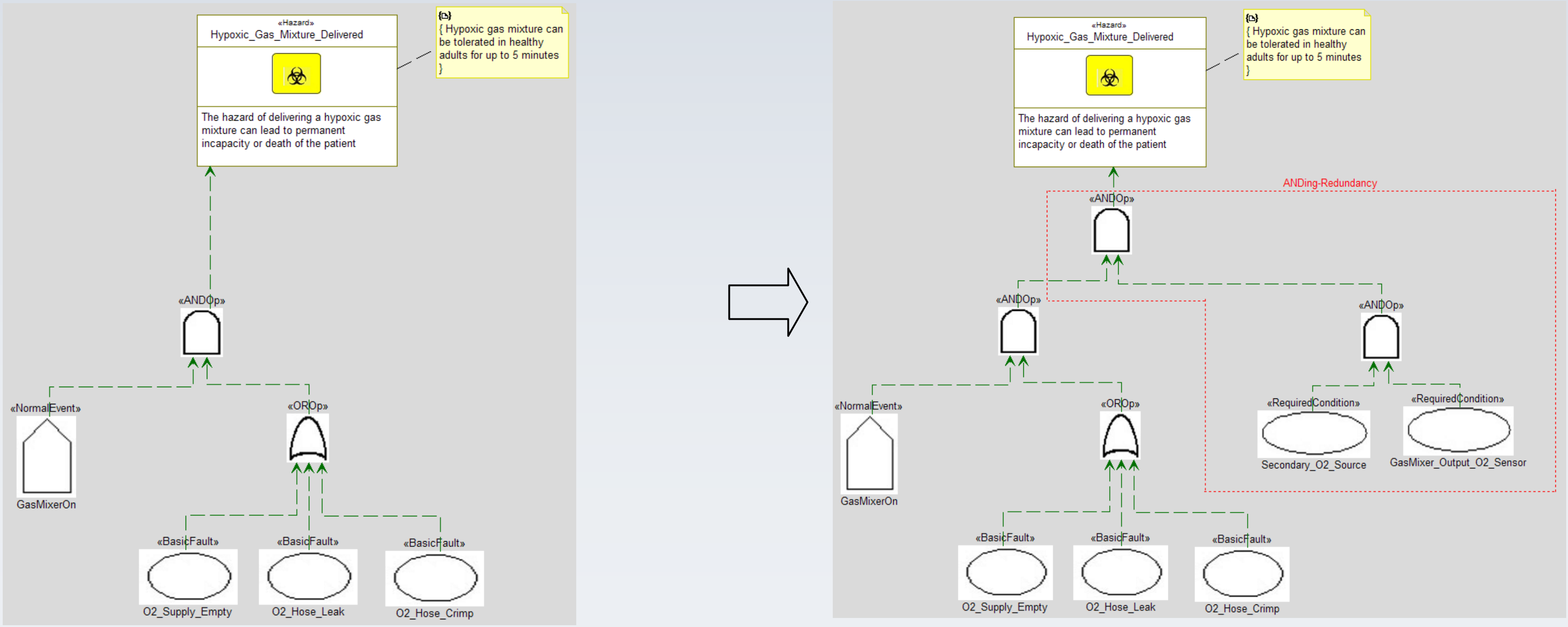
Fault Tree Analysis determines what combinations of conditions or events are necessary for a hazard condition to occur

Example Fault Tree Analysis



Design Redundancy for Safety

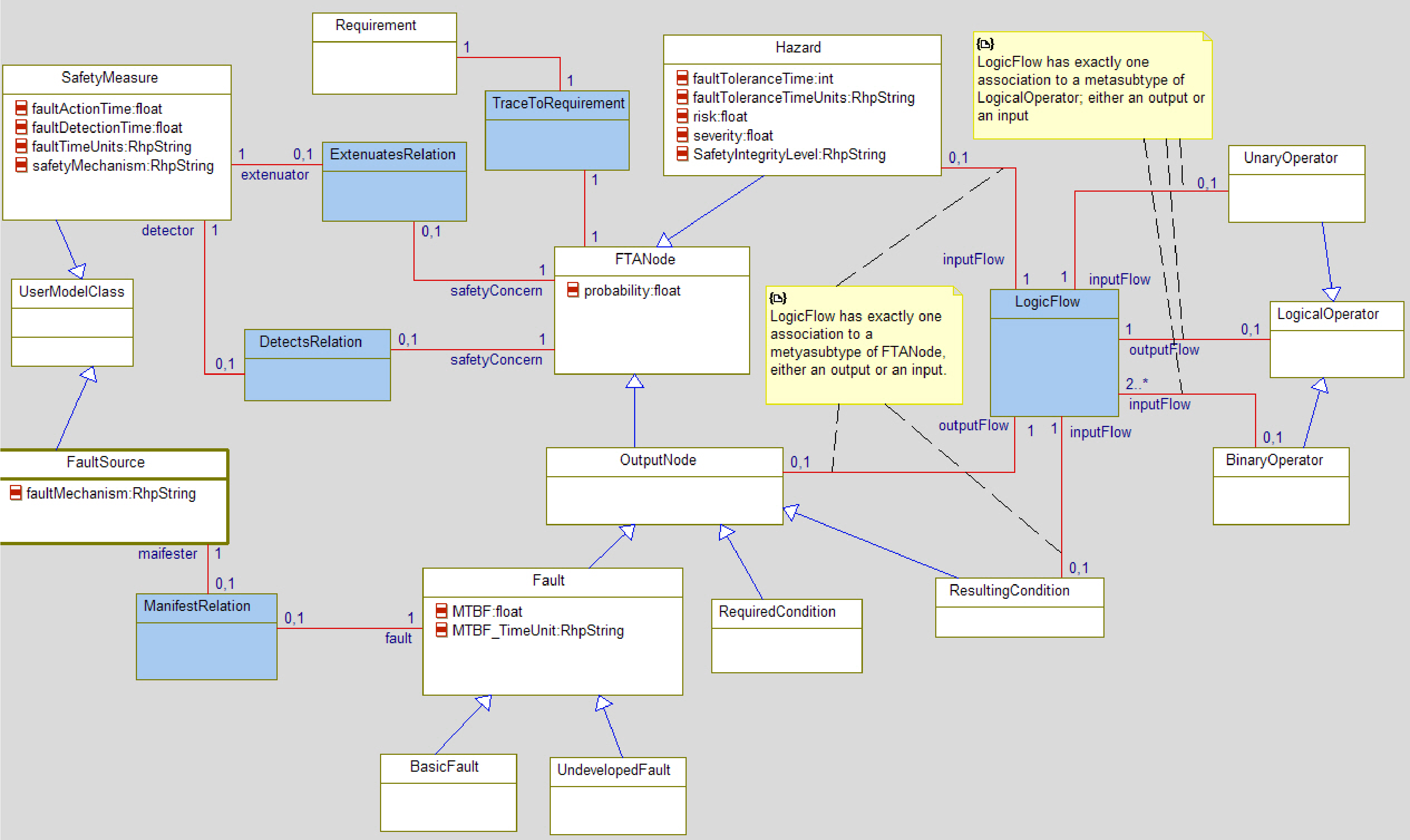
- The key to safe systems is to analyze the system and to identify the conditions and events that can lead to hazards
- Fault Tree Analysis (FTA) determines what logical combination of events and conditions lead to faults
- By adding “ANDing-redundancy”, architectural redundancy can be added


«Hazard»
Hypoxic_Gas_Mixture_Delivered
The hazard of delivering a hypoxic gas mixture can lead to permanent incapacity or death of the patient
{ Hypoxic gas mixture can be tolerated in healthy adults for up to 5 minutes }
«ANDOp»
«NormalEvent»
GasMixerOn
«OROp»
«BasicFault»
O2_Supply_Empty
«BasicFault»
O2_Hose_Leak
«BasicFault»
O2_Hose_Crimp
«Hazard»
Hypoxic_Gas_Mixture_Delivered
The hazard of delivering a hypoxic gas mixture can lead to permanent incapacity or death of the patient
{ Hypoxic gas mixture can be tolerated in healthy adults for up to 5 minutes }
ANDing-Redundancy
«ANDOp»
«ANDOp»
«ANDOp»
«NormalEvent»
GasMixerOn
«OROp»
«RequiredCondition»
Secondary_O2_Source
«RequiredCondition»
GasMixer_Output_O2_Sensor
«BasicFault»
O2_Supply_Empty
«BasicFault»
O2_Hose_Leak
«BasicFault»
O2_Hose_Crimp

Safety Metamodel

- The Safety Metamodel identifies and characterizes the important concepts (and their metadata) and their relations
- The metamodel serves as a blueprint for stereotypes in the profile

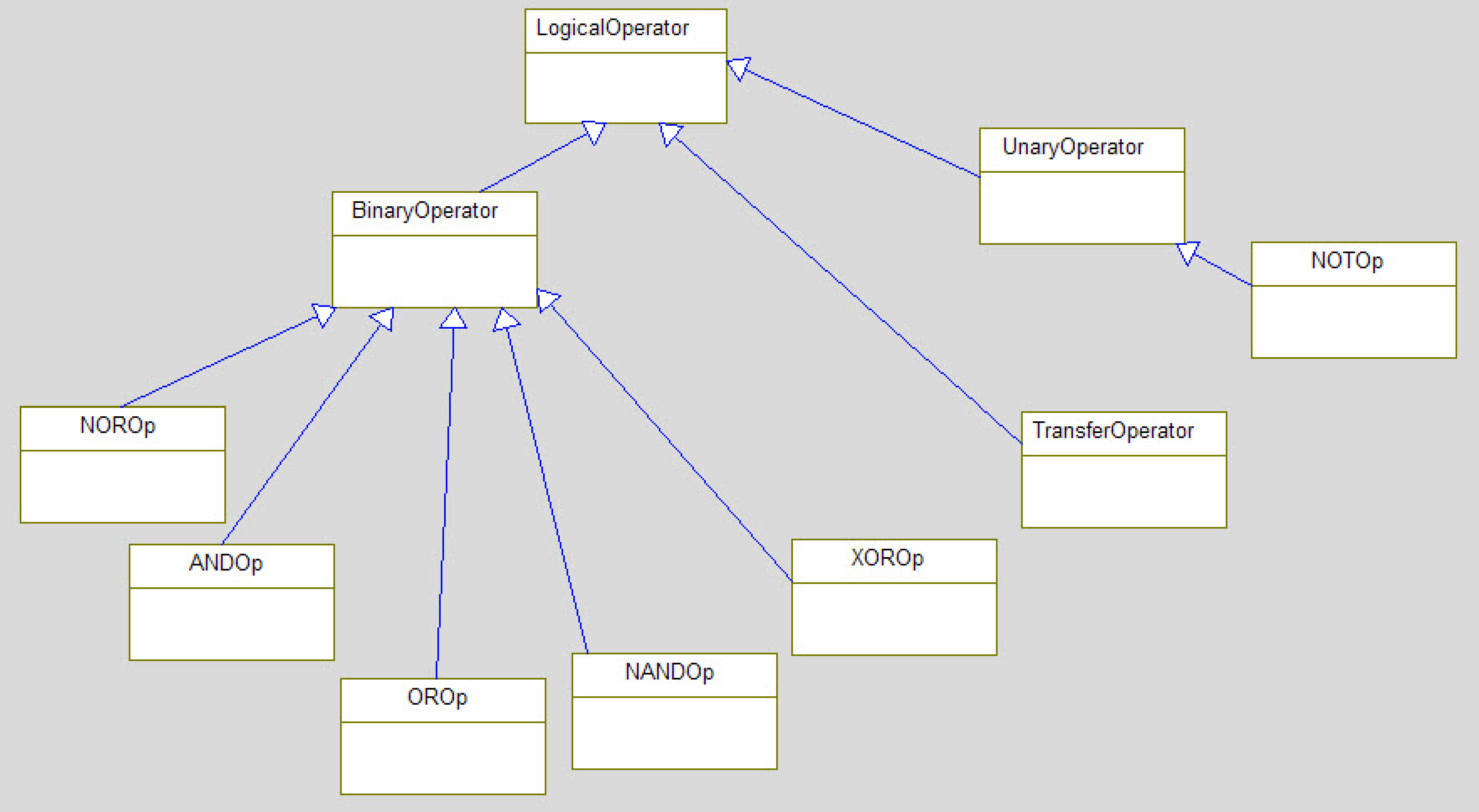
Safety Metamodel



LogicFlow has exactly one association to a metasubtype of LogicalOperator; either an output or an input

LogicFlow has exactly one association to a metyasubtype of FTANode, either an output or an input.

Safety Metamodel (Operators)



Metaclasses

- FTA Node
  - Description: An abstract metaclass providing the Probabilty metaattribute and various relations
  - Base metaclass: Class
  - Metadata
    - Probability [0.0 .. 1.0]
- Fault
  - Description: An abstract metaclass representing a non-conformant behavior of some kind
  - Base metaclass: FTA Node
  - Metadata
    - MTBF
    - MTBF Time Units
- Basic Fault
  - Description: A fault that cannot be further decomposed
  - Base metaclass: Fault
- Undeveloped Fault
  - Description: A fault that may be, but not is, further decomposed
  - Base metaclass: Fault

Metaclasses

- Required Condition
  - Description: A condition required as an input to an operator for a TRUE output
  - Base metaclass: FTANode
- Resulting Condition
  - Description: A condition resulting from the combination of other conditions and faults
  - Base metaclass: FTA Node
- Hazard
  - Description: A condition that inevitably leads to an accident or loss
  - Base metaclass: FTA Node
  - Metadata
    - Fault Tolerance Time
    - Fault Tolerance Time Units
    - Risk
    - Severity
    - Safety Integrity Level (SIL)

Metaclasses

- Requirement
  - Description: A capability or condition that must be satisfied by a system
  - Base metaclass: Class (from SysML)
  - Metadata
    - Text
    - Id
- Fault Source
  - Description: A model element that can manifest a fault
  - Base metaclass: Class
  - Metadata
    - Fault mechanism: string
- Safety Measure
  - Description: A model element that can detect or extenuate a fault
  - Base Metaclass: Class
  - Metadata
    - Fault action time
    - Fault detection time
    - Fault time units
    - Safety mechanism: string

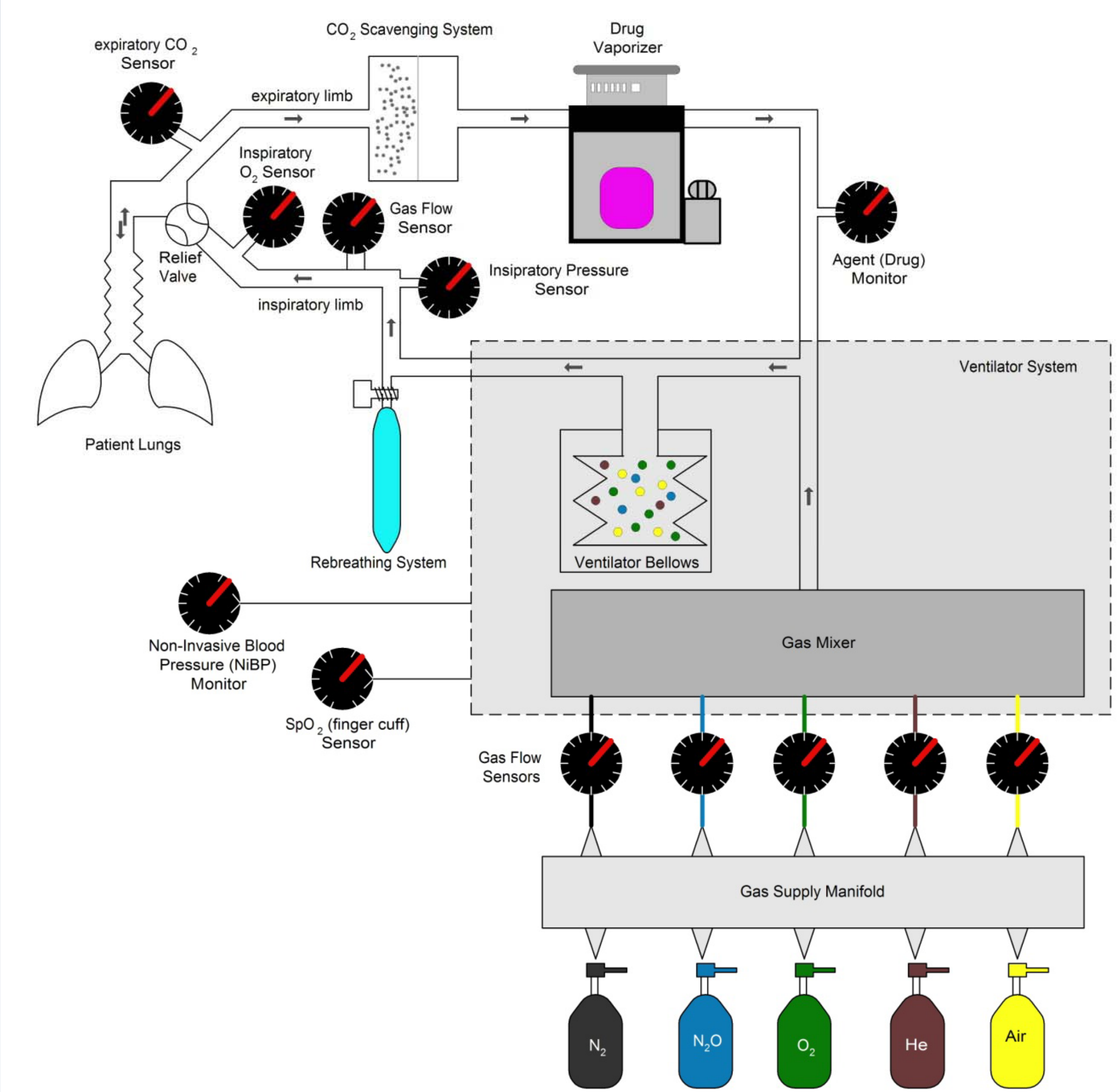
Metaclasses

- Logic Flow
  - Description: A “carrier” of boolean value
  - Base metaclass: Information flow
- Logical Operator
  - Description: An abstract metaclass for a function that performs logic on its inputs
  - Base metaclass: Class
- Unary Operator
  - Description: A logical operator with a single input and output
  - Base metaclass: Logical Operator
- Binary Operator
  - Description: A logical operator that takes two (or more) inputs to produce a single output
  - Base metaclass: Logical Operator
- Transfer Operator
  - Description: A binary operator whose purpose is to “connect” logic across multiple diagrams
  - Base metaclass: Binary operator

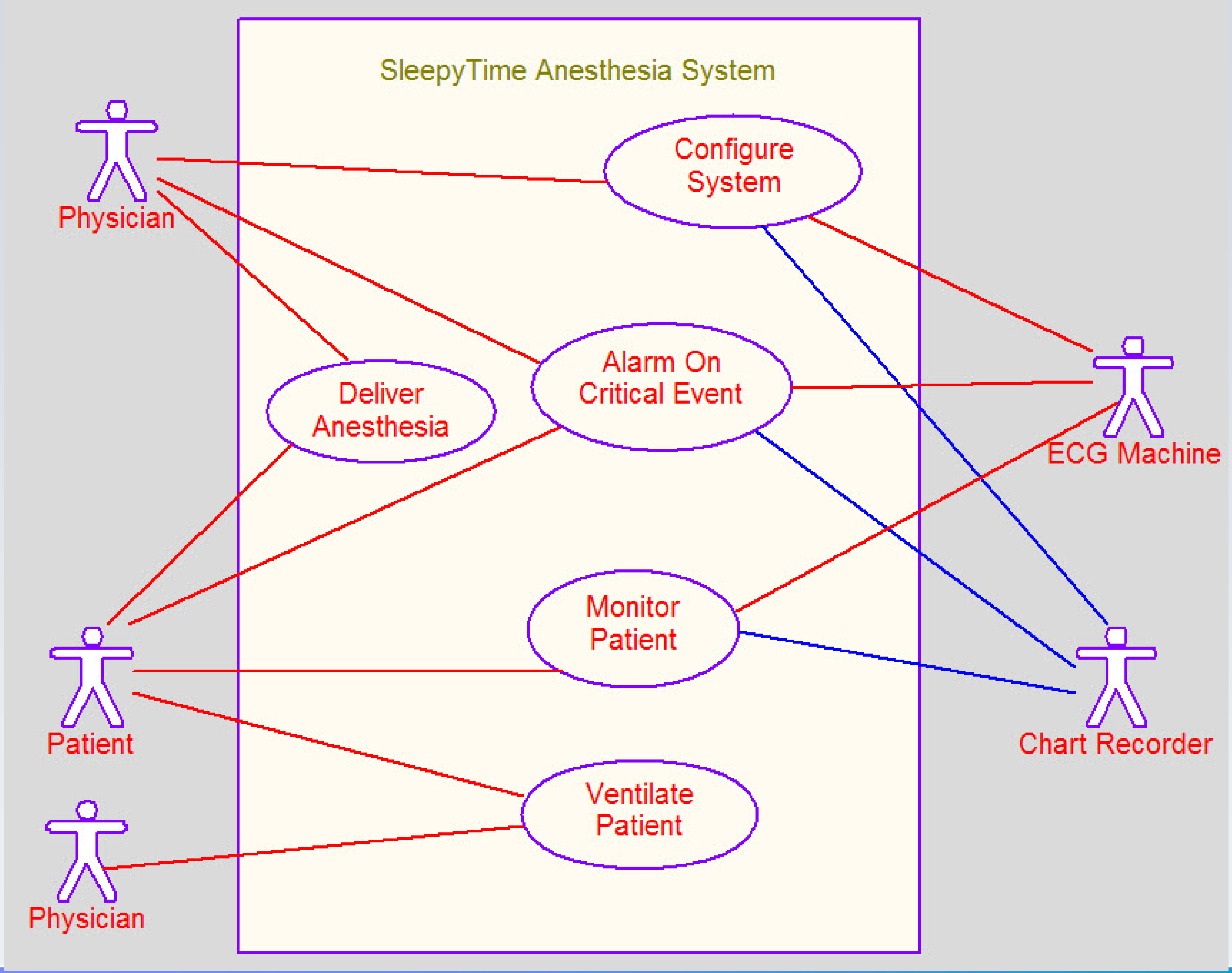
Metaclasses

- TraceToReq
  - Description: A relation from an FTA node to a requirement
  - Base metaclass: Dependency
- ManifestsRelation
  - Description: A relation from a Fault to a Fault Source or Class
  - Base metaclass: Dependency
- DetectsRelation
  - Description: A relation from an FTA node to a Safety Measure that detects a fault
  - Base metaclass: Dependency
- ExtenuatesRelation
  - Description: A relation from an FTA node to a Safety Measure that removes, mitigates, or extenuates a fault
  - Base metaclass: Dependency

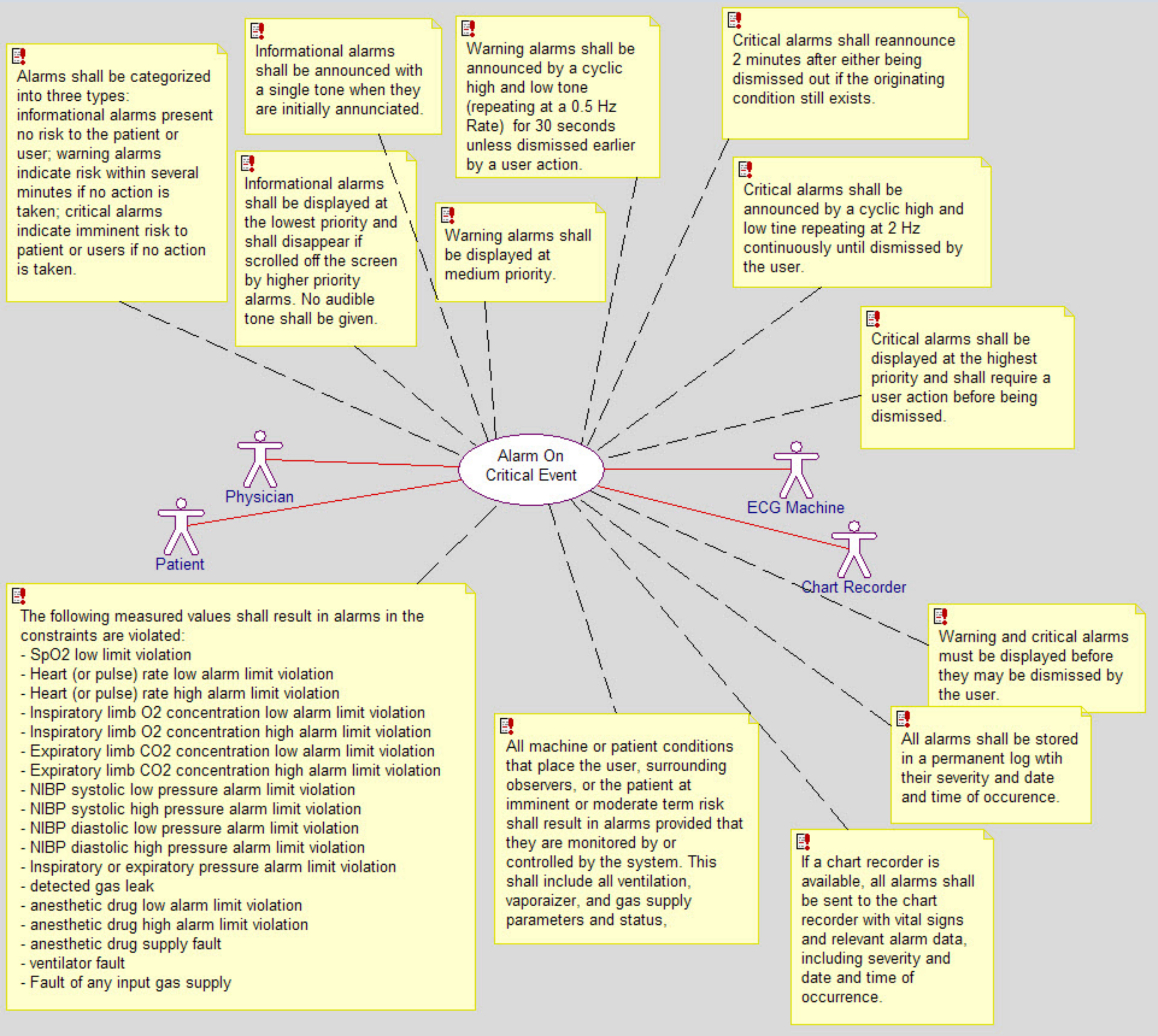
Safety Example: SleepyTime Anesthesia Machine



System Use Case Model



Alarm Requirements



Alarms shall be categorized into three types: informational alarms present no risk to the patient or user; warning alarms indicate risk within several minutes if no action is taken; critical alarms indicate imminent risk to patient or users if no action is taken.

Informational alarms shall be announced with a single tone when they are initially annunciated.

Informational alarms shall be displayed at the lowest priority and shall disappear if scrolled off the screen by higher priority alarms. No audible tone shall be given.

Warning alarms shall be announced by a cyclic high and low tone (repeating at a 0.5 Hz Rate) for 30 seconds unless dismissed earlier by a user action.

Warning alarms shall be displayed at medium priority.

Critical alarms shall reannounce 2 minutes after either being dismissed out if the originating condition still exists.

Critical alarms shall be announced by a cyclic high and low tine repeating at 2 Hz continuously until dismissed by the user.

Critical alarms shall be displayed at the highest priority and shall require a user action before being dismissed.

The following measured values shall result in alarms in the constraints are violated:
- SpO2 low limit violation
- Heart (or pulse) rate low alarm limit violation
- Heart (or pulse) rate high alarm limit violation
- Inspiratory limb O2 concentration low alarm limit violation
- Inspiratory limb O2 concentration high alarm limit violation
- Expiratory limb CO2 concentration low alarm limit violation
- Expiratory limb CO2 concentration high alarm limit violation
- NIBP systolic low pressure alarm limit violation
- NIBP systolic high pressure alarm limit violation
- NIBP diastolic low pressure alarm limit violation
- NIBP diastolic high pressure alarm limit violation
- Inspiratory or expiratory pressure alarm limit violation
- detected gas leak
- anesthetic drug low alarm limit violation
- anesthetic drug high alarm limit violation
- anesthetic drug supply fault
- ventilator fault
- Fault of any input gas supply

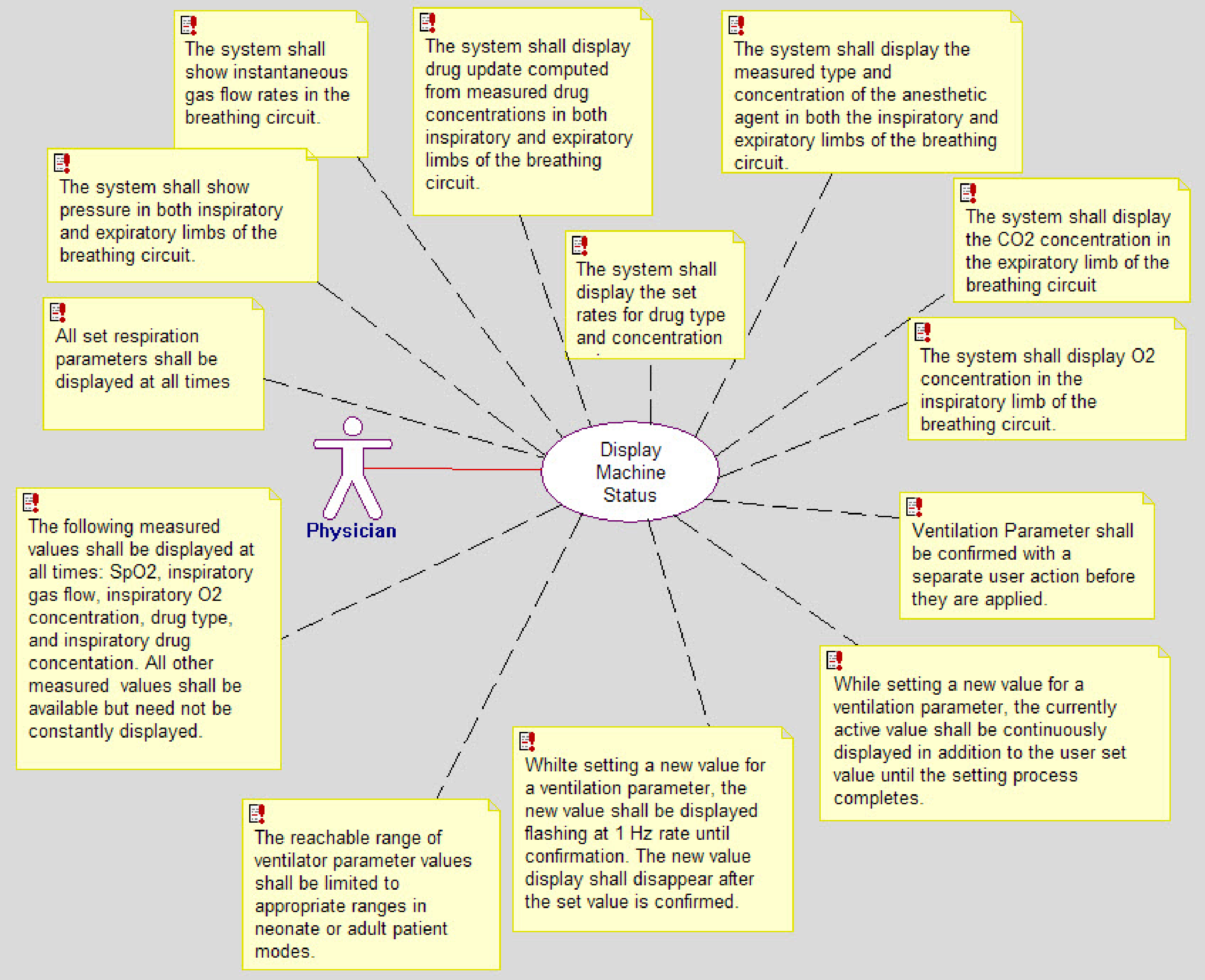
All machine or patient conditions that place the user, surrounding observers, or the patient at imminent or moderate term risk shall result in alarms provided that they are monitored by or controlled by the system. This shall include all ventilation, vaporaizer, and gas supply parameters and status,

If a chart recorder is available, all alarms shall be sent to the chart recorder with vital signs and relevant alarm data, including severity and date and time of occurrence.

All alarms shall be stored in a permanent log wtih their severity and date and time of occurence.

Warning and critical alarms must be displayed before they may be dismissed by the user.

Display Requirements



The system shall show instantaneous gas flow rates in the breathing circuit.

The system shall display drug update computed from measured drug concentrations in both inspiratory and expiratory limbs of the breathing circuit.

The system shall display the measured type and concentration of the anesthetic agent in both the inspiratory and expiratory limbs of the breathing circuit.

The system shall show pressure in both inspiratory and expiratory limbs of the breathing circuit.

The system shall display the set rates for drug type and concentration

The system shall display the CO2 concentration in the expiratory limb of the breathing circuit

All set respiration parameters shall be displayed at all times

The system shall display O2 concentration in the inspiratory limb of the breathing circuit.

The following measured values shall be displayed at all times: SpO2, inspiratory gas flow, inspiratory O2 concentration, drug type, and inspiratory drug concentation. All other measured values shall be available but need not be constantly displayed.

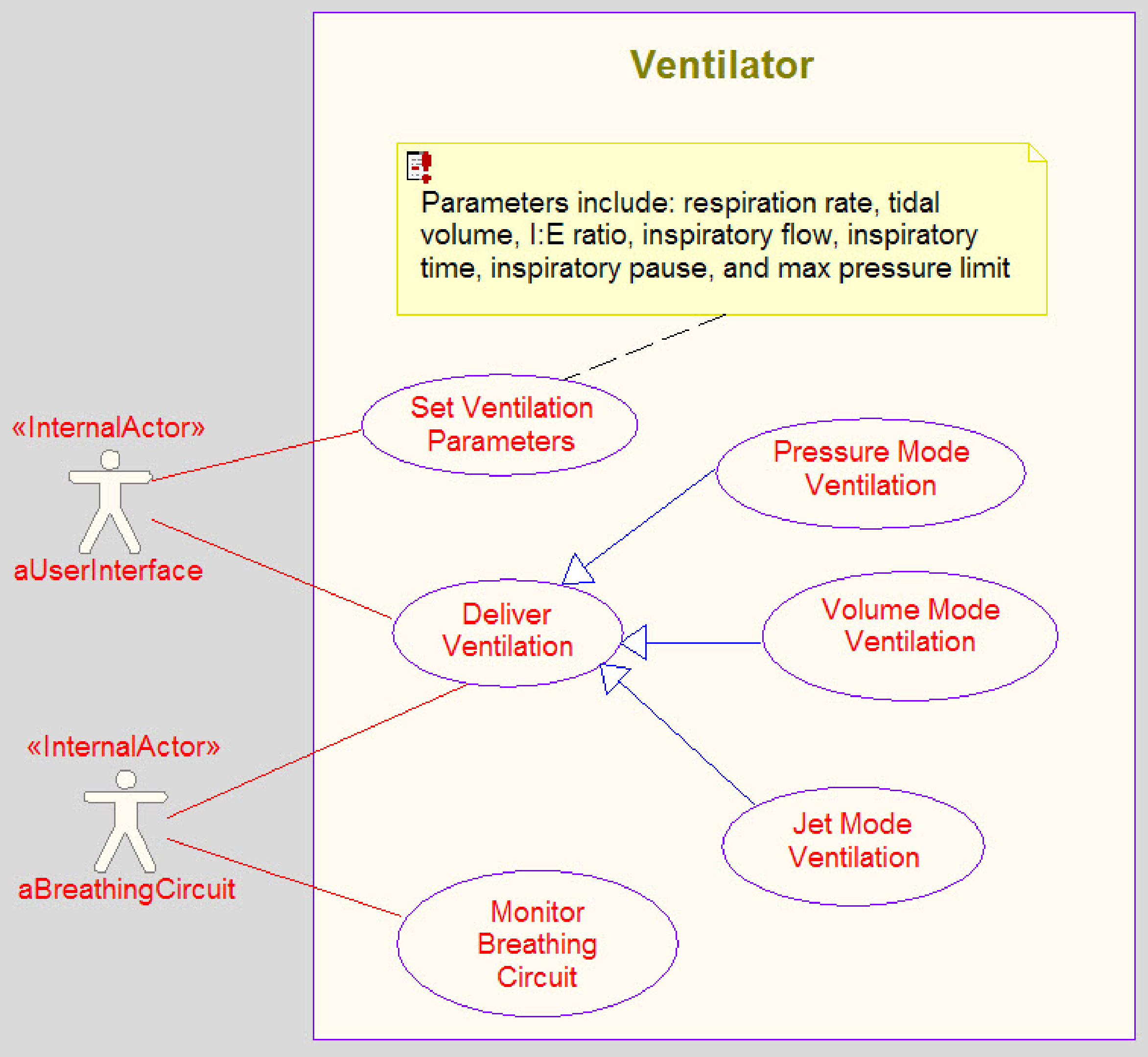
Ventilation Parameter shall be confirmed with a separate user action before they are applied.

While setting a new value for a ventilation parameter, the currently active value shall be continuously displayed in addition to the user set value until the setting process completes.

Whilte setting a new value for a ventilation parameter, the new value shall be displayed flashing at 1 Hz rate until confirmation. The new value display shall disappear after the set value is confirmed.

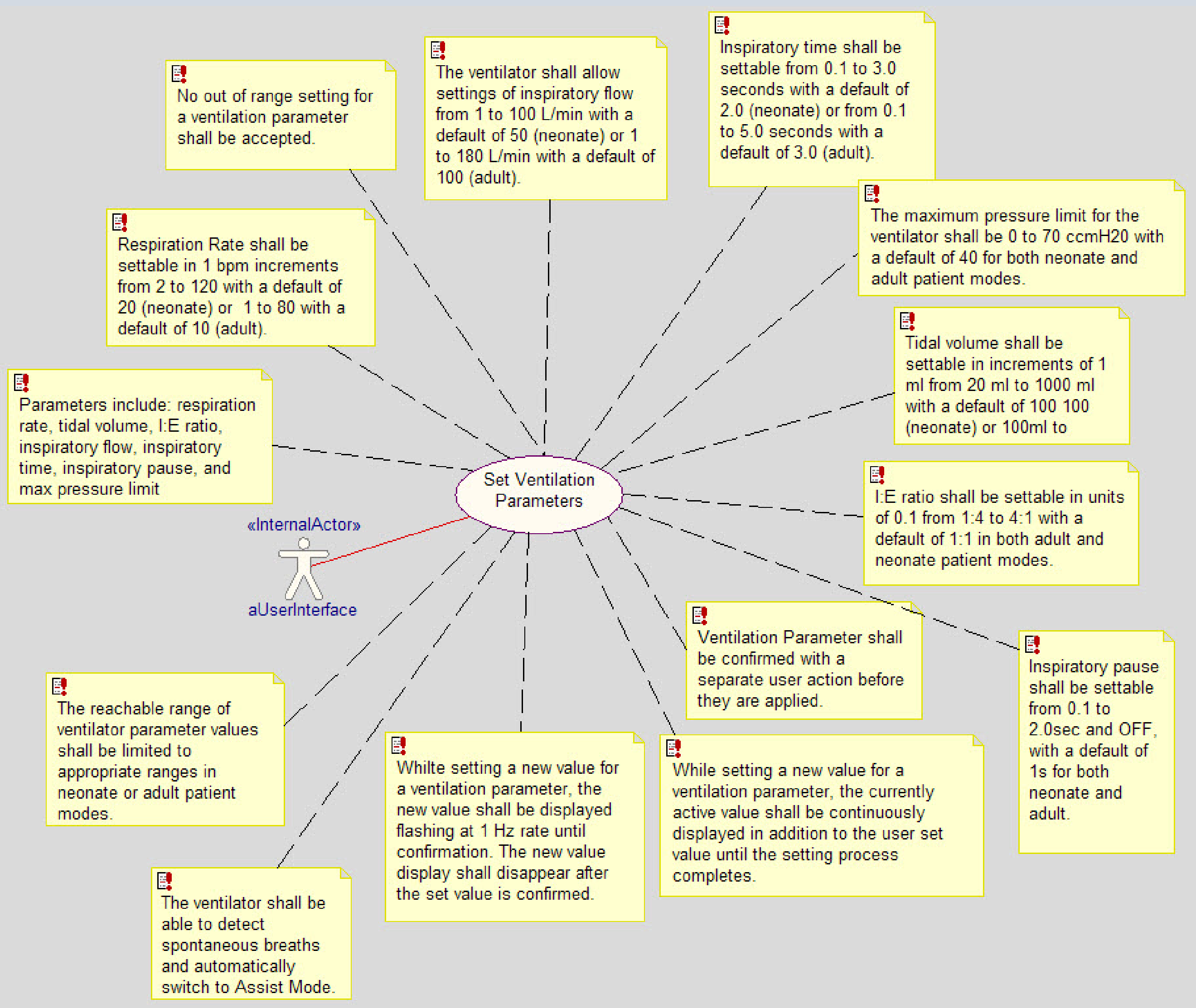
The reachable range of ventilator parameter values shall be limited to appropriate ranges in neonate or adult patient modes.

Ventilator Subsystem Use Case Model



Parameters include: respiration rate, tidal volume, I:E ratio, inspiratory flow, inspiratory time, inspiratory pause, and max pressure limit

Ventilator Requirements



No out of range setting for a ventilation parameter shall be accepted.

The ventilator shall allow settings of inspiratory flow from 1 to 100 L/min with a default of 50 (neonate) or 1 to 180 L/min with a default of 100 (adult).

Inspiratory time shall be settable from 0.1 to 3.0 seconds with a default of 2.0 (neonate) or from 0.1 to 5.0 seconds with a default of 3.0 (adult).

The maximum pressure limit for the ventilator shall be 0 to 70 ccmH20 with a default of 40 for both neonate and adult patient modes.

Respiration Rate shall be settable in 1 bpm increments from 2 to 120 with a default of 20 (neonate) or 1 to 80 with a default of 10 (adult).

Tidal volume shall be settable in increments of 1 ml from 20 ml to 1000 ml with a default of 100 100 (neonate) or 100ml to

Parameters include: respiration rate, tidal volume, I:E ratio, inspiratory flow, inspiratory time, inspiratory pause, and max pressure limit

I:E ratio shall be settable in units of 0.1 from 1:4 to 4:1 with a default of 1:1 in both adult and neonate patient modes.

Ventilation Parameter shall be confirmed with a separate user action before they are applied.

Inspiratory pause shall be settable from 0.1 to 2.0sec and OFF, with a default of 1s for both neonate and adult.

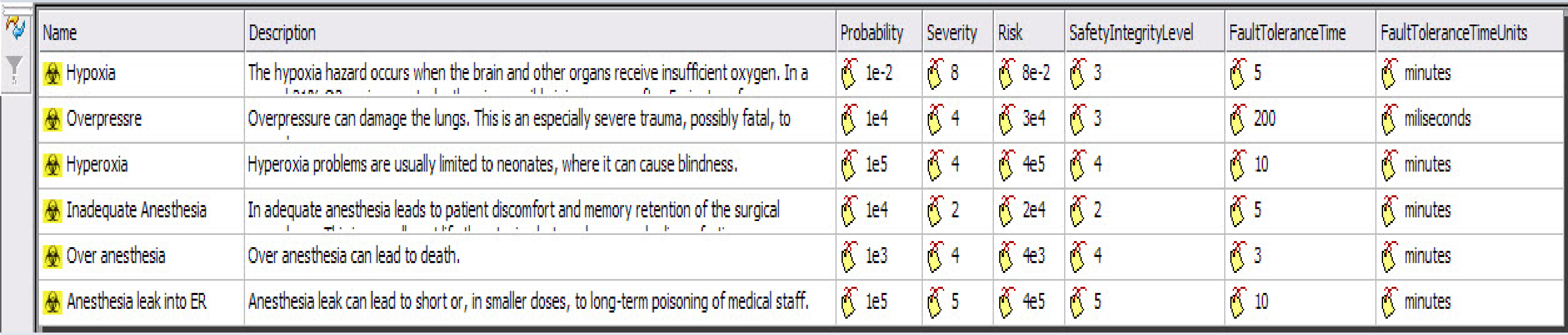
The reachable range of ventilator parameter values shall be limited to appropriate ranges in neonate or adult patient modes.

Whilte setting a new value for a ventilation parameter, the new value shall be displayed flashing at 1 Hz rate until confirmation. The new value display shall disappear after the set value is confirmed.

While setting a new value for a ventilation parameter, the currently active value shall be continuously displayed in addition to the user set value until the setting process completes.

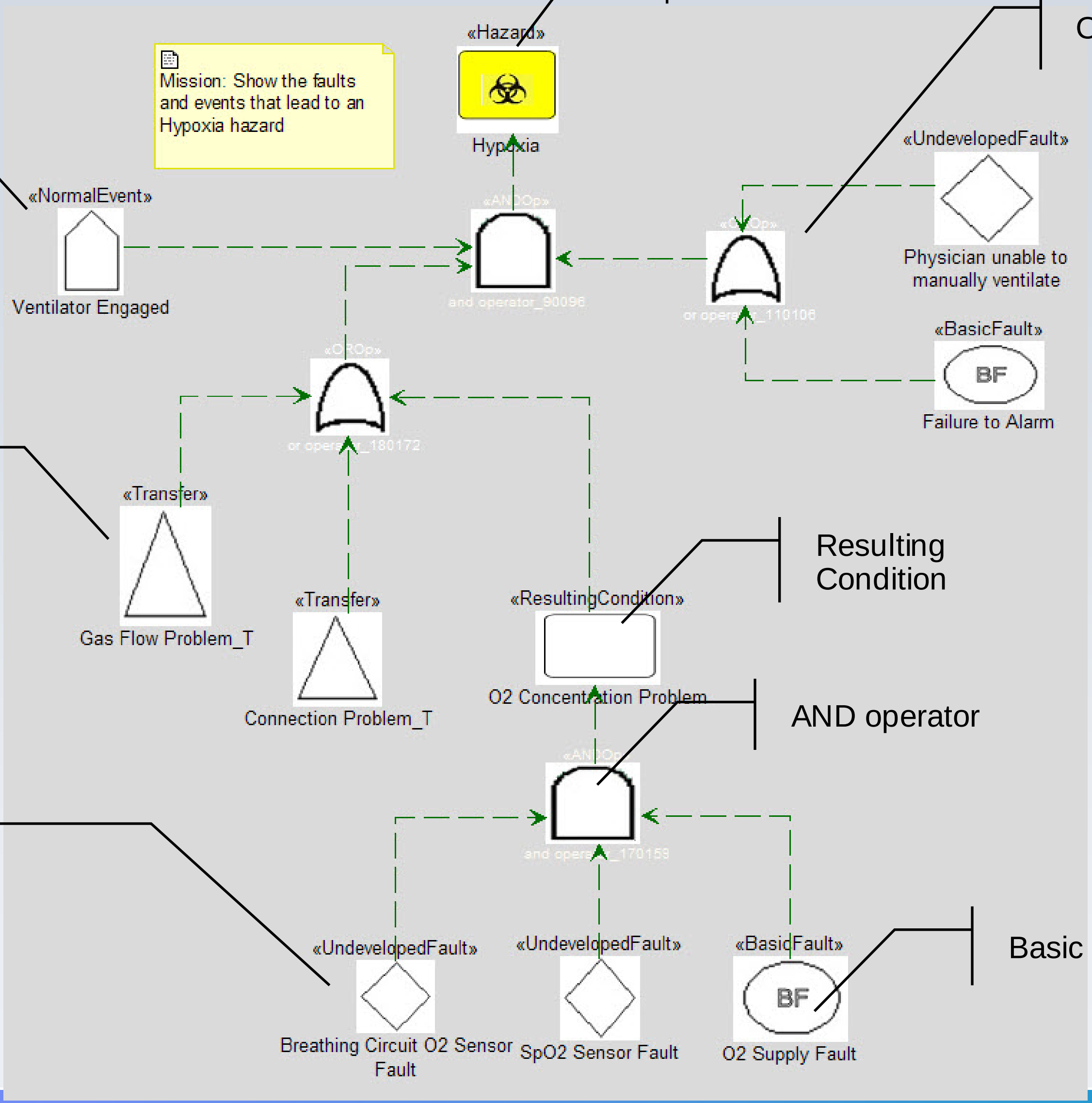
The ventilator shall be able to detect spontaneous breaths and automatically switch to Assist Mode.

Hazard Table (generated)



Name	Description	Probability	Severity	Risk	SafetyIntegrityLevel	FaultToleranceTime	FaultToleranceTimeUnits
Hypoxia	The hypoxia hazard occurs when the brain and other organs receive insufficient oxygen. In a	1e-2	8	8e-2	3	5	minutes
Overpressre	Overpressure can damage the lungs. This is an especially severe trauma, possibly fatal, to	1e4	4	3e4	3	200	miliseconds
Hyperoxia	Hyperoxia problems are usually limited to neonates, where it can cause blindness.	1e5	4	4e5	4	10	minutes
Inadequate Anesthesia	In adequate anesthesia leads to patient discomfort and memory retention of the surgical	1e4	2	2e4	2	5	minutes
Over anesthesia	Over anesthesia can lead to death.	1e3	4	4e3	4	3	minutes
Anesthesia leak into ER	Anesthesia leak can lead to short or, in smaller doses, to long-term poisoning of medical staff.	1e5	5	4e5	5	10	minutes

FTA Hypoxia Hazard



Hazard

OR operator

Normal Event

Transfer Operator

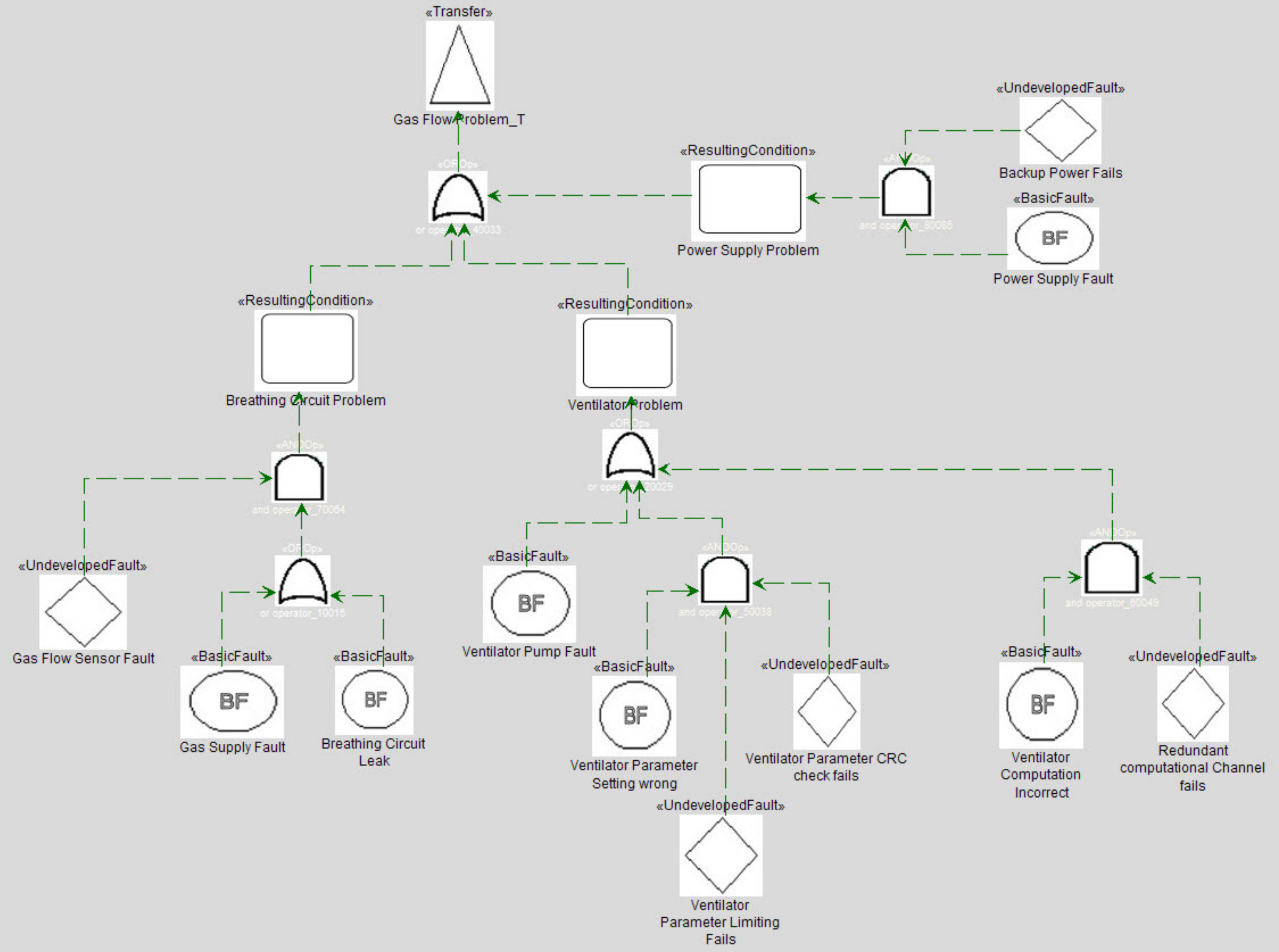
Resulting Condition

AND operator

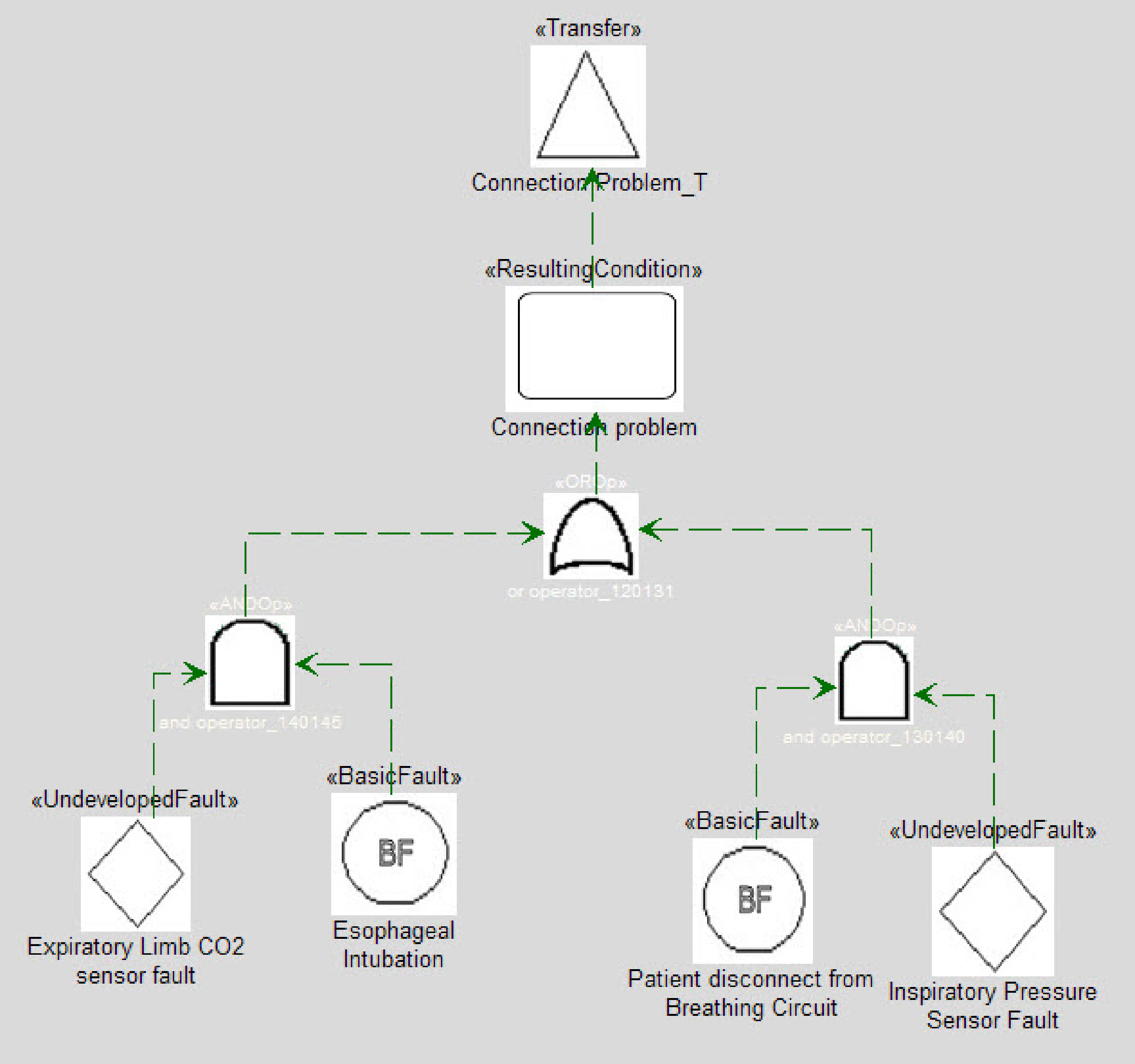
Undeveloped Fault

Basic Fault

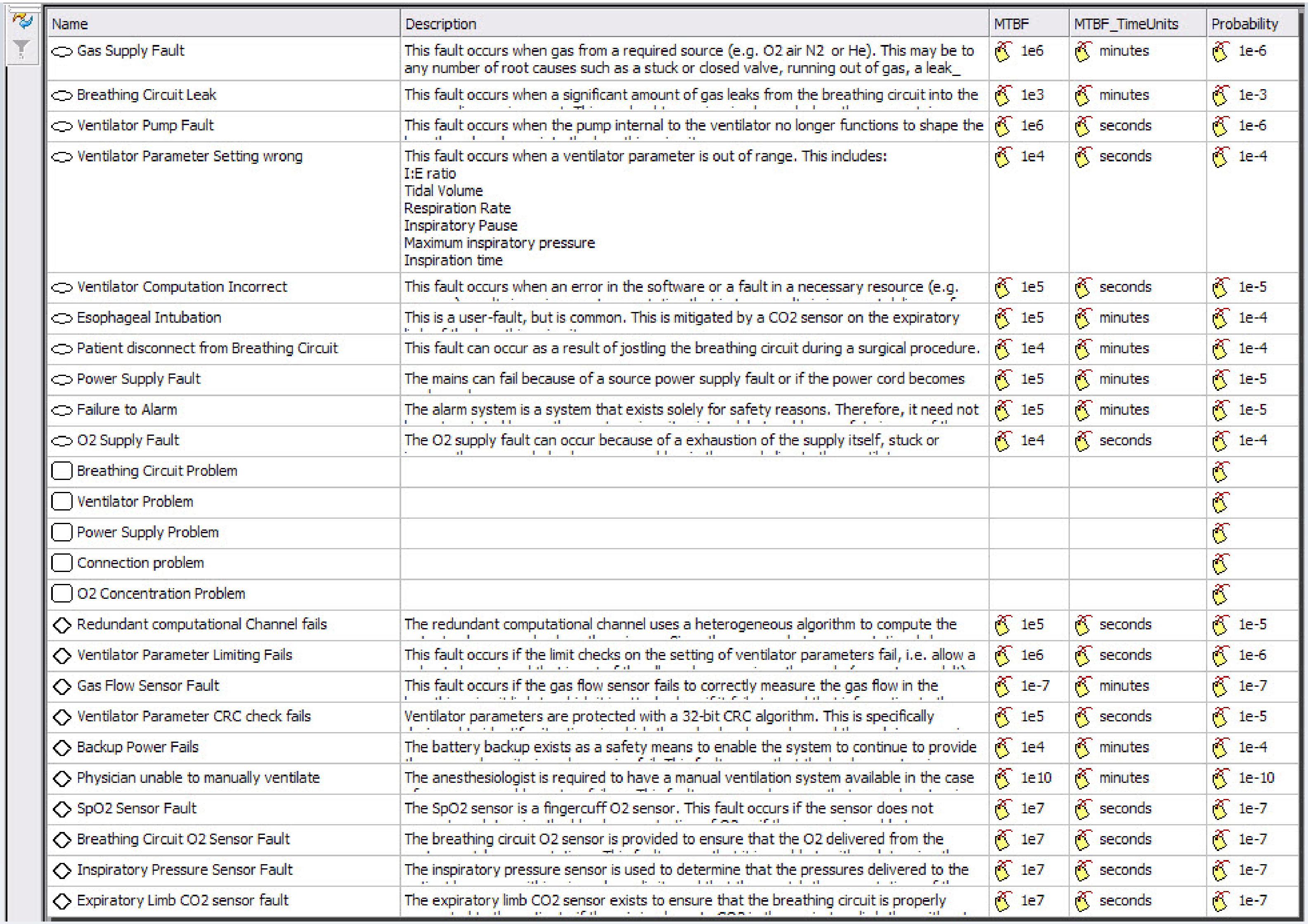
FTA Gas Flow Problem



FTA Gas Connection Problem

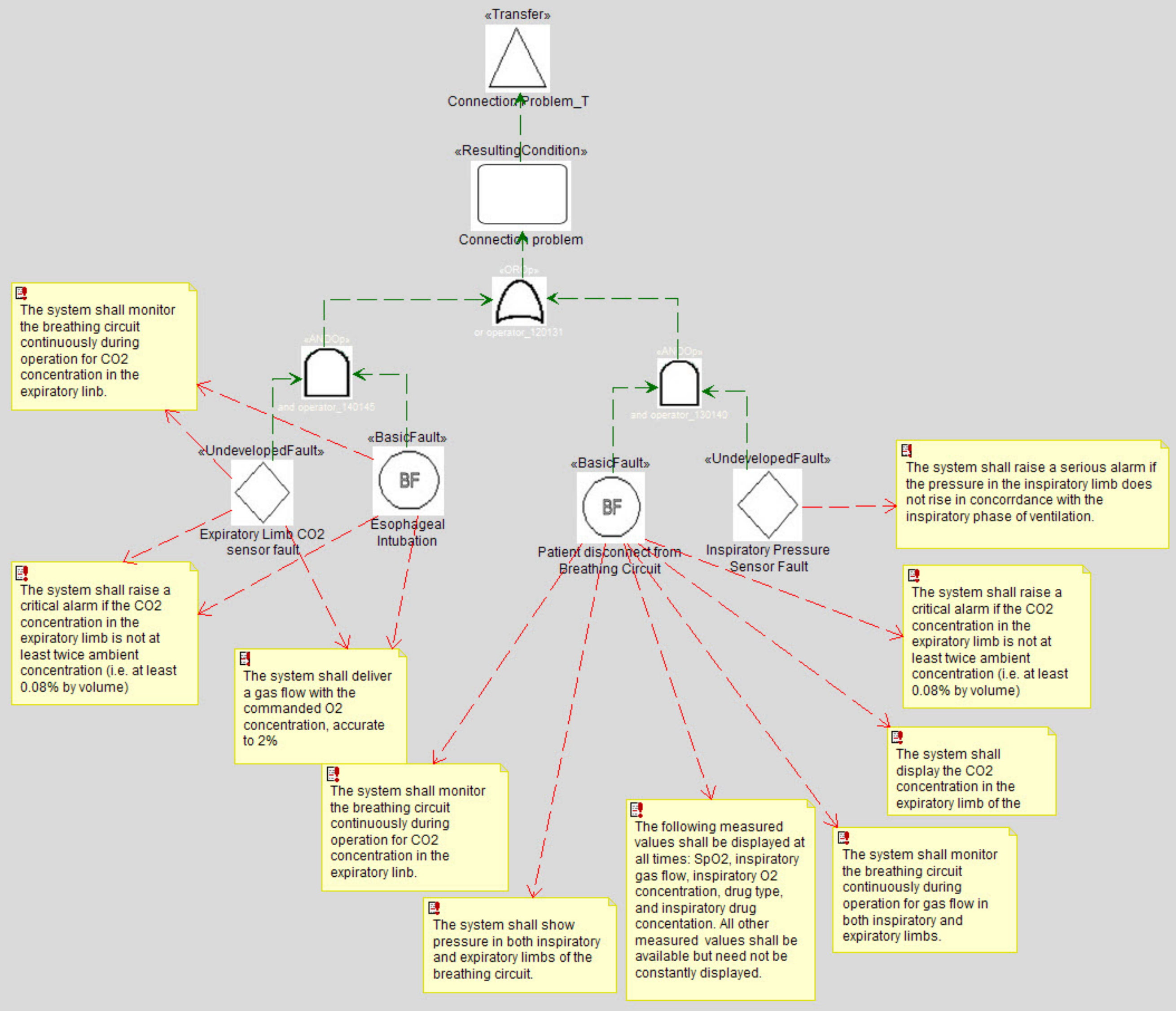


Fault Table



Name	Description	MTBF	MTBF_TimeUnits	Probability
⬭ Gas Supply Fault	This fault occurs when gas from a required source (e.g. O2 air N2 or He). This may be to any number of root causes such as a stuck or closed valve, running out of gas, a leak_	1e6	minutes	1e-6
⬭ Breathing Circuit Leak	This fault occurs when a significant amount of gas leaks from the breathing circuit into the	1e3	minutes	1e-3
⬭ Ventilator Pump Fault	This fault occurs when the pump internal to the ventilator no longer functions to shape the	1e6	seconds	1e-6
⬭ Ventilator Parameter Setting wrong	This fault occurs when a ventilator parameter is out of range. This includes: I:E ratio Tidal Volume Respiration Rate Inspiratory Pause Maximum inspiratory pressure Inspiration time	1e4	seconds	1e-4
⬭ Ventilator Computation Incorrect	This fault occurs when an error in the software or a fault in a necessary resource (e.g.	1e5	seconds	1e-5
⬭ Esophageal Intubation	This is a user-fault, but is common. This is mitigated by a CO2 sensor on the expiratory	1e5	minutes	1e-4
⬭ Patient disconnect from Breathing Circuit	This fault can occur as a result of jostling the breathing circuit during a surgical procedure.	1e4	minutes	1e-4
⬭ Power Supply Fault	The mains can fail because of a source power supply fault or if the power cord becomes	1e5	minutes	1e-5
⬭ Failure to Alarm	The alarm system is a system that exists solely for safety reasons. Therefore, it need not	1e5	minutes	1e-5
⬭ O2 Supply Fault	The O2 supply fault can occur because of a exhaustion of the supply itself, stuck or	1e4	seconds	1e-4
□ Breathing Circuit Problem				
□ Ventilator Problem				
□ Power Supply Problem				
□ Connection problem				
□ O2 Concentration Problem				
◇ Redundant computational Channel fails	The redundant computational channel uses a heterogeneous algorithm to compute the	1e5	seconds	1e-5
◇ Ventilator Parameter Limiting Fails	This fault occurs if the limit checks on the setting of ventilator parameters fail, i.e. allow a	1e6	seconds	1e-6
◇ Gas Flow Sensor Fault	This fault occurs if the gas flow sensor fails to correctly measure the gas flow in the	1e-7	minutes	1e-7
◇ Ventilator Parameter CRC check fails	Ventilator parameters are protected with a 32-bit CRC algorithm. This is specifically	1e5	seconds	1e-5
◇ Backup Power Fails	The battery backup exists as a safety means to enable the system to continue to provide	1e4	minutes	1e-4
◇ Physician unable to manually ventilate	The anesthesiologist is required to have a manual ventilation system available in the case	1e10	minutes	1e-10
◇ SpO2 Sensor Fault	The SpO2 sensor is a fingercuff O2 sensor. This fault occurs if the sensor does not	1e7	seconds	1e-7
◇ Breathing Circuit O2 Sensor Fault	The breathing circuit O2 sensor is provided to ensure that the O2 delivered from the	1e7	seconds	1e-7
◇ Inspiratory Pressure Sensor Fault	The inspiratory pressure sensor is used to determine that the pressures delivered to the	1e7	seconds	1e-7
◇ Expiratory Limb CO2 sensor fault	The expiratory limb CO2 sensor exists to ensure that the breathing circuit is properly	1e7	seconds	1e-7

Connecting FTA to Requirements (TraceToReq)



The system shall monitor the breathing circuit continuously during operation for CO2 concentration in the expiratory linb.

The system shall raise a critical alarm if the CO2 concentration in the expiratory limb is not at least twice ambient concentration (i.e. at least 0.08% by volume)

The system shall deliver a gas flow with the commanded O2 concentration, accurate to 2%

The system shall monitor the breathing circuit continuously during operation for CO2 concentration in the expiratory linb.

The system shall show pressure in both inspiratory and expiratory limbs of the breathing circuit.

The following measured values shall be displayed at all times: SpO2, inspiratory gas flow, inspiratory O2 concentration, drug type, and inspiratory drug concentation. All other measured values shall be available but need not be constantly displayed.

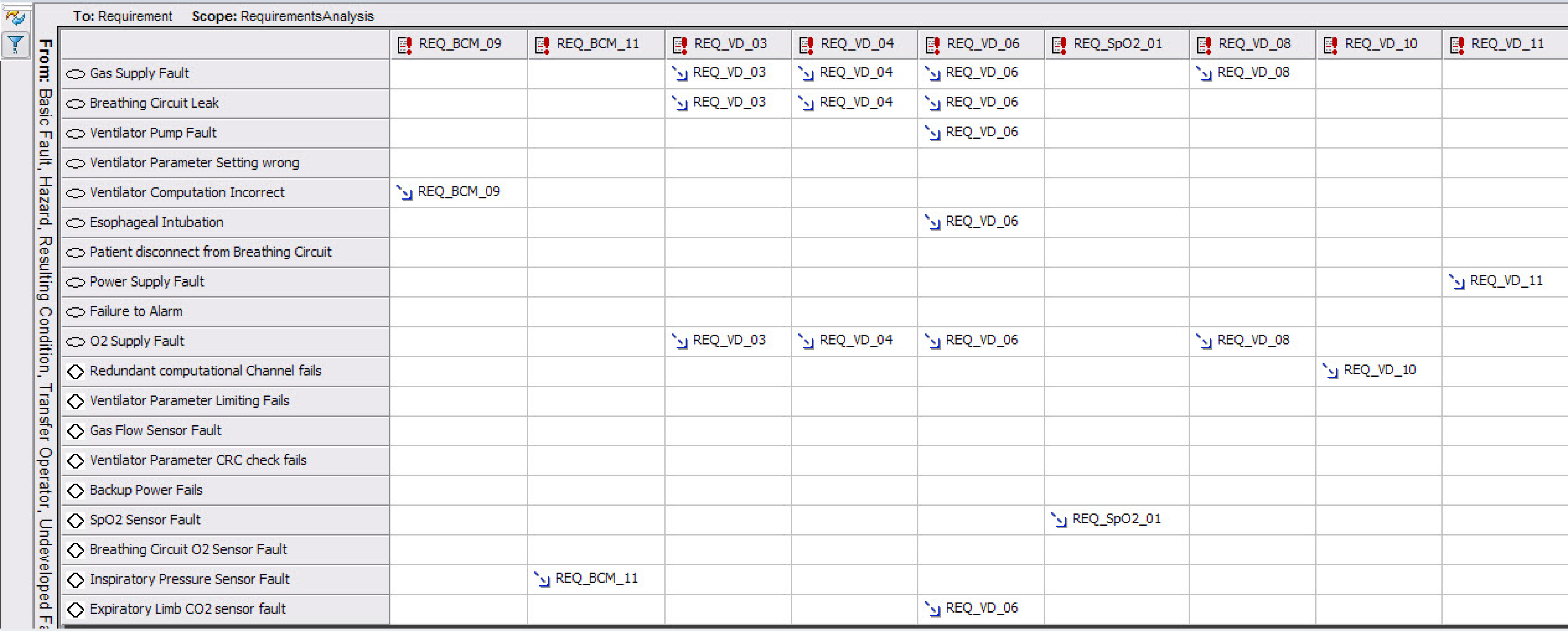
The system shall raise a serious alarm if the pressure in the inspiratory limb does not rise in concorrdance with the inspiratory phase of ventilation.

The system shall raise a critical alarm if the CO2 concentration in the expiratory limb is not at least twice ambient concentration (i.e. at least 0.08% by volume)

The system shall display the CO2 concentration in the expiratory limb of the

The system shall monitor the breathing circuit continuously during operation for gas flow in both inspiratory and expiratory limbs.

Fault-Requirement Matrix (generated)

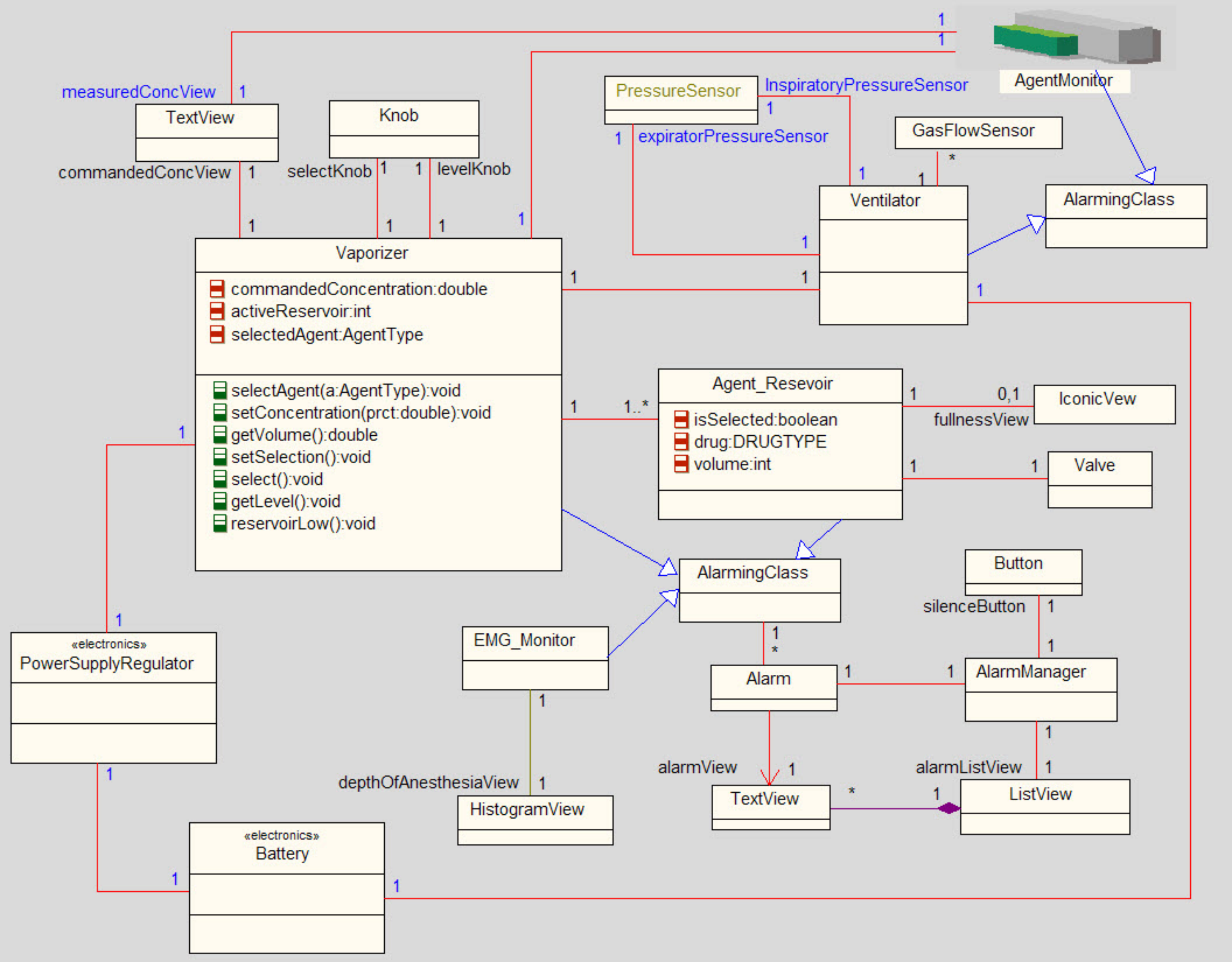


To: Requirement **Scope:** RequirementsAnalysis

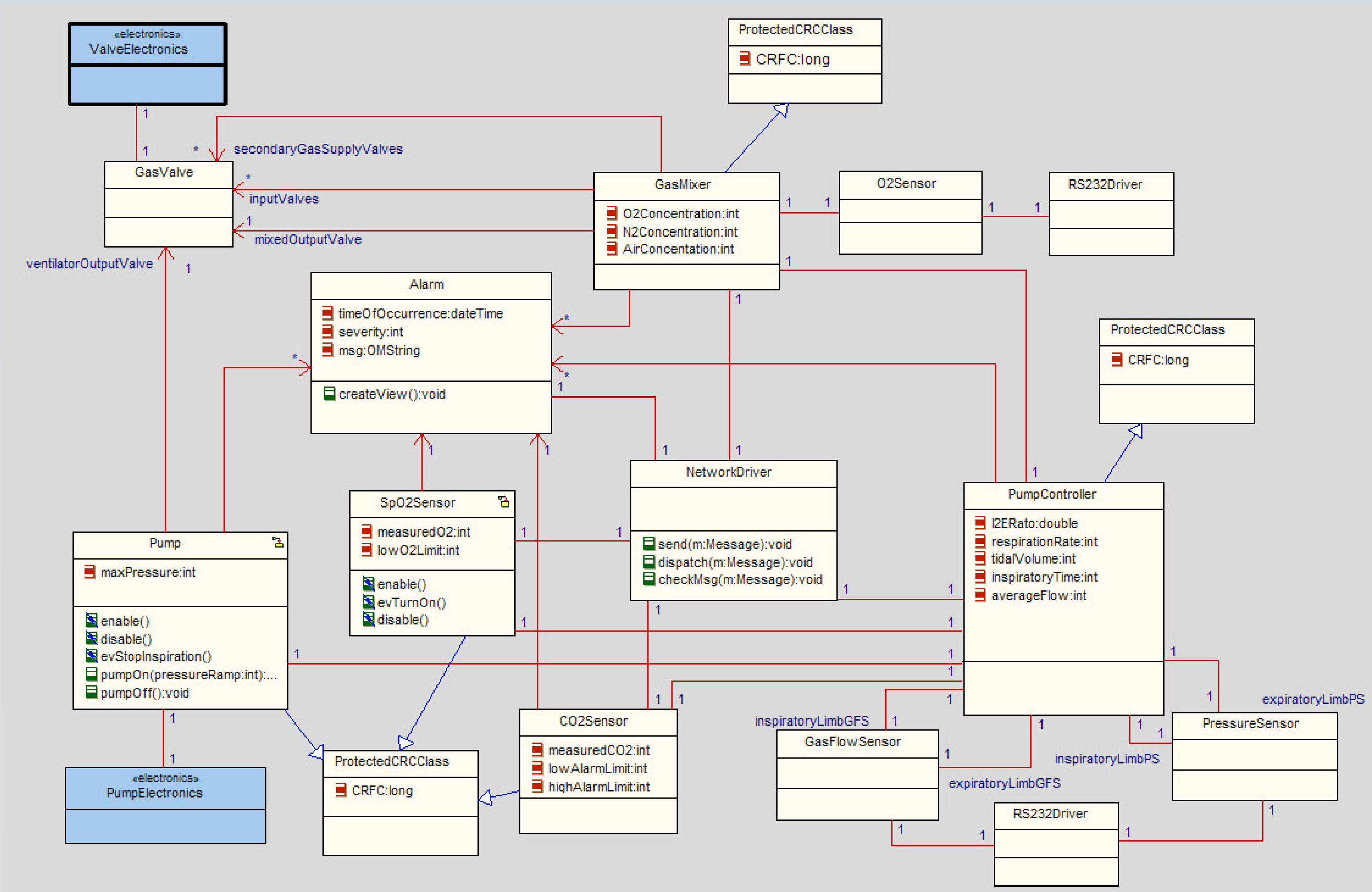
From: Basic Fault, Hazard, Resulting Condition, Transfer Operator, Undeveloped Fa

	REQ_BCM_09	REQ_BCM_11	REQ_VD_03	REQ_VD_04	REQ_VD_06	REQ_SpO2_01	REQ_VD_08	REQ_VD_10	REQ_VD_11
Gas Supply Fault			REQ_VD_03	REQ_VD_04	REQ_VD_06		REQ_VD_08		
Breathing Circuit Leak			REQ_VD_03	REQ_VD_04	REQ_VD_06				
Ventilator Pump Fault					REQ_VD_06				
Ventilator Parameter Setting wrong									
Ventilator Computation Incorrect	REQ_BCM_09								
Esophageal Intubation					REQ_VD_06				
Patient disconnect from Breathing Circuit									
Power Supply Fault									REQ_VD_11
Failure to Alarm									
O2 Supply Fault			REQ_VD_03	REQ_VD_04	REQ_VD_06		REQ_VD_08		
Redundant computational Channel fails								REQ_VD_10	
Ventilator Parameter Limiting Fails									
Gas Flow Sensor Fault									
Ventilator Parameter CRC check fails									
Backup Power Fails									
SpO2 Sensor Fault						REQ_SpO2_01			
Breathing Circuit O2 Sensor Fault									
Inspiratory Pressure Sensor Fault		REQ_BCM_11							
Expiratory Limb CO2 sensor fault					REQ_VD_06				

Analysis Model of the SleepyTime Machine



Analysis Model of the Ventilator Subsystem



«electronics» ValveElectronics

GasValve

secondaryGasSupplyValves

inputValves

mixedOutputValve

ventilatorOutputValve

ProtectedCRCClass
- CRFC:long

GasMixer
- O2Concentration:int
- N2Concentration:int
- AirConcentation:int

O2Sensor

RS232Driver

Alarm
- timeOfOccurrence:dateTime
- severity:int
- msg:OMString
- createView():void

ProtectedCRCClass
- CRFC:long

SpO2Sensor
- measuredO2:int
- lowO2Limit:int
- enable()
- evTurnOn()
- disable()

NetworkDriver
- send(m:Message):void
- dispatch(m:Message):void
- checkMsg(m:Message):void

PumpController
- I2ERato:double
- respirationRate:int
- tidalVolume:int
- inspiratoryTime:int
- averageFlow:int

Pump
- maxPressure:int
- enable()
- disable()
- evStopInspiration()
- pumpOn(pressureRamp:int):...
- pumpOff():void

«electronics» PumpElectronics

ProtectedCRCClass
- CRFC:long

CO2Sensor
- measuredCO2:int
- lowAlarmLimit:int
- highAlarmLimit:int

inspiratoryLimbGFS

GasFlowSensor

expiratoryLimbGFS

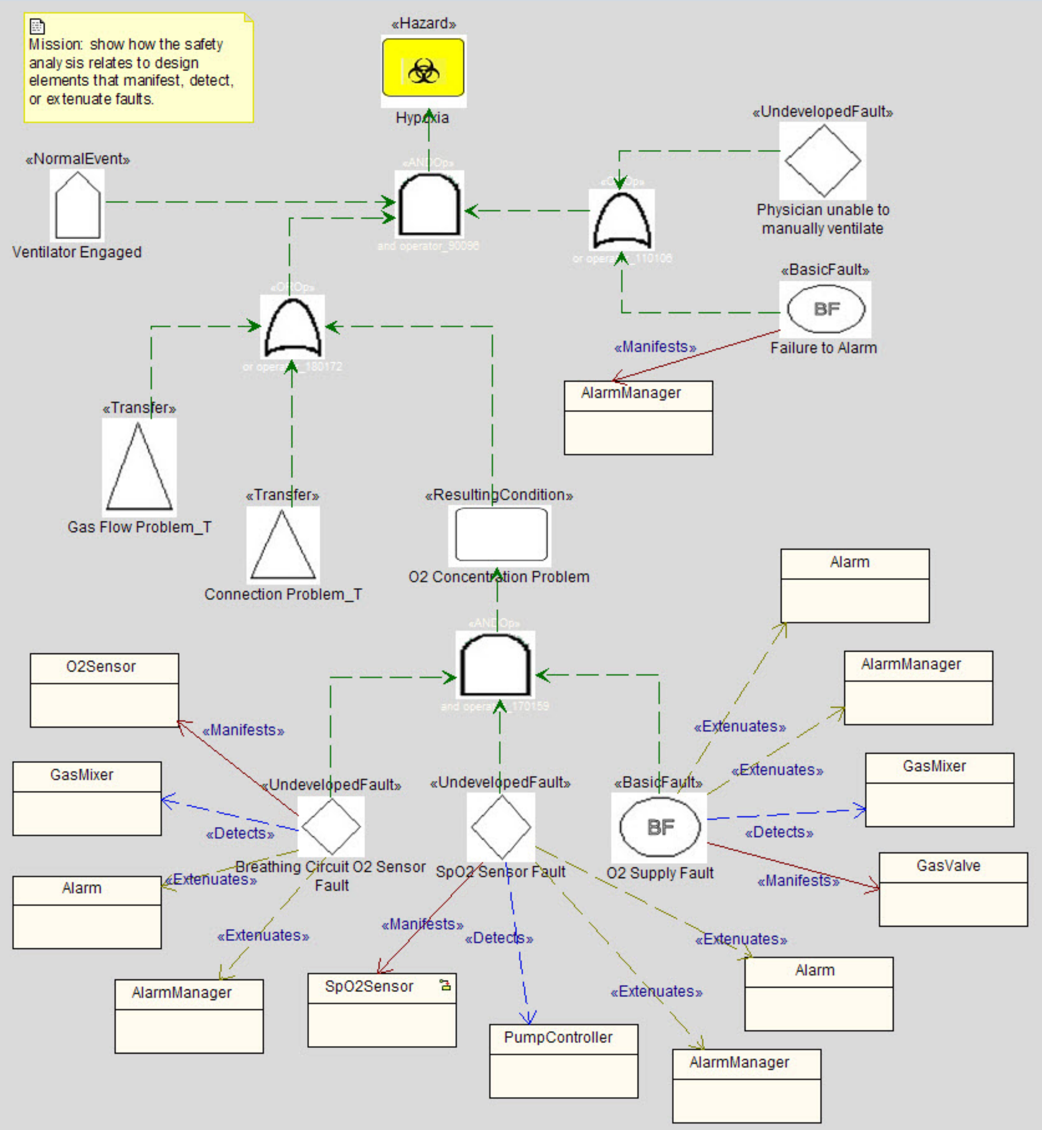
inspiratoryLimbPS

expiratoryLimbPS

PressureSensor

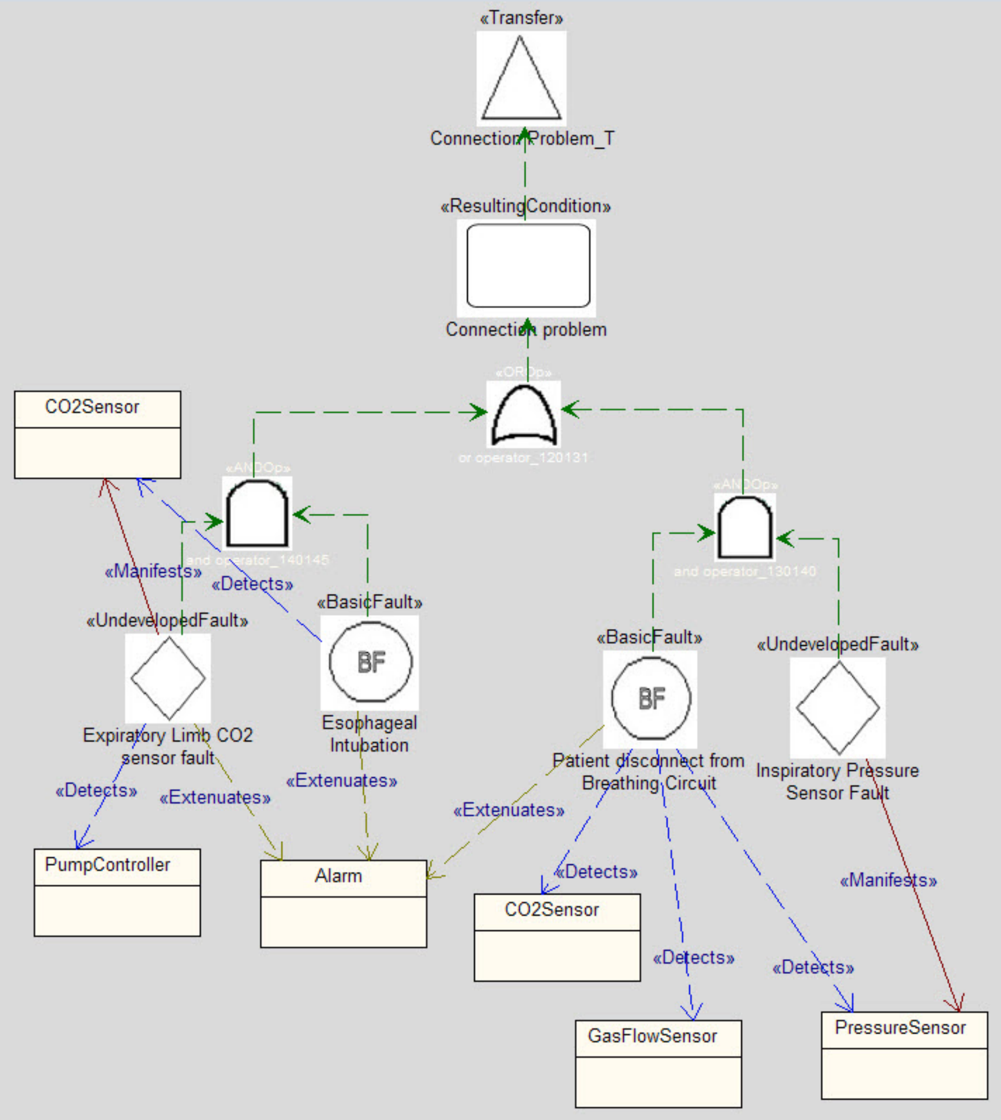
RS232Driver

FTA Hypoxia Hazard with Design Elements

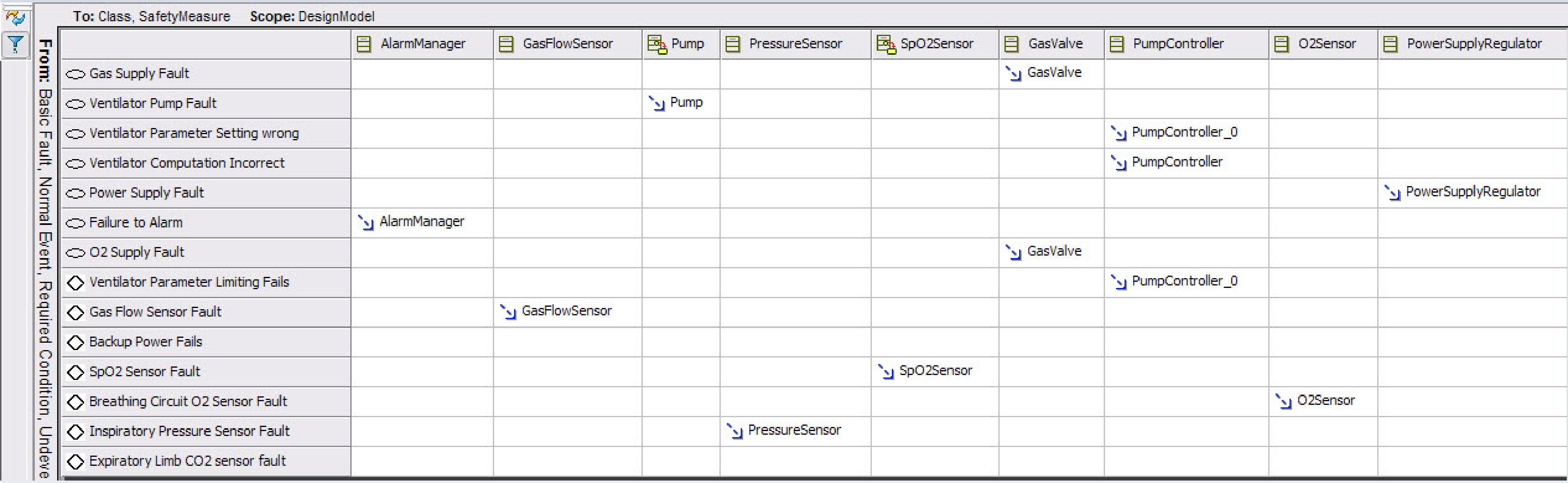


Mission: show how the safety analysis relates to design elements that manifest, detect, or extenuate faults.

FTA Connection Problem with Design Elements



Fault-Source Matrix (generated)

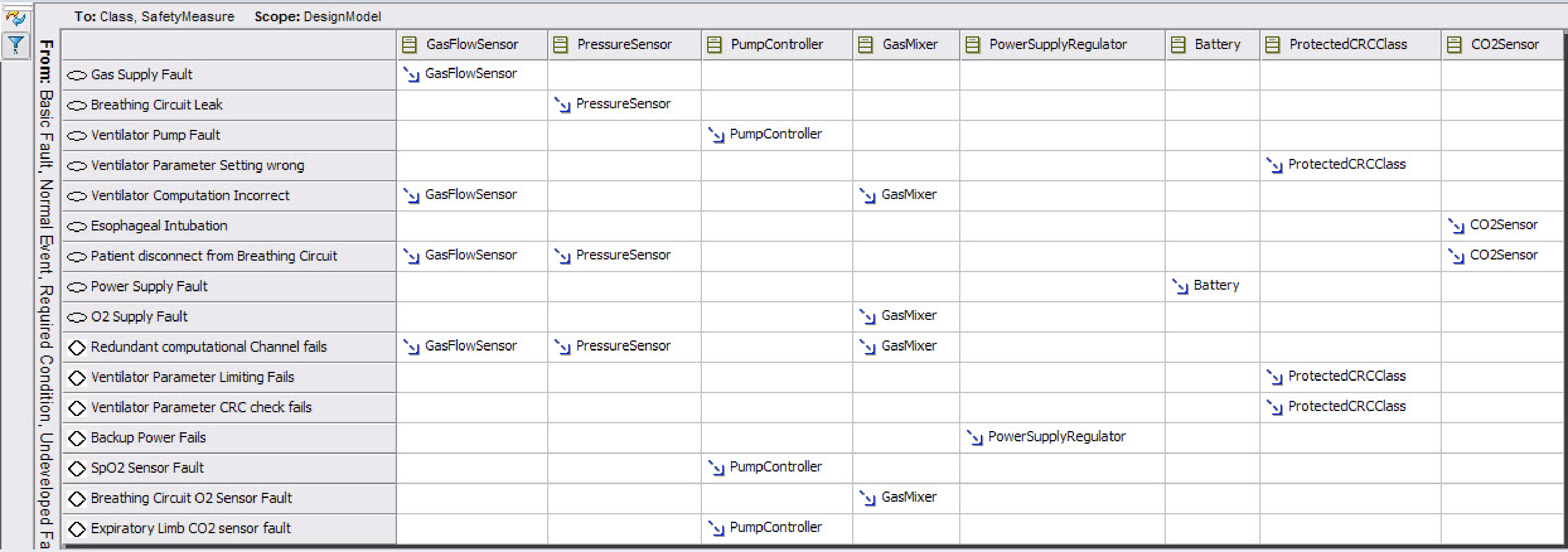


To: Class, SafetyMeasure **Scope:** DesignModel

From: Basic Fault, Normal Event, Required Condition, Undeve

	AlarmManager	GasFlowSensor	Pump	PressureSensor	SpO2Sensor	GasValve	PumpController	O2Sensor	PowerSupplyRegulator
Gas Supply Fault						GasValve			
Ventilator Pump Fault			Pump						
Ventilator Parameter Setting wrong							PumpController_0		
Ventilator Computation Incorrect							PumpController		
Power Supply Fault									PowerSupplyRegulator
Failure to Alarm	AlarmManager								
O2 Supply Fault						GasValve			
Ventilator Parameter Limiting Fails							PumpController_0		
Gas Flow Sensor Fault		GasFlowSensor							
Backup Power Fails									
SpO2 Sensor Fault					SpO2Sensor				
Breathing Circuit O2 Sensor Fault								O2Sensor	
Inspiratory Pressure Sensor Fault				PressureSensor					
Expiratory Limb CO2 sensor fault									

Fault Detection Matrix (generated)



To: Class, SafetyMeasure **Scope:** DesignModel

From: Basic Fault, Normal Event, Required Condition, Undeveloped Fa

	GasFlowSensor	PressureSensor	PumpController	GasMixer	PowerSupplyRegulator	Battery	ProtectedCRCClass	CO2Sensor
Gas Supply Fault	GasFlowSensor							
Breathing Circuit Leak		PressureSensor						
Ventilator Pump Fault			PumpController					
Ventilator Parameter Setting wrong							ProtectedCRCClass	
Ventilator Computation Incorrect	GasFlowSensor			GasMixer				
Esophageal Intubation								CO2Sensor
Patient disconnect from Breathing Circuit	GasFlowSensor	PressureSensor						CO2Sensor
Power Supply Fault						Battery		
O2 Supply Fault				GasMixer				
Redundant computational Channel fails	GasFlowSensor	PressureSensor		GasMixer				
Ventilator Parameter Limiting Fails							ProtectedCRCClass	
Ventilator Parameter CRC check fails							ProtectedCRCClass	
Backup Power Fails					PowerSupplyRegulator			
SpO2 Sensor Fault			PumpController					
Breathing Circuit O2 Sensor Fault				GasMixer				
Expiratory Limb CO2 sensor fault			PumpController					

Hazard Analysis (generated external file) Pg 1

Hazard	Description	Fault tolerance time	Fault tolerance time units	Probability	Severity	Risk	Safety integrity level
Hypoxia	The hypoxia hazard occurs when the brain and other organs receive insufficient oxygen. In a normal 21% O$_2$ environment, death or irreversible injury occurs after five minutes of no oxygen. If the patient is breathing 100% for a significant period of time, this time is about 10 minutes.	5	minutes	1.00E-02	8	8.00E-02	3
Overpressure	Overpressure can damage the lungs. This is an especially severe trauma, possibly fatal, to neonates.	200	milliseconds	1.00E+04	4	3.00E+04	3
Hyperoxia	Hyperoxia problems are usually limited to neonates, where it can cause blindness.	10	minutes	1.00E+05	4	4.00E+05	4
Inadequate anesthesia	Inadequate anesthesia leads to patient discomfort and memory retention of the surgical procedures. This is normally not life threatening but can be severely discomforting.	5	minutes	1.00E+04	2	2.00E+04	2
Over anesthesia	Over anesthesia can lead to death.	3	minutes	1.00E+03	4	4.00E+03	4
Anesthesia leak into ER	Anesthesia leak can lead to short or, in smaller doses, to long-term poisoning of medical staff.	10	minutes	1.00E+05	5	4.00E+05	5

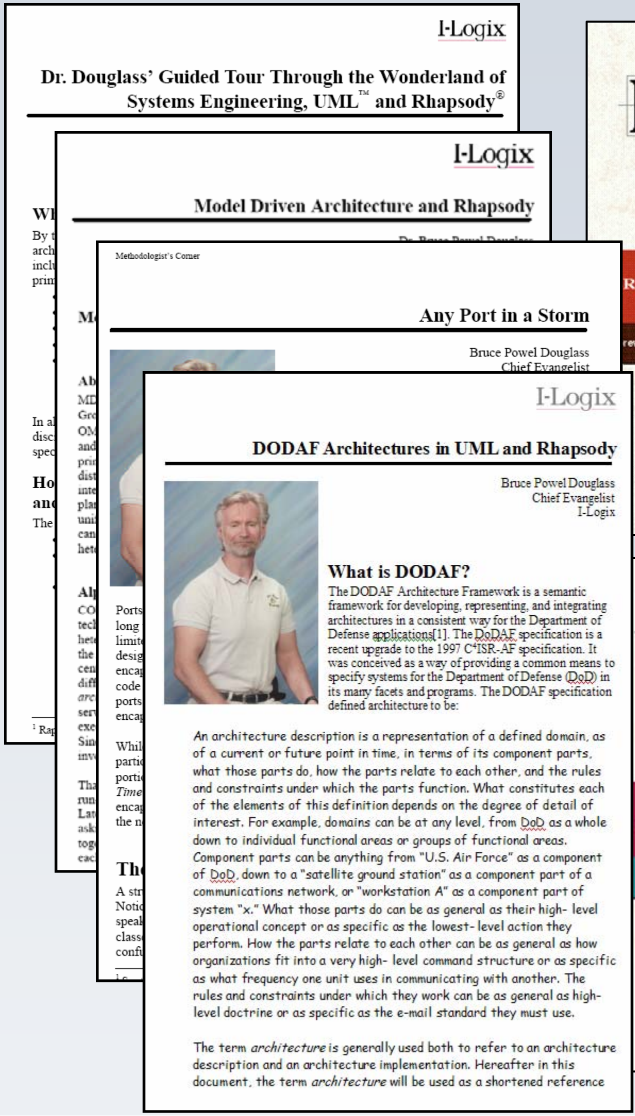
Hazard Analysis (generated external file) Pg 2

Hazard	Fault or event	Fault type	Fault description	MTBF	MTBF time units	Probability
Hypoxia	Ventilator engaged	NormalEvent				1
Hypoxia	Gas supply fault	BasicFault	This fault occurs when gas from a required source is unavailable. This may be due to any number of root causes, such as a stuck or closed valve, running out of gas or a leak.	1.00E+06		1.00E-06
Hypoxia	Breathing circuit leak	BasicFault	This fault occurs when a significant amount of gas leaks from the breathing circuit into the surrounding environment. This can lead to a poisoning hazard when the gas contains anesthetic drugs.	1.00E+03		1.00E-03
Hypoxia	Ventilator pump fault	BasicFault	This fault occurs when the pump internal to the ventilator no longer functions to shape the breath and push gas into the breathing circuit.	1.00E+06		1.00E-06
Hypoxia	Ventilator parameter setting wrong	BasicFault	This fault occurs when a ventilator parameter is out of range. This includes: -I:E ratio -Tidal Volume -Respiration Rate -Inspiratory Pause -Maximum inspiratory pressure -Inspiration time	1.00E+04		1.00E-04
Hypoxia	Ventilator computation incorrect	BasicFault	This fault occurs when an error in the software or a fault in a necessary resource (such as memory) results in an incorrect computation that in turn results in incorrect delivery of ventilation.	1.00E+05		1.00E-05

Hazard Analysis (generated external file) Pg 3

Fault or event	Requirements	Manifestors	Detectors	Extenuators
Gas supply fault	REQ_BCM_01	GasValve	GasFlowSensor	Alarm
Gas supply fault	REQ_VD_06			
Gas supply fault	REQ_VD_03			
Gas supply fault	REQ_VD_04			
Gas supply fault	REQ_VD_08			
Breathing circuit leak	REQ_VD_03		PressureSensor	Alarm
Breathing circuit leak	REQ_VD_04			
Breathing circuit leak	REQ_VD_06			
Ventilator pump fault	REQ_VD_06	Pump	PumpController	PumpController
Ventilator parameter setting wrong	REQ_vent_limit_range_on_patient_mode	PumpController	ProtectedCRCClass	Alarm
Ventilator parameter setting wrong	REQ_vent_parameter_out_of_range_setting			
Ventilator parameter setting wrong	REQ_Vent_confirmation			

References to enhance your Harmony



I-Logix

Dr. Douglass' Guided Tour Through the Wonderland of Systems Engineering, UML™ and Rhapsody®

I-Logix

Model Driven Architecture and Rhapsody

Methodologist's Corner

Any Port in a Storm

Bruce Powel Douglass
Chief Evangelist

I-Logix

DODAF Architectures in UML and Rhapsody

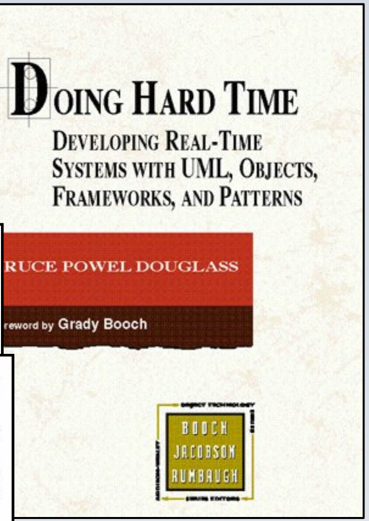
Bruce Powel Douglass
Chief Evangelist
I-Logix

What is DODAF?

The DODAF Architecture Framework is a semantic framework for developing, representing, and integrating architectures in a consistent way for the Department of Defense applications[1]. The DoDAF specification is a recent upgrade to the 1997 C^4ISR-AF specification. It was conceived as a way of providing a common means to specify systems for the Department of Defense (DoD) in its many facets and programs. The DODAF specification defined architecture to be:

An architecture description is a representation of a defined domain, as of a current or future point in time, in terms of its component parts, what those parts do, how the parts relate to each other, and the rules and constraints under which the parts function. What constitutes each of the elements of this definition depends on the degree of detail of interest. For example, domains can be at any level, from DoD as a whole down to individual functional areas or groups of functional areas. Component parts can be anything from "U.S. Air Force" as a component of DoD, down to a "satellite ground station" as a component part of a communications network, or "workstation A" as a component part of system "x." What those parts do can be as general as their high- level operational concept or as specific as the lowest- level action they perform. How the parts relate to each other can be as general as how organizations fit into a very high- level command structure or as specific as what frequency one unit uses in communicating with another. The rules and constraints under which they work can be as general as high-level doctrine or as specific as the e-mail standard they must use.

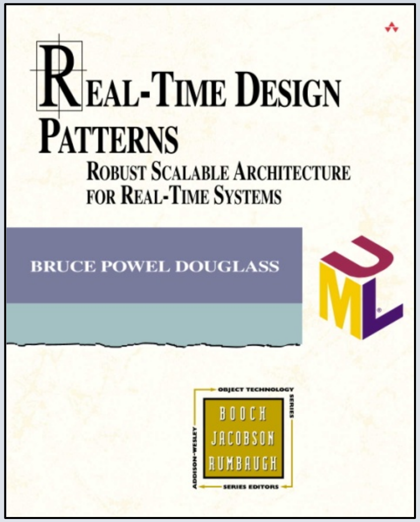
The term *architecture* is generally used both to refer to an architecture description and an architecture implementation. Hereafter in this document, the term *architecture* will be used as a shortened reference



Doing Hard Time
Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns

Bruce Powel Douglass
Foreword by Grady Booch

Object Technology Series
Booch Jacobson Rumbaugh
Addison-Wesley
Series Editors

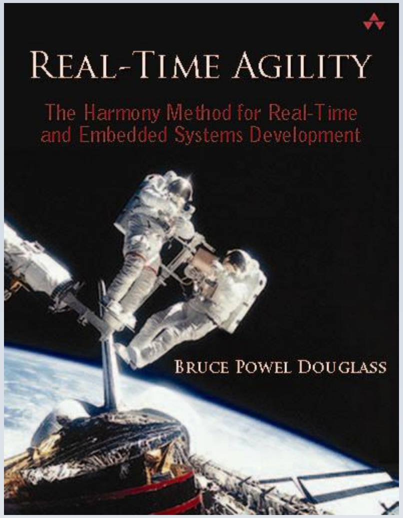


Real-Time Design Patterns
Robust Scalable Architecture for Real-Time Systems

Bruce Powel Douglass

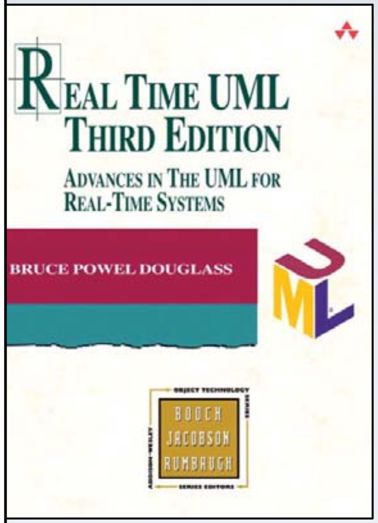
UML

Object Technology Series
Booch Jacobson Rumbaugh
Addison-Wesley
Series Editors



Real-Time Agility
The Harmony Method for Real-Time and Embedded Systems Development

Bruce Powel Douglass

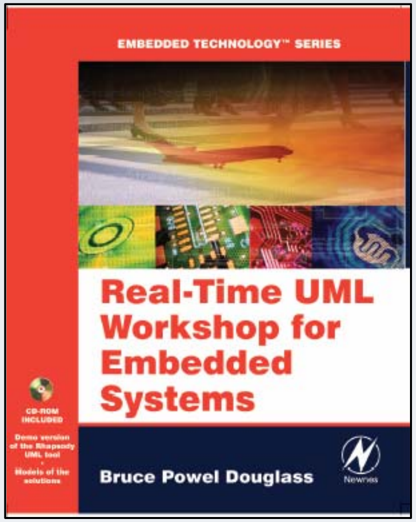


Real Time UML
Third Edition
Advances in the UML for Real-Time Systems

Bruce Powel Douglass

UML

Object Technology Series
Booch Jacobson Rumbaugh
Addison-Wesley
Series Editors



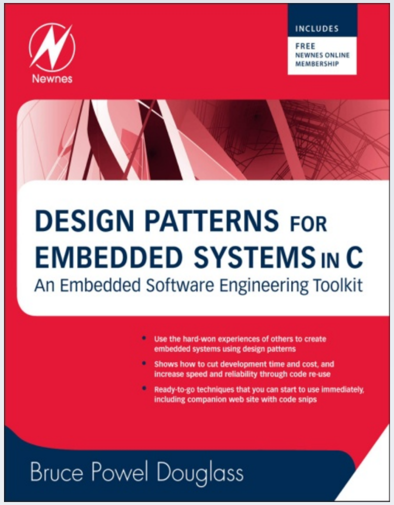
Embedded Technology™ Series

Real-Time UML Workshop for Embedded Systems

CD-ROM Included
Demo version of the Rhapsody UML tool
•
Models of the solutions

Bruce Powel Douglass

Newnes



Newnes

Includes
Free Newnes Online Membership

Design Patterns for Embedded Systems in C
An Embedded Software Engineering Toolkit

- Use the hard-won experiences of others to create embedded systems using design patterns
- Shows how to cut development time and cost, and increase speed and reliability through code re-use
- Ready-to-go techniques that you can start to use immediately, including companion web site with code snips

Bruce Powel Douglass