

S1000D Logic Engine – How does it work?

Vojtech Toman

Principal Software Engineer
EMC Corporation

Agenda

Process Data Module

Logic Engine

Logic Engine Implementation

Q/A

Process Data Module

Process Data Module

A specification for interactive processing structures

- Troubleshooting, testing etc.

Sequencing steps and/or other S1000D data modules

- “Open door 1, Open circuit breaker B, ...”

Information grouping

State information

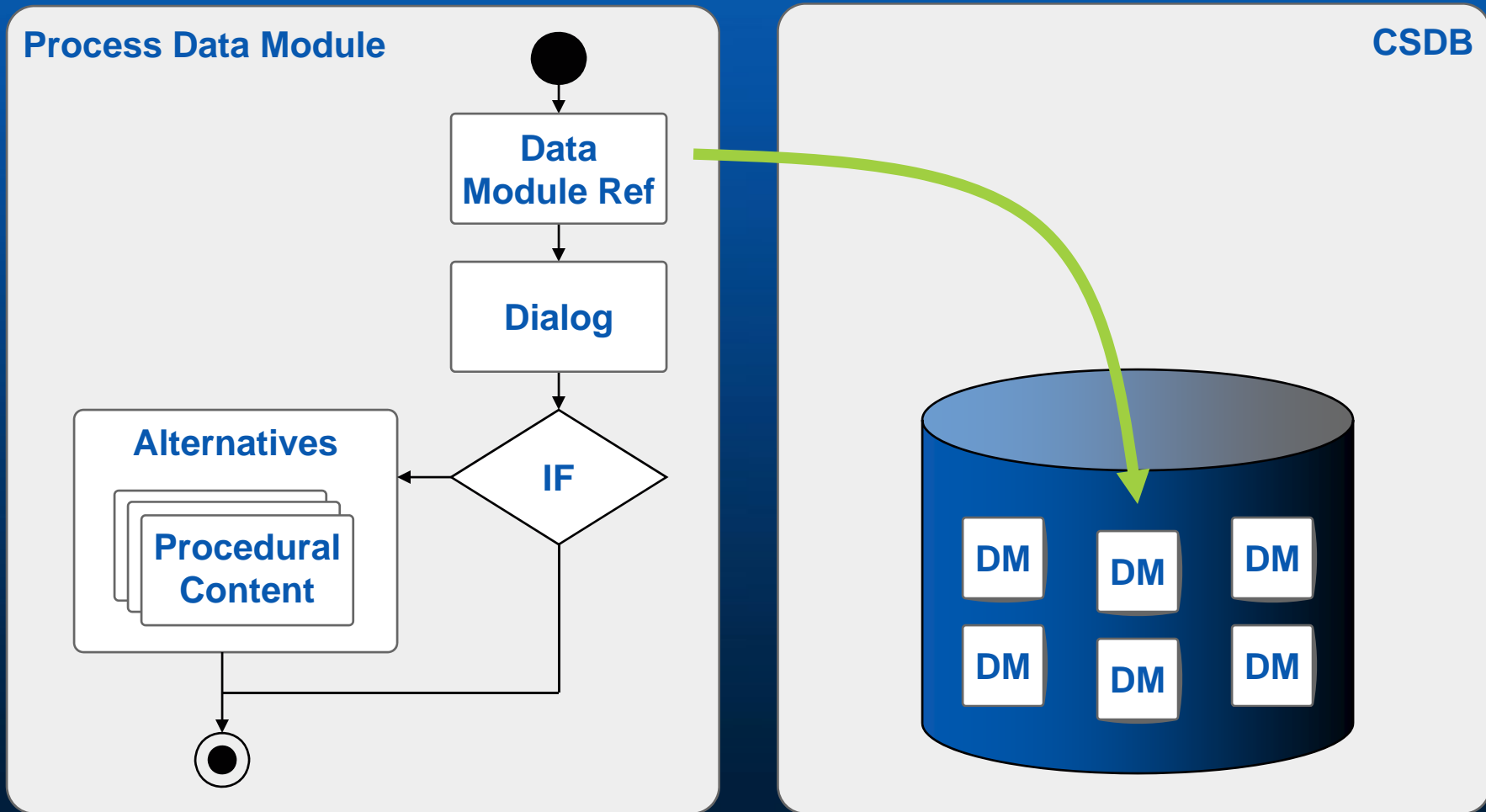
Sequencing based on state information

Context filtering based on state information

User interaction

Process Data Module

Overall Schema



Process Data Module

Logic Engine

Executes/interprets S1000D process data modules

Provides the user with the current view of the process

Implements the interactive presentation functionality

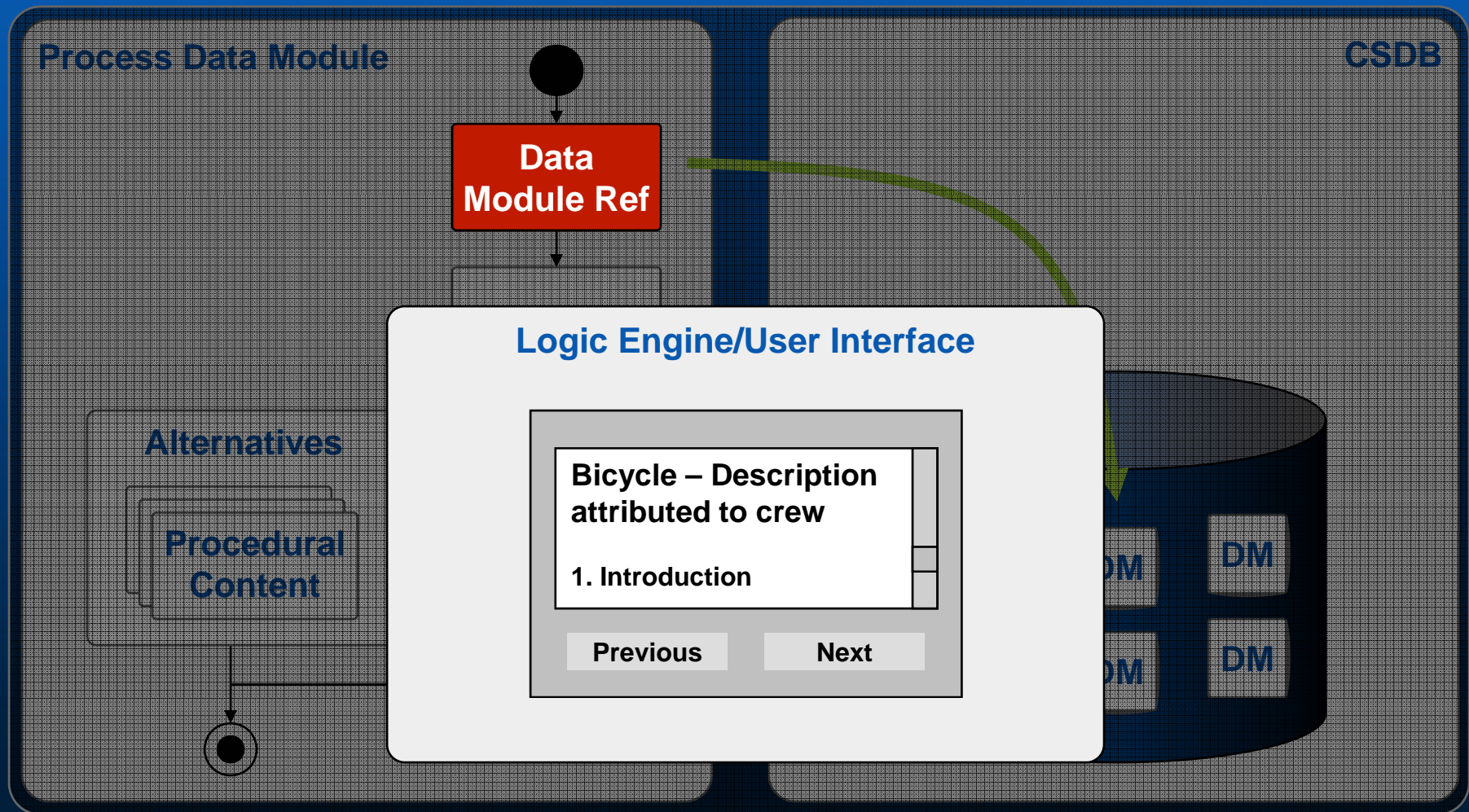
Maintains state information

Integrates with external systems

Error handling/reporting

Process Data Module

Process Execution



Process Data Module

Process Execution – IETP Example

The screenshot displays a software interface with a search results window and a preferences dialog box. The search results window shows a document titled "Bicycle - Riding a bicycle" with a description attributed to the crew. The preferences dialog box is open, showing options for "Left brake lever" and "Right brake lever".

Search - Results S1000DBIKE-AAA-D00-00-00-00AA-950A-D Bicycle - Riding a bicycle 01

Serial number=1B070701 | Service bulletin S001 - Chain guard=Pre

Unclassified

Bicycle - Description attributed to crew

References

Table 1. References

Data module/Technical publication	Title
Shifters - Description of how it is made	Shifters - Description of how it is made

1 **Introduction**
Data about the bicycle and its control system is given in this document. This data will help you operate the bicycle.

2.1 **Controls**
Data about the controls that follow is given in this document:

- [Steering](#)
- [Shifters](#)

is operated by* Left brake lever
 Right brake lever

OK Cancel

Process Data Module

Information Grouping

Data module node

- Defines the information that is to be displayed on the screen

Procedural Content

Lubricate the bicycle

1. Make sure the chain is clean and dry
2. Apply the lubricant to each roller of the chain

Previous

Next

Data Module Reference

Bicycle – Description attributed to crew

1. Introduction

Previous

Next

Dialog

Bicycle checkout

Tire pressure (PSI)

OK

Cancel

External Application

Audio player



OK

Cancel

Process Data Module

State Information

Variables

- Identified by name
- String, Integer, Real, Boolean, Set
- May be given an initial value or an “init-dialog”

State table

- List of variables and their values

Expressions

- Combinations of variables, static values, and operators
- Rich set of operators

Assertions

- Assignment of values to variables
- Variable = Expression

State Table

Variable	Value
product	“bi ke”
pressure	45
temperature	<i>NOVALUE</i>

Process Data Module

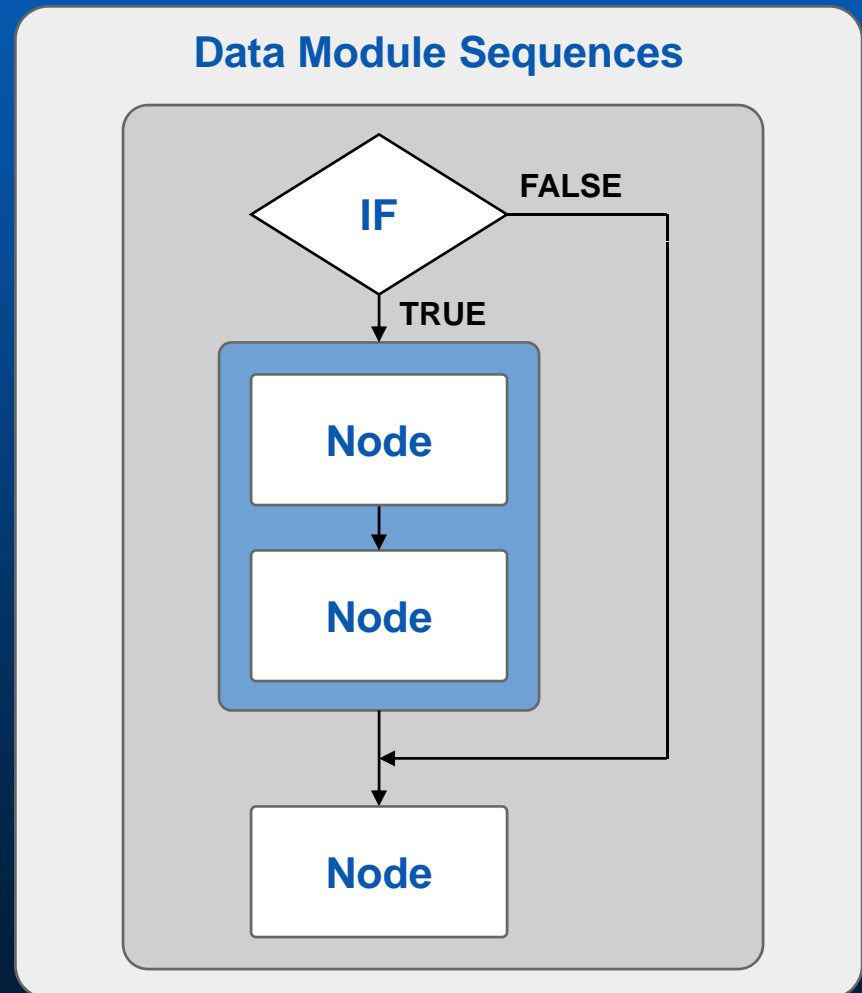
Sequencing of Information

Data module sequences

- Organize data module nodes into sequences
- Can be nested

Conditional processing

- If-then-else branching
- Loops



Process Data Module

Context Filtering

User is presented only information that applies to his situation

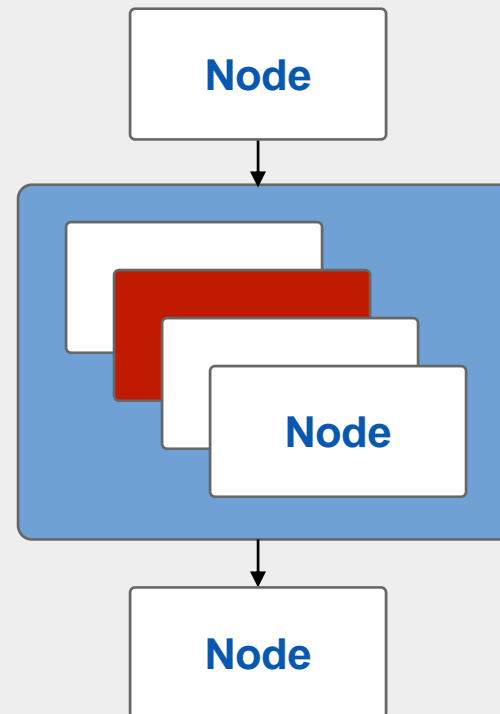
Applicability filtering

- “Standard” applicability
- Expression-based applicability

Alternative data module node

- Process the first applicable node in the list of data module nodes
- One procedure that handles different configurations of the product

Alternative Data Module Node



Process Data Module

Dialogs

Collection of user information

Two types of dialogs

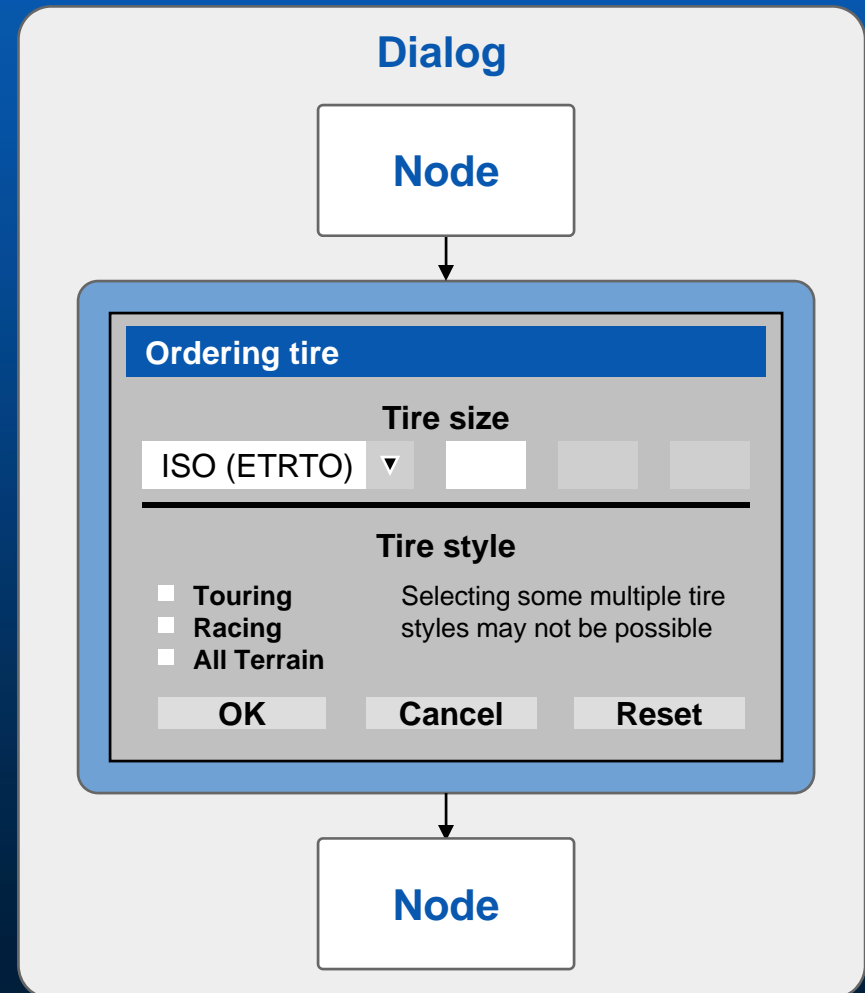
- Explicit dialogs
- "Init" dialogs

Five basic types of controls

- User entry
- Menu
- Button
- Message
- Dialog group

Interactive functionality

- Input validation
- Dynamic evaluation of dialog state



Process Data Module

External Applications

Calls to external applications

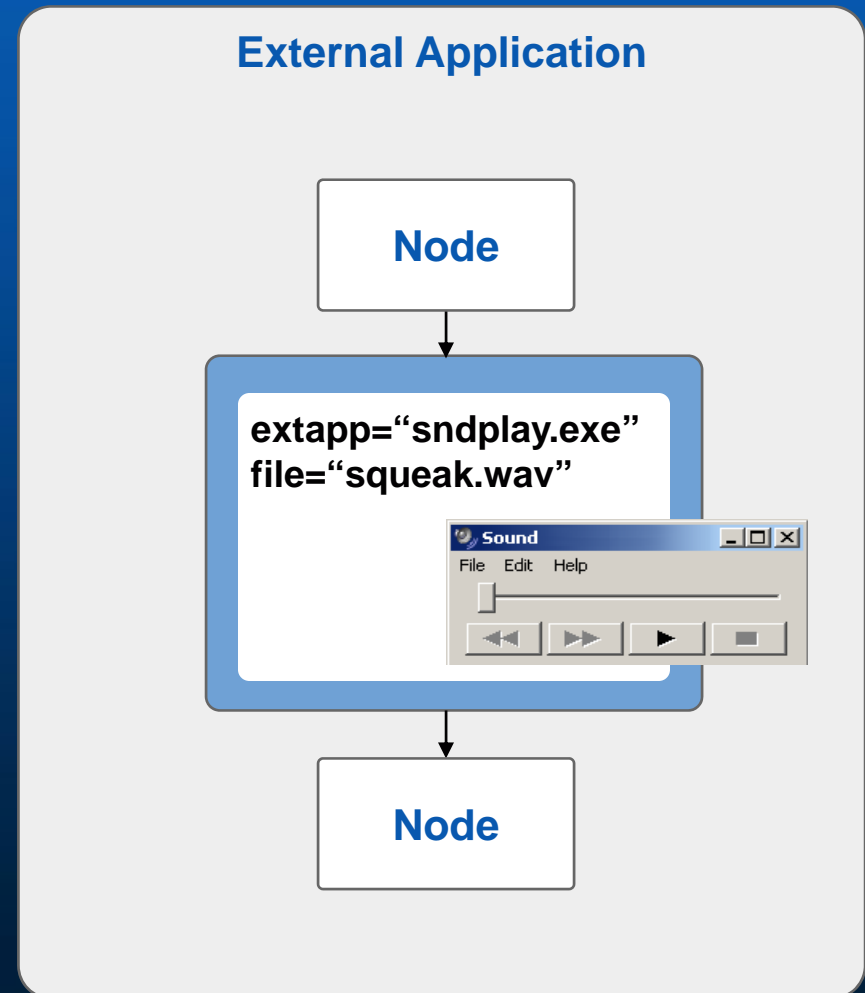
- Diagnostic tool, media player, ...
- External subsystem

Input parameters

- Name (optional)
- String value or a reference to a process variable

Return values

- Assigned to process variables
- Processed sequentially or by name

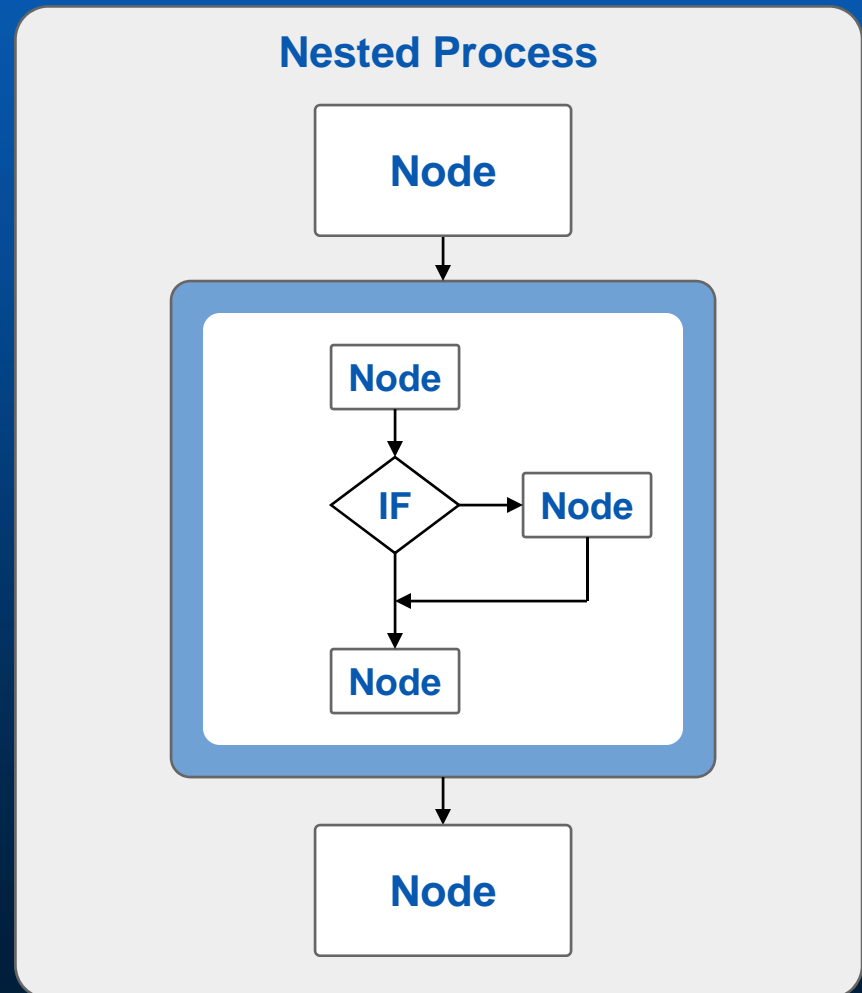


Process Data Module

Nested Processes

Data module node with a reference to a process data module

- Using other process data modules as “subroutines”
- Reuse of process data modules



Process Data Module

Would-be-nice Features

Reuse within a process data module

Try-catch functionality

Support for other dialog representations

- XForms

Logic Engine

Logic Engine

Executes/interprets S1000D process data modules

- Determines what to display next

Implements the interactive presentation functionality

- Navigation functions

Maintains state information

- Expression evaluation

Integrates with external systems

Error handling/reporting

Logic Engine

Navigation Functions

Next

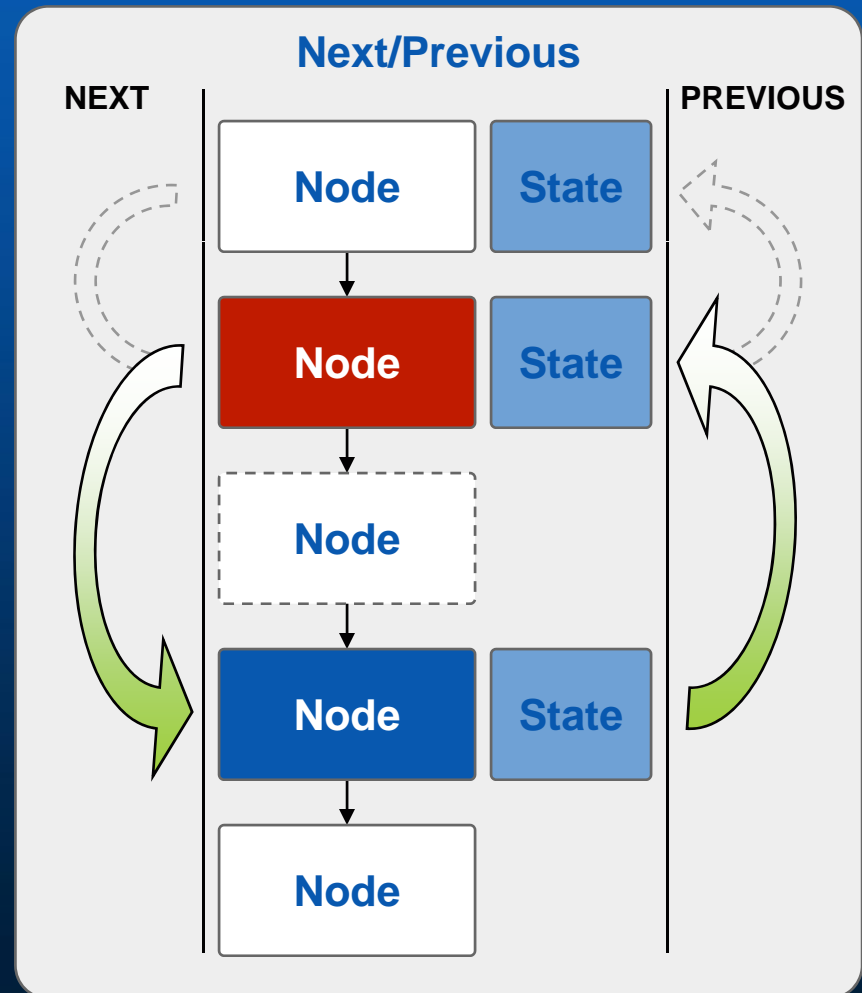
- Proceed with interpreting the process data module from the current position
- Applicability filtering

Previous

- Logical reverse of “Next”
- Restores the original state

Dialog-related

- Submit, Cancel, Reset



Logic Engine

Data Module Node Processing

Applicability

- Applicability associated with the node

Variable pre set

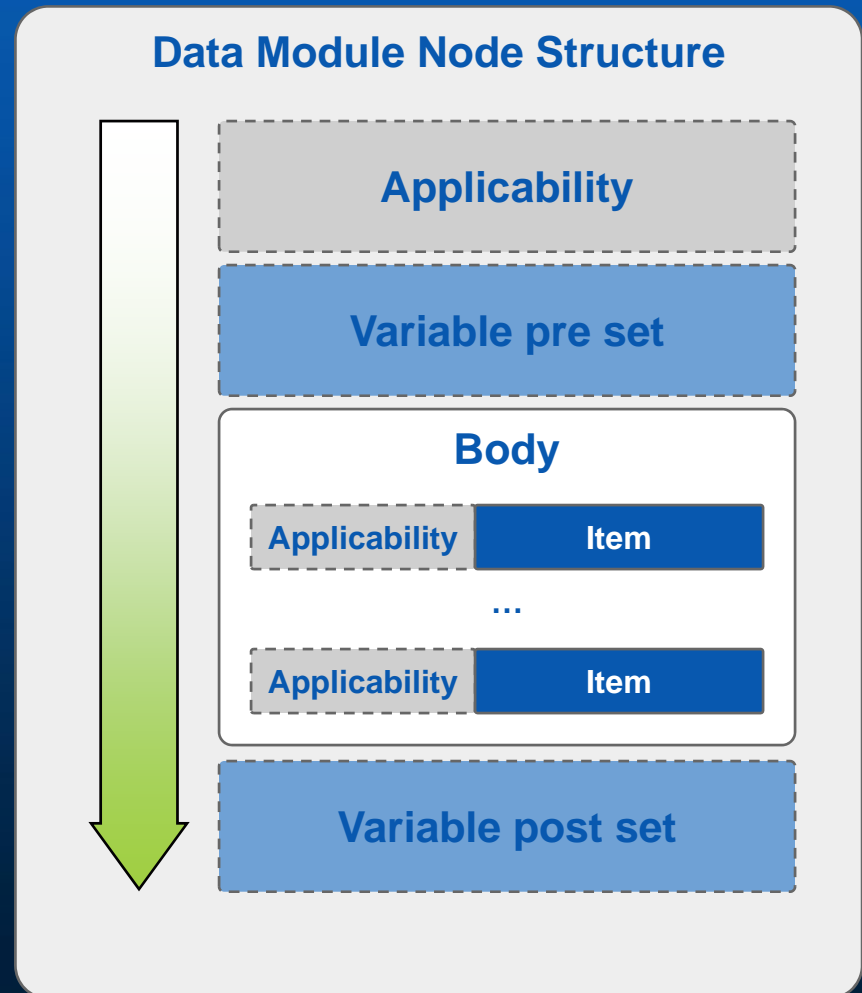
- Assertions to be evaluated before node is traversed

Body

- Contains a list of items
- Subject to applicability filtering

Variable post set

- Assertions to be evaluated after node has been traversed



Logic Engine

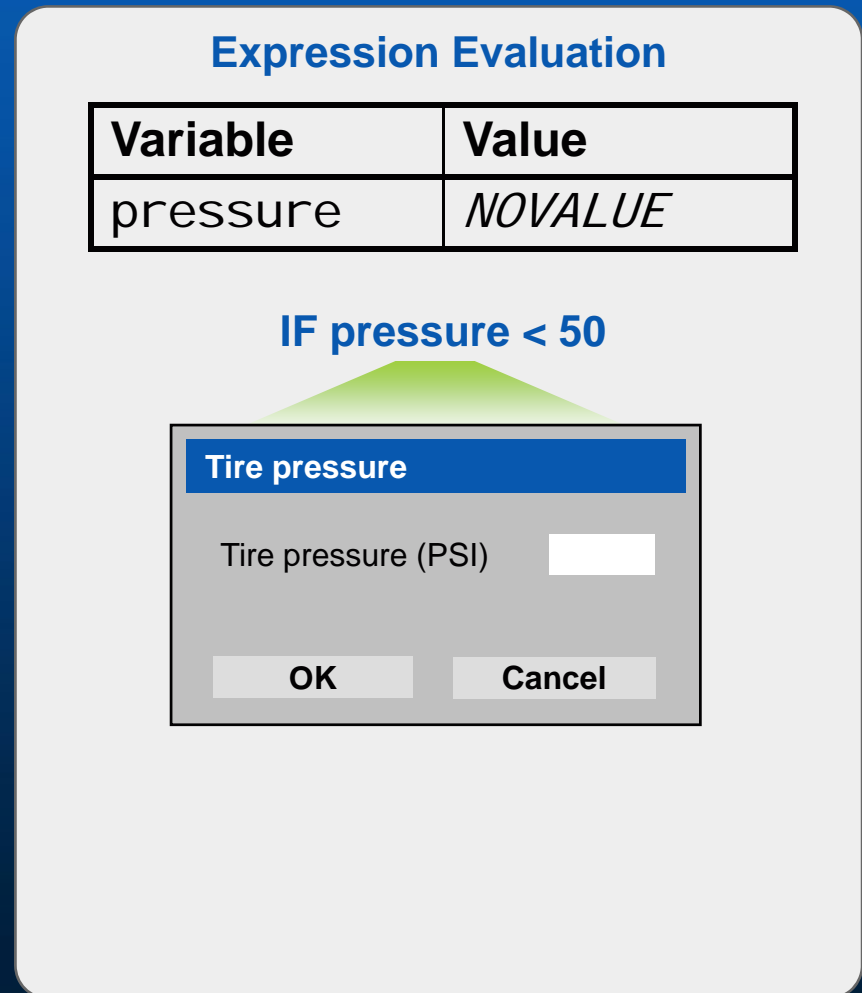
Expression evaluation

Fundamental feature of the process data module

- Updating of state table
- Evaluating conditions
- Evaluating applicability
- Validation

Dealing with uninitialized variables

- Display the "init-dialog"
- Fail



Logic Engine Implementation

Logic Engine

Architecture

User interface

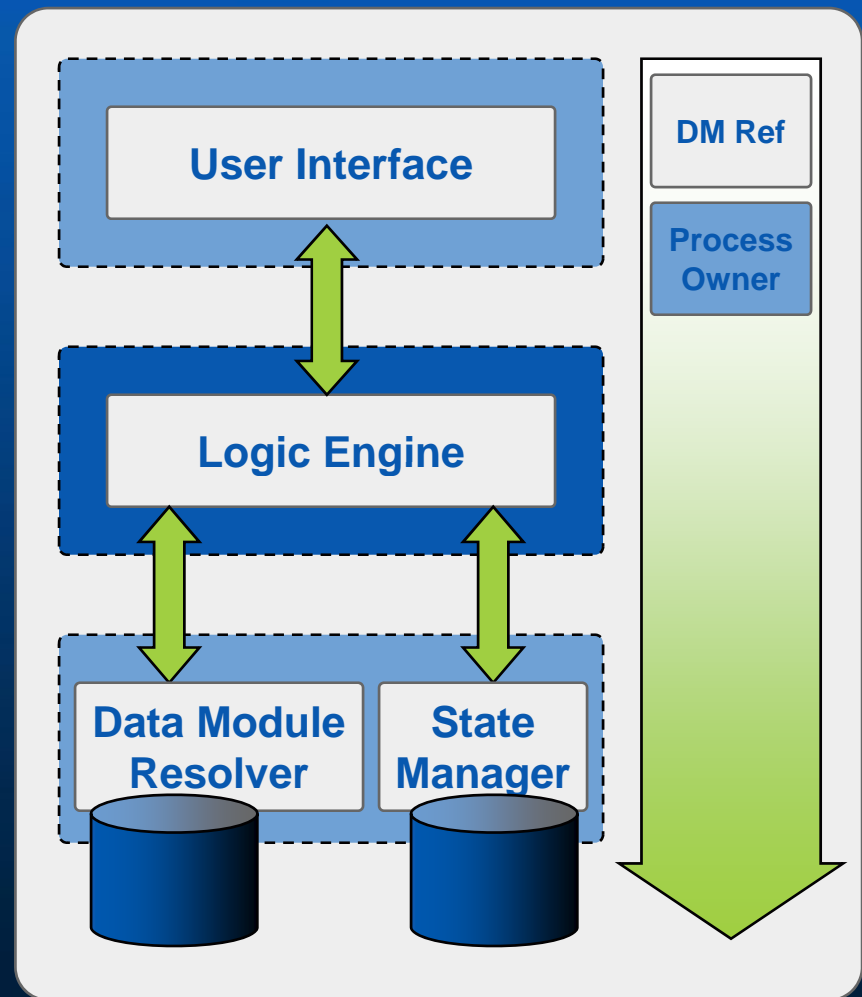
Logic Engine

Data Module Resolver

State Manager

Data Module Reference

Process Owner



Logic Engine

Process State Management

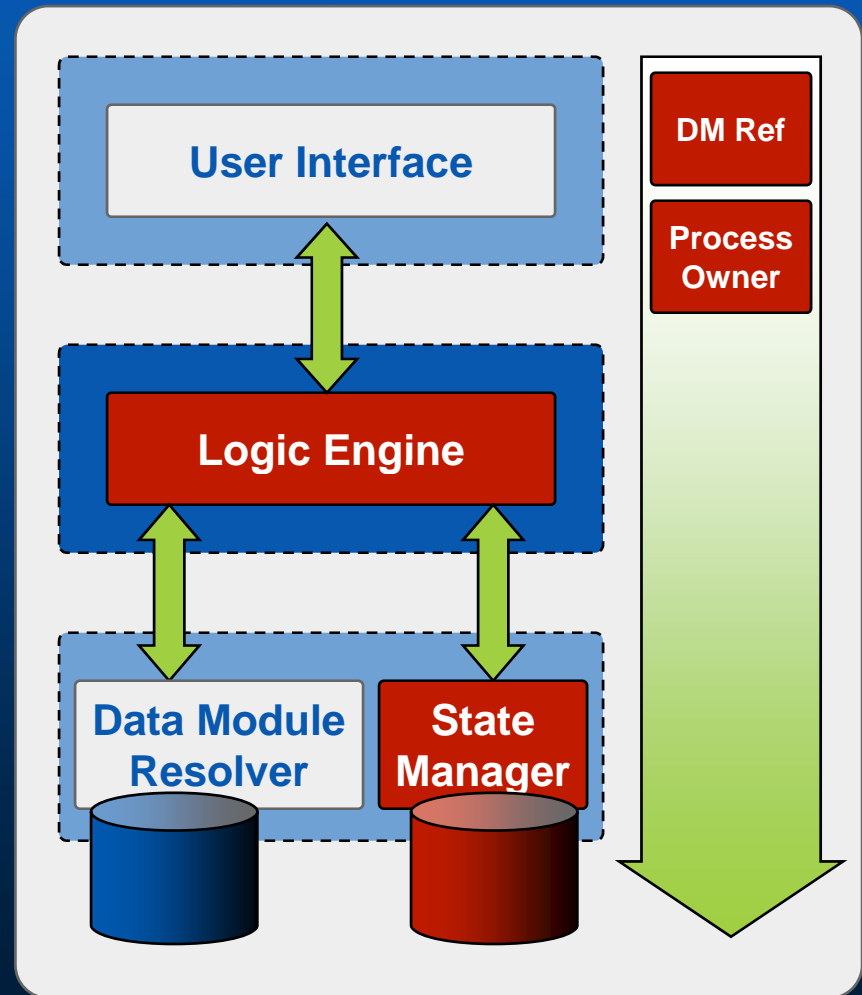
Logic engine maintains the state information of process instances

State information is externalized and stored as an XML document

- XML schema for representing the state information
- State Manager is responsible for the storage

State information of one process instance is identified by:

- Data module reference
- Process owner



Logic Engine

Process State Document

```
<s:processState xmlns:stat="http://www.emc.com/documentum/xml/le/state">
  <s:ref xlink:href="process/bike.xml" xmlns:xlink="http://www.w3.org/1999/xlink"/>

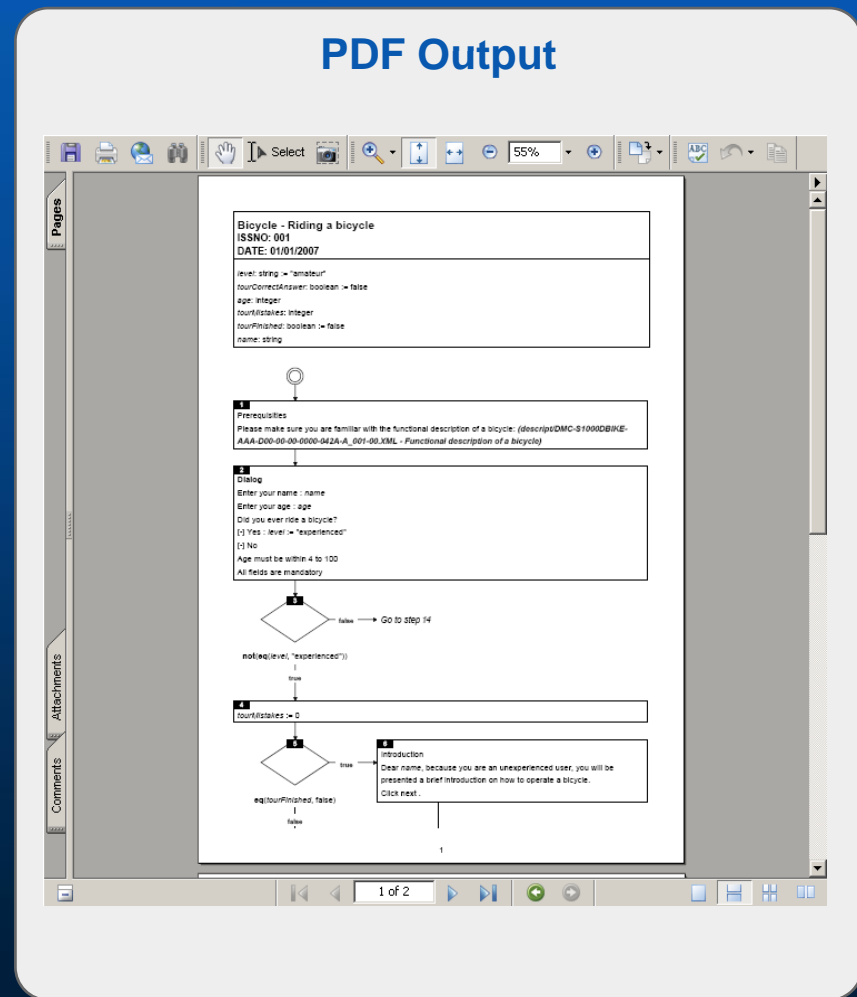
  <s:stateStack>
    <s:state>
      <s:baselineContext>
        <s:variables>
          <s:variable name="pressure">
            <s:value type="integer">45</s:value>
          </s:variable>
        </s:variables>
        <s:locator>
          <impl:locatorData xmlns:impl="http://www.emc.com/documentum/xml/le/impl">
            <impl:step id="19" position="EVAL_POSTSET"/>
          </impl:locatorData>
        </s:locator>
      </s:baselineContext>
    </s:state>
    ...
  </s:stateStack>
</s:processState>
```

Logic Engine

Process Data Module Rendering

Printable representations of process data modules

- PDF support



Questions and Answers

Vojtech Toman
toman_vojtech@emc.com

www.emc.com
www.x-hive.com