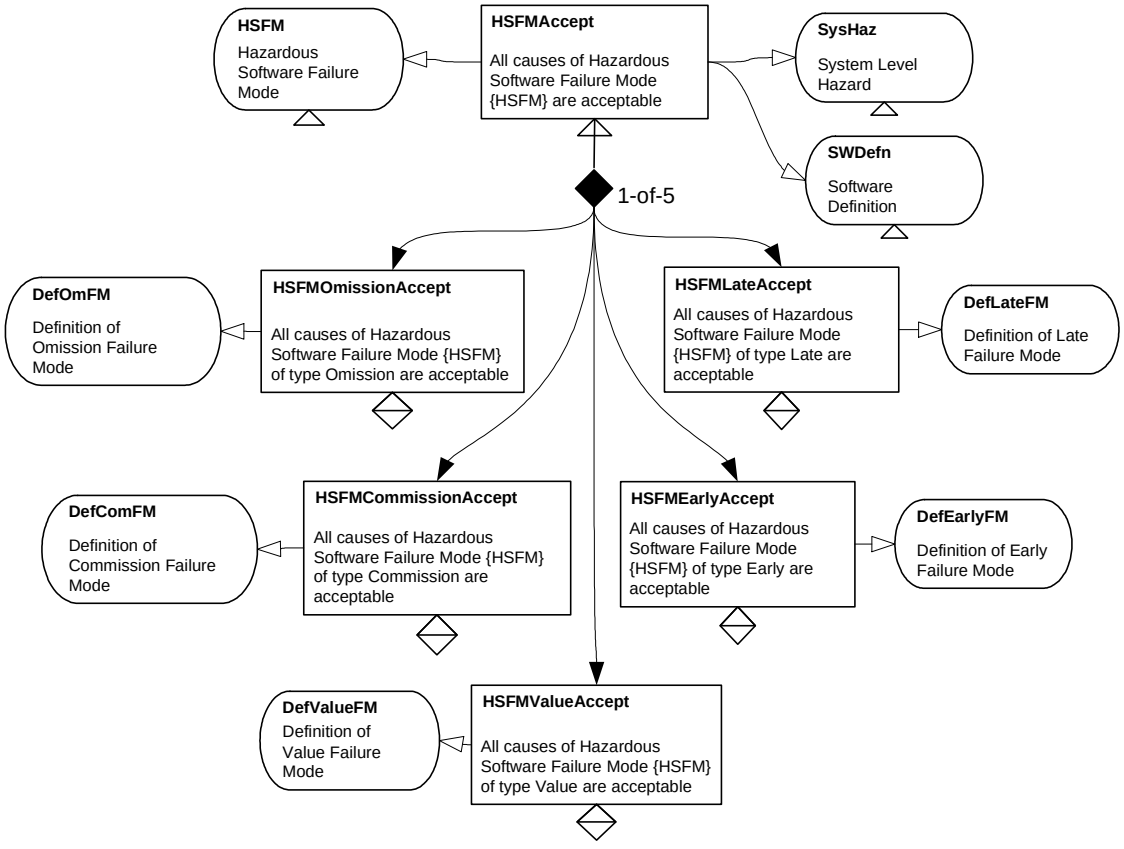
Hazardous Software Failure Mode Classification

Author(s)	Rob Weaver, John McDermid, Tim Kelly		
Created	18/09/00	**Last Modified**	20/04/04

Intent	The intent of this pattern is to provide a type classification for the hazardous failure mode that is the subject of the argument. The failure mode can be classified as one of Omission, Commission, Early, Late or Value.
Also Known As	
Motivation	By defining the hazard as a particular type, it is possible to focus the argument on the particular causes and associated evidence for that hazard type.

Structure



Participants	**HSFMAccept**	The overall objective of the argument – to provide sufficient support to the claim that the Hazardous Software Failure Mode under consideration is acceptably safe.

	HSFM	This context identifies the Hazardous Software Failure Mode, for which this pattern provides the classification.
	SysHaz	This context identifies the System Level Hazard to which the software failure mode contributes.
	SWDefn	This Software Definition should give a clear description of the system software. From the model it should be possible to determine how the failure mode contributes to the system level hazard and what type it is.
	HSFMOmissionAccept	This claim is a subtype of the **HSFMAccept** claim. It alters the original claim by classifying the type as a Omission Failure Mode.
	DefOmFM	To classify the failure mode as an Omission it is necessary to define this type. The failure mode type can be defined as: *The service is never delivered*
	HSFMCommission Accept	This claim is a subtype of the **HSFMAccept** claim. It alters the original claim by classifying the type as a Commission Failure Mode.
	DefComFM	To classify the failure mode as a Commission it is necessary to define this type. The failure mode type can be defined as: *The service is delivered when not required*
	HSFMEarlyAccept	This claim is a subtype of the **HSFMAccept** claim. It alters the original claim by classifying the type as an Early Failure Mode.
	DefEarlyFM	To classify the failure mode as Early it is necessary to define this type. The failure mode type can be defined as: *The service occurs earlier than intended*
	HSFMLateAccept	This claim is a subtype of the **HSFMAccept** claim. It alters the original claim by classifying the type as a Late Failure Mode.

	DefLateFM	To classify the failure mode as Late it is necessary to define this type. The failure mode type can be defined as: *The service occurs later than intended*
	HSFMValueAccept	This claim is a subtype of the **HSFMAccept** claim. It alters the original claim by classifying the type as a Value Failure Mode.
	DefValueFM	To classify the failure mode as Value it is necessary to define this type. The failure mode type can be defined as: *The information (data) delivered has the wrong value*
Collaborations	• The type of the Hazardous Software Failure Mode identified by **HSFMOmissionAccept**, **HSFMCommissionAccept**, **HSFMEarlyAccept**, **HSFMLateAccept** or **HSFMValueAccept**, is dependent upon the failure mode identified in **HSFM**, the System hazard it contributes to (**SysHaz**), the software definition (**SWDefn**) and the type definitions identified in **DefOmFM**, **DefComFM**, **DefEarlyFM**, **DefLateFM** and **DefValueFM**.	
Applicability	This pattern provides a classification of a hazardous software failure mode and it assumes that the failure mode has already been identified. In order to perform the classification, it assumes an understanding of the role of the failure mode in contributing to system level hazards.	
Consequences	After instantiating this pattern one of the following undeveloped goals will remain: • **HSFMOmissionAccept**, **HSFMCommissionAccept**, **HSFMEarlyAccept**, **HSFMLateAccept** or **HSFMValueAccept** This goal must be developed to satisfy the **HSFMAccept** claim.	
Implementation	To instantiate this pattern the implementer should consider the failure mode in the context of the software definition and the system level hazard to which it contributes. From this the type should be determined.	

	Possible Pitfalls • Not clearly understanding the role of the software failure mode in contributing to the system level hazard • Not understanding the different failure mode types.
Examples	None provided at this stage.
Known Uses	
Related Patterns	*Hazardous Software Failure Mode Decomposition* –This pattern provides a context for the overall objective **HSFMAccept**. *Effects of Other Components* - This pattern provides a context for the overall objective **HSFMAccept**. *Software Argument Approach* – This pattern can be used to decompose the undeveloped goals **HSFMOmissionAccept**, **HSFMCommissionAccept**, **HSFMEarlyAccept**, **HSFMLateAccept** and **HSFMValueAccept**. This pattern forms part of a software safety argument pattern catalogue, which includes the following patterns: *Component Contributions to System Hazards* *Hazardous Software Failure Mode Decomposition* *Hazardous Software Failure Mode Classification* *Software Safety Argument Approach* *Absence of Omission Hazardous Failure Mode* *Absence of Commission Hazardous Failure Mode* *Absence of Early Hazardous Failure Mode* *Absence of Late Hazardous Failure Mode* *Absence of Value Hazardous Failure Mode* *Effects of Other Components* *Handling of Software Failure Mode* *Handling of Hardware/Other Component Failure Mode*