Inca Workshop
August 26, 2010

Hands-on Tutorial #1
Reporter API and Repository

Review reporter APIs:

1) Overview of APIs: http://inca.sdsc.edu/releases/2.6/guide/userguide-one.html#REPORTER-API
2) Perl APIs: http://inca.sdsc.edu/releases/2.6/repdocs/perl.html
3) Python APIs: http://inca.sdsc.edu/releases/2.6/repdocs/python.html

Download a default set of reporters:

1) Get repository and untar it
```
% wget http://inca.sdsc.edu/releases/2.6/Inca-Reporter.tar.gz
% tar -zxvf Inca-Reporter.tar.gz
% cd Inca-Reporter-5.13398/bin
```
2) Set environment variables to include reporter libs
```
% setenv PERL5LIB ../lib/perl; setenv PYTHONPATH ../lib/python
```
3) Execute a reporter. Try running with -help, -version, and -log to see how the output changes.
```
% ./cluster.compiler.gcc.version
```

Write some reporters:

1) A base Reporter that reports the path of the user's shell ("$ENV{SHELL}" contains this) using the following <body> format:
```
<body>
  <shell>
    <ID>shell</ID>
    <path>/bin/bash</path>
  </shell>
</body>
```

2) A Version Reporter that determines the version of tcsh ("tcsh –c 'echo $version' " reports this), failing if tcsh is unavailable

3) A SimpleUnit Reporter that determines whether "finger" command can provide the user's name.

4) A Performance Reporter that reports system page swap activity over a specified interval (the program sar reports this; on some systems the command is "sar –g *interval* 1"; on others it's "sar –B *interval* 1"). Allow the user to specify the interval via the --interval

command-line option, with 30 seconds the default. Try using the "reportBodyShort" method to change the reporter body.

5) (Extra credit) A Version Reporter that extracts the manifests from the Java jar files found in a specified directory, then reports the Implementation-Version attribute found in each manifest as a subpackage version. Require the user to specify the directory to search for .jar files. (You can unpack the manifest from a jar file via "jar xf *jar* META-INF/MANIFEST.MF". Use the reporter's **tempFile** method to remove the META-INF directory when the reporter exits.)

6) Create a repository catalog for your reporters by running incpack in the directory where you created your reporters.

Answer Key

1) A base Reporter that reports the path of the user's shell

```
use Inca::Reporter;
my $reporter = new Inca::Reporter
  (version => 1, description => 'Reports the SHELL path');
$reporter->processArgv(@ARGV);
$reporter->setBody(
  $reporter->xmlElement('shell', 0,
    $reporter->xmlElement('ID', 1, 'shell'),
    $reporter->xmlElement('path', 1, $ENV{SHELL})
  )
);
$reporter->setResult(1);
$reporter->print();
```

2) A Version Reporter that determines the version of tcsh

```
use Inca::Reporter::Version;
my $reporter = new Inca::Reporter::Version(
  version => 1,
  description => 'Reports the version of tcsh',
  package_name => 'tcsh'
);
$reporter->processArgv(@ARGV);
$reporter->setVersionByExecutable
  ('tcsh -c \'echo $version\'', '^tcsh ([\d\.]+)');
$reporter->print();
```

3) A SimpleUnit Reporter that determines whether "finger" command can provide the user's name

```
use Inca::Reporter::SimpleUnit;
my $reporter = new Inca::Reporter::SimpleUnit(
  version => 1,
  description => 'Reports whether finger provides a name',
  unit_name => 'finger'
);
$reporter->processArgv(@ARGV);
my $output =
  $reporter->loggedCommand("finger $ENV{USER}");
if(!defined($output)) {
  $reporter->unitFailure("finger command failed: $!");
} elsif($? || $output !~ /Name:\s*.*/) {
  $reporter->unitFailure(0, "finger command failed: $output");
} else {
  $reporter->unitSuccess();
}
$reporter->print();
```

4) A Performance Reporter that reports system page swap activity over a specified interval

```
use Inca::Reporter::Performance;
my $reporter = new Inca::Reporter::Performance(
  version => 1,
  description => 'Reports page swap activity',
  measurement_name => 'swap'
);
$reporter->addArg
  ('interval', 'interval over which to measure', 30, '\d+');
$reporter->processArgv(@ARGV);
my $interval = $reporter->argValue('interval');
my $output = $reporter->loggedCommand("sar -g $interval 1");
if($? != 0) {
  $output = $reporter->loggedCommand("sar -B $interval 1");
}
if(!defined($output)) {
  $reporter->setResult(0, "sar command failed: $!");
} elsif($? ||
        $output !~ /^Average:\s*([\d\.]+)(\s*([\d\.]+))?/m) {
  $reporter->setResult(0, "sar command failed: $output");
} else {
  my $pgIn = defined($3) ? $1 : undef;
  my $pgOut = defined($3) ? $3 : $1;
  my $benchmark = $reporter->addNewBenchmark('swap');
  $benchmark->setParameter('interval', $interval);
  $benchmark->setStatistic('pagesout', $pgOut);
  $benchmark->setStatistic('pagesin', $pgIn) if defined($pgIn);
  $reporter->setResult(1);
}
$reporter->print();
```

5) A Version reporter that extracts the manifests from the Java jar files found in a specified directory, then reports the Implementation-Version attribute found in each manifest as a subpackage version

```
use Inca::Reporter::Version;
my $reporter = new Inca::Reporter::Version
  (version => 1, description => 'Reports jar file versions');
$reporter->addArg('dir', 'directory path');
$reporter->processArgv(@ARGV);
my $dir = $reporter->argValue('dir');
my $manifest = 'META-INF/MANIFEST.MF';
$reporter->tempFile('META-INF');
foreach my $jar(glob "$dir/*.jar") {
  $reporter->loggedCommand("jar xf $jar $manifest");
  next if $? != 0;
  my $output = $reporter->loggedCommand
    ("grep 'Implementation-Version:' $manifest");
```

```
  next if !defined($output) ||
          $output !~ /Implementation-Version:\s*(\S+)/;
  my $version = $1;
  my ($subpackage) = $jar =~ m#.*/(.*)#;
  $reporter->setSubpackageVersion($subpackage, $version);
}
$reporter->setResult(1);
$reporter->print();
```

6) From the Inca-Reporter-5.13398 directory:

```
% sbin/incpack bin/* -I lib/perl -I lib/python
```