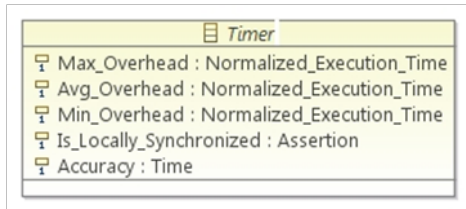


Laxitudes del meta-modelo MAST-2

1 Laxitudes relacionadas con los atributos de una clase

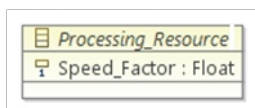
1.1 Timer



a) Los overheads pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	Max_Overhead ≥ Avg_Overhead ≥ Min_Overhead
Formulación OCL	<pre> context Timer inv i_1_1_a: Max_Overhead >= Avg_Overhead and Max_Overhead >= Min_Overhead and Avg_Overhead >= Min_Overhead </pre>
	<pre> context Mast_Model inv i_1_1_a: Timer.allInstances() -> forAll(t t.Max_Overhead >= t.Avg_Overhead and t.Max_Overhead >= t.Min_Overhead and t.Avg_Overhead >= t.Min_Overhead) </pre>

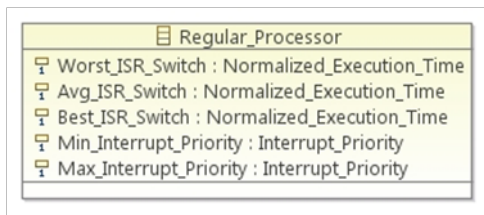
1.2 Processing_Resource



a) El speed factor puede ser no-positivo

Severidad de problema	Error
Restricción de integridad	Speed_Factor > 0.0
Formulación OCL	<pre> context Processing_Resource inv i_1_2_a: Speed_Factor > 0.0 </pre>
	<pre> context Mast_Model inv i_1_2_a: Processing_Resource.allInstances() -> forAll(pr pr.Speed_Factor > 0.0) </pre>

1.3 Regular_Processor



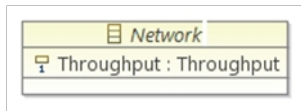
a) Las prioridades de interrupción pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	$\text{Max_Interrupt_Priority} \geq \text{Min_Interrupt_Priority}$
Formulación OCL	<pre> context Regular_Processor inv i_1_3_a: Max_Interrupt_Priority >= Min_Interrupt_Priority </pre>
	<pre> context Mast_Model inv i_1_3_a: Regular_Processor.allInstances() -> forAll(rp rp.Max_Interrupt_Priority >= rp.Min_Interrupt_Priority) </pre>

b) Los cambios de contexto por ISR pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	$\text{Worst_ISR_Switch} \geq \text{Avg_ISR_Switch} \geq \text{Best_ISR_Switch}$
Formulación OCL	<pre> context Regular_Processor inv i_1_3_b: Worst_ISR_Switch >= Avg_ISR_Switch and Worst_ISR_Switch >= Best_ISR_Switch and Avg_ISR_Switch >= Best_ISR_Switch </pre>
	<pre> context Mast_Model inv i_1_3_b: Regular_Processor.allInstances() -> forAll(rp rp.Worst_ISR_Switch >= rp.Avg_ISR_Switch and rp.Worst_ISR_Switch >= rp.Best_ISR_Switch and rp.Avg_ISR_Switch >= rp.Best_ISR_Switch) </pre>

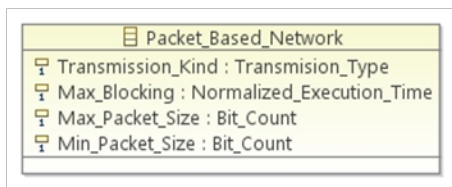
1.4 Network



a) El throughput puede ser negativo

Severidad de problema	Error
Restricción de integridad	$\text{Throughput} \geq 0.0$
Formulación OCL	<pre> context Network inv i_1_4_a: Throughput > 0.0 </pre>
	<pre> context Mast_Model inv i_1_4_a: Network.allInstances() -> forALL(n n.Throughput > 0.0) </pre>

1.5 Packet_Based_Network



a) Los tamaños de paquete pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	$\text{Max_Packet_Size} \geq \text{Min_Packet_Size}$
Formulación OCL	<pre> context Packet_Based_Network inv i_1_5_a: Max_Packet_Size >= Min_Packet_Size </pre>
	<pre> context Mast_Model inv i_1_5_a: Packet_Based_Network.allInstances() -> forALL(pbn pbn.Max_Packet_Size >= pbn.Min_Packet_Size) </pre>

1.6

1.7 AFDX_Link

AFDX_Link	
Max_Packet_Size	: Bit_Count
Min_Packet_Size	: Bit_Count
Max_HW_Tx_Latency	: Time
Min_HW_Tx_Latency	: Time
Avg_HW_Tx_Latency	: Time
Max_HW_Rx_Latency	: Time
Min_HW_Rx_Latency	: Time
Avg_HW_Rx_Latency	: Time
Ethernet_Overhead	: Bit_Count
Protocol_Overhead	: Bit_Count

a) Los tamaños de paquete pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	$\text{Max_Packet_Size} \geq \text{Min_Packet_Size}$
Formulación OCL	<pre>context AFDX_Link inv i_1_6_a: Max_Packet_Size >= Min_Packet_Size</pre>
	<pre>context Mast_Model inv i_1_6_a: AFDX_Link.allInstances() -> forALL(a a.Max_Packet_Size >= a.Min_Packet_Size)</pre>

b) Las latencias HW_Tx pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	$\text{Max_HW_Tx_Latency} \geq \text{Avg_HW_Tx_Latency} \geq \text{Min_HW_Tx_Latency}$
Formulación OCL	<pre>context AFDX_Link inv i_1_6_b: Max_HW_Tx_Latency >= Avg_HW_Tx_Latency and Max_HW_Tx_Latency >= Min_HW_Tx_Latency and Avg_HW_Tx_Latency >= Min_HW_Tx_Latency</pre>
	<pre>context Mast_Model inv i_1_6_b: AFDX_Link.allInstances() -> forALL(a a.Max_HW_Tx_Latency >= a.Avg_HW_Tx_Latency and a.Max_HW_Tx_Latency >= a.Min_HW_Tx_Latency and a.Avg_HW_Tx_Latency >= a.Min_HW_Tx_Latency)</pre>

c) Las latencias HW_Rx pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	$\text{Max_HW_Rx_Latency} \geq \text{Avg_HW_Rx_Latency} \geq \text{Min_HW_Rx_Latency}$
Formulación OCL	<pre>context AFDX_Link inv i_1_6_c: Max_HW_Rx_Latency >= Avg_HW_Rx_Latency and Max_HW_Rx_Latency >= Min_HW_Rx_Latency and Avg_HW_Rx_Latency >= Min_HW_Rx_Latency</pre>
	<pre>context Mast_Model inv i_1_6_c: AFDX_Link.allInstances() -> forALL(a a.Max_HW_Rx_Latency >= a.Avg_HW_Rx_Latency and a.Max_HW_Rx_Latency >= a.Min_HW_Rx_Latency and a.Avg_HW_Rx_Latency >= a.Min_HW_Rx_Latency)</pre>

1.8 Network_Switch

Network_Switch	
Max_Fixed_Fork_Latency	: Normalized_Execution_Time
Avg_Fixed_Fork_Latency	: Normalized_Execution_Time
Min_Fixed_Fork_Latency	: Normalized_Execution_Time
Max_Variable_Fork_Latency	: Normalized_Execution_Time
Avg_Variable_Fork_Latency	: Normalized_Execution_Time
Min_Variable_Fork_Latency	: Normalized_Execution_Time
Max_Delivery_Latency	: Normalized_Execution_Time
Avg_Delivery_Latency	: Normalized_Execution_Time
Min_Delivery_Latency	: Normalized_Execution_Time

a) Las latencias de bifurcación fija pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	Max_Fixed_Fork_Latency ≥ Avg_Fixed_Fork_Latency ≥ Min_Fixed_Fork_Latency
Formulación OCL	<pre>context Network_Switch inv i_1_7_a: Max_Fixed_Fork_Latency >= Avg_Fixed_Fork_Latency and Max_Fixed_Fork_Latency >= Min_Fixed_Fork_Latency and Avg_Fixed_Fork_Latency >= Min_Fixed_Fork_Latency</pre>
	<pre>context Mast_Model inv i_1_7_a: Network_Switch.allInstances() -> forAll(ns ns.Max_Fixed_Fork_Latency >= ns.Avg_Fixed_Fork_Latency and ns.Max_Fixed_Fork_Latency >= ns.Min_Fixed_Fork_Latency and ns.Avg_Fixed_Fork_Latency >= ns.Min_Fixed_Fork_Latency)</pre>

b) Las latencias de bifurcación variable pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	Max_Variable_Fork_Latency ≥ Avg_Variable_Fork_Latency ≥ Min_Variable_Fork_Latency
Formulación OCL	<pre>context Network_Switch inv i_1_7_b: Max_Variable_Fork_Latency >= Avg_Variable_Fork_Latency and Max_Variable_Fork_Latency >= Min_Variable_Fork_Latency and Avg_Variable_Fork_Latency >= Min_Variable_Fork_Latency</pre>
	<pre>context Mast_Model inv i_1_7_b: Network_Switch.allInstances() -> forAll(ns ns.Max_Variable_Fork_Latency >= ns.Avg_Variable_Fork_Latency and ns.Max_Variable_Fork_Latency >= ns.Min_Variable_Fork_Latency and ns.Avg_Variable_Fork_Latency >= ns.Min_Variable_Fork_Latency)</pre>

c) Las latencias de reparto pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	Max_Delivery_Latency ≥ Avg_Delivery_Latency ≥ Min_Delivery_Latency
Formulación OCL	<pre>context Network_Switch inv i_1_7_c: Max_Delivery_Latency >= Avg_Delivery_Latency and Max_Delivery_Latency >= Min_Delivery_Latency and Avg_Delivery_Latency >= Min_Delivery_Latency</pre>
	<pre>context Mast_Model inv i_1_7_c: Network_Switch.allInstances() -></pre>

	<pre>forAll(ns ns.Max_Delivery_Latency >= ns.Avg_Delivery_Latency and ns.Max_Delivery_Latency >= ns.Min_Delivery_Latency and ns.Avg_Delivery_Latency >= ns.Min_Delivery_Latency)</pre>
--	---

1.9 Network_Router

<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #ffffcc; border: 1px solid black; padding: 2px; margin-bottom: 5px;"> Network_Router </div> <ul style="list-style-type: none"> [-] Max_Fixed_Branch_Latency : Normalized_Execution_Time [-] Avg_Fixed_Branch_Latency : Normalized_Execution_Time [-] Min_Fixed_Branch_Latency : Normalized_Execution_Time [-] Max_Variable_Branch_Latency : Normalized_Execution_Time [-] Avg_Variable_Branch_Latency : Normalized_Execution_Time [-] Min_Variable_Branch_Latency : Normalized_Execution_Time </div>	
--	--

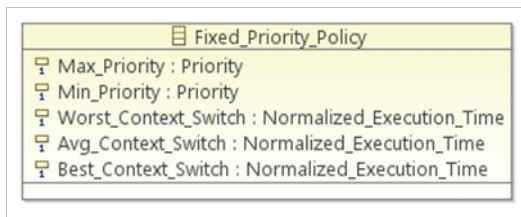
a) Las latencias de ramificación fija pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	Max_Fixed_Branch_Latency ≥ Avg_Fixed_Branch_Latency ≥ Min_Fixed_Branch_Latency
Formulación OCL	<pre>context Network_Router inv i_1_8_a: Max_Fixed_Branch_Latency >= Avg_Fixed_Branch_Latency and Max_Fixed_Branch_Latency >= Min_Fixed_Branch_Latency and Avg_Fixed_Branch_Latency >= Min_Fixed_Branch_Latency</pre>
	<pre>context Mast_Model inv i_1_8_a: Network_Router.allInstances() -> forAll(nr nr.Max_Fixed_Branch_Latency >= nr.Avg_Fixed_Branch_Latency and nr.Max_Fixed_Branch_Latency >= nr.Min_Fixed_Branch_Latency and nr.Avg_Fixed_Branch_Latency >= nr.Min_Fixed_Branch_Latency)</pre>

b) Las latencias de ramificación variable pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	Max_Variable_Branch_Latency ≥ Avg_Variable_Branch_Latency ≥ Min_Variable_Branch_Latency
Formulación OCL	<pre>context Network_Router inv i_1_8_b: Max_Variable_Branch_Latency >= Avg_Variable_Branch_Latency and Max_Variable_Branch_Latency >= Min_Variable_Branch_Latency and Avg_Variable_Branch_Latency >= Min_Variable_Branch_Latency</pre>
	<pre>context Mast_Model inv i_1_8_b: Network_Router.allInstances() -> forAll(nr nr.Max_Variable_Branch_Latency >= nr.Avg_Variable_Branch_Latency and nr.Max_Variable_Branch_Latency >= nr.Min_Variable_Branch_Latency and nr.Avg_Variable_Branch_Latency >= nr.Min_Variable_Branch_Latency)</pre>

1.10 Fixed_Priority_Policy



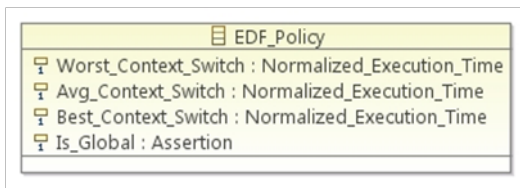
a) Las prioridades pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	$\text{Max_Priority} \geq \text{Min_Priority}$
Formulación OCL	<pre> context Fixed_Priority_Policy inv i_1_9_a: Max_Priority >= Min_Priority </pre>
	<pre> context Mast_Model inv i_1_9_a: Fixed_Priority_Policy.allInstances() -> forAll(p p.Max_Priority >= p.Min_Priority) </pre>

b) Los cambios de contexto pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	$\text{Worst_Context_Switch} \geq \text{Avg_Context_Switch} \geq \text{Best_Context_Switch}$
Formulación OCL	<pre> context Fixed_Priority_Policy inv i_1_9_b: Worst_Context_Switch >= Avg_Context_Switch and Worst_Context_Switch >= Best_Context_Switch and Avg_Context_Switch >= Best_Context_Switch </pre>
	<pre> context Mast_Model inv i_1_9_b: Fixed_Priority_Policy.allInstances() -> forAll(p p.Worst_Context_Switch >= p.Avg_Context_Switch and p.Worst_Context_Switch >= p.Best_Context_Switch and p.Avg_Context_Switch >= p.Best_Context_Switch) </pre>

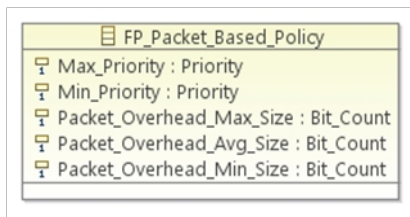
1.11 EDF_Policy



a) *Los cambios de contexto pueden estar desordenados*

Severidad de problema	Error
Restricción de integridad	Worst_Context_Switch ≥ Avg_Context_Switch ≥ Best_Context_Switch
Formulación OCL	<pre> context EDF_Policy inv i_1_10_a: Worst_Context_Switch >= Avg_Context_Switch and Worst_Context_Switch >= Best_Context_Switch and Avg_Context_Switch >= Best_Context_Switch </pre>
	<pre> context Mast_Model inv i_1_10_a: EDF_Policy.allInstances() -> forAll(p p.Worst_Context_Switch >= p.Avg_Context_Switch and p.Worst_Context_Switch >= p.Best_Context_Switch and p.Avg_Context_Switch >= p.Best_Context_Switch) </pre>

1.12 FP_Packet_Based_Policy



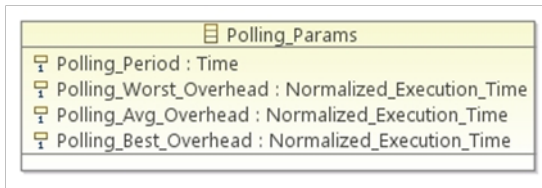
a) Las prioridades pueden estar desordenadas

Severidad de problema	Error
Restricción de integridad	$\text{Max_Priority} \geq \text{Min_Priority}$
Formulación OCL	<pre> context FP_Packet_Based_Policy inv i_1_11_a: Max_Priority >= Min_Priority context Mast_Model inv i_1_11_a: FP_Packet_Based_Policy.allInstances() -> forALL(p p.Max_Priority >= p.Min_Priority) </pre>

b) Los tamaños de overhead de paquete pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	$\text{Packet_Overhead_Max_Size} \geq \text{Packet_Overhead_Avg_Size} \geq \text{Packet_Overhead_Min_Size}$
Formulación OCL	<pre> context FP_Packet_Based_Policy inv i_1_11_b: Packet_Overhead_Max_Size >= Packet_Overhead_Avg_Size and Packet_Overhead_Max_Size >= Packet_Overhead_Min_Size and Packet_Overhead_Avg_Size >= Packet_Overhead_Min_Size context Mast_Model inv i_1_11_b: FP_Packet_Based_Policy.allInstances() -> forALL(p p.Packet_Overhead_Max_Size >= p.Packet_Overhead_Avg_Size and p.Packet_Overhead_Max_Size >= p.Packet_Overhead_Min_Size and p.Packet_Overhead_Avg_Size >= p.Packet_Overhead_Min_Size) </pre>

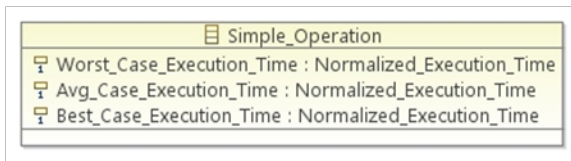
1.13 Polling_Params



a) Los overhead de sondeo pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	$\text{Polling_Worst_Overhead} \geq \text{Polling_Avg_Overhead} \geq \text{Polling_Best_Overhead}$
Formulación OCL	<pre> context Polling_Params inv i_1_12_a: Polling_Worst_Overhead >= Polling_Avg_Overhead and Polling_Worst_Overhead >= Polling_Best_Overhead and Polling_Avg_Overhead >= Polling_Best_Overhead </pre>
	<pre> context Mast_Model inv i_1_12_a: Polling_Params.allInstances() -> forAll(pp pp.Polling_Worst_Overhead >= pp.Polling_Avg_Overhead and pp.Polling_Worst_Overhead >= pp.Polling_Best_Overhead and pp.Polling_Avg_Overhead >= pp.Polling_Best_Overhead) </pre>

1.14 Simple_Operation



a) Los tiempos de ejecución pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	$\text{Worst_Case_Execution_Time} \geq \text{Avg_Case_Execution_Time} \geq \text{Best_Case_Execution_Time}$
Formulación OCL	<pre> context Simple_Operation inv i_1_13_a: Worst_Case_Execution_Time >= Avg_Case_Execution_Time and Worst_Case_Execution_Time >= Best_Case_Execution_Time and Avg_Case_Execution_Time >= Best_Case_Execution_Time </pre>
	<pre> context Mast_Model inv i_1_13_a: Simple_Operation.allInstances() -> forAll(so so.Worst_Case_Execution_Time >= so.Avg_Case_Execution_Time and so.Worst_Case_Execution_Time >= so.Best_Case_Execution_Time and so.Avg_Case_Execution_Time >= so.Best_Case_Execution_Time) </pre>

1.15 Enclosing_Operation

Enclosing_Operation
Worst_Case_Execution_Time : Normalized_Execution_Time
Avg_Case_Execution_Time : Normalized_Execution_Time
Best_Case_Execution_Time : Normalized_Execution_Time

a) Los tiempos de ejecución pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	Worst_Case_Execution_Time \geq Avg_Case_Execution_Time \geq Best_Case_Execution_Time
Formulación OCL	<pre>context Enclosing_Operation inv i_1_14_a: Worst_Case_Execution_Time >= Avg_Case_Execution_Time and Worst_Case_Execution_Time >= Best_Case_Execution_Time and Avg_Case_Execution_Time >= Best_Case_Execution_Time</pre>
	<pre>context Mast_Model inv i_1_14_a: Enclosing_Operation.allInstances() -> forAll(eo eo.Worst_Case_Execution_Time >= eo.Avg_Case_Execution_Time and eo.Worst_Case_Execution_Time >= eo.Best_Case_Execution_Time and eo.Avg_Case_Execution_Time >= eo.Best_Case_Execution_Time)</pre>

1.16 Message

Message
Max_Message_Size : Bit_Count
Avg_Message_Size : Bit_Count
Min_Message_Size : Bit_Count

a) Los tamaños de mensaje pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	Max_Message_Size \geq Avg_Message_Size \geq Min_Message_Size
Formulación OCL	<pre>context Message inv i_1_15_a: Max_Message_Size >= Avg_Message_Size and Max_Message_Size >= Min_Message_Size and Avg_Message_Size >= Min_Message_Size</pre>
	<pre>context Mast_Model inv i_1_15_a: Message.allInstances() -> forAll(m m.Max_Message_Size >= m.Avg_Message_Size and m.Max_Message_Size >= m.Min_Message_Size and m.Avg_Message_Size >= m.Min_Message_Size)</pre>

1.17 Sporadic_Event

<ul style="list-style-type: none"> ☰ Sporadic_Event ☰ Avg_Interarrival : Time ☰ Min_Interarrival : Time ☰ Distribution : Distribution_Type
--

a) Los tiempos entre llegadas pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	$Avg_Interarrival \geq Min_Interarrival$
Formulación OCL	<pre>context Sporadic_Event inv i_1_16_a: Avg_Interarrival >= Min_Interarrival</pre>
	<pre>context Mast_Model inv i_1_16_a: Sporadic_Event.allInstances() -> forAll(se se.Avg_Interarrival >= se.Min_Interarrival)</pre>

1.18 Delay

<ul style="list-style-type: none"> ☰ Delay ☰ Delay_Max_Interval : Time ☰ Delay_Avg_Interval : Time ☰ Delay_Best_Interval : Time

a) Los intervalos de retraso pueden estar desordenados

Severidad de problema	Error
Restricción de integridad	$Delay_Max_Interval \geq Delay_Avg_Interval \geq Delay_Min_Interval$
Formulación OCL	<pre>context Delay inv i_1_17_a: Delay_Max_Interval >= Delay_Avg_Interval and Delay_Max_Interval >= Delay_Best_Interval and Delay_Avg_Interval >= Delay_Best_Interval</pre>
	<pre>context Mast_Model inv i_1_17_a: Delay.allInstances() -> forAll(d d.Delay_Max_Interval >= d.Delay_Avg_Interval and d.Delay_Max_Interval >= d.Delay_Best_Interval and d.Delay_Avg_Interval >= d.Delay_Best_Interval)</pre>

2 Laxitudes relacionadas con una asociación

2.1 Mast_Model.Element_List



a) Los objetos de sincronización podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En un modelo MAST2, cada objeto de sincronización ha de tener nombre único
Formulación OCL	<pre> context Clock_Synchronization_Object inv i_2_1_a: Model.Element_List -> select(e e.oclIsTypeOf(Clock_Synchronization_Object)) -> forAll(c c <> self implies c.name <> self.name) </pre>
	<pre> context Mast_Model inv i_2_1_a: Element_List -> select(e e.oclIsKindOf(Clock_Synchronization_Object)) -> forAll(c1, c2 c1 <> c2 implies c1.name <> c2.name) </pre>

b) Los temporizadores podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En un modelo MAST2, cada temporizador ha de tener nombre único
Formulación OCL	<pre> context Timer inv i_2_1_b: Model.Element_List -> select(e e.oclIsKindOf(Timer)) -> forAll(t t <> self implies t.name <> self.name) </pre>
	<pre> context Mast_Model inv i_2_1_b: Element_List -> select(e e.oclIsKindOf(Timer)) -> forAll(t1, t2 t1 <> t2 implies t1.name <> t2.name) </pre>

c) Los recursos de procesamiento podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En un modelo MAST2, cada recurso de procesamiento ha de tener nombre único
Formulación OCL	<pre> context Processing_Resource inv i_2_1_c: Model.Element_List -> select(e e.oclIsKindOf(Processing_Resource)) -> forAll(pr pr <> self implies pr.name <> self.name) </pre>
	<pre> context Mast_Model inv i_2_1_c: Element_List -> select(e e.oclIsKindOf(Processing_Resource)) -> forAll(pr1, pr2 pr1 <> pr2 implies pr1.name <> pr2.name) </pre>

d) Los planificadores podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En un modelo MAST2, cada planificador ha de tener nombre único
Formulación OCL	<pre>context Scheduler inv i_2_1_d: Model.Element_List -> select(e e.oclIsKindOf(Scheduler)) -> forAll(s s <> self implies s.name <> self.name)</pre>
	<pre>context Mast_Model inv i_2_1_d: Element_List -> select(e e.oclIsKindOf(Scheduler)) -> forAll(s1, s2 s1 <> s2 implies s1.name <> s2.name)</pre>

e) Los recursos planificables podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En un modelo MAST2, cada recurso planificable ha de tener nombre único
Formulación OCL	<pre>context Schedulable_Resource inv i_2_1_e: Model.Element_List -> select(e e.oclIsKindOf(Schedulable_Resource)) -> forAll(sr sr <> self implies sr.name <> self.name)</pre>
	<pre>context Mast_Model inv i_2_1_e: Element_List -> select(e e.oclIsKindOf(Schedulable_Resource)) -> forAll(sr1, sr2 sr1 <> sr2 implies sr1.name <> sr2.name)</pre>

f) Los recursos de exclusión mútua podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En un modelo MAST2, cada recurso de exclusión mútua ha de tener nombre único
Formulación OCL	<pre>context Mutual_Exclusion_Resource inv i_2_1_f: Model.Element_List -> select(e e.oclIsKindOf(Mutual_Exclusion_Resource)) -> forAll(m m <> self implies m.name <> self.name)</pre>
	<pre>context Mast_Model inv i_2_1_f: Element_List -> select(e e.oclIsKindOf(Mutual_Exclusion_Resource)) -> forAll(m1, m2 m1 <> m2 implies m1.name <> m2.name)</pre>

g) Las operaciones podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En un modelo MAST2, cada operación ha de tener nombre único
Formulación OCL	<pre>context Operation inv i_2_1_g: Model.Element_List -> select(e e.oclIsKindOf(Operation)) -> forAll(o o <> self implies o.name <> self.name)</pre>
	<pre>context Mast_Model inv i_2_1_g: Element_List -> select(e e.oclIsKindOf(Operation)) -> forAll(o1, o2 o1 <> o2 implies o1.name <> o2.name)</pre>

h) Las transacciones podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En un modelo MAST2, cada transacción ha de tener nombre único
Formulación OCL	<pre>context End_To_End_Flow inv i_2_1_h: Model.Element_List -> select(e e.oclIsKindOf(End_To_End_Flow)) -> forAll(e2ef e2ef <> self implies e2ef.name <> self.name)</pre>
	<pre>context Mast_Model inv i_2_1_h: Element_List -> select(e e.oclIsKindOf(End_To_End_Flow)) -> forAll(e1, e2 e1 <> e2 implies e1.Name <> e2.Name)</pre>

i) Un modelo MAST2 podría no contener los elementos mínimos requeridos

Severidad de problema	Error
Restricción de integridad	<p>En un modelo MAST2 ha de haber al menos:</p> <ul style="list-style-type: none"> • 1 Computing_Resource • 1 Primary_Scheduler • 1 Thread • 1 Regular_End_To_End_Flow • 1 Simple_Operation
Formulación OCL	<pre>context End_To_End_Flow inv i_2_1_i: Model.Element_List -></pre>
	<pre>context Mast_Model inv i_2_1_i:</pre>

- j) Un modelo MAST podría contener una red de comunicaciones y podría no contener al menos dos recursos de computación albergando a los planificadores que planifican a los recursos planificables utilizados por los drivers de la red.

Severidad de problema	Error
Restricción de integridad	En un modelo MAST2
Formulación OCL	<pre>context End_To_End_Flow inv i_2_1_j: Model.Element_List -></pre>
	<pre>context Mast_Model inv i_2_1_k:</pre>

2.2 End_To_End_Flow.Workload_Event_List



- a) Eventos activadores de una transacción podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En una transacción, cada evento externo ha de tener nombre único
Formulación OCL	<pre>context WorkLoad_Event inv i_2_2_a: Owner.WorkLoad_Event_List -> forALL(we we <> self implies we.name <> self.name)</pre>
	<pre>context End_To_End_Flow inv i_2_2_a: WorkLoad_Event_List -> forALL(we1, we2 we1 <> we2 implies we1.name <> we2.name)</pre>

2.3 End_To_End_Flow.Internal_Event_List



a) Eventos internos de una transacción podrían compartir nombre

Severidad de problema	Error
Restricción de integridad	En una transacción, cada evento interno ha de tener nombre único
Formulación OCL	<pre> context Internal_Event inv i_2_3_a: Owner.Internal_Event_List -> forAll(ie ie <> self implies ie.name <> self.name) context End_To_End_Flow inv i_2_3_a: Internal_Event_List -> forAll(ie1, ie2 ie1 <> ie2 implies ie1.name <> ie2.name) </pre>

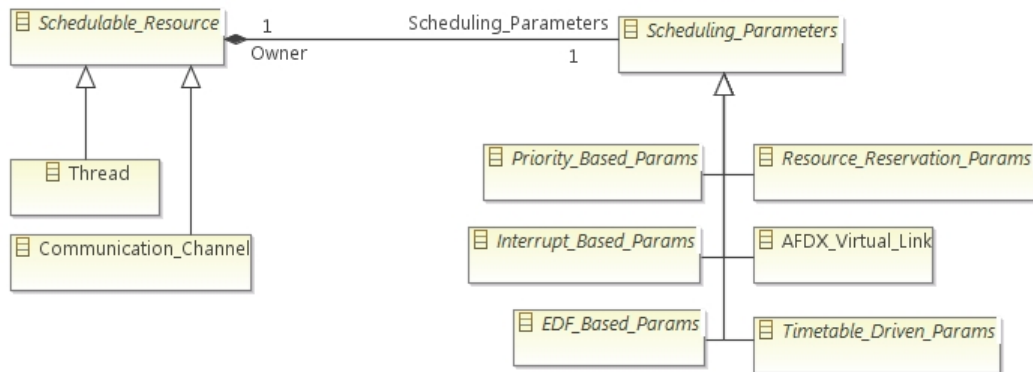
2.4 End_To_End_Flow.Event_Handler_List



a) Una transacción podría no contener los elementos mínimamente requeridos (1 actividad)

Severidad de problema	Error
Restricción de integridad	Toda transacción ha de contener al menos una actividad
Formulación OCL	<pre> context End_To_End_Flow inv i_2_4_a: Event_Handler_List -> exists(eh eh.ocIsTypeOf(Step)) context Mast_Model inv i_2_4_a: Element_List -> select(e e.ocIsKindOf(End_To_End_Flow)) -> forAll(e2ef e2ef.Event_Handler_List -> exists(eh eh.ocIsTypeOf(Step))) </pre>

2.5 Schedulable_Resource.Scheduling_Parameters

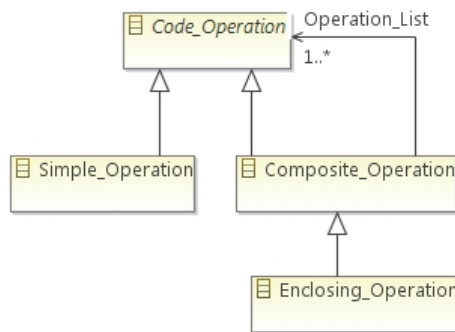


a) Un recurso planificable podría contener parámetros de planificación incompatibles con su naturaleza

Severidad de problema	Error
Restricción de integridad	Su tipo ha de corresponderse con el tipo de sus parámetros de planificación, según la tabla
Formulación OCL	<pre> context Thread inv i_2_5_a_I: Scheduling_Parameters.ocIsKindOf(Interrupt_Based_Params) or Scheduling_Parameters.ocIsTypeOf(Non_Preemptible_FP_Params) or Scheduling_Parameters.ocIsTypeOf(Fixed_Priority_Params) or Scheduling_Parameters.ocIsTypeOf(Polling_Params) or Scheduling_Parameters.ocIsKindOf(Periodic_Server_Params) or Scheduling_Parameters.ocIsKindOf(EDF_Based_Params) or Scheduling_Parameters.ocIsKindOf(Timetable_Driven_Params) context Communication_Channel inv i_2_5_a_II: Scheduling_Parameters.ocIsTypeOf(Non_Preemptible_FP_Params) or Scheduling_Parameters.ocIsTypeOf(Fixed_Priority_Params) or Scheduling_Parameters.ocIsTypeOf(Polling_Params) or Scheduling_Parameters.ocIsKindOf(Periodic_Server_Comm_Params) or Scheduling_Parameters.ocIsKindOf(EDF_Based_Params) or Scheduling_Parameters.ocIsKindOf(Timetable_Driven_Params) or Scheduling_Parameters.ocIsTypeOf(AFDX_Virtual_Link) context Mast_Model inv i_2_5_a_I: Element_List -> select(e e.ocIsTypeOf(Thread)) -> forAll(t t.ocAsType(Thread).Scheduling_Parameters.ocIsKindOf(Interrupt_Based_Params) or t.ocAsType(Thread).Scheduling_Parameters.ocIsTypeOf(Non_Preemptible_FP_Params) or t.ocAsType(Thread).Scheduling_Parameters.ocIsTypeOf(Fixed_Priority_Params) or t.ocAsType(Thread).Scheduling_Parameters.ocIsTypeOf(Polling_Params) or t.ocAsType(Thread).Scheduling_Parameters.ocIsTypeOf(Periodic_Server_Params) or t.ocAsType(Thread).Scheduling_Parameters.ocIsKindOf(EDF_Based_Params) or t.ocAsType(Thread).Scheduling_Parameters.ocIsKindOf(Timetable_Driven_Params)) context Mast_Model inv i_2_5_a_II: Element_List -> select(e e.ocIsTypeOf(Communication_Channel)) -> forAll(cc cc.ocAsType(Communication_Channel).Scheduling_Parameters.ocIsTypeOf(Non_Preemptible_FP_Params)) or cc.ocAsType(Communication_Channel).Scheduling_Parameters.ocIsTypeOf(Fixed_Priority_Params) or cc.ocAsType(Communication_Channel).Scheduling_Parameters.ocIsTypeOf(Polling_Params) or cc.ocAsType(Communication_Channel).Scheduling_Parameters.ocIsKindOf(Periodic_Server_Comm_Params) or cc.ocAsType(Communication_Channel).Scheduling_Parameters.ocIsKindOf(EDF_Based_Params) or cc.ocAsType(Communication_Channel).Scheduling_Parameters.ocIsKindOf(Timetable_Driven_Params)) or cc.ocAsType(Communication_Channel).Scheduling_Parameters.ocIsTypeOf(AFDX_Virtual_Link) </pre>

Schedulable resource type	Scheduling parameters type
Thread	<i>Interrupt_Based_Params</i> <i>Periodic_Server_Params</i> (<i>Sporadic_Server_Params</i>)
	<i>Non_Preemptible_FP_Params</i> <i>Fixed_Priority_Params</i> <i>Polling_Params</i> <i>EDF_Based_Params</i> <i>Timetable_Driven_Params</i>
Communication_Channel	<i>Periodic_Server_Comm_Params</i> (<i>Sporadic_Server_Comm_Params</i>) <i>ADFX_Virtual_Link</i>
Virtual_Schedulable_Resource	<i>Virtual_Resource_Params</i>
Virtual_Communication_Channel	<i>Virtual_Comm_Channel_Params</i>

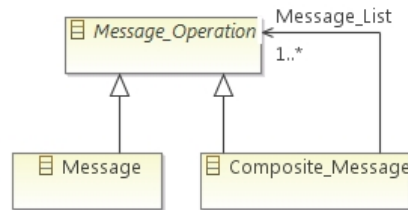
2.6 Composite_Operation.Operation_List



a) Una operación compuesta podría autoreferenciarse

Severidad de problema	Error
Restricción de integridad	Una operación compuesta no puede autoreferenciarse
Formulación OCL	<pre> context Composite_Operation inv i_2_6_a: not Operation_List -> includes(self) context Mast_Model inv i_2_6_a: Element_List -> select(e e.oclIsKindOf(Composite_Operation)) -> forAll(co not co.oclAsType(Composite_Operation).Operation_List -> includes(co)) </pre>

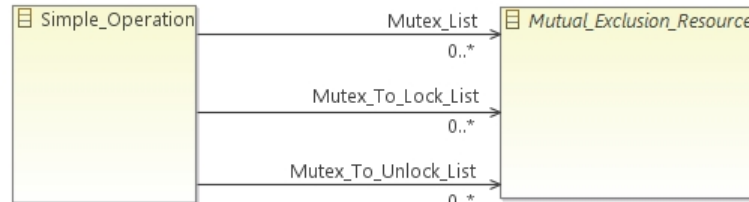
2.7 Composite_Message.Message_List



a) Un mensaje compuesto podría autoreferenciarse

Severidad de problema	Error
Restricción de integridad	Un mensaje compuesto no puede autoreferenciarse
Formulación OCL	<pre> context Composite_Message inv i_2_7_a: not Message_List -> includes(self) context Mast_Model inv i_2_7_a: Element_List -> select(e e.oclIsTypeOf(Composite_Message)) -> forAll(cm not cm.oclAsType(Composite_Message).Message_List -> includes(cm)) </pre>

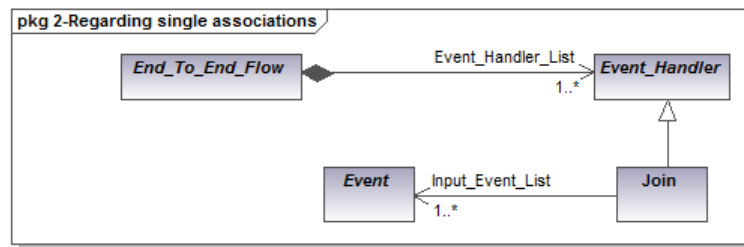
2.8 Simple_Operation.Mutex_List



a) Un mutex podría estar sin usar

Severidad de problema	Warning
Restricción de integridad	Un mutex ha de ser referenciado por al menos una operación simple
Formulación OCL	<pre> context Mutual_Exclusion_Resource inv i_2_8_a: Simple_Operation.allInstances() -> exists(so so.Mutex_List -> includes(self) or so.Mutex_To_Lock_List -> includes(self) or so.Mutex_To_Unlock_List -> includes(self)) context Mast_Model inv i_2_8_a: Element_List -> select(e e.oclIsKindOf(Mutual_Exclusion_Resource)) -> forAll(m Simple_Operation.allInstances() -> exists(so so.Mutex_List -> includes(m) or so.Mutex_To_Lock_List -> includes(m) or so.Mutex_To_Unlock_List -> includes(m))) </pre>

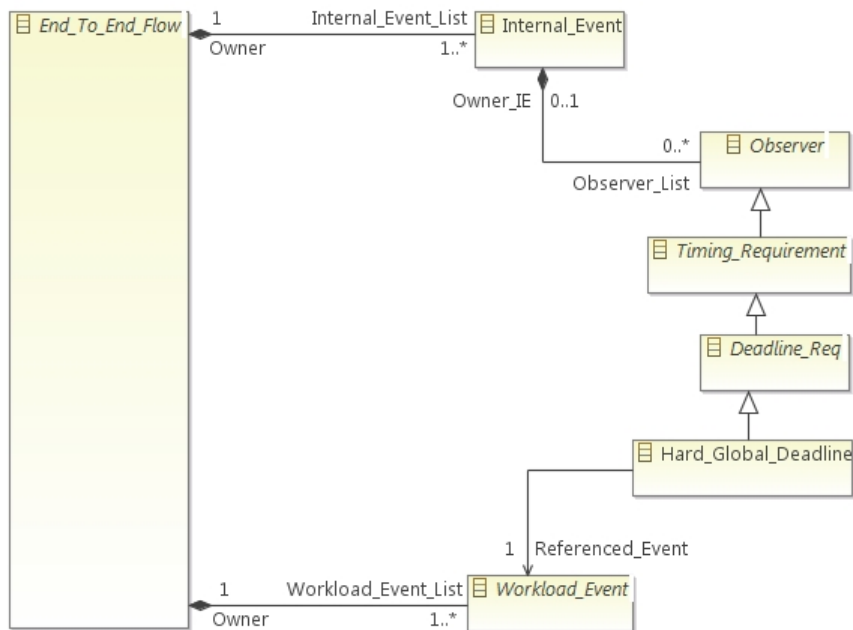
2.9 Join.Input_Event_List



a) *Un join podría tener eventos de entrada procedentes de distintos eventos externos*

Severidad de problema	Error
Restricción de integridad	Todos los eventos de entrada de un <i>join</i> han de proceder del mismo evento externo (el cual ha de pertenecer a la misma transacción que el <i>join</i>)
Formulación OCL	<pre> context Join inv i_2_9_a: </pre>

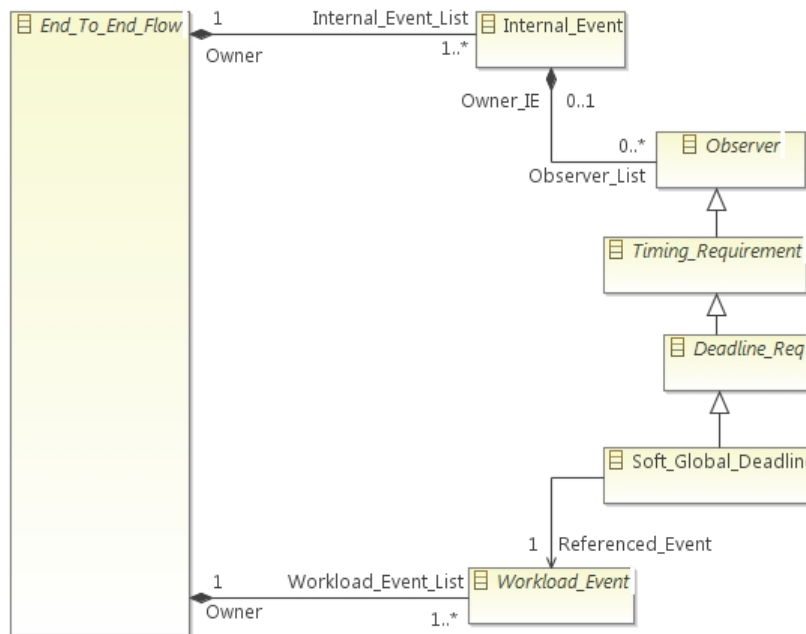
2.10 Hard_Global_Deadline.Referenced_Event



- a) *Un deadline global- estricto podría referenciar un evento externo diferente de aquel de donde procede su línea de flujo*

Severidad de problema	Error
Restricción de integridad	Un deadline global-estricto sólo puede referenciar al evento externo de donde procede su línea de flujo
Formulación OCL	<pre> context Hard_Global_DeadLine inv i_2_10_a: </pre>

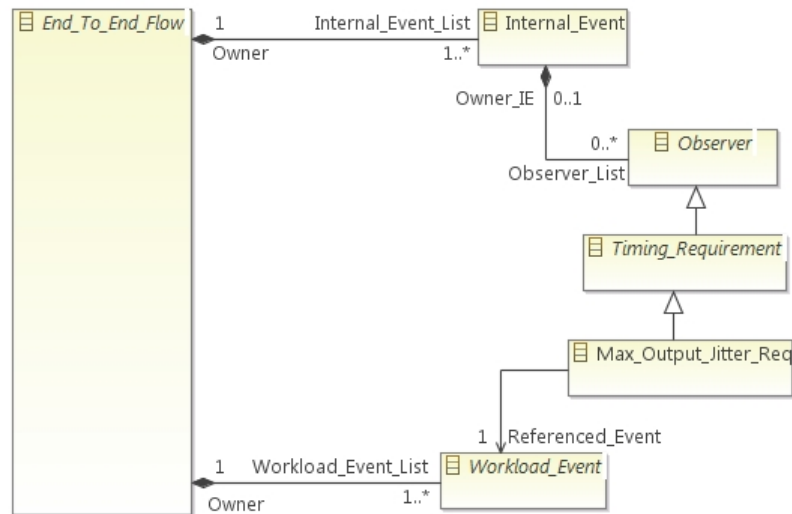
2.11 Soft_Global_Deadline.Referenced_Event



- a) *Un deadline global-laxo podría referenciar un evento externo diferente de aquel de donde procede su línea de flujo*

Severidad de problema	Error
Restricción de integridad	Un deadline global-laxo sólo puede referenciar al evento externo de donde procede su línea de flujo
Formulación OCL	<pre> context Soft_Global_DeadLine inv i_2_11_a: </pre>

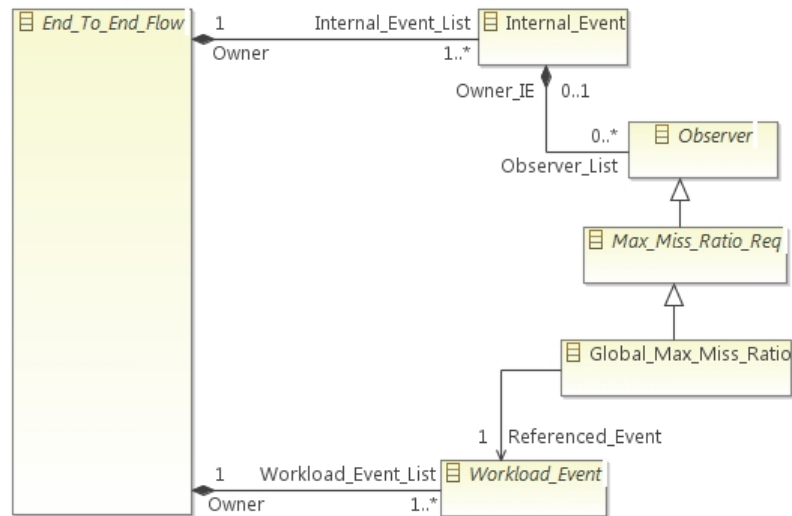
2.12 Max_Output_Jitter_Req.Referenced_Event



- a) Un requerimiento relativo al máximo jitter de salida podría referenciar un evento externo diferente de aquel de donde procede su línea de flujo

Severidad de problema	Error
Restricción de integridad	Un requerimiento relativo al máximo <i>jitter</i> de salida sólo puede referenciar al evento externo de donde procede su línea de flujo
Formulación OCL	<pre> context Max_Output_Jitter_Req inv i_2_12_a: </pre>

2.13 Global_Max_Miss_Ratio.Referenced_Event

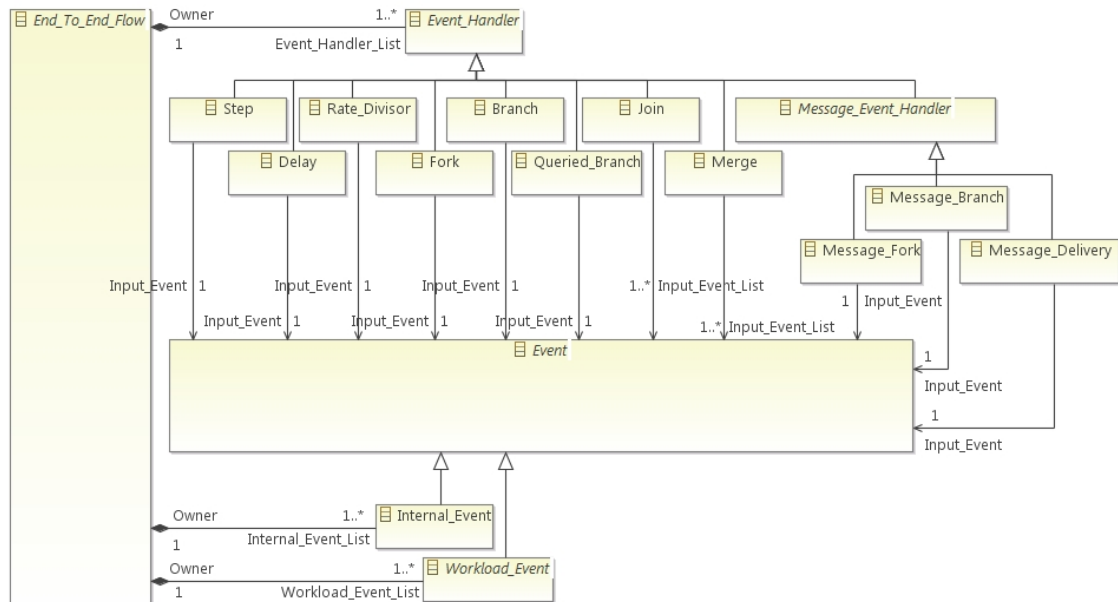


- a) *Un requerimiento relativo al máximo ratio de deadlines globales perdidos podría referenciar un evento externo diferente de aquel de donde procede su línea de flujo*

Severidad de problema	Error
Restricción de integridad	Un requerimiento relativo al máximo ratio de deadlines globales perdidos sólo puede referenciar al evento externo de donde procede su línea de flujo
Formulación OCL	<pre> context Global_Max_Miss_Ratio inv i_2_13_a: </pre>

2.14 {Step,
 Delay, Offset
 Rate_Divisor,
 Fork,
 Branch,
 Queried_Branch,
 Message_Delivery,
 Message_Fork,
 Message_Branch}.Input_Event

{Join,
 Merge}.Input_Event_List



a) Un manejador de eventos podría referenciar como entrada un evento de otra transacción

Severidad de problema	Error
Restricción de integridad	Un manejador de eventos sólo puede referenciar como entrada a un evento de su misma transacción
Formulación OCL	<pre> context Step inv i_2_14_a_Step: if (Input_Event.ocIsTypeOf(Internal_Event)) then Owner.Internal_Event_List -> includes(Input_Event) else Owner.Workload_Event_List -> includes(Input_Event) endif context Join inv i_2_14_a_Join: Input_Event_List -> forAll(e if (e.ocIsTypeOf(Internal_Event)) then Owner.Internal_Event_List -> includes(e) else </pre>

	<code>Owner.WorkLoad_Event_List -> includes(e)</code> <code>endif)</code>
--	---

b) *Un evento externo podría estar sin usar*

Severidad de problema	Warning
Restricción de integridad	Todo evento externo ha de ser entrada de al menos un manejador de eventos (de su misma transacción)
Formulación OCL	<pre> context WorkLoad_Event inv i_2_14_b: Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Step)) -> exists(s s.ocLAsType(Step).Input_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Delay)) -> exists(d d.ocLAsType(Delay).Input_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Rate_Divisor)) -> exists(rd rd.ocLAsType(Rate_Divisor).Input_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Fork)) -> exists(f f.ocLAsType(Fork).Input_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Branch)) -> exists(b b.ocLAsType(Branch).Input_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Queried_Branch)) -> exists(qb qb.ocLAsType(Queried_Branch).Input_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Join)) -> exists(j j.ocLAsType(Join).Input_Event_List -> includes(self)) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Merge)) -> exists(m m.ocLAsType(Merge).Input_Event_List -> includes(self)) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Message_Delivery)) -> exists(md md.ocLAsType(Message_Delivery).Input_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Message_Fork)) -> exists(mf mf.ocLAsType(Message_Fork).Input_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Message_Branch)) -> exists(mb mb.ocLAsType(Message_Branch).Input_Event = self) </pre>

c) *Un evento externo podría estar sobre referenciado*

Severidad de problema	Error
Restricción de integridad	Un evento externo no puede ser entrada de más de un manejador de eventos
Formulación OCL	<pre> context WorkLoad_Event inv i_2_14_c: (Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Step)) -> select(s s.ocLAsType(Step).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Delay)) -> select(d d.ocLAsType(Delay).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Rate_Divisor)) -> select(rd rd.ocLAsType(Rate_Divisor).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Fork)) -> select(f f.ocLAsType(Fork).Input_Event = self) -> size() + </pre>

	<pre> Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Branch)) -> select(b b.ocLAsType(Branch).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Queried_Branch)) -> select(qb qb.ocLAsType(Queried_Branch).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Join)) -> select(j j.ocLAsType(Join).Input_Event_List -> includes(self)) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Merge)) -> select(m m.ocLAsType(Merge).Input_Event_List -> includes(self)) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Message_Delivery)) -> select(md md.ocLAsType(Message_Delivery).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Message_Fork)) -> select(mf mf.ocLAsType(Message_Fork).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Message_Branch)) -> select(mb mb.ocLAsType(Message_Branch).Input_Event = self) -> size()) < 2 </pre>

d) Un evento interno podría estar sobre referenciado (como entrada)

Severidad de problema	Error
Restricción de integridad	Un evento interno no puede ser entrada de más de un manejador de eventos
Formulación OCL	<pre> context Internal_Event inv i_2_14_d: (Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Step)) -> select(s s.ocLAsType(Step).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Delay)) -> select(d d.ocLAsType(Delay).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Rate_Divisor)) -> select(rd rd.ocLAsType(Rate_Divisor).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Fork)) -> select(f f.ocLAsType(Fork).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Branch)) -> select(b b.ocLAsType(Branch).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Queried_Branch)) -> select(qb qb.ocLAsType(Queried_Branch).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Join)) -> select(j j.ocLAsType(Join).Input_Event_List -> includes(self)) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Merge)) -> select(m m.ocLAsType(Merge).Input_Event_List -> includes(self)) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Message_Delivery)) -> select(md md.ocLAsType(Message_Delivery).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Message_Fork)) -> select(mf mf.ocLAsType(Message_Fork).Input_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIIsTypeOf(Message_Branch)) -> select(mb mb.ocLAsType(Message_Branch).Input_Event = self) -> size()) < 2 </pre>

2.15 {Step,
 Delay, Offset
 Rate_Divisor,
 Join,
 Merge,
 Message_Delivery}.Output_Event

{Fork,
 Branch,
 Queried_Branch,
 Message_Fork,
 Message_Branch}.Output_Event_List

a) Un manejador de eventos podría referenciar como salida un evento interno de otra transacción

Severidad de problema	Error
Restricción de integridad	Un manejador de eventos sólo puede referenciar como salida a un evento interno de su misma transacción
Formulación OCL	<pre>context Step inv i_2_15_a_Step: Owner.Internal_Event_List -> includes(Output_Event) context Fork inv i_2_15_a_Fork: Output_Event_List -> forAll(e Owner.Internal_Event_List -> includes(e))</pre>

b) Un evento interno podría estar sin usar

Severidad de problema	Warning
Restricción de integridad	Todo evento interno ha de ser salida de al menos un manejador de eventos (de su misma transacción)
Formulación OCL	<pre>context Internal_Event inv i_2_15_b: Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Step)) -> exists(s s.ocLAsType(Step).Output_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Delay)) -> exists(d d.ocLAsType(Delay).Output_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Rate_Divisor)) -> exists(rd rd.ocLAsType(Rate_Divisor).Output_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Fork)) -> exists(f f.ocLAsType(Fork).Output_Event_List -> includes(self)) or</pre>

<pre> Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Branch)) -> exists(b b.ocLAsType(Branch).Output_Event_List -> includes(self)) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Queried_Branch)) -> exists(qb qb.ocLAsType(Queried_Branch).Output_Event_List -> includes(self)) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Join)) -> exists(j j.ocLAsType(Join).Output_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Merge)) -> exists(m m.ocLAsType(Merge).Output_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Message_Delivery)) -> exists(md md.ocLAsType(Message_Delivery).Output_Event = self) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Message_Fork)) -> exists(mf mf.ocLAsType(Message_Fork).Output_Event_List -> includes(self)) or Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Message_Branch)) -> exists(mb mb.ocLAsType(Message_Branch).Output_Event_List -> includes(self)) </pre>

c) Un evento interno podría estar sobre referenciado (como salida)

Severidad de problema	Error
Restricción de integridad	Un evento interno no puede ser salida de más de un manejador de eventos
Formulación OCL	<pre> context Internal_Event inv i_2_15_c: (Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Step)) -> select(s s.ocLAsType(Step).Output_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Delay)) -> select(d d.ocLAsType(Delay).Output_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Offset)) -> select(o o.ocLAsType(Offset).Output_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Rate_Divisor)) -> select(rd rd.ocLAsType(Rate_Divisor).Output_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Fork)) -> select(f f.ocLAsType(Fork).Output_Event_List -> includes(self)) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Branch)) -> select(b b.ocLAsType(Branch).Output_Event_List -> includes(self)) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Queried_Branch)) -> select(qb qb.ocLAsType(Queried_Branch).Output_Event_List -> includes(self)) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Join)) -> select(j j.ocLAsType(Join).Output_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Merge)) -> select(m m.ocLAsType(Merge).Output_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Message_Delivery)) -> select(md md.ocLAsType(Message_Delivery).Output_Event = self) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Message_Fork)) -> select(mf mf.ocLAsType(Message_Fork).Output_Event_List -> includes(self)) -> size() + Owner.Event_Handler_List -> select(eh eh.ocLIsTypeOf(Message_Branch)) -> select(mb mb.ocLAsType(Message_Branch).Output_Event_List -> includes(self)) -> size()) < 2 </pre>

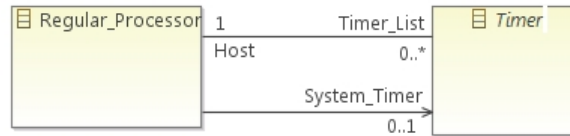
2.16 Internal_Event.Observer_List

a) *Cualquier evento interno podría tener observadores*

Severidad de problema	Error
Restricción de integridad	Sólo los eventos internos final de segmento pueden tener observadores
Formulación OCL	<pre>context inv i_2_16_a:</pre>
	<pre>context Mast_Model inv i_2_16_a: Step.allInstances() -> forAll(s s.Owner.Internal_Event_List -> includes(s.Output_Event))</pre>

3 Laxitudes relacionadas con conjuntos de asociaciones

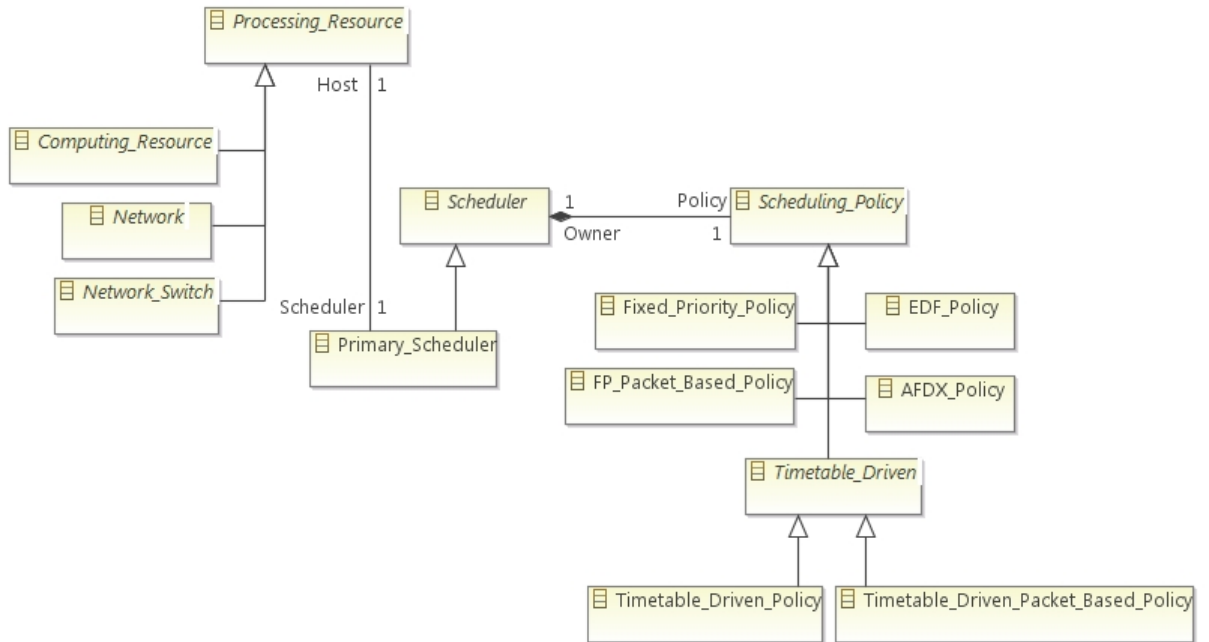
3.1 Regular_Processor.System_Timer Regular_Processor.Timer_List



a) Un procesador podría tener un temporizador de sistema no incluido en su lista de temporizadores

Severidad de problema	Error
Restricción de integridad	El temporizador de sistema ha de ser uno de entre los referenciados en la lista de temporizadores
Formulación OCL	<pre> context Regular_Processor inv i_3_1_a: if not System_Timer.oclIsUndefined() then Timer_List -> includes(System_Timer) else true endif </pre> <pre> context Mast_Model inv i_3_1_a: Regular_Processor.allInstances() -> forALL(rp rp.Timer_List -> includes(rp.System_Timer)) </pre>

3.2 Primary_Scheduler.Host Scheduler.Policy



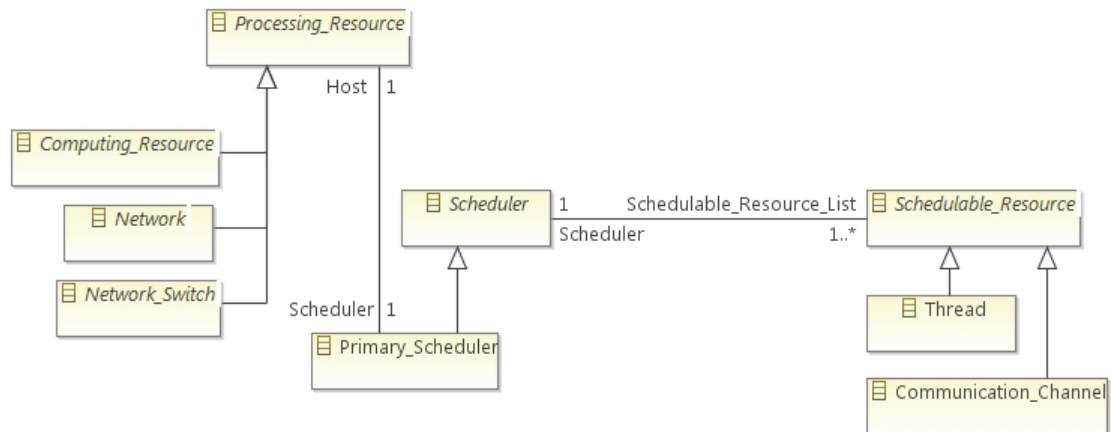
a) Un planificador primario podría tener una política de planificación incompatible con su host

Severidad de problema	Error
Restricción de integridad	El tipo de la política de planificación de un planificador ha de corresponderse con el tipo de su host según la siguiente tabla
Formulación OCL	<pre> context Primary_Scheduler inv i_3_2_a: ((Policy.ocIsTypeOf(Fixed_Priority_Policy) or Policy.ocIsTypeOf(EDF_Policy) or Policy.ocIsTypeOf(Timetable_Driven_Policy)) and Host.ocIsKindOf(Computing_Resource)) or ((Policy.ocIsTypeOf(FP_Packet_Based_Policy) or Policy.ocIsTypeOf(EDF_Policy) or Policy.ocIsTypeOf(Timetable_Driven_Packet_Based_Policy)) and Host.ocIsTypeOf(Packet_Based_Network)) or (Policy.ocIsTypeOf(AFDX_Policy) and Host.ocIsTypeOf(AFDX_Link)) context Mast_Model inv i_3_2_a: Primary_Scheduler.allInstances() -> forALL(ps ((ps.Policy.ocIsTypeOf(Fixed_Priority_Policy) or ps.Policy.ocIsTypeOf(EDF_Policy) or ps.Policy.ocIsTypeOf(Timetable_Driven_Policy)) and ps.Host.ocIsKindOf(Computing_Resource)) or ((ps.Policy.ocIsTypeOf(FP_Packet_Based_Policy) or ps.Policy.ocIsTypeOf(EDF_Policy) or ps.Policy.ocIsTypeOf(Timetable_Driven_Packet_Based_Policy)) and ps.Host.ocIsKindOf(Packet_Based_Network)) or (ps.Policy.ocIsTypeOf(AFDX_Policy) and ps.Host.ocIsKindOf(AFDX_Link))) </pre>

Scheduling policy type	Host type
------------------------	-----------

Fixed_Priority_Policy	<i>Computing_Resource</i>
Timetable_Driven_Policy	
EDF_Policy	Packet_Based_Network
FP_Packet_Based_Policy	
Timetable_Driven_Packet_Based_Policy	
AFDX_Policy	AFDX_Link

3.3 Primary_Scheduler.Host Schedulable_Resource.Scheduler



a) Un recurso planificable podría ser incompatible con el tipo de host de su planificador (primario)

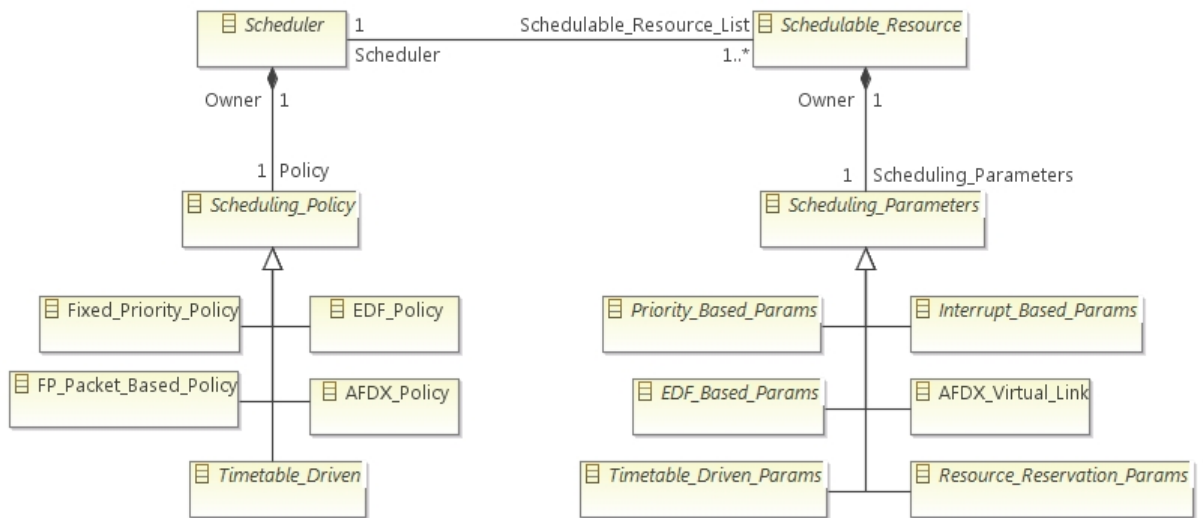
Severidad de problema	Error
Restricción de integridad	El tipo de un recurso planificable ha de corresponderse con el tipo del host de su planificador (primario) según la siguiente tabla
Formulación OCL	<pre> context Thread inv i_3_3_a_I: self.Scheduler.ocLAsType(Primary_Scheduler).Host.ocLIsKindOf(Computing_Resource) context Communication_Channel inv i_3_3_a_II: self.Scheduler.ocLAsType(Primary_Scheduler).Host.ocLIsKindOf(Network) context Mast_Model inv i_3_3_a_I: Thread.allInstances() -> forAll(t t.Scheduler.ocLAsType(Primary_Scheduler).Host.ocLIsKindOf(Computing_Resource)) context Mast_Model inv i_3_3_a_II: Communication_Channel.allInstances() -> forAll(c c.Scheduler.ocLAsType(Primary_Scheduler).Host.ocLIsKindOf(Network)) </pre>

Schedulable resource type	Host type
Thread	Computing_Resource
Communication_Channel	Network

3.4 Scheduler.Policy

Schedulable_Resource.Scheduling_Parameters

Schedulable_Resource.Scheduler



a) Un recurso planificable podría tener parámetros de planificación incompatibles con la política de planificación de su planificador (primario)

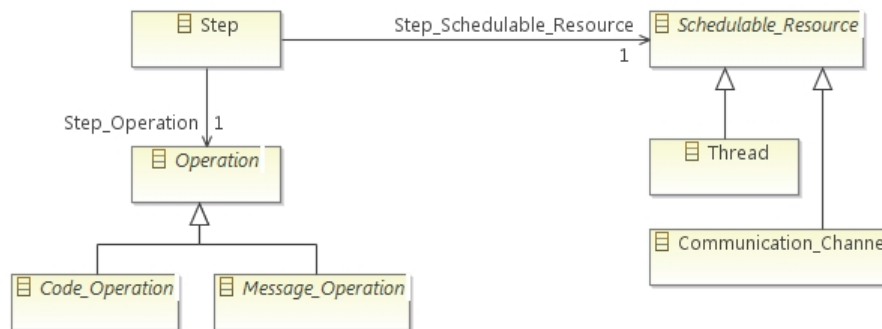
Severidad de problema	Error
Restricción de integridad	El tipo de los parámetros de planificación de un recurso planificable ha de corresponderse con el tipo de la política de planificación de su planificador (primario) según la siguiente tabla
Formulación OCL	<pre> context Schedulable_Resource inv i_3_4_a: self.Scheduling_Parameters.ocIsKindOf(Priority_Based_Params) and self.Scheduler.Policy.ocIsTypeOf(Fixed_Priority_Policy) or self.Scheduling_Parameters.ocIsKindOf(Interrupt_Based_Params) and self.Scheduler.Policy.ocIsTypeOf(Fixed_Priority_Policy) or self.Scheduling_Parameters.ocIsKindOf(EDF_Based_Params) and self.Scheduler.Policy.ocIsTypeOf(EDF_Policy) or self.Scheduling_Parameters.ocIsKindOf(Timetable_Driven_Params) and self.Scheduler.Policy.ocIsTypeOf(Timetable_Driven_Policy) or self.Scheduling_Parameters.ocIsTypeOf(AFDX_Virtual_Link) and self.Scheduler.Policy.ocIsTypeOf(AFDX_Policy) context Mast_Model inv i_3_4_a: Schedulable_Resource.allInstances() -> forAll(sr sr.Scheduling_Parameters.ocIsKindOf(Priority_Based_Params) and sr.Scheduler.Policy.ocIsTypeOf(Fixed_Priority_Policy) or sr.Scheduling_Parameters.ocIsKindOf(Interrupt_Based_Params) and sr.Scheduler.Policy.ocIsTypeOf(Fixed_Priority_Policy) or sr.Scheduling_Parameters.ocIsKindOf(EDF_Based_Params) and sr.Scheduler.Policy.ocIsTypeOf(EDF_Policy) or sr.Scheduling_Parameters.ocIsKindOf(Timetable_Driven_Params) and sr.Scheduler.Policy.ocIsTypeOf(Timetable_Driven_Policy)) </pre>

Scheduling parameters type	Scheduling policy type
<i>Priority_Based_Params</i>	Fixed_Priority_Policy
<i>Interrupt_Based_Params</i>	
<i>EDF_Based_Params</i>	EDF_Policy
<i>Timetable_Driven_Params</i>	Timetable_Driven_Policy
AFDX_Virtual_Link	AFDX_Policy
<i>Resource_Reservation_Params</i>	????????

b) Un recurso planificable con parámetros de planificación basados en prioridades fijas podría tener una prioridad fuera de rango

Severidad de problema	Error
Restricción de integridad	La prioridad ha de estar dentro del rango especificado por la política (basada en prioridades fijas) del planificador
Formulación OCL	<pre> context Thread inv i_3_4_b_I: if self.Scheduling_Parameters.ocIsKindOf(Priority_Based_Params) then self.Scheduling_Parameters.ocAsType(Priority_Based_Params).Priority <= self.Scheduler.Policy.ocAsType(Fixed_Priority_Policy).Max_Priority and self.Scheduling_Parameters.ocAsType(Priority_Based_Params).Priority >= self.Scheduler.Policy.ocAsType(Fixed_Priority_Policy).Min_Priority else true endif context Communication_Channel inv i_3_4_b_II: if self.Scheduling_Parameters.ocIsKindOf(Priority_Based_Params) then self.Scheduling_Parameters.ocAsType(Priority_Based_Params).Priority <= self.Scheduler.Policy.ocAsType(FP_Packet_Based_Policy).Max_Priority and self.Scheduling_Parameters.ocAsType(Priority_Based_Params).Priority >= self.Scheduler.Policy.ocAsType(FP_Packet_Based_Policy).Min_Priority else true endif </pre>

3.5 Step.Step_Operation Step.Step_Schedulable_Resource

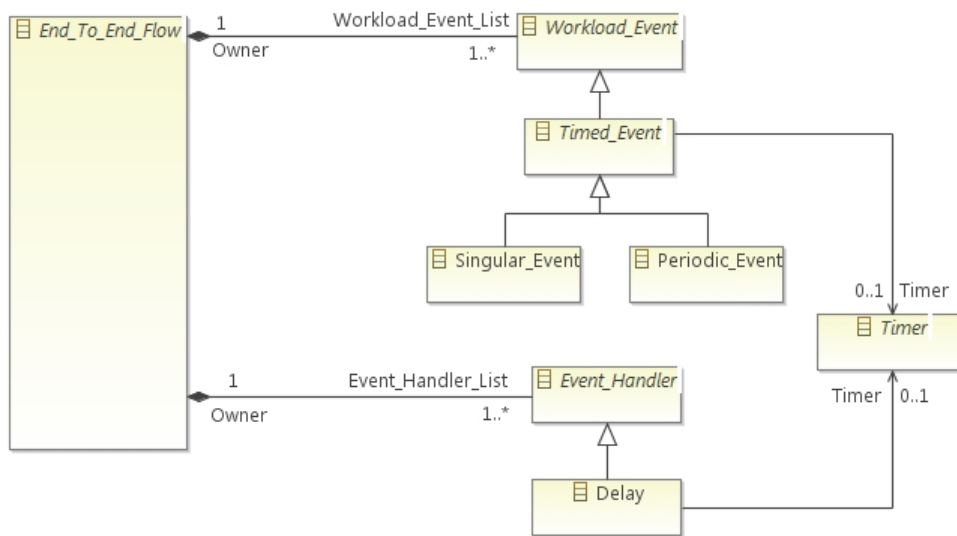


a) Una actividad podría estar asociada a una pareja imposible de operación / recurso planificable

Severidad de problema	Error
Restricción de integridad	El tipo de la operación asociada a una actividad y el tipo del recurso planificable asociado han de corresponderse según la siguiente tabla
Formulación OCL	<pre> context Step inv i_3_5_a: Step_Operation.ocIsKindOf(Code_Operation) and Step_Schedulable_Resource.ocIsTypeOf(Thread) or Step_Operation.ocIsKindOf(Message_Operation) and Step_Schedulable_Resource.ocIsTypeOf(Communication_Channel) </pre>
	<pre> context Mast_Model inv i_3_5_a: Step.allInstances() -> forAll(s s.Step_Operation.ocIsKindOf(Code_Operation) and s.Step_Schedulable_Resource.ocIsTypeOf(Thread) or s.Step_Operation.ocIsKindOf(Message_Operation) and s.Step_Schedulable_Resource. ocIsTypeOf(Communication_Channel)) </pre>

Schedulable resource type	Operation type
Thread	Code_Operation
Communication_Channel	Message_Operation

3.6 Timed_Event.Timer Delay.Timer



a) Un temporizador podría estar sin utilizar

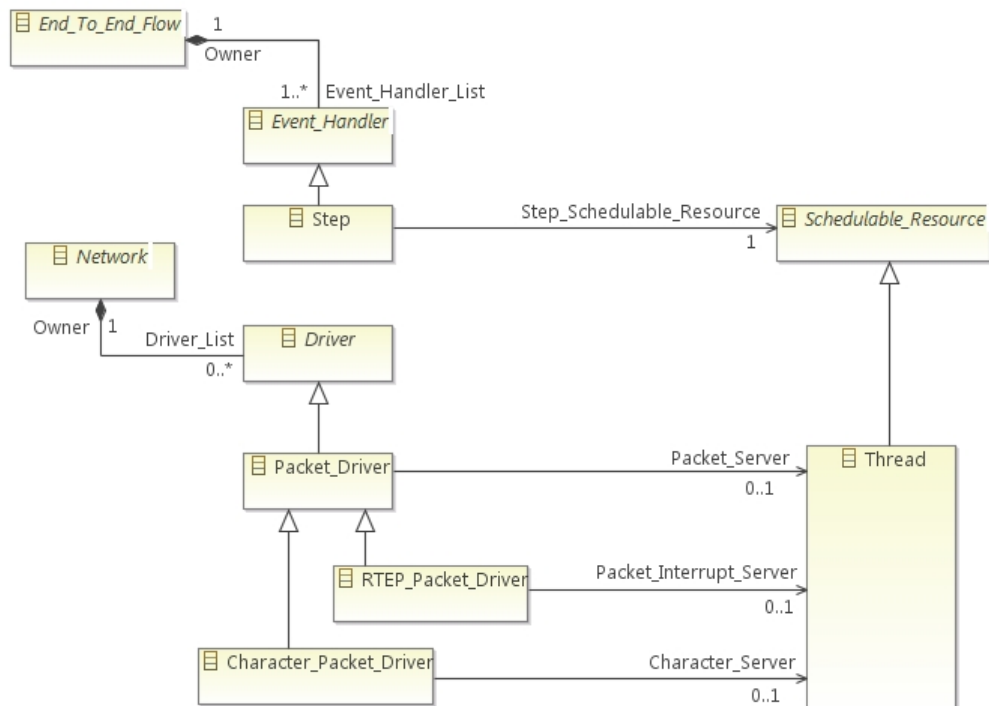
Severidad de problema	Warning
Restricción de integridad	Todo temporizador ha de ser referenciado por al menos una transacción (a través de sus eventos temporizados o retrasos)
Formulación OCL	<pre> context Timer inv i_3_6_a: Timed_Event.allInstances() -> exists(te te.Timer = self) or Delay.allInstances() -> exists(d d.Timer = self) </pre>

3.7 Step.Step_Schedulable_Resource

Packet_Driver.Packet_Server

Character_Packet_Driver.Character_Server

RTEP_Packet_Driver.Packet_Interrupt_Server



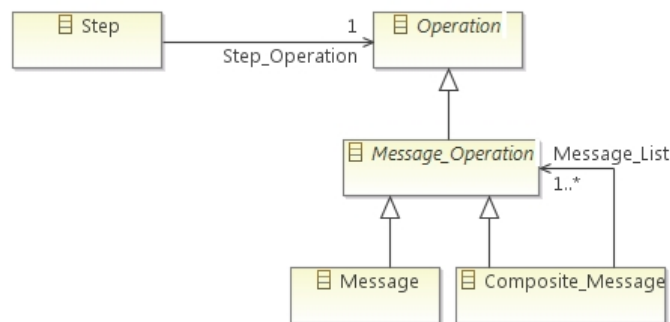
a) Un thread podría estar sin usar

Severidad de problema	Warning
Restricción de integridad	Todo thread ha de estar referenciado por al menos una transacción (a través de sus actividades) OR por al menos un driver
Formulación OCL	<pre> context Thread inv i_3_7_a: Step.allInstances() -> exists(s s.Step_Schedulable_Resource = self) or Packet_Driver.allInstances() -> exists(pd pd.Packet_Server = self) or Character_Packet_Driver.allInstances() -> exists(cpd cpd.Character_Server = self) or RTEP_Packet_Driver.allInstances() -> exists(rpd rpd.Packet_Interrupt_Server = self) </pre>

b) Un canal de comunicaciones podría estar sin usar

Severidad de problema	Warning
Restricción de integridad	Todo canal de comunicaciones ha de estar referenciado por al menos una transacción (a través de sus actividades)
Formulación OCL	<pre> context Communication_Channel inv i_3_7_b: Step.allInstances() -> exists(s s.Step_Schedulable_Resource = self) </pre>

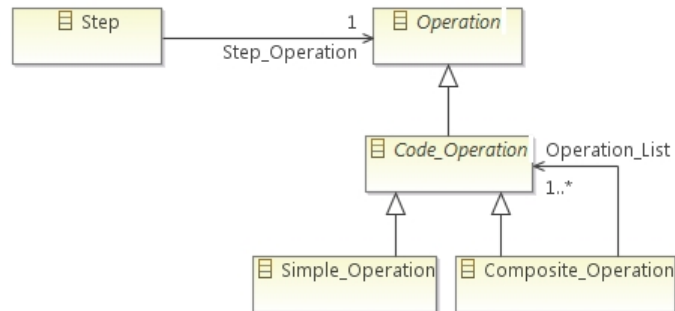
3.8 Step.Step_Operation Composite_Message.Message_List



a) Un mensaje podría estar sin utilizar

Severidad de problema	Warning
Restricción de integridad	Todo mensaje ha de ser referenciado por al menos una transacción (a través de sus actividades) o por al menos un mensaje compuesto
Formulación OCL	<pre> context Message_Operation inv i_3_8_a: Step.allInstances() -> exists(s s.Step_Operation = self) or Composite_Message.allInstances() -> exists(cm cm.Message_List -> includes(self)) </pre>

3.9 Step.Step_Operation Composite_Operation.Operation_List Packet_Driver.Packet_Send_Operation etc.



a) Una operación de código podría estar sin utilizar

Severidad de problema	Warning
Restricción de integridad	Toda operación de código ha de ser referenciada por al menos una transacción (a través de sus actividades) o por al menos un driver o por al menos una operación compuesta.
Formulación OCL	<pre> context Code_Operation inv i_3_9_a: Step.allInstances() -> exists(s s.Step_Operation = self) or Packet_Driver.allInstances() -> exists(pd pd.Packet_Receive_Operation = self or pd.Packet_Send_Operation = self) or Character_Packet_Driver.allInstances() -> exists(cpd cpd.Character_Receive_Operation = self or cpd.Character_Send_Operation = self) or RTEP_Packet_Driver.allInstances() -> exists(rpd rpd.Packet_ISR_Operation = self or rpd.Packet_Discard_Operation = self or rpd.Packet_Retransmission_Operation = self or rpd.Token_Retransmission_Operation = self or rpd.Token_Manage_Operation = self or rpd.Token_Check_Operation = self) or Composite_Operation.allInstances() -> exists(co co.Operation_List -> includes(self)) </pre>

3.10 Simple_Operation.Mutex_List

Simple_Operation.Mutex_To_Lock_List

Simple_Operation.Mutex_To_Unlock_List

- a) *Un mutex podría aparecer en la lista de mutexes de una operación y también en su lista de mutexes tomados*

Severidad de problema	Error
Restricción de integridad	Un mutex que aparece en la lista de mutexes de una operación no puede aparecer en su lista de mutexes tomados
Formulación OCL	<pre>context Simple_Operation inv i_3_10_a: not ((not Mutex_List -> isEmpty()) and (not Mutex_To_Lock_List -> isEmpty()))</pre>
	<pre>context Mast_Model inv i_3_10_a: Simple_Operation.allInstances() -> forAll(so not (so.Mutex_List -> size() > 0 and so.Mutex_To_Lock_List -> size() > 0))</pre>

- b) *Un mutex podría aparecer en la lista de mutexes de una operación y también en su lista de mutexes liberados*

Severidad de problema	Error
Restricción de integridad	Un mutex que aparece en la lista de mutexes de una operación no puede aparecer en su lista de mutexes liberados
Formulación OCL	<pre>context Simple_Operation inv i_3_10_b: not ((not Mutex_List -> isEmpty()) and (not Mutex_To_Unlock_List -> isEmpty()))</pre>
	<pre>context Mast_Model inv i_3_10_b: Simple_Operation.allInstances() -> forAll(so not (so.Mutex_List -> size() > 0 and so.Mutex_To_Unlock_List -> size() > 0))</pre>

- c) *Una actividad podría intentar tomar un mutex que ha sido tomado por otra actividad*

Severidad de problema	Error
Restricción de integridad	Un mutex no puede ser tomado por una actividad si no se encuentra libre
Formulación OCL	<pre>context inv i_3_10_c:</pre>

d) *Un mutex podría permanecer tomado al final de la línea de flujo*

Severidad de problema	Error
Restricción de integridad	Todo mutex debe ser liberado antes de terminarse la transacción
Formulación OCL	context inv i_3_10_d: