Date: 2014-05-07

Changes from CySeMoL v2.0 to v2.1

(1) Softened coloring scheme of calculation results

Motivation. This change is introduced due to comfort.

Effect. The change impacts how calculation results are visualized in an object model.

(2) Added possibility to input injective evidence in object models, for faster calculation using rejection sampling and Metropolis-Hastings sampling

Motivation. Previously, providing much evidence in models, lead to slow calculations when using rejection and Metropolis-Hastings sampling. To address the performance problem, a new concept of injective evidence was introduced to CySeMoL.

Function. Unlike the original (classical) concept of evidence, injective evidence unconditionally overrides the derivation of an attribute's value, for which the evidence is provided. Hence, and unlike for the classical evidence, a whole sample across the model is not rejected if a derived attribute value lacks consistency with the provided injective evidence.

Usage. Similarly to inputting the classical evidence, the user needs to select a defense mechanism (or an attack step) of an asset in a model. Subsequently, the property browser, usually to the right from the main modeling control, will include properties named *Functioning_EvidenceToInject* and *Functioning_InjectEvidence* (for attack steps, they would be named *Likelihood_EvidenceToInject* and *Likelihood_InjectEvidence*, respectively). The latter property, *InjectEvidence*, indicates whether injective evidence should be used. The former property, *EvidenceToInject*, indicates what specific evidence should be injected. Both properties contain sub-property called *Evidence*, which contains an OCL expression that has to yield true, false or no value (=> false)). For example, in order to set positive injective evidence for that a specific installation of an operating system is fully patched, the user needs to select that defense mechanism (i.e., *HasAllPatches* on the operating system), and set its properties *Functioning_EvidenceToInject*.*Evidence* to true, and *Functioning_InjectEvidence*.*Evidence* to true. Similarly, in order to set negative injective evidence for that a security awareness program is conducted for some people in the architecture, the user needs to select that security awareness program's defense mechanism called *TrainingConducted*, and set its properties *Functioning_EvidenceToInject*.*Evidence* to false, and *Functioning_InjectEvidence*.*Evidence* to true.

Note. Although injective evidence is not valid to use in all cases, it is safe to use in cases of direct derivation of value. For example, one can use injective evidence instead of the classical one for the availability of defense mechanisms that one ultimately knows are present (or absent), or where one can specify an ultimately trusted probability distribution of their availability. Injective evidence should not be used on any attributes that are being calculated based on the value of another attribute in the object model, because doing so could make the calculation result erroneous. It is generally invalid (unsafe) to use injective evidence for attack steps.

Date: 2014-05-07

(3) Correction of default availability of defense mechanisms (50%-50% now) – for defense mechanisms that do not override this, and those for which no injective evidence is provided

Motivation. In previous version of CySeMoL, the default generic availability of defense mechanisms was set to true, which disallowed simulating full uncertainty of the availability of defense mechanisms that do not further override the value by their own derivation, which affects a subset of defense mechanisms in CySeMoL. Full uncertainty presupposes that there is an equal (50%-50%) chance of the availability being true as false, which the default generic derivation.

Effect. Improvement in calculation.