Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

MODULAR AND EXTENSIBLE PIPELINES FOR RESIDENTIAL ENERGY DEMAND MODELING AND SIMULATION

Swapna Thorve
Anil Vullikanti
Samarth Swarup
Henning Mortveit
Madhav Marathe

Biocomplexity Institute
University of Virginia
Charlottesville 22903
Virginia, USA

ABSTRACT

The landscape of residential energy modeling is changing rapidly. With increase in the availability of data, 'Modeling & Simulation' systems are becoming ubiquitous. However, reusing or extending these simulations is complicated due to sparse commonality in design and interoperability. One solution to this conundrum is developing modular and extensible pipelines. In this paper, we define a set of five pipelines inspired by microservices-oriented architecture. Four modular pipeline templates are defined, *Data Processing Pipeline*, *Modeling and Simulation Pipeline*, *Validation Pipeline*, *Visual Analytics Pipeline*; each encapsulating details of important tasks in modern-day complex systems. In addition, one custom pipeline is developed, for composing tasks that can be executed concurrently, called *Parallelizable Pipeline*. We instantiate this pipeline architecture for designing a synthetic energy demand modeling system. The value of the pipeline is demonstrated via three case studies – two of these studies provide new insights into issues related to equity and climate change impact.

1 INTRODUCTION

Modeling energy demand in the residential sector is becoming increasingly important to understand how to mitigate climate change, develop sustainable policies, perform energy efficiency retrofits, improve grid operations, and plan for future energy generation. Energy related datasets are becoming available to researchers from open and proprietary sources for analyses and developing models. This has led to a massive growth in techniques used for modeling residential sector energy consumption. A detailed review of techniques and types of datasets used in modeling efforts are listed in the following works (Swan and Ugursal 2009; Verwiebe et al. 2021). Due to lack of space, we only cite important methodology reviews and not individual models.

In particular, bottom-up modeling (e.g., agent-based models) and simulations are gaining importance in this domain since it allows for a detailed modeling approach (Rai and Henry 2016; Bustos-Turu et al. 2016; Tian and Chang 2020). Simulations developed using a bottom-up approach for modeling energy demands offer ample opportunities to understand heterogeneity in occupant behaviors, study effects of climate change on different population segments, or plan for solar adoptions in particular neighborhoods. For example, they allow simulation of disaggregated energy demand (Thorve et al. 2018) or simulate effects of electric vehicle adoption in a region (Bustos-Turu et al. 2014).

Bottom-up modeling techniques are highly data-driven and may become complex very quickly, thus making it hard to maintain them or replicate them. As a result, there can be multiple models with a

similar goal, but dissimilar in input data, a modeling component, or applicable to a limited spatial and temporal scope. This makes it difficult to re-use these modeling frameworks even if researchers make their simulation source code available. This is mainly because these frameworks have little commonality in design, e.g., there is no separation of concerns, making the framework tightly coupled and inflexible. There is also a lack of software infrastructure for addressing extensibility, reproducibility, composability, reusability, and interoperability for simulations. Establishing software design principles for developing modular and extensible frameworks for simulation tasks has great value in terms of accelerating development of bottom-up approach modeling frameworks in a reliable way and increasing human productivity. This will also be an important step towards democratization of simulations.

1.1 Contributions

We propose a design process rooted in software engineering principles for developing a flexible system architecture for energy demand modeling and simulation. Microservices-oriented architecture and Pipes & Filters architectural styles are applied to develop pipelines in simulations.

A set of five highly composable pipelines are defined to resemble the algorithmic workflows representing common processes such as data munging, modeling, validation, and visualization in modeling and simulation frameworks. The four pipelines are – (*i*) Data Processing Pipeline (DPP), (*ii*) Modeling and Simulation Pipeline (MSP), (*iii*) Validation Pipeline (VP), and (*iv*) Visual Analytics Pipeline (VAP). The fifth pipeline is called Parallelizable Pipeline (PP) that is influenced by dataflow paradigm for composing tasks that can be executed simultaneously.

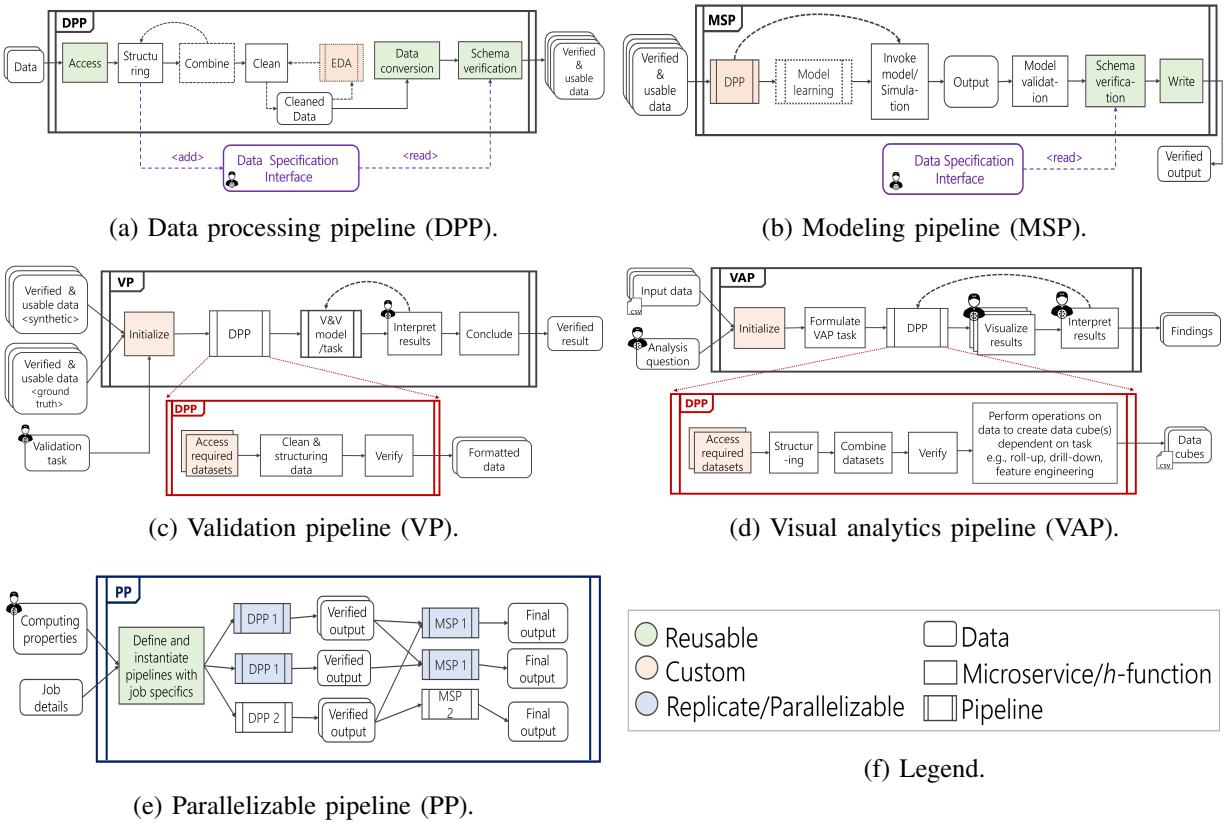
The proposed pipelines handle big data efficiently in a multi-level data processing approach. Multiple DPPs can be employed for processing different aspects of a dataset (e.g. convert raw data into a processed format, combine two processed datasets, store dataset in multiple formats and so on). Energy demand modeling requires domain knowledge and data context to fully understand its potential. This is accomplished in two ways – creating specialized functions in pipelines, and creating interface for handling domain context for datasets.

The value proposition of the proposed pipeline architecture is shown through three case studies. They provide an insight into three types of perspectives of the pipelines. The first case study demonstrates that pipelines are highly extensible, reduce effort involved in reproducibility, thereby enabling rapid development. As an example we show the process of substituting existing datasets in the simulations. The second study performs analyses of the simulation data by adding metadata from census to study effects of socio-economic variables on energy use. The analytics pipelines ingests large amount of data and generates insights via visualizations even at high spatial resolution. The third case study simulates climate change scenarios for observing change in energy demand at high spatial resolution. We can study important social good questions by modifying, adding, and reusing our pipelines in a timely and efficient manner.

Paper organization. First, we provide a brief literature review about pipelines and energy modeling. Then, we describe the general structure of pipelines, microservices, and their responsibilities. Once the formal model of our pipeline templates is defined, the proposed pipeline architecture is instantiated for energy demand modeling that generates high resolution synthetic energy data. Three case studies are described to illustrate the modularity, reusability, maintainability, and extensibility of the pipeline framework followed by a summary section.

2 RELATED WORK

Workflows and pipelines have been designed in many domains (e.g. genetics, bioinformatics, smart grid, online games) for automation of tasks, improved efficiency, re-usability, and better control of elements (Stoudt et al. 2021; Asaithambi et al. 2020; Cedeno-Mieles et al. 2020; Khalilnejad 2020; Simmhan et al. 2013). These works have shown that pipeline/workflow frameworks have supported streamlining of complex analyses tasks and duplicating or updating micro-tasks in tedious experiments much simpler.



(a) Data processing pipeline (DPP).

(b) Modeling pipeline (MSP).

(c) Validation pipeline (VP).

(d) Visual analytics pipeline (VAP).

(e) Parallelizable pipeline (PP).

(f) Legend.

Figure 1: **Proposed pipeline templates.** Five pipeline templates following the *pipe and filter* architectural design pattern are proposed for different stages of data-driven simulations. Filters are composed of modular functions (*h*-functions) that have properties of microservices-oriented architecture. Functions are chained together by data pipes. The user icon indicates that some functions require user input/domain expertise.

Several works have focused on designing reusable and reliable workflows in different application areas. Some examples are (Simmhan et al. 2009; Khalilnejad 2020; Raj et al. 2020). Data processing is an integral part of a large scale system and comes with many challenges (Pervaiz et al. 2019). It is essential to focus on understanding data and processing it appropriately to retain its value (Sambasivan et al. 2021). Koehler et al. (Koehler et al. 2017) present a methodology to automate data wrangling process by incorporating data context via user annotated schemas and rule based data repairs. (Khalilnejad 2020) presents a scalable time series data processing pipeline for building-level energy data on a HPC platform. (Simmhan et al. 2013) presents a cloud-based machine learning pipeline for dynamic demand response in smart grids. This pipeline performs data ingestion, machine learning modeling, and interaction with the system.

In spite of these efforts, there is scarcity of guidelines on software infrastructure for developing ABM simulations in the domain of energy demand modeling. The importance of designing a complete system, including software for all stages of the process, such as data processing, modeling and simulation, validation, visualization, and analytics, has not been addressed. Thus, we propose a set of five composable and extensible pipelines for designing ABM systems and simulations.

3 PIPELINES

Our pipelines are inspired by two designs: the microservices-oriented architecture (MSA) (Cerny et al. 2018; Wolff 2016; Mark 2016; Salah et al. 2016; Bayser et al. 2021) and the *Pipes and Filters* architectural

design pattern (Len, Paul, Clements, and Rick 2012). One of the biggest benefits of MSA is its usefulness for big data applications because of the ease of extensibility it provides. MSA consists of loosely coupled, reusable, specialized, and independent modules/functions that often work independently of one another. Thus, one unit module can work with its input(s) as a standalone entity with little to no dependencies. This gives the function enough room to be scaled in an individual fashion. The pipe and filter design pattern treats the filter as a black box function that can communicate with another filter using specific sets of channels called pipes. These pipes can be data, messages, or other information required by the filter. This type of architectures make it easy to maintain the system for rapid development and integration of workflows. These architectures provide flexibility so that only certain processes can be activated while keeping the remaining system untouched. Thus, they provide many benefits that are desirable properties for building bottom-up simulations.

A pipeline is a sequence of components, where each component takes a set of input(s) and produces a set of output(s). We define each component of the pipeline as a microservice (or *h*-function). Modularity and loose coupling characteristics of a microservice gives a clean structure to the pipelines, resulting in application of the *Pipes and Filters* pattern (Len et al. 2012). In our case, filters are microservices which encapsulate a functionality and pipes serve as connectors for data streams between two filters. Thus, a pipeline has chained and cooperative microservices assembled in a Pipe and Filter pattern to provide functionalities. We proceed by formalizing the pipeline framework and instantiating it for our application. Notations for the pipelines are described in Table 1.

Table 1: Notations.

Notation	Description
$\mathcal{P}$	The set of instances of pipeline templates.
DPP	The Data Processing Pipeline $DPP \in \mathcal{P}$.
MSP	The Modeling and Simulation Pipeline $MSP \in \mathcal{P}$.
VAP	The Visual Analytics Pipeline $VP \in \mathcal{P}$.
VP	The Validation Pipeline $VP \in \mathcal{P}$.
PP	The Parallelization Pipeline $PP \in \mathcal{P}$.
$\mathcal{H}$	The set of all microservices in the pipeline framework/system.
H	The set of microservices employed in a pipeline; $H \subset \mathcal{H}$.
h	A software implementation of a function as a microservice (*h*-function); $h \in H$.
R	A collection of all the datasets employed in the system stored in their original format along with any metadata. Our system stores raw data in formats such as flat files (e.g. csv files), pdf, images, and shapefiles.
r	An unprocessed dataset in its original format; $r \in R$.
D	A collection of curated, verified, and usable datasets obtained by processing datasets from collection R. Datasets in D are cleaned and stored in readily usable formats such as csv files, text files, and excel sheets so they can be easily utilised by other services in the system.
d	A curated, verified, and usable dataset; $d \in D$.
$\mathcal{I}$	A Data Specification Interface (DSI) stores information/metadata about different datasets $d \in D$ for lookup purposes.
I_d	A tuple $I_d = (a_d, s_d, f_d, l_d, e_d) \in \mathcal{I}$ for the dataset d where a_d is the access type of d, s_d is the schema, l_d is the location where d is stored, and e_d is the name of the dataset.

Energy demand modeling requires domain knowledge and context to fully understand the data potential. One way to achieve this is by creating interfaces for handling domain context for datasets. We call this interface Data Specification Interface (DSI) that incorporates domain knowledge while synthesizing data

from disparate sources. The domain expert/analyst aids in defining and annotating schemas for datasets. Our DSI has a global scope and is used for specifying a data-product schema s_d that will be populated by a data processing pipeline after ingesting a data source. The analyst/domain expert defines target schemas and annotates datasets. Let $\mathcal{I}$ be the DSI. Let I be a tuple in $\mathcal{I}$ which corresponds to a record for dataset d employed in the framework. Then, $I \in \mathcal{I}$ and $I = (a_d, s_d, f_d, l_d, e_d)$, where a_d stores access type and access properties of the data d, s_d is the target schema for data d, f_d is the storage format of d (e.g. database, flat files, etc), l_d is the location of the data, e_d is the name of the data. The user can perform classic CRUD (create, read, update and delete) operations on $\mathcal{I}$.

We define five pipelines in this work: four pipeline templates referred to as the (*i*) Data Processing Pipeline (DPP), (*ii*) Modeling and Simulation Pipeline (MSP), (*iii*) Validation Pipeline (VP), (*iv*) Visual Analytics Pipeline (VAP), and a custom pipeline called the (*v*) Parallelization Pipeline (PP) based on dataflow paradigms. A pipeline is constructed through a composition of microservices/*h*-functions and/or pipelines as building blocks. The 'composability' attribute of *h*-functions makes them highly reusable, modular, and independent. They can encapsulate a number of specialized services, support parallelization, and can flexibly be adapted for specific tasks.

Data Processing Pipeline DPP*(R,H).* The goal of this type of pipeline is to ingest raw data $r \in R$ and produce verified and usable data $d \in D$ in a specific format (i.e. target schema) s_d. This pipeline can also perform operations on multiple data $d \in D$ to produce a new data $d^{'} \in D$. In this process, the pipeline ingests data, cleans data, performs EDA (exploratory data analysis), create/update records in DSI (create mappings and schema definition), and store the verified data on a file system (optional). First, data access type is determined (e.g., read file, query database), unsupported requirements are addressed, and are then converted into *h*-functions. This is followed by data cleaning activities such as missing value omission/imputation, addressing duplication, and other EDA tasks. If multiple datasets are input to the pipeline, then data augmentation may also be a function in the pipeline. Once the data is formatted per target schema defined by user, the pipeline adds record(s) $I_d = (a_d, s_d, f_d, l_d, e_d)$ in the DSI $\mathcal{I}$ and stores 'verified and usable data' on the disk at location l_d. Figure 1(a) shows a DPP template. This pipeline performs the heavy lifting tasks such as data munging, data profiling, data synthesis, and creating/annotating schemas so that they can be easily assimilated in other pipelines in a uniform way. DPP pipelines are a first line of action for many long workflows in the framework (e.g., adding/replacing a dataset). Although this pipeline resembles many features of a typical data munging pipeline, we distinguish it by adding a way to handle data context/domain knowledge by defining a DSI. Thus, a DPP is defined as a transformation $DPP\colon R \times H \rightarrow D \times \mathcal{I}$ sending (r,h) to (d, I_d). This supports separation of concern and adds value to our pipelines.

Modeling and Simulation Pipeline MSP*(D,H).* Output of one or more DPPs is input to a MSP. Some of the *h*-functions in this pipeline may run simulations, invoke already trained ML models, train and test a ML model, perform model predictions, develop first-principle models, or validate model generated data. Some other data related functions include data conversions, and schema verification for input- and output data that are encapsulated by the DPP in this pipeline. Figure 1(b) shows a MSP template.

Validation Pipeline VP*(D,H,v).* The input to the VP comes from two different datasets to be compared/evaluated. The validation task *v* is also an input (user defined) so as to trigger and initialize the correct VP. The data is then converted to the required format and fed into the verification and validation (V&V) function/model/task. Results are then verified, visualized, and arrive at a conclusion depending upon *v*. Figure 1(c) shows a VP template. Thus, a VP is a transformation that maps data sets $D = (d_1, \ldots, d_n)$ to a verified result using the *h*-functions $H = (h_1, \ldots, h_n)$.

Visual Analytics Pipeline VAP. We define this pipeline in our application for special purpose. This pipeline extends the framework to incorporate scenario/intervention analysis. One example of this pipeline use is to study how energy use differs in income groups and population groups in a region. Our simulations generate high resolution data, and this pipeline is apt at converting data into insight. Figure 1(d) shows a VAP template. It is important to note that this pipeline takes domain expert/analyst queries as one of the inputs and the conceptualizes the task.

Parallelization Pipeline PP$(DPP_1,\ldots,DPP_n,MSP_1,\ldots,MSP_m,D,H,z)$. This type of pipeline can be built to run parallel instances of slow pipelines/*h*-functions within pipelines to improve runtime of the system or individual pipeline. The composition of such pipelines is completely user defined. Once the elements are appropriately assembled for the given task and computation details (e.g. number of instances) are provided in input z, the pipeline execution can be automated to produce desired results.

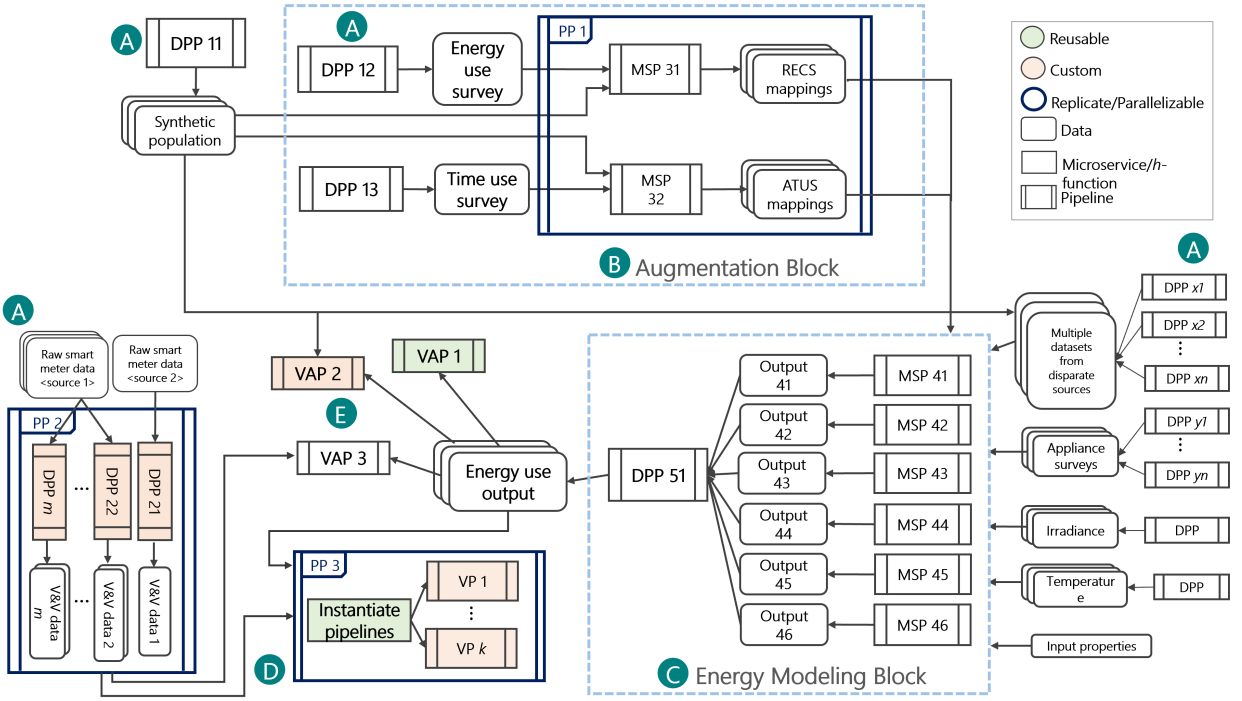


Figure 2: **Pipeline framework for residential energy demand modeling.** The figure shows a system-level view of pipeline interactions for the modeling and generation of synthetic energy demand. All the blocks marked with *A* indicate these are the first set of processes for ingesting a variety of data sources in different formats and converting them into usable data. Once the datasets are ready, we proceed with augmentation of a few important datasets (e.g. synthetic population) with domain-related information. These processes lay the foundation for high resolution simulations. The DPP and MSP pipelines for augmentation of synthetic population are shown in the *Augmentation Block* denoted by *B*. Pipelines encapsulated in Parallelizable Pipelines reduce execution time of larger tasks (e.g. *PP*1 runs pipeline chains independently in the *Augmentation Block*). *Energy Modeling Block* (*C*) takes inputs from datasets in *D*. Several data-driven and first principle MSPs generate disaggregated energy demand timeseries at household level. Then, we validate (denoted by *D*) the simulated data with ground truth with multiple procedures (VP). One can process this high resolution data to study characteristics of the generated dataset using VAP. The box in pink is highlighted for case study 1.

4 ENERGY DEMAND MODELING PIPELINE FRAMEWORK

The residential energy demand modeling framework generates household-level synthetic energy demand profiles for different end-uses at an hourly resolution using a bottom-up modeling approach. End-uses modeled are heating and cooling, hot water use, refrigerator, lighting, TV, and other appliances such as cooktops and oven, dishwasher, washer and dryer. Different models and multiple datasets from disparate

sources are used in modeling different end-uses. Figure 2 shows this in the blue dotted box 'Energy Modeling Block'.

We design the energy modeling simulation in a bottom-up approach. A bottom-up approach in simulations relies on detailed designing of components of system and then integrate these components in a meaningful and recursive way until the system is whole. This gives way to formulating a pipeline framework and delineation of different types of tasks in large-scale systems such as data processing, modeling, and validation. We instantiate the pipeline templates to outline our residential energy demand modeling framework as shown in Figure 2.

Data processing is one of the most important tasks in our system since we have data from a variety of sources. The datasets have multiple formats and resolutions (e.g. by minute, normative day, annual, statistical representations of the entire population, small samples of population groups). These datasets differ largely in volume. For example, the synthetic population is statistical representation of households in a region (e.g. Virginia state has 3M households) whereas the energy use survey is available for 5k households. We follow a multi-layered approach for processing the data through DPPs. In the first stage, raw datasets employed in the system are converted into a *verified and usable data* via a DPP. These are marked by *A* in Figure 2. Then, as required by the modeling task, we harmonise different datasets and/or engineer model features to create appropriate inputs for DPPs.

Depending upon the runtime of subsequent pipelines, the outputs may or may not be written to disk. This is one of the advantages of having microservices. Data can be stored on disk after benchmark actions so as to avoid re-running tedious pipelines. MSPs denote modeling pipelines in the framework. MSPs perform specialized data transformations, feature engineering, and build the model. For example, MSP31 trains a multivariate ML model using a survey population and is used for prediction on synthetic population. The output ('RECS mappings') of this pipeline is written to disk since this task is performed only once and it is compute intensive.

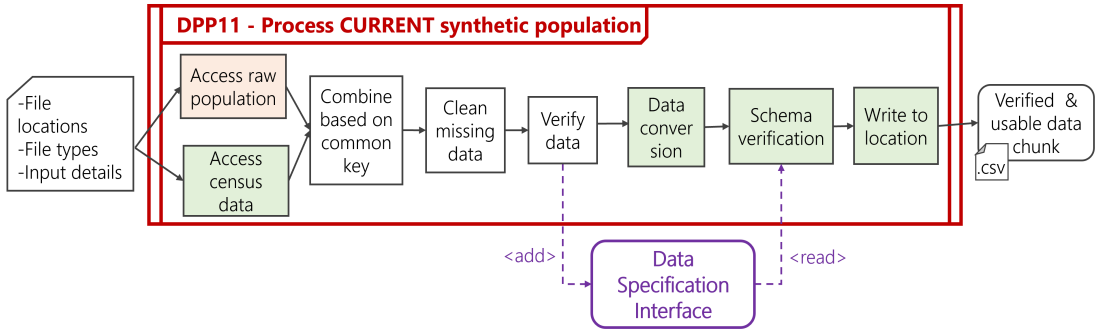
The synthetic population and augmentation outputs are input to the *Energy Modeling Block*. This block is responsible for generating energy demand profiles at household-level and hourly resolution for different end-uses. Thus, we see multiple MSPs in this block. For example, MSP46 is the modeling pipeline for simulating the duration and time of appliance use such as dishwasher, laundry appliances, and cooking appliances. DPP within this pipeline will harmonise appliance surveys with household information from synthetic population, and occupancy information from respective ATUS mapping, and appliance ownership information from the respective RECS mapping. Further details of the pipeline framework for energy demand modeling can be found in Figure 2.

Our pipeline framework can separate domain invariant *h*-functions from context-aware *h*-functions that incorporate domain knowledge. Considering the big data aspect in our framework, we harness the power of DPPs to address the volume, variety, and veracity of datasets in a staggered multi-layered approach.

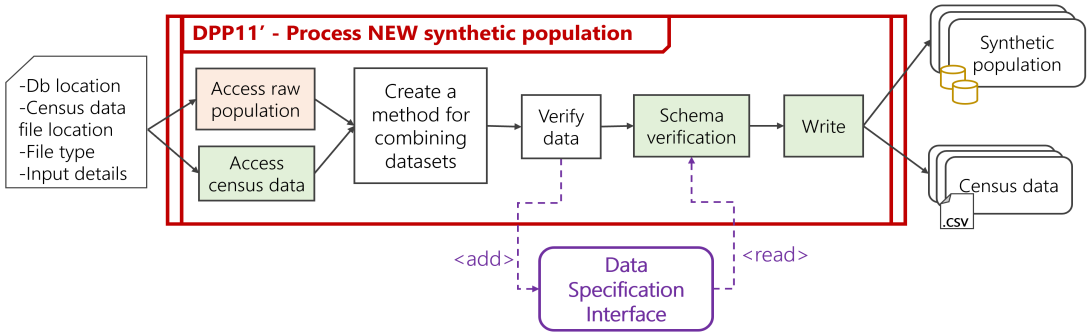
5 CASE STUDIES

This section outlines three case studies to demonstrate the value of the proposed pipeline architecture. The first case study shows how a new data source can be substituted for an existing data source in the system. This study highlights that having an extensible software infrastructure in place, speeds up effort needed by researchers to add new functionality to the system. Case studies 2 and 3 analyze energy related questions at different spatial resolutions in Virginia (VA) state, U.S. for 3.3 million households. Case study 2 analyzes effects of social, economic, and dwelling characteristics on energy consumption. The case study shows how VAP pipelines are used to formulate these studies to reduce researcher's time for conducting this experiment. Case study 3 shows examples of modeling and simulating future energy related scenarios such as effects of global warming (specifically, temperature rise) in different regions in VA. This experiment adds a new dataset in the system, executes the energy demand modeling framework, and then uses VAP pipelines to analyze the relevant datasets and report findings through data cubes and visualizations. The

modularity of pipelines demonstrates how easy it is to extend the current architecture to study future climate change scenarios.



(a) DPP11



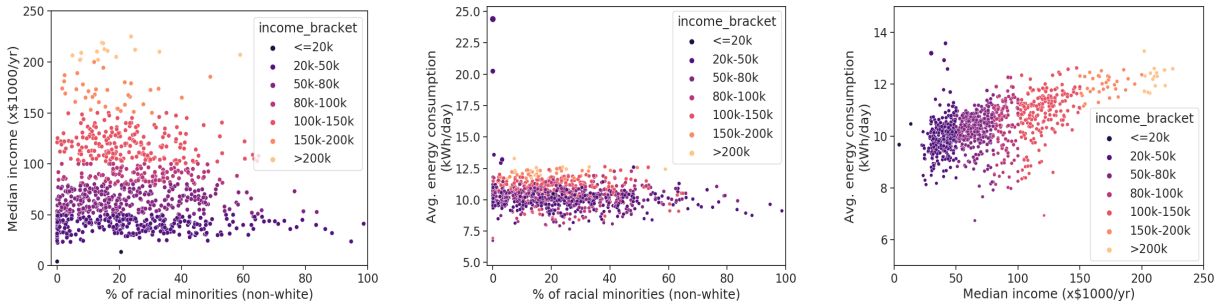
(b) DPP11'

Figure 3: **Data substitution.** This figure shows an example of data substitution. Let dataset d is processed by DPP11 and dataset $d^{'}$ be processed by DPP11'. In the process of substituting the synthetic population dataset from d to $d^{'}$ we replace the pipelines from DPP11 by DPP11'. The individual components within the pipelines are the microservices/$h-$functions that process small pieces of information.

5.1 Study 1: Data Substitution

One of the major benefits of using microservices oriented architecture for big data applications is the ease of extensibility it provides. Extensions can be through the addition of new data/functions or through modification of existing data/functions. The goal of this case study is to show how pipelines (specifically DPP) can be used to replace an existing data source with a new data source in the framework (e.g., substituting a synthetic population data d with another synthetic population data $d^{'}$. This study highlights modularity and extensibility characteristics of the pipeline framework.

We replace the existing synthetic population dataset d with a new dataset $d^{'}$. Overall, we want to make minimum number of changes to the system while replacing a data source. Figure 3(b) shows the new data pipeline DPP11' that will be substituted in place of pipeline in Figure 3(a) DPP11. Note that, DPP11 will be substituted by DPP11' in Figure 2. A new data pipeline is developed for $d^{'}$ since the format and method of accessing this dataset is different than that of dataset d. $d^{'}$ population is accessible via a database whereas the current access mechanism for d is flat files. Thus, DSI is updated and we add a new 'access' microservice for $d^{'}$. When the pipeline is replaced in Figure 2, the overall operating mechanism of the system does not change. We only substitute DPP to switch the dataset. Thus, the pipeline architecture is able to accommodate these changes with ease.



(a) Income vs. racial minority (b) Energy use vs. racial minority (c) Energy use vs. income

Figure 4: Energy use is simulated for a summer day in Virginia. A dot in the scatter plot represents a census tract. (a) Higher income bracket population seems to reside census tracts with lower percentage of racial minorities (correlation=-0.08). (b) Slightly negative correlation between energy use and % of racial minority groups (correlation=-0.13). (c) Higher income groups consume more energy (correlation=0.46).

5.2 Study 2: Socio-economic Analyses of Synthetic Energy Demand Data

This goal of this case study is to examine the effects of social, economic, and dwelling features on energy use in different census tracts of Virginia. Two analyses are perfomed – *(i)* the effects of income and race on energy use, *(ii)* the influence of floor area on energy use in urban, rural, and cluster areas. This study is outlined to highlight composability and extensibility characteristics of the pipeline framework. It also shows that with a software architecture in place for designing pipelines, it is very easy to perform such analyses, thus increasing human productivity.

Income, race, and energy use. A VAP is designed to analyze demographic data from census and energy output from the modeling system. Aggregation operations are performed on the data to roll-up from household level to census tract level generating data cubes on spatial resolution, income brackets, and race. Figure 4 shows that energy use tends to increase with income and decrease with increase in minority groups. The pipelines aid in querying different datasets, combining them, and generating data cubes across multiple dimensions. This process is extremely time efficient to generate results from the VAP.

Floor area and energy use in urban and rural areas. A VAP is designed to analyze energy use vs. floor area at census tract level in Virginia. Floor area and energy demand are both output of the energy modeling framework. Aggregation operations are performed on the data to roll-up from household level to census tract level generating a data cube. The data cube is then augmented with urban and rural annotations at census tract level by processing census shapefiles. Figure 5 displays a scatter plot of energy usage vs. median floor area at census tract level for Virginia state and the VAP designed for this case study. At a glance, we can see which census tracts can potentially be targeted for decarbonization (e.g. quadrants labeled 'Large floor area, High energy' and 'Small floor area, High Energy').

5.3 Study 3: Examining Effects of Climate Change

This goal of this experiment is to examine the effects of climate change in different regions of Virginia. This study is outlined to highlight reproducibility, reusability, composability, scalability, and extensibility characteristics of the pipeline framework. Three different climate change scenarios are simulated for a summer day in VA. Representative Concentration Pathway (RCP) scenarios that limits global warming are simulated for average temperature rise corresponding to 3.6F (RCP 2.6), 5.4F (RCP 4.5), and 9F (RCP 8.5). A new DPP is composed for generating future temperature data under each of the scenarios at county level. A schema is added in the DSI and annotated by the user/researcher. This is dataset is then plugged in the energy demand modeling framework. This shows that the framework is extensible and reusable.

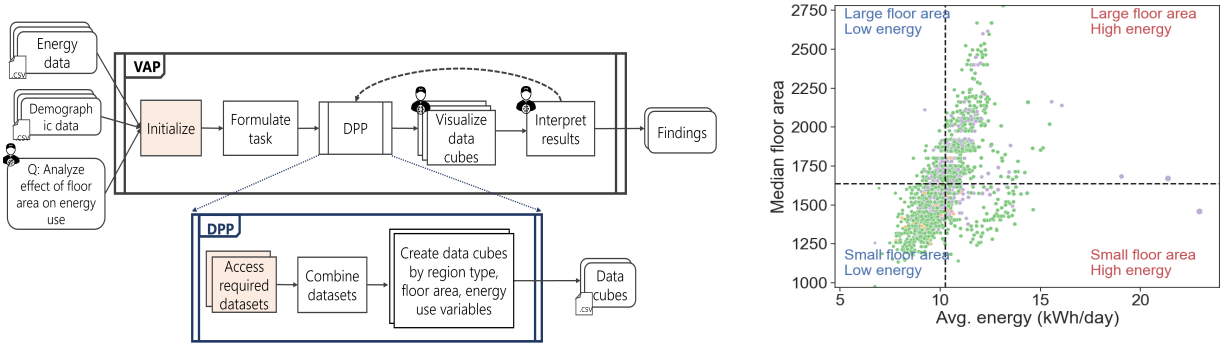
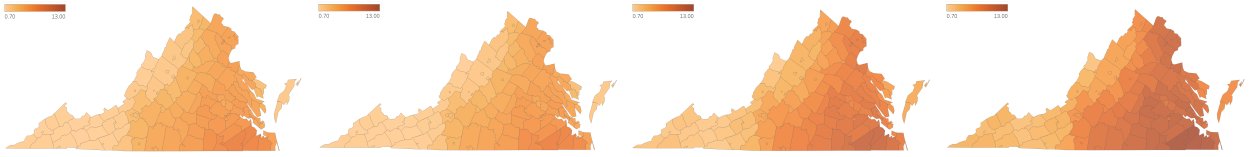


Figure 5: Energy use vs. floor area: The VAP is shown on the left and the scatter plot on the right displays energy usage vs. median floor area at census tract level. Each point is colored to display its area type. Quadrants are drawn by plotting averages for the axes (correlation = 0.546).

The same energy demand modeling framework is used to reproduce results for all the RCP scenarios. The researcher can execute each of these scenarios in parallel and speed up the process of obtaining results. A VAP is developed for analyzing the output data. The output of this pipeline are datacubes aggregated from household level to county level for different scenarios. This pipeline collates the data very easily to formulate a researcher defined question and analyze the results via visual aids. Figure 6 shows the effect of climate change on air conditioner energy use for a summer day under different scenarios. The simulation results are shown for 8 July 2014, RCP 2.6, RCP 4.5, and RCP 8.5 scenarios. The southeast counties of Virginia are the most vulnerable to climate change. The temperature change is shown in histogram (Figure 6(c)).



(a) 8 Jul, 2014 (b) RCP2.6 (c) RCP 4.5 (d) RCP 8.5

Figure 6: **Effect of climate change in Virginia**. Heatmaps are used to show average increase in energy usage by air conditioners on a summer day in Virginia. The results are shown at county level. It is observed that southeast counties are the most vulnerable to climate change.

6 SUMMARY

The proposed pipeline templates implement important tasks performed by modern-day complex software systems. Note that, each pipeline is composed of loosely coupled microservices. Thus, the templates can be extended/tweaked for architecting software systems in other domains too. One such example is designing analytical and data processing microservices for smart city transportation (Asaithambi et al. 2020). Koehler et al. demonstrate an example of incorporating domain knowledge in big data systems (Koehler et al. 2017). In another example, microservices oriented architecture is adopted for designing controlled networked social science experiments. A set of five highly composable and extensible pipelines for modeling residential energy demand have been presented along with modular and specialized building blocks called *h*-functions in data, modeling, validation, and analytics pipelines. A fifth custom pipeline is proposed based on dataflow paradigm for composing pipelines to speed up the execution time of long-running tasks. This conceptual

approach of our pipelines satisfies reproducibility, reusability, separation of concern, high maintainability, and extensibility properties of efficient software design. Domain knowledge and data context is incorporated in the pipelines via specialized microservices and a DSI. Our case studies illustrate that pipelines offer great potential to study intervention scenarios in social good applications with minimum effort.

ACKNOWLEDGMENTS

This work is partially supported by University of Virginia Strategic Investment Fund award number SIF160, NSF EAGER CMMI-1745207, NSF Grant OAC-1916805, and the UVA 3 Cavalier Grant titled "Exploring critical roadblocks to decarbonization: distributed energy resources and the grid" (166914-WMS5F-LC00295-30015).

REFERENCES

Asaithambi, S. P. R., R. Venkatraman, and S. Venkatraman. 2020. "MOBDA: Microservice-Oriented Big Data Architecture for Smart City Transport Systems". *Big Data and Cognitive Computing* 4(3).

Bayser, M., V. C. V. B. Segura, L. G. Azevedo, L. P. Tizzei, R. M. Thiago, E. F. S. Soares, and R. Cerqueira. 2021. "DevOps and Microservices in Scientific System Development". *Computing Research Repository.* abs/2112.12049.

Bustos-Turu, G., K. H. van Dam, S. Acha, C. N. Markides, and N. Shah. 2016. "Simulating Residential Electricity and Heat Demand in Urban Areas Using an Agent-based Modelling Approach". In *IEEE International Energy Conference*, 1–6. April 4th-8th, Leuven, Belgium.

Bustos-Turu, G., K. H. van Dam, S. Acha, and N. Shah. 2014. "Estimating Plug-in Electric Vehicle Demand Flexibility Through an Agent-based Simulation Model". In *IEE PES Innovative Smart Grid Technologies, Europe*, 1–6. Oct 12th-15th, Istanbul, Turkey.

Cedeno-Mieles, V., Z. Hu, Y. Ren, X. Deng, N. Contractor, S. Ekanayake, J. M. Epstein, B. J. Goode, G. Korkmaz, C. J. Kuhlman, D. Machi, M. Macy, M. V. Marathe, N. Ramakrishnan, P. Saraf, and N. Self. 2020, 11. "Data Analysis and Modeling Pipelines for Controlled Networked Social Science Experiments". *PLOS ONE* 15(11):1–58.

Cerny, T., M. J. Donahoo, and M. Trnka. 2018, Jan. "Contextual Understanding of Microservice Architecture: Current and Future Directions". *ACM Special Interest Group on Applied Computing Review* 17(4):29–45.

Khalilnejad, A. 2020. "Automated Pipeline Framework for Processing of Large-scale Building Energy Time Series Data". *PLOS ONE* 15(12):1–22.

Koehler, M., A. Bogatu, C. Civili, N. Konstantinou, E. Abel, A. A. A. Fernandes, J. Keane, L. Libkin, and N. W. Paton. 2017. "Data Context Informed Data Wrangling". In *2017 IEEE International Conference on Big Data*, 956–963. Dec 11th-14th, Boston, MA, USA.

Len, B., Paul, Clements, and K. Rick. 2012. *Software Architecture in Practice, Third Edition*. New Jersey: Addison-Wesley Professional.

Mark, R. 2016. *Microservices vs. Service-Oriented Architecture*. California: O'Reilly Media, Inc.

Pervaiz, F., A. Vashistha, and R. Anderson. 2019. "Examining the Challenges in Development Data Pipeline". In *Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies*, 13–21. Jul 3rd-5th, Accra, Ghana.

Rai, V., and A. D. Henry. 2016, Jun. "Agent-based Modelling of Consumer Energy Choices". *Nature Climate Change* 6(6):556–562.

Raj, A., J. Bosch, H. H. Olsson, and T. J. Wang. 2020. "Modelling Data Pipelines". In *46th Euromicro Conference on Software Engineering and Advanced Applications*, 13–20. Aug 26th-28th, Portoroz, Slovenia.

Salah, T., M. Jamal Zemerly, C. Y. Yeun, M. Al-Qutayri, and Y. Al-Hammadi. 2016. "The Evolution of Distributed Systems Towards Microservices Architecture". In *11th International Conference for Internet Technology and Secured Transactions*, 318–325. Dec 5th-7th, Barcelona, Spain.

Sambasivan, N., S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo. 2021. "Everyone Wants to Do the Model Work, Not the Data Work: Data Cascades in High-Stakes AI". In *Conference on Human Factors in Computing Systems (CHI)*. May 8th-13th, Yokohama, Japan.

Simmhan, Y., S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna. 2013. "Cloud-Based Software Platform for Big Data Analytics in Smart Grids". *Computing in Science Engineering* 15(4):38–47.

Simmhan, Y., C. van Ingen, A. Szalay, R. Barga, and J. Heasley. 2009. "Building Reliable Data Pipelines for Managing Community Data Using Scientific Workflows". In *Fifth IEEE International Conference on e-Science*, 321–328. Dec 9th-11th, Oxford, UK.

Stoudt, S., V. N. Vásquez, and C. C. Martinez. 2021, 03. "Principles for Data Analysis Workflows". *PLOS Computational Biology* 17(3):1–26.

Swan, L. G., and V. I. Ugursal. 2009. "Modeling of End-use Energy Consumption in the Residential Sector: A Review of Modeling Techniques". *Renewable and Sustainable Energy Reviews* 13(8):1819–1835.

Thorve, S., S. Swarup, A. Marathe, Y. Chungbaek, E. K. Nordberg, and M. V. Marathe. 2018. "Simulating Residential Energy Demand in Urban and Rural Areas". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 548–559. Gothenburg, Sweden: Institute of Electrical and Electronics Engineers, Inc.

Tian, S., and S. Chang. 2020. "An Agent-based Model of Household Energy Consumption". *Journal of Cleaner Production* 242:118378.

Verwiebe, P. A., S. Seim, S. Burges, L. Schulz, and J. Müller-Kirchenbauer. 2021. "Modeling Energy Demand: A Systematic Literature Review". *Energies* 14(23).

Wolff, E. 2016. *Microservices: Flexible Software Architecture*. United States: Leanpub.

AUTHOR BIOGRAPHIES

SWAPNA THORVE is a PhD student in the Computer Science department at University of Virginia and a graduate research assistant in Network Systems Science and Advanced Computing, Biocomplexity Institute and Initiative. Her email address is st6ua@virginia.edu.

ANIL VULLIKANTI is a Professor in the Biocomplexity Institute and Initiative, and the Department of Computer Science. His interests are in network science, and the foundations of AI and machine learning. His email address is vsakumar@virginia.edu.

HENNING S. MORTVEIT is an associate professor in the Biocomplexity Institute and Initiative and the Department of Engineering Systems and Environment at the University of Virginia. His research covers massively interacting systems, their mathematical structures, theory, and related software design and computational architectures. His email address is Henning.Mortveit@virginia.edu.

SAMARTH SWARUP is a Research Associate Professor in the Biocomplexity Institute and Initiative at the University of Virginia. His research interests are in large-scale agent-based simulations and machine learning applied to problems in public health and social science. His email address is swarup@virginia.edu.

MADHAV MARATHE is a professor of Computer Science and a Distinguished professor in Biocomplexity, University of Virginia. He is a fellow of ACM, SIAM, IEEE and AAAS. His research interests are modeling and simulations, network science, sustainability, AI. Email marathe@virginia.edu