

Framework Invasiveness:

- A study of how differences between scientific modeling frameworks impacts the quality of the model implementation
 - Olaf David, Wes Lloyd
 - USDA-ARS, USDA-NRCS
- Colorado State University Civil Engineering
Colorado State University Computer Science

Framework Invasiveness

- How does the use of a particular framework impact an application's quality?
- Quality with respect to
 - Maintainability
 - Portability
 - Reusability
 - Understandability (of code)

Framework Invasiveness - 2

- Coupling between application and framework code
 - Use of Framework APIs
 - Use of Framework Interfaces
 - Use of Framework Custom Data Types
- Mass of boilerplate code
 - Mass should account for both Lines of Code (LOC) and source code size in bytes
- External framework dependencies
 - Required non-framework APIs

Framework Invasiveness – 3

- Other framework couplings
 - Language dependencies
 - Platform dependencies
 - For example CCA required Redhat 8
 - Organizational dependency
 - Training cost/investment
 - Hardware cost/investment
 - Support cost/investment

Research Questions

- How does framework invasiveness impact software quality?
 - In order to answer the question we must first quantify invasiveness
- How do we measure framework to application invasiveness?
 - Perform an analysis of models implemented in various frameworks, languages

Invasiveness Properties

- Number of Framework specific data types used/uses
- Number of framework specific functions used/uses (API calls)
- LOC of model implementation
 - Boilerplate LOC of model implementation
 - Application LOC of model implementation
- Boilerplate to Application LOC ratio
- Size of Framework
 - Number of classes/modules/functions
 - Size (LOC)

Model Implementation Investigation

- Thornthwaite Water balance model
 - Models allocation of water among components of hydrological system
- Model
 - FORTRAN Implementation = 244 LOC
 - 8 Modeling Components
 - Climate, Daylen, HamonET, Snow, Soil moisture, Runoff, Output, Controller

Invasiveness Measures

- Source code size (bytes)
- Lines of Code (LOC)
- Framework dependent datatypes used/uses
- Framework specific functions used/uses
- Framework dependent code lines of code (FDLOC)
 - Does not compile if framework removed
 - Must be refactored if framework removed or changed

Model Implementation Investigation - 2

- Model implementation in 4 Languages, 5 Frameworks
 - Produce identical output
 - Same functional decomposition
 - Time step driven by input datafile
- How do the implementations differ?
 - Preliminary Measurements

Source Code Size

- Thornthwaite

* ESMF C implementation uses global data

Implementation	Size (bytes)
Plain C++	12437
OMS Java	21846
NGMF Java	11843
ESMF C*	24002
ESMF Fortran	44473
Plain Fortran	10678
CCA Java (java source only, no infrastructure)	581901
OpenMI Java	40408

Lines of Code

- Thornthwaite

* ESMF C implementation uses global data

Implementation	Size
Plain C++ (.cpp and .h files)	364
OMS Java (java source only, no infrastructure)	444
NGMF Java (java source only, no infrastructure)	240
ESMF C* (.cpp and .h files)	496
CCA Java (java source only, no infrastructure)	9926
Plain Fortran	244
ESMF Fortran	700
OpenMI Java	532

Framework Datatype Usage

- Thornthwaite

* ESMF C implementation uses global data

Implementation	Framework datatypes used	Framework datatype uses
Plain C++ (.cpp and .h files)	0	0
Plain Fortran (.F90 files)	0	0
OMS Java (java source only, no infrastructure)	4	36
NGMF Java (java source only, no infrastructure)	1	1
ESMF C* (.cpp and .h files)	8	30
ESMF Fortran (.F90 files)	4	113
CCA Java (java source only, no infrastructure)	12	91
OpenMI Java	10	137

Framework Specific Functions

- Thornthwaite

* ESMF C implementation uses global data

Implementation	Framework functions used	Framework function uses
Plain C++ (.cpp and .h files)	0	0
Plain Fortran (.F90 files)	0	0
OMS Java (java source only, no infrastructure)	3	4
NGMF Java (java source only, no infrastructure)	4	18
ESMF C* (.cpp and .h files)	9	55
ESMF Fortran (.F90 files)	8	138
CCA Java (java source only, no infrastructure)	?	1480*
OpenMI Java	20	231

Framework Dependent Lines of Code (FDLOC)

- Thornthwaite

* ESMF C implementation uses global data

Implementation	LOC	FDC LOC	% FDC
Plain C++ (.cpp and .h files)	364	0	0.00%
Plain Fortran (.F90 files)	244	0	0.00%
OMS Java (java source only, no infrastructure)	444	324	72.90%
NGMF Java (java source only, no infrastructure)	244	87	35.60%
ESMF C* (.cpp and .h files)	496	229	46.16%
ESMF Fortran (.F90 files)	700	270	39.00%
CCA Java (java source only, no infrastructure)	9926	9716	97.88%
OpenMI Java	532	305	57.00%