3.4. Hazardous Contribution Software Safety Argument Pattern

Hazardous Contribution Software Safety Argument Pattern			
Author	Richard Hawkins		
Created	09/12/08	**Last modified**	08/06/09

INTENT

This pattern provides the structure for arguments that potential hazardous failures that may arise at {tier n} are acceptably managed.

MOTIVATION

At each tier of software development it is possible that hazardous failures may manifest themselves. This argument demonstrates how the hazardous failures are prevented. This is achieved in two ways. Firstly potential hazardous failure modes are identified, and appropriate SSRs defined in response. Secondly, the absence of design errors which could cause hazardous failures must also be demonstrated. It should be noted that this aspect of the argument will often consider more generally how errors are removed from the design.

STRUCTURE

The structure of this argument pattern is shown in Figure 6 below.

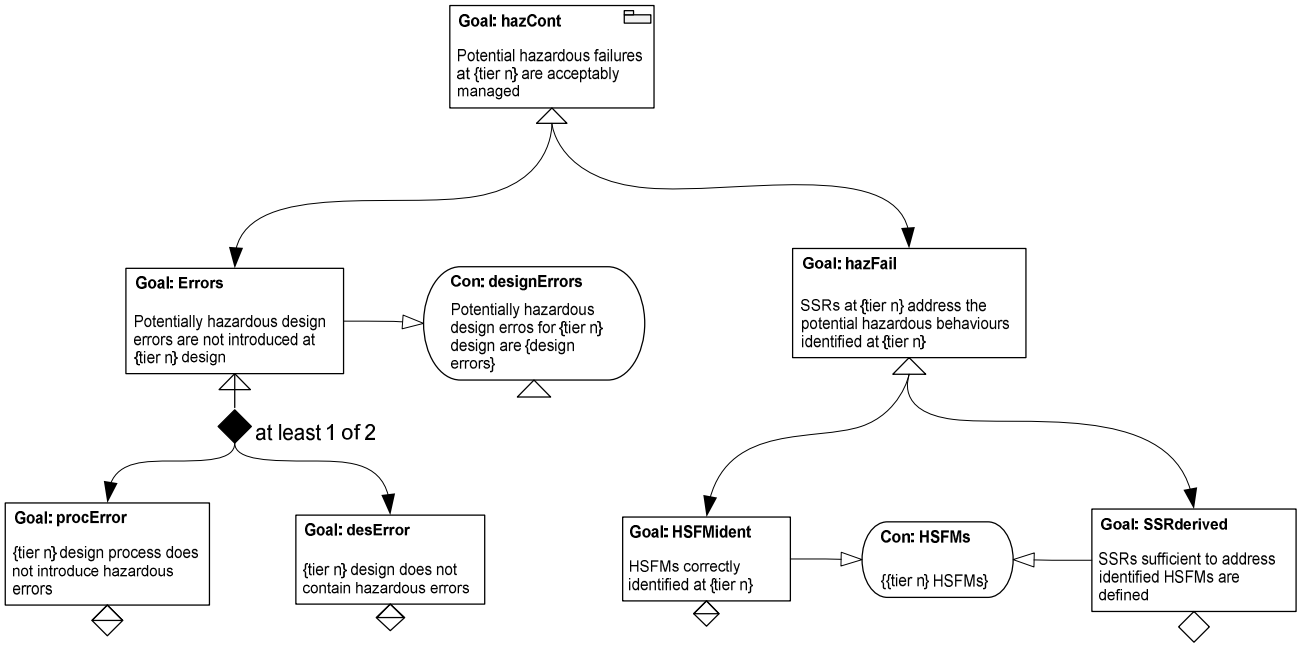

Figure 6 - Hazardous Contribution Software Safety Argument Pattern Structure

PARTICIPANTS

Goal: hazCont

This is a public goal in a separate argument module which can be referenced from other software safety argument modules using an away goal reference. This claim is applicable wherever an argument is being presented over the tiers of the software development lifecycle. {tier n} refers to the current tier being considered in the argument. This goal claims that the potential hazardous failures at the current tier are acceptably managed.

Goal: Errors

The design process at any tier may be flawed. This goal claims that potentially hazardous design (or code) errors have not been introduced at the current tier. This supported by arguing about the design process adopted at the current tier, and about the design artefact itself.

Goal: desError

This goal claims that the design (or code) produced at the current tier does not contain potentially hazardous errors.

Goal: procError

This goal must be supported by argument and evidence about the integrity of the design process that is used at the current tier. Note that at the lowest level tiers this may include the coding process.

Goal: hazFail

This goal claims that SSRs are identified, sufficient to address the potential hazardous behaviours identified at {tier n}. The goal is supported by demonstrating that hazardous software failure modes (HSFMs) (that is failure of the software which could contribute to a hazard at the system level) at {tier n} are sufficiently identified, and that each of these HSFMs is addressed through the definition of one or more SSRs.

APPLICABILITY

This pattern should be applied as part of any hazard-directed software safety argument to provide a warrant for an argument that SSRs from one development tier are adequately addressed at the next tier.

CONSEQUENCES

Once this pattern has been instantiated, a number of elements will remain undeveloped and requiring support. 'Goal: deviations' must be supported by an argument provided in a 'deviations' safety argument module. An instance of 'Goal: HSFMaddress' must be supported for each HSFM identified at {tier n}. 'Goal: HSFMs' must also be supported.

IMPLEMENTATION

The techniques most appropriate to use to identify potential deviations from intended behaviour at each tier will vary. Appendix B provides some examples of the types of hazard and failure analysis techniques that may be used at some of the possible tiers.

RELATED PATTERNS

This pattern is used to provide context to the Software contribution safety argument pattern.