15.2 Advanced Arguments

Advanced Arguments adds in additional features to support

- Strengths - to allow the inclusion of strength of belief in an argument
- Additional AssertionDeclarations - an additional AssertionDeclarationKind EnumerationLiteral ("byRule") and an assertionDeclartionRule AssociationEnd to allow the extension of AssertionDeclarations to include user-defined specifications
- Additional InferenceTypes - a new Enumeration InferenceTypeKind that adds in support for inductive (default), deductive, abductive and user-defined specifications (through "byRule" EnumerationLiteral and inferenceRule AssociationEnd) on AssertedInferenceRelation
- Explicit Truth values (through isTrue Attribute on Assertion), or
- (possible Multi-valued) assertionValues and varied logic (e.g. Three-valued logic) using assertionValue AssociationEnd on Assertion

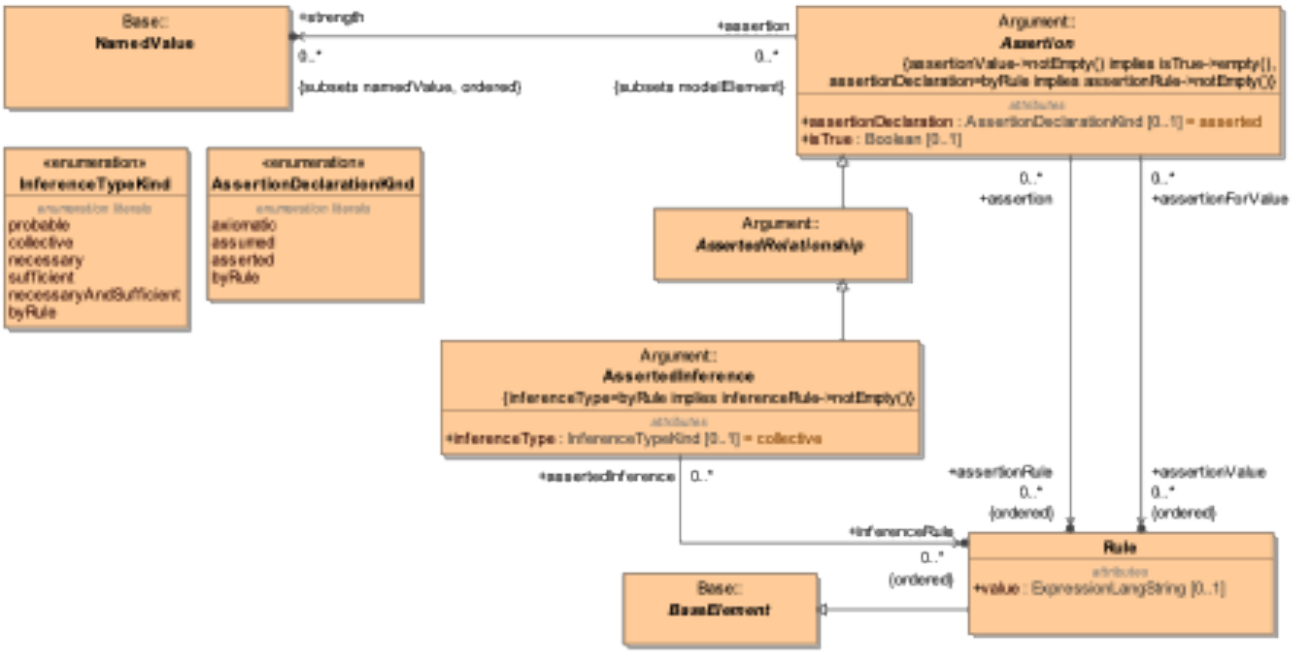

Figure 15.2.1 – Advanced Arguments Diagram

15.2.1 AssertionDeclarationKind [Additions to 13.7]

EnumerationLiteral

- byRule - allows the specification of an extension to AssertionDeclarationKind (e.g. which might allow for Modal logics)

15.2.2 Assertion (abstract) [Additions to 13.9]

Attributes

isTrue : Boolean [0..1] - whether an Assertion has a Boolean value or not. No value (cardinality of 0) can be used to show undecidable things.

AssociationEnds

strength : NamedValue [0..*] {subsets namedValue, ordered} - Strength values for the assertion
assertionDeclarationRule : Rule [0..*] {ordered} - Allows specification of assertionDeclaration "byRule"
assertionValue : Rule [0..*] {ordered} - Allows for specification of assertion value (e.g. Three-valued logic value)

Semantics

SACMDependencies can be defined between strengths, and SACMConstraints can be used to calculate values of strengths based on other strengths.

Constraints

ByRuleDeclarationRequiresRule
If assertionDeclaration is byRule then assertionRule must reference at least one Rule
inv: assertionDeclaration=byRule implies assertionRule->notEmpty()

AssertionValueImpliesNoIsTrue
If there is an assertionValue then isTrue must be empty
inv: assertionValue->notEmpty() implies isTrue->empty()

15.2.3 Rule

Rules allow extension of Argumentation for other kinds of logics and reasoning.

Supertype

Base::BaseElement

Attributes

value : ExpressionLangString [0..1] - Value for the rule. The name can be the content or one can have a name and then the value contains the content.

Semantics

When multiple rules are referenced, lower index rules take precedence over higher index rules.

15.2.4 InferenceTypeKind

InferenceTypeKind enumerates inference types which control the semantics of the AssertedInferenceRelationship.

EnumerationLiteral

- probable - (P) infers that the source make the target more probable (usually shown with strengths)
- collective - (C) [default] infers that all sources are Necessary for the target, taken together all sources together are Sufficient for the target (meant for inductive arguments)
- necessary - (N) infers that the source is Necessary for the target
- sufficient - (S) infers that the source is sufficient for the target
- necessaryAndSufficient - (NS) infers that the source is necessary and sufficient for the target
- byRule - (R) infers that the source has some inference to the target as specified by the Rule in the inferenceRule

Note: these can be counter-inferences by setting isCounter=true.

15.2.5 AssertedInference [Additions to 13.13]

Attributes

inferenceType : InferenceTypeKind [0..1] = collective - specifies the inference type for this inference

AssociationEnds

inferenceRule : Rule [0..*] {ordered} – if inferenceType=byRule, this specifies the ordered set of Rules for the inference type

Constraints

ByRuleMustHaveRule
If inferenceType is byRule then inferenceRule must reference at least one Rule.
inv: inferenceType=byRule implies inferenceRule->notEmpty()

15.2.6 Examples

SACM provides the ability to argue using Deductive, Inductive, Abductive, and user-defined byRule inferencing. The various logics inferencing can be controlled by setting inferenceType using the InferenceTypeKind Enumeration.

In the Abduction Example, there is an argument that Bill is more likely the Killer (raises the probability strength to .6) because Bill has a Yellow Coat. This inference is not defeated because The Killer wore a yellow coat. If the Killer did not wear a Yellow Coat (i.e. that claim is defeated) then the AssertedInference aaa (default inferenceType=collective, i.e. arguing defeation rather than truth value) would defeat the probable inference.

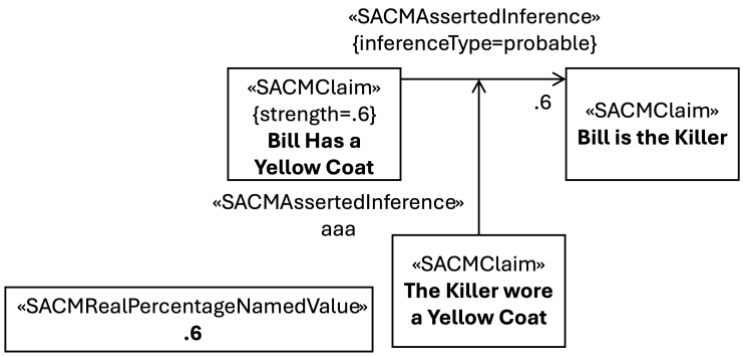

Figure 15.2.6.1 – Abduction Example

The byRule (user-defined) example, ClaimA using TriLogicInference infers ClaimB (which is asserted to be Possible) and has value of MidValue. This facility in SACM allows for user-defined (e.g. Modal) logic to be defined within the SACM model.

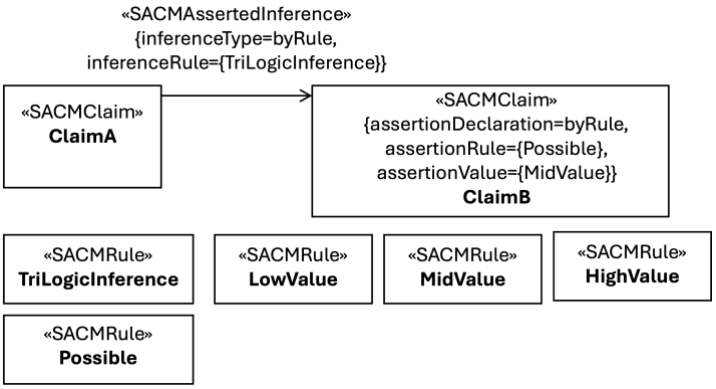

Figure 15.2.6.2 – byRule Example