

Replicode tutorial: The basics

Helgi Pall Helgason
CADIA / Reykjavik University

in this tutorial...

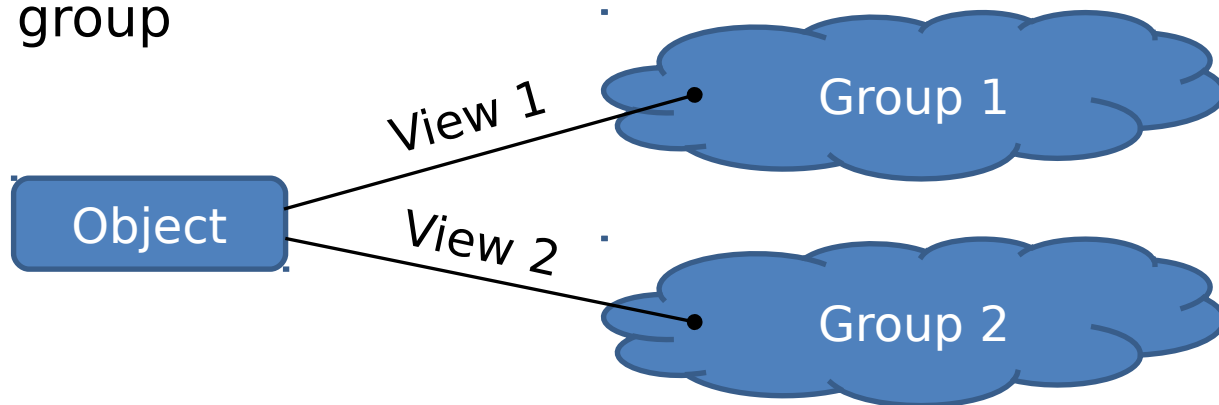
- Replicode at a glance
- How to download and compile the Replicode source code
- Getting started

Replicode at a glance

- Replicode consists of:
 - Programming language
 - Executive (handles runtime operation)
 - Development tools (compiler, decompiler, etc.)
- Designed to implement systems based on numerous short programs that run in parallel
- Replicode systems are *dynamic production systems*
 - Each program can be viewed as a “rule” that performs some action when the condition it specifies are met
 - New programs can be created at run-time
- Time is handled in an explicit fashion

Groups

- Replicode uses workspaces called **groups**
 - The three standard groups are: root, stdin and stdout
 - Groups control visibility of program/data
 - Each group has it's own set of control parameters
- A single object (e.g. program) can exist in multiple groups
- **Views** are a set of parameters that associate an object with a group



Programs

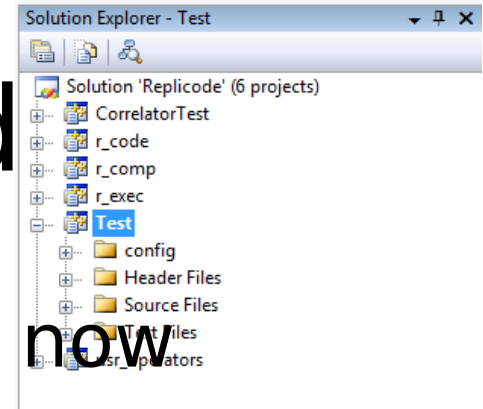
- Replicode programs are reactive and data-driven
 - Execute only when they find the input they want (except in special cases)
- For a Replicode program to run, it must have an **activation value** greater or equal than the **activation threshold** of its group
- Any Replicode object must have a **saliency value** greater or equal to the **saliency threshold** of the group to be an input for a program
- Replicode programs are first defined and then instantiated
 - Similar to traditional OO programming (class/instance)
 - The instantiation may require parameters (template arguments)

Downloading and compiling

- Currently, the Replicode source code can be downloaded from:
 - <http://cadia.ru.is/svn/repos/replicode/>
- An additional project, CoreLibrary, is also required:
 - <http://cadia.ru.is/svn/repos/CoreLibrary/>
 - Place on same directory level as Replicode
- Microsoft Visual Studio should be used to compile the source code
 - Visual Studio 2008 recommended
- Compilation warnings can be ignored

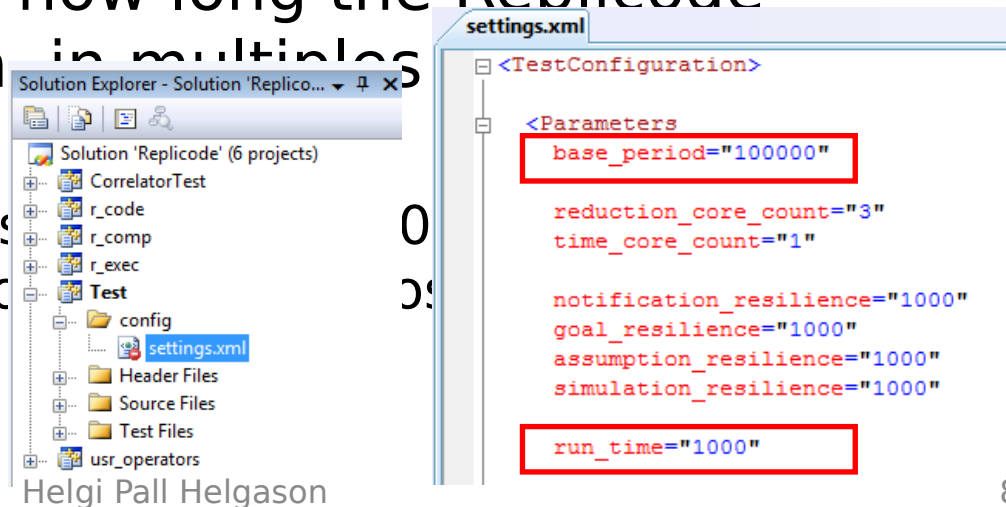
Getting started

- The **Test** project is the focus for now
- It contains the code for a program that:
 - Starts a Replicode executive
 - Loads and compiles a specified Replicode source code file
 - Runs the loaded code on the Replicode executive for a specified amount of time
 - Decompiles and prints out the resulting image (state of the system)



The Test project

- The duration of execution is specified in the configuration file
- **base_period** is a basic unit of time in nanoseconds
 - Duration is specified as a multiple of this value in many cases
- **run_time** controls how long the Replicode executive shall run in multiples of base period
 - If the base_period is 1.000 equals 1 second



the **Test** project

- The configuration file also specifies which Replicode source code file to load

```
settings.xml*
goal_resilience="1000"
assumption_resilience="1000"
simulation_resilience="1000"

run_time="1000"

probe_level="2"

usr_operator_path="./usr_operators.dll"
usr_class_path="./Test/user.classes.replicode"
source_file_name="./Test/test.3.replicode"
```

- Note: It may depend on your code for how long you wish to run the executive.
- Remember: The output of your programs is only shown **after** the executive has completed its operation.

In conclusion

- You should now have a compiled version of Replicode
- To run one of the examples, simply point the **source_file_name** parameter in settings.xml to the desired example source code file and run the Test project
- Following tutorials will explain the displayed output
- Additional material:
 - Replicode specification (v1.1)
 - Slides from Eric Nivel's Replicode presentation given at Reykjavik University (September 2010)