MODEL AND SIMULATION SCALABILITY TRAITS FOR INTERACTION (NEXUS) MODELING OF WATER AND ENERGY SYSTEMS

Mostafa D. Fard
Hessam S. Sarjoughian

Arizona Center for Integrative Modeling & Simulation
School of Computing and Augmented Intelligence
Arizona State University
Tempe, AZ, USA

ABSTRACT

It is beneficial to combine simulation models via I/O data exchanges. The Knowledge Interchange Broker (KIB) modeling approach can be used to develop interaction models that also have time, state, operations, and concurrency. A unique advantage of the interaction model is the composed models can have their own specifications. The KIB is used to model the nexus of the water and energy models of city metropolises. The RESTful WEAP, LEAP, and DEVS-Suite are used to model and simulate the composition of hybrid water, nexus, and energy models. The performance measurements of the simulations of these integrated simulators are evaluated. The results show the DEVS and Algorithmic interaction models have nearly identical computational times. These simulation times are contrasted with the use of links that share data between WEAP and LEAP models. This research highlights the interaction model flexibility and visibility at almost twice the computational time cost for data sharing.

Keywords: Hybrid Models, Interaction Models, Knowledge Interchange Broker, Water-Energy Systems.

1 INTRODUCTION

Effective management of interdependent water and energy systems is critical to sustainable development for cities, countries, regions, and the entire planet. Over the past few decades, analyzing issues within the Water-Energy Nexus (WEN) has become a topic of increasing attention for the scientific and policy communities (Dai et al. 2018). The rapid population, urban growth, and global economic development adversely affect limited natural and man-made resources, including water, energy, food, and land. Modeling the WEN is a challenging endeavor that requires extensive data on specific study areas (Zhang et al. 2019). Furthermore, the water and energy systems have bi-directional dependencies. For example, water is needed to cool the power plants, and energy is required to pump, treat, and distribute water. In addition to these internal interactions between the systems, external factors such as climate change, population growth, and economic instabilities increase the complexity of the Water-Energy system.

Domain experts can participate in collective work using existing tools to better use previously acquired knowledge and experience. Frameworks reduce the effort and resources needed for model development and simulation studies. The proprietary Water Evaluation and Planning (WEAP) (Sieber, Swartz, and Huber-Lee 2005) and Low Emissions Analysis Platform (LEAP) (Heaps 2022) tools used in this research are specialized for modeling, simulating, and evaluating the water and energy systems.

In this paper, we detail the use of a DEVS-Based Interaction Model (DEVS-IM) (Fard and Sarjoughian 2021b) developed for composing models defined using the WEAP and LEAP systems. We show the nexus of the water and energy models can be modeled independently to create flexible and scalable WEN models. This paper demonstrates the use of the DEVS-IM framework for composing the water and energy models of the Phoenix Active Management Area (Phoenix AMA) at scale. We also evaluate the computational cost of the RESTful Componentized WEAP (Fard and Sarjoughian 2021a) and Componentized LEAP simulators with a RESTful DEVS-Suite simulator relative to the Algorithmic-IM (Fard and Sarjoughian 2020) and the internally linked WEAP & LEAP systems for the Phoenix AMA (Fard et al. 2020).

2 BACKGROUND

2.1 WEAP & LEAP Tools with their Internal Linkage

The Water Evaluation and Planning (WEAP) system is a tool for modeling, simulating, and evaluating water systems. Models are defined as a network of water supply and demand entities (nodes) that are connected via transportation entities (links). A WEAP model defines water allocation from different sources through preferences and mass balance constraints. The WEAP tool is widely used globally for water allocation and management (Gao, Christensen, and Li 2017; Psomas et al. 2016; Amin et al. 2018). The Low Emissions Analysis Platform (LEAP) system is an integrated modeling and simulation tool that can be used to calculate energy consumption, production, and extraction in all sectors of an economy.

The structure of the WEAP and LEAP models are defined using different types of entities (e.g., *River*, *Demand Site*, and *Reservoir* in the WEAP system; and *Demand*, *Transformation*, and *Resource* in the LEAP system) with their Data (as the input) and Result (as the output) variables. Both systems are based on mass balance equations using the variables of the entities. They have a scenario-based approach that studies typically include a historical period (known as the Current Accounts). These models are simulated to test their abilities to replicate known statistical data using multiple forward-looking scenarios. Defining the time granularity for the water and energy models has an essential distinction in the WEAP and LEAP systems. The time-step in the WEAP system divides a year into a finite equal number of segments (e.g., yearly, monthly, or daily). Unlike time-steps in the WEAP system, the time granularity in the LEAP system (called time-slices) must be defined one by one, and the size of the segments can be different subject to the time-slices equal to a whole year.

From a model coupling perspective, the WEAP and LEAP systems use an internal linking mechanism that can bi-directionally share data to read the Data and Result variables from one to another. The WEAP-LEAP Internal Linkage allows basic data sharing but has some limitations. First, the WEAP and LEAP models must have the same time interval (start year and end year of the simulation) and the same time granularity in a year (the same time-steps/time-slices in a year). Otherwise, some data is lost. Second, both tools must run on the same machine. Third, the WEAP-LEAP Internal Linkage strongly depends on the internal structures and behaviors of the water and energy models. Any change related to the interactions in one model may cause changes in other models. Forth, the WEAP and LEAP tools must be executed manually and sequentially in the WEAP-LEAP Internal Linkage. Furthermore, the use of internal linkage for the WEAP and LEAP tools limits defining data transformations that should have multiple time resolutions and different control rules. The DEVS-IM framework lifts these limitations.

2.2 Componentized WEAP and Componentized LEAP Frameworks

Experts develop models of systems in terms of their parts and relationships. This approach allows some parts of a system to be modeled in detail while others are kept simple. This approach to multi-resolution modeling is important for constraining model complexity and scale. Furthermore, the need for component-based modeling and simulation is evident for understanding the interactions (nexus) among different parts of an integrated system (e.g., the WEN model) (Hoff 2011).

The Componentized WEAP and Componentized LEAP RESTful frameworks are designed and developed, given the significance of combining the defined model in the WEAP and LEAP systems with each other (to define the WEN model). Other types of modeling and simulation tools may be changed to be used for specialized domains (e.g., Food, Climate, and Economics). These frameworks have the same schemas for WEAP's and LEAP's projects, scenarios, and entities with variables mapped to separate components with input and output data. The WEAP and LEAP models are mapped to meta-components using the Ecore modeling methodology. In other words, the Componentized WEAP and Componentized LEAP frameworks are provided as a set of REST APIs to extract the structure of the defined models in the WEAP and LEAP systems (Fard and Sarjoughian 2021a). They also have a set of APIs to read and/or write the Data and Result variables (the simulation data) and other APIs to control the simulation execution.

2.3 Algorithmic Interaction Model

An Algorithmic Interaction Model (Algorithmic-IM) is developed based on the Knowledge Interchange Broker (KIB) approach (Sarjoughian 2006) for coupling the defined models in the WEAP and LEAP systems using the Componentized WEAP and Componentized LEAP frameworks (Fard and Sarjoughian 2021a). The interactions between the water and energy models in the Algorithmic-IM are defined as separate models. Thus, the water and energy models do not have direct knowledge of each other. The input and output relationships between the composed models are defined as two levels of hierarchically structured components (*Modules* and *Transformations*). The Algorithmic-IM has a cyclic, discrete time-step, and synchronous fixed execution protocol (consisting of six sequential steps); see Section 4.2 of (Fard and Sarjoughian 2020). The outcome was a concurrent and bidirectional interaction model for data mappings between the water and energy models (Fard and Sarjoughian 2020). However, this interaction model did not separate domain-specific model specification from its simulation execution protocol (a fundamental principle of the KIB modeling approach for model composability).

2.4 DEVS-Based Interaction Model

A component-based, hierarchical modeling approach that aligns with system thinking helps develop, reuse, and raise the interaction models' maintainability. The parallel Discrete Event System Specification (DEVS) has strong modularity, hierarchy, and support for discrete-time state transitions with inputs and outputs (Chow and Zeigler 1994). A DEVS-Based Interaction Model (DEVS-IM) is developed based on the DEVS formalism for coupling the defined models in the WEAP and LEAP systems using the Componentized WEAP and Componentized LEAP frameworks (Fard and Sarjoughian 2021a). The DEVS-IM model can be defined using *Input Connector*, *Output Connector*, *Process*, *Task*, *Port*, and *Coupling* entities. It also supports defining an interface for external systems (WEAP and LEAP systems) via *System*, *Component*, and *Function* entities. The DEVS-IM is supported with a RESTful framework to define the models' structure and store them in the MongoDB database. After defining the model via REST APIs, a code generator is used to generate meaningful java classes for the DEVS-Suite Simulator. Finally, the DEVS-Suite Simulator is used to define the behavior, simulate, test, and debug the interaction model (ACIMS 2022b, McLaughlin and Sarjoughian 2020). The DEVS-IM framework is grounded in system theory and component-based modeling. The model specification and execution protocol are separated in the DEVS-IM framework. It supports the model reusability, flexibility, and maintainability traits essential for developing realistic simulations of coupled energy and water systems.

3 RELATED WORK

Researchers with a variety of perspectives have been investigating the inter-relations between Water and Energy resources and others such as Climate and Food. These inter-relations were considered at different model scales: Urban, National, Regional, Global (Newell, Goldstein, and Foster 2019). Furthermore, the WEAP and LEAP tools are widely used by domain experts to establish the coupled model of the water and energy systems. For example, a study uses the WEAP and LEAP tools to design 26 scenarios to explore the energy/water saving of different policies in Beijing, China, over 2015 to 2050 (Liu et al. 2021). Another

study in Sacramento, California, over the period 1980 to 2001 with weekly time-step was undertaken on four climate scenarios to represent the impact of future temperature and precipitation extremes (Dale et al. 2015). In another study, the WEAP and LEAP were used for Xiamen, China, to define eleven future scenarios designed to explore the impacts of different factors on the urban WEN model from both supply and demand sides (Lin et al. 2019).

The Phoenix Active Management Area is a subject study area for understanding the WEN model. The water and energy systems serve as both demand and supply. A study of the WEN for the Phoenix AMA is carried out (Mounir, Mascaro, and White 2019; Guan et al. 2020) using the internally linked WEAP and LEAP tools. The models defined for the Phoenix AMA have 119 and 291 different entities. Also, there are nine interconnections from the energy system to the water system and 172 interconnections from the water system to the energy system. A WEN model has been developed using the Algorithmic-IM for the Phoenix AMA (Fard et al. 2020). It replicates the same nexus which was developed using the WEAP-LEAP Internal Linkage. The Algorithmic-IM model for the Phoenix AMA consists of 2 modules and 89 transformations. The computational cost of the Algorithmic-IM is approximately twice that of the WEAP-LEAP Internal Linkage. Furthermore, the DEVS-IM has a theoretical modeling foundation while also having a similar computational performance as the Algorithmic-IM (cf. Section 4.3 in (Fard et al. 2020)). The findings described in this paper show the formal system-theoretic modeling approach is advantageous, for example, in model verification.

4 PHOENIX AMA WATER-ENERGY NEXUS MODELING VIA DEVS-IM

Figure 1 presents the main steps to model and simulate water-energy nexus systems using the DEVS-IM framework. First, the structure of the interaction model must be defined. Second, code is generated for the skeleton of a complete project in the DEVS-Suite simulator. Third, the behavior of the interaction model is defined by identifying the external connectors and the data transformations under sequential and synchronous control schemes. Forth, the DEVS-Suite simulator is used to test, debug, and run the interaction model. This section describes the developed Phoenix AMA model using the DEVS-IM framework (Fard and Sarjoughian 2021b).

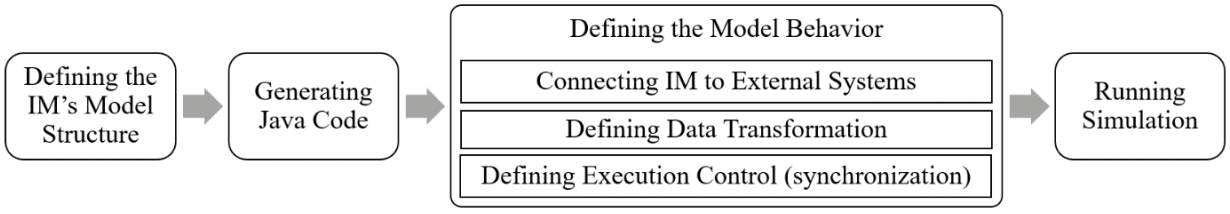


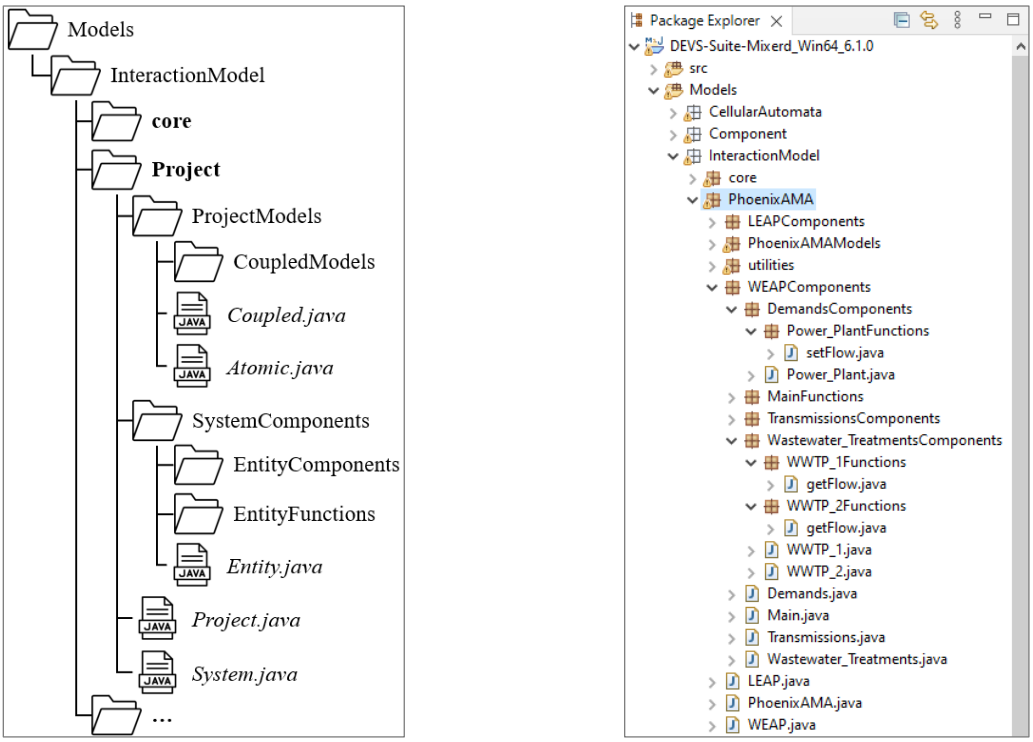
Figure 1: Steps of developing a model in the DEVS-IM Framework.

In the first step, the DEVS-IM REST APIs are used to define the structure of the Phoenix AMA Water-Energy model (via around 2,000 APIs). This DEVS-IM has two interfaces for the entities belonging to the Componentized WEAP and Componentized LEAP systems. The interfaces are defined using 181 *Component* and 144 *Function* elements (see Section 2.4). The interaction model has 15 *Process*, 91 *Task*, and 201 *Connector* elements. The interaction model has around 307 atomic and coupled DEVS models with 942 ports and 740 couplings. This interaction model replicates the same WEN model developed using the WEAP-LEAP Internal Linkage (same as the developed model using the Algorithmic-IM framework). The DEVS-IM model structure is verified for correctness before storing it in MongoDB.

In the second step, the skeleton of the DEVS-Suite Simulator project is created using the stored data in the MongoDB database (see Figure 2.a). A package with the same name as the DEVS-IM project (called *Project*) is added under the InteractionModel package (which is under the Models package). The root of the interaction model (a coupled DEVS model) is implemented using Java with the same name as the DEVS-IM project and a package (DEVS-IM project's name + Models) for the sub-models (i.e., Project.java

file and *ProjectModels* package). The same approach is applied to implement the Process entities of the interaction model (i.e., *Coupled.java* file and *CoupledModels* package). Also, all Task entities of the interaction model are implemented using Java (i.e., *Atomic.java* file). This approach is used for the whole hierarchy of the interaction model. Likewise, each external system is implemented using Java with the same name as the external system and a package (external system's name + Components) for its sub-components (i.e., *System.java* file and *SystemComponents* package). Each component is implemented using Java and two packages for its sub-components and functions (i.e., *Entity.java* file and *EntityComponents* and *EntityFunctions* packages). This approach is also used to define an interface for the WEAP or LEAP systems. Figure 2.b presents a portion of the generated files and packages for the DEVS-IM model in the DEVS-Suite Simulator. The "PhoenixAMA.java" file and "PhoenixAMAModels" package contain the required files to model the interaction model. Also, the "WEAP.java" and "LEAP.java" files and "WEAPComponents" and "LEAPComponents" packages contain the required files to define the interfaces for the WEAP and LEAP systems.

In Figure 2, the "core" package under the "InteractionModel" package contains the main classes for the elements of the DEVS-IM model (Fard and Sarjoughian 2021b). Default behaviors are defined for the Input Connector and Output Connector elements. Each Output Connector of the interaction model is connected to at least one Function of the external systems. The modeler must define the functionality of the Function elements in the external systems. For example, the modeler must define how the "setFlow" function (`Models/InteractionModel/PhoenixAMA/WEAPComponents/DemandsComponents/Power_PlantFunction/setFlow.java` in Figure 2.b) calls the "`…/Water/Phoenix-AMA/DemandSites/Power%20Plant/Inputs/Monthly%20Demand/Current%20Accounts`" API from the Componentized WEAP framework to write data to the "Monthly Demand" input variable of the "Power Plant" demand site in the "Current Accounts" scenario of the Phoenix AMA model in the WEAP system.



(a) Folder structure schema (b) The Phoenix AMA DEVS-IM model

Figure 2: DEVS-Suite Simulator generated project via DEVS-IM framework.

Having an interface for the external system increases the model complexity, but it has two main advantages. First, it promotes the interaction model to define the external system at different levels of abstraction or ignore some parts of the actual model. Second, it helps to have a high level of modularity between how to

get data from the external systems and how the interaction model is manipulating the sent/received data. As an example, the Phoenix AMA WEAP model has six demand-site entities, but just one of them is used inside the Phoenix AMA DEVS-IM model (presented via `Models/InteractionModel/PhoenixAMA/WEAPComponents/DemandsComponents/Power_Plant.java` in Figure 2.b).

Figure 3 shows a portion of the Phoenix AMA DEVS-IM model in the DEVS-Suite Simulator's SimView which hides the sub-models of some coupled models (presented in black-box mode). For example, the sub-models of the "Municipal_CAP" and "Municipal_SRP" coupled models are hidden in Figure 3. The black-box mode in the DEVS-Suite Simulator allows hierarchical viewing of large-scale models. The white-box mode (shown for "Municipal_Groundwater" coupled model in Figure 3) presents one level of the hierarchy for the coupled models. This viewing mode is for developing and debugging complex models via step-by-step tracking of the input and output messages among models and monitoring the states of atomic models.

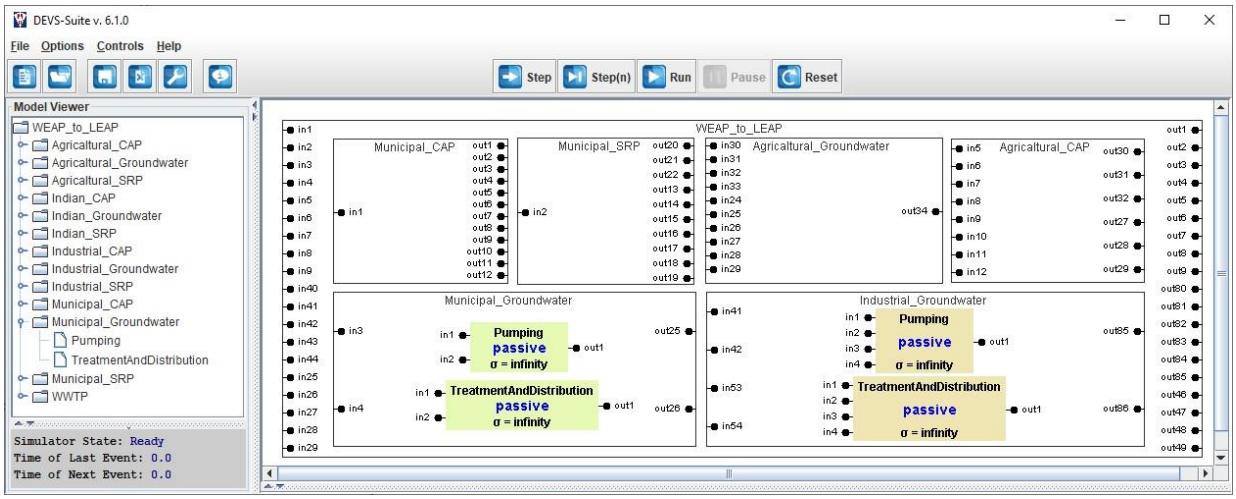


Figure 3: Hierarchical WEAP-LEAP Portion of the Phoenix AMA DEVS-IM model depicted in the DEVS-Suite Simulator's SimView.

Figure 4 presents another portion of the DEVS-IM model in the DEVS-Suite Simulator. The output connectors are connected to the Componentized WEAP and Componentized LEAP systems. The purple models are used to control the execution, and the gray models are used for time-based data transformations. Applying the data transformation to the received data from the external systems must be defined by the modeler via adding behavior to the *Task* elements (the Atomic models in the DEVS-Suite Simulator). Also, the execution control for the whole interaction model must be defined in a *Task* element named "Control". The "Control" model defines the ordering of receiving/sending data from/to the external systems and the order of executing the WEAP and LEAP systems. As shown in Figure 4, the "Control" model has five inputs ("start", "LEAP Executed", "LEAP Input Applied", "WEAP Executed", and "WEAP Input Applied") and four outputs ("Get LEAP Data", "Get WEAP Data", "Run LEAP", and "Run WEAP"). The "Control" model replicates the execution of the WEAP and LEAP systems via the WEAP-LEAP Internal Linkage.

Figure 5 presents a state machine for the "Control" protocol of the Phoenix AMA DEVS-IM model. Initially, two "`WEAPAppliedCount`" and "`LEAPAppliedCount`" variables (indicate the number of write data on the WEAP and LEAP systems) are set to zero, and the state is changed to "`Idle`". By receiving a message on the "start" input port, a message will be sent on the "runWEAP" output port, and the state will be changed to "`Running WEAP`". As shown in Figure 4, the message will be sent to the "Execute_WEAP" output connector, which calls an API from the Componentized WEAP framework to run the WEAP simulation. After completing the execution, a message will be sent on the "out" output port of the "Execute_WEAP" output connector. This message will be transferred to the "WEAP-Executed" input connector and then will be transferred to the "WEAP Executed" input port of the "Control" model.

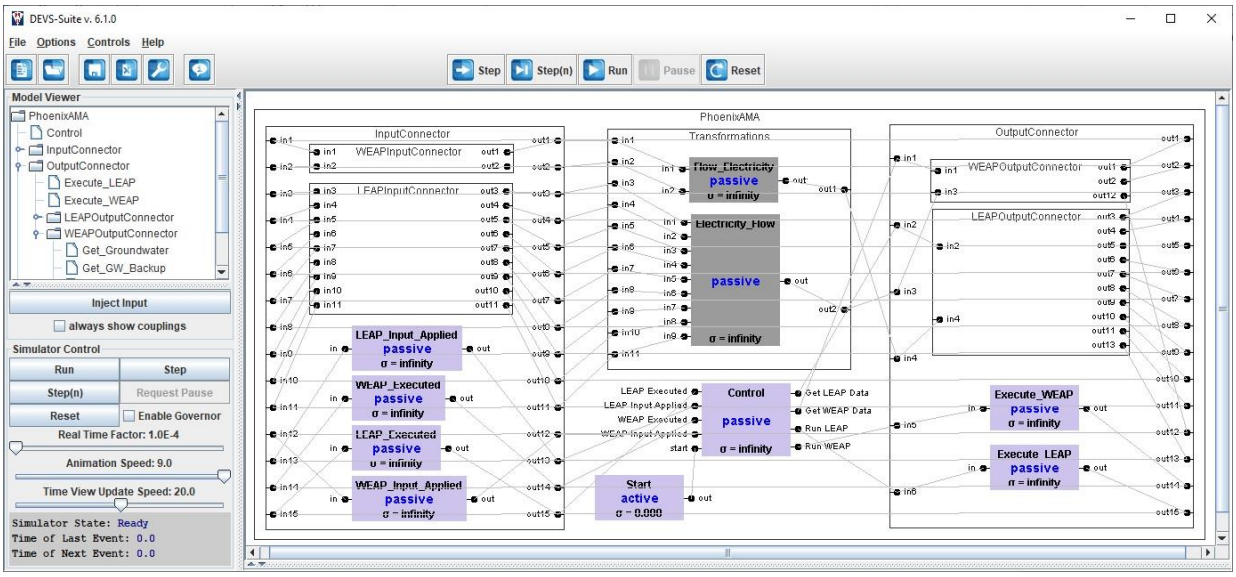


Figure 4: A portion of the Phoenix AMA DEVS-IM model shown in the DEVS-Suite Simulator.

As shown in Figure 5, the arrived message changes the state from the "`Running WEAP`" to the "`Getting WEAP Data`", and a message is sent on the "Get WEAP Data" output port. The output message will be sent to the output connectors to get data from the WEAP system (by calling the corresponding APIs from the Componentized WEAP framework). The received data from the WEAP system will be sent to the input connectors and then to a *Task* element to apply data transformation (e.g., the "Flow_Electricity" in Figure 4). After that, the transformed data will be sent to an output connector to send it to the LEAP system via calling a proper API from the Componentized LEAP framework. Then, the output connector sends an acknowledgment of applying data which will receive on the "LEAP Input Applied" input port of the "Control" model. The "`LEAPAppliedCount`" variable in Figure 5 increments one unit by receiving this message and checks the variable's value. The state will not change if the variable's value is less than the total number of write processes that must be applied to the LEAP system (the value 88 in Figure 5). Otherwise, the state will be changed to "`Running LEAP`" and the "`LEAPAppliedCount`" variable sets to zero. The same scenario happens for running the LEAP system, getting data from the LEAP system, and applying the transformed data to the WEAP system (see Figure 4 and Figure 5).

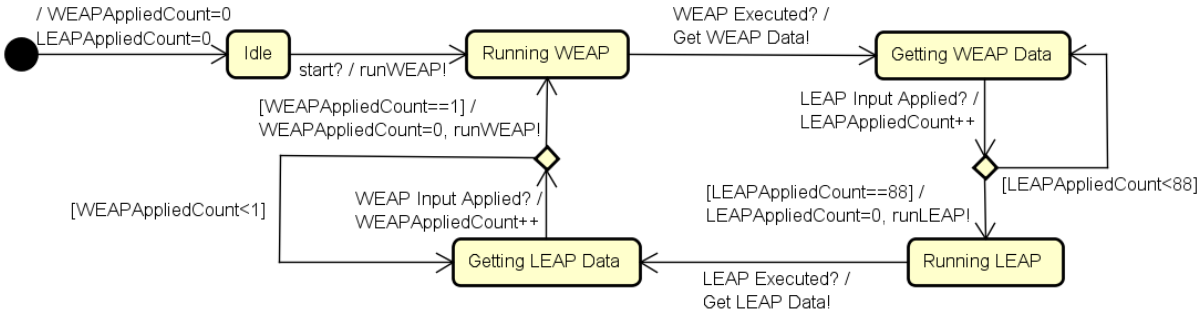


Figure 5: A state machine for the *Control* task element of the Phoenix AMA DEVS-IM model.

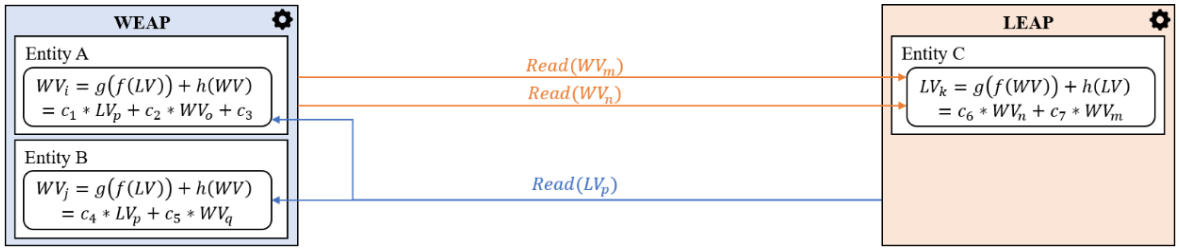
5 EVALUATION

Verification is the process of determining that a model implementation and its associated data accurately represent the developer's conceptual description and specifications (answering the question "Have we built the model, right?"). Validation is the process of determining the degree to which a simulation model and

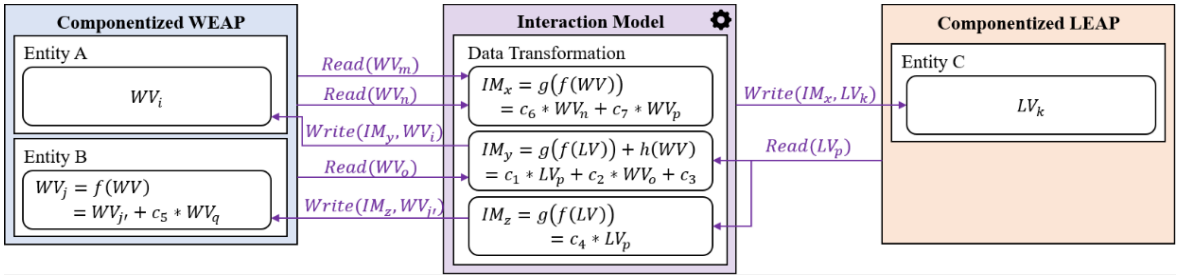
its associated data accurately represent the real world from the perspective of the intended uses of the model (answering the question "Have we built the right model?"). As mentioned in Section 4, the structure of the defined model is verified during the modeling process using the DEVS-IM framework.

5.1 Model Validation

The developed DEVS-IM Phoenix-AM model, described in Section 4, is converted from the corresponding Algorithmic-IM model (Fard et al. 2020). Moreover, both Algorithmic-IM and DEVS-IM models are replications of the Phoenix AMA model developed by domain experts using the WEAP-LEAP Internal Linkage (Guan et al. 2020; Mounir, Mascaro, and White 2019). Figure 6 illustrates the parts and their relationships for the Internal Linkage and Interaction Model. The $Read(x)$ on the arrows between the systems means reading the x variable from the source system. Also, the $Write(x, y)$ means writing the values of the variable x (from the source system) to the y variable (from the target system). The WV and LV in the formulas represent the WEAP Variables and LEAP Variables, respectively. In the WEAP-LEAP Internal Linkage (see Figure 6.a), the value of a WEAP/LEAP variable drives from some function(s) of the WEAP variables, some function(s) of the LEAP variable, and some constants. As an example, the values of the WV_i calculated by reading the LV_p variable from the LEAP system, reading the WV_o variable from the WEAP system, and constant coefficient values (i.e., c_1, c_2, and c_3).



(a) WEAP-LEAP Internal Linkage



(b) Interaction Model

Figure 6: Comparing the WEAP-LEAP Internal Linkage and Interaction Model.

In the Interaction Model approach (Figure 6.b), the data transformation formulas can be completely or partially applied to the data in the interaction model. For example, the computation formulas for the WV_i in Figure 6.b applied completely, and the computation formulas for the WV_j applied partially to the data in the Interaction Model. Completely applying data transformation to the Interaction Model formalizes and reduces the complexity of the relationship between the WEAP and LEAP models. In the WEAP-LEAP Internal Linkage, the systems are responsible for reading the required data from the other system and applying the data transformation before starting the execution (the $Read(x)$ on the arrows between the system in Figure 6.a). Whereas, in our approach, the Interaction Model is responsible for reading the data from the system, performing data transformations, writing the transformed data to the system, and controlling the execution of the componentized WEAP and LEAP systems (i.e., manage the $Read(x)$ and $Write(x, y)$ operations for time-driven data transformations shown in Figure 6.b).

A variety of methods are used to validate simulation models; for example, "*Face Validity*", "*Historical Data Validation*", "*Sensitivity Analysis*", etc. (Law 2019). The validation in this research is based on the

"*Comparison to Other Models*" method. The WEAP-LEAP Internal Linkage model is considered to be the real-world data for validating the DEVS-IM model. The initial states of the Phoenix AMA model in the WEAP and LEAP systems are the same as those in the WEAP-LEAP Internal Linkage and DEVS-IM models. The water and energy Phoenix AMA systems are defined as deterministic mass-balance equations. Given the same WEAP and LEAP models with the same initial states, the transformation formulas defined in the Interaction Model should have the same results as those of the internal linkage.

The results of executing the Phoenix AMA DEVS-IM model are validated in two scenarios. First, the WEAP model is executed (considering zeros for the dependent variables to the LEAP system), its results are read, and the transformed data are applied to the LEAP model (see Figure 5). Then, the LEAP system is executed (using the applied data by the Interaction Model), its results are read, and the transformed data are applied to the WEAP model. At this step, the results of both WEAP-LEAP Internal Linkage and DEVS-IM approaches are exported to CSV files. Then, a separate application (written in Java programming language) compares the results based on the components, variables, years, and time steps. The outcome shows that the results are identical in most cases. In some cases, the results have negligible differences ($\sim 10^{-6}$) due to the transferred data precision and computation/rounding mechanism. The WEAP and LEAP APIs allow extracting the data from their systems up to 15 digits (maximum six decimal places). However, their computation engines may use data with higher precision. Second, the same mechanism, but first running the LEAP system and then the WEAP system, is applied. Consequently, simulating the WEN model via the WEAP-LEAP Internal Linkage and the DEVS-IM have nearly identical results.

5.2 Performance Evaluation

The stages in their execution cycles are examined in detail to show the different modeling approaches described in the previous sections. Table 1, Table 2, and Figure 7 present the allocated time and their order for different execution steps for one round of simulating the Phoenix AMA model using three simulation approaches. In the WEAP-LEAP Internal Linkage, the WEAP and LEAP systems are running alternatively (the WEAP system runs first in this experiment). As shown in Table 1, the execution time of each system includes reading the required data (interconnection between the systems) from the other system (the amount is not distinguishable from the outside), applying the data transformations, then computing the results. As shown in Figure 7, the total time for an execution cycle for the WEAP-LEAP Internal Linkage is $\delta t_w + \delta t_l$. The periods for the water and energy models to read the data from one another are δt_{wr} and δt_{lr} ($0 < \delta t_{wr} < \delta t_w$ and $0 < \delta t_{lr} < \delta t_l$). For one complete simulation round, the WEAP system first reads the LEAP data; then, the water model executes. Next, the LEAP system first reads the WEAP data, and then the energy model executes (see Figure 7). The execution time of the Phoenix AMA model using the WEAP-LEAP Internal Linkage is 394.5 seconds, with around 19% and 81% of computation time consumed by the WEAP and LEAP systems, respectively.

Table 1: Phoenix AMA model time allocation for one round using WEAP-LEAP Internal Linkage.

	WEAP Execution		LEAP Execution		**Total**	**Quartile 1**	**Quartile 3**
	Read Data from WEAP	Apply Data Transformation	Read Data from LEAP	Apply Data Transformation			
WEAP-LEAP Internal Linkage	76.1		318.4		394.5	389.4	395.7

The numbers in Table 1 and Table 2 are the average of 10 different runs presented in the *second* unit. The last three columns show the center (median) and the spread of the data (Quartile 1 and Quartile 2). The same experiment has been performed for the same Phoenix AMA model (for the internal linkage and Algorithmic-IM) using older versions of the WEAP and LEAP system (Fard et al. 2020), which results in different execution times.

As mentioned before, the defined Phoenix AMA models in Algorithmic-IM and DEVS-IM frameworks replicate the execution regime in the WEAP-LEAP Internal Linkage (ACIMS 2022a). Table 2 presents the

order of different steps and their allocated time for the Phenix-AMA model simulated via Algorithmic-IM and DEVS-IM approaches. As shown in Figure 7, a complete simulation round in both approaches starts by running the WEAP system. Then, the WEAP results are read by calling the proper Componentized WEAP APIs in the DEVS-IM model. In the third step, the data transformations are applied to the received data. Finally, the results are sent to the LEAP system in the fourth step (via calling the Componentized LEAP APIs). The same scenario applies in the other direction, which means running the LEAP system, reading the data from the LEAP system, applying the data transformation, and writing the results to the WEAP system. In both approaches, applying the data transformation in the interaction model takes a negligible amount of time. The significant computation time is for writing data to the LEAP system (91% of the Algorithmic-IM approach and 93% of the DEVS-IM approach). It was observed in some of our experiments that based on the version of the WEAP, LEAP, and third-party dependencies and the free resources of the hardware/software, this step was as fast as around 600 seconds. Based on our last experiments, the total execution time for one round of the Phoenix AMA model via the Algorithmic-IM and DEVS-IM are 901.7 and 870.1 seconds, respectively. The Algorithmic-IM and DEVS-IM approaches have around 120% computational overhead compared to the WEAP-LEAP Internal Linkage approach.

Table 2: Phoenix AMA model time allocation for one round using Interaction Model approaches.

	WEAP Execution	Read Data from WEAP	IM Data Transformation	Write Data to LEAP	LEAP Execution	Read Data from LEAP	IM Data Transformation	Write Data to WEAP	**Total**	**Quartile 1**	**Quartile 3**
Algorithmic-IM	50	9.6	0.044	824	65.4	24.8	0.025	1.3	901.7	889.9	928.5
DEVS-IM	49.1	9.3	0.044	811.6	65.7	23.7	0.008	1.3	870.1	856.5	896.6

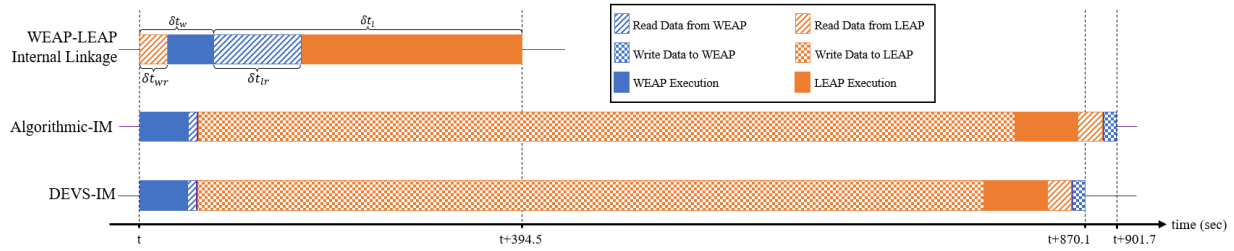


Figure 7: Phoenix AMA model execution time allocation via three simulation approaches.

A standalone desktop computer with Windows10 64-bit OS with four Core i5 Intel CPU and 20 GB RAM is used for all the experimental results included in this paper. The Componentized WEAP and Componentized LEAP frameworks are implemented using the NodeJS and TypeScript frameworks. The main third-party dependencies used in the frameworks are *Typescript-Node 8.10.2* for using Typescript in the NodeJS server-side application; *Express 4.17.13* is used to build a web application and APIs; *Routing-Controller 0.9.0* is used to create structured, declarative, and organized class-based controllers; *Body-Parser 1.19.1* to parse the incoming request to the webserver, and *Winax 3.3.4* is used to define *ActiveXObject* in NodeJS (create WEAP and LEAP instance in server-side applications). The Algorithmic-IM and DEVS-IM frameworks are implemented using Java 11. The Jersey (Kalin 2013) framework is used to call the RESTful web services for the Componentized WEAP and Componentized LEAP frameworks. The WEAP tool (version 2021.0101) and the LEAP tool (version 2020.1.0.56 32-Bit) are used for the above demonstration example.

6 CONCLUSIONS

Models of hybrid systems should be grounded on theories, methods, and frameworks that can facilitate the development of accurate and precise simulation scenarios. The interactions between different models that

are coupled should have well-defined models. They are essential for conducting trustworthy experiments and evaluations essential for decision-making for large, high-complexity hybrid water-energy systems. The I/O modular, component-based interactions models are a key for building frameworks that promote the development of composable hybrid models. The developed RESTful framework demonstrates the DEVS Interaction Model directly contributes to the transparency of building high fidelity simulations of water-energy systems. The computational cost of simulating the Phoenix Active Management Area water-energy system using the interaction model is about twice the cost of the obfuscated data sharing used in the WEAP-LEAP internal linkage. However, the benefit of composing hybrid water and energy models with interaction models outweighs its higher computational cost as compared with data sharing, especially when the key consideration is flexible model composability, not the amount of time it takes to simulate hybrid models.

ACKNOWLEDGMENTS

This research is supported by NSF under Grant #CNS-1639227. We thank the anonymous referees for their reviews on an earlier version of this paper.

REFERENCES

ACIMS 2022a. "Arizona Center for Integrative Modeling and Simulation". https://acims.asu.edu/modsim/weap-kib-leap/. Accessed June 01, 2022.

ACIMS 2022b. "Arizona Center for Integrative Modeling and Simulation". https://acims.asu.edu/devs-suite/. Accessed June 01, 2022.

Amin, A., J. Iqbal, A. Asghar, and L. Ribbe 2018. "Analysis of Current and Future Water Demands in the Upper Indus Basin under IPCC Climate and Socio-Economic Scenarios Using a Hydro-Economic WEAP Model". *Water*. 537.

Chow, A. C. H., and B. P. Zeigler. 1994. "Parallel DEVS: A Parallel, Hierarchical, Modular Modeling Formalism". *Winter Simulation Conference*, Orlando, Florida, US.

Dai, J., S. Wu, G. Han, J. Weinberg, X. Xie, X. Wu, X. Song, B. Jia, W. Xue, and Q. Yang. 2018. "Water-Energy Nexus: A Review of Methods and Tools for Macro-Assessment". *Applied Energy*. pp. 393-408.

Dale, L. L., N. Karali, D. Millstein, M. Carnall, S. Vicuña, N. Borchers, E. Bustos , and et al. 2015. "An Integrated Assessment of Water-Energy and Climate Change in Sacramento, California: How Strong is the Nexus?". *Climatic Change*. pp. 223-235.

Fard, M. D., and H. S. Sarjoughian. 2021a. "A RESTful Framework Design for Componentizing the Water Evaluation and Planning (WEAP) System". *Simulation Modelling Practice and Theory*. 102199.

Fard, M. D., and H. S. Sarjoughian. 2021b. "A RESTful Persistent DEVS-Based Interaction Model for the Componentized WEAP and LEAP RESTful Frameworks". *Winter Simulation Conference.* Phoenix, Arizona, USA.

Fard, M. D., and H. S. Sarjoughian. 2020. "Coupling WEAP and LEAP Models Using Interaction Modeling". *Spring Simulation Conference*, Virtual Event

Fard, M. D., H. S. Sarjoughain, I. Mahmood, A. Mounir, X. Guan, and G. Mascaro. 2020. "Modeling the Water-Energy nexus for the Phoenix Active Management Area". *Winter Simulation Conference*, Virtual Event.

Gao, J., P. Christensen, and W. Li. 2017. "Application of the WEAP Model in Strategic Environmental Assessment: Experiences from a Case Study in an Arid/Semi-Arid Area in China". *Journal of Environmental Management*. pp. 363-371.

Guan, X., G. Mascaro, D. Sampson, and R. Maciejewski. 2020. "A Metropolitan Scale Water Management Analysis of the Food-Energy-Water Nexus". *Science of The Total Environment*. 134478.

Heaps, C. G. 2022. "LEAP: The Low Emissions Analysis Platform." *Stockholm Environment Institute. Somerville, MA, USA.* https://leap.sei.org. Accessed June 01, 2022.

Hoff, H. 2011. “Understanding the Nexus: Background Paper for the Bonn2011 Conference”. *Bonn2011*, Bonn, Nordrhein-Westfalen, Germany

Kalin, M. 2013. “Java Web Services: Up and Running: A Quick, Practical, and Thorough Introduction”. O'Reilly Media, Inc.

Law, A. M. 2019. “How to Build Valid and Credible Simulation Models”. *Winter Simulation Conference.* National Harbour, Maryland, USA.

Lin, J., J. Kang, X. Bai, H. Li, X. Lv, and L. Kou. 2019. “Modeling the Urban Water-Energy Nexus: a Case Study of Xiamen, China”. *Journal of Cleaner Production*. pp. 680-688.

Liu, G., J. Hu, C. Chen, L. Xu, N. Wang, F. Meng, B. F. Giannetti, F. Agostinho, C. M. Almeida, and M. Casazza. 2021. “LEAP-WEAP Analysis of Urban Energy-Water Dynamic Nexus in Beijing (China)”. *Renewable and Sustainable Energy Reviews*. 110369.

McLaughlin, M. B., and H. S. Sarjoughian. 2020. “DEVS-scripting: a Black-Box Test Frame for DEVS Models”. *Winter Simulation Conference*, Virtual Event.

Mounir, A., G. Mascaro, and D. D. White. 2019. “A Metropolitan Scale Analysis of the Impacts of Future Electricity Mix Alternatives on the Water-Energy Nexus”. *Applied Energy*. 113870.

Newell, J. P., B. Goldstein, and A. Foster. 2019. “A 40-Year Review of Food–Energy–Water Nexus Literature and its Application to the Urban Scale”. *Environmental Research Letters*. 073003.

Psomas, A., Y. Panagopoulos, D. Konsta, and M. Mimikou. 2016. “Designing Water Efficiency Measures in a Catchment in Greece using WEAP and SWAT Models”. *Procedia engineering*. pp. 269-276.

Sarjoughian, H. S. 2006. “Model Composability”. *Winter Simulation Conference*. Arlington, Virginia, USA.

Sieber, J., C. Swartz, and A. H. Huber-Lee. 2005. “Water Evaluation and Planning System (WEAP): User Guide”. Boston: Stockholm Environment Institute.

Zhang, P., L. Zhang, Y. Chang, M. Xu, Y. Hao, S. Liang, G. Liu, Z. Yang, and G. Wang. 2019. “Food-Energy-Water (FEW) Nexus for Urban Sustainability: A Comprehensive Review”. *Resources, Conservation & Recycling*. vol. 142, pp. 215-224.

AUTHOR BIOGRAPHIES

MOSTAFA D. FARD is a Ph.D. candidate in the Computer Science program in the School of Computing and Augmented Intelligence (SCAI) at Arizona State University, Tempe, AZ, USA. His email address is smd.fard@asu.edu.

HESSAM S. SARJOUGHIAN is an Associate Professor of Computer Science and Computer Engineering in the School of Computing and Augmented Intelligence (SCAI) at Arizona State University, and the co-director of the Arizona Center for Integrative Modeling & Simulation (ACIMS), Tempe, AZ, USA. His research interests include model theory, poly-formalism modeling, collaborative modeling, simulation for complexity science, and M&S frameworks/tools. His email address is sarjoughian@asu.edu.