

Semi-Automatic Regression Testing With IPAT-S

by

Eric Kemp-Benedict

© 2004 Eric Kemp-Benedict

The layout for this document was designed by Mirto Silvio Busico
(m.busico@ieee.org) as an OpenOffice.org Writer template.

Index

1 Starting Up.....	2
1.1 Introduction.....	2
1.2 A Testing Framework in a Nutshell.....	2
2 An Example.....	4
2.1 Introduction.....	4
2.2 The Example, Step-by-Step.....	4
2.3 Notes on the Procedure.....	7

1 Starting Up...

1.1 Introduction

IPAT-S is a scripting language (a special-purpose computer programming language) for quantitative scenario analysis. Most scenario modeling projects evolve in an iterative, incremental fashion. In such projects, a basic requirement is that any functionality added in one iteration should not break the functionality introduced in previous iterations. This feature is shared by software development projects, and a technique from software development, the use of *regression tests*, can be applied to scenario modeling as well. Regression tests compare key outputs from the software to an archived set of outputs from previous runs. Regression tests can make progress much faster and reduce the level of stress and uncertainty on the modeling team.

The main output from a scenario model is a set of *indicators* – quantitative outputs that change over time. While there are different ways to structure a modeling iteration, in most projects each iteration will produce a set of indicators that have been cleared by relevant stakeholders. These indicators should not change in future iterations, and the use of regression tests can ensure this.

Creating a regression testing framework is very easy to implement in IPAT-S, and this document explains how to do it. It is called “semi-automatic regression testing” because the procedure as described requires some steps on the modeler’s part. With some extra work it could be fully automated. Readers interested in applying the technique can download the free and open-source IPAT-S software and documentation from the IPAT-S web site, <http://ipat-s.kb-creative.net>.

1.2 A Testing Framework in a Nutshell

Section 2 gives a full example of implementing a simple regression framework. Here, a skeleton framework is outlined.

The skeleton framework uses a batch file and a freely-available program called “diff”. The `diff` program shows the differences between two text files, and can be downloaded for free in a package called “UnxUtils.”¹ First download and install UnxUtils,² and create the following one-line batch file, which should be placed in the same folder as the script you are testing.

```
regress.bat:  
  
diff %1 %2 > %3
```

1 A good source for UnxUtils is the GNU Win II site, at <http://gnuwin.epfl.ch/apps/unxutils/en/>. This version has a nice Windows installer that adds the utilities to the path. Alternatively, you can go to the UnxUtils site itself, <http://unxutils.sourceforge.net/>. In this case you will need to modify the path yourself.

2 A good thing for a modeler to have in any case! In addition to `diff`, the distribution includes the powerful `gawk` and `sed` utilities, which can be used to transform text files from one format to another.

The main components in a testing framework are given in the following script fragment, which uses the “regress.bat” batch file. In the script fragment, missing code is indicated by “...” Elements of the framework are in bold.

```
# Year definitions
...
# Dimension definitions
...

## Switches for regression testing
logical makeTest = false # Flag to create the archived outputs
logical runTest = false # Flag to compare outputs to the archive

# Body of script
...
...

## Applying the regression tests
if makeTest then
  clear 'archive.txt'
  set output 'archive.txt'
endif

if runTest then
  clear 'current.txt'
  set output 'current.txt'
endif

if makeTest .or. runTest then
  # List of report statements
  ... Add indicators here ...
  reset output
endif

if runTest then
  wait "regress.bat" "archive.txt current.txt regressout.txt"
  start "notepad.exe" "regressout.txt"
endif
```

With this framework, when `makeTest` is set to `true`, the archived set of outputs is generated. When `runTest` is set to `true`, the current outputs are compared to the archive.

The result of running `diff` is sent to the file “regressout.txt,” which is then shown in Notepad.exe. If the file is blank, then the archived and current outputs are identical, and the script passes the test. If it is not blank, it will show how the two files differ, giving a clue to the modeler where to edit the script.

Over time, as the script is developed, indicators can be added to the script on the line marked by “... *Add indicators here* ...” before creating the archive. The script is designed so that the same set of indicators is exported when the archive and the current set are being generated.

The next section shows a complete example.

2 An Example

2.1 Introduction

The testing framework given in Section 1.2 is very general, and can be used with little modification in many situations. In this section it is applied to a simple script, the “LandUseChange.ips” script that is distributed with the IPAT-S installation. To successfully implement this example, you must first install the `diff` utility and create the batch file “regress.bat” given in Section 1.2.

2.2 The Example, Step-by-Step

1. Open IPAT Studio (the IDE distributed with IPAT-S) and browse for a script.



2. Search for the script “LandUseChange.ips” in the **samples** folder distributed with IPAT Studio.

```

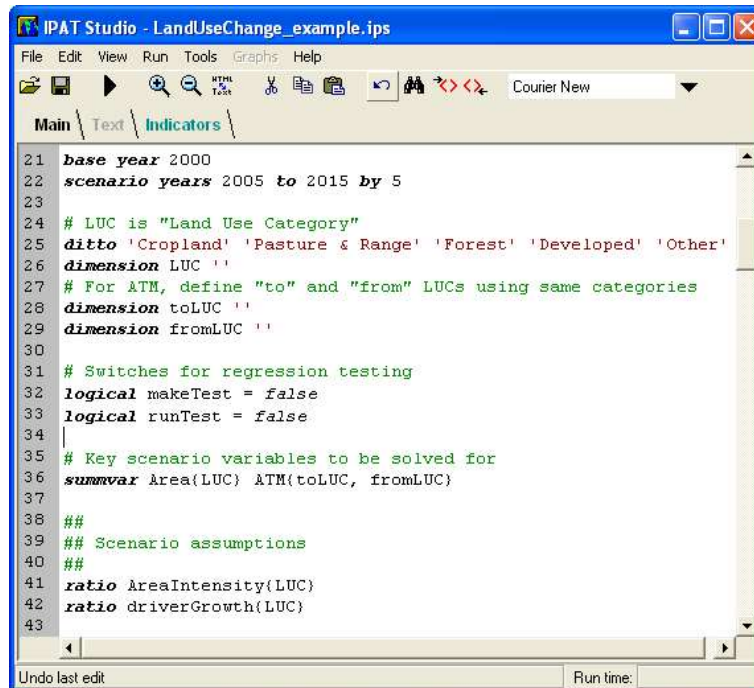
1 |comment:
2 | This script implements land use change in a scenario by calcula
3 | an area transition matrix (ATM). The scenario examines future c
4 | in US land area based on historical patterns and future demogra
5 | It uses the historical area transition matrix for 1982-1997 fro
6 | US National Resource Inventory (1997 NRI, revised Dec. 2000,
7 | http://www.nrcs.usda.gov/technical/NRI/1997/summary_report/tab
8 |
9 | The ATM is solved for in the scenario using a linear goal progr
10 | The scenario ATM is set as close to an historically-based targ
11 | as possible, while meeting the requirements for agricultural an
12 | developed land.
13 |
14 | Because a transition matrix approach is used, rather than net c
15 | in land area, it is possible to keep track of the loss of exist
16 | forest as well as additions of new forest land separately. This
17 | useful because it can take several decades for new forest to ac
18 | the same biomass density of standing forest.
19 | :comment
20 |
21 | base year 2000
22 | scenario years 2005 to 2015 by 5
23 |

```

3. Go to “File | Save As” in IPAT Studio and save the file under another name, such as “LandUseChange_example.ips”.
4. Scroll to line 30, just after the dimension declarations, and insert the lines

```
# Switches for regression testing
logical makeTest = false
logical runTest = false
```

The file should now look like



```
IPAT Studio - LandUseChange_example.ips
File Edit View Run Tools Graphs Help
Main \ Text \ Indicators \
21 base year 2000
22 scenario years 2005 to 2015 by 5
23
24 # LUC is "Land Use Category"
25 ditto 'Cropland' 'Pasture & Range' 'Forest' 'Developed' 'Other'
26 dimension LUC ''
27 # For ATM, define "to" and "from" LUCs using same categories
28 dimension toLUC ''
29 dimension fromLUC ''
30
31 # Switches for regression testing
32 logical makeTest = false
33 logical runTest = false
34
35 # Key scenario variables to be solved for
36 sumvar Area(LUC) ATM(toLUC, fromLUC)
37
38 ##
39 ## Scenario assumptions
40 ##
41 ratio AreaIntensity(LUC)
42 ratio driverGrowth(LUC)
43
Undo last edit Run time:
```

5. Scroll to the end of the script and insert the lines

```
## Applying the regression tests
if makeTest then
  clear 'archive.txt'
  set output 'archive.txt'
endif

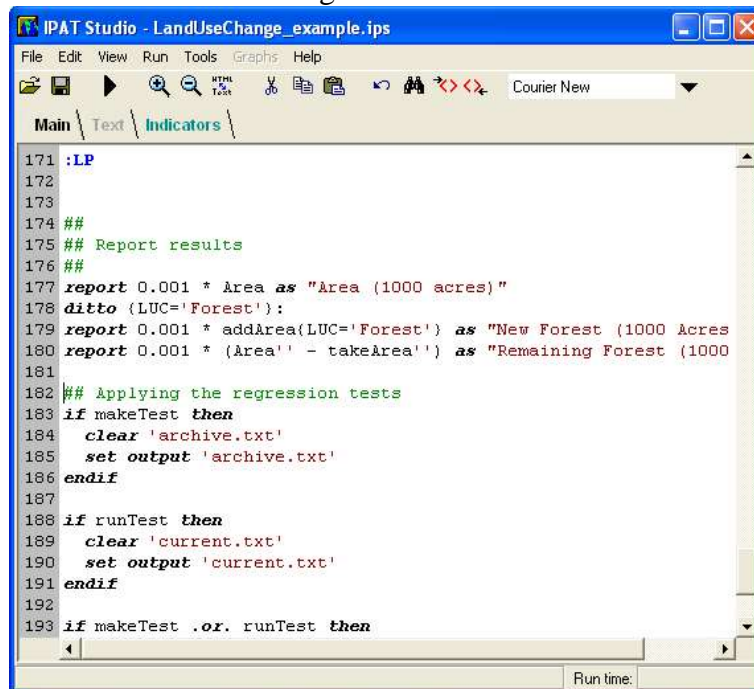
if runTest then
  clear 'current.txt'
  set output 'current.txt'
endif

if makeTest .or. runTest then
  report Area as "Area (acres)"
  reset output
endif

if runTest then
  wait "regress.bat" "archive.txt current.txt regressout.txt"
```

```
start "notepad.exe" "regressout.txt"
endif
```

The script should now look something like



```
IPAT Studio - LandUseChange_example.ips
File Edit View Run Tools Graphs Help
Main | Text | Indicators |
171 :LP
172
173
174 ##
175 ## Report results
176 ##
177 report 0.001 * Area as "Area (1000 acres)"
178 ditto (LUC='Forest'):
179 report 0.001 * addArea(LUC='Forest') as "New Forest (1000 Acres"
180 report 0.001 * (Area' - takeArea') as "Remaining Forest (1000"
181
182 ## Applying the regression tests
183 if makeTest then
184 clear 'archive.txt'
185 set output 'archive.txt'
186 endif
187
188 if runTest then
189 clear 'current.txt'
190 set output 'current.txt'
191 endif
192
193 if makeTest .or. runTest then
```

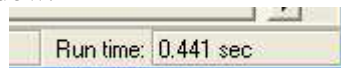
Note that the inserted lines are nearly identical to the lines in the testing framework in Section 1.2. The only difference is that for this script a specific indicator has been chosen – the land area. The relevant line should be on or close to line 195 in IPAT Studio.

6. Create the archived test results:

- Scroll to line 32 and change `makeTest` from *false* to *true*.
- Press the “Run Special” button or press F9 to run the script.



Note: The script will not produce any output, but “Run Time” will be updated in the lower right of the program window.



- Return `makeTest` to its original value (that is, change it from *true* to *false*).

7. Run the test with no changes to make sure there are no errors:

- Scroll to line 33 and change `runTest` from *false* to *true*.
- Press the “Run Special” button or F9 to run the script. A blank Notepad window should open.³ This indicates that the test values are identical to the archived values.

³ If instead you get an error saying that the batch file could not be found, make sure that it is in the same folder as the script. If you get an error saying that `diff.exe` could not be found, then install the `diff` utility.

8. Change the script and then run the test to see how a failed regression test is reported.

- Scroll to line 45 and change the `AreaIntensity` growth rate for cropland from -0.7% to -0.8% .

```

42 ratio driverGrowth(LUC)
43
44 # Three LUCs are set exogenously
45 AreaIntensity(LUC='Cropland') = gr(<-0.8>/100) #SLIDER -group "i
46 AreaIntensity(LUC='Pasture & Range') = gr(<-0.3>/100) #SLIDER -i
47 AreaIntensity(LUC='Developed') = gr(<1.0>/100) #SLIDER -group "i
48

```

- Press the “Run Special” button or F9 to run the script. Notepad should open, with a file looking something like

```

regressout.txt - Notepad
File Edit Format View Help
2c2
< "Area (acres)", "Cropland", ,, 376998, 363986, 351424, 339295
-----
> "Area (acres)", "Cropland", ,, 376998, 362157, 347901, 334206
4c4
< "Area (acres)", "Forest", ,, 406955, 416242, 424357, 431087
-----
> "Area (acres)", "Forest", ,, 406955, 418071, 427880, 436176

```

This file indicates that the test that was just run differs from the archived file. Also, it indicates where the differences appear – in the areas of cropland and forest.

- Return `runTest` to its original value (that is, change it from *true* to *false*).

That completes the regression test example.

2.3 Notes on the Procedure

It is easy to add regression tests to any IPAT-S script using the testing framework in this document. In a project of any complexity, it is a good idea to create a regression test and apply it whenever the model is modified. It removes a lot of uncertainty from the modelers and makes it possible to move forward with confidence.