BUREAU OF RECLAMATION

Technical Memorandum

Water Temperature Modeling Platform: Model Framework Selection and Design (DRAFT)

Central Valley Project Water Temperature Modeling Platform

California-Great Basin Region

U.S. Department of the Interior **June 2021**

Mission Statements

The U.S. Department of the Interior protects and manages the Nation's natural resources and cultural heritage; provides scientific and other information about those resources; and honors its trust responsibilities or special commitments to American Indians, Alaska Natives, and affiliated Island Communities.

The mission of the Bureau of Reclamation is to manage, develop, and protect water and related resources in an environmentally and economically sound manner in the interest of the American public.

Water Temperature Modeling Platform: Model Framework Selection and Design (DRAFT)

Central Valley Project Water Temperature Modeling Platform

California-Great Basin Region

prepared by

United States Department of the Interior Bureau of Reclamation

California-Great Basin

With Technical Support by:

Watercourse Engineering, Inc.

Cover Photo: Keswick Dam on the Sacramento River by John Hannon

Contents

Contents

Figures

Tables

Abbreviations and Acronyms

API	Application Programming Interface
CVP	Central Valley Project
GUI	Graphical User Interface
I/O	Input/Output
IT	Information Technology
QA	Quality Assurance
Reclamation	U.S. Department of the Interior, Bureau of Reclamation
TCD	Temperature Control Device
TMDL	Total Maximum Daily Load
TMP	Temperature Management Plan
WTMP	Water Temperature Modeling Platform

Chapter 1 Introduction

Flow and water temperature simulation models are useful and necessary tools to support resource managers in their understanding of the temperature dynamics in U.S. Bureau of Reclamation (Reclamation) CVP reservoirs and downstream river reaches. Such tools support evaluation of how operational decisions and various influencing factors can affect water temperature in reservoirs and rivers, as well as the resulting potential impacts to fishery species sensitive to water temperature. The improvement of models, modeling approach, and associated tools to support operational decision making is considered a necessary adaptation strategy that takes advantage of ongoing technological advancement, additional information, and data. Reclamation's objective for the development of the Water Temperature Modeling Platform (WTMP) is the effective and efficient management of resources for downstream regulatory and environmental requirements within the context of an uncertain environment. A primary development goal of the WTMP is to provide realistic predictions of downstream water temperatures with sufficient confidence to carry out the necessary planning for seasonal, real-time, and long-term study applications while also describing situational risk and uncertainty.

Models of large complex reservoir-river systems have been developed for a wide range of applications (DeGeorge et al. 2018, USACE 2016, Goode at al. 2010, Modini 2010). Reservoir and river reaches can be modeled as discrete components, with individual models for each reservoir or river reach, or with a modeling system as an interconnected network. A modeling system is a single software package (e.g., HEC5Q (HEC 1999, HEC 2000) or HEC ResSim (HEC 2021b)) that incorporates all system components (e.g., discrete reservoirs and river reaches) and their inter-connections. However, a single model may not represent all potential characterizations desired by resource managers (Buahin and Horsburgh 2018).

A framework is a software application, or set of applications, that can be used to streamline model use and automate repetitive tasks, making the modeling process more efficient and more robust (David et al. 2013, David et al 2010). A modeling framework (framework) provides a means to represent reservoir-river systems as either a suite of linked but discrete models, with a modeling system, or a combination of the two approaches (Buahin and Horsburgh 2018, Elliot et al. 2014, Petty 2014). Frameworks can range from a fixed set of linked models that are run in sequence (passing information from upstream reaches to downstream reaches) to a complex set of interchangeable models that include operational consideration (e.g., operating rules/targets, capacities, demands) (Clark et al. 2015). For the CVP, there is a need for both high resolution, discrete reservoir and/or river element models that can represent more detailed representations, as well as a modeling system that can accommodate system wide operations in a computationally efficient manner.

This document first provides an overview of modeling frameworks, including the purposes for using a framework, basic framework modeling process and system architecture, and defines key roles for members of a modeling team. Subsequently, requirements for a WTMP framework are reviewed, followed by a discussion of framework selection criteria, which were developed based on WTMP framework requirements. Frameworks selected for consideration are then introduced and evaluated based on the established selection criteria and recommendations for a WTMP framework. Finally, a description of proposed framework design for the WTMP is provided. River, reservoir, and system model selection is presented in the accompanying *Technical Memorandum 6: Water Temperature Modeling Platform: Model Selection*.

Chapter 2 Modeling Framework Overview

The modeling framework overview section identifies the primary purposes and benefits of a modeling framework, then describes basic modeling processes and modeling framework processes. Next, key roles and skills for members of a modeling team are described with respect to constructing, maintaining, and using a framework. Last, examples of two types of modeling framework architecture are described. This overview provides important information for framework selection because design choices influence selection criteria.

Purposes and Benefits of a Modeling Framework

A modeling framework is a software application that can accommodate the use of multiple models, be used to streamline model application and automate repetitive tasks to make the modeling process more efficient and more robust (David et al. 2010). A modeling framework is not itself a model. Instead, a modeling framework combines and controls models, allowing decision makers to expand the scope, and extend the usefulness, of models for decision support (Buahin and Horsburgh 2018, David et al 2013). At the same time, a framework can make model use easier and enhance model efficiency, consistency, adaptability, and transparency (David et al. 2010). Specifically, a framework can provide the following benefits (Gil et al. 2021, Zeng et al. 2020, Buahin and Horsburgh 2018, Zhang et al. 2017, vanGerven et al. 2015, CEIWR-HEX 2010, Sanders 1999):

- Accommodating/integrating multiple models of varying spatial and temporal resolution, different system representations, and even differing disciplines.
- Facilitating the use of multiple models, individually or in a sequence, including the use of a graphical user interface.
- Reducing input error, and errors in passing data from one model to another.
- Reducing the time it takes to carry out modeling activities by automating repetitive tasks.
- Standardizing data management and reporting.
- Ensuring consistent data sharing among models of different complexity and temporal resolution.
- Managing updates and addition of new features, including scalability.
- Reducing the requirement for training on model-specific details for file editing and information flow by automating repetitive modeling tasks and providing an environment for automated linking of discrete models.
- Leveraging available technology to take advantage of new models, computational efficiency, and other advances.

Basic Modeling Processes

Modeling frameworks are able to provide the aforementioned benefits because nearly all models follow common patterns of organization and application. Typical computational model configuration, input, and output are illustrated in Figure 2-1. Model configuration information includes data, model control inputs, and calibration parameters that are created with the initial implementation of a generalized numerical model for application to a specific physical system. Model input, including boundary conditions and operational data, must be prepared for each simulation (i.e., a model run). Simulation-specific model output is created as model computation is carried out.

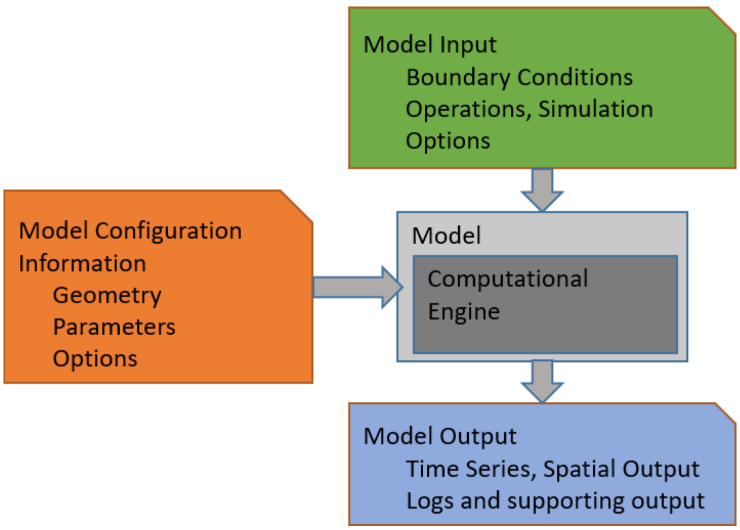

Figure 2-1. Typical computational model configuration, input, and output.

Important in the framework and model development and application process is an understanding of the necessary technical capabilities of modeling team members and their roles in support of modeling efforts. Developing and configuring models, and implementing those models for production use within frameworks, requires a variety of skill sets and roles that involves a team of people, although any individual may fill more than one role. Descriptions of these modeling team roles are listed below.

- **Model Developer:** Develops and maintains model logic flow and specifications for models, including input and output, computational engine, and other associated activities.
- **Expert Modeler:** Configures and calibrates each model for a particular application.
- **Power User:** Configures automated processing for pre- and post- processing, designs reports, manages model linkages.
- **Model Operator:** Carries out modeling studies.

- **IT Support:** Manages the IT infrastructure to facilitate team modeling and provide connectivity to web data sources.

Specific modeling activities and how they might be distributed across team members with varying levels of experience, skills, and responsibilities are listed in Table 2-1.

Model operators are responsible for ensuring that the input data are correct and are formatted for model input (pre-processing), for validating that model calculations were successful, and for creating the desired analysis products from model output (post-processing). Periodically, expert modelers may be asked to update configuration data and/or model calibration parameters.

Table 2-1. Modeling framework activities and associated team roles ("X" indicates the activity is associated with a team role, "NA" indicates a blank cell).

Activity	Model Developer	Expert Modeler	Power User	Model Operator	IT Support
Update, modify, change the computational engine source code	X	NA	NA	NA	NA
Change the input/output data content or formats utilized by the computational engine	X	NA	NA	NA	NA
Initial configuration/calibration of a model, updates to configuration/calibration	X	X	NA	NA	NA
Set up connections to input data sources and linkages between models	NA	X	X	NA	NA
Set up scripts for pre-processing, calculation, post-processing	NA	X	X	NA	NA
Design output reports	NA	X	X	NA	NA
Create new model runs, carry out simulations	NA	X	X	X	NA
Prepare reports and data products from model runs	NA	X	X	X	NA
Network file system management and connectivity to web data sources	NA	NA	NA	NA	X

Modeling Framework Process

The foundation of the modeling framework process is formalizing the flow of information into and out of a set of numerical models to allow automation of production modeling tasks. This process includes:

- Management of model configuration information,
- Collection and formatting of boundary conditions and time dependent controls,
- Passing of information between models (model coupling or linking), and

- Results processing.

The general concept of a modeling framework supporting multiple models is shown in Figure 2-2, including data sources that provide input to a model framework, and a model framework that includes automated pre-processing, model linking, and automated post-processing that provides output to modeling products. The model framework must provide for definition and selection of a consistent set of model configuration files and linking strategies to define the basis for a simulation. Automated pre-preprocessing is used to gather time dependent information from one or more data sources and then format that data to meet the input requirements of the specific numerical models used in the simulation. Once results are computed, automated post-processing can be used to read data from model output files in their native formats and to create standard reports or other data products for distribution. A modeling framework leverages automation to preform sets of simulations such as sensitivity analysis, ensemble simulation, or Monte Carlo analysis (Clark et al. 2015, vanGerven et al. 2015).

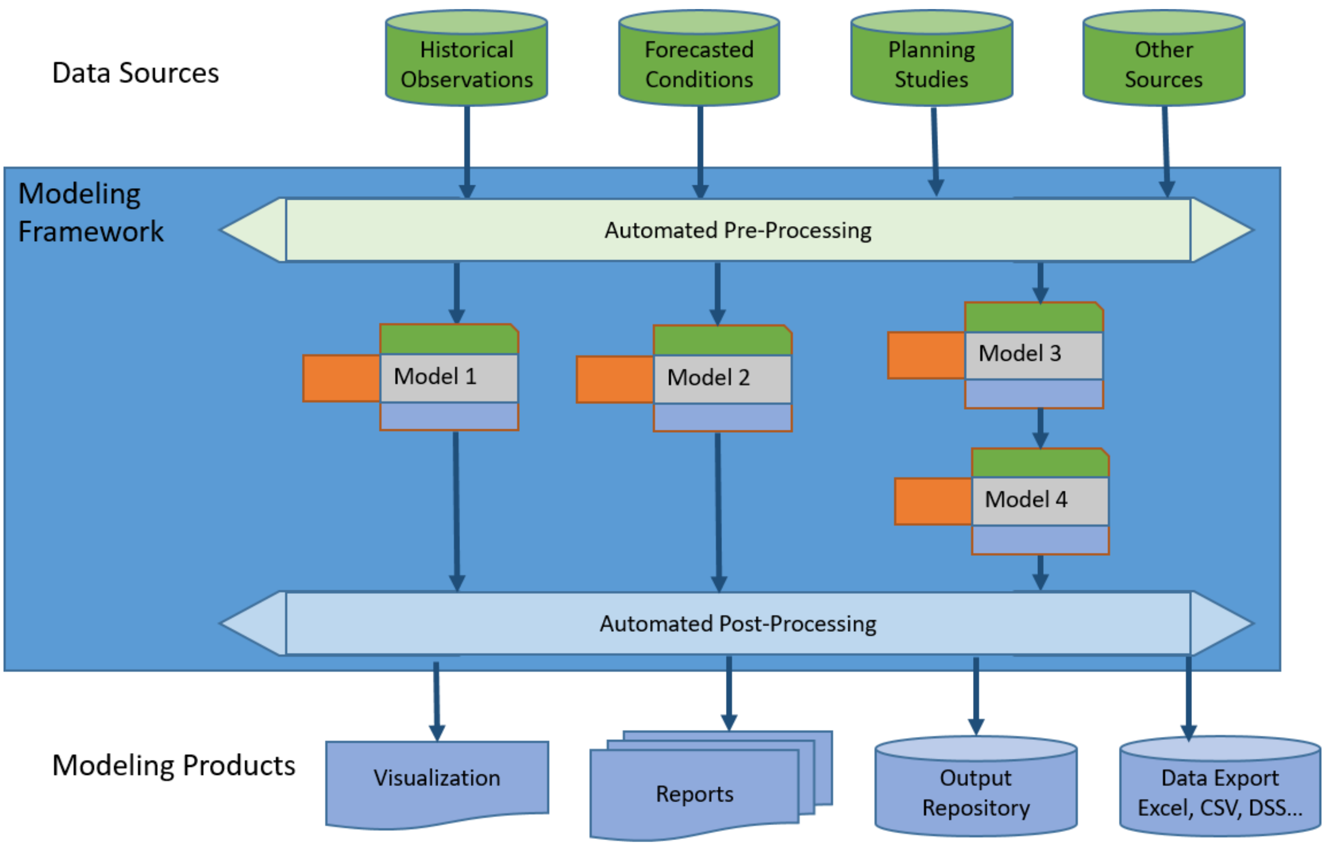

Figure 2-2. Generalized modeling framework supporting multiple models. Data sources provide input to a modeling framework that includes automated pre-processing, linked models, and automated post-processing that provides output to modeling products.

Software System Architecture

Software system architecture refers to the fundamental components of a software system and the relationships and information flow between components (see for example Bass, et al. (2003)). In the context of developing a numerical modeling framework, the system architecture should consider selection of the framework itself, where the framework software will reside, information flow into and out of the framework, identification of supporting systems such as a Data Management system. Design choices will be driven by overall modeling objectives and how the framework and models would be used within an organization, as well as the required modeling skill level of modeling team members. One of the key architectural design decisions will be related to the whether the system will be focused on distributed modeling workstations or a more corporate, server or cloud-based approach.

Within an organization, model use strategies can range from a single modeler using a desktop computer or workstation to a full enterprise approach. In a full enterprise approach, multiple users interact through a web interface to access data and modeling services distributed across multiple, possibly cloud-based, servers. The approach to model use preferred by Reclamation will be established through ongoing discussion. Two example model use strategies are presented in the following sections. The example model use strategies demonstrate how an understanding of use strategy can guide the development of the system architecture.

Workstation-Based Architecture Supporting a Modeling Team Within a Single Organization

For a modeling team within a single organization, modeling software is installed on each team member's workstation, shared data can be downloaded from a commonly accessible server, and selected output can be posted back to the server after model simulations are completed (Figure 2-3). Although primary model use would be within a local area network, use from remote workstations is possible if the server is made to be internet accessible. Key elements of this workstation-based use strategy include:

- Modeling software package and base data sets are downloaded from file server and installed locally on workstations.
- Boundary conditions and other model information are collected and managed in a common database on the file server.
- Model runs and post-processing are carried out on local workstations.
- Selected model configurations and results sets are uploaded to the server to share with the team (with appropriate safeguards in place to prevent overwriting of data).

This use strategy is similar in concept to modeling by a single user. Speed of computation will depend on the specifications of the individual workstations. Providing a common file server facilitates collaborative modeling across the team. Although this approach requires some information technology (IT) support to manage the file sharing infrastructure, the level of IT support is much less than in the enterprise modeling architecture discussed below.

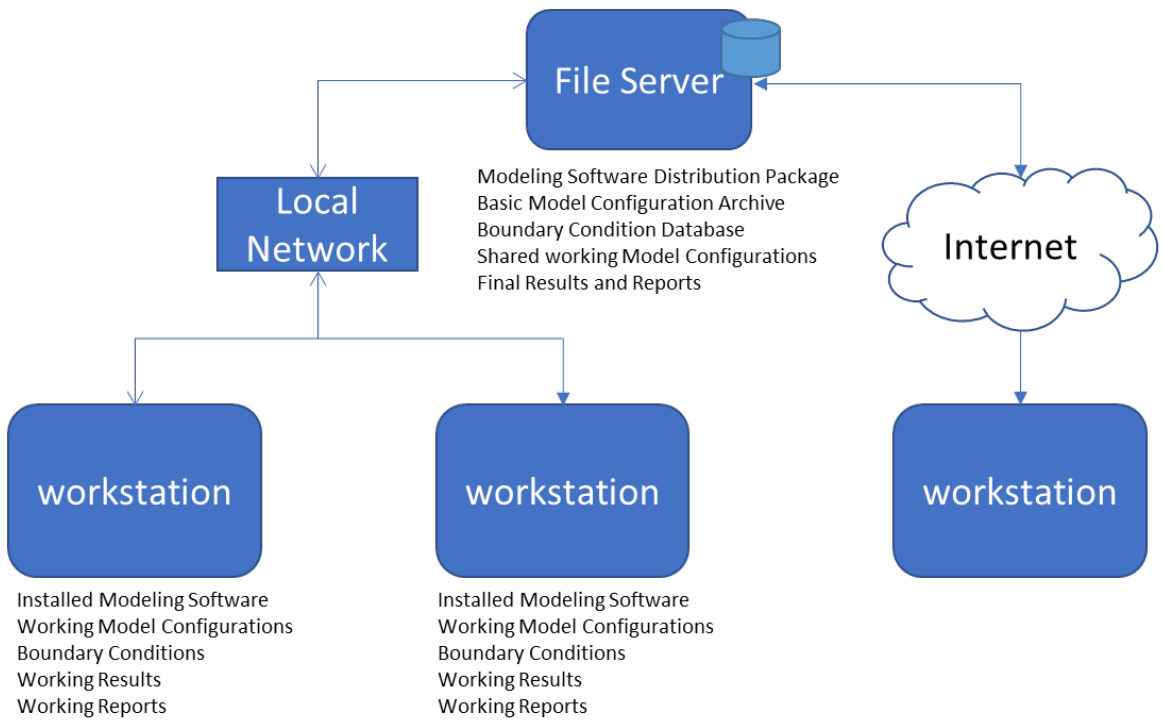

Figure 2-3. Workstation-based modeling architecture.

Enterprise Modeling Architecture

In the enterprise modeling architecture, data management and computation are performed on enterprise servers or in a cloud computing environment (Figure 2-4). Users interact with the modeling system through a web-enabled application interface; no modeling software is installed on a user's workstation. Key elements of this configuration include:

- Modeling software, boundary conditions, and model configurations are all managed on an enterprise server system.
- Users access the system through a web application – a simple client-side software application or possibly a standard web browser.
- Computation could be carried out on hardware managed by the enterprise, or through commercial cloud services.

The advantage to the enterprise modeling use strategy is that model users are not directly involved with details of data management, and computation can be spread across multiple servers (virtual machines) in a high-performance computing environment (e.g., use of "supercomputers"). This approach requires more investment in the modeling IT infrastructure and in on-going IT support.

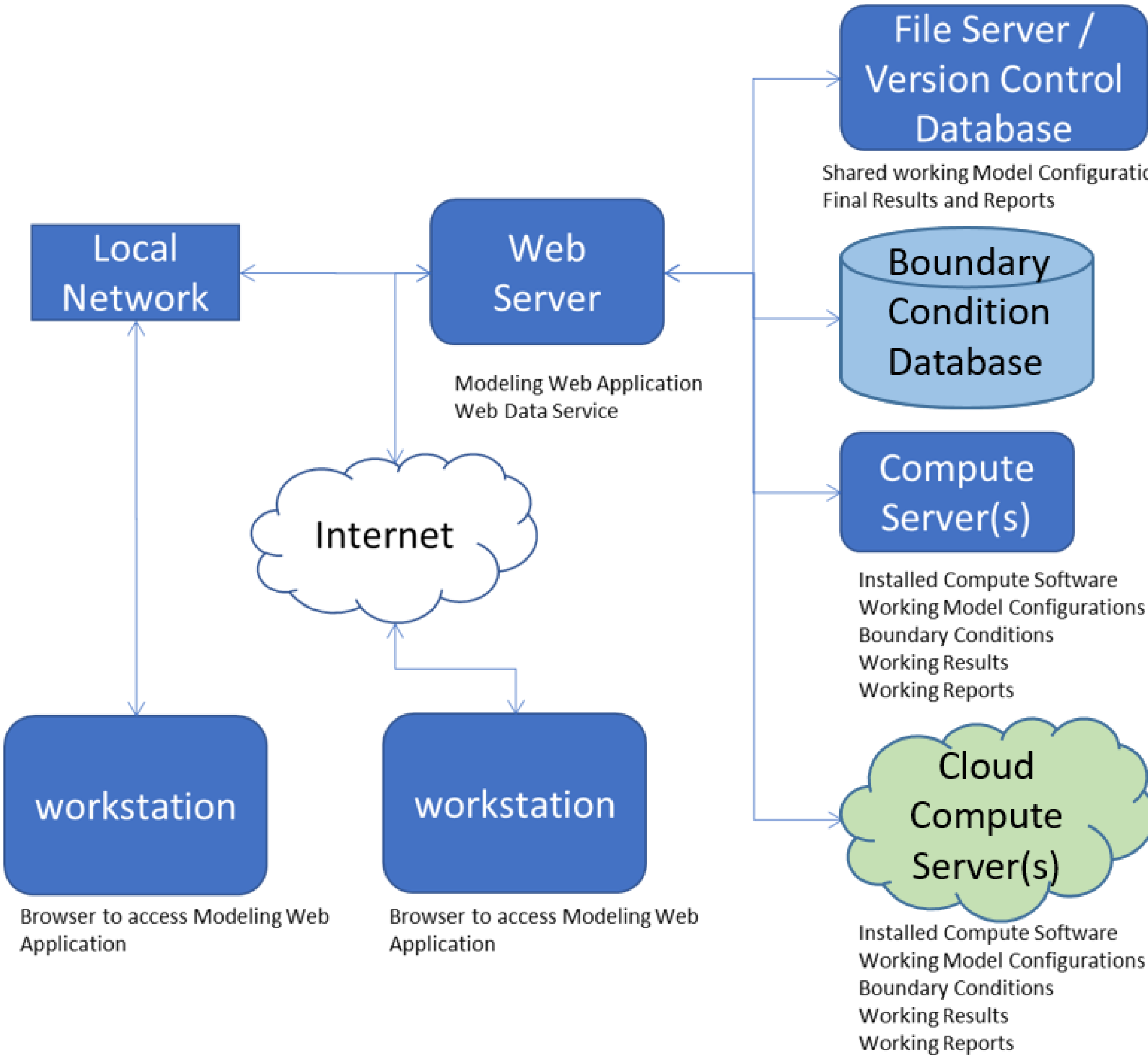

Figure 2-4. Enterprise modeling architecture.

This page intentionally left blank.

Chapter 3 Framework Implementation Requirements for the WTMP

Chapter 2 presents concepts and information that could apply to develop any modeling framework. Presented herein are implementation requirements for a modeling framework specific to the WTMP, based on project goals and objectives, feedback from Reclamation, and experience from the technical team. Consideration of these specific needs guides the development of evaluation criteria for selecting a modeling framework software for application (Chapter 4).

A set of evaluation criteria was developed based on Reclamation's objectives for development of the WTMP and required modeling capabilities that include (but are not limited to):

- Network representing physical linkages and physical characteristics (selective withdrawal leakage, thermal curtain behavior, submerged structures, etc.) between the facilities,
- Inflow volume, timing, and temperature forecasting/modeling,
- Temperature and stratification forecasting/modeling based on forecasted inflow, forecasted meteorological conditions, and a forecasted release schedule, and
- Simulation of selective withdrawal operation for automation provided downstream temperature target and specified forecasted release schedule.

Implementation requirements listed below consider the required framework elements and capabilities as well as how models are used, what data is required, information flow, input and output reporting, management of model versions, how new models are introduced to the framework, and other factors. To support Reclamation's WTMP, a framework should be able to:

- Efficiently use several models, individually or in a sequence, or use in concert with a system model;
- Support workflows for several typical modeling activities;
- Utilize common boundary conditions and operational controls across models;
- Manage modeling scenarios for record keeping and reference/reuse;
- Create reports using common report formats across models;
- Provide version control of model executable programs and configuration data sets;
- Allow for incorporation of new modeling tools; and
- Focus on the efficiency of production modeling activities.

Each of the framework implementation requirements for the WTMP are defined in the following sections.

Efficiently Use Several Models, Individually or in a Sequence

Reclamation has historically used a variety of temperature models for specific analyses including discrete component models, system models, statistical models, and others. This trend will likely continue, especially given the requirements for seasonal forecasting, long-term planning analysis, and real-time applications. The modeling framework should support the models Reclamation has used historically, allow for models to be updated, and permit other models to be added over time as Reclamation's modeling needs evolve. Further, the framework must support information transfers between models so that output from one model becomes input to another model for system simulation using component river or reservoir models.

Support Workflows for Several Typical Modeling Activities

A primary use of the WTMP will be to support the development of an annual Temperature Management Plan (TMP) on the Sacramento River. Producing the TMP involves making seasonal forecasts multiple times from early spring through summer of each year. Establishing common tasks, such as model testing (validation) and hindcasting activities offers users efficient ways to compare model prediction to observations and provides an additional measure of model performance and confidence. Seasonal and long-term forecasting, which utilize different sources for boundary conditions and have different reporting requirements, can each be supported by a framework.

Utilize Common Boundary Conditions and Operational Controls Across Models

Most of the input data required for water temperature simulation is similar from one model to another. Managing boundary condition data in a consistent manner will allow for the efficient use of multiple models (or model versions) for specific purposes (e.g., high resolution models for detailed analysis and lower resolution models for rapid reconnaissance analysis).

Manage Modeling Scenarios for Record Keeping and Reference/ Reuse

Using models in combination – with each model having parameter sets that may be customized for particular purposes – often leads to the development of a large number of different multi-model configurations. A model framework can provide a system for saving and re-using combinations of models and parameter sets, extending the configuration-saving features of the individual model programs. In the HEC frameworks (HEC-WAT (HEC 2017) and HEC-RTS (HEC 2021a)) these combinations are referred to as "alternatives." Alternatives (scenarios) can be used to form groups of models for a particular task, such as producing a seasonal report, comparing several alternate operation strategies, or for preserving sets of parameters as part of a calibration exercise.

Create Reports Using Common Report Formats Across Models

Most temperature models produce similar outputs (e.g., daily flow, hourly water temperature) at key locations in the model domains, allowing standardization of reports (calibration/validation, seasonal, and long-term planning reports) to use output from any temperature model. For example, a high-resolution reservoir model will share some common outputs with a simplified reservoir model, such as reservoir release flow and temperature or cold water pool volume. For common outputs, there can be common reporting of results and comparison to historical data. The reporting mechanism should also support analysis of detailed results that may only be produced by high-resolution models (e.g., temperature profiles at multiple locations in a reservoir).

Compliance with the accessibility requirements of Section 508 of the Rehabilitation Act for public-facing outputs or reports is important. Standardization of data types and formats should be used to simplify compliance.

Provide Version Control of Model Executable Programs and Configuration Data Sets

Periodically, new versions of model executable programs may become available, and model configuration data or calibrated parameters may need to be updated due to collection of new physical data, new operational approaches, or calibration to recent historical data. The modeling framework must allow for management of model updates, as well as updates of configuration data files, preferably in coordination with a version control repository or artifact management system to manage changes over time.

Allow for Introduction of New Modeling Tools

The modeling framework must allow for the introduction of new modeling tools that were not included during the original framework configuration. The framework must be flexible enough to accommodate introduction of new computational models and pre- and post-processing tools without significant effort.

Focus on the Efficiency of Production Modeling Activities

Although the modeling framework may support a variety of uses, the focus must be on the production use of temperature models to support Reclamation's temperature management responsibilities. Having a streamlined and easy-to-learn user interface for model operators is a high priority and can serve to efficiently introduce new operators to temperature modeling in an accessible manner.

Chapter 4 Framework Evaluation Criteria

This section presents a set of framework evaluation criteria that were developed in cooperation with Reclamation and the technical team, based on the framework implementation goals and objectives presented in the previous chapter. Model framework evaluation criteria pertain to model support, data management, user interface, and software installation and configuration. Eight sub-categories of criteria summarized in tables below are:

- Criteria based on model types that may be utilized in the WTMP modeling framework (Table 4-1);
- Criteria based on type of model coupling utilized by a model framework (Table 4-2);
- Criteria based on forms of workflow control utilized when running a sequence of models in model framework (Table 4-3);
- Criteria based on model configuration and time series data management application that may be utilized in a framework (Table 4-4);
- Criteria based on user interface capabilities the framework can provide to improve the useability, efficiency, and transparency of modeling activities (Table 4-5).
- Criteria based on locations of model and framework configuration and time series data storage (Table 4-6);
- Criteria based on where computations are performed (Table 4-7); and
- Criteria based on type of software application model with which model operators will interact (Table 4-8).

Table 4-1. Model framework evaluation criteria based on model types that may be utilized in the WTMP modeling framework.

Criterion	Notes	Importance
CEQUAL-W2	Direct support for specific models expected to be used by Reclamation may significantly reduce the effort implementing the modeling framework.	Must
HEC-5Q	Direct support for specific models expected to be used by Reclamation may significantly reduce the effort implementing the modeling framework.	Prefer
HEC-ResSim	Direct support for specific models expected to be used by Reclamation may significantly reduce the effort implementing the modeling framework.	Prefer

Criterion	Notes	Importance
HEC-RAS	Direct support for specific models expected to be used by Reclamation may significantly reduce the effort implementing the modeling framework.	Prefer
General command line models	Models typically run from a command line can generally be run in an automated mode on a server.	Must
General GUI based models	Models that are embedded within a user interface may be more difficult to automate from a modeling framework.	Prefer
Scripted processes	Many pre- and post-processing activities can be accomplished with scripts.	Must
Excel worksheets	Leveraging Excel-based tools such as the Jacobs WQ visualization worksheet may be advantages.	Prefer

Table 4-2. Model framework evaluation criteria based on types of model coupling utilized in modeling framework.

Criterion	Notes	Importance
Loose coupling	Loose coupling allows one model output to pass as input to another model after completing simulation over a prescribed time window.	Must
Tight coupling	Tight coupling allows exchanges of data within model computation loops, typically relying on inter-process communication based on a common API such as Open Modeling Interface (OpenMI).	Unnecessary

Table 4-3. Model framework evaluation criteria based on forms of workflow control utilized when running a sequence of models in a modeling framework.

Criterion	Notes	Importance
Linear sequence	Simple sequence where output from one model becomes input to the next model in the sequence.	Must
IF-THEN-ELSE conditionals	Branching structure based on conditional evaluation, useful to reduce run-time when some models do not need to be computed under all conditions.	Prefer
Loops	Iteration loops are the foundation for ensemble, Monte Carlo, and sensitivity simulations.	Prefer
Ensemble Sets	Running a set of simulations utilizing an ensemble of boundary conditions.	Prefer
Monte Carlo Iteration	Running a large set of simulations where boundary conditions are utilized.	Prefer
Sensitivity Analysis	Making a series of runs varying one or more model parameters over a given range.	Prefer
Uncertainty Analysis	Including computations to place confidence limits on model results.	Prefer

Table 4-4. Model framework evaluation criteria based on model configuration and time series data management application that may be utilized in a framework.

Criterion	Notes	Importance
Data Acquisition	Collection, validation, and correction of new field data.	Prefer
Boundary Condition Management	Linking historical or planning time series data to models.	Must
Alternative Configurations	Managing model configuration files for existing or proposed conditions.	Must
Analysis Period Specifications	Managing model simulation periods.	Must
Simulation (run) Management	Managing and executing sets of model runs.	Must
Forecasting Support	Managing model configuration and output for regular forecasting.	Prefer
Planning Support	Organizing sets of input and output for a planning workflow.	Must
Configuration Version Control	Managing changes in model configuration data through version control.	Prefer
Result Posting and Archiving	Result posting and archiving.	Prefer
Distributable data and model versions	Allow selected data and models to be publicly distributed.	Must

Table 4-5. Model framework evaluation criteria based on user interface capabilities the framework can provide to improve the useability, efficiency, and transparency of modeling activities.

Criterion	Notes	Importance
Configure model linking	Configure, manage, and display boundary condition and model-to-model linking.	Must
Model parameter editing	Editing of at least the primary model data.	Prefer
Run control	Provide user interface (UI) to facilitate run management.	Must
Alternative Management	Provide a UI to manage alternatives.	Must
Plotting Results	Provide a UI to do at least basic visualization of model results.	Must
Reporting	Provide a UI to create at least basic reports.	Must
Workflow Guidance	Provide an interface to facilitate standard workflows.	Prefer

Table 4-6. Model framework evaluation criteria based on locations of model and framework configuration and time series data storage.

Criterion	Notes	Importance
Desktop Workstation	Decision depends on preferred modeling team organization and IT infrastructure, and influences workflow.	Must
Local Server	Decision depends on preferred modeling team organization and IT infrastructure, and influences workflow.	Must
Cloud Server	Decision depends on preferred modeling team organization and IT infrastructure, and influences workflow.	Prefer

Table 4-7. Model framework evaluation criteria based on where computations are performed.

Criterion	Notes	Importance
Desktop Workstation	Decision depends on preferred modeling team organization and IT infrastructure, and influences workflow and computational performance.	Must
Local Server	Decision depends on preferred modeling team organization and IT infrastructure, and influences workflow and computational performance.	Prefer
Cloud Server	Decision depends on preferred modeling team organization and IT infrastructure, and influences workflow and computational performance.	Prefer

Table 4-8. Model framework evaluation criteria based on type of software application model with which model operators will interact.

Criterion	Notes	Importance
Desktop Application	Decision depends on preferred modeling team organization and IT infrastructure, and influences workflow.	Must
Web Application	Decision depends on preferred modeling team organization and IT infrastructure, and influences workflow.	Unnecessary

Chapter 5 Recommendation

Eight framework systems were examined for potential use in the WTMP for the Sacramento/Trinity, American, and Stanislaus River basins. These frameworks were identified through web searches and a very useful review of modeling frameworks by the New Zealand National Institute for Water and Atmosphere (Elliot et al. 2014). Reclamation utilized information developed in this technical memorandum to frame internal discussions on the framework criteria, attributes, and importance to the WTMP objectives. This information was reviewed with the technical team to assess tradeoffs and balance the various attributes of identified frameworks to arrive at a final recommendation.

The frameworks that were considered are:

- Object Modeling System/Cloud Services Integration Platorm (OMS3/CSIP) (David, et al. 2010; David et al. 2014)
- Earth System Modeling Framework (ESMF) (Collins, et al. 2005)
- HydroCouple (Buahin & Horsburgh 2018)
- Community Surface Dynamics Modeling System (CSDMS) (Peckham et al. 2012; Hutton et. al. 2020)
- Delft FEWS (Werner et al. 2004; Werner et al. 2012)
- Delta Shell (Donchyts & Jagers 2010)
- HEC-WAT (HEC 2017; Dunn & Baker 2010; DeGeorge et al. 2014)
- HEC-RTS (HEC 2021a)

Each framework was evaluated using the criteria described in Chapter 4, and the results are presented in tables below. In the tables, “Y” indicates the framework provides good suitability to the needs of the WTMP (based on review of specific frameworks and experience from the technical team familiar with these key software and hardware attributes), without requiring customization or additional programming. A “Y*” indicates the framework can meet the project’s need for this capability, but some adjustment or creative application may be required. A “C” indicates a framework can provide a capability, but effort is required for customization or coding. Specifically, a “C” indicates more than simply configuring the framework to match a particular region; it indicates that a substantial effort by a knowledgeable engineer or programmer is required to deliver the desired capability. Each “C” in the table is assigned “1”, “2”, or “3” to indicate a relative level of effort, from low to high, of customization or coding needed for the framework to deliver the desired capability.

Model frameworks were evaluated using criteria that pertain to model support, data management, user interface, and software installation and configuration. Comparison of framework capabilities are summarized below, using eight sub-categories of evaluation criteria:

- Criteria based on model types that may be utilized in the WTMP modeling framework (Table 5-1);
- Criteria based on type of model coupling utilized by a model framework (Table 5-2);
- Criteria based on forms of workflow control utilized when running a sequence of models in model framework (Table 5-3);
- Criteria based on model configuration and time series data management application that may be utilized in a framework (Table 5-4);
- Criteria based on user interface capabilities the framework can provide to improve the useability, efficiency, and transparency of modeling activities (Table 5-5).
- Criteria based on locations of model and framework configuration and time series data storage (Table 5-6);
- Criteria based on where computations are performed (Table 5-7); and
- Criteria based on type of software application model with which model operators will interact (Table 5-8).

Development of the WTMP will require programming of custom features and extensions to the framework. Four development languages are utilized by the eight candidate frameworks. Java and Python are preferred because WTMP will incorporate existing models and data systems that already rely on those languages. The primary development language utilized by each framework is indicated by an "X" in Table 5-9.

Appendix A includes additional details for candidate frameworks considered for WTMP application.

Table 5-1. Model framework comparison per criteria based on model types that may be utilized in the WTMP modeling framework. Y indicates the framework provides good suitability for WTMP; Y* indicates the framework meets WTMP needs with some adjustment; C1, C2, and C3 indicate the framework can be customized to meet WTMP needs with relatively low, moderate, and high level, respectively, of effort of customization and coding.

Criterion	Importance	OMS3/CSIP	ESMF	HydroCouple	CSDMS	Delft-FEWS	Delta Shell	HEC-WAT	HEC-RTS
CEQUAL-W2	Must	C3	C3	Y*	C3	C3	C2	Y	C2
HEC-5Q	Prefer	C3	C3	C3	C3	C3	C2	C2	C2
HEC-ResSim	Prefer	C3	C3	C3	C3	Y	C2	Y	Y
HEC-RAS	Prefer	C3	C3	C3	C3	Y	C2	Y	Y
General command line models	Must	C3	C3	C3	C3	C2	C2	C2	C2
General GUI based models	Prefer	C3	C3	C3	C3	C3	C3	C3	C3
Scripted processes	Must	C3	C3	C3	C3	C2	C2	C1	C1
Excel worksheets	Prefer	C3	C3	C3	C3	C3	C3	C3	C3

Table 5-2. Model framework comparison per criteria based on type of model coupling utilized by a model framework. Y indicates the framework provides good suitability for WTMP; Y* indicates the framework meets WTMP needs with some adjustment; C1, C2, and C3 indicate the framework can be customized to meet WTMP needs with relatively low, moderate, and high level, respectively, of effort of customization and coding.

Criterion	Importance	OMS3/CSIP	ESMF	HydroCouple	CSDMS	Delft-FEWS	Delta Shell	HEC-WAT	HEC-RTS
Loose coupling	Must	Y	Y	Y	Y	Y	Y	Y	Y
Tight coupling	Unnecessary	Y	Y	Y	Y	C3	Y	C3	C3

Table 5-3. Model framework comparison per criteria based on forms of workflow control utilized when running a sequence of models in model framework. Y indicates the framework provides good suitability for WTMP; Y* indicates the framework meets WTMP needs with some adjustment; C1, C2, and C3 indicate the framework can be customized to meet WTMP needs with relatively low, moderate, and high level, respectively, of effort of customization and coding.

Criterion	Importance	OMS3/CSIP	ESMF	HydroCouple	CSDMS	Delft-FEWS	Delta Shell	HEC-WAT	HEC-RTS
Linear sequence	Must	Y	Y	Y	Y	Y	Y	Y	Y
IF-THEN-ELSE conditionals	Prefer	C3	C3	C1	C1	C3	C1	C2	C3
Loops	Prefer	C3	C3	C2	C2	Y*	C1	C2	C2
Ensemble Sets	Prefer	C3	C3	C3	C2	Y*	C3	C2	Y
Monte Carlo Iteration	Prefer	C3	C3	C3	C2	C3	C3	Y	C3
Sensitivity Analysis	Prefer	C3	C3	C3	C2	C3	C2	C2	C3
Uncertainty analysis	Prefer	C3	C3	C3	C2	C3	C2	C2	C3

Table 5-4. Model framework comparison per criteria based on model configuration and time series data management application that may be utilized in a framework. Y indicates the framework provides good suitability for WTMP; Y* indicates the framework meets WTMP needs with some adjustment; C1, C2, and C3 indicate the framework can be customized to meet WTMP needs with relatively low, moderate, and high level, respectively, of effort of customization and coding.

Criterion	Importance	OMS3/CSIP	ESMF	HydroCouple	CSDMS	Delft-FEWS	Delta Shell	HEC-WAT	HEC-RTS
Data Acquisition	Prefer	C3	C3	C3	C3	Y	C3	C2	Y
Boundary Condition Management	Must	C2	C2	C2	Y*	Y	Y	Y	Y
Alternative Configurations	Must	C2	C2	C2	C2	C2	Y	Y	Y
Analysis Period Specifications	Must	C2	C2	C2	C2	Y	C3	Y	Y
Simulation (run) Management	Must	C2	C2	C2	C2	Y	Y	Y	Y
Forecasting Support	Prefer	C3	C3	C3	C3	Y	C3	C2	Y
Planning Support	Must	C3	C3	C3	C3	C3	C3	Y	C2
Configuration Version Control	Prefer	C3	C3	C3	C3	C3	C3	C2	Y*
Result Posting and Archiving	Prefer	C3	C3	C3	C3	C3	C3	C2	Y
Distributable data and model versions	Must	Y	Y	Y	Y	Y	Y	Y	Y

Table 5-5. Model framework comparison per criteria based on user interface capabilities the framework can provide to improve the useability, efficiency, and transparency of modeling activities. Y indicates the framework provides good suitability for WTMP; Y* indicates the framework meets WTMP needs with some adjustment; C1, C2, and C3 indicate the framework can be customized to meet WTMP needs with relatively low, moderate, and high level, respectively, of effort of customization and coding.

Criterion	Importance	OMS3/CSIP	ESMF	HydroCouple	CSDMS	Delft-FEWS	Delta Shell	HEC-WAT	HEC-RTS
Configure model linking	Must	Y	Y*	Y	Y*	Y*	Y	Y	Y
Model parameter editing	Prefer	C3	C3	C3	C3	C2	C2	Y	Y
Run control	Must	C2	C2	C2	C2	Y	Y	Y	Y
Alternative Management	Must	C3	C3	C3	C3	C3	C3	Y	Y
Plotting Results	Must	C3	C3	C3	C3	Y	Y	Y	Y
Reporting	Must	C3	C3	C3	C3	C3	C2	C2	C2
Workflow Guidance	Prefer	C3	C3	C3	C3	C3	C3	C2	Y

Table 5-6. Model framework comparison per criteria based on locations of model and framework configuration and time series data storage. Y indicates the framework provides good suitability for WTMP; Y* indicates the framework meets WTMP needs with some adjustment; C1, C2, and C3 indicate the framework can be customized to meet WTMP needs with relatively low, moderate, and high level, respectively, of effort of customization and coding.

Criterion	Importance	OMS3/CSIP	ESMF	HydroCouple	CSDMS	Delft-FEWS	Delta Shell	HEC-WAT	HEC-RTS
Desktop Workstation	Must	Y	Y*	Y	C3	Y	Y	Y	Y
Local Server	Must	C2	Y*	C2	Y*	Y*	C2	C2	Y
Cloud Server	Prefer	C2	Y*	C2	Y*	Y*	C2	C2	C2

Table 5-7. Model framework comparison per criteria based on where computations are performed. Y indicates the framework provides good suitability for WTMP; Y* indicates the framework meets WTMP needs with some adjustment; C1, C2, and C3 indicate the framework can be customized to meet WTMP needs with relatively low, moderate, and high level, respectively, of effort of customization and coding.

Criterion	Importance	OMS3/CSIP	ESMF	HydroCouple	CSDMS	Delft-FEWS	Delta Shell	HEC-WAT	HEC-RTS
Desktop Workstation	Must	Y	Y*	Y	C3	Y	Y	Y	Y
Local Server	Prefer	Y	Y*	C2	Y	C2	C2	Y*	C2
Cloud Server	Prefer	Y*	Y*	C2	Y	C2	C2	Y*	C2

Table 5-8. Model framework comparison per criteria based on type of software application model with which model operators will interact. Y indicates the framework provides good suitability for WTMP; Y* indicates the framework meets WTMP needs with some adjustment; C1, C2, and C3 indicate the framework can be customized to meet WTMP needs with relatively low, moderate, and high level, respectively, of effort of customization and coding.

Criterion	Importance	OMS3/CSIP	ESMF	HydroCouple	CSDMS	Delft-FEWS	Delta Shell	HEC-WAT	HEC-RTS
Desktop Application	Must	Y*	Y*	Y*	C3	Y	Y	Y	Y
Web Application	Prefer	C3	C3	C3	Y*	C3	C3	C3	C3

Table 5-9. Primary development language for the framework. X indicates primary development language, NA indicates a blank cell.

Primary Development Language	Preference	OMS3/CSIP	ESMF	HydroCouple	CSDMS	Delft-FEWS	Delta Shell	HEC-WAT	HEC-RTS
Java	Prefer	X	NA	NA	NA	X	NA	X	X
Python	Prefer	NA	NA	X	X	NA	NA	X	X
.NET	NA	NA	NA	NA	NA	NA	X	NA	NA
C/C++/FORTRAN	NA	NA	X	X	X	NA	NA	NA	NA

Upon review of Reclamation's modeling needs and the criteria described above, HEC-WAT is recommended as the model framework platform for the WTMP. Reclamation's application requires a software package that will run on a user's desktop and will provide support for a production workflow. Of the frameworks we examined, only the two Deltares frameworks (Delft-FEWS and Delta Shell) and the two HEC frameworks (HEC-WAT and HEC-RTS) provided that support. The other frameworks under consideration provide a variety of tools for combining and running models but would require extensive new user interface development for this application.

The Deltares FEWS and HEC-RTS applications have well developed production user interfaces but are better suited to short-term forecasting and decision support. Deltares Delta Shell and HEC-WAT can both support modeling workflows that are appropriate to Reclamation's needs. However, HEC-WAT has the advantage of already having well-developed systems for managing and comparing alternative operating schemes, while Delta Shell would require new code to support alternative management and comparison. In addition, all of the target modelling applications -- CEQUAL-W2, HEC-ResSim, HEC-RAS, and HEC-5Q – can readily be run in HEC-WAT. Although this capability does not guarantee that the specific requirements for those models in this project will be supported "out-of-the-box," the risk of failure or need for extensive code development due to incompatibility between the framework and those models is dramatically reduced with HEC-WAT.

This page intentionally left blank.

Chapter 6 Proposed Framework Design

A brief overview of the proposed design for the modeling framework based on HEC-WAT is presented in this section. This design overview is not intended as a detailed technical specification, but rather a high-level introduction to the implementation approach. Design elements discussed here include support of computational models, the spatial and temporal domain, system architecture, and organization of data.

Computational Models

The WTMP will utilize both river-reservoir system models and discrete component models as well as linked models where output from one model becomes input to another model (discussed in *Technical Memorandum 6: Water Temperature Modeling Platform: Model Selection*). HEC-WAT interacts with numerical models through a "plug-in" Application Programming Interface (Plug-in API). HEC-WAT currently supports HEC-ResSim and CE-QUAL-W2 through existing plug-ins. The legacy HEC5Q model can be run through a generalized "Scripted" plug-in. The USACE Hydrologic Engineering Center (HEC) intends to support HEC-RAS water quality simulation within HEC-WAT soon (Mark Jensen, personal communication). If Reclamation requires new models to be included in the WTMP in the future, they can be supported through the "Scripted" plug-in or through custom plug-in development.

Spatial and Temporal Domain

The spatial domain for each of the three river-reservoir systems represented by the modeling framework will be based on the largest domain used by the numerical models included in the framework.

Discrete numerical models can be applied in various combinations within the framework. These various discrete models are used to represent a number of reservoir and river system components and key physical features that impact temperature simulation, including:

- Sacramento/Trinity River System
    - Shasta Lake and Shasta Dam temperature control device
    - Keswick Reservoir
    - Sacramento River from Keswick Dam to Bend Bridge
    - Trinity Lake

    - Lewiston Lake (including temperature control curtain and diversion to Whiskeytown Lake)
    - Trinity River from Lewiston Dam to North Fork Trinity River
    - Whiskeytown Lake (including temperature control curtains and diversions to Keswick Reservoir)
    - Clear Creek from Whiskeytown Dam to the Sacramento River
- American River System
    - Folsom Lake and temperature control device (shutter system)
    - Lake Natoma
    - American River from Nimbus Dam to the Sacramento River
- Stanislaus River System
    - New Melones Lake (including submerged dam)
    - Tulloch Lake
    - Goodwin Diversion Dam
    - Stanislaus River from Goodwin Dam to the San Joaquin River.

The approximate spatial domain for each system is presented in Figure 6-1.

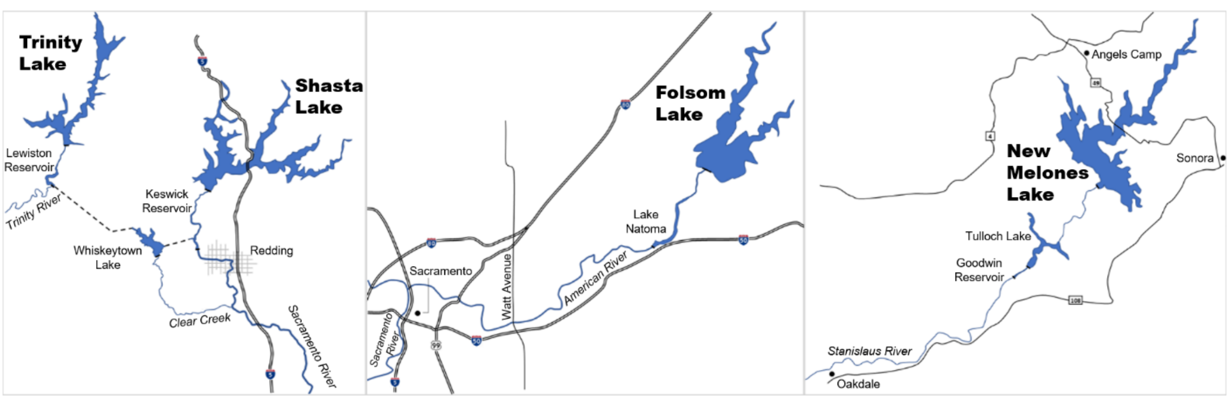

Figure 6-1. Approximate spatial domain for the upper Sacramento River system (left), American River (center), and Stanislaus River (right) study areas.

The temporal domain for modeling activities will be established by the time dependent data stored in the project Data Management System. The modeling framework will be capable of extracting time dependent data from any time window available in the Data Management System, ranging from a few hours to several years. The computational time step for individual models will be established by the configuration of each discrete model and/or system model employed in the framework.

System Architecture

Based on direction from Reclamation for the WTMP, a modeling framework implemented with HEC-WAT will be installed as a desktop application on workstations used by the modeling team. This approach is expected to require less direct IT support than might be required by an enterprise modeling architecture as discussed in section 2 above. However, some IT support may be required during installation of the framework software on users' workstations.

The framework will interact with the project databases via a Data Management Server to extract time dependent data, post key modeling results, and access and update model executable programs and configuration files. Production modeling, result processing, and report generation will be managed by the modeling framework on team workstations. These design concepts are illustrated in Figure 6-2.

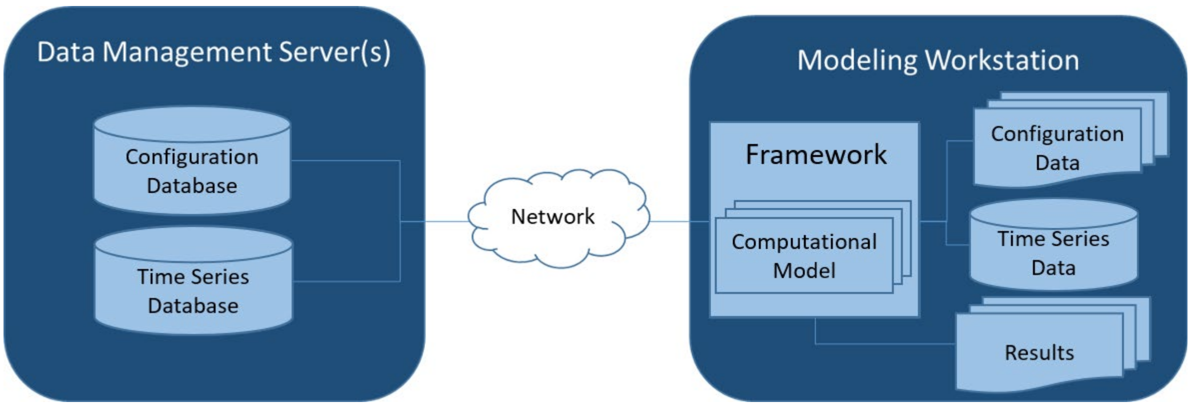

Figure 6-2. Illustration of framework installation on modeling workstation and communication with data management server(s).

Organization of Data

Each numerical model has particular requirements for input and output data files. The HEC-WAT model framework software organizes modeling data in a directory hierarchy on the modeling workstation that accommodates the requirements of each numerical model used in the framework. The following paragraphs describe some of the data management design concepts that are most important to production use of the modeling framework.

Model framework installation files and base model configuration files will be maintained on a version control/artifact management component of the project Data Management System (Reclamation 2021). The version control/artifact management software will allow a modeler to download the latest framework software and configuration files to the modeler's workstation. There may be special permissions required for installation of the framework software on workstation, depending on Reclamation IT protocols.

HEC-WAT results files are organized in a directory hierarchy referred to as a WAT "study." A separate WAT study will be created for each of the river-reservoir systems included in the project – upper Sacramento/Trinity Rivers, American River, and Stanislaus River. The WAT studies will be available for download from the version control/artifact management component of the Database Management System.

In HEC-WAT, a "simulation" represents one or more computational processes in a user-defined sequence and includes the linkage to external boundary conditions as well as the linkage between discrete models within the framework, where applicable. For example, a simulation could incorporate a single system model for the upper Sacramento River system (e.g., Shasta Lake, Keswick Reservoir, and the Sacramento River from Keswick Dam to Red Bluff) or it could include three discrete models separately representing the individual components of Shasta Reservoir, Keswick Reservoir, and the Sacramento River from Keswick Dam to Red Bluff. "Base" simulations represent a primary modeling scenario to which alternative modeling scenarios can be compared.

Creation, execution, and post processing of production model runs will be facilitated through a custom user interface. Further, when production model runs are created through HEC-WAT, copies of the base simulation configuration files will be copied into a new subdirectory hierarchy to facilitate tracking and documentation of the model runs. The modeling framework will include a "post" feature that allows selected model results to be posted to the Database Management System.

Chapter 7 Modeling Framework Selection and Design Summary

Reclamation's objective for the development of the Water Temperature Modeling Platform (WTMP) is the effective and efficient management of CVP resources for downstream regulatory and environmental requirements within the context of an uncertain environment. As stated earlier, a primary development goal of the WTMP is to provide realistic predictions of downstream water temperatures with sufficient confidence to carry out the necessary planning for seasonal, real-time, and long-term study applications while also describing situational risk and uncertainty.

For the CVP, there is a need for both a broader network model that can accommodate the large complex reservoir-river networks for temperature management purposes and a framework that can accommodate the more detailed models that represent specific operation of facilities (including the applicable temperature management infrastructure). The framework construct also offers features for multiple model use, data consistency across models, workflow processing, automation, and scenario management which support project goals.

This document presents Reclamation's requirements for a WTMP framework, framework selection criteria that were developed based on those requirements, and an evaluation of modeling frameworks based on those criteria. HEC-WAT satisfied the required criteria and was selected to be the modeling framework for Reclamation's WTMP.

This page intentionally left blank.

Chapter 8 Citations

Bass, Len, Paul Clements, and Rick Kazman. 2003. Software architecture in practice. Boston: Addison-Wesley.

Buahin, C.A., and J.S. Horsburgh. 2018. Advancing the Open Modeling Interface (OpenMI) for Integrated Water Resources Modeling. Environmental Modelling & Software, vol. 108, pp. 133–153., doi:10.1016/j.envsoft.2018.07.015.

Clark, M. P., B. Nijssen, J.D. Lundquist, D. Kavetski, D.E. Rupp, R.A. Woods, J.E. Freer, E.D. Gutmann, A.W. Wood, L.D. Brekke, J.R. Arnold, D.J. Gochis, and R.M. Rasmussen. 2015. A unified approach for process-based hydrologic modeling: 1. Modeling concept, WaterResour. Res., 51, 2498–2514, doi:10.1002/2015WR017198.

Collins, Nancy & Theurich, Gerhard & Deluca, Cecelia & Suarez, Max & Trayanov, Atanas & Balaji, V. & Li, Peggy & Yang, Weiyu & Hill, Chris & Da Silva, Arlindo. (2005). Design and Implementation of Components in the Earth System Modeling Framework. IJHPCA. 19. 341-350. 10.1177/1094342005056120.

David, O., J. Ascough, W. Lloyd, T. R. Green, K. Rojas, G. Leavesley and L. Ahuja. 2013. A software engineering perspective on environmental modeling framework design: The Object Modeling System. Environ. Model. Softw. 39: 201-213.

David, O., J.C. Ascough II, G.H. Leavesley, L. Ahuja. 2010. Rethinking Modeling Framework Design: Object Modeling System 3.0. International Environmental Modelling and Software Society (iEMSs), 2010 International Congress on Environmental Modelling and Software Modelling for Environment's Sake, Fifth Biennial Meeting, Ottawa, Canada.

DeGeorge, John & Ackerman, Mark & Baker, Penni. (2014). Columbia River Treaty 2014/2024: Integrating Numerical Models and HEC-WAT Plug-In Technology. 1661-1669. 10.1061/9780784413548.164. (Conference: World Environmental and Water Resources Congress 2014).

DeGeorge, J., S. Andrews, T. Steissberg, Z. Zhang. 2018. Development and an Integrated Water Quality Modeling Engine for HEC-RAS, HEC-ResSim, and HEC-HMS. Presentation at the 13th International Conference on Hydroscience & Engineering. Chongqing, China. June 18-22, 2018.

Donchyts, Gennadiy & Jagers, Bert. (2010). DeltaShell-an open modelling environment. Proceedings of 2010 International Congress on Environmental Modelling and Software, Ottawa, Canada.

Dunn, Christopher & Baker, Penni. (2010). A Watershed Modeling Tool, HEC-WAT. 99-99. 10.1061/41148(389)99. (Conference: Proceedings of the 2010 Watershed Management Conference).

Elliott, Alexander & Turek, Gabriella & Rutledge, Daniel & Ritchie, Alistair & Herzig, Alexander & Snow, Val. (2014). Framework for Interoperable Freshwater Models: Testing and

Recommendations. NIWA Client Report HAM2014-013. National Institute for Water and Atmosphere, New Zealand.

Gil, Y., D. Garijo, D. Khider, C.A. Knoblock, V. Ratnakar, Maximiliano Osorio, H. Vargas, Minh Pham, Jay Pujara, B. Shbita, B. Vu, Y-Y Chiang, D. Feldman, Yijun Lin, H. Song, V. Kumar, A. Khandelwal, M. Steinbach, K. Tayal, S. Xu, S. Pierce, L. Pearson, D. Hardesty-Lewis, E. Deelman, R. Mayani, Y. Shi, L. Leonard, K. Cobourn, Z. Zhang, C. Duffy. 2021. Artificial Intelligence for Modeling Complex Systems: Taming the Complexity of Expert Models to Improve Decision Making. Computer Science ACM Trans. Interact. Intell. Syst. 11. 1-49.

Goode, D.J., E.H. Koerkle, S.A. Hoffman, R.S. Regan, L.E. Hay,, and S.L. Markstrom. 2010. Simulation of runoff and reservoir inflow for use in a flood-analysis model for the Delaware River, Pennsylvania, New Jersey, and New York, 2004-2006: U.S. Geological Survey Open-File Report 2010-1014, 68 p.

HEC (U.S. Army Corps of Engineers, Institute for Water Resources – Hydrologic Engineering Center). 1999. Water Quality Modeling of Reservoir System Operations Using HEC-5, TD-24.

HEC (U.S. Army Corps of Engineers, Institute for Water Resources – Hydrologic Engineering Center). 2000. HEC-5, HEC-5, Simulation of Flood Control and Conservation Systems, Appendix on Water Quality Analysis, CPD-5Q.

HEC (U.S. Army Corps of Engineers, Institute for Water Resources – Hydrologic Engineering Center). 2017. HEC-WAT, Water Analysis Tool User's Manual, Version 1.0. CPD-88.

HEC (U.S. Army Corps of Engineers, Institute for Water Resources – Hydrologic Engineering Center). 2021a. HEC-RTS, Real Time Simulation User's Manual, Version 3.2.1 CPD-96.

HEC (U.S. Army Corps of Engineers, Institute for Water Resources – Hydrologic Engineering Center). 2021b. HEC-ResSim, Reservoir System Simulation User's Manual, Version 3.3. CPD-82.

Hutton et al., (2020). The Basic Model Interface 2.0: A standard interface for coupling numerical models in the geosciences. Journal of Open Source Software, 5(51), 2317.

Modini, C. 2010. Using HEC-ResSim for Columbia River Treaty Flood Control. Presentation at 2nd Joint Federal Interagency Conference, Las Vegas, NV, June 27 - July 1, 2010.

Peckham, Scott D., Hutton, Eric W.H., Norris, Boyana (2012). A Component-Based Approach to Integrated Modeling in the Geosciences: The Design of CSDMS. Computers and Geosciences Computers & Geosciences, Volume 53, pp.3-12.

Petty, M.D., J. Kim, S.E. Barbosa, and J, Pyun. 2014. Software Frameworks for Model Composition. Modeling and Simulation Engineering. Vol 2014, 18pp.

Sanders, W.H. 1999. Integrated Frameworks for Multi-Level and Multi-formalism Modeling. Proceedings of the 8th International Workshop on Petri Nets and Performance Models, Zaragoza, Spain, Sept. 9.

U.S. Army Corps of Engineers (USACE). 2016. Missouri River Recovery Management Plan Time Series Data Development for Hydrologic Modeling (Draft). Prepared by the Northwestern

Division Omaha and Kansas City Districts, Hydrologic Engineering Branch, Engineering Division. September.

U.S. Bureau of Reclamation (Reclamation). 2021. Technical Memorandum: Water Temperature Modeling Platform: Data Management Plan (DRAFT). September.

van Gerven, L.P.A., R.J. Brederveld, J.J. de Klein, D.L. DeAngelis, A.S. Downing, M. Faber, D.J. Gerla, J. Hoen, J.H. Janse, A.B.G. Janssen, M. Jeuken,B.W. Kooi, J.J. Kuiper, B. Lischke,S. Liu, T. Petzoldt, S.A. Schep, S. Teurlincx, C. Thiange, D. Trolle, E.H. van Nes, W.M. Mooij. 2015. Advantages of concurrent use of multiple software frameworks in water quality modelling using a database approach. Fundamental and Applied Limnology Volume 186 Nr. 1-2, p. 5 – 20.

Werner, Micha & Schellekens, Jaap & Gijsbers, Peter & Dijk, M. & Akker, O. & Heynert, K.. (2012). The Delft-FEWS flow forecasting system. Environmental Modelling & Software. 40. 10.1016/j.envsoft.2012.07.010.

Werner, M., van Dijk, M., Schellekens, J., 2004. DELFT-FEWS: an open shell flood forecasting system. In: Liong, S., Phoon, K., Babovic, V. (Eds.), Proceedings of the 6th International Conference on Hydroinformatics. World Scientific Publishing Company, Singapore, p. 12051212.

Zeng, S. H. Du, J. Xia. 2020. Development of an interface-oriented add-in modeling framework for integrated water system simulation and its application. Environmental Modelling and Software. 134. December.

Zhang, G., Y. Li, Chong Chen, R. Zhou, Dan Chen and Qingguo Zhou. 2017. A Formal Framework for Integrated Environment Modeling Systems. ISPRS Int. J. Geo Inf. 6: 47.

This page intentionally left blank.

Appendix A: Candidate Frameworks

OMS3/CSIP: Object Modeling System

Developer: Colorado State University
https://alm.engr.colostate.edu/cb/project/oms
Brief Description: Interoperable and lightweight modeling framework for component-based model and simulation development on multiple platforms
Programming Language: Java
Most Recent Updates: 2016
Open Source: Yes
Public API: Yes
Freely Distributable: Yes
Method of integrating models: OMS/CSIP API
Model Coupling: varies
Includes Graphical User Interface (GUI) for Model Operators: No

Observations based on brief initial investigation:

- This system is not a feasible solution to this project modeling need
    - Software difficult to use. New Java classes are required for nearly all applications.
    - Scripting is fragile, can easily fail to work, and is not designed to inform you of causes and troubleshooting approaches.
    - The GUI is only useful for modifying and running scripts, not managing the project or the simulation.
    - Appears that the system hasn't been updated since 2016.
- Cloud Services Innovation Platform (CSIP) appears to have been directed toward greater modularization and flexible UI across platforms.

ESMF: Earth System Modeling Framework

Developer: NASA, NOAA, NCAR, DoD
https://earthsystemmodeling.org/
Brief Description: High-performance, flexible software infrastructure for building and coupling weather, climate, and related Earth science applications
Programming Language: Fortran, C, Python
Most Recent Updates: 2020
Open Source: Yes
Public API: Yes
Freely Distributable: Yes
Method of integrating models: ESMF API
Model Coupling: varies
Includes Graphical User Interface (GUI) for Model Operators: No

Observations based on brief initial investigation:

- Provides extensive control over workflow, but it needs to be coded.
- ESMF is the framework, NUOPC is a Fortran utility package for commonly used utilities and gridded components.
- No GUI for configuring or running models, but NUOPC provides a task-oriented interface to component models.
- Focus on gridded components, using NetCDF standard for some supported data formats – applications are mostly global-scale models of atmosphere and oceans.
    - Underlying technology for the National Weather Model.
- Needs to be compiled before it can be used. No installer available. Requires multiple compilers and libraries to build.
- This seems to be a powerful, but low-level tool that would require experienced developers to implement.

HydroCouple

Developer: University of Utah
http://www.hydrocouple.org/
Brief Description: Cross platform component-based modeling framework for integrated modeling for environmental and earth science applications
Programming Language: C, C++, Fortran, Python
Most Recent Updates: 2020
Open Source: Yes
Public API: Yes
Freely Distributable: Yes
Method of integrating models: OpenMI
Model Coupling: tight
Includes Graphical User Interface (GUI) for Model Operators: No

Observations based on brief initial investigation:

- Example applications at GitHub include water temperature models.
- A graphical configuration editor –HydroCouple Composer – is available.
- A Python API is available.
- Documentation seems to be incomplete.
- Unable to get an example running in the short time allocated to this investigation.
- Significant custom development would be required for this project.
- A CEQUAL-W2 component exists.
    - HydroCouple/CE-QUAL-W2 Component
- Directly runs CE-QUAL-W2 Fortran code instead of running an EXE, not sure how viable this is since some CE-QUAL-W2 models require a specific version of the CE-QUALW2 exe.
    - API and GUI don't seem to be actively updated; last update was 2019, but the Python code was last updated in June 2020.
- Hard to troubleshoot errors, the limited and incomplete documentation does not provide steps for assisting user, and installation guide is limited.
- Framework has some promising attributes, but it was not possible to set up a significant test in the available time.

CSDMS: Community Surface Dynamics Modeling System

Developer: University of Colorado
https://csdms.colorado.edu/
Brief Description: Cyber-infrastructure to promote the quantitative modeling of earth surface processes
Programming Language: Python (others)
Most Recent Updates: 2021
Open Source: Yes
Public API: Yes
Freely Distributable: Yes
Method of integrating models: BMI
Model Coupling: varies
Includes Graphical User Interface (GUI) for Model Operators: No, but supported by Sandia's Dakota project

Observations based on brief initial investigation:

- Basic Model Interface (BMI) is used to link models.
- Provides utilities for creating and modifying BMI models.
    - CSDMS Tools
    - Tools include pymt, a Python package for coupling BMI models.
- Web Modeling Tool (WMT) performs simulations with a linked set of models.
    - Does not support sets of alternatives.
- CSDMS has developed Dakotathon – a Python interface to Dakota, which is a toolkit developed by Sandia Labs for optimization and uncertainty quantification. This is not a GUI for operational use but provides a means to allow any BMI model written in Python to be wrapped into a scripted Dakota experiment.
- CSDMS supports a library of data sets.
- CSDMS Web Modeling Tool (WMT) supports high-performance cluster implementation of models built from CSDMS components.
    - CSDMS Web Modeling Tool
- CSDMS has an active community of researcher contributors, but the web site does not include any examples of production use.

Delft-FEWS

Developer: Deltares (Delft)
https://oss.deltares.nl/web/delft-fews/
Brief Description: Framework for a forecasting system utilizing a variety of models
Programming Language: Java
Most Recent Updates: 2021
Open Source: No
Public API: Yes
Freely Distributable: Yes
Method of integrating models: FEWS Model Plug-in API
Model Coupling: loose
Includes Graphical User Interface (GUI) for Model Operators: No (?)

Observations based on brief initial investigation:

- Deltares is a large, stable organization and FEWS has a substantial user community.
- FEWS has extensive documentation of the framework.
- FEWS targets real-time operational decision support.
- Natively supports time series and paired data.
- Adapters have been created for HEC-RAS , HEC-ResSim, and HMS.
- Can be challenging and time consuming to set up.
- Not recommended for uncertainty/sensitivity analysis.

Delta Shell

Developer: Deltares
https://oss.deltares.nl/web/delft-fews/
Brief Description: Integrated modelling environment which provides a platform that can be used to integrate various models, data and tools
Programming Language: .Net (C#)
Most Recent Updates: 2021
Open Source: Yes
Public API: Yes
Freely Distributable: Yes
Method of integrating models: Delta Shell Model Plug-in API
Model Coupling: loose
Includes Graphical User Interface (GUI) for Model Operators: Yes
Observations based on brief initial investigation:

- Flexible UI but would require custom development to support the models needed for this project.

HEC-WAT: Watershed Analysis Tool

Developer: USACE Hydrologic Engineering Center
https://www.hec.usace.army.mil/software/hec-wat/
Brief Description: model integration tool that allows multi-disciplinary teams to perform water resources studies and risk analysis
Programming Language: Java
Most Recent Updates: 2021
Open Source: No (at this time)
Public API: Yes
Freely Distributable: Yes
Method of integrating models: WAT Model Plug-in API or Jython/Python Scripting
Model Coupling: loose
Includes Graphical User Interface (GUI) for Model Operators: Yes
Observations based on brief initial investigation:

- Project team has experience with this framework.
- HEC-WAT specifically targets alternative comparisons as its primary purpose, which aligns well with Reclamation's needs.
- Has native support for HEC-ResSim and HEC-RAS.
- Applications using HEC-5Q and CEQUAL-W2 already exist and can be used for guidance and as a basis for new applications.

HEC-RTS: Real Time Simulation

Developer: USACE Hydrologic Engineering Center
https://www.hec.usace.army.mil/software/hec-rts/
Brief Description: Comprehensive data acquisition and hydrologic modeling system for short-term decision support of water control operations in real time
Programming Language: Java
Most Recent Updates: 2021
Open Source: No (at this time)
Public API: Yes
Freely Distributable: Yes
Method of integrating models: RTS Model Plug-in API or Jython/Python Scripting
Model Coupling: loose
Includes Graphical User Interface (GUI) for Model Operators: Yes
Observations based on brief initial investigation:

- Project team has experience with this framework.
- HEC-RTS targets short-term operational decision support as its primary purpose, making it less suitable than HEC-WAT for Reclamation's needs.
- Has native support for HEC-ResSim and HEC-RAS.
