Error Handling in Variables

Danny Haynes
The MITRE Corporation

Overview

- Quick review of variables
- Where errors may be encountered
- Issues with error handling in variables
  - Variable values must conform to datatype
  - What does it mean to report errors?
- Wrap-up

Quick review of variables

- Variables are containers that can hold one or more values of the same datatype and are identified with a globally unique ID
  - Can be referenced from object entities, state entities, and variables
- constant_variable
  - No frills … just values
- external_variable
  - Obtain values from sources outside of an OVAL Definitions document at run-time
    - XCCDF Benchmark document, OVAL Variables document, etc.
- local_variable
  - Manipulate values with functions
  - Obtain values from sources local to an OVAL Definitions document
    - Collected items and other variables

Where errors may be encountered

- constant_variable
  - Values don't match the specified datatype
- external_variable
  - Values don't match the specified datatype
  - Values don't conform to the specified possible values or restrictions
  - External source cannot be found
- local_variable
  - Same problems as constant and external variables (it can reference them)
  - Functions encounter errors
  - Referenced objects are non-existent or are missing information
- Object and state entities that refer to variables with no values

Variable values must conform to datatype

- **Specified in the documentation, but, not enforced**
  - "*The set of values identified by a variable must comply with the specified datatype, otherwise an error should be reported.*" [oval-def:variable]
  - **No schema types or Schematron rules**
    - **Left for interpreters to enforce**
  - **No casting ... not even integer to float!**

- **Casting is allowed in other places**
  - "*If the data being worked on is not of the correct type, a cast should be attempted before reporting an error.*" [oval-def:FunctionGroup]
  - "*Another way of thinking of things is that the datatype attribute specifies how the data should be cast before performing the operation (note that the default datatype is 'string').*" [oval-def:EntityAttributeGroup/@datatype]
  - **Legal casts are not documented**

- **Do we really want variable values to conform to the datatype?**

Looking forward

- Remove the requirement that variable values must conform to the specified datatype
  - Values would be loosely typed
  - No more datatype attribute on variables
- Value types must only be enforced when an operation is applied
  - Object and state entities
  - Function input parameters
- We need to consider how to deal with external_variables
  - Restrictions do not allow for the specification of a datatype
- To make all of this work we need type casting
  - Need to specify legal casts
  - Literal variable values are of type string

What does it mean to report errors?

- Let's look at an example
  - "…If the variable being referenced does not have a value…If the element is part of a state declaration, then the state element referencing it will evaluate to error." [oval-def:EntityAttributeGroup/@var_ref]
  - "If a variable does not have any value when referenced by an OVAL State an error should be reported during OVAL analysis." [oval-defEntityAttributeGroup/@var_check]

- Unfortunately, how to handle errors is not clear
  - Opportunity for inconsistent results

- We need an implementation-independent way to ensure that we handle errors consistently!

Looking forward

- **Luckily, we already have a way using flags!**
  - **Unfortunately it is not well documented**

Number of Components with Flag				
Error	Complete	…	Not applicable	Resulting Flag
1+	0+	…	0+	Error
0	1+	…	0	Complete
0	0+	…	0	Incomplete
0	0+	…	0	Does not exist
0	0+	…	0	Not collected
0	0+	…	1+	Not applicable

- **We need define more of these tables**
  - **Variables**
    - **Functions**
    - **Components**
  - **Combining object entity flags and how it affects the collected_object flag**
    - **Object entity refers to an incomplete variable**
    - **Object entity refers to a complete variable**
  - **Determining state entity flags and how it affects evaluation results**
    - **State entity refers to an incomplete variable**

- **Do all of the flag values make sense?**

Wrap-up

- Documentation, documentation, documentation
  - Fix inconsistences
  - Further specify existing documentation
  - Develop new documentation
    - Legal casting
    - Tables for combining flags
- Are there other issues that we should consider?
- Other questions, comments, or concerns?