



Title: User Guide for Modular GSN plugin

Contents

1	Introduction.....	1
1.1	Concept.....	2
1.2	Symbology.....	2
2	Installation.....	5
3	Usage scenario	6
3.1	Public Goals.....	7
3.2	Away Goals.....	7
3.3	Navigating to a linked module or goal	9
3.4	Solved by Contract.....	10
3.5	Rescan Packages	10

1 Introduction

This document is a user guide to explain the use of the Modular GSN plugin and schema for ASCE 4.2.

This plugin and schemas allow you to link together a collection of modules within a project.

Current versions of the product covered by this guide are:

- Modular GSN plugin
 - adelard modular gsn.bundle - version 0.1.8
- Modular GSN Schemas
 - Modular GSN 0.1e
 - Modular GSN 0.1e – Modular Architecture View

This user guide assumes the reader is familiar with GSN and the Modular GSN concepts.

1.1 Concept

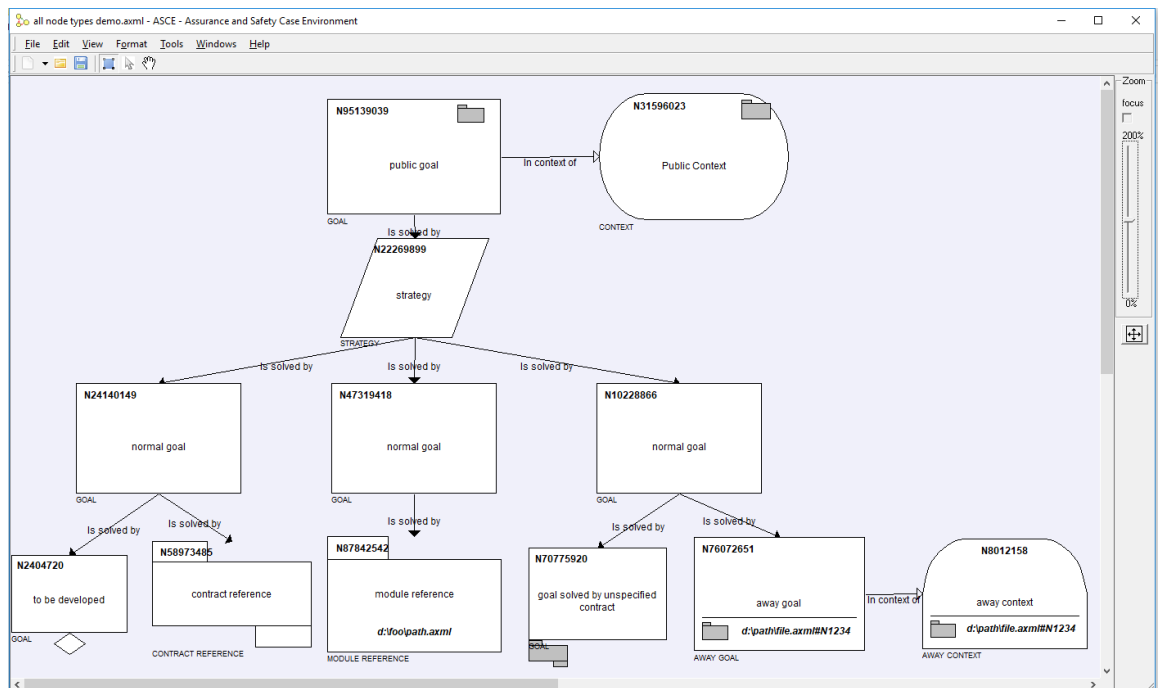
Modular GSN symbology is defined in the GSN community standard¹. This specifies the extensions to “standard” GSN whereby goals may be solved by goals or arguments held in other modules.

Currently the following aspects have been implemented:

- package/architectural level view, referencing sub modules
- modular GSN symbology for new elements and relationships
- public indicator for goals, solutions, contexts and other elements
- referencing away goals/away contexts/away solutions between modules
- a function to update the package view to show references between the modules

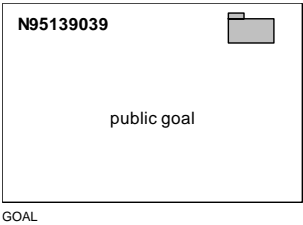
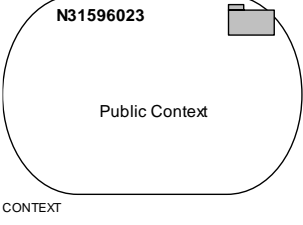

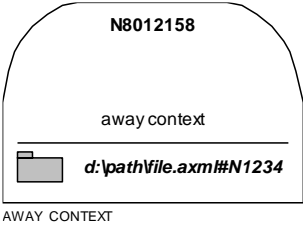
1.2 Symbology

The example of symbology supported is shown below:



¹ http://www.goalstructuringnotation.info/documents/GSN_Standard.pdf

The different node types at the module view in addition to the standard GSN node types are as follows:

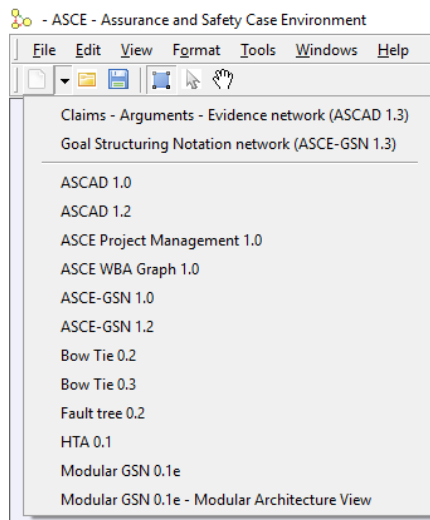
Node type	Option	Symbol shown	Notes
Goal	<i>Is Publicly visible = true</i>		A public goal that may be referenced from other modules
Context	<i>Is Publicly visible = true</i>		A public context that may be referenced from other modules
Away Goal	<i>Away module reference and Away Node reference being set</i>		An Away Goal – a reference to a goal contained in another module
Away Context	<i>Away module reference and Away Node reference being set</i>		An Away Context – a reference to a context contained in another module

Node type	Option	Symbol shown	Notes
Away Solution	<i>Away module reference and Away Node reference being set</i>		An Away Solution – a reference to a solution contained in another module
Module Reference	<i>Away module reference is set</i>		A reference to a remote module
Contract Reference	<i>Away module reference is set</i>		A reference to a remote module that is a Modular GSN contract
Goal solved by contract	<i>Solved By contract = true</i>		Goal to be supported by a Modular GSN contract, which is not yet specified.

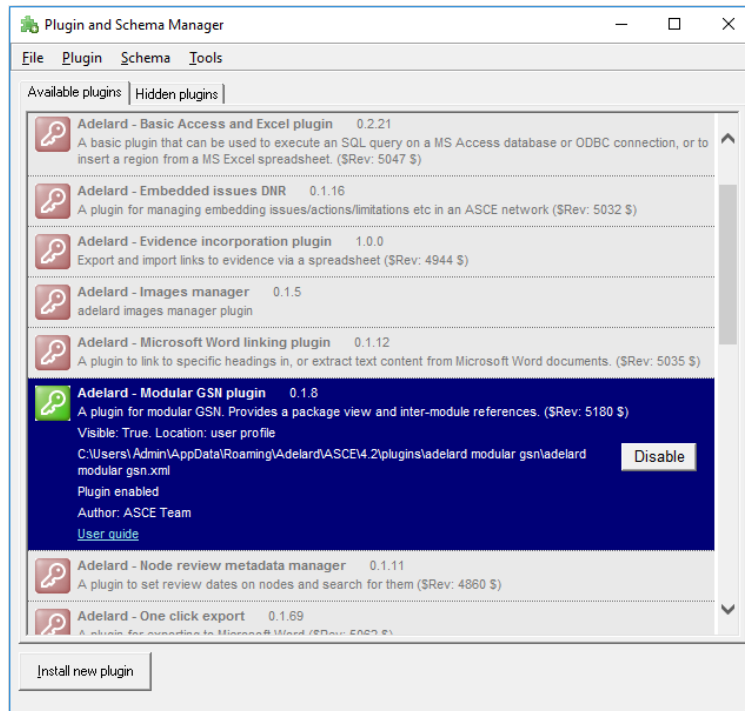
2 Installation

Use the ASCE Plugin Manager to install the schemas and plugin bundle file. Install them to the user profile folder (not system folder).

Then restart ASCE - the schemas should be visible from the *File->New* menu.



The modular GSN plugin will need to be enabled through the ASCE Plugin Manager.



The modular GSN plugin for ASCE is only loaded when an ASCE network using the Modular GSN extensions is opened. If you see extra menu items on the node context menu in such networks, the plugin is successfully loaded.

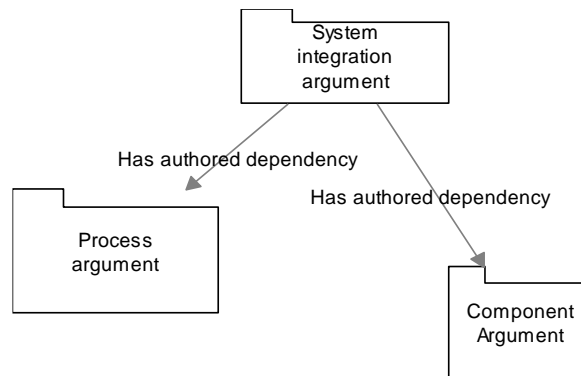
3 Usage scenario

The schemas define node and link types, and their visual display:

- architecture/package view - to define the list of modules and contracts in a case
- module view - an enhancement of the core GSN notation to allow defining public goals, and linking to away goals and contexts etc.

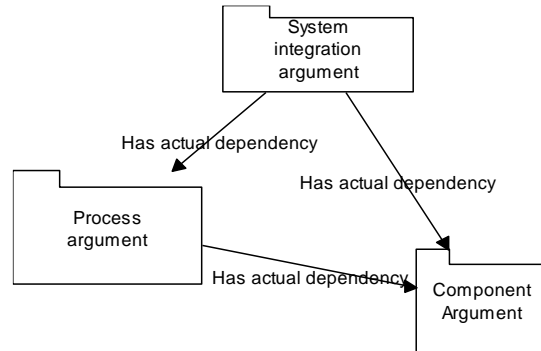
These are typically used together in the following way.

The architecture is where you specify the modules that will exist in the project. At the early stages this may well be merely a statement of intent as the modules referred to may not yet exist. Still, the modules to be created can be defined and broad relationships can be specified. Over time each node in the module view should be linked to an actual module ASCE file. A number of "authored dependency" links can be created to convey the intent of how the modules will eventually be linked together.



Within each module file you create the actual content in a similar way to normal GSN construction. Elements can be linked across the module boundary by specifying that some node refers to a remote element in another network. In this way a set of relationships are authored between the modules. You can navigate from a node to its linked node in another network.

Once a number of relationships have been created, from within the architecture view you can then rescan packages (*Tools -> Modular GSN -> Rescan packages*). This will analyse the linked modules in the project and create a number of discovered links on the architecture view. This summarises the actual dependencies between the modules. As the number of dependencies increase, if appropriate you may remove the "authored dependency" relationships to leave the actual dependencies displayed.



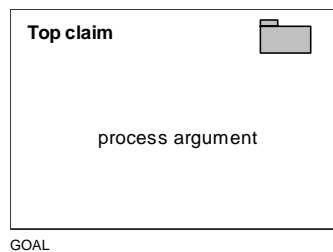
3.1 Public Goals

In Modular GSN schema, you can mark a node (typically a goal or context) as being “publicly visible” using the node properties dialog. This means it will be visible to other networks to be linked to.

Node properties	
Reference	NG5139039
Id	Top claim
Title	process argument
Node type	Goal
Development required	False
Instantiation required	False
Is publicly visible	True
Traffic light	Off
Away module	
Away node reference	
Solved by contract	False
Spectrum 1	Off

When linking to a module, only the publicly visible nodes will be available as the target of the link.

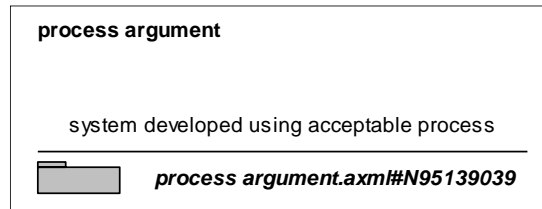
In Modular GSN, a publicly visible element has a folder decorator in the top right hand corner:



3.2 Away Goals

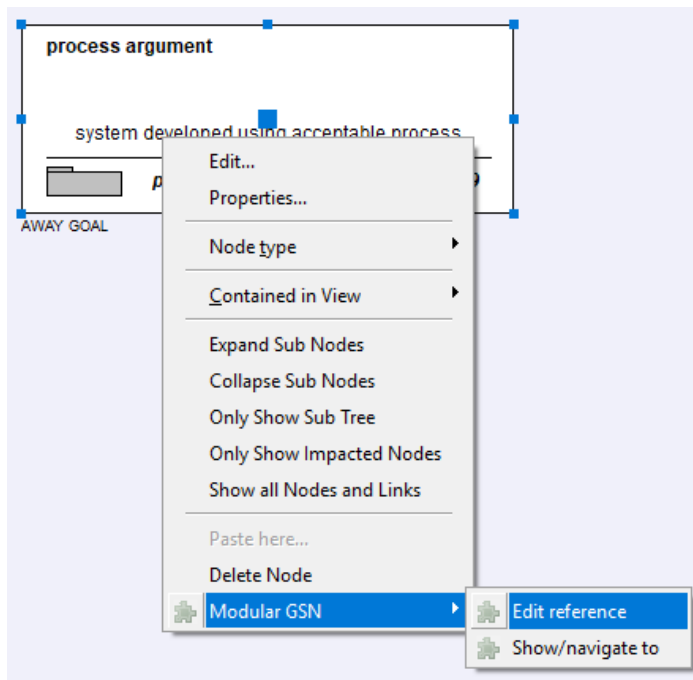
An “Away Goal” is a goal, whose resolution is implemented in another GSN module. An Away Goal is linked to a remote public goal.

It has a folder decorator beneath a line to indicate it has a remote reference:



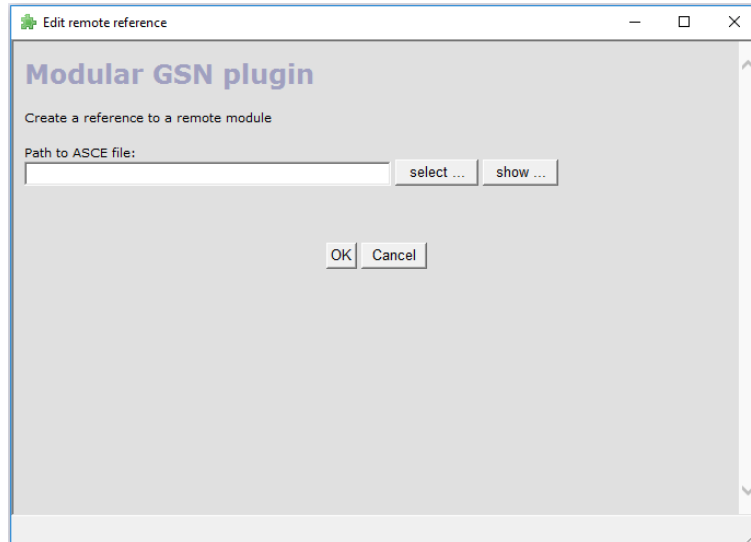
AWAY GOAL

To set up a reference to a remote public goal, select the *Modular GSN* -> *Edit reference* entry from the node's context menu:

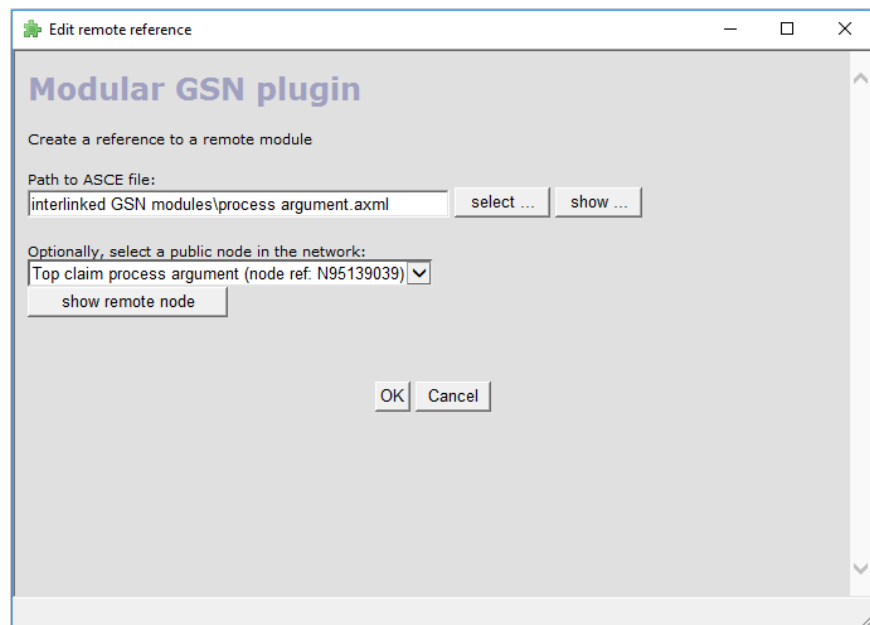


This will then show the following form to view and update the remote reference.

In the architecture/package view, you can set the reference to the module:

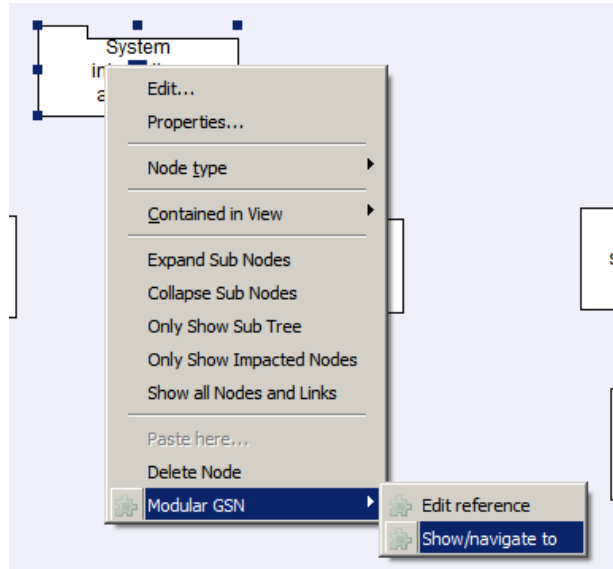


In the module view you can set a reference to a remote module and optionally a public element in that module:



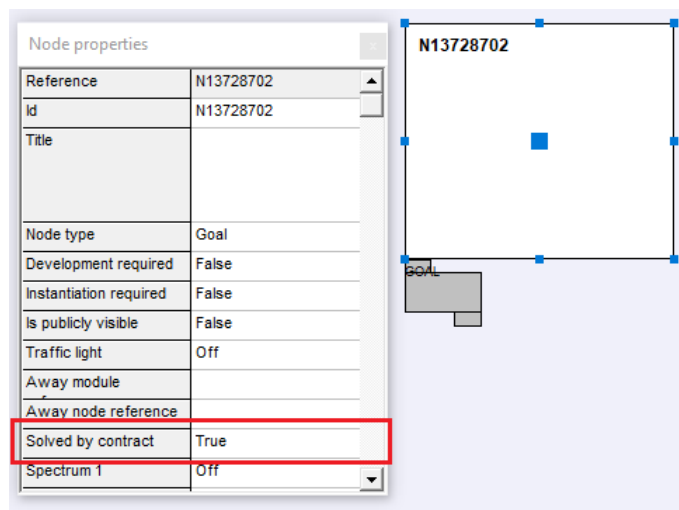
3.3 Navigating to a linked module or goal

Choose the *Modular GSN* -> *Show/navigate to* option from the node context menu:



3.4 Solved by Contract

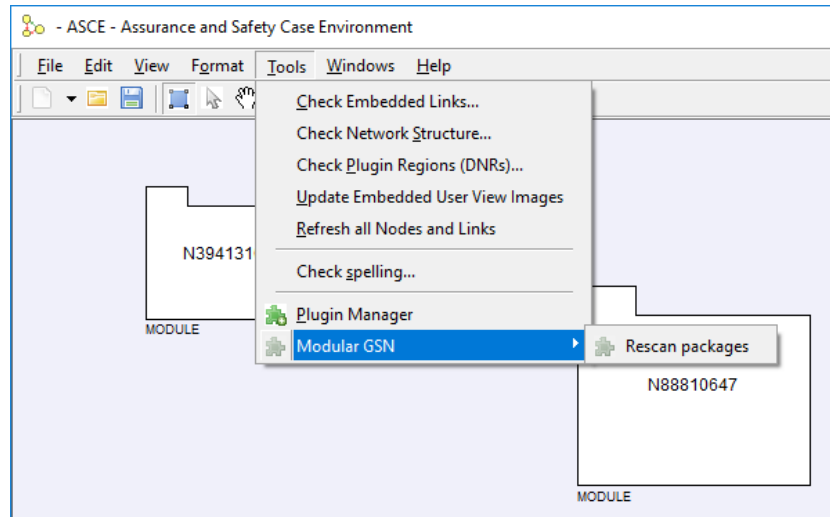
In a GSN module you can denote that a goal is solved by a contract by setting the appropriate node property:



3.5 Rescan Packages

Once the references to the modules are defined and some relationships exist between the modules, you can update the architecture view to show a summary of the links that exist.

In the architecture view choose *Tools -> Modular GSN -> Rescan packages* menu:



This feature will scan the referenced modules contained in the architecture view and update the view to show the actual dependencies between the modules, based on the references found within those modules.