PySAR

Model-Driven Engineering that doesn’t hurt

Created by Cornel Izbașa / cornel.izbasa@opensynergy.com

”Easy” AUTOSAR MDE using Eclipse

- Your stuff
- ARTOP
- Sphinx
- EMF
- Eclipse

Some Concerns

- Initial development cost
  Product with many deps - Maven, Gradle, Ant...
- Maintainability
  Embedded devs depend on model engineers
- Safety qualification
  Daunting component stack size and complexity
- Licenses (EPL + ARTOP + yours)
  Too many components => legal gray zone
- Questionable community
  Few incentives to publish fixes and ehancements

PySAR

Python **S**imple **A**rchitecture **R**endering

- Model access and code generation framework
  Covers common phases of the MDE workflow
- Written in Python
  Embedded devs can use it easily
- Open, small and dynamic
  Easy fixes, tailoring and extensions

AUTOSAR-like Model Access

```python
from pysar import ar

if __name__ == '__main__':
  ar.set_schema('somesar_4.xsd')
  root = ar.AR(ar.read('ecuc.xml'))

  Demo = root.find_first(
    'MODULE_CONFIGURATION',
    DefinitionRef='/SOMESAR/SomeDefs/Demo')
  print Demo

  DemoGeneral = Demo.getDemoGeneral
  print DemoGeneral

  debounce_counter_support = DemoGeneral.getDemoDebounceCounterBasedSupport
  print debounce_counter_support

  status_changed_callbacks = DemoGeneral.getDemoCallbackDTCStatusChanged
  for callback in status_changed_callbacks:
    print callback.getDemoCallbackDTCStatusChangedFnc

  print Demo.getDefinition

  system = root.find_first('SYSTEM')
  print system.getShortName

  system_signal = root.find_first('SYSTEM_SIGNAL')
  print system_signal
  print system_signal.getDynamicLength
```

Output

```
AR(MACC_MODULE_CONFIGURATION_VALUES : Demo)
AR(MACC_CONTAINER_VALUE : DemoGeneral)
False
Some_Function
Other_Function
AR(MACC_MODULE_DEF : Demo)
System
AR(SYSTEM_SIGNAL : SYSTEM_SIGNAL1)
False
```

AUTOSAR-like Code Generation

```
from pysar import ar
from pysar import ar_cg

def gen_cfg_h(module):
  def_name = module.getDefinition.getShortName
  cg = ar_cg.CodeGen('bsw/' + def_name + '_Cfg_h.j2')
  return cg.generate(
    header_base = def_name.upper(),
    Demo=module)

if __name__ == '__main__':
  ar.set_schema('somesar_4.xsd')
  root = ar.AR(ar.read('ecuc.xml'))
  Demo = root.find_first(
        'MODULE_CONFIGURATION',
        DefinitionRef='/SOMESAR/SomeDefs/Demo')
  print gen_cfg_h(Demo)
```

MSN_Cfg_h.j2

```
{% block header %}
#ifndef {{ header_base }}_CFG_H
#define {{ header_base }}_CFG_H
{% endblock %}

{% block source %}
{% endblock %}

{% block footer %}
#endif /* {{ header_base }}_CFG_H */
{% endblock %}
```

Code Templates

Dem_Cfg_h.j2

```
{% extends "bsw/MSN_Cfg_h.j2" %}
{% block header %}{{super()}}{% endblock %}

{% block source %}
#define DEMO_DEBOUNCE_COUNTER_BASED_SUPPORT STD_{{
  Demo.getDemoGeneral.getDemoDebounceCounterBasedSupport and 'ON' or 'OFF' }}

{% for callback in Demo.getDemoGeneral.getDemoCallbackDTCStatusChanged %}
extern StatusChangedCallback_t {{
  callback.getDemoCallbackDTCStatusChangedFnc }}(DTCId_t, Status_t, Status_t);
{% endfor %}
{% endblock %}

{% block footer %}{{super()}}{% endblock %}
```

Output

```
#ifndef DEMO_CFG_H
#define DEMO_CFG_H

#define DEMO_DEBOUNCE_COUNTER_BASED_SUPPORT STD_OFF

extern StatusChangedCallback_t Some_Function(DTCId_t, Status_t, Status_t);
extern StatusChangedCallback_t Other_Function(DTCId_t, Status_t, Status_t);

#endif /* DEMO_CFG_H */
```

Architecture

- Python framework
  Easy, widely supported, dynamic language
- lxml.objectify for model access
- Jinja2 for code generation
  Tried and true for Web apps
- Applicable to any models
  Initially tuned for AUTOSAR-like models
- Independent of AUTOSAR artifacts
  User supplies AUTOSAR schema and model element name mappings
- Designed for openness and freedom
  Not encumbered by legal or technical dependencies