# *Distributed Hydrologic Model for AWIPS OB8.2*

Software Design Document

Version 1.2

Revision History

| **Date** | **Version** | **Description** | **Author** |
|---|---|---|---|
| 03//29/2007 | 1.0 | Preliminary version | AiVo |
| 04/02/2007 | 1.1 | Updates from internal review | Lee Cajina |
| 04/05/2007 | 1.2 | Updates from PAL review | Chris Dietz |

# *Table of Contents*

1. Overview ..... 4
2. System Architecture ..... 4
3. High-Level Design ..... 5
4. DHM Mods in OB8.2 ..... 8
   4.1 Mod Descriptions ..... 8
      4.1.1 DHM Precipitation Mods Description ..... 8
      4.1.2 DHM Sac-State Mods Description ..... 8
   4.2 Design Details ..... 8
5. Multiple Upstream Inflows in OB8.2 ..... 8
   5.1 Description ..... 8
   5.2 Design Details ..... 9
6. Improve OB8.1DHM performance ..... 9
   6.1 Design ..... 9
7. Improve Script used to Copy Grid Data to User's Local Directory ..... 9
   7.1 Design ..... 9
8. DHM Grid Editor GUI Program ..... 10
   8.1 Design Description ..... 10
9. Allow users to define times other than 12Z for grid output ..... 11
   9.1 Description ..... 11
   9.2 Design ..... 11
10. Package Naming Convention ..... 12
11. Error, Warning, Information Strategy ..... 12

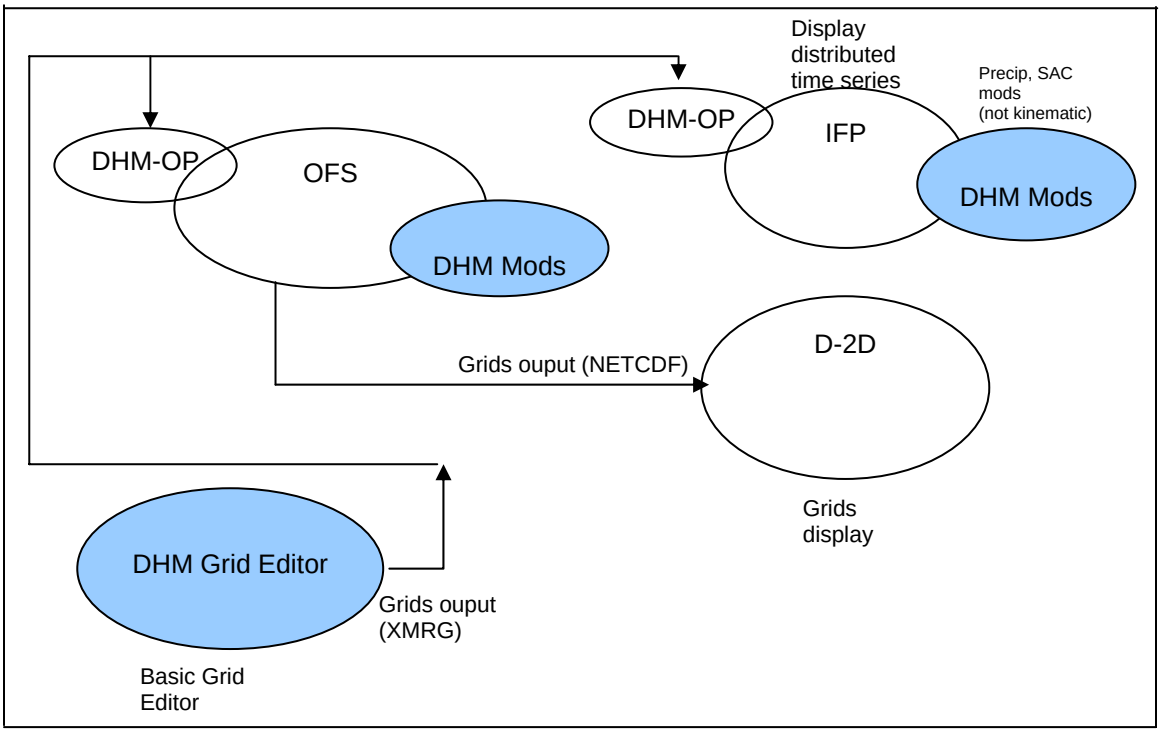
# 1. Overview

This document provides design information at the level of a developer being tasked with implementing some of these changes. Requirements for the enhancements to be made to the Distributed Hydrologic Model software for OB8.2 are documented in the OSIP CONOPS and Requirement Specification Document. The design components needed to satisfy these new requirements can be classified into the following groups:

a) Allow users to specify precipitation modifications (Mods)
b) Allow users to specify SAC state Mods
c) Allow users to model basins which use multiple upstream inflows
d) Improve OB8.1DHM performance (i.e. the time needed to simulate a basin during operations)
e) Improve an existing script used to copy grid data to a user's local directory
f) Allow users to define times other than 12Z for grid output
g) Develop a utility to create operational DHM grids from a previous calibration

# 2. System Architecture

Figure 1: AWIPS Operational Baseline OB8.2



# 3. High-Level Design

A comparison of DHM design features in OB8.2 compared to OB8.1 is provided in the table below.

| **Design Feature** | **AWIPS OB8.1** | **AWIPS OB8.2** |
|---|---|---|
| Programming Languages | Java 1.5, C, $C^{++}$, Fortran | Java 1.5, C, $C^{++}$, Fortran |
| Mode of Use | **Batch Mode**: as an operation within OFS/FCST<br><br>**GUI Mode**: as an operation, with IFP | **Batch Mode**: as an operation within OFS/FCST<br>**GUI Mode**: as an operation, with IFP |
| Graphical Displays | **Timeseries**: Tulsa Plot and/or PLOT-TS operations through IFP (XMOTIF)<br><br>**Grids**: D2D viewer | **Timeseries**: Tulsa Plot and/or PLOT-TS operations through IFP (XMOTIF)<br><br>**Grids**: D2D viewer<br><br>**new** **Mod Utility**: Precipitation mod display through IFP (JAVA), Sac State mod display through IFP (JAVA)<br><br>**new** **Calibration Utility**: stand-alone application to transfer calibrated grid values to operations (JAVA) |
| Libraries | 4 shareware jar files<br>• commons-io1.1.jar<br>• toolsUI-2.2.12.jar<br>• jgrapht-0.6.0.jar<br>• commons-collections-3.1.jar<br><br>5 OHD developed jar files<br>• dhm.jar<br>• dhm-tests.jar | 4 shareware jar files<br>• **update** commons-io1.3.1.jar<br>• toolsUI-2.2.12.jar<br>• jgrapht-0.6.0.jar<br>• commons-collections-3.1.jar<br><br>5 OHD developed jar files<br>• dhm.jar<br>• dhm-tests.jar |

| | | |
|---|---|---|
| | • ofs.jar<br>• distrouting.jar<br>• simplexmrg.jar<br><br>2 OHD developed shared libraries<br>• libsimplexmrg.so<br>• libdistrouting.so | • ofs.jar<br>• distrouting.jar<br>• **update** simplexmrg replaced with rdhmutilities.jar<br>• **new** dhmguis.jar<br><br>2 OHD developed shared libraries<br>• **update** libsimplexmrg.so replaced with librdhmutilites.so<br>• libdistrouting.so |
| Scripts | **Batch Mode**:<br>ofs – used to set CLASSPATH and LD_LIBRARY_PATH environment variables prior to executing DHM<br><br>**GUI Mode**:<br>start_ifp_nwsrfs – used to set CLASSPATH and LD_LIBRARY path environment variables prior to executing DHM<br><br>get_ofs_data – used to copy precipitation grids (XMRGs) to a local directory prior to executing DHM | **Batch Mode:**<br>ofs – used to set CLASSPATH and LD_LIBRARY_PATH environment variables prior to executing DHM<br><br>**new** abort_nwsrfs – used to cleanup any log files created by abnormal termination of FCST<br><br>**GUI Mode:**<br>start_ifp_nwsrfs – used to set CLASSPATH and LD_LIBRARY path environment variables prior to executing DHM<br><br>**update** get_ofs_data – remove portion of script used to copy precipitation grids (XMRGs)<br><br>**new** abort_nwsrfs – used to cleanup any log files created by abnormal termination of IFP (same script used in batch mode) |

| | | |
|---|---|---|
| | | **new** ifp_cleanup – used to cleanup any temporary DHM grids copied for executing DHM within IFP |
| Memory<br>(Physical disk space and RAM) | RAM – default AWIPS environment<br><br>Disk Space – default AWIPS environment and a local partition created on each workstation (/data/dhm: 1 GB) on each LX and XT workstation | RAM – default AWIPS environment<br><br>**update** Disk Space – Will be writing more data to /data/dhm in OB8.2 due to changes associated with user-selected times; size of /data/dhm on each LX and XT workstation needs to be increased from 1GB to 2GB. |
| Application Default Tokens<br>(APPS_DEFAULTS) | dhm_data_dir – directory with input/output XMRG data<br><br>dhm_d2d_data_dir – directory used to write output viewable in D2D<br><br>dhm_d2d_notify_bin – directory with dhmNotify exe; used to ping D2D notification server<br><br>ifp_griddb_dir – directory with user's local set of precipitation XMRGs (used when running DHM-OP through IFP) | dhm_data_dir – directory with input/output XMRG data<br><br>dhm_d2d_data_dir – directory used to write output viewable in D2D<br><br>dhm_d2d_notify_bin – directory with dhmNotify exe; used to ping D2D notification server<br><br>ifp_griddb_dir – directory with user's local set of precipitation XMRGs (used when running DHM-OP through IFP)<br><br>**new** ifp_dhm_data_dir – directory with user's local set of dhm input/output data (used when running DHM-OP through IFP) |

# 4. DHM Mods in OB8.2

The enhancements related to DHM mods in OB8.2 (a and b in section 1) are stated as requirements 45, 45.1, 45.2, 46 in the Requirements Document. Implementing these requirements has a GUI and computations component. Details of the GUI component are further explained in the document "Distributed Hydrologic Modeling User Interface Document", version 1.2. The computations component is explained here.

## 4.1 Mod Descriptions

### 4.1.1 DHM Precipitation Mods Description

A precipitation mod is a multiplier used to edit the hourly accumulated precipitation for each grid cell in a basin. Precipitation mods are applied over a time interval within the observed data period. A precip mod decreases or increases the observed hourly accumulated precipitation in each cell by multiplying the mod value and the original cell value. Precipitation mods are identified using a keyword, three dates and one value.

### 4.1.2 DHM Sac-State Mods Description

A DHM Sac-State Mod is one or more multipliers used to edit the Sacramento model state variables for each grid cell in a basin. A Sac-State mod decreases or increases the model computed sac state in each cell by multiplying the mod value and the original cell value. Sac-State Mods are specified using {state name, mod value} pairs. Each Sac-State mod can specify 1 - 6 pairs.

## 4.2 Design Details

The data flow diagram in Figure 3 gives a systems perspective of how mods (both precipitation and sac states) are processed. As shown in the figure, mods are first entered by the user and then move through various DHM components until they are finally stored as Java objects. The mod objects are then used in the Sacramento Model algorithm as shown in the class diagram in Figure 5.

# 5. Multiple Upstream Inflows in OB8.2

The enhancements related to allowing for multiple upstream flows (c in Section 1) in OB8.2 are updates to the software needed to meet requirements 84 and 85 in the requirements document.

## 5.1 Description

When modeling downstream basins (basins not at the top of the channel network) the model's results are dependent on one or more previously computed upstream flow(s).

Prior to OB8.2, DHM only allowed for one upstream flow in its algorithm. In OB8.2, DHM will be enhanced to allow for up to five upstream flows.

### 5.2 Design Details

The bulleted list below describes the changes needed to enhance DHM to accept up to five upstream flows.

- Update DHM-OP definition to allow more than one inflow card
- Update DHM-OP execution to use multiple input time series
- Update C to java initialization (JNI calls)
- Update Java modeler to store multiple upstream inflows in existing Hashtable

Figure 4 is a diagram showing the sequence steps each upstream flow timeseries takes as it's used in DHM. Figure 6 shows how the upstream flow timeseries are used by the algorithm.

## 6. Improve OB8.1DHM performance

### 6.1 Design

It was recognized with the OB7.2 and OB8.1 delivery of DHM that there are performance issues involving reading precipitation grid data during the DHM model run in the IFP Program. These issues exist with the OFS batch Program, as well, but they are not as noticeable since it does not have a graphical interface. In OB8.2 the Java classes used to read precipitation XMRGs were re-factored to improve performance; no design changes were made.

## 7. Improve Script used to Copy Grid Data to User's Local Directory

### 7.1 Design

It was recognized with the OB7.2 and OB8.1 delivery of DHM that there is the potential for the disk space allocated for DHM to fill up. The problem is due to copying large amounts of XMRG grid files to a user's local directory for running DHM from IFP. The problem was noticed by IFP users not using DHM.

The new approach to copy XMRGs will copy the exact number of XMRG files needed, instead of doing a blanket copy of all files. This will be possible because the code used to copy files will be moved inside of the DHM module where the start date and end date are known. The following is more information on changes that will be made to facilitate this requirement:

- Remove the copying of XMRGs from the "get_ofs_data"script.
- Move XMRG copying to 1st DHM-OP in each Forecast Group; and make it a temporary copy ( model run start date and end date are known; only need to copy a subset of XMRG files)
- Remove temporary XMRGs when quitting IFP Program

Benefits of new approach

- Unlikely disk space fill up
  5 days observed – 120 files (24hours per day and 5 days total)
  24 hours qpf – 48 files
  168 * .5 MB = 84 MB per IFP session. The current data directory is allocated 1Gigabytes in disk space.
- Copying upon selecting a forecast group allows users to refresh any changes to XMRG files within an IFP session
- Copying as part of DHM execution does not effect non DHM-OP users

# 8. DHM Grid Editor GUI Program

The new DHM Grid Editor GUI program included in OB8.2 is an interim tool which allows sites to edit the DHM grids. NOTE: this program will be superseded by the Graphical Forecast Editor (GFE) in OB8.3.

The Grid Editor addresses requirements 89, 89.1, 89.2, 89.2.1, 89.2.2, 89.3, 89.4. More information on the GUI is given in the attached GUI document. Some information on its design is given below.

## 8.1 Design Description

DhmGridEditor is a graphical user interface that will provide DHM users the capability to create DHM model parameter and state grids (SAC Runoff and Kinematics Routing) and monthly PET grids by specifying a multiplier or in the case of SAC state grids by specifying a percentage of maximum. Users can also create DHM model Parameter and State grids and monthly PET grids by specifying a constant value. All Grid edits are done on a per-basin basis.

The Java program will be developed as a stand alone program to be used before running the DHM operation in NWSRFS. Its primary purpose is to create the Parameter, State, and PET grids needed by the AWIPS version of DHM.

The following steps will be made to facilitate the requirement:

a) Create a main GUI display window with capability to specify:
   - Input directory
   - Output data directory

- Connectivity file name and path
- List of selectable basin id

Initial values from an input ASCII file

b) Create a SAC –parameter and States display window that allow a user to change input value
c) Create a Kinematic –parameter and States display window that allow a user to change input value.
d) Create a PET –display window that allow a user to change input value.
e) Write new values to the output grid data file. Output grid data should be able to view via any existing utility which reads XMRG files.

See Figure 2 for DHM Grid Editor's Java class layout.

# 9. Allow users to define times other than 12Z for grid output

The enhancements in OB8.2 related to e in Section 1 address requirements 86.2 and 86.5 in the requirements document. This requirement was also identified as a high priority during beta testing of OB7.2.

## 9.1 Description

In the OB7.2 and OB8.1 versions of DHM, grids showing model results (sac/kinematic states and channel flow) are only written during batch mode runs of DHM-OP.

During beta testing of OB7.2 DHM, RFC forecasters expressed the need to view model results during interactive runs of DHM (through IFP). In addition to the grids already written out during batch mode runs, there may be other model results users are interested in.

## 9.2 Design

Configuring DHM to write out results during runs from within IFP will require users to spawn a GUI from IFP to select which dates/layers they wish to see. More information on the look of this GUI is given in the document "Distributed Hydrologic Modeling User Interface Document", version 1.2. From a computational standpoint we will refactor some of the existing classes used to write out model results.

Maintenance of these additional grids will follow the approach used for precipitation XMRGs (this approach was updated in OB8.2 – see Section 7).

# 10. Package Naming Convention

The standard OHD package naming convention, beginning with "ohd.hseb.dhm", will be adopted. Sub-package name convention will follow the guidelines established by Sun. See http://java.sun.com/docs/codeconv/html/CodeConventions.doc8.html.

# 11. Error, Warning, Information Strategy

The strategy for displaying Error/Warning messages is to follow the existing Error/Warning Message guidelines for the NWSRFS, instead of the DHM-unique strategy used in OB7.2 and OB8.1. All Error/Warning Message will now be printed in the output of the batch OFS batch Program and IFP Program. The error message will be displayed in the background window. All the output log error files will be stored in the user's OFS output data directory.

# Appendix A

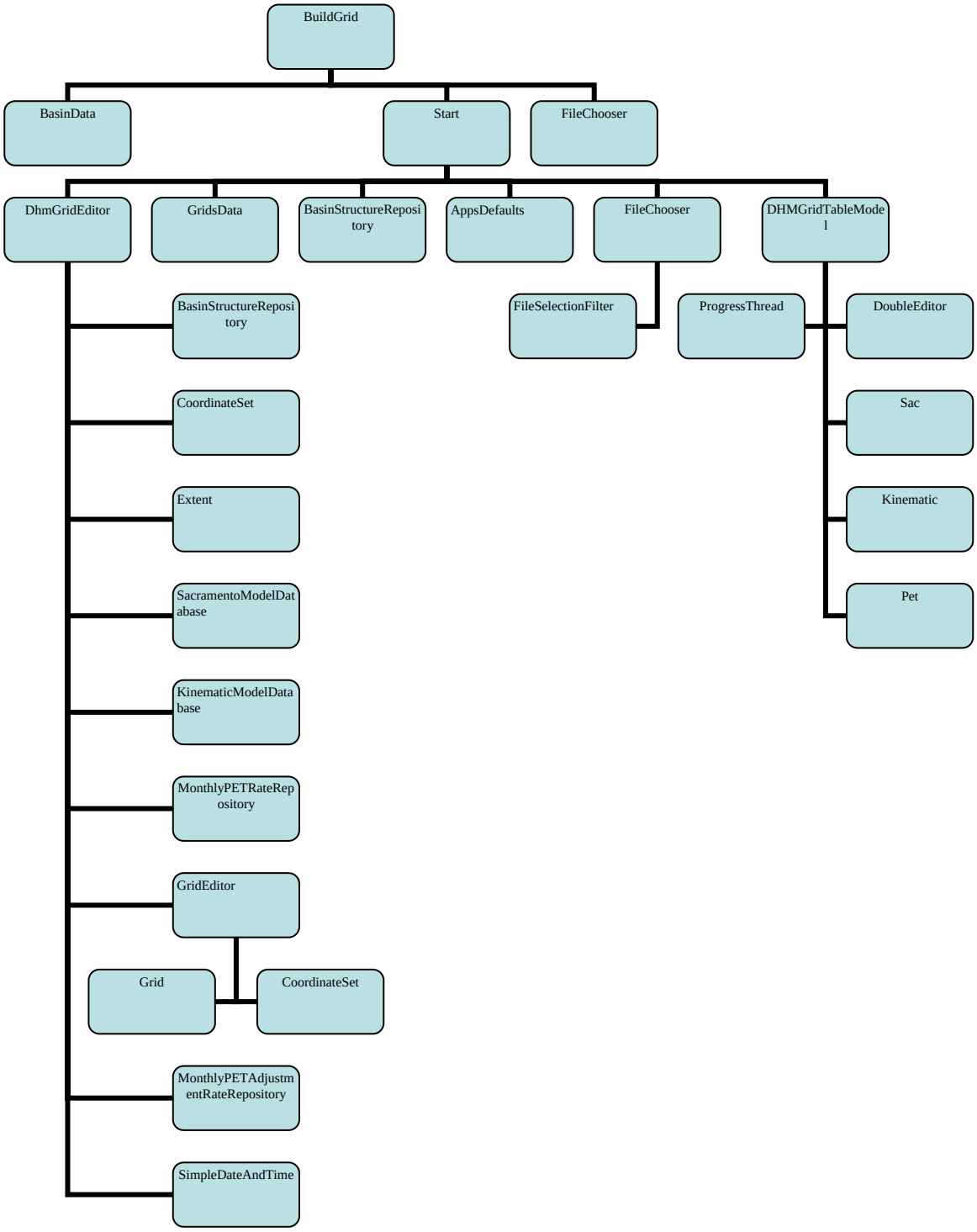



Figure 2: DhmGridEditor Utility Class Diagram

# DHM MODS (OB8.2)

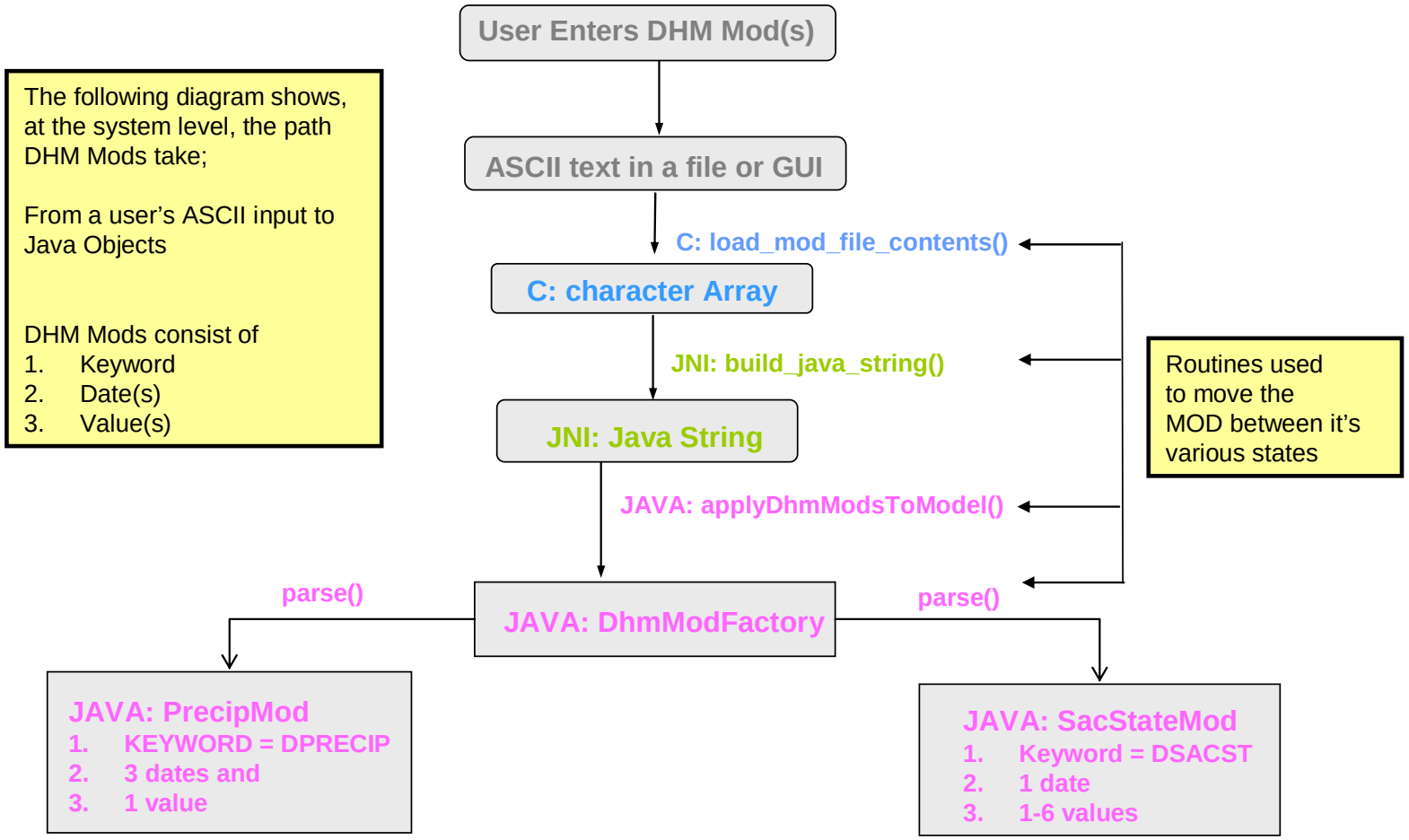



Figure 3: DHM-Mods Flow Diagram

# DHM Upstream Flow (OB8.2)

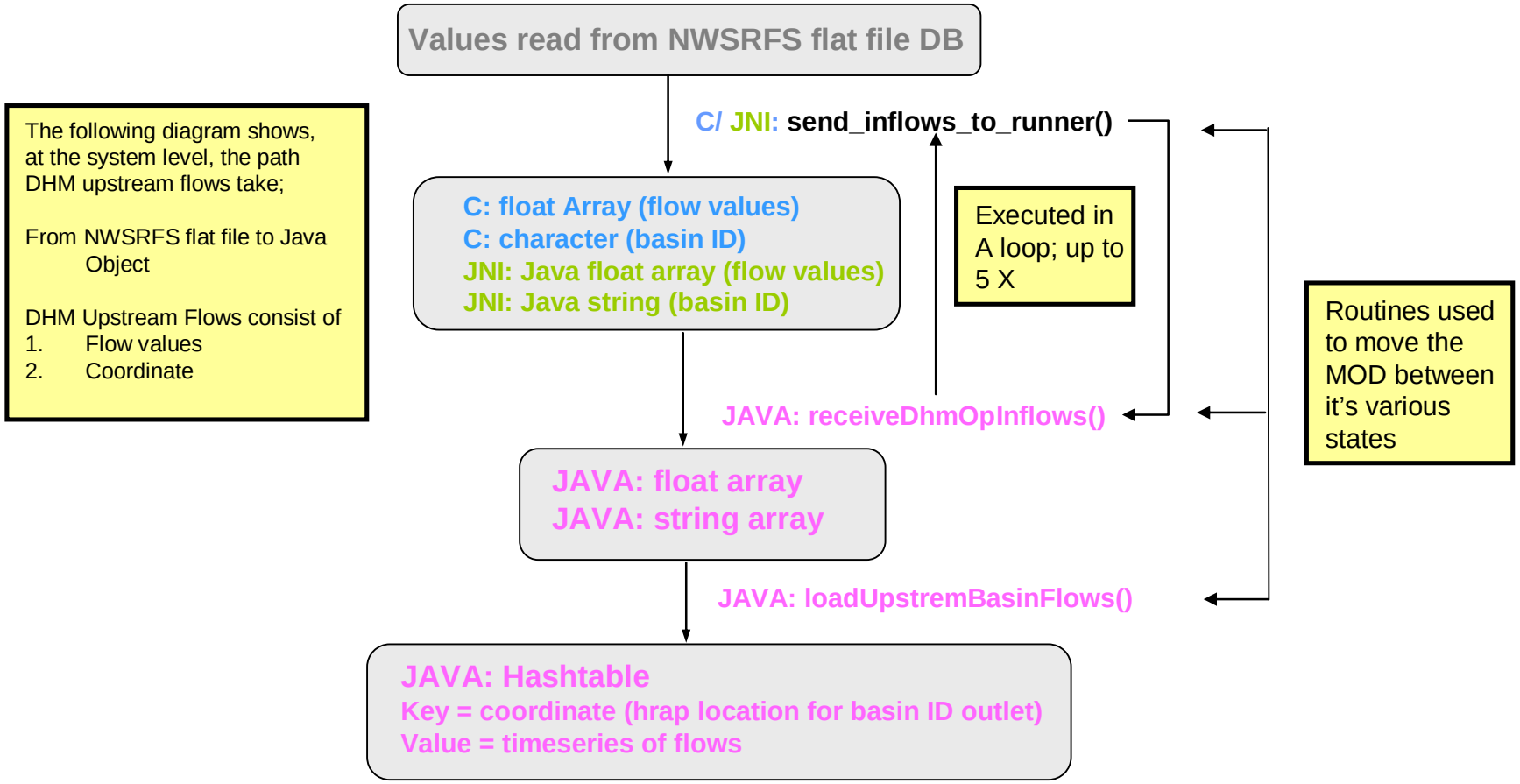



Figure 4: DHM-Upstream Flows Flow Diagram

# DHM Class Diagram For Mod Feature in OB8.2

The following diagram is not a complete DHM Class Diagram; only the Class' related to DHM MODS

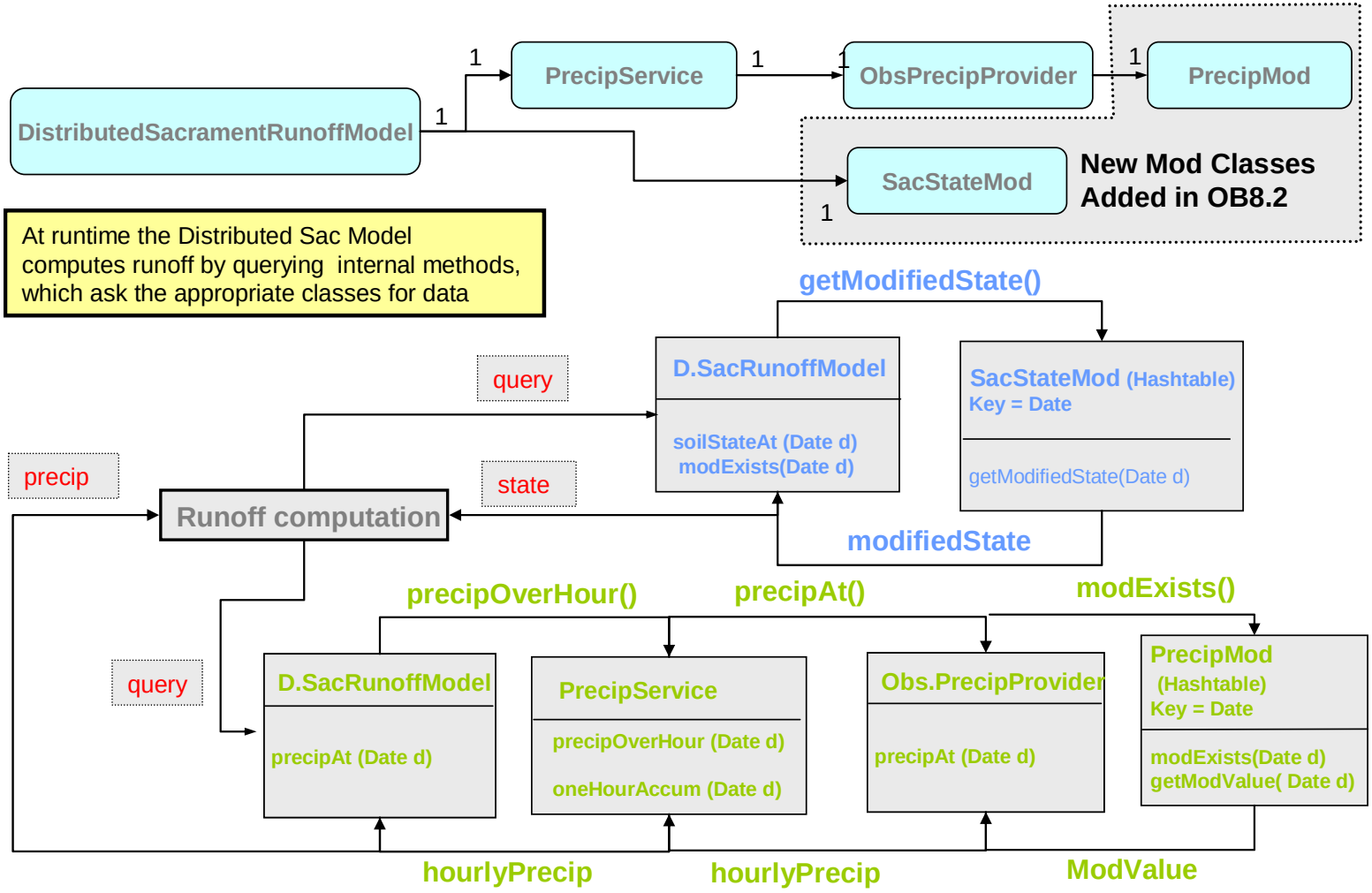



Figure 5: DHM-Mods Class Diagram

# DHM Class Diagram For Multiple Upstream Flows Feature in OB8.2

The following diagram is not a complete Class Description; just the parts relevant to upstream flows

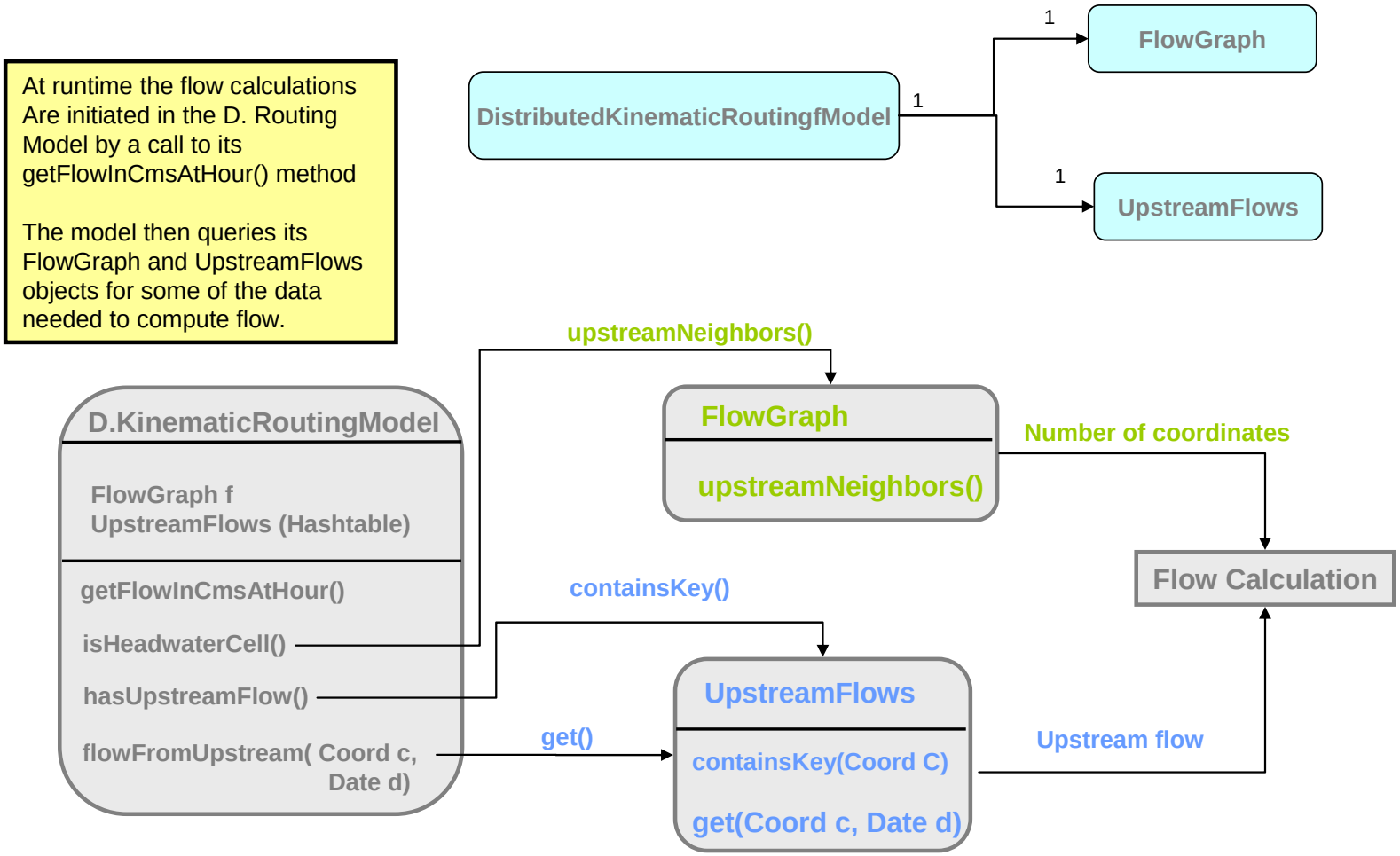



Figure 6: DHM-Upstream Flows Class Diagram