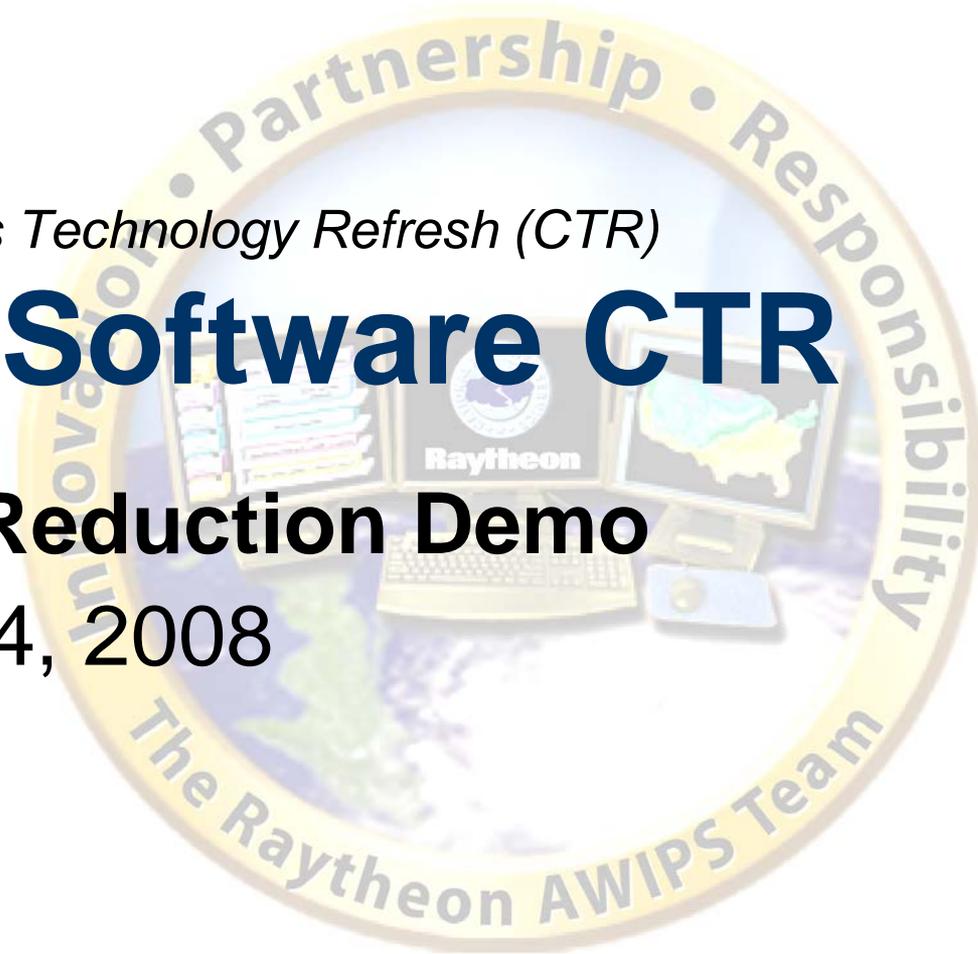


AWIPS Continuous Technology Refresh (CTR)

AWIPS Software CTR

T09: Risk Reduction Demo

September 4, 2008



This document includes data that shall not be duplicated, used, or disclosed – in whole or in part – outside the Government for any purpose other than to the extent provided in contract DG133W-05-CQ-1067. However, the Government shall have the right to duplicate, use, or disclose the data to the extent provided in the contract. This restriction does not limit the Government's right to use information contained in this data if it is obtained from another source without restriction. The data subject to this restriction are contained in all sheets.

T09 Grid Derived Parameter Pattern

New Pattern based on Python Scripts

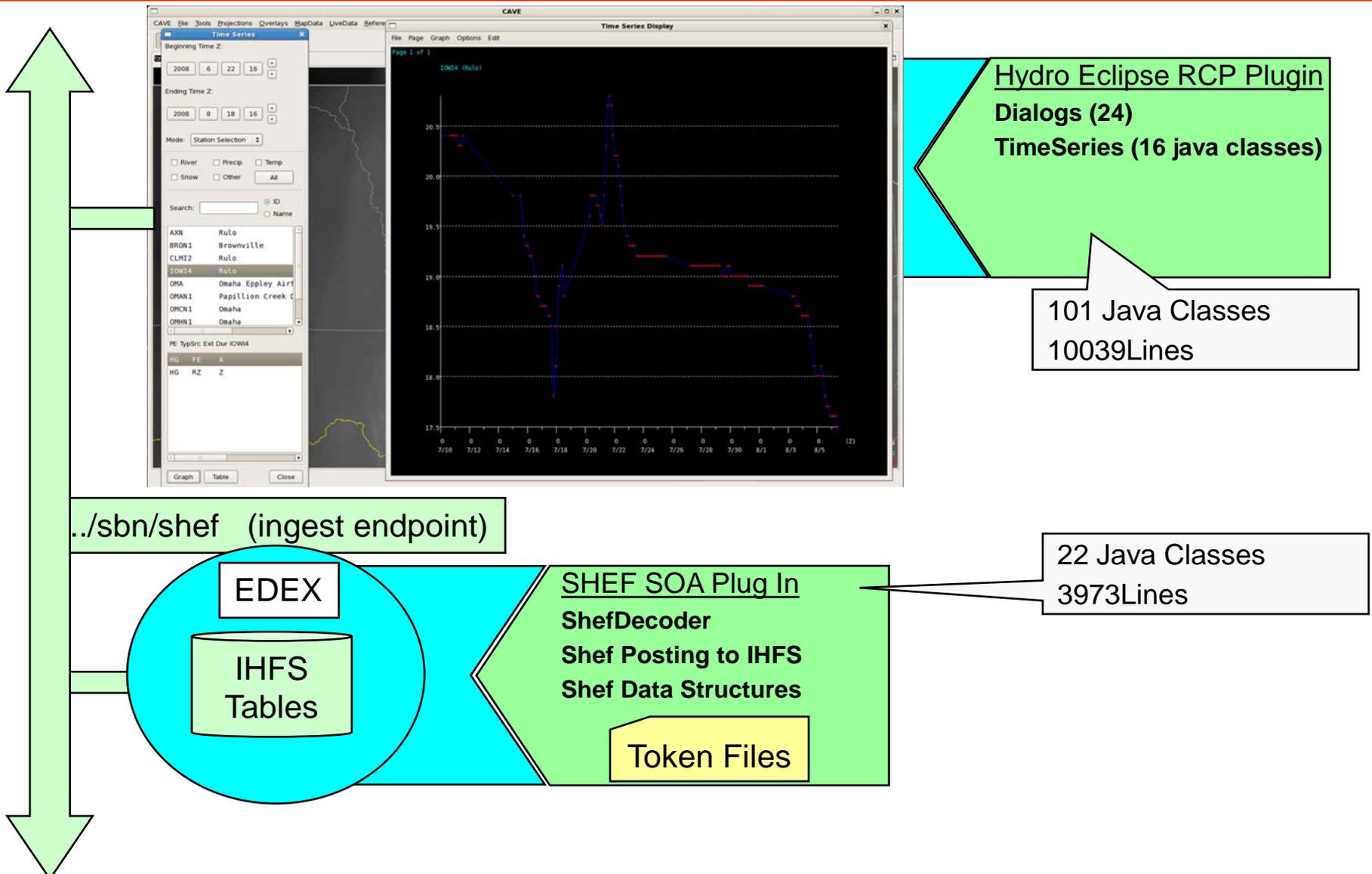
The image shows a workflow for creating a derived parameter pattern. On the left, a Python script editor displays the code for a script named 'DpD.py'. A yellow callout box points to the variable definitions: 'Python Variables Set: Id, Name, Units'. The code includes imports for numpy and functions, variable definitions for 'DpD', and a class 'DerivedParameter' with an 'execute1' method. A yellow callout box points to the 'execute1' method: 'Core MetLib FORTRAN ported to py'. On the right, a file browser shows a directory structure with a 'functions' folder highlighted in green. A green arrow points from the file browser to the script editor, and another green arrow points from the script editor to the file browser. A large green arrow at the bottom points from the file browser to the script editor, labeled 'Derived Parameters Python Scripts'. The background of the script editor shows a contour plot of a map with green contour lines and a red boundary line.

```
#####  
# -----  
# Calculate D  
# Humidity(0 t  
# -----  
#####  
from numpy import  
from numpy import  
from numpy import  
from functions.na import nan_filled  
from functions.na import nan_greater  
from functions.co stants import bad  
from functions.co stants import ourNaN  
import DpT  
  
variableId = "DpD"  
variableName = "Dew point depression"  
variableUnit = "Celsius"  
  
parameters1 = "|T|RH"  
  
class DerivedParameter():  
  
    ##  
    # Calculate dewpoint depression (in degrees  
    # (degrees K) and relative humidity(0 to 100  
    # This function can operate on numpy arrays  
    #  
    # @param T: Temperature in degrees K  
    # @param RH: relative humidity from 0 to 100  
    # @return: dewpoint depression in degrees K  
    # @rtype: numpy array of Python floats or Python float  
    def execute1(self, T, RH):  
        "Calculate dewpoint depression(K) from Temperature(K) and \  
        relative humidity(0 to 100)"  
        DpTCalc = DpT.DerivedParameter()  
        val = DpTCalc.execute1(T,RH)  
        DpD = T - val  
        DpD = nan_filled(DpD, ourNaN)  
        return DpD
```

T09 Hydro Advanced Development

End to End Thread from SHEF ingest to TimeSeries

Hydro



T09 Demo Scenario

Features that illustrate the architecture

- D2D Features
 - Wind Streamlines (Zoom and Pan)
 - Redbook Graphics
 - Derived Parameters (Adding {i.e. DpD.py}, modifying)
- GFE Features
 - GridManager (Show eclipse “ViewPart” features)
 - Running Smart Tools and editing, interpolation
 - Formatter Launcher (NOW product), Make Hazards
- Hydro Perspective
 - Timeseries from ingested SHEF data