3.3. SSR Identification Software Safety Argument Pattern

SSR Identification 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 software safety requirements (SSRs) from a previous tier of development have been adequately captured at the next tier of development through the allocation, decomposition, apportionment or interpretation of the SSRs from the previous tier. This is achieved either through making design decisions which mitigate the SSR, or through the definition of additional SSRs.

STRUCTURE

The structure of this argument pattern is shown in Figure 5 below.

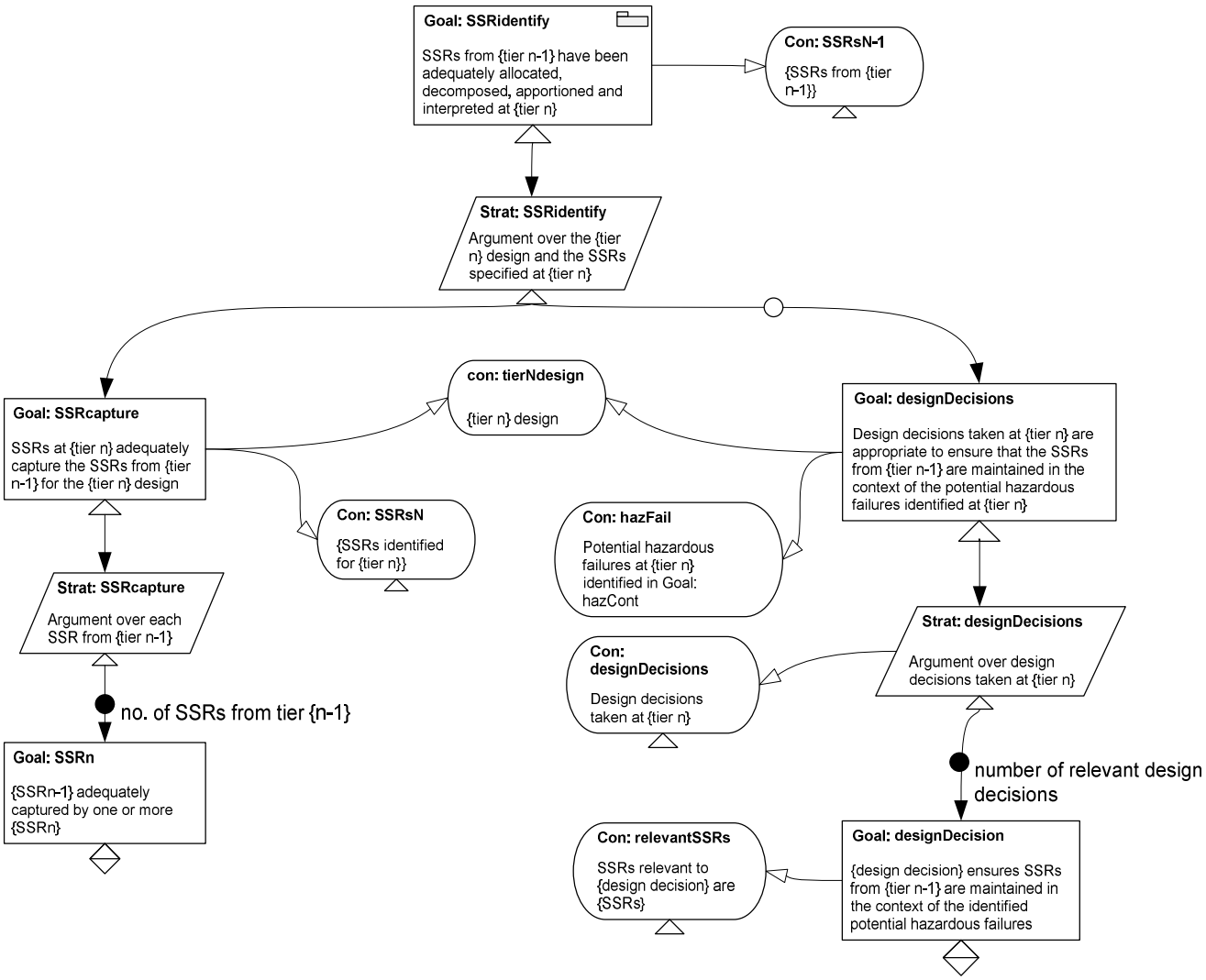

Figure 5 – SSR Identification Software Safety Argument Pattern Structure

PARTICIPANTS

Goal: SSRidentify

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. {tier n-1} refers to the previous tier of development. At each tier it is necessary to demonstrate that the SSRs from {tier n-1} are adequately captured in the design of {tier n}.

Strat: SSRidentify

This is achieved either through making design decisions at {tier n} which facilitate the satisfaction of the {tier n-1} SSR, or through the definition of SSRs for {tier n} which consider the {tier n} design. In some cases a mixture of appropriate design decision and SSR definition might be required to capture all of the {tier n} SSRs. In other cases just one approach may be sufficient, this will depend on a number of factors including the nature of the SSRs, which tier is being considered and the nature of the design of {tier n}. The Argument justification software safety argument pattern may be used to justify the adopted strategy.

Goal: SSRcapture

This goal claims that the design of {tier n} has been considered in order to define SSRs for {tier n} which adequately capture the SSRs from {tier n-1}.

Con: tierNdesign

The design of {tier n} will be determined by the design decisions made, some of which may have been influenced by {tier n-1} SSRs. The {tier n} design will also determine the nature of the SSRs defined at {tier n}. This context is therefore common to both 'Goal: SSRcapture' and 'Goal: designDecisions'.

Goal: SSRn

An instance of this goal is created for each SSR from {tier n-1}. To adequately reflect each {tier n-1} SSR, one or more SSRs may be required at {tier n}.

Goal: designDecisions

It may be possible to facilitate the satisfaction of some of the {tier n-1} SSRs through decisions taken in the design of {tier n}. For example, a decision to have redundant components may be taken in order to help satisfy a SSR relating to the availability of an item of data. Alternatively a decision may be taken to introduce into the design a mechanism for detecting and handling failures which may lead to the breach of an SSR. It may also be possible, for example, to prevent interference between components through ensuring physical or logical partitioning in the design. This goal allows claims to be made that such decisions reflect the SSRs from {tier n-1}. The appropriate of the design decisions will depend upon the nature of the SSRs.

Goal: designDecision

An instance of this goal is created for each design decision taken which is relevant to the satisfaction of a SSR from {tier n-1}. Each instance of this goal requires a supporting argument which demonstrates how the design feature supports the SSR satisfaction.

Con: relevantSSRs

This context specifies the SSRs which this design decision helps to satisfy.

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. An instance of 'Goal: SSRn' must be supported for each SSR from {tier n-1}. An argument should be provided which demonstrates that one or more SSRs specified at {tier n} adequately capture the {tier n-1} SSR for the design at {tier n}. An instance of 'Goal: designDecision' must be supported for each design decision which was made to facilitate the satisfaction of SSRs at {tier n}. 'Goal: HSFMdetect' and 'Goal: SSRprevent', if created, must also be supported.

IMPLEMENTATION