

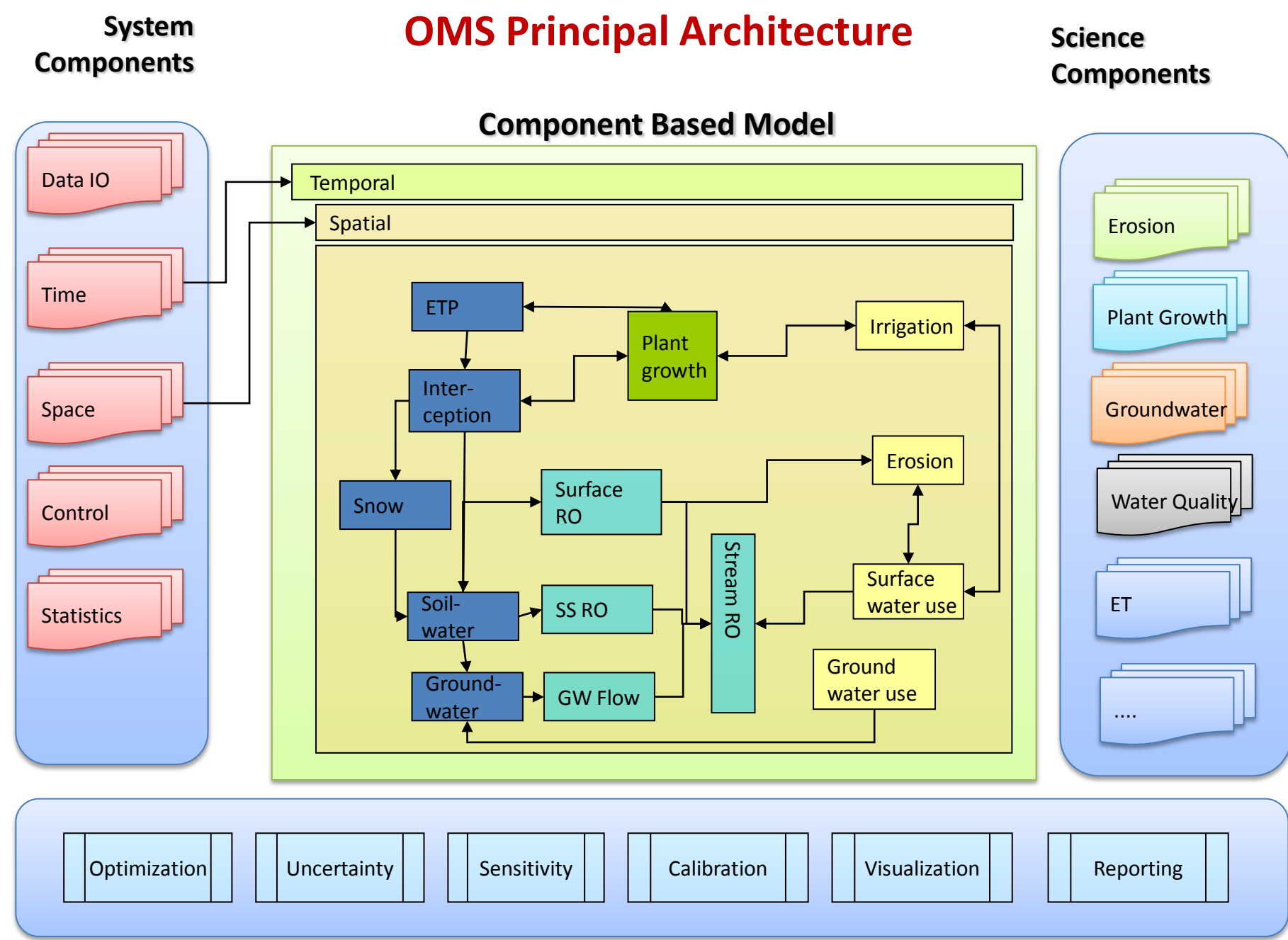
Introduction

The Object Modeling System (OMS) is a framework for agro-environmental model development, data provisioning, testing, validation, and deployment. It provides a bridge to transfer technology from research to program delivery. A modeling team uses the framework to conceive and design models, build science-components, aggregate components into models, calibrate and validate them, provision the models with data, and deploy them for widespread use. OMS contains four platforms:

- 1) Model development and validation
- 2) Data provisioning
- 3) Model services deployment
- 4) Knowledge base and metadata management

OMS was developed by USDA-ARS and CSU, released in 2004, and integrated into the USDA-NRCS technical architecture in 2008. Features of current version OMS 3.0 include:

- Non-invasive data-driven architecture
- Language interoperability (Java, Fortran, C, C++)
- Many calibration and validation tools
- Cloud services deployment infrastructure

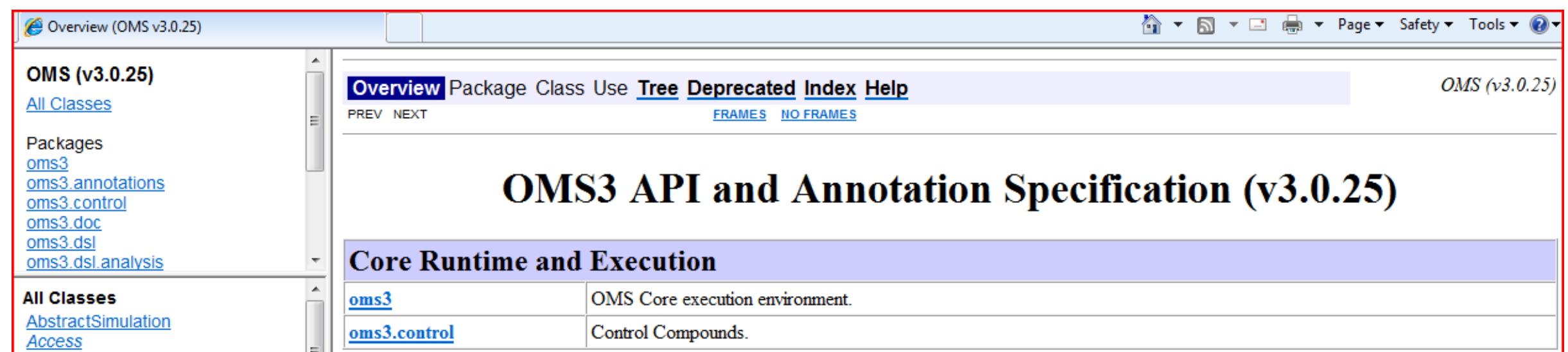


OMS currently supports NRCS water supply forecasting, next generation ARS CEAP model AgES-Watershed, deployment of the RUSLE2 model service for the NRCS Conservation Delivery Streamlining Initiative (CDSI) and Keystone Center Field to Market, IAEA Nile River Basin water balance modeling, CSU Environmental Risk Assessment Management System (eRAMS), Precipitation and Runoff Modeling System (PRMS) integration with the NOAA-NWS Community Hydrologic Prediction System (CHPS), and hydrologic modeling by HydroloGIS (Italy). OMS envisions support to six primary model bases:

- 1) AgES-Watershed (in progress)
- 2) AgES-Farm/Field (planned)
- 3) AgES-Forecasting (in progress)
- 4) AgES-Environmental Services (planned)
- 5) AgES-Conservation Practice Design (in progress)
- 6) AgES-Research (in progress)

Challenge

Most agro-environmental models focus on few natural resource concerns, whereas conservation programs usually address many concerns, forcing conservationists to use multiple models. The models are stovepipes often with redundant processes and data. Models can be integrated by (1) creating one really big tightly coupled stovepipe model, (2) chaining models together, or (3) creating a model base of science components aggregated into model instances. As models or model bases increase in size and complexity to address a greater number of resource concerns, understanding model structure, concepts, and relationships in a systematic way becomes crucial. Currently, most agro-environmental models analyze less than 5 resource concerns and the most comprehensive, 10-15 resource concerns. USDA-NRCS has streamlined their resource concern list for conservation planning and program eligibility from 80 plus to 32 concerns. Despite this reduction, the most effective way to design and build a science model service infrastructure to support analysis of these concerns remains a challenge. The OMS knowledge base platform can facilitate the integration of models and components into the model bases supporting analysis of a large number of resource concerns. This poster summarizes progress with the knowledge base platform to date.



OMS Knowledge Base Platform Description and Objectives

The OMS knowledge base platform consists of tools for systematically understanding the science concepts contained in a model or model base, the structure of the model, and how it operates. The tools include concept models in Web Ontology Language (OWL), class and sequence diagrams in Unified Modeling Language (UML), entity relationship diagrams (ERD), and auto-generated documentation from model packages. The tools can be used to analyze existing models or to conceptualize and design new models and components.

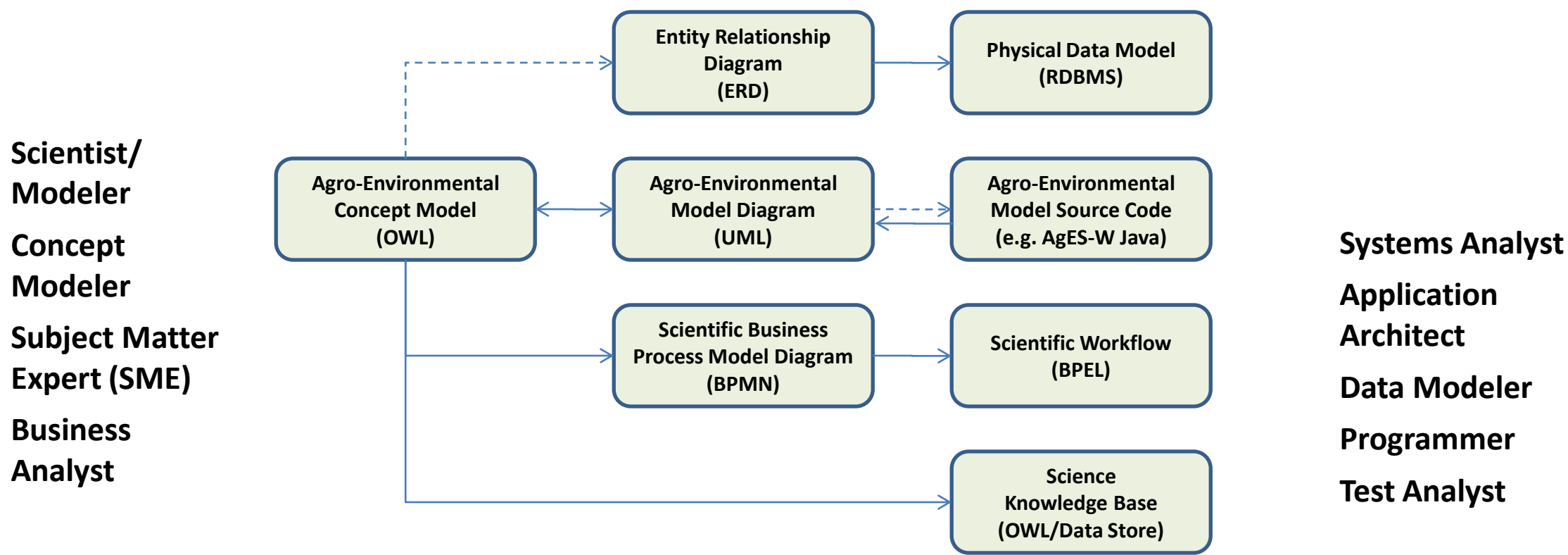
The OMS model knowledge product suite, as an example for the Java Thornthwaite model, would contain Thornthwaite.jar (model), Thornthwaite.src (source code), Thornthwaite.uml (as-built model structure and design), Thornthwaite.owl (concept model), Thornthwaite.erd (as-built entity relationship model, if applicable), and Thornthwaite.pdf (auto-generated documentation also in .html or .doc format).

The objectives of the OMS knowledge base platform are to help:

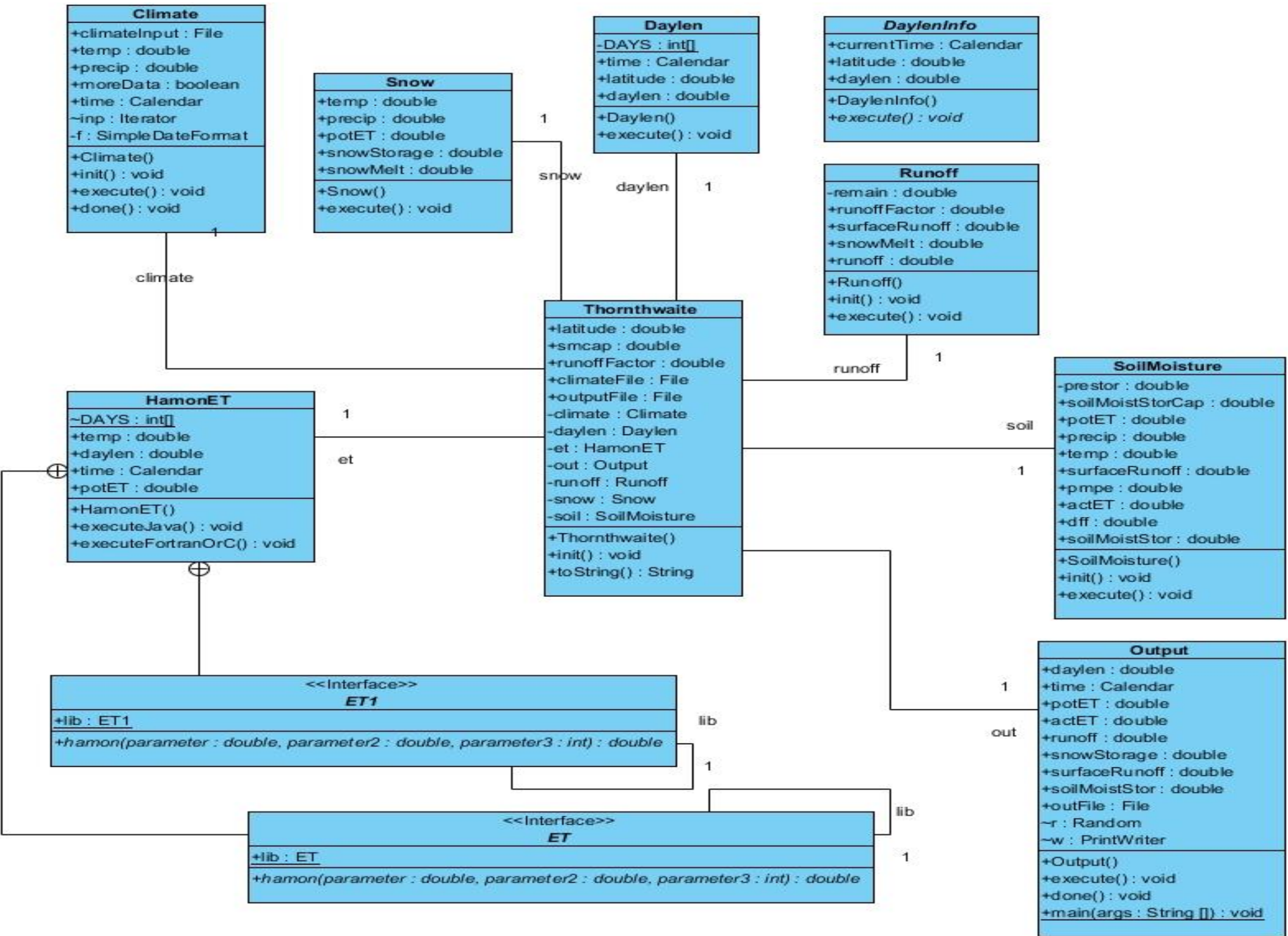
- 1) Capture concepts, attributes, properties, and relationships in existing models (as-built knowledge documentation)
- 2) Harmonize concepts, attributes, properties, and relationships across models within a business domain
- 3) Conceptualize and design new models and components and enhancements to models and model bases.

Concept Modeling

One method increasingly used to develop and understand the concepts for a biophysical model is to create an ontology (concept model) using description logic (DL), a family of formal knowledge representation languages, specifically OWL developed for the semantic web (to infuse intelligence into web pages). Using an ontology editor, such as Protégé from Stanford University, a concept modeling session can establish and annotate classes corresponding to scientific concepts, and object and data properties reflecting relationships among concepts. Classes, their object and data properties can carry forward into UML as classes, associations, and attributes where the model structure and design takes shape. The knowledge codified in OWL also can be useful as a reference for creating entity relationship diagrams for database design. The OWL concept model can carry forward to Business Process Management Notation (BPMN) for workflow design. And the ontology can be fleshed out and reserved for use in a knowledge base application. The concept model can serve as a quality gate for model integrity.



Ontology can help modeling communities manage the complexity of creating and linking hundreds of concepts and thousands of variables and parameters. The OWL to UML (and UML to OWL) transition can help bridge understanding among subject matter experts and computer programmers..

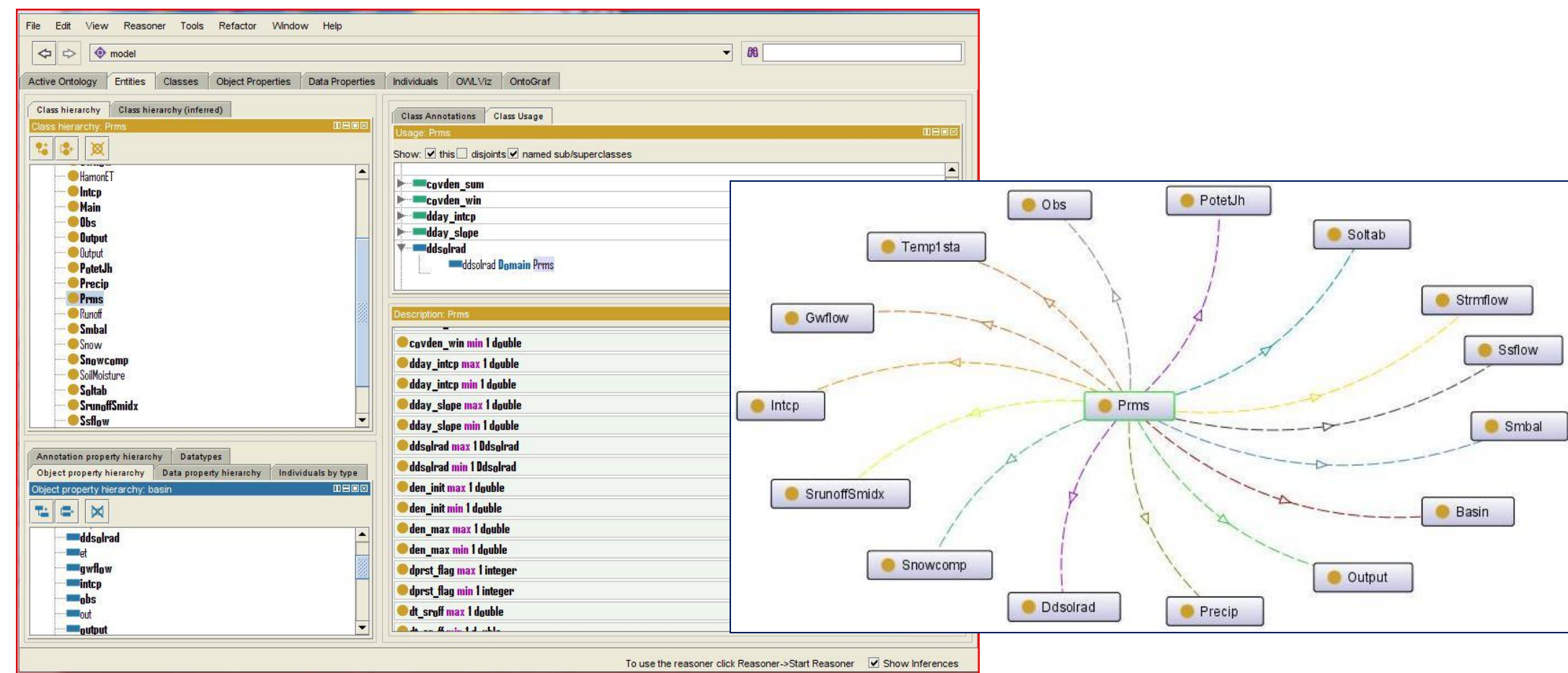


Thornthwaite Water Balance Model
Reversed Engineered UML Class Diagram

Tools and Process

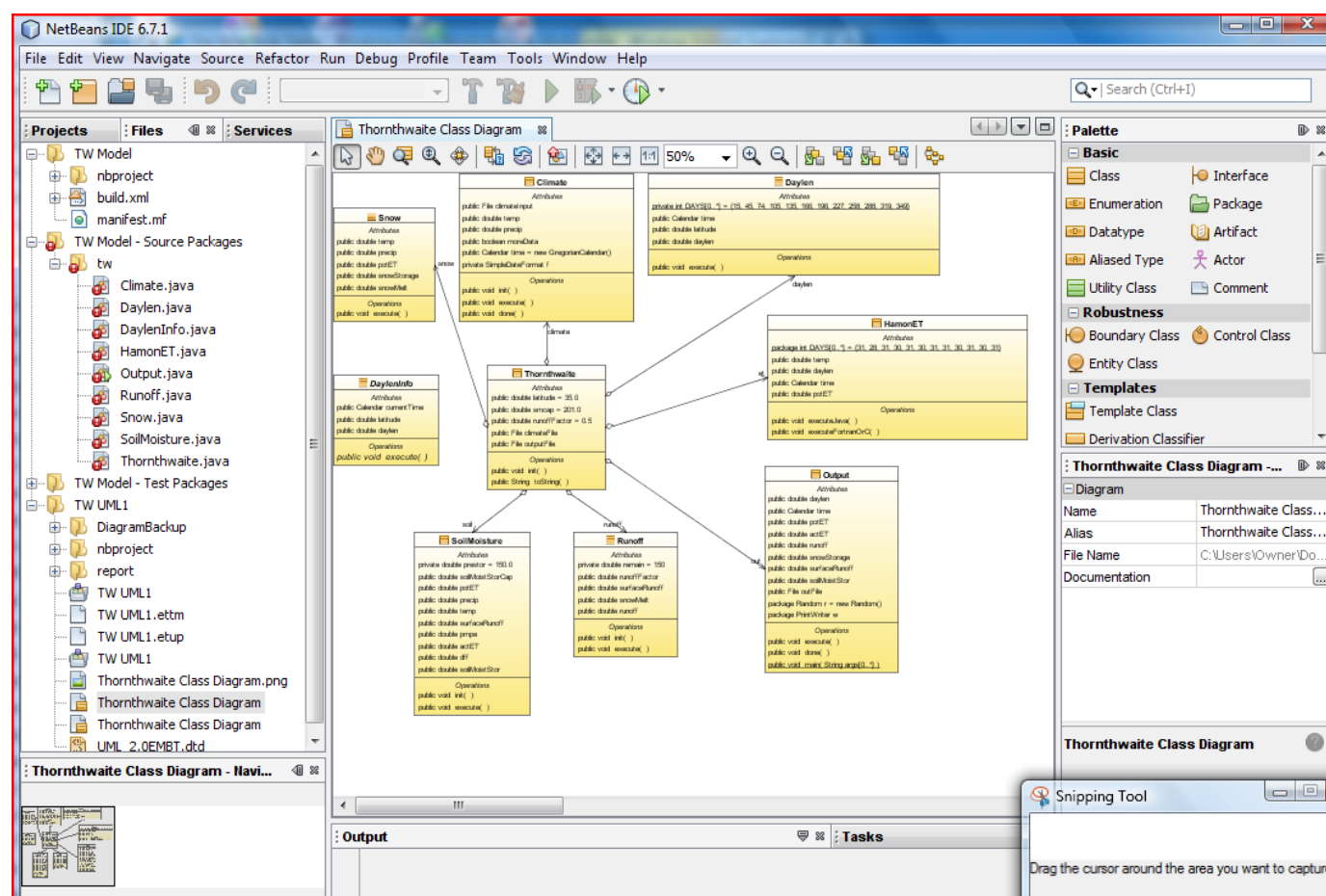
The following tools have been used to analyze and create knowledge about OMS developed models:

- 1) Protégé Ontology Editor 4.0.2 or 4.1b (creating and modifying concept models in OWL)



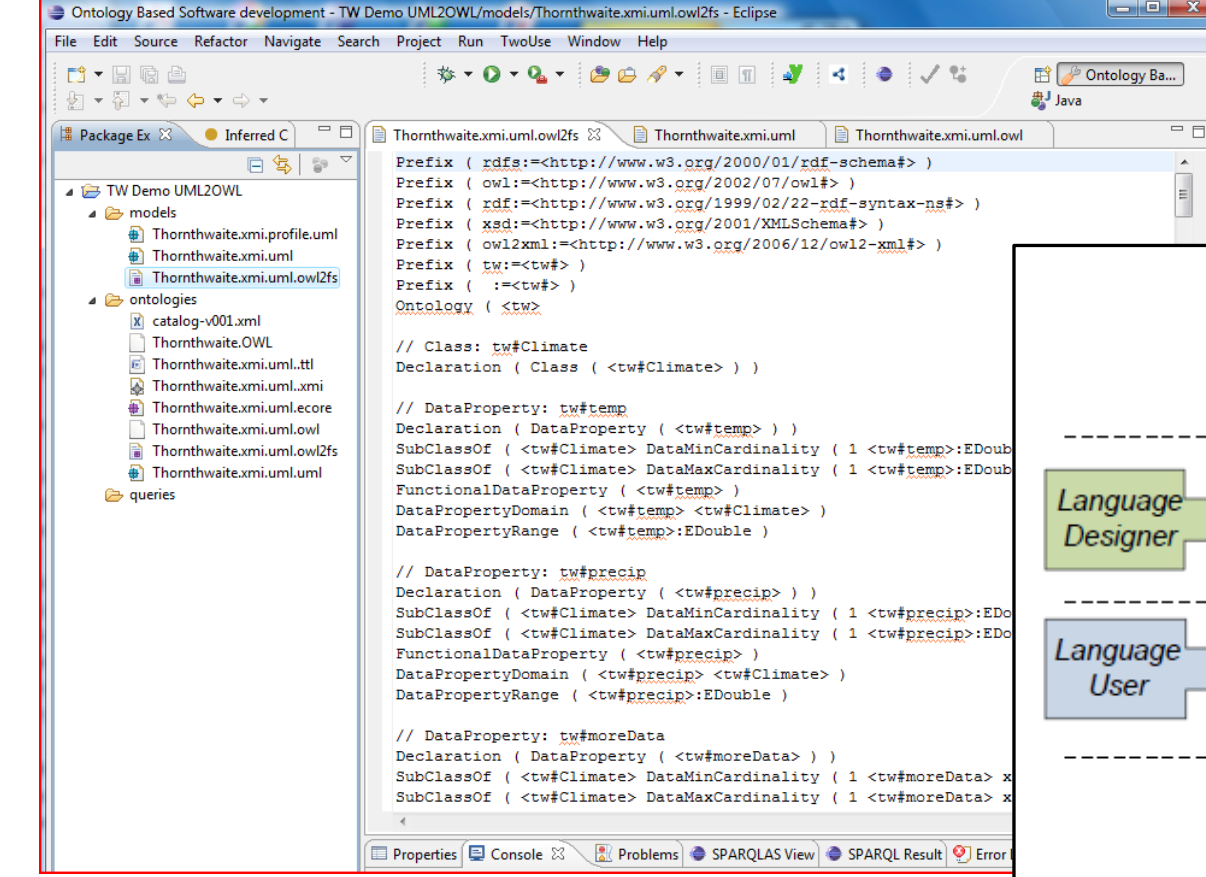
- 2) Netbeans 6.7.1 UML Plug-in or commercial Visual Paradigm UML 8.0 (reverse engineering existing models to UML)

Example: Thornthwaite.jar → Thornthwaite.uml → Thornthwaite.xmi

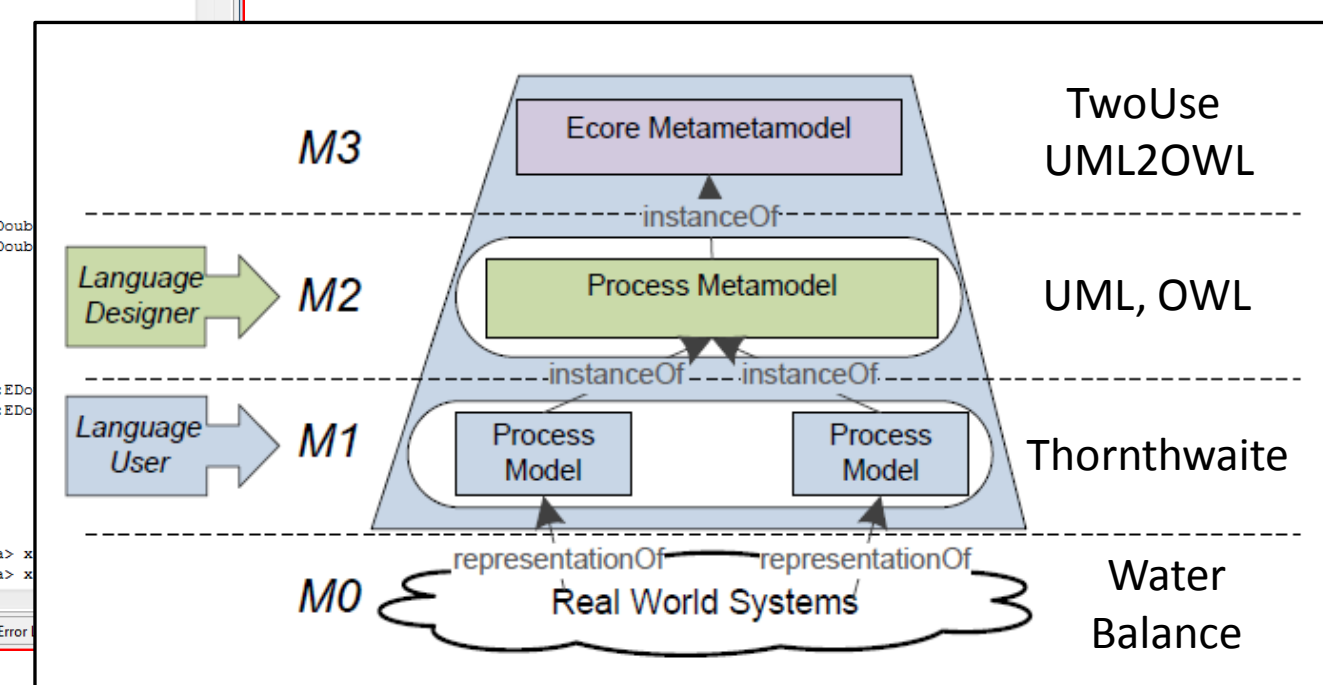


- 3) Eclipse TwoUse Plug-in (transforming UML to OWL)

Example: Thornthwaite.xmi → Thornthwaite.owlfs → Thornthwaite.owl

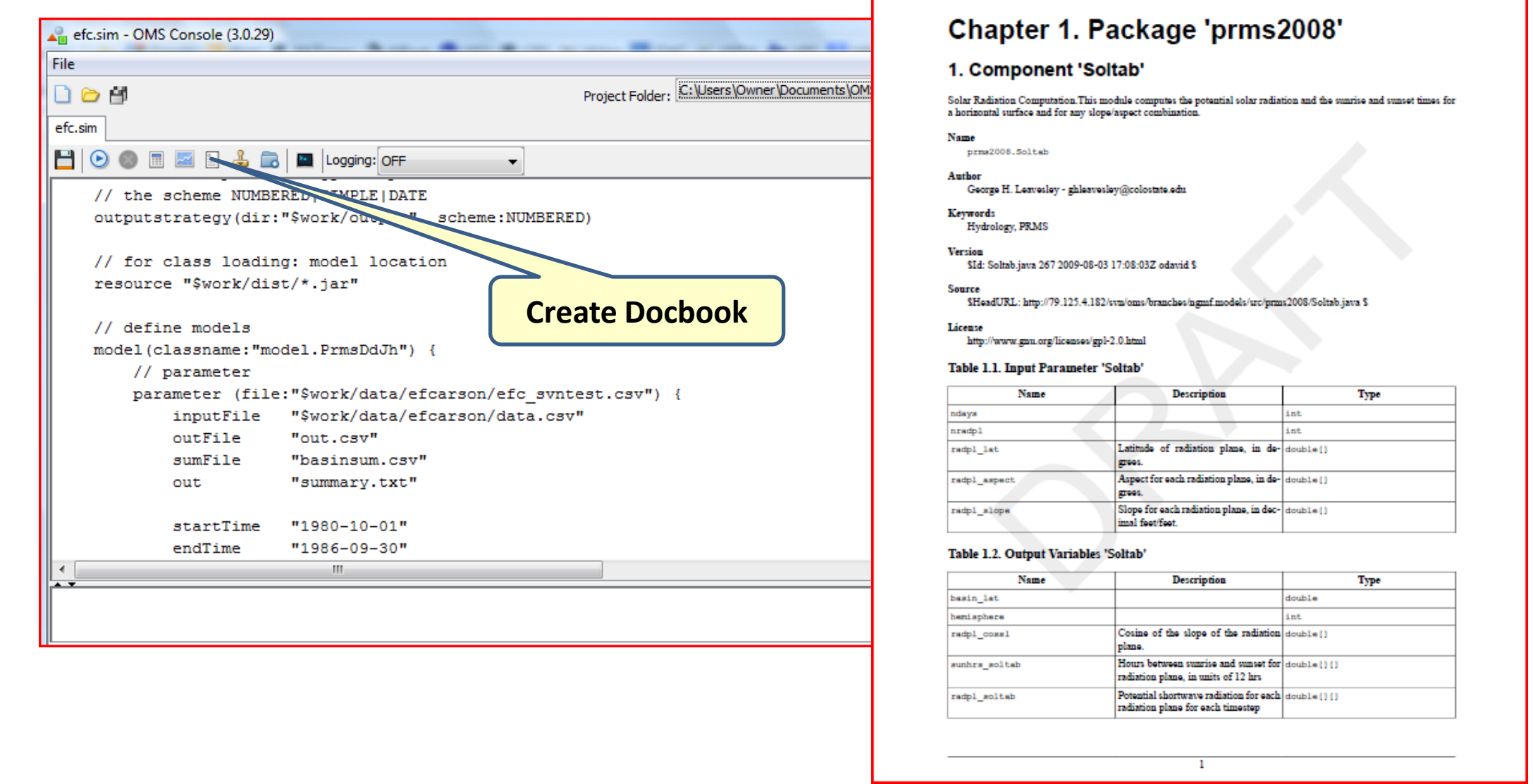


After Staab et al, 2010

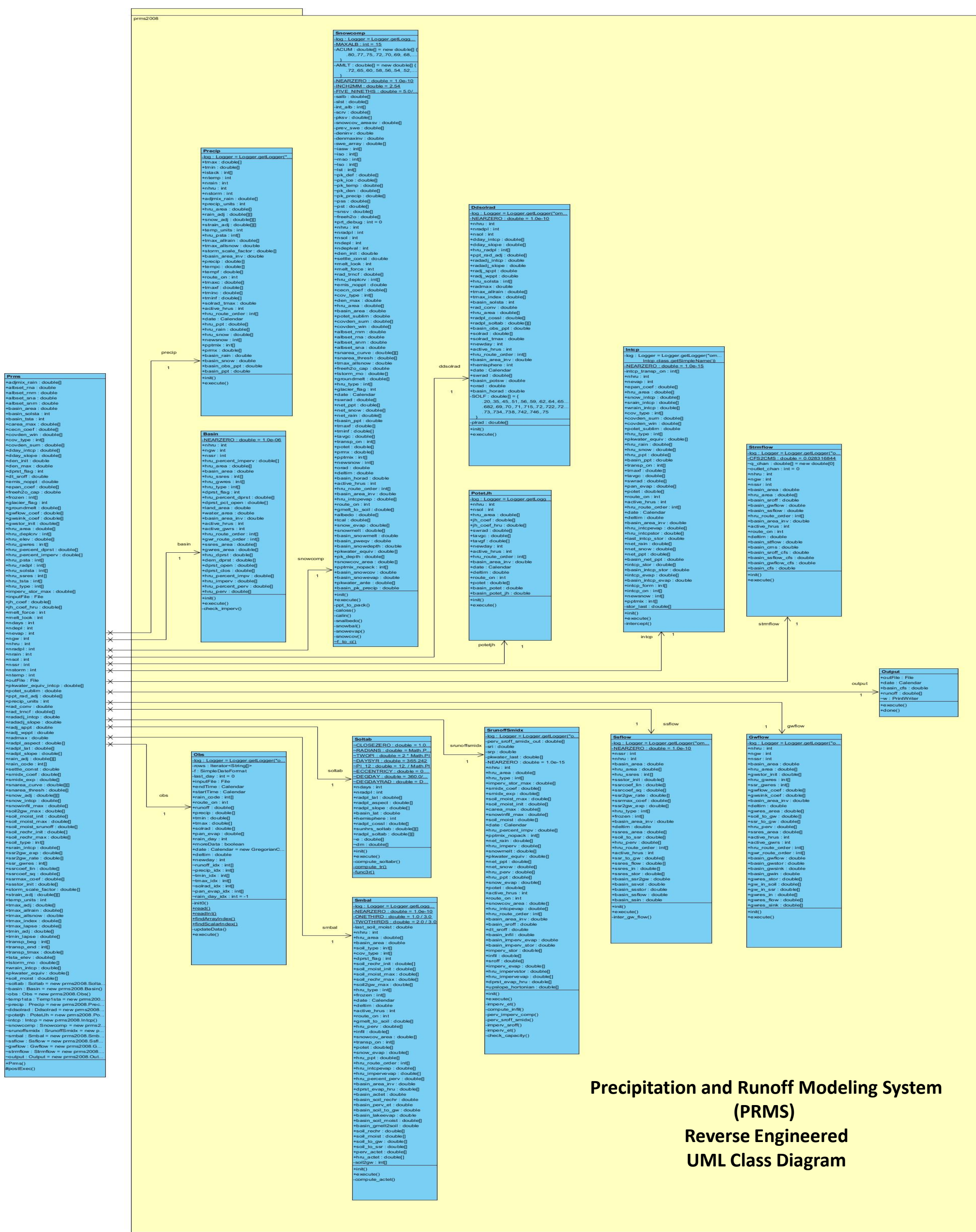


- 4) OMS Auto-Documentation (extracting knowledge from existing models)

Example: prms2008.jar → prms2008.xml → prms2008.pdf



The primary requirement for reverse engineering existing models is a component-based design. Models written in Fortran or C should contain a component design and JNA-binding.



Precipitation and Runoff Modeling System
(PRMS)
Reverse Engineered
UML Class Diagram

Discussion

Current tools efficiently reverse engineer OMS models to UML, although expert manipulation often is required to properly display UML content from large models in diagrams, such as AgES-Watershed. UML to OWL transformations are trickier. Model package by ontology interaction can preclude complete TwoUse transformation of UML to OWL. The OMS team and the TwoUse team in Germany are communicating to work through problems encountered in transformations. Netbeans UML product currently does not format to the UML syntax required for TwoUse transformation. Automated OWL to UML transformation processes have not yet been worked out for OMS requirements. Protégé diagramming plug-ins need enhancement to meet OMS needs. Also enhancements probably are needed to better express OMS model data flows in UML output. Nevertheless, these tools are useful in their current state for eliciting and conserving knowledge in OMS models.

References

- Anthanasiadis et al. 2009. Ontology for seamless integration of agricultural data and models. Proc. 3rd International Conference on Metadata and Semantics Research (MTSR'09), Milan, Italy, 282-293.
- Parreiras and Staab. 2010. Using ontologies with UML class-based modeling: the TwoUse approach. Data and Knowledge Engineering 69(11):1194-1207.
- W Staab et al. 2010. Model driven engineering with ontology technologies. Proc. 6th International Conference on Semantic Technologies for Software Engineering (ReasoningWeb'10), Technical University of Dresden, Germany, 62-98.
- WC3. 2009. OWL 2 Web Ontology Language Primer. Accessed 03-22-2011 at <http://www.w3.org/TR/2009/REC-owl2-primer-20091027/#Datatypes>

Contacts

- Jack R. Carlson (Jack.Carlson@colostate.edu)
Olaf David (ODavid@colostate.edu)
George H. Leavesley (GHLLeaves@rams.colostate.edu)
Department of Civil and Environmental Engineering, Colorado State University, Fort Collins, Colorado
- James C. Ascough II (Jim.Ascough@ars.usda.gov)
USDA-ARS Agricultural Systems Research Unit, Fort Collins
Ken W. Rojas, (Ken.Rojas@frc.usda.gov)
USDA-NRCS Information Technology Center, Fort Collins