IMPaCT Wizard – Basic Functionality

This document specifies the activities of the IMPaCT wizard that provide its basic functionalities. It also provides a preliminary specification of interfaces for using such activities.

Future work may consist of adding activities that provide more complicated features, and specifying additional interfaces.

Note that the user of terms such as ‘must’, ‘may’, ‘can’ etc. comply with the usual RFC (albeit that we do not capitalize such words).

# ACTIVITIES

This chapter specifies the activities that can be executed and the constraints that apply when the result is delivered.

## Create/Edit Component

The result of this activity is that a (new or modified) component exists that satisfies the following constraints:

1. the component has an ID that distinguishes it from all other components;
2. the component has a (non-empty) description;
3. the interface of the component is specified, which means that a set of ports has been specified for that component, and every port satisfies the following constraints:
   1. the port has a name (id) that allows it to be distinguished from all other ports on the component;
   2. the port is either an input or an output (not: none, or both);
   3. the port may have a (configuration) type assigned (e.g. ‘float’);
   4. the minimum number of wires for the port may be specified;
   5. the maximum number of wires for the port may be specified;
   6. only if the port is an input port, then a default value may be specified;

## Create/Edit Compound

The result of this activity is that a (new or modified) compound exists that satisfies the following constraints:

1. the compound has an ID that distinguishes it from all other compounds;
2. the compound has a (non-empty) description;
3. the interface of the compound is specified, which means that a set of ports has been specified for that compound, and every port satisfies the following constraints:
   1. the port has a name (id) that allows it to be distinguished from all other ports on the compound;
   2. the port is either an input or an output (not: none, or both);
   3. the port may (not) have a (configuration) type assigned (e.g. ‘float’);
   4. the minimum number of wires for the port may (not) be specified;
   5. the maximum number of wires for the port may (not) be specified;
   6. (only) if the port is an input port, then
      1. it may be assigned the property of being a ‘constant port’ (meaning that a constant must provide its value);
      2. a default value may be specified;
4. the compound may contain components, each of which satisfies the following constraints:
   1. the component has been defined (i.e. satisfies the requirements of section 1.1)
   2. the component has an ID that distinguishes it from all other components within the compound
5. the compound may contain constants, each of which satisfies the following constraints:
   1. the constant has a name (id) that allows it to be distinguished from all other constants in the compound;
   2. the constant can be assigned the property of ‘configuration constant’, which means that when it is part of a fully configured application or project, it must have been assigned a value.
   3. the constant can only be connected to a port that has the property of being a ‘constant port’;
   4. the constant must be connected to at least one Port;
   5. every port to which the constant is connected is a port on a component within the compound;
   6. if the constant and a port(s) to which it is connected are both typed, then these types are the same;
6. the compound may contain (proxy)wires each of which satisfies the following constraints:
   1. the wire connects a single (source) port to one or more target ports;
   2. the source of a (proxy)wire is either an input port of the compound, or an output port of a component within the compound;
   3. the target of a (proxy)wire is either an output port of the compound, or an input port of a component within the compound;
   4. a (proxy)wire cannot connect (to) a port that has the property of being a ‘constant port’ (that’s the task of constants);
   5. if two ports that are connected by one (proxy)wire have (configuration) types, then these types must be the same;

## Create/Edit Application

The result of this activity is that a (new or modified) application exists that satisfies the following constraints:

1. the application has an ID that distinguishes it from all other applications;
2. the application has a (non-empty) description;
3. the application may contain components, each of which satisfies the following constraints:
   1. the component has been defined (i.e. satisfies the requirements of section 1.1);
   2. the component has an ID that distinguishes it from all other components within the application;
4. the application may contain compounds, each of which satisfies the following constraints:
   1. the compound has been defined (i.e. satisfies the requirements of section 1.2);
   2. the compound has an ID that distinguishes it from all other compounds within the application;
5. the application may contain constants, each of which satisfies the following constraints:
   1. the constant has a name (id) that allows it to be distinguished from all other constants in the application;
   2. the constant can be assigned the property of ‘configuration constant’, which means that when it is part of a fully configured application or project, it must have been assigned a value.
   3. the constant can only be connected to a port that has the property of being a ‘constant port’;
   4. the constant must be connected to at least one Port;
   5. every port to which the constant is connected is a port on a component or (outer perimeter of a) compound within the application;
   6. if the constant and a port(s) to which it is connected are both typed, then these types are the same;
6. the application may contain wires each of which satisfies the following constraints:
   1. the wire connects a single (source) port to one or more target ports;
   2. the source of a wire is an output port of a component or (outer perimeter of a) compound within the application;
   3. the target of a wire is an input port of a component or (outer perimeter of a) compound within the application;
   4. a wire cannot connect (to) a port that has the property of being a ‘constant port’ (that’s the task of constants);
   5. if two ports that are connected by one wire have (configuration) types, then these types must be the same;

## Create/Edit Project

The result of this activity is that a (new or modified) project exists that satisfies the following constraints:

1. the project has an ID that distinguishes it from all other projects;
2. the project has a (non-empty) description;
3. the project may contain applications, each of which satisfies the following constraints:
   1. the application has been defined (i.e. satisfies the requirements of section 1.3);
   2. the application has an ID that distinguishes it from all other applications within the project;

## Configure a Project

The result of this activity is that an existing project not only satisfies all constraints as mentioned in section 1.4, but also the following ones:

1. every constant that is within an application or compound that is within the scope of the project, and has the property of ‘configuration constant’, has been assigned a value;
2. the project has been assigned the property ‘configuration is confirmed’.

# INTERFACES

This chapter specifies the interfaces that are made available, and for every interface the activities that it can execute and any constraints that may apply.

## Interface for Developer/Device manufacturer

This interface allows a user to do the following activities:

1. Create/Edit Component
2. Create/Edit Compound

Constraints that hold for this interface:

1. If logins are supported, the user can
   1. create components and compounds, which will subsequently be owned by the organization that the user is a part of;
   2. edit components or compounds that are owned by the organization that the user is part of;
2. The user cannot assign the property of ‘configuration constant’ to constants.

## Interface for Engineer

This interface allows a user to do the following activities:

1. Create/Edit Application
2. Create/Edit Project

Constraints that hold for this interface:

1. If logins are supported, the user can
   1. create applications and projects, which will subsequently be owned by the organization that the user is a part of;
   2. edit applications and projects that are owned by the organization that the user is a part of.

### Interface for Technician

This interface allows a user to do the activity ‘Configure a Project’.

Constraints that hold for this interface:

1. If logins are supported, the user can operate on projects that are owned by the organization that the user is a part of, as well as information within such projects.
2. The user can only edit constants that are part of (compounds that are part of) applications that are part of projects that the user can operate on.