International Congress on Environmental Modelling and Software

7th International Congress on Environmental Modelling and Software

Jun 16th, 9:00 AM - 10:20 AM

The CSDMS Standard Names: Cross-Domain Naming Conventions for Describing Process Models, Data Sets and Their Associated Variables

Scott D. Peckham
University of Colorado, INSTAAR, scott.peckham@colorado.edu

Scott D. Peckham, "The CSDMS Standard Names: Cross-Domain Naming Conventions for Describing Process Models, Data Sets and Their Associated Variables" (June 16, 2014). *International Congress on Environmental Modelling and Software.* Paper 12.
http://scholarsarchive.byu.edu/iemssconference/2014/Stream-A/12

International Environmental Modelling and Software Society (iEMSs)
7th Intl. Congress on Env. Modelling and Software, San Diego, California, USA
Daniel P. Ames, Nigel W. T. Quinn, and Andrea E. Rizzoli (Eds.)
http://www.iemss.org/society/index.php/iemss-2014-proceedings

The CSDMS Standard Names: Cross-Domain Naming Conventions for Describing Process Models, Data Sets and Their Associated Variables

Scott D. Peckham [a]

[a] University of Colorado, INSTAAR, 1560 30th Street, Boulder, CO (Scott.Peckham@colorado.edu)

Abstract: The CSDMS (Community Surface Dynamics Modeling System) modeling framework provides mechanisms that allow models and data sets from different contributors to be automatically coupled in a plug-and-play manner to create composite models. In developing this capability, a key challenge has been that of **automatic semantic mediation** or name matching, because each model or data set (here called a resource) uses its own set of terms for input and output variable names. These names are often domain-specific or abbreviated. In order for the CSDMS framework to determine whether one model's output variable is appropriate to be used as another model's input variable, a standardized and precise description of each variable is required. If additional information (e.g. a variable's units) is also provided in a standardized way, the framework can also automatically apply conversions (e.g. unit conversion) when needed so that coupled resources can share the numerical value of a variable. This paper's purpose is to (1) define this semantic mediation problem in terms of design criteria for a desired solution, (2) compare alternate approaches to the problem and (3) propose a new set of naming conventions, the CSDMS Standard Names, that meet the stated design criteria. Although this work is ongoing, CSDMS Standard Names are currently used within the CSDMS framework and are proving to be an effective solution strategy for this problem.

Keywords: automatic semantic mediation; naming conventions; variable names; model metadata; lingua franca

1 Introduction and Motivation

The CSDMS repository contains a growing set of 225 open-source models and tools that together are able to solve a wide variety of problems in the domain of earth surface process dynamics. Each solves a particular problem and has its own set of input variables (required data that it must get from another model or data set) and output variables (data it can compute using physical laws and approximations). The breadth of these models means that by coupling them together in various ways, their combined capabilities can provide predictions for a much larger set of science problems. These models often complement one another, with one model providing an output variable that another model needs as an input variable. However, since the models were developed by different people from different science domains, they each use different names and abbreviations for their variables in their source code. One model might be able to compute the volumetric flow rate at a river mouth, calling it "streamflow", and another model might need this same variable to do its calculations, calling it "discharge" (or just "Q", the standard abbreviation used by hydrologists). The two models might also use different units for this variable. With only minimal changes to model source code, how can the model coupling framework understand the semantic equivalence, "see" the possibility for sharing this variable, automatically link the two models and adjust for the fact that they use different units?

1.1 Four Possible Approaches to the Semantic Mediation Problem

Mediation by users at runtime. At the time of coupling, present the internal variable names of each resource to the user and rely on their semantic knowledge to manually achieve each pairing. This approach is currently used by the OpenMI Configuration Editor (Gregersen et al., 2007). Cons: (1) The model user may not have sufficient domain knowledge, especially when coupling cross-domain components. (2) A coupled model configuration may require sharing a large number of variables; this is then labor intensive and must be repeated each time the model is used. Pros: High accuracy for expert users.

Mediation by reasoning software. Framework developers create “intelligent” software that tries to determine which variable names are semantically equivalent through some kind of reasoning algorithm. This algorithm may use “crawling” or “scraping” methods to automatically harvest domain knowledge from various sources. Cons: (1) It is very difficult to capture domain knowledge in software. (2) There will be cases that cannot be resolved using available information. (3) Matches may not be accurate and need to be verified by a domain expert. Pros: (1) Large amounts of data can be processed to build up domain knowledge. (2) High accuracy may eventually be possible.

Mediation by framework developers. At the time a resource is contributed to a repository, framework developers ask the contributor questions to clarify the semantic meaning of every internal variable name. This information must then be encoded in a machine-readable form, e.g. by mapping into a standardized list of variable names. Cons: (1) Contributors may not want to participate. (2) This is labor intensive for framework developers and not scalable to a large number of resources. Pros: (1) High accuracy.

Mediation by contributors using a standardized list. At the time a resource is contributed to a repository, the contributor searches a standardized list to find semantically equivalent names for each of their resource’s variable names. These matches are recorded as a mapping, such as a Python dictionary or “crosswalk”, and included with the contribution. Pros: (1) High accuracy. (2) One-time cost. (3) No effort for users. Cons: (1) A contributor may not want to provide the mapping.

2 Design Objectives for Standardized Variable Names

Avoid ambiguous variable names. It should be easy for a resource contributor (domain expert) to find the right match in a standardized list without assistance. Distinct variables need unique, descriptive names. Many terms (especially quantity names) are reused with different meanings within different contexts or science domains. Examples include: conductivity (electrical, hydraulic, ionic, thermal), concentration (mass, molar, number and volume), stress (deviatoric, normal, shear), viscosity (dynamic shear, dynamic volume, kinematic shear, kinematic volume, extensional, eddy) and vorticity (absolute, potential, relative). Many terms have precise meanings with associated units, such as speed (vs. velocity), flux and rate. Naming conventions must therefore be expressive enough to avoid ambiguity while striving for short, concise names that eliminate ambiguity.

Avoid domain-specific terminology. Standard variable names should use terms that are understood by most geoscientists, regardless of domain. For example, some domains use “tendency” as a synonym for “time derivative”, but the latter term is much more widely understood and is also more easily extended to other types of derivative.

Use generic or already-standardized object names. Words like river, stream, brook and creek are essentially synonyms for the same type of object, although they convey some information regarding size. While including an object name in a variable name provides many benefits, it is counterproductive to support synonyms because it makes finding valid matches more complicated. Instead, a generic object name such as “channel” should be chosen to represent a class of similar objects. In chemistry, there are widely used standard names for the elements and most known chemical compounds, (e.g. the standard names of the International Union of Pure and Applied Chemistry, IUPAC).

Support for approximate or closest matches. Some models may apply to any object of a certain type (e.g. a tree or fish), while other models may only apply to a specific object of that type (e.g a species of tree or fish). If a variable name contains the object type name in addition to words that identify a specific object (e.g. bluejack oak tree), it may be possible for automatic semantic mediation software to find the closest match when an exact match cannot be found (e.g. oak tree or tree).

Ability to specify multiple objects. Most quantity names, such as temperature, only require one object (which may be a *medium*) to be specified (e.g. air, snow, soil, water). However, there are some quantity names that require two objects (or substances) to be specified. In some cases, one object may be contained within or moving through the other. This occurs for quantities like: chemical affinity, concentration, diffusion coefficient, miscibility, partial pressure, refraction index, relative saturation, solubility and volume fraction. To quantify friction with a kinetic or static friction coefficient, it is necessary to specify the two materials that are in contact (e.g. rubber and concrete). To quantify the distance between two objects, or the bond dissociation energy in molecules, two objects must also be specified.

Avoid mixing object names into quantity names. Quantity names should be general enough to be reused with different objects, but domain-specific quantity names often imply a certain object. For example, in hydrology the focus is on water, and names like: relative humidity (relative saturation applied to water vapor in the atmosphere), streamflow (volume outflow rate applied to water in a stream as the control volume) and rainrate (liquid equivalent precipitation rate of water in the atmosphere) are used. Note that “liquid water equivalent” also violates this criteria. Jupiter’s moon, Titan, has a “methane cycle” similar to Earth’s hydrologic cycle, and quantities should be reusable in that context.

Parsability and strict adherence to rules. Strict adherence to rules allows the parts of a name to be extracted via simple string parsing. This allows software to be written that: (1) checks the validity of a proposed name, (2) automatically constructs valid names from other lists, (3) finds partial or closest matches when there isn’t an exact match and (4) builds an ontology from a set of names.

Natural grouping by object via alphabetization. When a domain expert searches an alphabetized list for a match, it is convenient for related names to appear close to one another, even those created by applying a mathematical operation to an existing name.

Support for mathematical operations. In modeling especially, many important input and output variables result from applying a mathematical operation (e.g. integral, derivative, function, etc.) to an existing quantity. Many models utilize vector or tensor quantities, so operations are needed to describe associated scalar quantities (as detailed in Section 3.3).

Support for dimensionless numbers. Dimensionless numbers (e.g. Froude, Peclet, Reynolds, Richardson, Stokes) are often used in models and are computed from other system variables. They typically serve as “deciders” of when the system will transition to some new state (e.g. turbulent vs. laminar or subcritical vs. supercritical flow).

Support for mathematical and physical constants. These constants occur frequently in models and should be retrievable from a model. Examples include the “standard gravity constant” for Earth, denoted by “little g” and the “universal gravitational constant”, denoted by “big G” (from Newton’s law of gravitation). Each planet has its own standard gravity constant and solar constant. Constants like these appear throughout physics.

Support for empirical parameters. Many empirical laws allow one system variable to be computed as a function of one or more other variables. Power laws are very common, in which case the “law” has both a coefficient and one or more exponents. For example, Manning’s formula for flow resistance in channels has a parameter called “Manning’s n”. Other examples occur in rheology (stress-strain laws) and soil physics.

Support for incoming or outgoing flow rates and fluxes. Fluxes and flow rates are used to quantify the rate at which mass, momentum, energy, volume or moles move into or out of a control volume. The word *rate* implies “per unit time”, so a *mass flow rate* has SI units of $kg\ s^{-1}$. The definition of a *flux* in this context is a “flow rate per unit area”, so a *mass flux* has SI units of $kg\ m^{-2}\ s^{-1}$. Often, the name of the process that is producing the flux or flow rate can be chosen to clarify whether flow is into or out of the control volume (e.g. irradiation vs. radiation or inflow vs. outflow). To avoid ambiguity, the control volume that the process is acting on must be specified (i.e. the object) *and* a directional process name or an extra adjective (e.g. incoming or incident) must be included in the variable name. In hydrology, the word *discharge* may be used for a volume flow rate that is either incoming or outgoing, while its opposite, *recharge*, is used for a different concept.

Support for reference quantities. Many quantities are defined with respect to a reference value of some quantity such as height, pressure, temperature or wavelength. For example, wind speed is often reported for a reference height of 10 meters above the ground. The *standard refraction index* for light traveling through air is based on a specific reference wavelength of 589 nm. Standard temperature and pressure (stp) is another example. Separate names should allow either the reference value itself (e.g. 10 m or 589 nm) or the value of a named quantity at the reference value to be retrieved.

Support an arbitrary number of assumptions for each name. In order to completely specify how a model or data set interprets or uses a given standard name, it may be necessary to provide a list of assumptions (i.e. clarifications, explanations, conditions, exceptions, provisos, measurement units, etc.). These may apply to either the object or quantity part of a name. Including these assumptions in the name itself can lead to unwieldy names or the omission of assumptions. As valuable metadata, it should be possible to associate an arbitrary number of assumptions with a given name.

3 The CSDMS Naming Conventions

Although the CSDMS Standard Names are still a work in progress, their naming conventions already address most of the design criteria described in the previous section. Work is underway with several groups to develop standard names for different geoscience domains (e.g. hydrology, oceanography, meteorology, seismology) and to align them with those used by other organizations, modeling frameworks and published data sets.

The naming conventions of the CSDMS Standard Names are based on object-oriented principles. Each name is a unique string that has an *object part* and a *quantity part*.

$$\text{standard variable name} = \text{object name} + \text{quantity name}. \qquad (1)$$

This name is used by framework software to retrieve corresponding numerical values stored either in RAM or in a file. The word *quantity* is used to describe an attribute that can be *quantified*, usually with one or more numbers. The EAV (Entity Attribute Value) data model is closely related, except that here the entity (object) and attribute (quantity) are encoded as standardized strings and then joined (with double underscores) to create a *unique label* or *key* that is used to access a stored number or array of numbers. While the object and quantity names are simple strings, the numbers are stored as binary data and may have a complex data structure, such as a 3D array of temperature values or a 3D vector field of velocities that span the model’s computational grid.

Both the object part and quantity part of the name must contain enough adjectives and modifiers (separated by underscores) to eliminate ambiguity and identify a specific object and a specific quantity associated with that object. In the English language, adjectives and modifiers are added to the left of an object name (a noun) to describe an object more fully. This is the case for both object names (e.g. tree, oak tree, bluejack oak tree) and quantity names (e.g. conductivity, hydraulic conductivity, saturated hydraulic conductivity, effective saturated hydraulic conductivity). Usually, no more than four adjectives are required to remove ambiguity. By contrast, when referring to a part of a larger (or containing) object, it is common to start with the name of the largest or outermost object and then add the names of nested parts to the *right* (e.g. bicycle wheel spoke, truck engine cylinder). These *subobjects* may require their own adjectives (on the left) for clarification. Software capable of distinguishing between adjectives and nouns would then be able to break such a name into its subobjects. When an object name itself requires more than one word and both are nouns, as in “carbon dioxide”, a hyphen or other special character can be inserted to make it possible for software to distinguish between a single object and two nested objects. However, this convention has not yet been adopted by CSDMS.

The object part is necessary to avoid ambiguity since the same quantity (e.g. temperature) may be applied to several different objects (e.g. air, snow, soil, water), even within a single model. This could be an object in the real world for which data has been collected and stored (i.e. in a file), or an object that appears in a computer model. These naming conventions can therefore be used for either data sets or models.

A complete description of the CSDMS naming conventions cannot be given in the space available here. Many reusable patterns have been identified by analyzing variable names from many different science domains and models. See the CSDMS website: csdms.colorado.edu/wiki/CSDMS_Standard_Names for details. Topics are divided into separate sections such as Basic Rules, Object Name Templates, Quantity Name Templates, Operation Templates, Process Names and Assumption Names. Links to numerous Wikipedia articles provide additional, supporting or background information. Any number of standardized assumption names can be associated with a CSDMS standard variable name by using <assume> tags in an XML-based, CSDMS Model Metadata File.

3.1 How the Standard Names are Used

CSDMS uses the fourth approach in Section 1.1 to achieve automated semantic mediation. Contributors of models or data sets are asked to map each of their own terms to the appropriate "long name" in the list of CSDMS Variable Names. For models or data sets this can be done by implementing a CSDMS Basic Model Interface (BMI), which means adding some standardized functions to the model code, or a model-type wrapper to a data set (Peckham et al., 2013). The BMI functions provide self-description, variable getters and setters and model control functions (i.e. initialize, update, finalize). The developer maps each of their model's internal variable names to a standard name, then uses that mapping to write BMI getter and setter functions that can be called by the modeling framework using a CSDMS Variable Name as the argument. The getter functions return the value(s) of the requested variable. Two other BMI functions (i.e. *get_input_var_names* and *get_output_var_names*) return standard name lists of the model's input and output variables so the framework knows what variables the model can provide or needs. The developer **does not** change the variable names used in the model source code. While this requires a modest amount of additional work on the part of the model developer, it only requires (1) finding a matching CSDMS Variable Name for each of the model's input and output variable names, (2) writing a small amount of new code as opposed to making changes to existing code and (3) possibly saving additional assumptions for some variable names as XML in a Model Metadata File. This is a one-time cost that provides the highest possible accuracy for semantic matching. Once completed, all future semantic mediation that is required to couple the model to other (similarly prepared) models or data sets can be fully automatic and this is a major benefit for users.

3.2 Standardized Process Names

Much of science is concerned with the study of natural and physical processes, so perhaps it should not be surprising that a large number of **quantity names** are constructed from a process name. From this work, it became clear that **process names** can be viewed as *nouns derived from verbs*, usually by adding one of these word endings:

tion	(e.g. absorption, convection, radiation)
sion	(e.g. conversion, dispersion, submersion)
ing	(e.g. melting, swimming, upwelling)
age	(e.g. drainage, seepage, storage)
y	(e.g. discovery, recovery, reentry)
ance	(e.g. acceptance, disturbance, maintenance)
ment	(e.g. alignment, improvement, recruitment)
al	(e.g. arrival, disposal, removal, retrieval) and
sis	(e.g. osmosis, metamorphosis, dialysis, paralysis).

However, for process names that end in "ing", it is common to drop the ending, as in: *flow_rate, lapse_rate, melt_rate, shear_stress, start_time* and *tilt_angle*. Most process names can be paired with "_rate" to create a quantity name that indicates how fast the process occurs (e.g. *infiltration_rate* and *evaporation_rate*). Other examples of this process name + quantity name pattern are: *dilution_ratio, drainage_area, escape_speed, gestation_period, identification_number, inclination_angle, penetration_depth, protection_factor, relaxation_time, return_period, striking_distance, turning_radius* and *vibration_frequency*. A collection of 525 examples of this pattern and over 1300 standardized process names can be found at: http://csdms.colorado.edu/wiki/CSN_Process_Names.

3.3 Standardized Operation Names

Mathematical operations may be applied to quantity names to produce new quantity names (usually with new units). A standardized list of **operation names** has been created to support the CSDMS Variable Names. In a standard variable name, the reserved word **of** is inserted after each operation name. Multiple operations can be composed (that is, chained together) when necessary, as in: *time_derivative_of_northward_component_of_velocity*. This use of the word **of** mirrors spoken math and also serves as a delimiter for extracting operation names from a quantity name. Operation names can be grouped into the following categories:

Time Derivatives. e.g. *time_derivative, 2nd_time_derivative*.

Spatial Derivatives. e.g. *eastward_derivative, northward_derivative, upward_derivative, x_derivative, y_derivative, z_derivative, offshore_derivative, longshore_derivative*.

General Derivatives. e.g. *Y_derivative*, where Y can be a one-word base quantity such as temperature or pressure, or can use multiple words when necessary.

Space and Time Integrals. e.g. *time_integral, area_integral, area_time_integral, volume_integral, line_integral*.

Functions of One Variable. e.g. *abs, anomaly, cos, exp, increment, limit, log, log10, minus, sin, sgn, sqrt, square, tan, tanh, 3rd_power, 4th_power*. Note that *increment* would be used for the (positive or negative) amount by which a quantity changes, perhaps after one model time step.

Statistical Operations. e.g. *max, mean, median, min, mode, standard_deviation, variance*. Similar to the pattern used for derivatives and integrals, terms like domain_max or time_max (e.g. peak discharge) can be used to indicate how the statistic is computed. (Standardized assumptions can also be used.)

Operations on Vectors that Return Scalars. e.g. *azimuth_angle, zenith_angle, divergence, laplacian, magnitude* and *X_component* (where X may be cross_stream, down_stream, downward, eastward, northward, offshore, longshore, u, v, x, y, or z). Some operations act on two vector quantities and return a scalar, such as: *cross_product_of_X_and_Y, dot_product_of_X_and_Y* and *dot_product_angle_of_X_and_Y*.

Operations that Return Vectors. e.g. *curl, opposite* (of vector) and *gradient* (of scalar).

Operations on Tensors that Return Scalars. e.g. *first_invariant (or trace), second_invariant*.

3.4 Examples from the CSDMS Standard Variable Names

atmosphere_carbon-dioxide partial_pressure
atmosphere_water liquid_equivalent_precipitation_rate
bedrock_surface time_derivative_of_elevation
channel_bed manning_coefficient
channel_cross-section wetted_perimeter
channel_cross-section width_to_depth_ratio
earth_axis tilt_angle
earth_ellipsoid equatorial_radius
earth_orbit eccentricity

earth_mean-sea-level-datum_air__pressure
glacier_glen-law__exponent
lake_water__volume_inflow_rate
land_surface__diffuse_shortwave_irradiation_flux
land_surface__longwave_radiation_flux
rubber_concrete__kinetic_friction_coefficient
sea_water__northward_component_of_velocity
soil_clay__volume_fraction
soil__saturated_hydraulic_conductivity
water_suspended-sediment__volume_concentration
watershed_outlet_water__time_max_of_volume_outflow_rate

3.5 Standardized Assumption Names

In the CSDMS Standard Names, the word **assumption** is taken as a broad term that can include things like *conditions, simplifications, approximations, limitations, conventions, provisos* and other forms of clarification. Every process-based model is built from many such assumptions and simplifications. When a scientist describes the details of a model to another scientist, many fairly standardized and widely recognized phrases are used. The CSDMS Assumption Names attempt to capture these. Several major categories have been identified, with many examples in each. In the near future, a "smarter" CSDMS modeling framework will use this **standardized model metadata** to help users determine whether, and the degree to which, different models in its repository are compatible for coupling.

Model metadata categories include: **Boundary conditions:** (e.g. no slip boundary condition), **Conserved quantities:** (e.g. mass, momentum, energy), **Coordinate system:** (e.g. Cartesian coordinate system), **Angle conventions:** (e.g. clockwise from north convention), **Dimensionality:** (e.g. 2 dimensional), **Equations used:** (e.g. Navier Stokes equation), **Closures:** (e.g. eddy viscosity turbulence closure), **Flow-type assumptions:** (e.g. laminar flow, hydraulically smooth flow), **Fluid-type assumptions:** (e.g. compressible fluid, Herschel-Bulkley fluid), **Geometry assumptions:** (e.g. trapezoid shaped), **Named model assumptions:** (e.g. Green-Ampt infiltration model), **Thermodynamic processes:** (e.g. isenthalphic process), **Stochastic processes:** (e.g. Poisson event process), **Named approximations:** (e.g. Boussinesq approximation), **Averaging methods:** (e.g. Reynolds averaged, depth averaged), **Grid assumptions:** (e.g. Arakawa C grid), **Numerical methods:** (e.g. leapfrog method, implicit method) and **State of matter:** (e.g. liquid phase).

4 Comparison to Two Other Naming Systems

4.1 CF Standard Names

The CF (Climate and Forecasting) Standard Names are used in the domain of ocean and atmosphere modeling, for labeling output variables from models that have been saved into NetCDF files (CF Metadata Group, 2011). The Earth System Modeling Framework (ESMF) uses a subset of these names (Dunlap et al., 2008) to support model coupling (e.g. 14 names in the NUOPC Field Dictionary). They have also been used for IPCC (Intergovernmental Panel on Climate Change) studies.

While these names are intended to be unambiguous and support mathematical operations (called transformations in CF, and applied to the whole name), they only meet a few of the design criteria described previously. To be specific, they (1) use domain-specific terminology (e.g. "tendency" instead of "time derivative"), (2) have complicated construction guidelines (vs. rules) that are not applied consistently, (3) they don't include provisions for dimensionless numbers, mathematical and physical constants, empirical parameters or reference quantities, (4) they do not have a natural grouping of related names and (5) they include assumptions in the name using "assuming_".

There are currently 2514 CF standard names, with 183 new proposed names under review (some since 2010). The names have been grouped into categories with Atmospheric Chemistry as the largest category with 968 names. However, a large fraction of these result from selectively applying 8 to 26 quantity name patterns (e.g. *atmosphere_mass_content_of_X*) to each of about 90 chemical species (e.g. ammonia, benzene, carbon dioxide, toluene, xylene.) There are 661 names that include the *due_to* construct to identify one possible source contribution to a given quantity. Other constructs that are meant to capture an assumption are: *expressed_as* , *assuming* and *excluding_* and are used in 197 names. Mathematical operations (called transformations) are supported but are used in very few names, except for *tendency_of* (i.e. time derivative), used in 690 names. Other name categories are Atmospheric Dynamics (146), Cloud (128), Hydrology (176), Ocean Dynamics (32), Radiation (179), Sea Ice (64) and Surface (229). The purpose of this summary is not to be overly critical, but rather to clarify the relatively narrow scope of these names, despite their number and widespread use. A crosswalk between these and the CSDMS Standard Names is under development.

4.2 CUAHSI Variable Name Controlled Vocabulary

The CUAHSI Controlled Vocabulary (CV) is composed of 13 separate CVs (CUAHSI, 2011). Together, these provide standardized names that can be used to populate fields in tables that support CUAHSI ODM v1.1 (Observational Data Model). The *VariableNameCV* provides 614 standardized names that support hydrologic observations (or measurements). All but about 136 of these names are for water chemistry and biota. Like many CVs, these are very much a work in progress and are designed to be extended by the hydrologic community. However, they currently meet very few of the design criteria discussed previously. They also contain very few names for branches of hydrology such as snow hydrology, open channel flow, infiltration theory, evaporation, soil physics, glaciology and hydrometeorology. Many terms are domain-specific and not precise enough to be used for semantic mediation across heterogeneous resources. See Zaslavsky et al. (2012) and references therein.

5 Conclusions and Future Work

The problem of semantic mediation between heterogeneous resources (e.g models and data sets) has been examined and articulated as a set of design criteria for a desired solution. A set of cross-domain naming conventions called the CSDMS Standard Names was described that meets these design criteria. While still a work in progress, a more complete description can be found on the CSDMS website at: csdms.colorado.edu/wiki/CSDMS_Standard_Names. This work has expanded beyond naming conventions for variable names to include a set of standardized **model assumption names**. These are divided into several categories and provide standardized **model metadata** that allows the science and assumptions of a model to be described in detail. Future, smart frameworks could use this metadata to determine whether, and the degree to which, different models are compatible for coupling.

Acknowledgments

Funding for this work by NSF grant #1226297 (CSDMS) and NSF EarthCube grants #1343811 (Earth System Bridge) and #1343800 (GeoSoft) is gratefully acknowledged.

References

CF Metadata Group (2011) NetCDF Climate and Forecast (CF) Metadata Convention-CF Standard Names, http://cf-pcmdi.llnl.gov/documents/cf-standard-names/. Accessed March 2014.

CUAHSI (2011) Master Controlled Vocabulary Registry for ODM 1.1, http://his.cuahsi.org/mastercvreg/. Accessed March 2014.

Dunlap, R.L., L. Mark, S. Rugaber, V. Balaji, J. Chastang, L. Cinquini, C. DeLuca, D. Middleton and S. Murphy (2008) Earth System Curator: Metadata infrastructure for climate modeling, *Earth Science Informatics*, **1**(3–4), 131–149.

Gregersen, J.B., P.J.A. Gijsbers and S.J.P. Westen (2007) OpenMI: Open modelling interface, *J. Hydroinformatics*, 09.3, 175–191.

Peckham, S.D., E.W.H. Hutton and B. Norris (2013) A component-based approach to integrated modeling in the geosciences: The Design of CSDMS, *Computers & Geosciences, special issue: Modeling for Environmental Change*, **53**, 3–12.

Zaslavsky, I., D. Valentine, R. Hooper, M. Piasecki, A. Couch, A. Bedig (2012) Community practices for naming and managing hydrologic variables, *AWRA 2012 Spring Specialty Conf., March 26-28*.