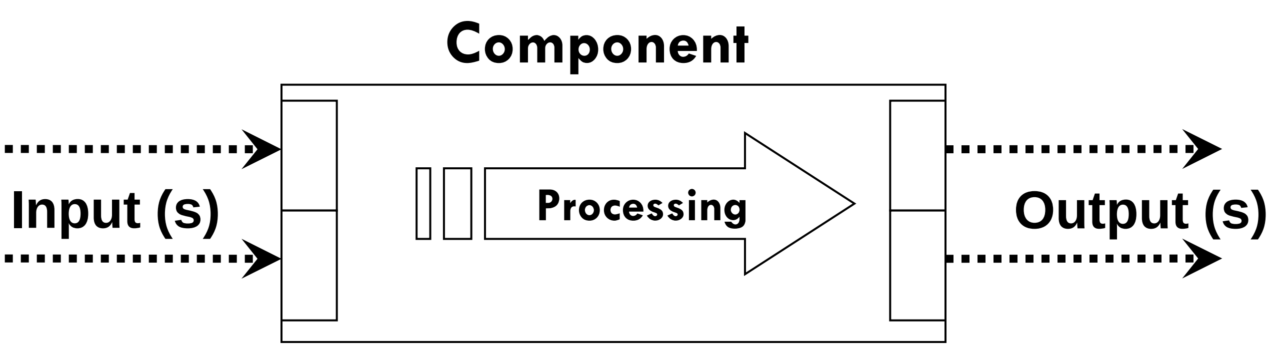
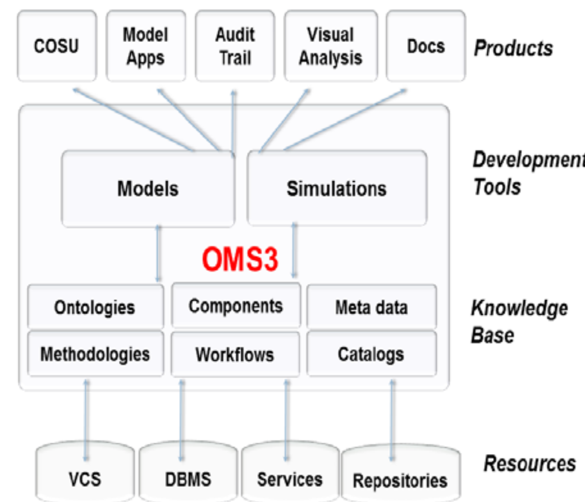
Development of a Combined Wind and Water Erosion Model (WWEM) for the Object Modeling System

J.C. Ascough II[1], D.C. Flanagan[2], C.C. Truman[3], O. David[4]

[1]USDA-ARS, Fort Collins, CO USA; [2]USDA-ARS, West Lafayette, IN USA; [3]USDA-ARS, Tifton, GA, USA; [4]Colorado State Univ., Fort Collins, CO USA

Component library
(erosion processes, statistical analysis, visualization, …)





Initial WWEM Motivation

2004 - NRCS Re-evaluated its Need for Erosion Prediction Technology from the USDA-ARS:

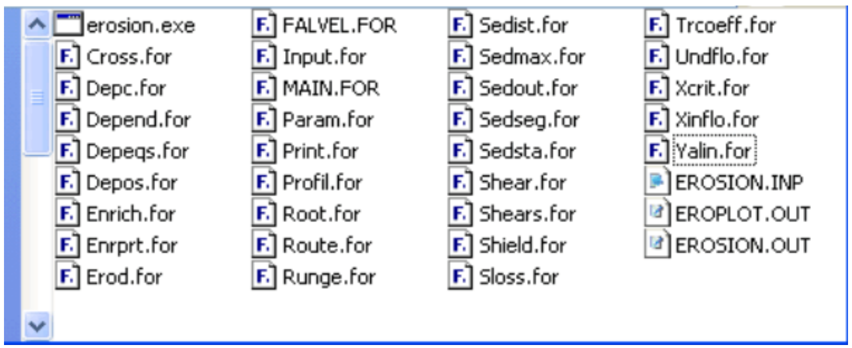
1. Short-term → Delivery of the WEPS model so that NRCS could test and subsequently implement it in their field offices as a replacement for the WEQ.

2. Long-term → "NRCS proposes to collaborate with ARS to build a single process based model to make erosion prediction calculations. NRCS proposes that this model be capable of making rainfall induced rill and interrill erosion computations, as well as computations for wind erosion together or independently of one another. This model would naturally incorporate the technologies currently in WEPS, the Water Erosion Prediction Project (WEPP)…"

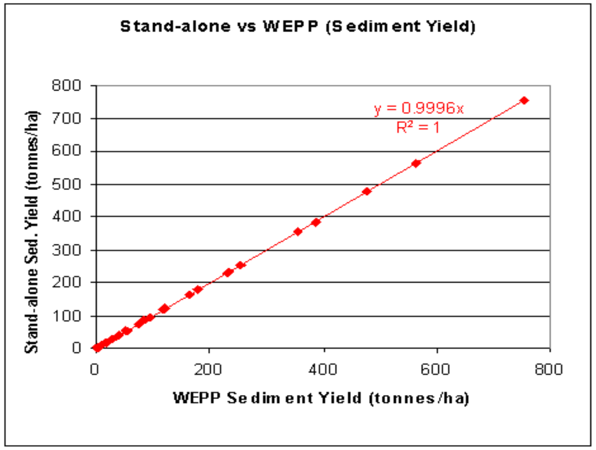
(Letter from L.E. Clark, NRCS Deputy Chief, dated March 1, 2004).

WWEM History: I. Initial Single Event Model Under the Object Modeling System (OMS) Version 1

September 2004 to June 2005

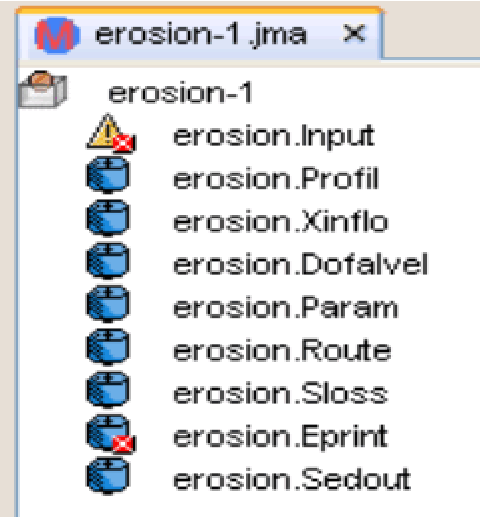


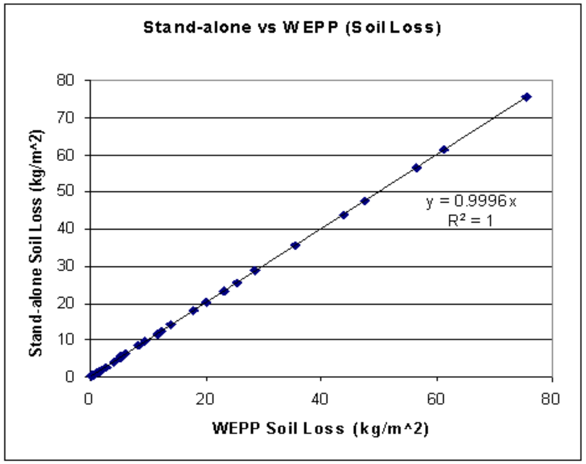
Disaggregation of WEPP v2004.7 hillslope model components → Route.for and underlying subroutines + new Main.for



Comparison of soil loss and sediment yield results for initial erosion stand-alone program and original WEPP v2004.7 model

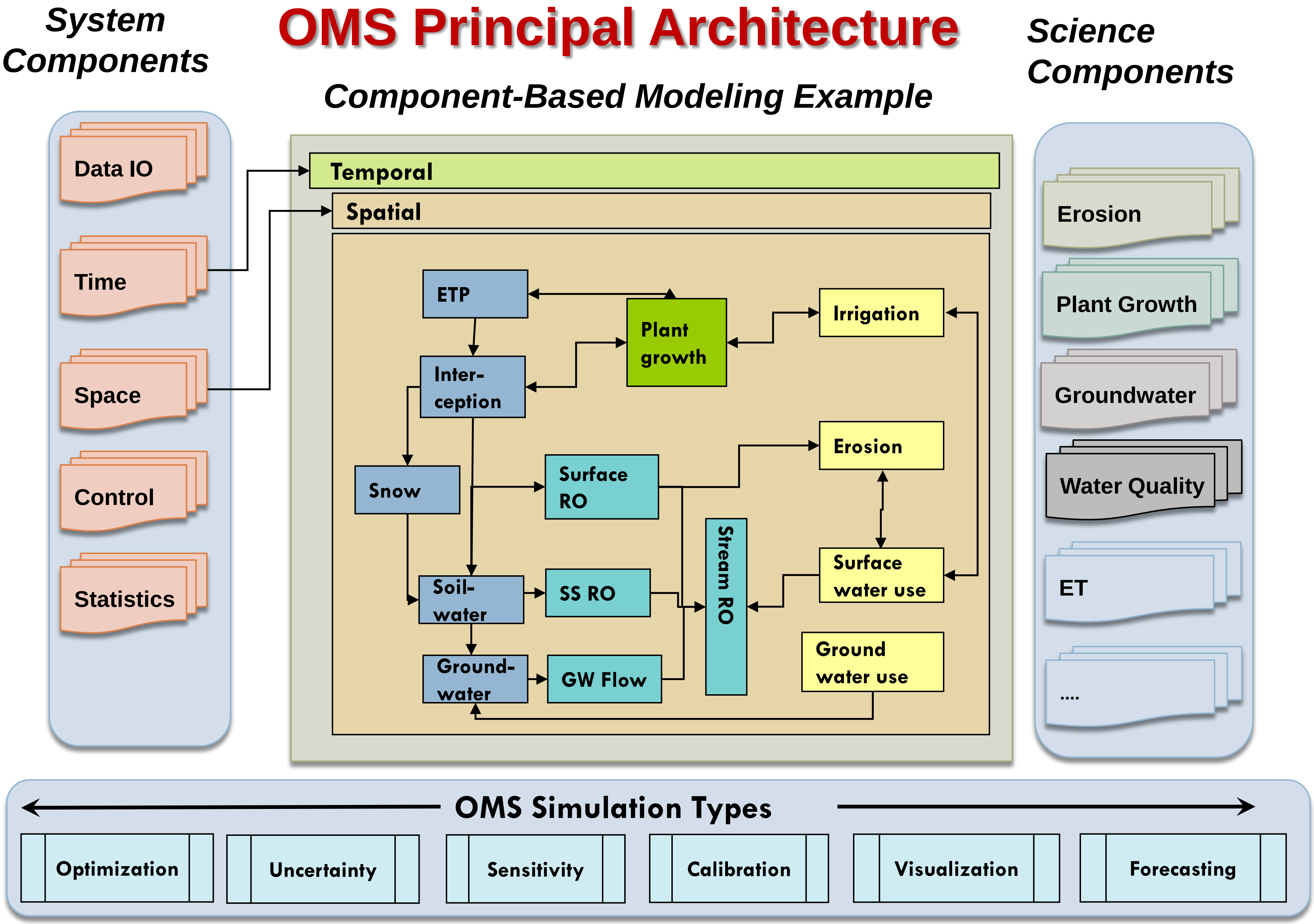
Integration of WEPP v2004.7 hillslope model components into OMS Version 1 – OMS controls space/time looping





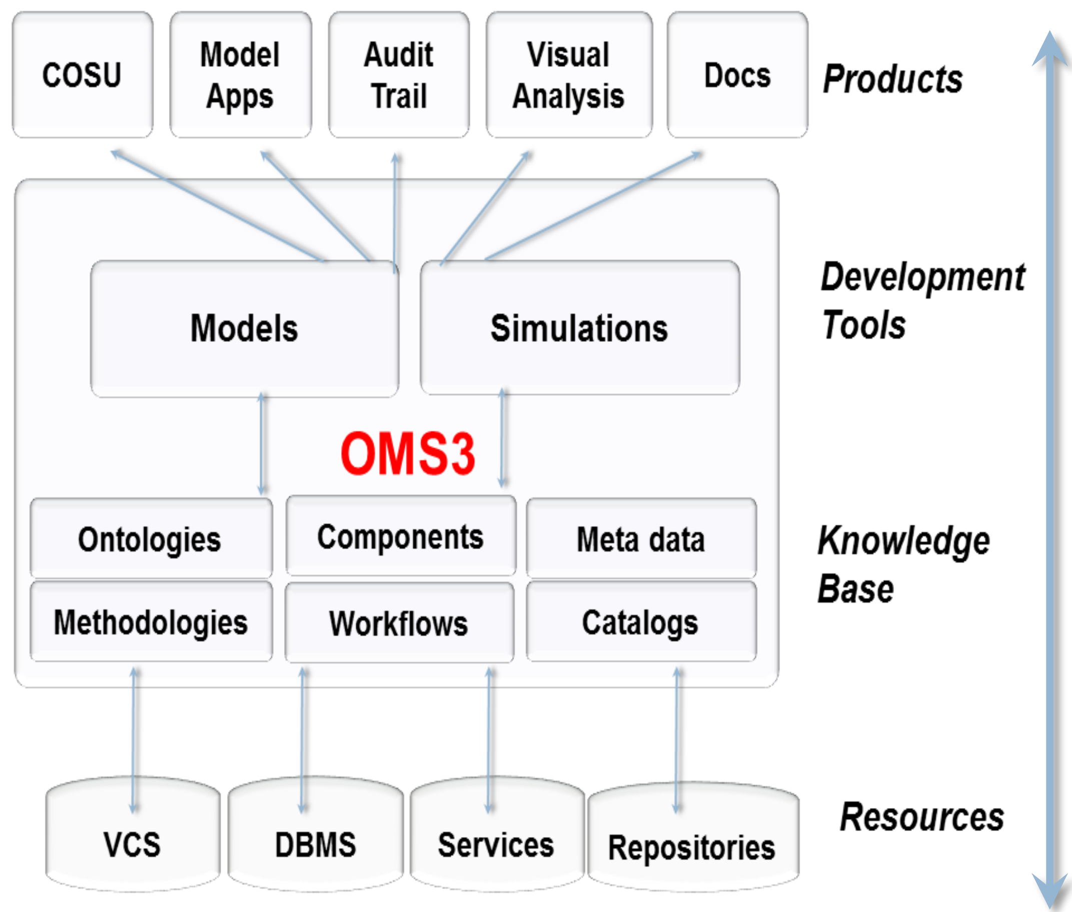
OMS Principal Architecture

Component-Based Modeling Example



Object Modeling System (OMS) Overview

- Software Framework for Environmental Modeling
  - “Science Building Blocks”
- Improve model code quality and modularity to support reuse and substitutability
- Reduce redundancy in model development while allowing for flexible changes in science
- Improve overall model maintainability and deployment - simplify technology transfer
- Reduce IT integration challenges for researchers

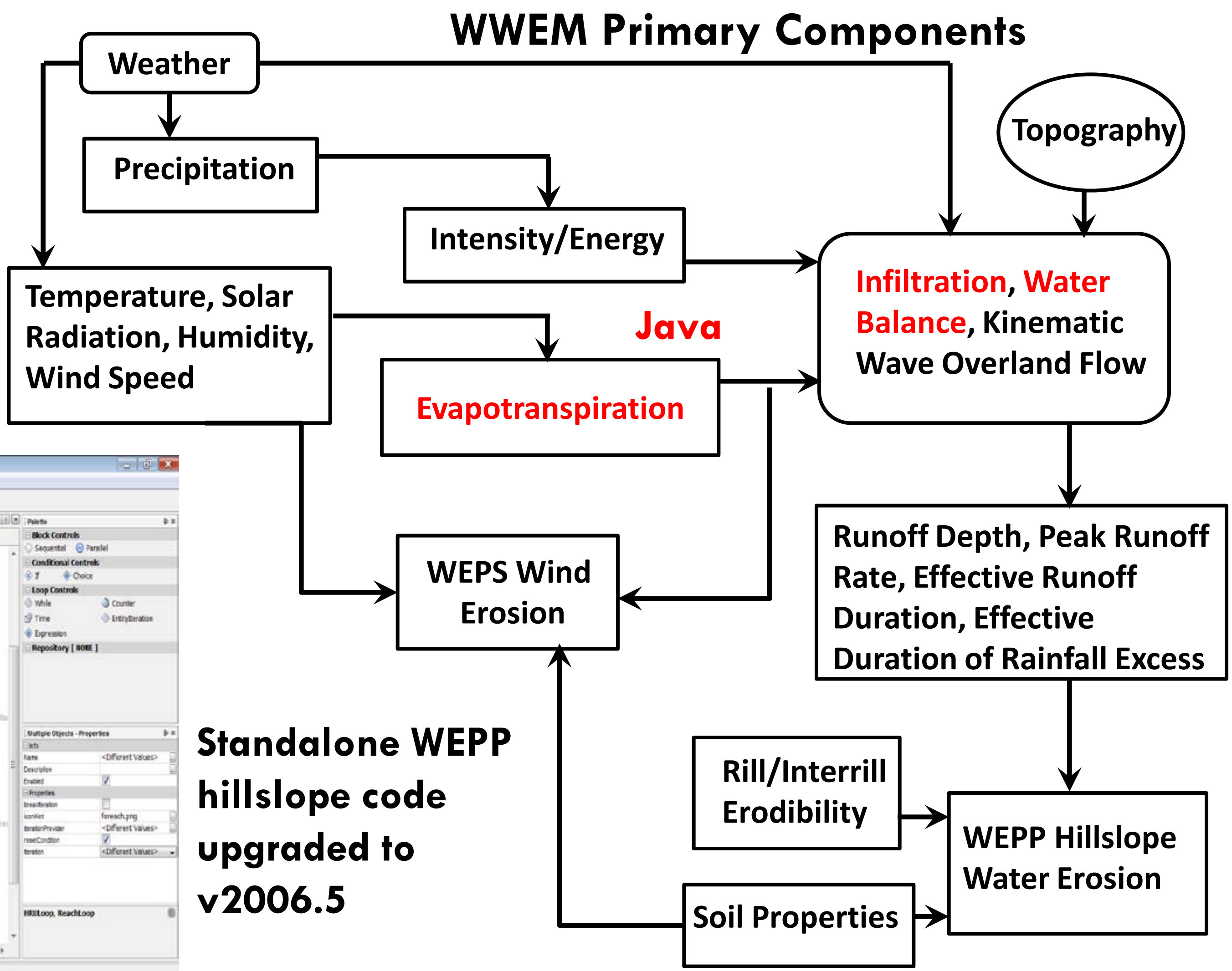


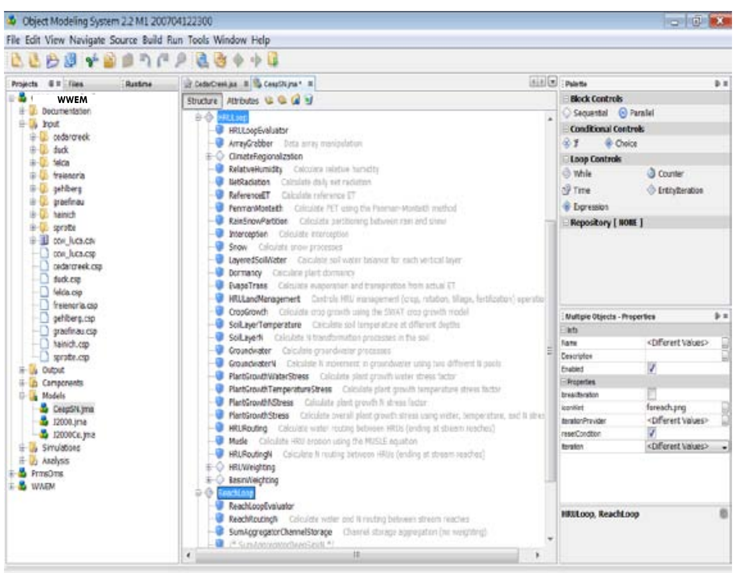
http://www.javaforge.com/project/oms

WWEM History: II. WWEM Components Expanded Under OMS Version 2

2006-2008

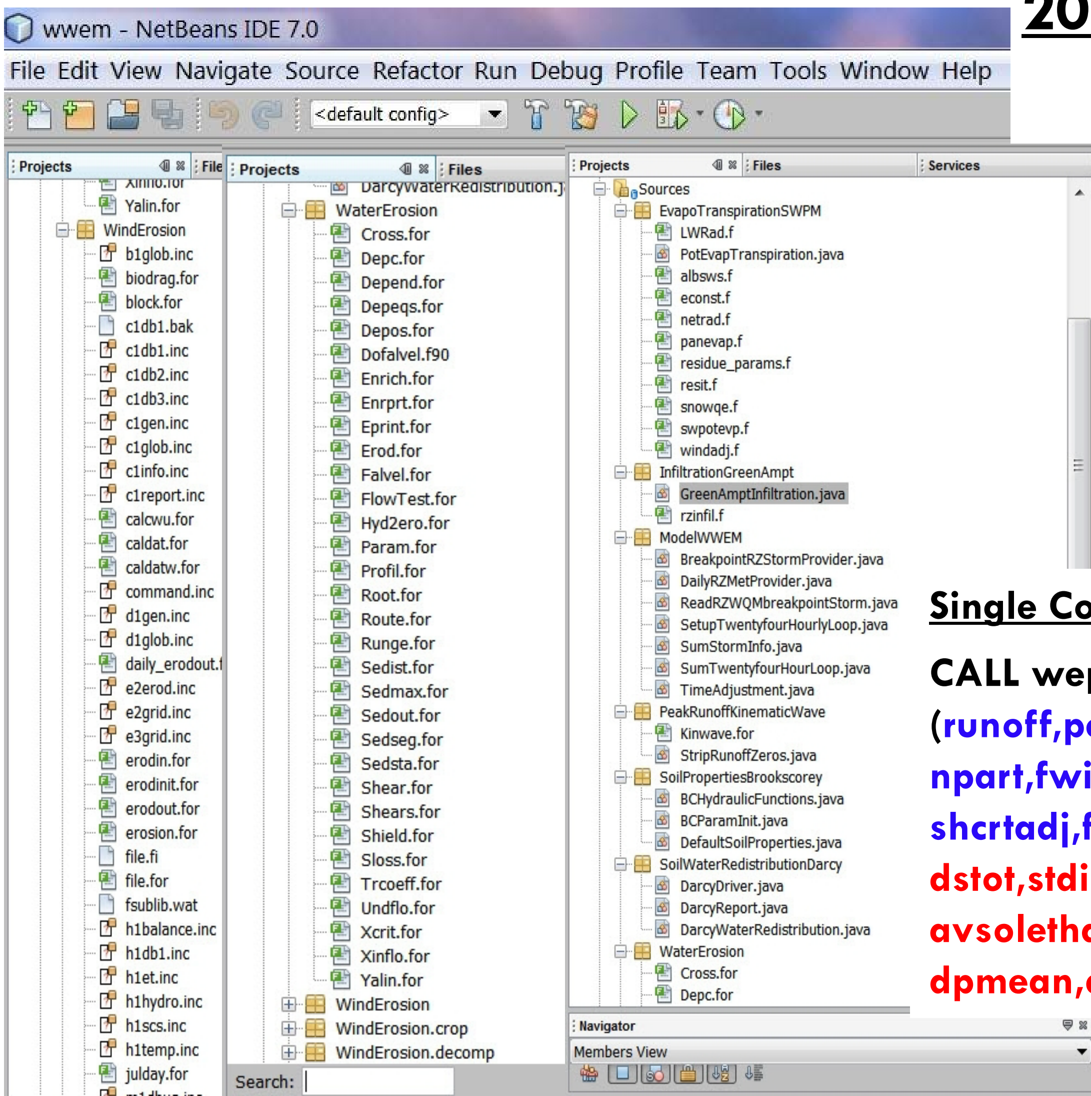
Added infiltration, kinematic wave, ET, snow process, soil water redistribution, and WEPS core wind erosion components





Standalone WEPP hillslope code upgraded to v2006.5

WWEM History: III. WWEM Components Integrated Under OMS Version 3



2010-2011

- WEPP code upgraded to v2008.907 (upgrade to v2010.1 almost completed)
- Space/time control transferred back to model components
- No GUI
- OMS Annotation-Based

Single Component Entry Point

CALL weperos (runoff,peakro,effdrn,effint,effdrr,qout,qsout, npart,fwidth,rspace,width,efflen,kiadj,kradj, shcrtadj,frcsol,frctrl,rrc,nslpts,slplen, dstot,stdist,avsolekgm,avsolekg,avsolekgm2, avsoletha,dtmean,dtmax,dtmxpt,irdgdx, dpmean,dpmax,dpmxpt,enrato)

WWEM Components Modified for OMS3 Annotation Design

Modeling Component

= POJO + Annotations

Plain Old Java Object

Meta data provide **execution control and connectivity,** execution support, and documentation/repository support

Simple Java Example Component with Annotated I/O Fields

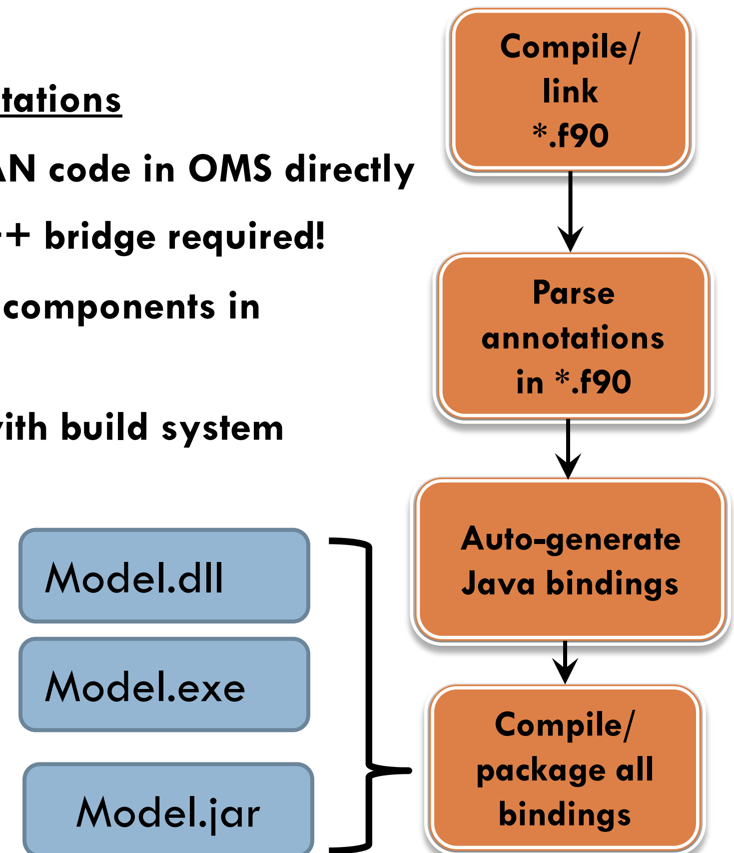
```
public class CircleArea {
  @In  public double radius;
  @Out public double area;

  @Execute
  public void runme() {
    area = Math.PI * radius * radius; }}
```

- Tag the fields being used for input and output with @In and @Out
- Required meta data for OMS
- Also applies to other languages such as FORTRAN, C, and C++

FORTRAN Annotations

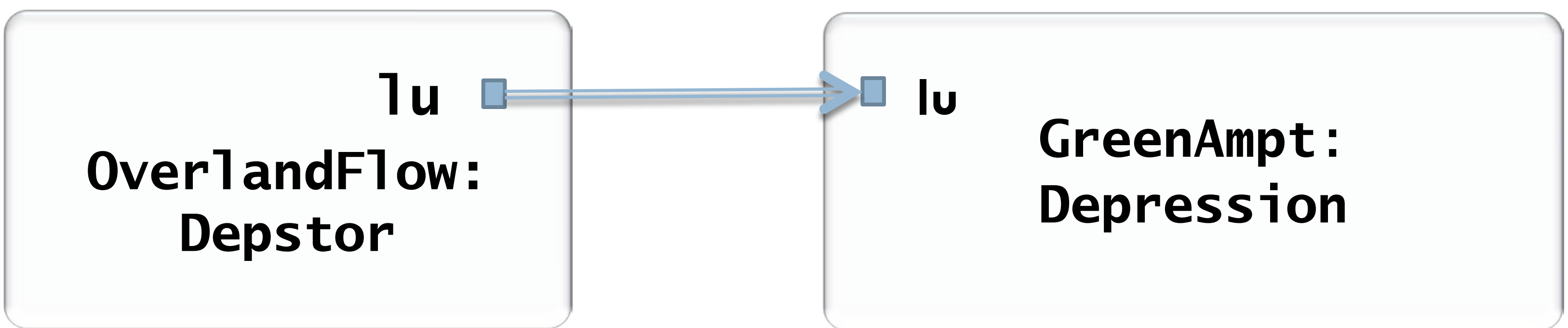
- Use FORTRAN code in OMS directly
  - No C/C++ bridge required!
- Define OMS components in FORTRAN
- Integrated with build system



Using Annotations to Link Model Components in OMS3


```
class OverlandFlow  {
   @Out Depstor
   public LandUnit lu;
   ...
}
```

```
class GreenAmpt {
   @In Depression
   public LandUnit lu;
   ...
}
```




```
…
out2in(Depstor, "hru",
Depression, "hru");
```

… out2in, field2in, out2field, feedback, …

"Constructor Code"

WWEM Evaluation for Water Erosion Using Constant/ Variable Rate Rainfall Intensity Data

Objective

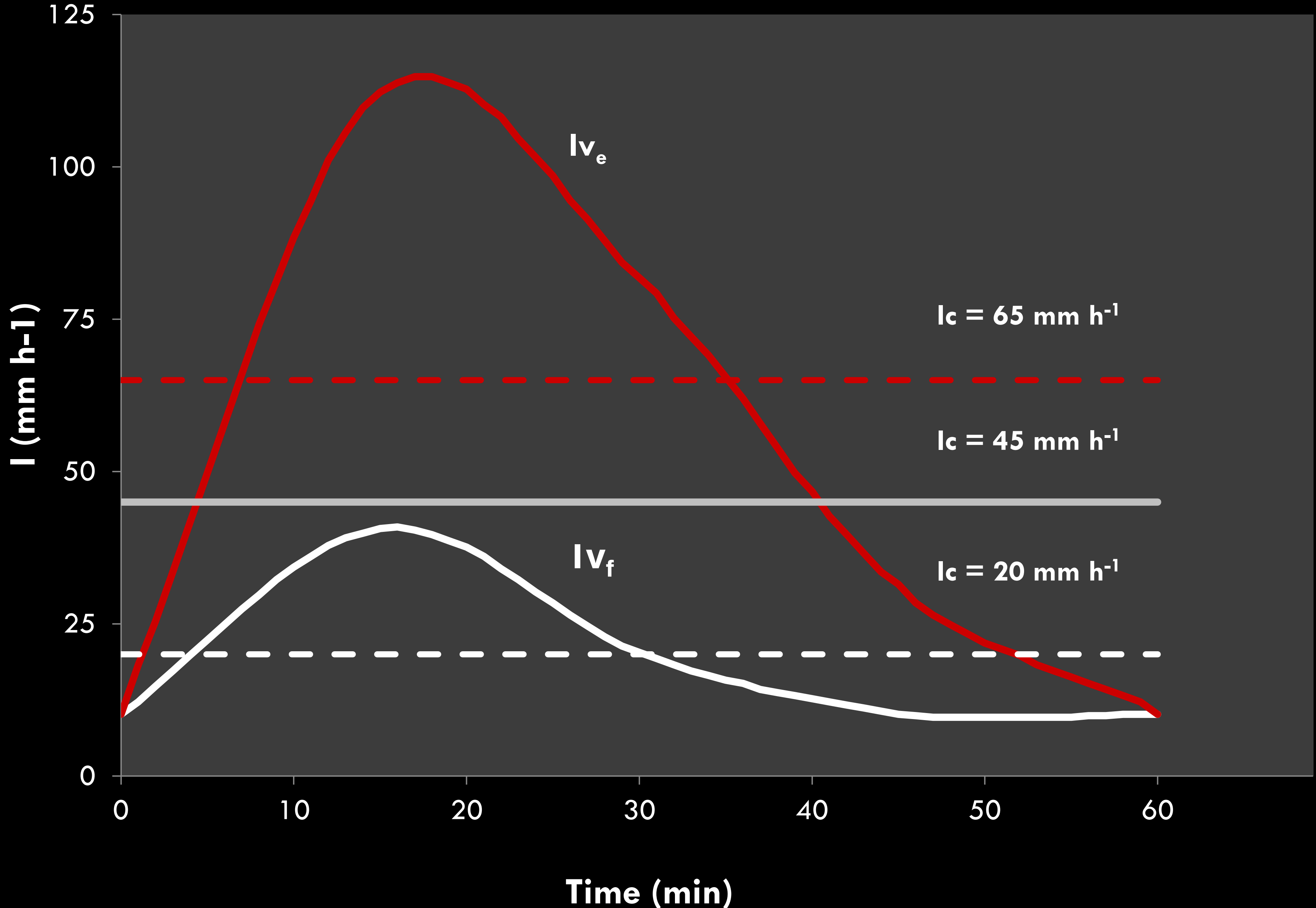
Quantify effects of constant (I_c) and variable (I_v) rainfall intensity patterns (I_{vf} - typical and I_{ve} - extreme) on runoff and sediment losses from a conventionally tilled Ft. Collins sandy clay loam

Rainfall Simulator

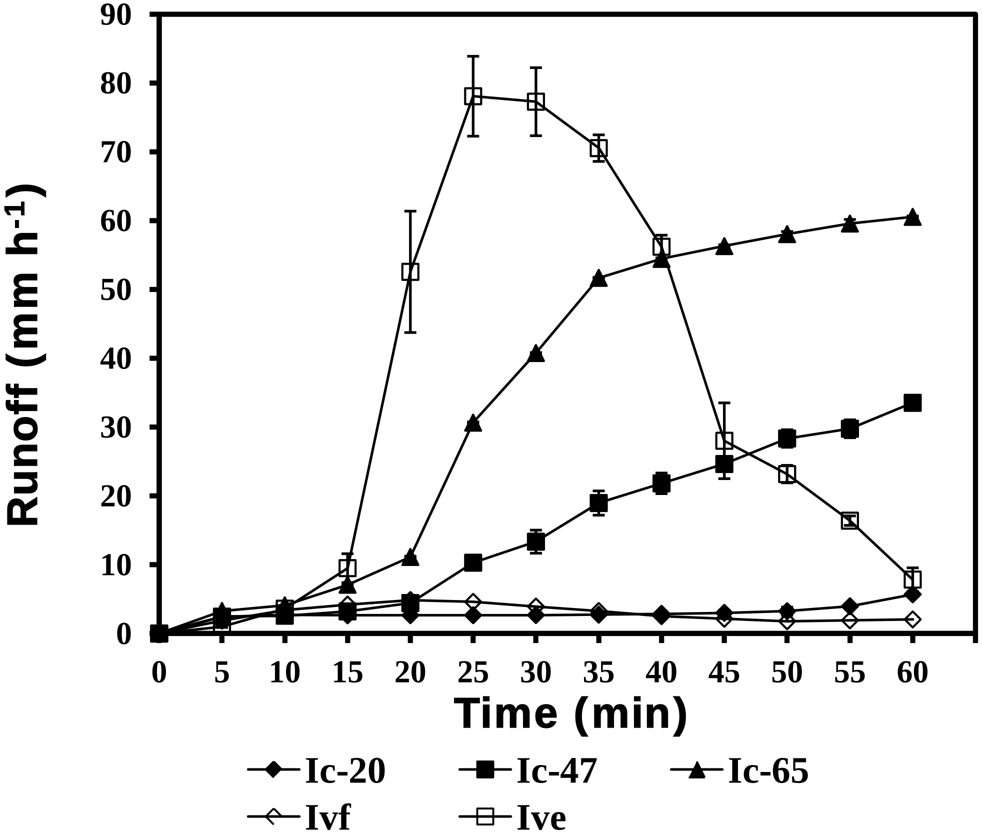
Simulated rainfall applied to each 6-m^2 plot with an oscillating nozzle rainfall simulator (80150 Veejet nozzles, ~2.3-mm median drop size). Simulator was placed 3 m above each 6-m^2 plot.

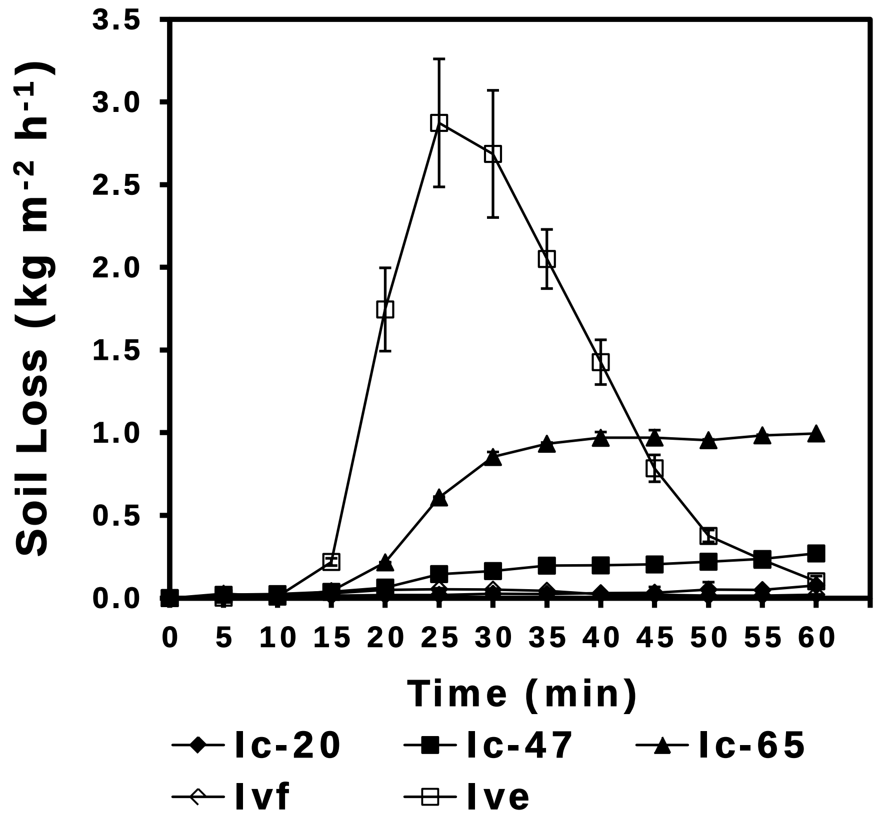
Methods

- Runoff (R) and sediment (E) were measured continuously from each 6-m^2 plot at 5-min. intervals for 60 mins.
- Treatments (5 rainfall intensity patterns: I_c=20 mm/h, I_{vf}, I_c=45 mm/h, I_c=65 mm/h, I_{ve})
- n=3



I (mm h-1)
Time (min)
Iv$_e$
Iv$_f$
Ic = 65 mm h^{-1}
Ic = 45 mm h^{-1}
Ic = 20 mm h^{-1}
0
25
50
75
100
125
0
10
20
30
40
50
60


Runoff (mm h-1)
0
10
20
30
40
50
60
70
80
90
0 5 10 15 20 25 30 35 40 45 50 55 60
Time (min)
Ic-20
Ic-47
Ic-65
Ivf
Ive


Soil Loss (kg m-2 h-1)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0 5 10 15 20 25 30 35 40 45 50 55 60
Time (min)
Ic-20
Ic-47
Ic-65
Ivf
Ive

```
@P, soilLayerBulkDensity, "{1.4, 1.4, 1.5, 1.5, 1.5, 1.5}"
@P, soilLayerClay, "{0.135, 0.22, 0.24, 0.31, 0.2, 0.2}"
@P, soilLayerDepth, "{5.0, 10.0, 20.0, 77.0, 90.0, 150.0}"
@P, soilLayerOrgmat, "{0.015}“
.
.
.
@P, frac, "{0.035,  0.140,  0.243,  0.348,  0.234,  0.0, 0.0, 0.0 ,0.0 ,0.0}"
@P, frcflw, "{0.0,  0.0,  0.0,  0.0,  0.0 , 0.0 ,0.0 ,0.0, 0.0, 0.0}"
@P, frcly, "{1.000,  0.000 , 0.261 , 0.105 , 0.000 , 0.0 ,0.0, 0.0 ,0.0, 0.0}"
@P, frcsol, "1.11"
@P, frctrl, "1.11"
@P, frorg, "{0.111,  0.000,  0.029,  0.012,  0.000,  0.0, 0.0, 0.0, 0.0, 0.0}"
@P, frslt, "{0.000,  1.000,  0.739,  0.182,  0.000,  0.0, 0.0, 0.0, 0.0, 0.0}"
@P, frsnd, "{0.000,  0.000,  0.000,  0.713,  1.000 , 0.0, 0.0, 0.0, 0.0, 0.0}“
.
.
.
@P, fwidth, "100.0"
@P, kiadj, "1719000.0"
@P, kradj, "0.0077"
@P, ktrprv, "0.0"
@P, nslpts, "2"
@P, qsout, "0.0"
@P, rrc, "0.018"
@P, rspace, "1.0"
@P, shcrtadj, "3.178"
@P, shrspv, "0.0“
.
```

Input parameter file - wwem.csv

Initial K_i, K_r, and τ_c initially estimated from WEPP baseline erodibility calculations and then adjusted

Statistical Evaluation – Runoff and Erosion

Intensity (mm h^{-1})	R_T Observed (mm)	R_T Predicted (mm)	PBIAS[1] (%)	E_T Observed (kg ha^{-1})	E_T Predicted (kg ha^{-1})	PBIAS (%)
20	2.7	2.4	-11.1	278	266	-4.3
I_{vf}	3.0	3.4	13.3	268	294	9.7
47	16.1	14.1	-12.4	1382	1182	-14.5
65	38.6	32.5	-15.8	5732	4841	-15.5
I_{ve}	35.3	41.8	18.3	9680	11602	19.9

[1] PBIAS = $\sum (P_i - O_i / O_i) * 100.0$

Results

- Runoff and erosion underpredicted for I_c rainfall events
- Runoff and erosion overpredicted for I_v rainfall events
- WWEM model predictions worsen with increasing rainfall intensity

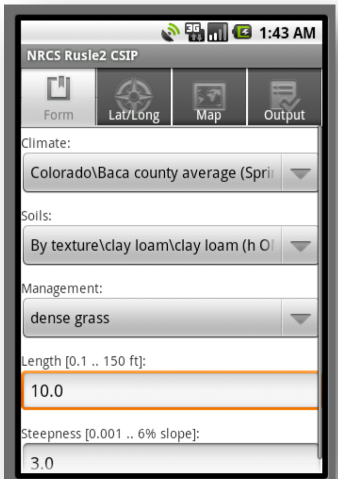
Conclusions

- WWEM contains component-based erosion prediction technology that can easily be linked to existing models, e.g., RZWQM2
- Easily accessible code base for testing and improving WEPP hillslope erosion code
- Excellent academic tool for teaching students erosion prediction modeling fundamentals
- OMS3 provides a vehicle for transferring erosion prediction technology through new methods of delivery

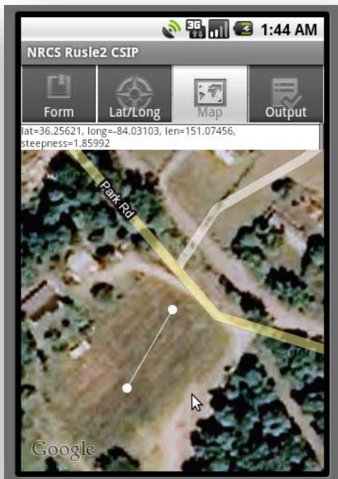
Cloud Services Innovation Platform (CSIP)

- Implement Modeling Infrastructure for NRCS that is:
  - Cost effective (→ Cloud)
  - Highly interoperable
  - Component-based (→ OMS3)
  - Computationally scalable
  - Scalable for data (→NoSQL)
- Prototype selected models via CSIP
  - RUSLE2, WEPP Hillslope …

RUSLE2 Mobile Smartphone Android Application

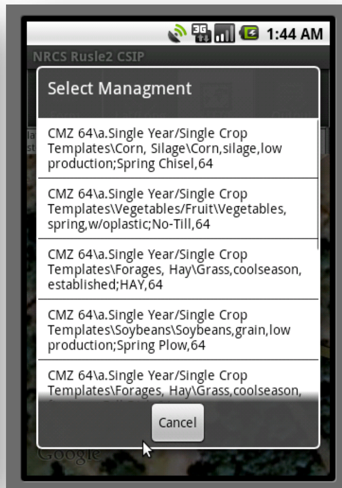


Manual Parameter Selection

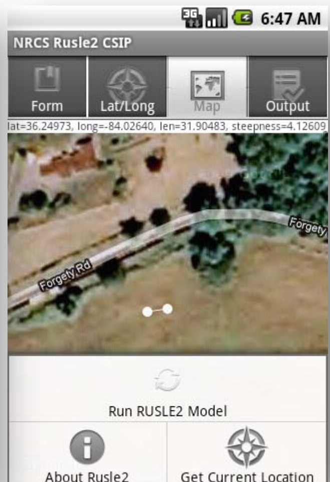


Transect Definition

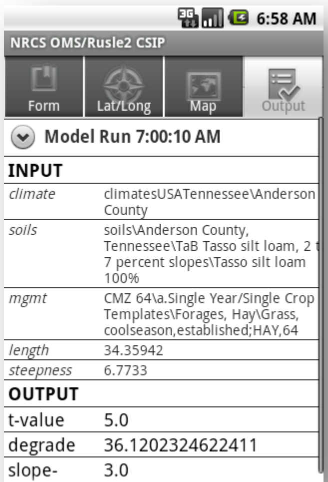
USGS Elevation Service



Location-Based Management Selection



Remote Model Execution of RUSLE2 in CSIP/OMS3



View RUSLE2 Model Results