# Averiti Software

The Averiti system provides a number of editor, viewing, and analysis applications to assist in the building and use of domain models. These include:

- Subsystem Editor
- Subsystem Builder
- Import Subsystem
- System Development
- Subsystem Analysis
- Subsystem Viewer
- Averiti Scripting

These are activated from the Averiti Control Panel, illustrated in Figure 1.
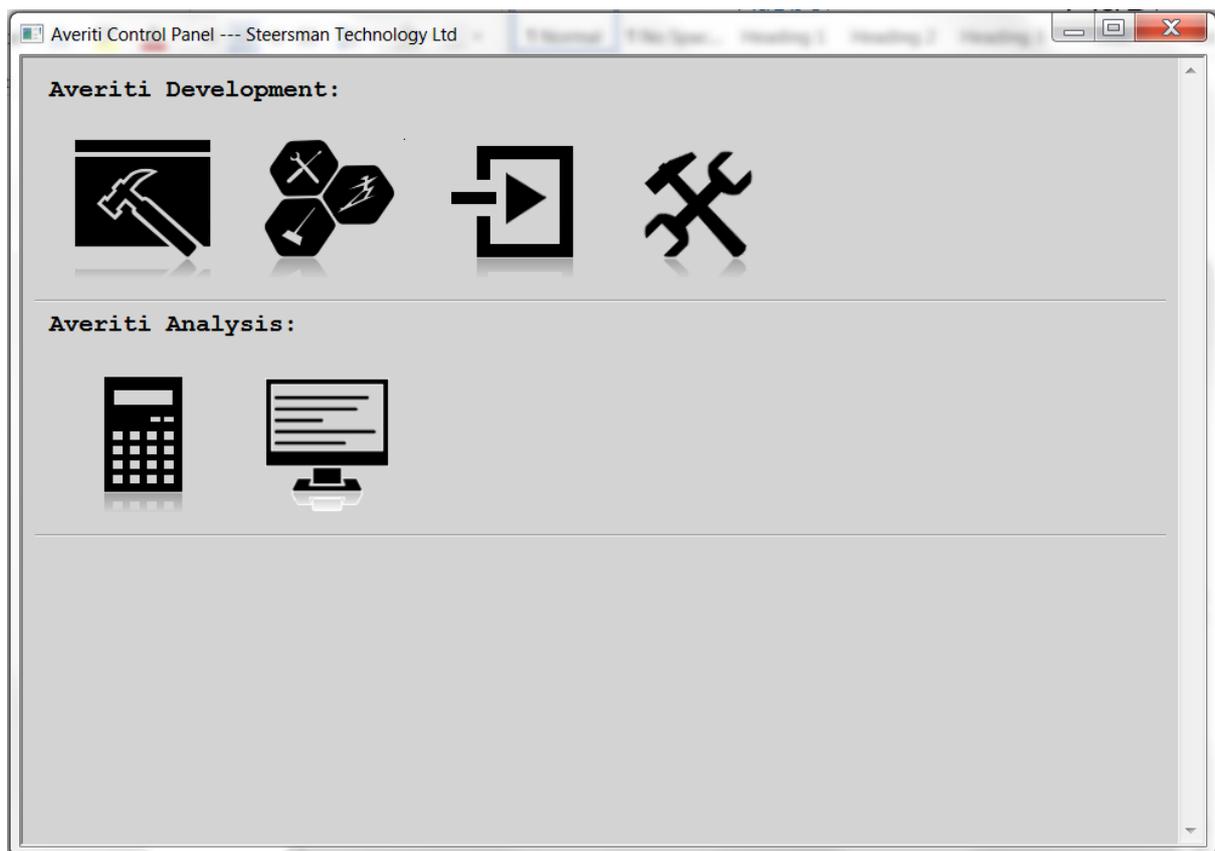
**Figure 1: Averiti Control Panel**

Each Averiti application is described in more detail in the following sections.

Subsystem Editor

The Subsystem Editor provides an XML editor which allows the user to specify subsystems using the Averiti XML notation which are compliant with the subsystem.xsd schema. The user may select to edit an existing subsystem. Once the subsystem has been edited, the user can then select to save and name the subsystem. Subsystems are saved in individual XML files, one per subsystem. This single file per subsystem method was chosen to facilitate the reuse and sharing of subsystems XML

definitions. All subsystem files that comprise a system must be saved in the same folder as each other.

Figure 2 shows a screenshot of the Subsystem Editor.
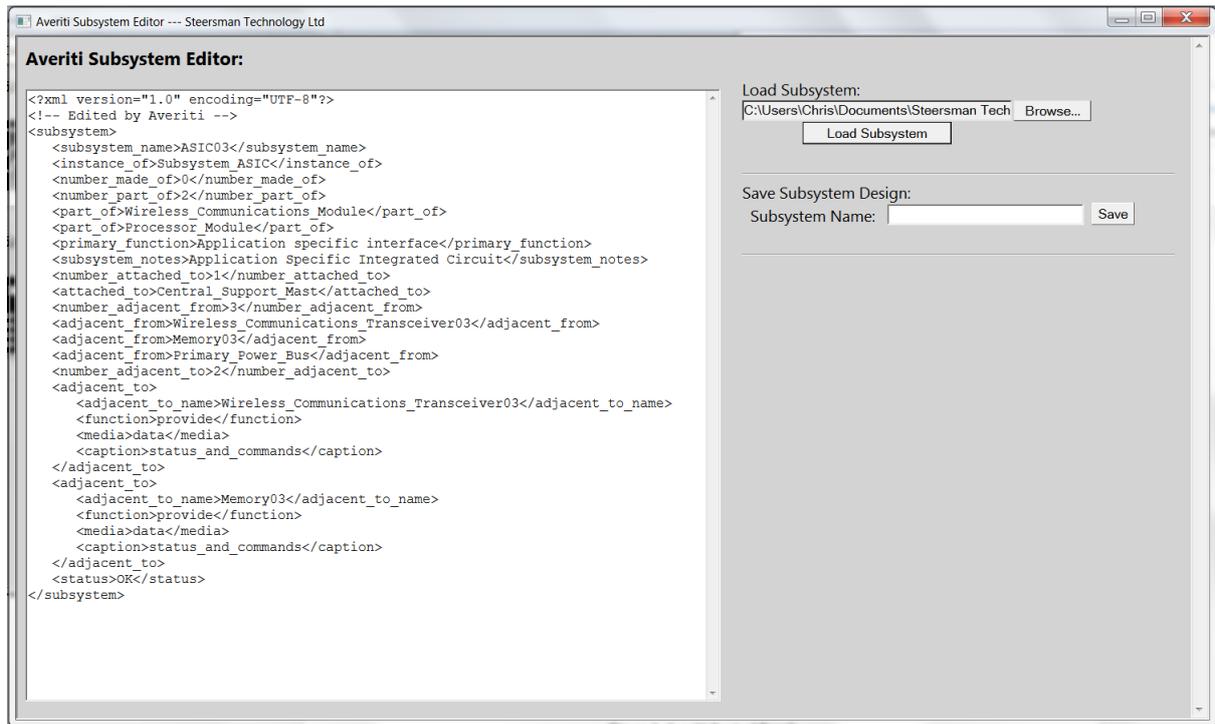
Figure 2: Averiti Subsystem Editor Screenshot

Subsystem Builder

The Subsystem Builder provides an XML editor which allows the user to specify subsystems using the Averiti XML notation which are compliant with the subsystem.xsd schema. The user may select to create a subsystem either based on a blank subsystem XML template or select a prototype subsystem from a domain library. For any case the user must populate the attribute values with the data that is required for the subsystem in question. Once the subsystem has been defined, the user can then select to save and name the subsystem.

Figure 3 shows a screenshot of the Subsystem Builder.

Averiti Subsystem Builder:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited by Averiti -->
<subsystem>
    <subsystem_name>null</subsystem_name>
    <instance_of>null</instance_of>
    <number_made_of>0</number_made_of>
    <made_of>null</made_of>
    <number_part_of>0</number_part_of>
    <part_of>null</part_of>
    <primary_function>null</primary_function>
    <subsystem_notes>null</subsystem_notes>
    <number_attached_to>0</number_attached_to>
    <attached_to>null</attached_to>
    <number_adjacent_from>0</number_adjacent_from>
    <adjacent_from>null</adjacent_from>
    <number_adjacent_to>0</number_adjacent_to>
    <adjacent_to>
        <adjacent_to_name>null</adjacent_to_name>
        <function>null</function>
    <media>null</media>
    <caption>null</caption>
    </adjacent_to>
<status>null</status>
</subsystem>
```

Load Subsystem Template: [Load Template]

Clear Subsystem Template: [Clear Template]

Load Prototype:
[_____] [Browse...]
[Load Subsystem]

Save Subsystem Design:
Subsystem Name: [_____] [Save]

Figure 3: Averiti Subsystem Builder Screenshot

Import Subsystem

Any subsystem defined using either the Subsystem Editor application or the Subsystem Builder application must be imported into the systems folder for the current project. The Import Subsystem application allows this to be achieved and allows the user to rename the imported subsystem on import, for example, to comply with a project naming convention.

Figure 4 shows a screenshot of the Import Subsystem.

Import Subsystem --- Steersman Technology Ltd

Select Subsystem Import Options:

Select Subsystem to Import:

[_____] [Browse...]

Enter Subsystem Name: [_____]

[Import system]

Figure 4: Averiti Import Subsystem Screenshot

System Development

The System Development application provides an XML editor which allows the user to specify a system as a collection of subsystems defined in a system XML file in accordance with the system.xsd schema. The user may select to create a new system from the system template, in which case an empty XML file in the system format is displayed to the user, which can then be edited with the system names. Alternatively an existing system file can be selected, in which case the existing system file data is presented to the user for editing. In either case the user can select to name and save the system file once editing is complete. The system file describing the subsystems of a system must be saved in the same folder as the subsystem XML files.

Figure 5 shows a screenshot of the System Builder.
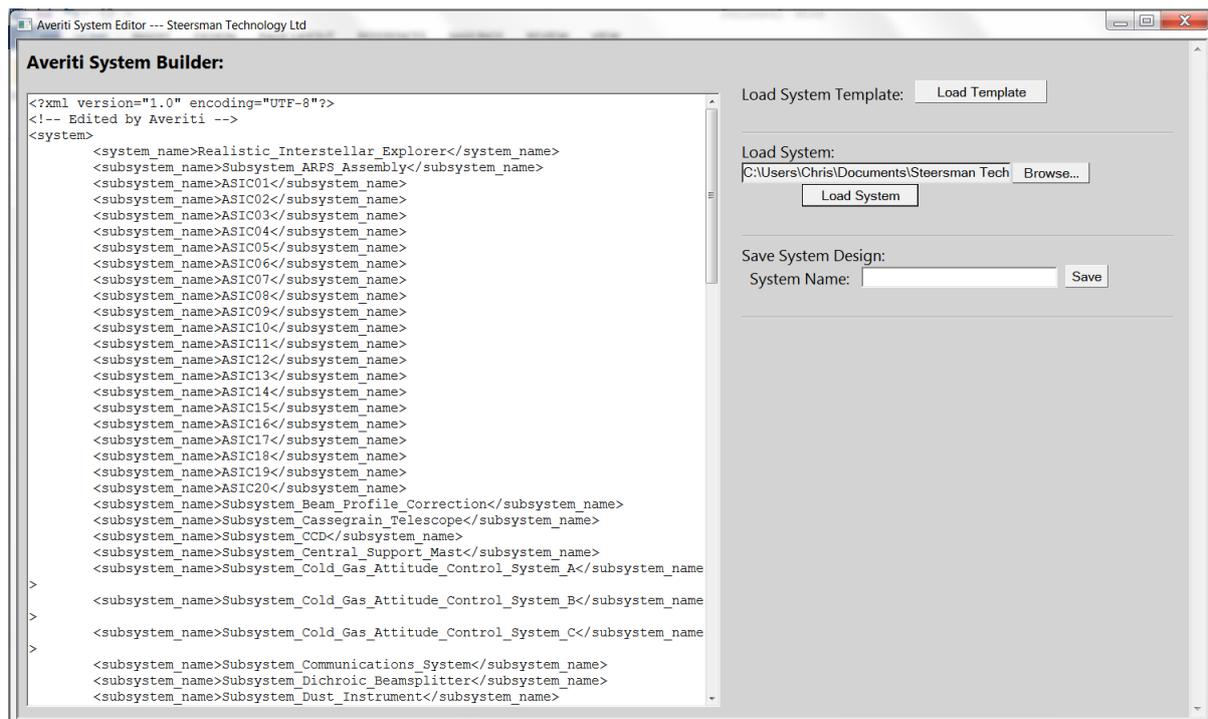
Figure 5: Averiti System Development Screenshot
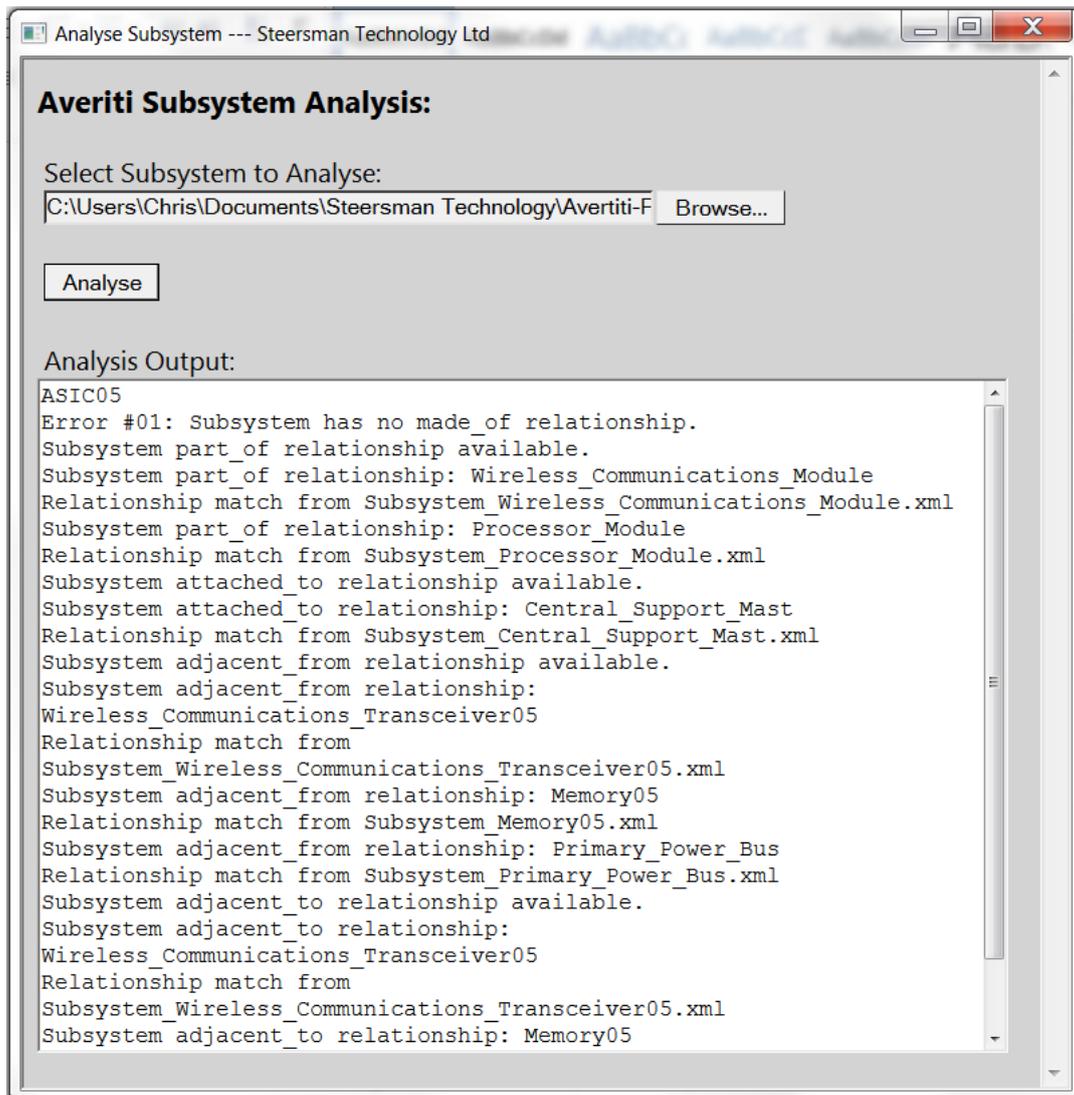
Averiti Subsystem Analysis

The Averiti Subsystem Analysis application allows the user to select a Subsystem for analysis. The analysis process checks the relationships defined in the Subsystem such as:

- Made_of
- Part_of
- Attached_to
- Adjacent_to

Errors are raised where relationships do not exist or are not referenced correctly.

The application opens each of the Subsystems referenced in the relationships and checks that the corresponding return path relationship is included. In this way the consistency of the model can be checked.

The output of the analysis is displayed in the application dialog, as shown in Figure 6, and is saved as a text file named after the subsystem (e.g. analysis_output_Propellant_Tank.txt) in the output folder.

Figure 6: Averiti Subsystem Analysis Screenshot

Averiti View Subsystem

The Averiti View Subsystem application allows the user to select a Subsystem to view. The application shows the subsystem's various relationships, as shown in Figure 7.
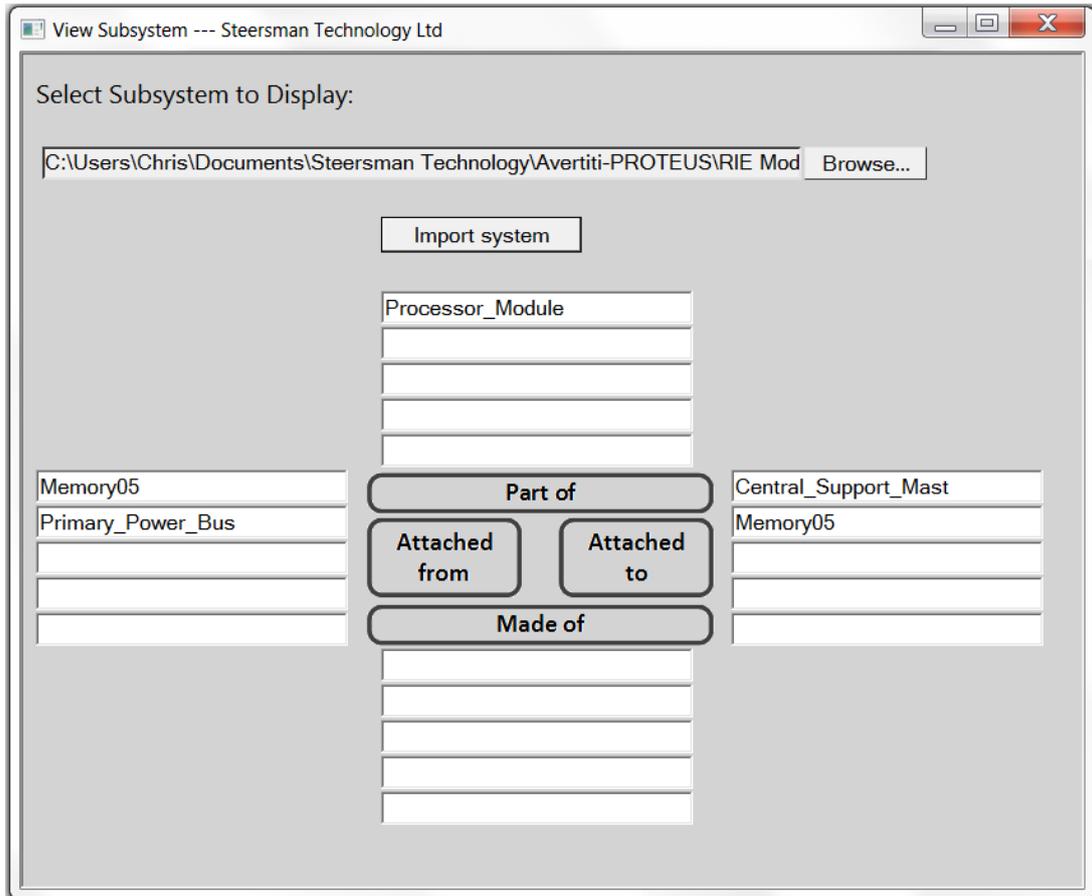
Figure 7: Averiti View Subsystem Screenshot