Canonical Artifact Formalization to Support the Automatic Generation of Hazard Relation Diagrams

Bastian Tenbergen[†,‡], *Thorsten Weyer*[‡]

[†] *Department of Computer Science, State University of New York at Oswego, United States*

[‡] *paluno – The Ruhr Institute for Software Technology, University of Duisburg-Essen, Germany*

1. Canonical Artifact Type Formalization

1.1. Functional Requirements

In the following, we simplify the UML activity diagrams proposed in to incorporate only those concepts needed to express system functionality that is input for hazard analysis worksheets (i.e. activities and actions). Since these are equivalent with regard to their consideration in hazard analyses, we use the term "activity" to describe opaque actions, activites, events, etc. In addition, it is to note that in contrast to semantic formalization of UML activity diagrams, e.g., as proposed in, the following formalism represents a purely canonical form of simplified UML activity diagram representations.

For this work, every activity diagram $actD$ can be described as a tuple

$$actD = (ad, A^{ad}, E^{ad}, P^{ad}, C^{ad}) \tag{1.1}$$

where ad is a unique, user-defined name of the activity diagram, $A^{ad} = \left\{a_1^{ad}, a_2^{ad}, \dots, a_n^{ad}\right\}$ is a set of activities in that activity diagram, $E^{ad} = \{e_1^{ad}, e_2^{ad}, \dots, e_n^{ad}\}$ is a set of activity edges, $P^{ad} = \{p_1^{ad}, p_2^{ad}, \dots, p_n^{ad}\}$ is a set of input/output ports, and $C^{ad} = F^{ad} \cup J^{ad} \cup D^{ad} \cup X^{ad}$ is a set of control nodes, which consists of a set of fork nodes $F^{ad} = \{f_1^{ad}, f_2^{ad}, \dots, f_n^{ad}\}$, a set of join nodes $J^{ad} = \{j_1^{ad}, j_2^{ad}, \dots, j_n^{ad}\}$, a set of decision nodes $D^{ad} = \{d_1^{ad}, d_2^{ad}, \dots, d_n^{ad}\}$, and a set of merge nodes $X^{ad} = \{x_1^{ad}, x_2^{ad}, \dots, x_n^{ad}\}$. An activity edge can be described as a tuple

$$e = (src^e, m^e, tar^e) | src^e, tar^e \in (A^{ad} \cup P^{ad} \cup C^{ad}) \tag{1.2}$$

where the source src^e and the target tar^e is an activity, a port, or a control node, respectively, with a message m^e being transmitted over the activity edge. An activity edge is called a control

flow if $m^e = \varepsilon$ and a data flow if $m^e \neq \varepsilon$. A port p is called an input port if it is the source of a data flow e, i.e.

$$P_{in}^{ad} \subseteq P^{ad} | \forall p \in P^{ad} : \exists e = (src^e, m, tar^e) \in E^{ad} \\ \wedge p = src^e \wedge m \neq \varepsilon \tag{1.3}$$

A port p is called an output port if it is the target of a data flow e, i.e.

$$P_{out}^{ad} \subseteq P^{ad} | \forall p \in P^{ad} : \exists e = (src^e, m, tar^e) \in E^{ad} \\ \wedge p = tar^e \wedge m \neq \varepsilon \tag{1.4}$$

To further simplify the following formalizations, it is assumed that Activities are specified without ActivityParameterNodes and that function parametrization is achived through the message on an activity edge.

A fork node $f \in F^{ad}$ is a control node that has one incoming activity edge and at least two outgoing activity edges which carry the same message, i.e.

$$\forall f \in F^{ad} : \exists e^{in} = \left(src^{in}, m^{in}, tar^{in}\right) \in E^{ad} \wedge tar^{in} = f \\ \wedge \exists E' = \{e_1, e_2, \dots, e_n\} \subseteq E^{ad} \wedge n \geq 2 \\ \wedge \forall e = (src^{out}, m^{out}, tar^{out}) \in E' : src^{out} = f \wedge m^{in} = m^{out} \tag{1.5}$$

A join node is a control node with multiple incoming edges and one outgoing edge:

$$\forall j \in J^{ad} : \exists E' = \{e_1, e_2, \dots, e_n\} \subseteq E^{ad} \wedge n \geq 2 \\ \wedge \forall e = \left(src^{in}, m^{in}, tar^{in}\right) \in E' \wedge tar^{in} = j \\ \wedge \exists e^{out} = (src^{out}, m^{out}, tar^{out}) \in E^{ad} \wedge src^{out} = \mathrm{j} \tag{1.6}$$

A decision node $d \in D^{ad}$ is a control node that has one incoming activity edge and at least two outgoing activity edges carrying the same message, and where each outgoing edge has a different guard:

$$\forall d \in D^{ad} : \exists e^{in} = \left(src^{in}, m^{in}, g^{in}, tar^{in}\right) \in E^{ad} \\ \wedge tar^{in} = d \wedge \exists E' = \{e_1, e_2, \dots, e_n\} \subseteq E^{ad} \wedge n \geq 2 \\ \wedge \forall e = (src^{out}, m^{out}, g^{out}, tar^{out}) \in E' \wedge src^{out} = d \\ \wedge g_{n-1}^{out} \neq g_n^{out} \wedge m^{in} = m_n^{out} \tag{1.7}$$

In (1.2), we defined activity edges as a 3-tuple without a guard, as the guard plays no role for the remainder of the formalization. However, in order to differentiate decision nodes from fork nodes, it is necessary to specifiy activity edges as a 4-tuple, as shown in (1.7).

A merge node is a control node merging multiple incoming edges after a decision node into one outgoing activity edge, i.e.

$$\begin{aligned} &\forall \mathrm{x} \in X^{ad}: \exists E' = \{e_1, e_2, \dots, e_n\} \subseteq E^{ad} \wedge n \geq 2 \\ &\wedge \forall e = \left(src^{in}, m^{in}, tar^{in}\right) \in E' \wedge tar^{in} = \mathrm{x} \\ &\wedge \exists e^{out} = (src^{out}, m^{out}, tar^{out}) \in E^{ad} \wedge src^{out} = \mathrm{x} \end{aligned} \tag{1.8}$$

Figure 1-1 shows the excerpt of the functional requirements of an Adaptive Cruise Control.

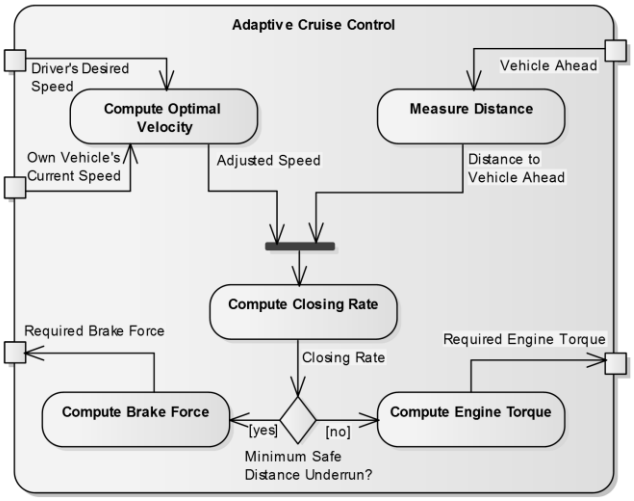

Figure 1-1 Excerpt of the Functional Requirements of the Example ACC.

The activity diagram in Figure 1-1 can be written as:

$$actD^{Adaptive\ Cruise\ Control} = \begin{pmatrix} Adaptive\ Cruise\ Control, \\ A^{Adaptive\ Cruise\ Control}, \\ E^{Adaptive\ Cruise\ Control}, \\ P^{Adaptive\ Cruise\ Control}, \\ C^{Adaptive\ Cruise\ Control} \end{pmatrix} \tag{1.9}$$

In order to increase readability of the example equations hereinafter, we introduce an abbreviated notation of the equations. In the following, the superscript indicating the name of an entity shall be left out if the references name is the name of the entity defined in the tuple. Therefore, (1.9) can be written in the following abbreviated form:

$$actD = (Adaptive\ Cruise\ Control, \mathrm{A}, \mathrm{E}, \mathrm{P}, \mathrm{C}) \tag{1.10}$$

The name of this activity diagram can be any unique ID or human-readable designation, e.g.:

$$ad = Adaptive\ Cruise\ Control \tag{1.11}$$

As can be seen, the activity diagram contains five activities, three input ports, two output ports, and two control nodes, one decision node and one fork node. In order to create Hazard Relation Diagrams automatically, it is necessary to provide a unique name to control nodes, although UML does not prescribe control nodes to have an identity. In the following, we use anonymous unique names for ports, fork nodes, join nodes, and merge nodes as well as the human-readable decision.

$$A^{Adaptive\ Cruise\ Control} = \left\{ \begin{matrix} Compute\ Optimal\ Velocity, \\ Measure\ Distance, \\ Compute\ Closing\ Rate, \\ Compute\ Brake\ Force, \\ Compute\ Engine\ Torwque \end{matrix} \right\} \tag{1.12}$$

$$P^{Adaptive\ Cruise\ Control} = \{i_1, i_2, i_3, o_1, o_2\} \tag{1.13}$$

$$D^{Adaptive\ Cruise\ Control} = \{Minimum\ Safe\ Distance\ Underrun?\} \tag{1.14}$$

$$F^{Adaptive\ Cruise\ Control} = \{f_1\} \tag{1.15}$$

$$C^{Adaptive\ Cruise\ Control} = F^{Adaptive\ Cruise\ Control} \cup D^{Adaptive\ Cruise\ Control} \tag{1.16}$$

Activity edges describe data flows and control flows between activity, ports, and control nodes. The activity diagram in Figure 1-1 contains the following eleven activity edges:

$$E^{Adaptive\ Cruise\ Control} = \tag{1.17}$$

$$\left\{\begin{array}{c} \binom{i_1, Driver's\ Desired\ Speed,}{Compute\ Optimal\ Velocity}, \binom{i_2, Own\ Vehicle's\ Current\ Speed,}{Compute\ Optimal\ Velocity}, \\ \binom{Compute\ Optimal\ Velocity,}{Adjusted\ Speed, f_1}, (i_3, Vehicle\ Ahead, Measure\ Distance), \\ \binom{Measure\ Distance,}{Dinstance\ to\ Vehicle\ Ahead, f_1}, \binom{f_1, Adjusted\ Speed +}{Distance\ to\ Vehicle\ Ahead,\ Compute\ Closing\ Rate}, \\ \binom{Compute\ Closing\ Rate, Closing\ Rate,}{Minimum\ Safe\ Distance\ Underrun?}, \binom{Minimum\ Safe\ Distance\ Underrun?,}{Closing\ Rate, Compute\ Engine\ Torque}, \\ \binom{Compute\ Engine\ Torque,}{Required\ Engine\ Torque, o_1}, \binom{Minimum\ Safe\ Distance\ Underrun?,}{Closing\ Rate, Compute\ Brake\ Force}, \\ \binom{Compute\ Brake\ Force,}{Required\ Brake\ Force, o_2} \end{array}\right\}$$

1.2. Hazard Analysis Results

The term "hazard analysis result" has thus far been used as an umbrella term which describes the output of some type of hazard analysis, documented in a hazard analysis worksheet. A "hazard analysis result" yields potential hazards which may be induced by hazard-inducing requirements, the trigger conditions for each hazard, and a possible safety goal which has been defined in response to the hazard. Hazard Relation Diagrams extend the functional requirements of the system (see Section 1.1) by the information elicited during hazard analysis. Therefore, a hazard analysis result can be described as a function on some activity diagram which yields a hazard list, i.e.

$$fha(actD) = H \tag{1.18}$$

with $actD$ being the activity diagram according to (1.1) and H being the resulting hazard list consisting of a number of hazards, $H = \{h_1, h_2, \dots, h_n\}$. A hazard can be described as a tuple:

$$h = (a, tc, sg) \tag{1.19}$$

with a unique hazard name h, which has an associated tuple consisting of an activity $a \in A^{actD}$ that gives rise to the hazard, a trigger condition tc, and the safety goal sg that has been conceived in response to the hazard. An example is shown in Table 1-1.

Table 1-1. Excerpt of the Functional Hazard Analysis.

System	Adaptive Cruise Control		Functional Hazard Analysis			
Hazard-Inducing Requirement	Hazard		Effect	Trigger Condition	Safety Goal	
	ID	Description			ID	Description
Compute Brake Force	H1	Yaw Momentum Causes Oversteering	Loss of Control, causing Crash	Driving through Curve and (High Vehicle Velocity or Low Road Surface Friction)	SG1	Prevent Loss of Control in Curves
					SG3	Adjust Brake Force In-dependently for All Four Wheels

In Table 1-1, an excerpt o the functional hazard analysis is shown, featuring only hazard H1 and both of its associated safety goals and it's trigger conditions. In this example, n refers to the description of the hazard in Table 1-1, i.e. *Yaw Momentum Causes Oversteering*. Since there are two safety goals associated with hazard H1, sg could refer the description of either, i.e. *Prevent Loss of Control in Curves* (SG1 in Table 1-1) or *Adjust Brake Force Independently for All Four Wheels* (SG2 in Table 1-1).

The trigger conditions pertaining to a hazard are described as a binary tree, in which, like in the example in, nodes AND- or OR-nodes associate exactly two binary trigger condition subtrees:

$$tc = \left(tc_{left}, r, tc_{right}\right) : r \in \{c, conj, disj\} \\ \vee\ (r = c) \longrightarrow tc_{left}, tc_{right} = \emptyset \tag{1.20}$$

In (1.20), tc_{left}, tc_{right} represent the left and right binary trigger condition subtree, respectively, which are associated through the root r. The root may be a conjunction (i.e. AND-node) $conj$, indicating that both the left and right subtree must evaluate to true for the entire tree tc to evaluate to true. Alternatively, the root may be a disjunction (i.e. OR-node) $disj$, in which either or both subtrees must evaluate to true for the entire tree tc to evaluate to true. If the root is neither a conjunction nor a disjunction, the root may be is a human-definable condition in the operational context of the system c, in which case it is implied that there are not further subtrees (i.e. $tc_{left}, tc_{right} = \emptyset$). The trigger condition from Table 1-1 is shown in Figure 1-2.

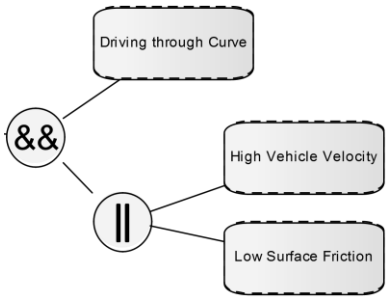

Figure 1-2 Trigger Condition of Hazard H1 from Table 1-1 as a Binary Tree.

In this example, the conditions *Driving through Curve*, *High Vehicle Velocity*, and *Low Road Surface Friction* have been defined. There is one disjunction associating *High Vehicle Velocity* with *Low Road Surface Friction*, thereby forming a subtree. There is also one disjunction, which associates the subtree with the condition *Driving through Curve*. This binary trigger condition tree can be formalized as follows:

$$tc^{h_1} = \left(conj, \begin{pmatrix} High\ Vehicle\ Velocity, disj, \\ Low\ Road\ Surface\ Friction \end{pmatrix}\right) \tag{1.21}$$

The FHA excerpt in Table 1-1 can hence be formalized as follows:

$$fha(Adaptive\ Cruise\ Control) = \{h_1\} \tag{1.22}$$

$$h_1^{Yaw\ Momentum\ Causes\ Oversteer} = \left(\begin{array}{c} Compute\ Brake\ Force, \\ \left(\begin{pmatrix} Driving\ through\ Curve, conj, \\ High\ Vehicle\ Velocity, disj, \\ Low\ Road\ Surface\ Friction \end{pmatrix}, \right. \\ \left. Prevent\ Loss\ of\ Control\ in\ Curve \right) \end{array} \right) \tag{1.23}$$

1.3. Partial Mitigations

Partial mitigations subsume the hazard-mitigating requirements that are meant to mitigate a hazard. We have there introduced the modeling concept “mitigation partition” as the notational element to highlight hazard-mitigating requirements within a Hazard Relation Diagram. We introduced the concept “mitigation template” to document the ontological dependencies between hazards, trigger conditions, safety goals, and hazard-mitigating requirements. Mitigation templates are used to document transformation steps to be enacted on activity diagrams containing hazard-inducing requirements in order to create activity diagrams containing hazard-mitigating requirements with are enriched with hazard analysis results in order to generate Hazard Relation Diagrams. A mitigation template hence represents a mitigation partition in the generated Hazard Relation Diagram. Table 1-2 shows a mitigation template that serves this purpose. In the example in Figure 1-3, the principle mechanism to specify hazard-mitigating requirements by means of the mitigation template from Table 1-2 is outlined.

Table 1-2. Mitigation Template to Specify Changes to Activity Diagrams containing Hazard-Inducing Requirements in order to automatically generate Hazard-Mitigating Requirements.

Mitigation Name	Conceptual Mitigation name								
Reference	Source Activity Diagram name								
Hazard	Hazard name								
Insert Activity	**ID**	**Activity Name**							
Insert Activity Edge	**ID**	**Source**		**Message**		**Guard**		**Target**	
Insert Pin	**ID**	**Pin Name**							
Insert Control Node	**ID**	**Node Type**		**Node Name**					
Remove Activity	**ID**	**Activity Name**							
Remove Activity Edge	**ID**	**Source**		**Message**		**Guard**		**Target**	
Remove Pin	**ID**	**Pin Name**							
Remove Control Node	**ID**	**Node Name**							
Substitute Activity	**ID**	**Old Activity Name**				**New Activity Name**			
Substitute Activity Edge	**ID**	**Old Source**	**Old Message**	**Old Guard**	**Old Target**	**New Source**	**New Message**	**New Guard**	**New Target**
Substitute Pin	**ID**	**Old Pin Name**				**New Pin Name**			
Substitute Control Node	**ID**	**Old Node Name**				**New Node Type**		**New Node Name**	

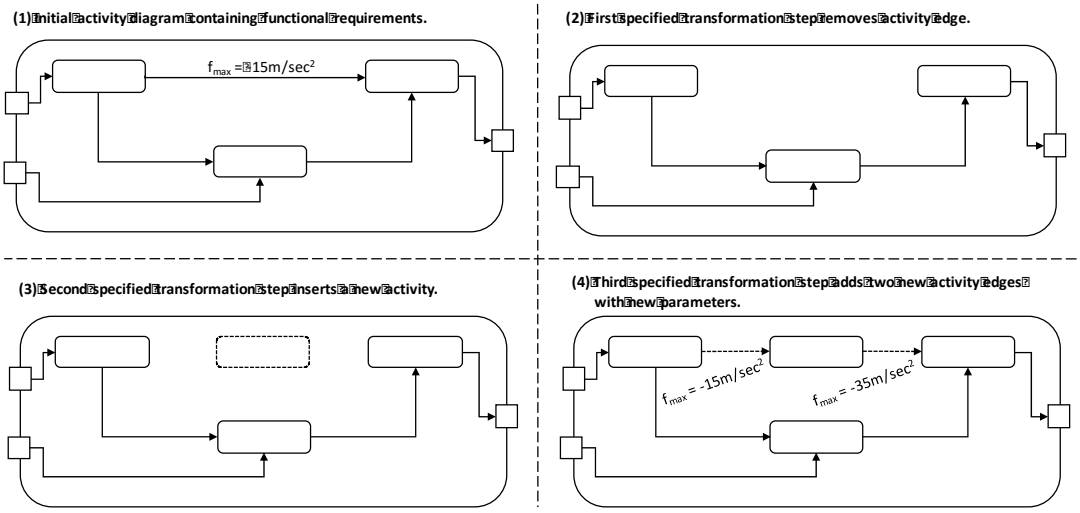

Figure 1-3 Simplified Example of Transformation Steps to Create Hazard-Mitigating Requirements.

In this example, an activity edge bearing a specific message (panel (1) in Figure 1-3) is replaced by a new activity and corresponding incoming and outgoing activity edges. As can be seen in panel (2), the stakeholder first specifies a transformation step to remove the old activity edge. Next, as can be seen in panel (3), the stakeholder specified an insertion opration to add a new activity. Finally, in panel (4), the stakeholder specifies two new activity edges with new messages to add a "context" (in the sense of the call order indicated by incoming and outgoing activity edges) to the new activity. This example also illustrates the case, where a hazard-mitigating requirement

corresponds to the change of a parameter for an activity. Parameter nodes for UML activities and opaque actions are represented in a simplified manner using the message payload on activity edges (see Section 1.1). This means that in a case, where a hazard-mitigating requirement consists of a changed parameter for an activity (e.g., the substitution of a value, tolerance, worst case execution time, etc.) can be represented using mitigation templates by substituting activity edges with modified messages.

By specifying hazard-mitigating requirements by means of transformation steps, partial mitigations can be specified as shown in Table 1-3. The example in Table 1-3 shows the partial mitigation for the hazard-mitigating requirements for Hazard H1. In the following, the canonical formalization of mitigation templates is outlined.

Table 1-3. Example of a Specified Mitigation Template to Mitigate Hazard H1 from according to Mitigation 1.

Mitigation	Mitigation for Hazard H1				
Reference	ACC Functional Requirements				
Corresponding Hazard	Yaw Momentum Causes Oversteering				
Insert Activity Edge	**ID**	**Source**	**Message**	**Guard**	**Target**
	4	i4	Yaw Rate	true	Compute Brake Force
	5	Compute Brake Force	Limited Brake Force	true	$\max brake\ force * 0.85^{\frac{1}{yaw}}$
	6	$\max brake\ force * 0.85^{\frac{5}{yaw}}$	Limited Brake Force	yes	o2
	7	$\max brake\ force * 0.85^{\frac{5}{yaw}}$	Limited Brake Force	no	Compute Brake Force
Insert Pin	**ID**	**Pin Name**			
	3	i4			
Insert Control Node	**ID**	**Node Type**	**Node name**		
	2	Decision	$\max brake\ force * 0.85^{\frac{5}{yaw}}$		
Remove Activity Edge	**ID**	**Source**	**Message**	**Guard**	**Target**
	1	Compute Brake Force	Required Brake Force	true	o2

We outlined that in practice, it might be the case that a hazard is mitigated by introducing hazard-mitigating requirements to different locations within the same activity diagram, across multiple activity diagrams, or both. This means that multiple mitigation templates may be needed in order to visualize the hazard's conceptual mitigation in the generated Hazard Relation Diagram, i.e. one mitigation template for each mitigation partition. It follows, that the modeling concept "conceptual mitigation" is a collective set of partial mitigations, each of which represents a set of removed, inserted, or substituted activity diagram modeling elements:

$$CM = \{pm_1, pm_2, \dots, pm_n\} \tag{1.24}$$

Each partial mitigation pm is a tuple consisting of a human-readable name mn identifying the common conceptual mitigation. This means that all partial mitigations with the same name make

up the conceptual mitigation. In addition, the partial mitigation pm consists of a source activity diagram $actD$ according to (1.1) upon which the partial mitigation shall be committed, and a hazard h according to (1.19) that shall be mitigated. A partial mitigation furthermore consists of a set of elements removed, inserted, or substituted, respectively. For some partial mitigation pm, it follows:

$$pm = (mn, actD, h, R, I, S) \tag{1.25}$$

Any type of element el of the referenced activity diagram can be inserted, removed, or substituted (i.e. $el \in A^{actD} \cup E^{actD} \cup P^{actD} \cup C^{actD}$). To simplify the following formalizations, substitutions are expressed as a remove operation followed by an insertion operation. Hence, elements to be removed, inserted, or substituted can be written as

$$R = \{el_1, el_2, \dots, el_n\} \tag{1.26}$$

$$I = \{el_1, el_2, \dots, el_n\} \tag{1.27}$$

$$S = \left\{\left(el_1^{old}, el_1^{new}\right), \left(el_2^{old}, el_2^{new}\right), \dots, (el_n^{old}, el_n^{new})\right\} \tag{1.28}$$

It must also be noted, that insertion, removal, and substitution operations may result in syntactically invalid activity diagrams. For example, if an activity is removed, any activity edge that had the removed activity as a source or target will be "dangling" in the sense of Well-formedness Rule 9. For this reason, we have defined several well-formedness rules pertaining to the syntactic correctness of the activity diagrams underlying Hazard Relation Diagrams. However, these well-formedness rules only pertain to syntactic correctness as it is relevant for the generation of Hazard Relation Diagram and do not cover any case of syntactic incorrectness of activity diagrams. It therefore remains the responsibility of the developer to ensure that that hazard-mitigating requirements are specified in a syntactically correct manner.

In case there are two or more partial mitigation with the same name mn, in which the same hazard h and the same activity diagram $actD$ are referenced, but where the elements to be inserted, removed, or substituted differ, the following holds:

$$\forall\left(pm_i, pm_j\right) | pm_i = (mn_i, h_i, actD_i, R_i, I_i, S_i) \wedge \tag{1.29}$$

$$pm_j = \left(mn_j, h_j, actD_j, R_j, I_j, S_j\right) \wedge mn_i = mn_j \wedge$$

$$h_i = h_j \wedge actD_i = actD_j \wedge (R_i \cup I_i \cup S_i) \cap \left(R_j \cup I_j \cup S_j\right) = \emptyset$$

In this case, all partial mitigations belong to the same conceptual mitigation and must be enacted on the activity diagram $actD$. This corresponds to Case 2 from [REJ Paper]. If there are two or more partial mitigations with the same name mn, in which different hazards are referenced, regardless of the activity diagrams referenced therein, this represents an example of Case 4 from [REJ Paper]. This means the same conceptual mitigation is a candidate to mitigate multiple, independent hazards. In this case, the changes from all partial mitigations belong to the same conceptual mitigation, but must be enacted on each respective activity diagram for the conceptual mitigation to be completely enacted. This corresponds to Case 3 from [REJ Paper].

If there are two or more partial mitigations with the same name mn, in which different hazards are referenced, regardless of the activity diagrams referenced therein, the following must hold:

$$\forall\left(pm_i, pm_j\right) | pm_i = (mn_i, h_i, actD_i, R_i, I_i, S_i) \wedge$$
$$pm_j = \left(mn_j, h_j, actD_j, R_j, I_j, S_j\right) \wedge mn_i = mn_j \wedge h_i \neq h_j \quad (1.30)$$

This case corresponds to Case 4 from [REJ Paper] and is indicative of the same conceptual mitigation being a candidate to resolve multiple, independent hazards.

In case two or more partial mitigations have different names, they pertain to two or more conceptual mitigations (see above). If in all of these partial mitigation, the same hazard h is referenced, regardless of the activity diagrams referenced therein, the partial mitigations represent an example of Case 5 from [REJ Paper], where alternative conceptual mitigations to resolve the same hazard, for which the following holds:

$$\forall\left(pm_i, pm_j\right) | pm_i = (mn_i, h_i, actD_i, R_i, I_i, S_i) \wedge$$
$$pm_j = \left(mn_j, h_j, actD_j, R_j, I_j, S_j\right) \wedge mn_i \neq mn_j \wedge h_i = h_j \quad (1.31)$$

Partial mitigation can be documented using mitigation templates. Mitigation templates document the specific changes that are enacted on the source activity diagram containing the hazard-inducing requirements in order to generate a new activity diagram containing the hazard-mitigating requirements. shows a mitigation template that serves this purpose.

As can be seen, the partial mitigation in in Table 1-3 bears the human-readable name "Mitigation for Hazard H1". Furthermore, the partial mitigation references the activity diagram name given in (1.11) and the hazard from (1.23). Therefore:

$$mn = Mitigation\ for\ Hazard\ H1 \tag{1.32}$$

$$actD = Adaptive\ Cruise\ Control \tag{1.33}$$

$$h = Yaw\ Momentum\ Causes\ Oversteering \tag{1.34}$$

No element is substituted in Table 1-3. One activity edge is removed, namely the data flow from the activity "Compute Brake Force" to the output pin o_2:

$$e_1^{remove} = \begin{pmatrix} Compute\ Brake\ Force, \\ Required\ Brake\ Force, true, o_2 \end{pmatrix} \tag{1.35}$$

In consequence, the set of activity edges to be removed is as follows:

$$R = \{e_1^{remove}\} \tag{1.36}$$

In addition, several elements are inserted. Specifically, one additional input pin called i_4 is added along with an activity edge that transmits the yaw rate to the activity "Compute Brake Force". Furthermore, a decision node is inserted along with three data flows as follows:

$$e_1^{insert} = i_4 \tag{1.37}$$

$$e_2^{insert} = (i_4, Yaw\ Rate, Compute\ Brake\ Force) \tag{1.38}$$

$$e_3^{insert} = \left(\max brake\ force * 0.85^{\frac{5}{yaw}}\right) \tag{1.39}$$

$$e_4^{insert} = \begin{pmatrix} Compute\ Brake\ Force, Limited\ Brake\ Force, \\ \max brake\ force * 0.85^{\frac{5}{yaw}} \end{pmatrix} \tag{1.40}$$

$$e_5^{insert} = \left(\max brake\ force * 0.85^{\frac{1}{yaw}}, Limited\ Brake\ Force, \\ no, Compute\ Brake\ Force\right) \tag{1.41}$$

$$e_6^{insert} = \begin{pmatrix} \max brake\ force * 0.85^{\frac{5}{yaw}}, \\ Limited\ Brake\ Force, yes, o_2 \end{pmatrix} \tag{1.42}$$

$$I = \{e_1^{insert}, e_2^{insert}, e_3^{insert}, e_4^{insert}, e_5^{insert}, e_6^{insert}\} \tag{1.43}$$

The partial mitigation pm_i in Table 1-3 can hence be written as

$$pm_1 = \begin{pmatrix} Mitigation\ for\ Hazard\ H1, \\ Adaptive\ Cruise\ Control, \\ h_1, R, I, S \end{pmatrix} \tag{1.44}$$

with $S^{Mitigation\ for\ Hazard\ H1} = \emptyset$ because Table 1-3 does not specify any elements to be substituted. Since there is only one mitigation partition, the partial mitigation from Table 1-3 is the only in order to mitigate hazard H1. Therefore, the conceptual mitigation for this hazard is the set containing $pm_1^{Mitigation\ for\ Hazard\ H1}$:

$$CM^{Mitigation\ for\ Hazard\ H1} = \{pm_1\} \tag{1.45}$$

Each partial mitigation belonging to the same conceptual mitigation must be contained in a Hazard Relation Diagram in the form of one mitigation partition. It is to note however, that mitigation partitions do not depict the human-readable name, as this is merely used to group those partial mitigations that belong to the same conceptual mitigation.

1.4. Hazard Relation Diagram

A Hazard Relation Diagram $hazRD$ is described as the following tuple:

$$hazRD = (hrd, AD, h, tc^h, sg^h, CM, hr, HA) \tag{1.46}$$

where hrd is a unique designation of the Hazard Relation Diagram. In a Hazard Relation Diagram, a set of activity diagrams $AD^{hrd} = \{ad_1^{hrd}, ad_2^{hrd}, \dots, ad_n^{hrd}\}$ according to (1.1) is associated with exactly one hazard h^{hrd} from (1.19), its trigger condition tc^h in its Boolean representation according to (1.20), and the safety goal sg^h. Figure 1-4 shows a Hazard Relation Diagram overlaid with the formal elements from (1.46).

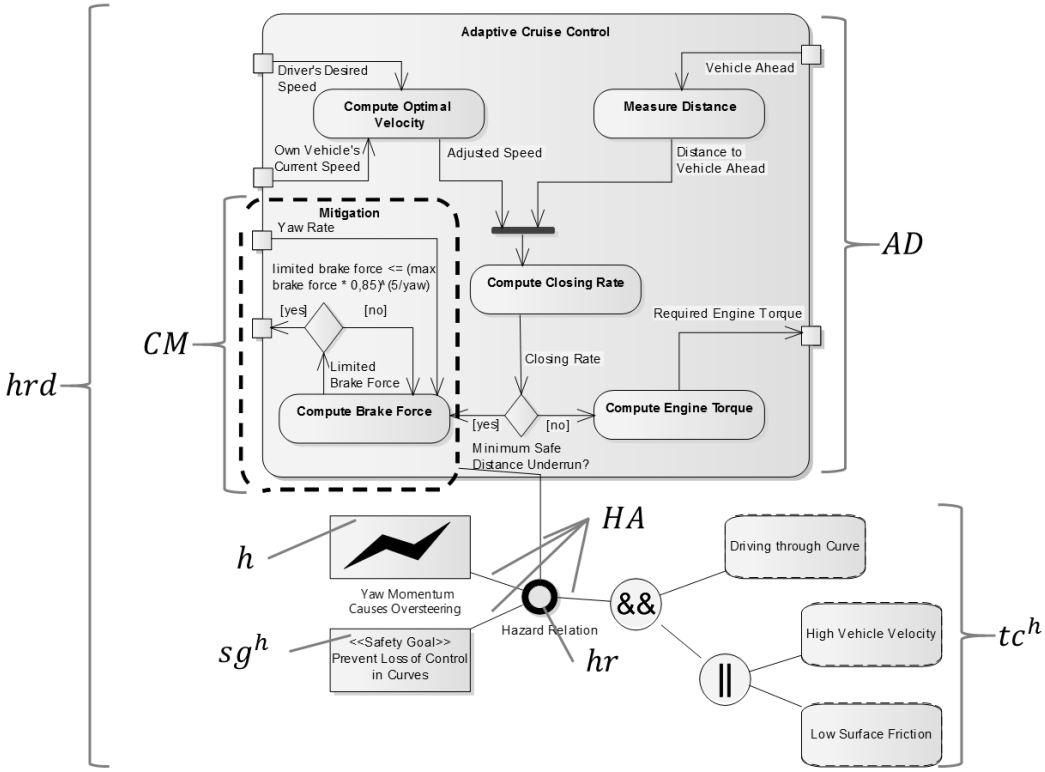

Figure 1-4 Hazard Relation Diagram overlaid with the formal elements from (1.46).

Since the well-formedness rules prescribe that the trigger condition and the safety goal pertain to the hazard, we define[1] the following relationship:

$$tc^h \in_t h \tag{1.47}$$

$$sg^h \in_t h \tag{1.48}$$

In addition to trigger conditions, safety goals, and the hazard they pertain to, Hazard Relation Diagrams comprise a conceptual mitigation CM^{hrd} according to (1.45) which in turn consists of a set of partial mitigations according to (1.25). For each partial mitigation $pm \in CM^{hrd}$, the activity diagram referenced in the partial mitigation must correspond to one of the activity diagrams in AD^{hrd}:

$$\forall pm \in CM^{hrd} | \exists ad \in AD^{hrd} : ad \in_t pm \tag{1.49}$$

---

[1] Here and in the following, we use the operator $x \in_t U$ to denote that some element x is element of some tuple U, regardless of its index within U. Since the tuples in this research are unambiguous due to their membership order and type-specificity, the following holds as a simplification of [**Error! Reference source not found.**]: $\in_t \coloneqq \exists x | U = (u_1, u_2, \dots, u_n) \wedge x = u_t, 0 < t < n$

As outlined in Section 1.3, each partial mitigation is depicted in the Hazard Relation Diagram using a mitigation partition. However, in contrast to the partial mitigation, the mitigation partition in the Hazard Relation Diagram will not include items that have been removed from the source activity diagram containing the hazard-inducing requirements. This is because a mitigation partition signifying removed elements would be empty, which may introduce ambiguity as to what information is highlighted in the mitigation partition. Yet, validation will not take into account individual changes, but instead will consider whether the collective impact of all enacted changes on the requirements are adequate to mitigate a hazard (see Section 1.2).

Hazard Relation Diagram consists furthermore of a Hazard Relation hr^{hrd} and a set of hazard associations $HA = \{ha_1, ha_2, \dots, ha_n\}$. Hazard associations can be seen as tuples connecting the Hazard Relation with either the hazard, the trigger condition, the safety goal, or the mitigation partitions:

$$ha^{hrd} = (srt, end) | srt = hr^{hrd} \vee \begin{pmatrix} end = h^{hrd} \\ \wedge\, end = tc^h \wedge end = sg^h \wedge \\ end = pm^h : \forall pm \in CM^{hrd} \end{pmatrix} \tag{1.50}$$

We have shown a Hazard Relation Diagram which contains a conceptual mitigation for Hazard H1. The hazard-mitigating requirements upon which this Hazard Relation Diagram is based are generated by applying the changes from the partial mitigation in Table 1-3 on the activity diagram. According to (1.46), the Hazard Relation Diagram above can hence be written as:

$$hazRD = \begin{pmatrix} HRD\ for\ Hazard\ H1\ and\ Mitigation\ M1, \\ AD, h, tc^h, sg^h, CM, hr, HA \end{pmatrix} \tag{1.51}$$

As the name for this Hazard Relation Diagram, the following human-readable designation was assigned:

$$hrd = HRD\ for\ Hazard\ H1\ and\ Mitigation\ M1 \tag{1.52}$$

The hazard depicted above is hazard H1 in Table 1-1, which has the safety goal “Prevent Loss of Control in Curve”, and the trigger conditions from (1.21). The hazard in this Hazard Relation Diagram is therefore the same as the one in as shown in (1.23) and the trigger conditions are the same as in (1.21),

$$h^{HRD\ for\ Hazard\ H1\ and\ Mitigation\ M1} = h_1 \tag{1.53}$$

$$tc^h = tc^{h_1} \tag{1.54}$$

while the safety goal is:

$$sg^h = Prevent\ Loss\ of\ Control\ in\ Curve \tag{1.55}$$

According to (1.45), the conceptual mitigation in the Hazard Relation Diagram is a set consisting of exactly one partial mitigation:

$$CM = \{pm^{Mitigation\ for\ Hazard\ H1}\} \tag{1.56}$$

Therefore, the set of activity diagrams is also a set, specifically the activity diagram containing hazard-inducing requirements (upon which the partial mitigation is applied):

$$AD = \{actD^{Adaptive\ Cruise\ Control}\} \tag{1.57}$$

Since there is only one mitigation partition in the Hazard Relation Diagram, this means that there are exactly four hazard associations, which connect the Hazard Relation, hr, to the hazard, the safety goal, the trigger conditions, and the mitigation partition, respectively:

$$ha_1 = (hr\ , \mathrm{h}) \tag{1.58}$$

$$ha_2 = (hr, sg^h) \tag{1.59}$$

$$ha_3 = (hr, tc^h) \tag{1.60}$$

$$ha_4 = (hr, pm^{Mitigation\ for\ Hazard\ H1}) \tag{1.61}$$

$$HA = \{ha_1, ha_2, ha_3, ha_4,\} \tag{1.62}$$