



CADSWES

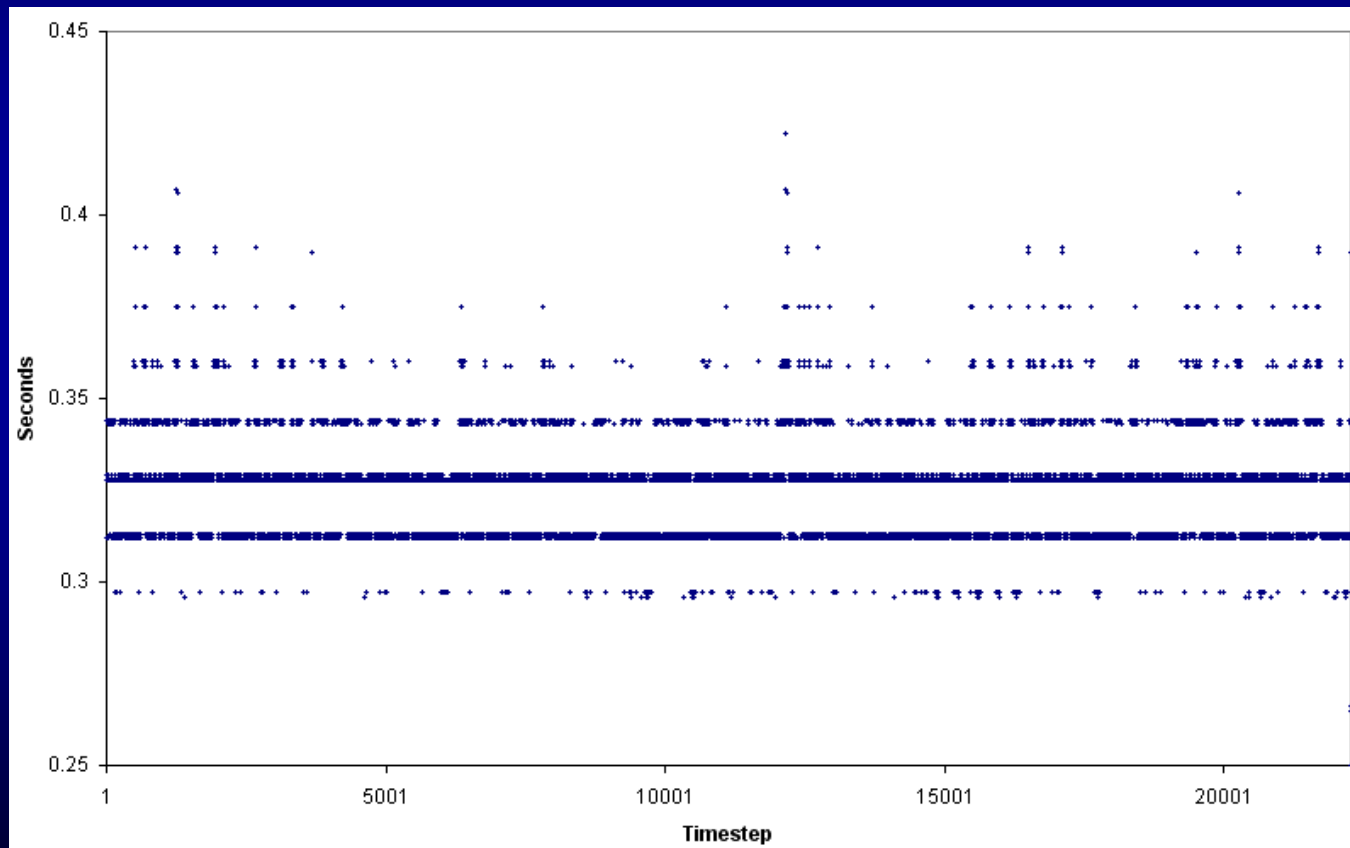
University of Colorado

Center for Advanced Decision Support for Water and Environmental Systems

Improving Performance

RiverWare User Group Meeting
August 13-14, 2008

Time per timestep



Decreasing run times: what you can do

- Turn off diagnostics
- Close unused windows, other applications
- Check and configure system paging
- RBS: Analyze and improve your policy
- Run in batch mode
- Divide the model into multiple runs
- Buy hardware

Improving the run time of a RBS policy

➤ Step 1: Analysis

- RPL set analysis tool
- RunTime() function

➤ Step 2: Modify policy

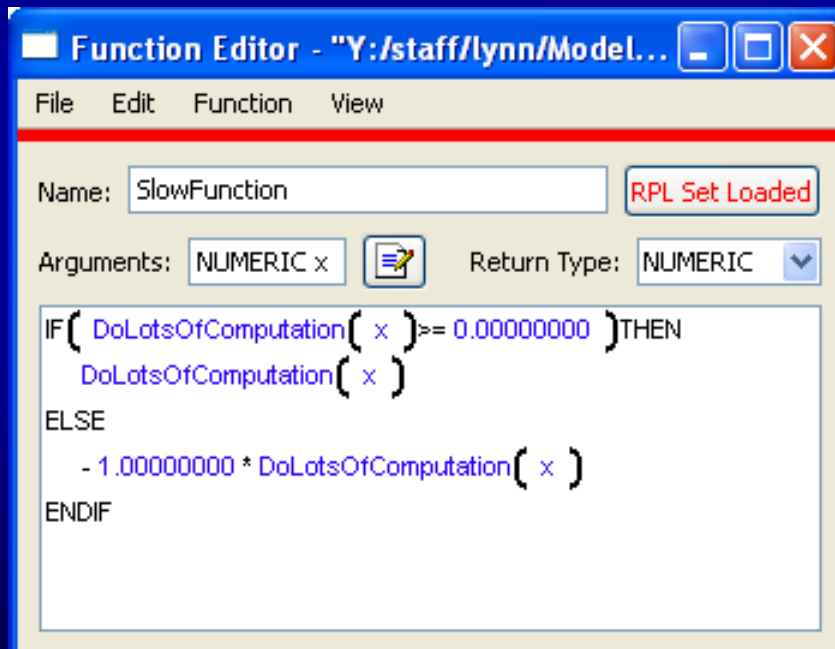
- Focus on bottlenecks
- Don't sacrifice clarity

➤ Repeat as necessary

How to create efficient RPL expressions

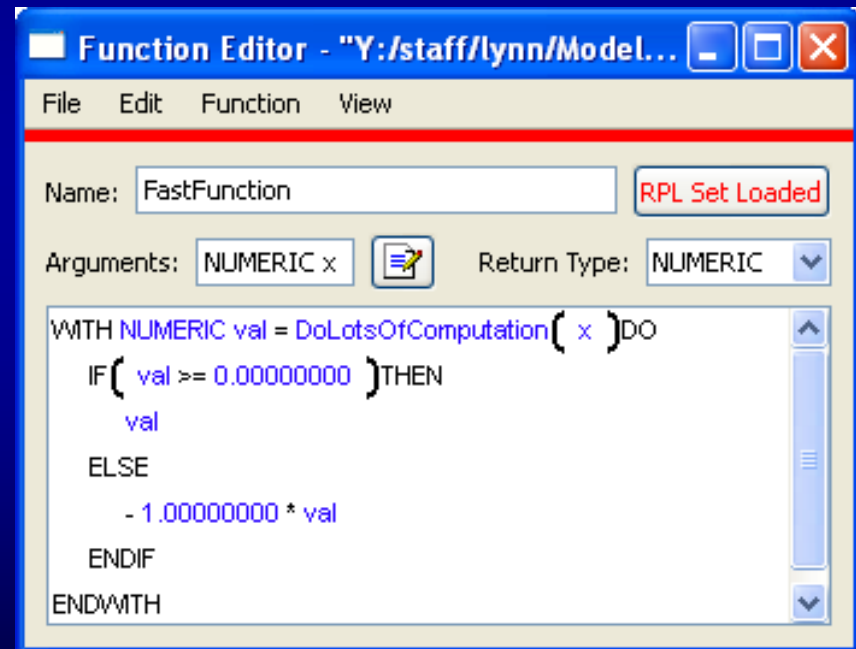
- Prefer predefined functions
- Use functional decomposition
- Use WITH expressions
- Minimize LIST and STRING processing
- Use OBJECT and SLOT values
- Use functions with no arguments
- Use time invariant functions

Effective use of WITH expressions



The screenshot shows a window titled "Function Editor - 'Y:/staff/lynn/Model...". The menu bar includes "File", "Edit", "Function", and "View". The "Name" field contains "SlowFunction" and there is a red "RPL Set Loaded" button. The "Arguments" field is "NUMERIC x" and the "Return Type" is "NUMERIC". The code in the editor is:

```
IF ( DoLotsOfComputation ( x ) >= 0.00000000 ) THEN
  DoLotsOfComputation ( x )
ELSE
  - 1.00000000 * DoLotsOfComputation ( x )
ENDIF
```



The screenshot shows a window titled "Function Editor - 'Y:/staff/lynn/Model...". The menu bar includes "File", "Edit", "Function", and "View". The "Name" field contains "FastFunction" and there is a red "RPL Set Loaded" button. The "Arguments" field is "NUMERIC x" and the "Return Type" is "NUMERIC". The code in the editor is:

```
WITH NUMERIC val = DoLotsOfComputation ( x ) DO
  IF ( val >= 0.00000000 ) THEN
    val
  ELSE
    - 1.00000000 * val
  ENDIF
ENDWITH
```

Decreasing run times: what CADSWES can do

- Focused performance studies
 - Arkansas River (35%)
 - URGWOM (20%)
 - Tennessee Valley (95%)
- Create new RPL predefined functions
- Redesign software components

Re-implementation of Series Slots

- Purpose: to reduce the memory needs of large models.

Original Implementation (4.9 and earlier releases)

Series Slot

Series Slot meta-data
(units, display precision,
linked-slots list, ...)

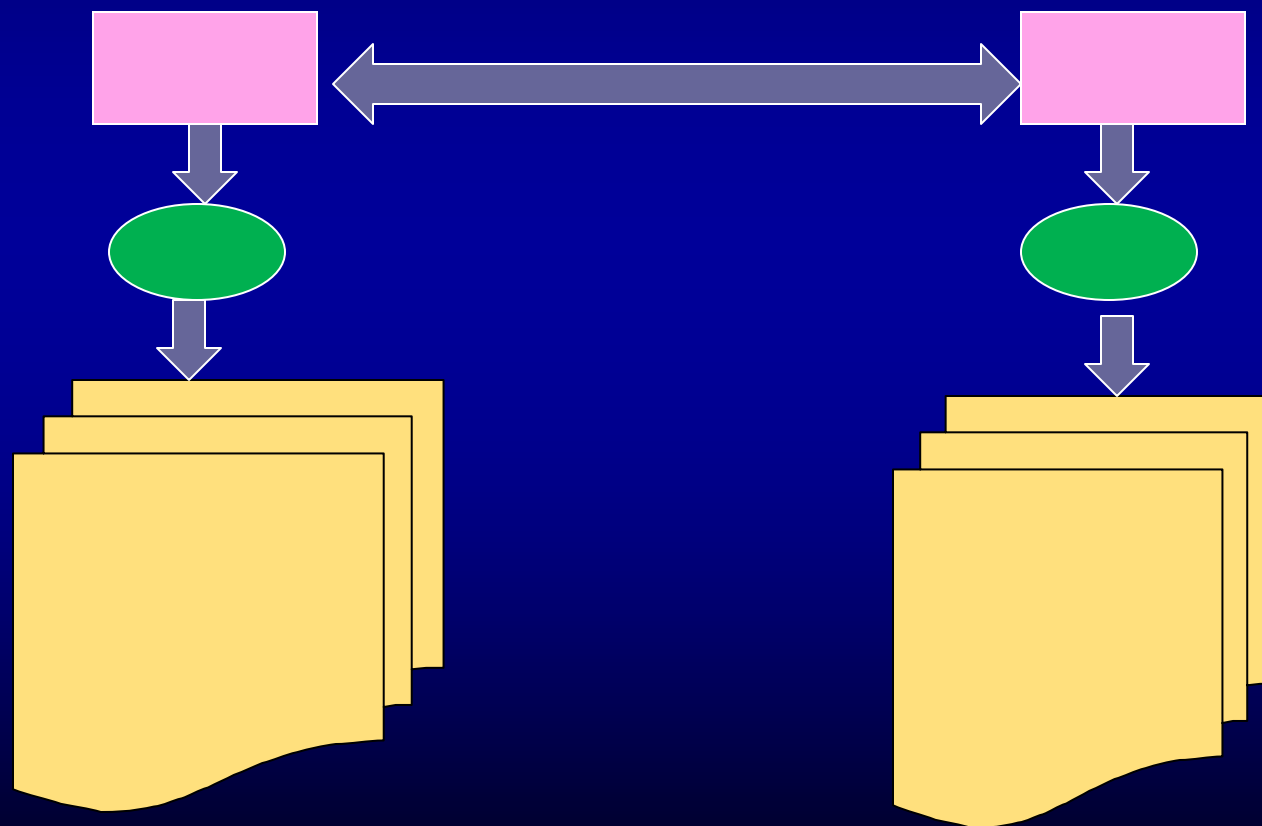
End date
Start date

Series < series struct >

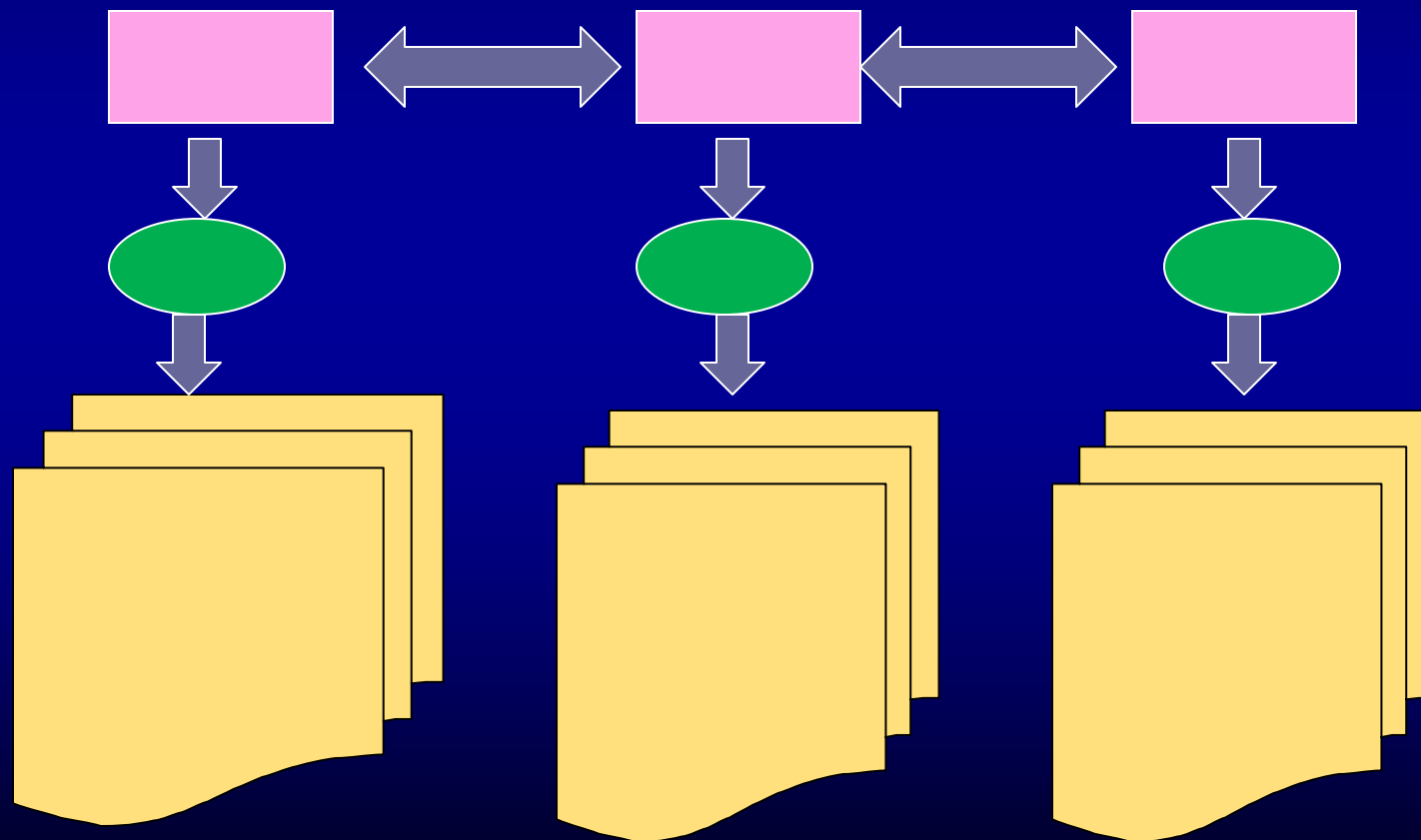
Series meta-data
(start date, end date,
timestep size, ...)

1: Value
2: Iteration count
3: Flags (input/rule/output/
method/account/
surcharge release/ ...)
4: Priority (for rules runs)
5: Set-by (for detecting
overdetermination)

Linked Simulation Series Slots



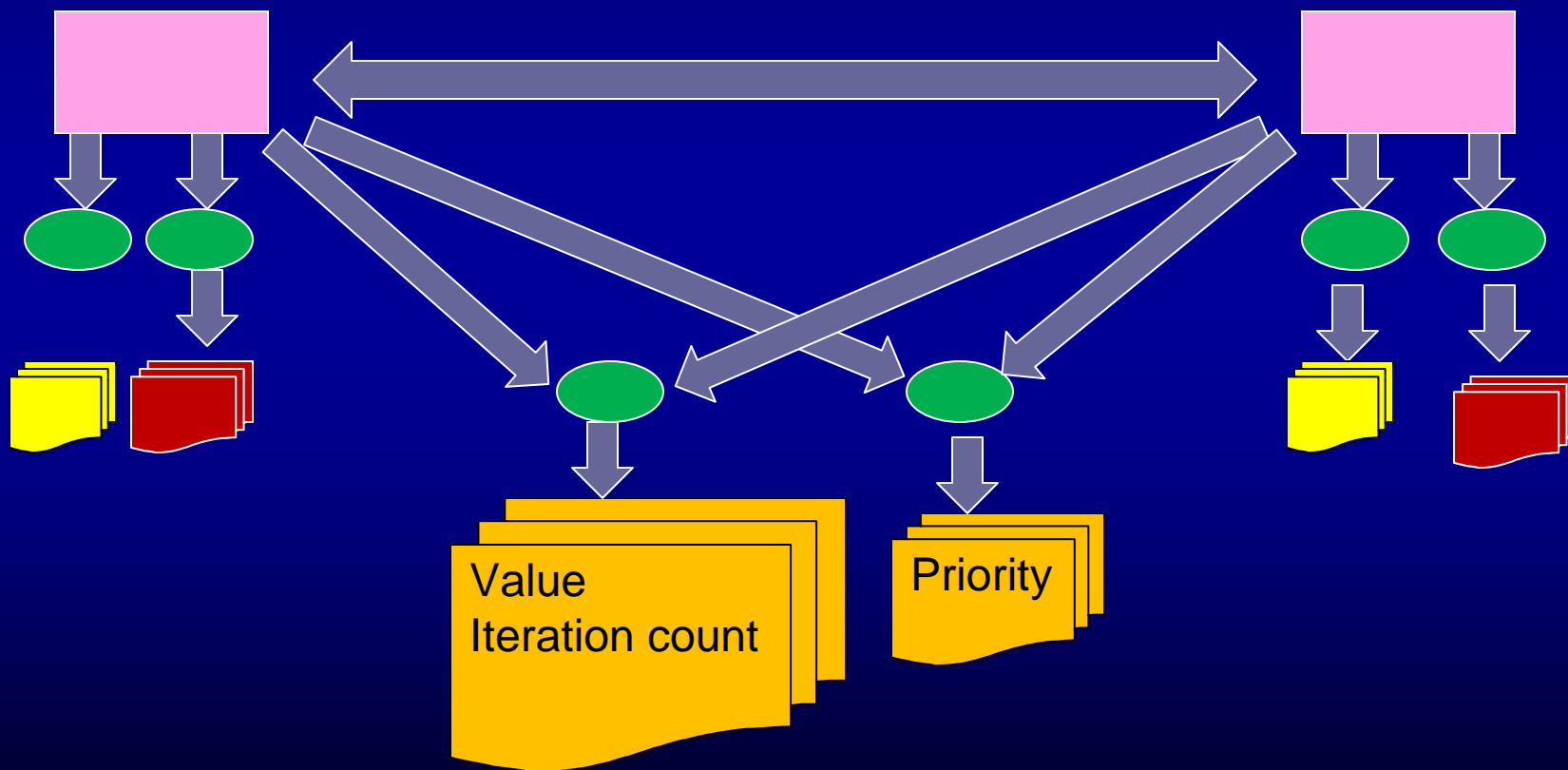
Linked Accounting Series Slots



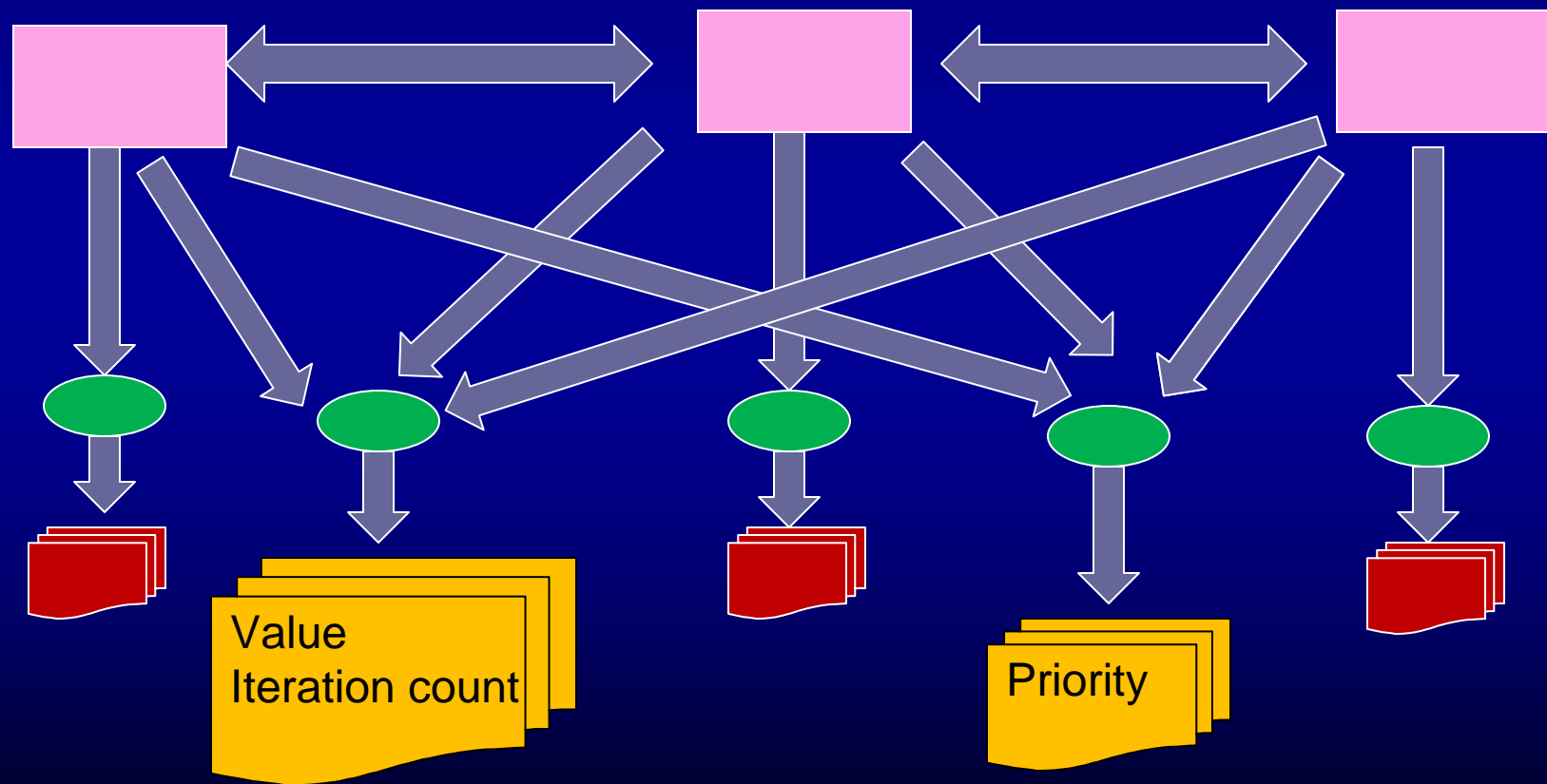
Data & Meta-data

- Can be shared between linked slots:
 - Value
 - Iteration Count
 - Priority
- Private – cannot be shared:
 - Set-by information (to detect over-determination), not needed for account slots
 - Flags (e.g. Method vs Propagated)

Linked Slots on Simulation Objects: 2-way sharing



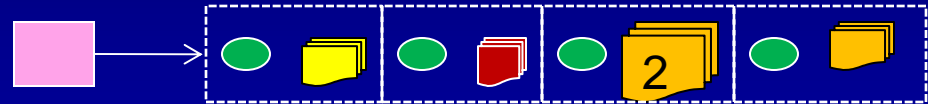
Linked Slots on Accounts: 3-way sharing



Series Data & Meta-data

➤ Simulation Object Slots

• Linkable



- Set-by Flags {Value, Iteration-count } Priority

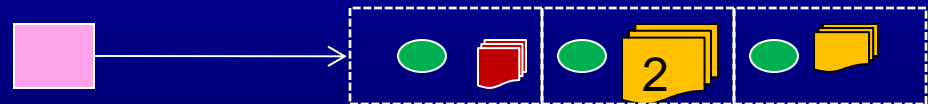
• Not Linkable



- { Set-by Flags Value Iteration-count Priority }

➤ Account Slots

• Linkable



- Flags {Value, Iteration-count } Priority

• Not Linkable



- { Flags Value Iteration-count Priority }

Phased Implementation

- ✓ Accounting-slots-sharing case first (simpler in its details than simulation-slots-sharing case)
- ✓ Eliminate Set-by series on accounting slots
- ✓ Share the sharable series on linked accounting slots
- Insert implementation class between SeriesSlot and Series<>
- Make non-linkable accounting slots use smaller structure (no set-by)
- Apply multiple series to linkable simulation object slots

Observations for work completed so far ... (more to come)

- Process size change (Solaris) +.2% to -27%
- Run time change (Windows) 0% to +11%
(expected space/time trade-off; should be reduced with up-coming work)

New Model File Format

- Old: series { datum, flags }
- New: series { datum } , series { flags }
 - Uses run-length encoding of each series
 - Compression rate is model-dependent:
 - Few-timestep model files may get larger
 - Many-timestep model files get smaller
 - Observed: 29% - 92% reduction in un-gzipped model size
 - 38% - 90% reduction in model-load time, gzipped or not