Construction of Complex Stock-and-Flow Models Based on Modular Decomposition and System Integration Using Shared State Variables and Subdivided Time Steps

Bo Hu[1], Bo Zhang[2]
[1] Universität der Bundeswehr München, Neubiberg, Germany
[2] Information Center, Ministry of Ecology and Environment, Beijing, China

Abstract

This paper proposes a new method for modular development and integration of system dynamics models. Existing system dynamics models can be decomposed into several modules, or modules can be directly built as required. These modules each embody specific functions, can operate independently, and are less complex than the entire model, making them easier to understand. Such modules can be combined using shared state variables and subdivided time step techniques introduced here to achieve complete model functionality. The purpose of developing this method is to lay the foundation for establishing a System Dynamics module library, taking the first step towards a "model factory".

Keywords: model decomposition and composition, module library, model factory

1. Introduction

In today's era of rapid social and technological development, various complex systems are prevalent in numerous natural and social domains, such as ecosystems, financial markets, transportation networks, and large-scale engineering systems. These systems have intricate internal structures, with extremely frequent interactions among their constituent elements, exhibiting high levels of dynamism and uncertainty. As the scale of systems continues to expand and the complexity of their functions continues to increase, traditional system analysis and modeling methods are increasingly unable to cope with the challenges posed by complex systems.

System dynamics, as a discipline dedicated to studying the dynamic behavior of complex systems, provides an effective means for deeply understanding the underlying principles of complex systems by constructing models that simulate internal feedback mechanisms, information transmission, and interrelationships between variables. However, when faced with ultra-large-scale and complex systems, directly constructing holistic dynamic models is often fraught with difficulties, not only easily leading to chaos during the model construction process but also posing significant obstacles to subsequent model maintenance, debugging, and expansion. Against this backdrop, the concept of modular decomposition has emerged and gradually become a key strategy for solving the challenges of complex system dynamics modeling.

Modularity is crucial for developing complex systems, facilitating easier management during development, especially with multi-team collaboration; enhancing overall system modifiability due to the local ease of module changes; and making the entire system more easily designed and understood (DeRemer & Kron 1976). Over the years, several modular applications have also emerged in the field of system dynamics. For instance, a modular approach has been implemented in the healthcare sector, where multiple models from various areas, such as population dynamics, disease dynamics, healthcare, and healthcare financing, are loosely coupled (Djanatliev et al., 2012). Elmasry and Größler (2016) presented their research efforts and achievements in adopting modular models in the field of supply chain. In the construction industry, Karl (2016) introduced a system dynamics library (SDL) approach to enable "multidisciplinary teams to develop joint models from varying perspectives".

Increasing system complexity necessitates robust architectures and effective methodologies. Eberlein and Hines (1996) advocated for standardized molecules to improve model quality. Tignor and Myrtveit (2000) distinguished between classes and sub-models in system dynamics, requiring sub-models to be standalone. In their modeling of dryland salinity in Australia, Khan and McLucas (2008) adhered to the V-model approach, placing a strong emphasis on integrating multiple modules into a comprehensive model. These modules were not only designed to be executed independently, but were also intended to have their own user interfaces with control elements. Yeager et al. (2014) presented a new platform for system dynamics modeling that supports detailed and object-oriented modeling. To enhance the integration of diverse system dynamics models and data, an MDaaS architecture (Arto et al. 2014) was proposed, which is characterized by models and data sources acting as services that communicate with each other solely through the exchange of state variables.

To further advance modular modeling in the field of system dynamics, this paper introduces a method for model decomposition and composition that can be implemented within the Vensim® environment, building upon the previously mentioned MDaaS architecture. The purpose of developing this method is to lay the foundation for establishing a system dynamics module library.

We will first present the proposed procedure for modularization and integration. This procedure will then be demonstrated through two examples, followed by a concluding discussion and summary.

2. Shared State Variables and Subdivided Time Steps for Model Decomposition and Integration

The essential requirement for a modularization concept is that: (1) it allows one module to be written with little knowledge of the code in another module, and (2) it allows modules to be reassembled and replaced without reassembling the entire system (DeRemer & Kron 1976). Gamma et al. (1994) emphasized the importance of finding appropriate objects, determining object granularity, and specifying object interfaces as the primary concerns in software design. Meyer (1997) proposed five fundamental requirements for modularization: decomposability, composability, understandability, continuity, and protection.

System dynamics modules can be developed through a mixed approach of decomposing existing models and building individual modules from scratch. The modules thus generated can be combined into a final model. We have the following requirements for each module:

1. It must have a clearly defined functional scope and accordingly include one or more core stocks.
2. The number of nodes should not exceed a certain limit, such as 70 nodes, and should be easily displayed within a single view of the user interface of the model development and simulation environment, such as Vensim®.
3. It should be capable of independent simulation within the model development and simulation environment. This means that the module's input parameters should only include stocks and constants, while output parameters can include several other variables in addition to these.

The third point is particularly crucial from a practical modeling and simulation perspective. Modules that meet this criterion can be independently created, tested and optimized. Several such modules can be integrated into a single model using the following steps:

1. Copy and paste all modules into a single Vensim file, with each module on a separate view.
2. Unify the names of identical stocks and constants across all modules.

Sometimes, more than one module may change the same stock. In this case, a technique of subdivided time steps can be employed:

1. Subdivide each time step of the entire model's simulation calculation into several smaller steps, for example, two, four or eight sub-steps.
2. In each sub-step, any stock accepts changes from at most one module.
3. The entire model's simulation calculation is completed by performing all time steps in the designed sequence.

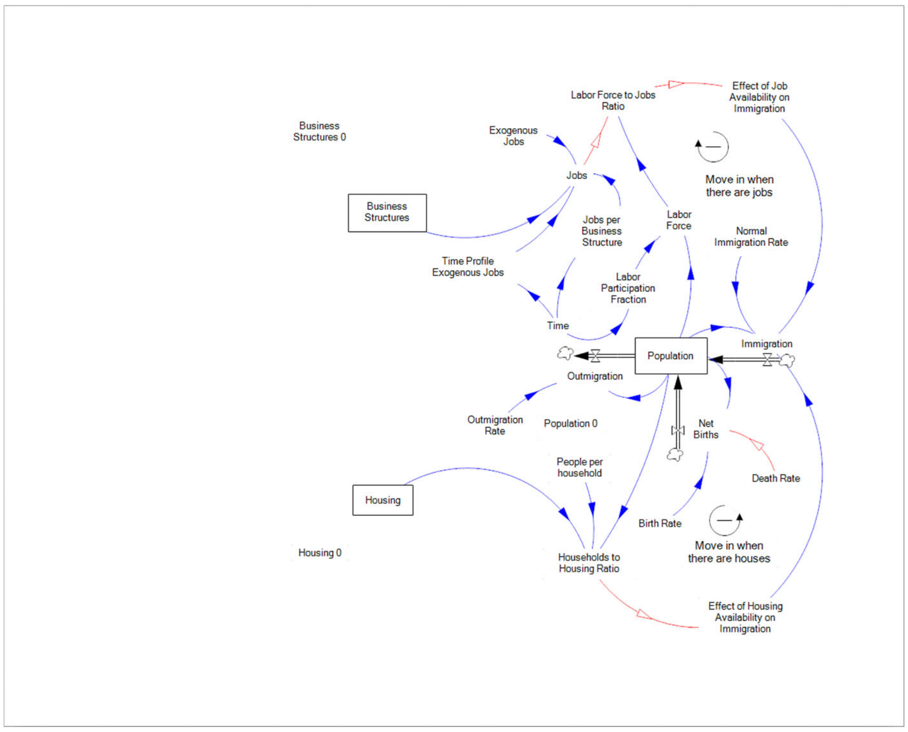
Decomposing an existing model into several modules is, to some extent, a process of re-modeling. The following method can generally be used:

1. Based on the functions assigned to the module, determine one or several core stocks of the module, i.e., the stocks that the module's operation will change.
2. Delete all variables and constants that will not cause changes in these core stocks.
3. All remaining variables can be concentrated into one page and stored separately as a module.

3. Examples of stock and flow model decomposition and integration

3.1 URBAN1 and an extension

In this section, we will present an example of modularization and reintegration of the URBAN1 model (Ghaffarzadegan et al. 2011). We first decomposed the original model into three independently operable sub-models using Vensim, titled "Population", "Business Structures" and "Housing," respectively. We then combined them into an integrated model. After confirming that the integrated model functions consistently with the pre-split model, we will also use the three sub-models to replace the "Businesses" model in the town development model established in (Qian et al. 2018) to achieve the purpose of model expansion.



Labor Force to Jobs Ratio
Effect of Job Availability on Immigration
Business Structures 0
Exogenous Jobs
Move in when there are jobs
Jobs
Business Structures
Jobs per Business Structure
Labor Force
Normal Immigration Rate
Time Profile Exogenous Jobs
Labor Participation Fraction
Time
Immigration
Population
Outmigration
Outmigration Rate
Population 0
Net Births
People per household
Death Rate
Housing
Birth Rate
Move in when there are houses
Housing 0
Households to Housing Ratio
Effect of Housing Availability on Immigration

Figure 1: Module Population of URBAN1

The first module revolves around the stock of `population`. The Figure 1 illustrates the sub-model that describes population dynamics based on the birth rate, mortality rate, and migration processes. Population growth is particularly influenced by other factors such as housing and job opportunities. The sub-model retains the two stock variables representing `housing` and `business structures`, while removing all components from the original model that only change them.

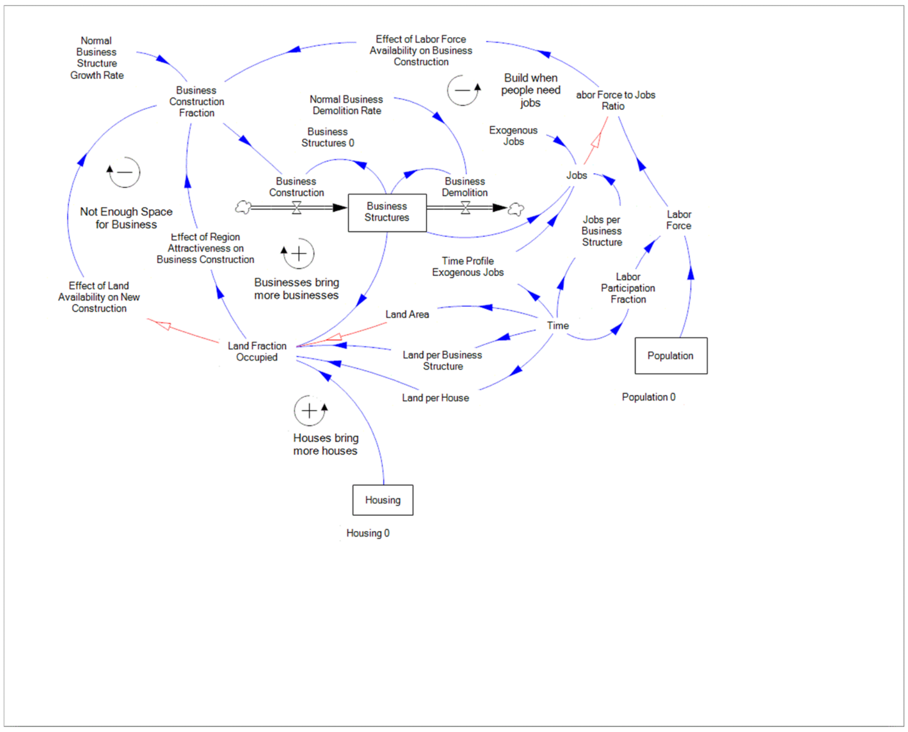

Normal Business Structure Growth Rate
Effect of Labor Force Availability on Business Construction
Build when people need jobs
abor Force to Jobs Ratio
Business Construction Fraction
Normal Business Demolition Rate
Business Structures 0
Exogenous Jobs
Jobs
Business Construction
Business Structures
Business Demolition
Not Enough Space for Business
Effect of Region Attractiveness on Business Construction
Jobs per Business Structure
Labor Force
Time Profile Exogenous Jobs
Businesses bring more businesses
Labor Participation Fraction
Effect of Land Availability on New Construction
Land Area
Time
Land Fraction Occupied
Land per Business Structure
Population
Land per House
Population 0
Houses bring more houses
Housing
Housing 0

Figure 2: Module Business Structures of URBAN1

The second module is centered around the stock of `business structures`. Figure 2 shows the sub-model that describes the mechanisms of business expansion, job creation, and land utilization. Business growth is also influenced by other conditions such as labor supply and land resources. The sub-model retains the two stocks of `housing` and `population` and their relationship with `business structures`, while removing all components from the original model that only change `housing` and `population`.

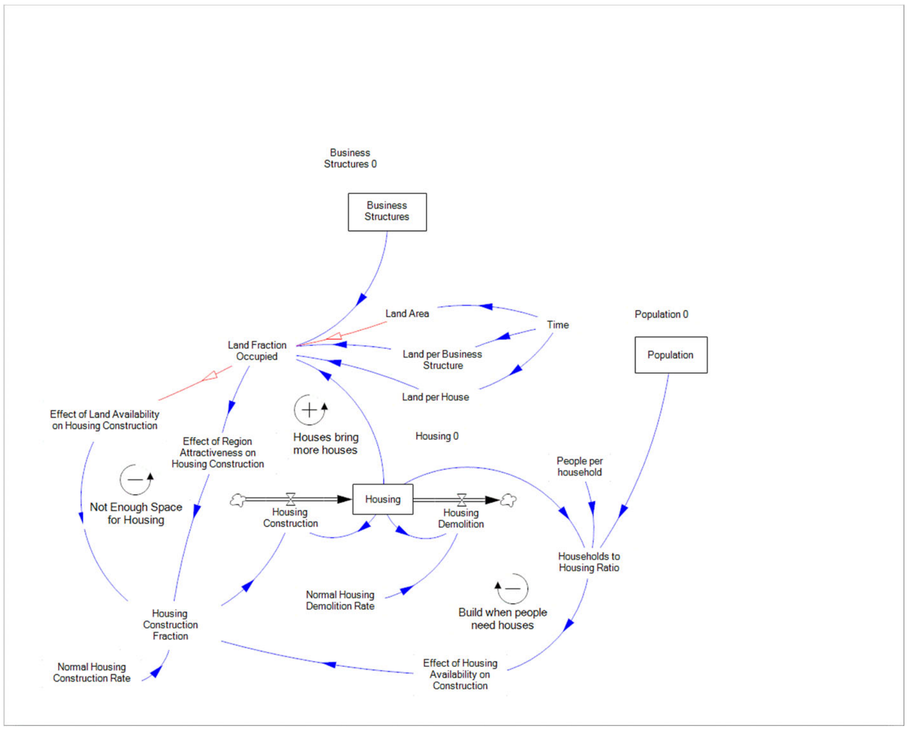

Business Structures 0
Business Structures
Land Area
Time
Population 0
Population
Land Fraction Occupied
Land per Business Structure
Land per House
Effect of Land Availability on Housing Construction
Effect of Region Attractiveness on Housing Construction
Houses bring more houses
Housing 0
People per household
Not Enough Space for Housing
Housing Construction
Housing
Housing Demolition
Households to Housing Ratio
Normal Housing Demolition Rate
Build when people need houses
Housing Construction Fraction
Normal Housing Construction Rate
Effect of Housing Availability on Construction

Figure 3: Module Housing of URBAN1

The third module encompasses the `housing` subsystem. The Figure 3 illustrates the processes of housing construction, demolition, and availability in response to demographic changes and land constraints. Housing development is influenced by factors such as land availability, regional construction dynamics, and the relationship between housing needs and available space. As the population grows, the demand for housing rises, altering household distribution and shaping further expansion. The sub-model retains the two stocks of `population` and `business structures` and their relationship to housing development, while removing all components from the original model that only change population and company numbers.

Each of the resulting three modules can be run and debugged independently. When integrating these three modules into a single model, simply copy the three modules into one model file and unify the names of the three stocks in each module using, for example, the Merge tool in Vensim. For convenience, the same tool can also be used to unify the names of all external parameters in each module.

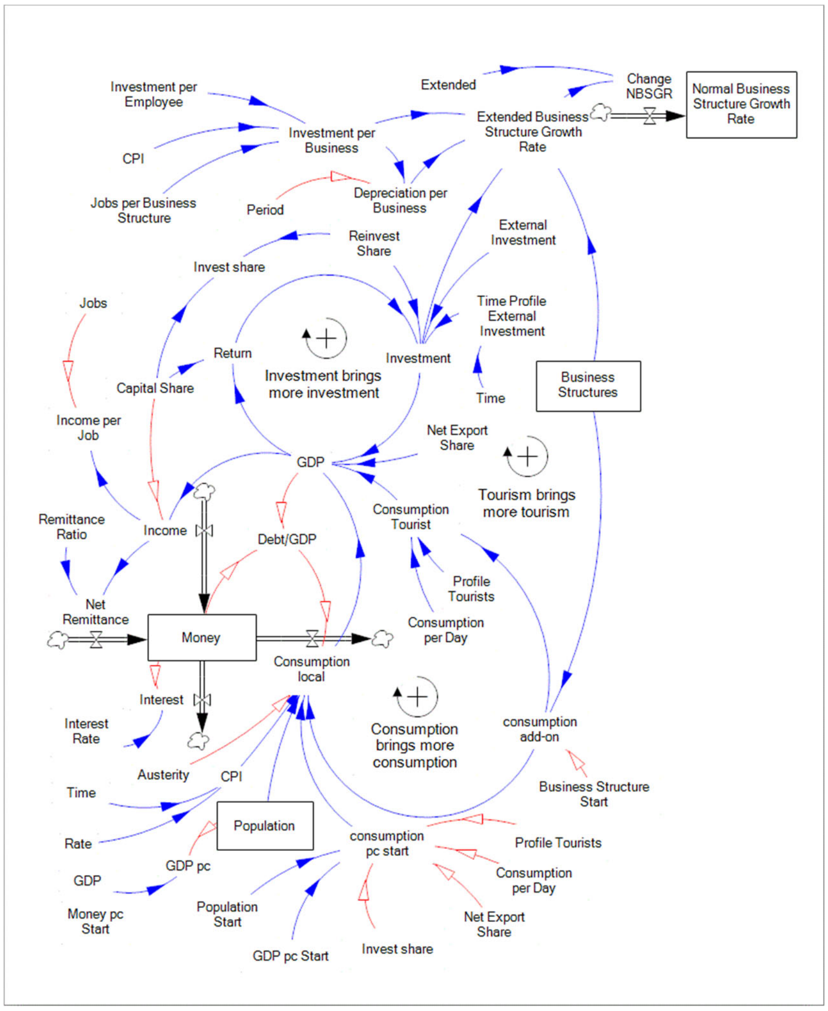


Figure 4: Expanding the URBAN1 model through the inclusion of investment and consumption dynamics

This method can also be used to expand the model. The module shown in Figure 4 can be run and debugged independently. Based on the two state variables of `Population` and `Business Structures` provided by URBAN1, it introduces factors such as investment, local resident consumption, and tourist consumption to simulate and analyze their impact on the `Normal Business Structure Growth Rate`. When integrating this module with the three modules mentioned earlier, this variable needs to be unified as a state variable.

3.2 Power2045

In this section, we present an additional example of modularization and integration. The original Power2045 model (Hu et al. 2024) has been divided into four modules, each representing a subsystem for renewable energy supply:

- **WDPV**: electricity consumption and production from wind, solar, and other renewable sources;
- consumer-side **load shifting**;
- pumped **storage** hydropower;
- **dispatchable** thermal power plants and plants for synthetic fuel production.

As shown in Figure 5, three of the modules consider both the technical and economic aspects of the subsystems. A model assembled from these three modules enables a detailed hourly simulation of electricity supply over a year. The WDPV, Storage, and Dispatchable modules are executed sequentially in each of the 8760 time steps, reflecting a hierarchy of energy utilization. Renewable energy is prioritized, followed by mechanical energy storage, and finally, conversion to chemical energy or dispatchable generation. To maintain consistency throughout the simulation, state variables are shared globally across all modules, while other variables are kept local within each module.

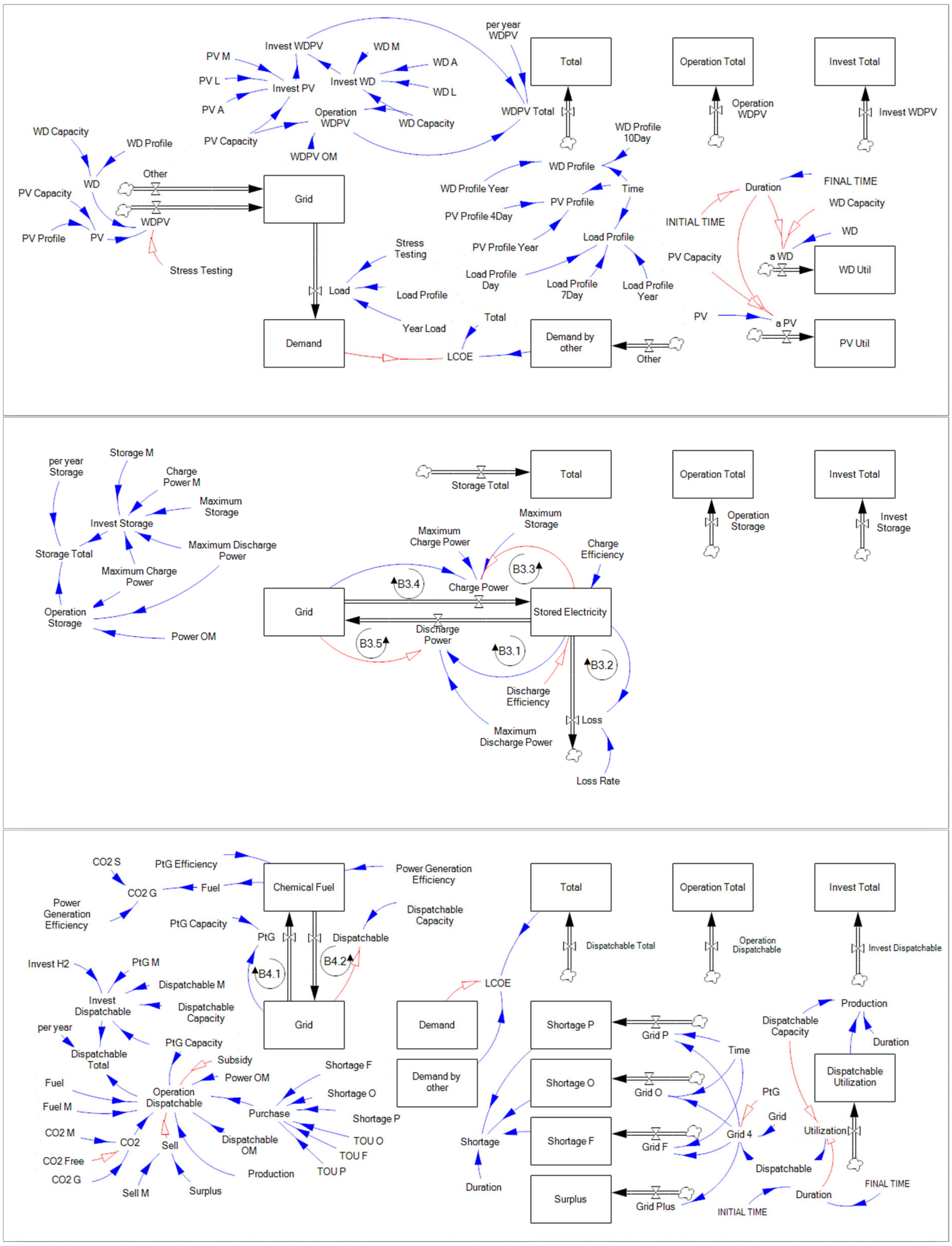


Figure 5: Three modules of the Power2045 model (from top to bottom): WDPV, electricity storage, and dispatchable generation

All three modules include a stock `Grid` that balances electricity generation and consumption. The mechanisms embedded in the storage and dispatchable modules ensure a largely balanced grid, potentially including the purchase of electricity from external sources.

A majority of the variables within these three modules are dedicated to economic analysis. Each subsystem's investment, operational costs, and annualized total costs are computed. By summing up the relevant stocks and dividing by the total annual electricity consumption, the Levelized Cost of Energy (`LCOE`) is calculated. Notably, the electricity generation and costs of other renewable energy sources are not included in these calculations.

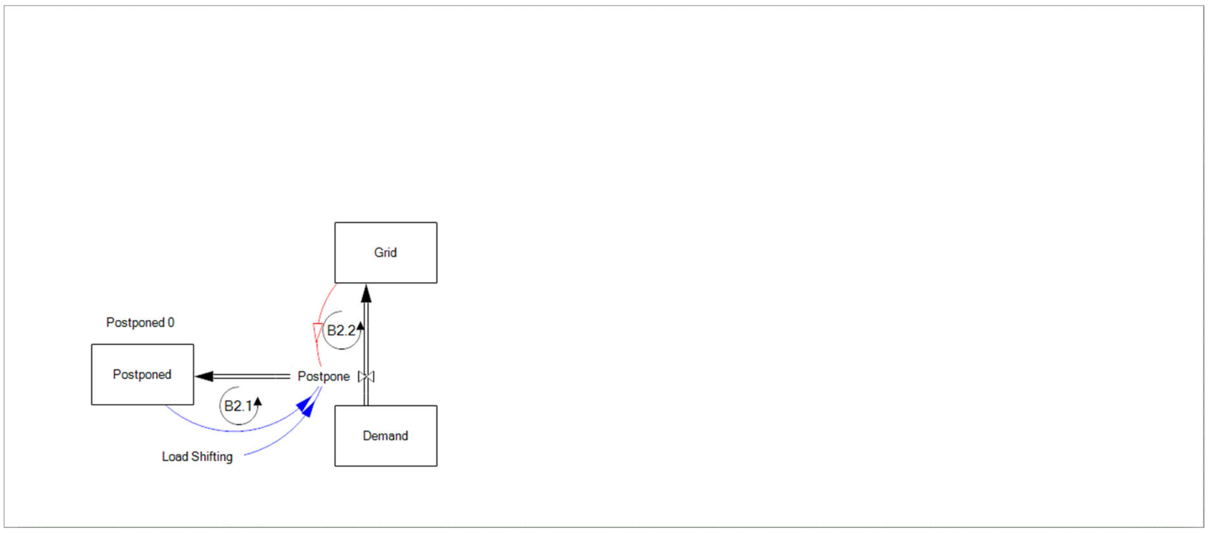


Figure 6: An addition module of the Power2045 model: load shifting

One of the four modules, consumer-side load shifting, only considers power-engineering aspects and does not include economic calculations (Figure 6). This module can be easily integrated with the previous three modules to form a new model. In each time step, the modules WDPV, load shifting, storage, and dispatchable generation are executed sequentially, with shared state variables ensuring a continuous simulation. This modeling reflects the priority of first utilizing renewable energies, then optimizing through load shifting, and finally resorting to storage and conventional generation to ensure grid balance.

Table 1 lists the five stocks that are changed by more than one module among the four modules to be combined. When combining these four modules, in addition to unifying the names of these five stocks, the aforementioned technique of subdivided time steps is also employed. The specific steps are as follows:

1. Subdivide the model simulation time from 1 hour into four 0.25-hour time steps.
2. Introduce four variables, `MS1`, `MS2`, `MS3`, and `MS4`. These four variables take a value of 4 during their respective subdivided time steps and 0 during other subdivided time steps. For example:
```
MS2 = IF THEN ELSE( MODULO(Time * 1/TIME STEP, 1/TIME STEP ) >= 1
        :AND: MODULO(Time * 1/TIME STEP, 1/TIME STEP ) < 2, 1/TIME STEP
, 0 )
```
3. Multiply the changes in all stocks due to inflow and outflow by the corresponding MSx factor, such as:
```
Grid = MS1 * (-Load+Other+WDPV - Grid) +
        MS2 * Postpone +
        MS3 * (Discharge Power - Charge Power) +
        MS4 * (Dispatchable - PtG)
```

Table 1: State variables changed by more than one module

Module	Grid	Demand	Total	Operation total	Invest total
WDPV	x	x	x	x	x
Load shifting	x	x			
Storage	x		x	x	x
Dispatchable	x		x	x	x

The simulation results of the combined model match those of the original model.

4. Concluding discussion

This report introduced a method to decompose existing stock-and-flow models into several Vensim .mdl files. These files, along with other directly created .mdl files, can be independently simulated and calculated, demonstrating specific functions. Their complexity is significantly lower than the entire model, making them easier to understand and optimize. With the aid of the Vensim environment, these .mdl files can be combined into a single .mdl file that possesses the functionality of the entire model. The feasibility of this method has been preliminarily demonstrated using two published models as examples.

The purpose of developing this method is to lay the foundation for establishing a stock and flow module library. The main function of such a module library is to provide modules, aiming to simplify and accelerate the construction of system dynamics applications through integration.

However, a major drawback of the method introduced in this report is its low degree of automation. Currently, the process of splitting existing models into modules and recombining them is entirely manual, which increases the likelihood of errors, and the number of existing modules is also relatively low. To establish a module library of a certain scale and use it to build entirely new models, rather than just recomposing existing ones, more software support is needed to achieve at least semi-automation of model decomposition and composition. During the module integration process, attention should also be paid to feedback loops that span more than one module, as these feedback loops usually have a greater impact on the overall system behavior than feedback loops within a single module.

In a development environment with increasing economic and social uncertainty, the demand for complex system modeling is surging. Rapidly building scalable and verifiable decision support models has become a core challenge. The modular decomposition method proposed in this paper will demonstrate broader application prospects in the field of System Dynamics.

By further standardizing module interfaces and functional encapsulation, a methodological framework can be provided for building a "model factory". In the future, with the aid of AI-assisted automated decomposition tools, existing model libraries can be expanded into a resource pool containing thousands of standardized modules, and complex systems can be rapidly assembled through intelligent combination algorithms. This modular ecosystem will not only reduce model development costs but also enable dynamic module replacement to meet decision-making needs in different scenarios. For example, in areas such as climate change, ecological environment, and economic development, customized models adapted to specific policies or perturbations can be rapidly generated based on the module library. Furthermore, combined with real-time data integration and cloud platform technology, the modular model factory can also support distributed collaborative modeling, driving a paradigm shift in interdisciplinary and inter-organizational complex system research. With the improvement of semi-automated toolchains, this technology is expected to become the infrastructure for nextgeneration System Dynamics application development, significantly enhancing humanity's modeling capabilities to address global complex challenges.

References

Arto et al. 2014 Klaus Arto, Bo Hu, Armin Leopold, Silja Meyer-Nieberg, Goran Mihelcic, Jan Stutzki, Tim Tepel: Modellbasierende Früherkennung politischer Krisen: Prototyp einer serviceorientierten Plattform. *Multikonferenz der Wirtschaftsinformatik, 278-293,* Paderborn, 26.-28.02.2014

DeRemer & Kron 1976 DeRemer, Frank, and Hans H. Kron: Programming-in-the-large versus programming-in-the-small. *IEEE Transactions on Software Engineering, **2**: 80-86,* 1976

Djanatliev et al. 2012 Djanatliev, A., German, R., Kolominsky-Rabas, P., & Hofmann, B. M.: Hybrid simulation with loosely coupled system dynamics and agent-based models for prospective health technology assessments. *Proceedings of the 2012 winter simulation conference (WSC), 1-12,* IEEE, 2012, December

Elmasry & Größler 2018 Elmasry, Aly and Größler, Andreas: Supply chain modularity in system dynamics. *System Dynamics Review, **34**: 462-476,* 04 December 201

Gamma et al. 1994 Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns : elements of reusable object-oriented software. ISBN 0-201-63361-2, Addison-Wesley, 1994

Ghaffarzadegan et al. 2011 Ghaffarzadegan, Navid, John Lyneis, and George P. Richardson: How small system dynamics models can help the public policy process. *System Dynamics Review, **27.1**: 22-44,* 2011

Hines & Eberlein 1996 Hines, J., & Eberlein, R.: Molecules for modelers. *Proceedings of the International System Dynamics Society,* System Dynamics Society, Cambridge, 1996

Hu et al. 2024 Bo Hu, Bo Zhang, Yonghua Li, Junshen Zhang: Towards carbon neutrality: Optimizing generation and storage capacities in Germany and carbon pricing in China. *Sustainable Production and Consumption, **46**: 703-716,* Elsevier, May 2024

Karl 2016 Karl, Christian K.: System Dynamics Libraries-An approach to develop modular-oriented simulation models. *2016 International System Dynamics Conference,* 2016

Khan & McLucas 2008 Khan, Naeem U., and Alan C. McLucas: A Case Study in Application of Vee Model of Systems Engineering to System Dynamics Modelling of Dryland Salinity in Australia. *School of Information Technology and Electrical Engineering,* 2008

Meyer 1997 Meyer, Bertrand: Object-oriented software construction. Second Edition, Prentice Hall, Englewood Cliffs, 1997

Qian et al. 2018 Ying Qian, Fang Yu, Bo Hu: A System Dynamics Model for Developing Small Towns with Characteristic Features. *The 2018 International Conference of the System Dynamics Society, Proceedings,* Reykjavík, Iceland, August 6-10, 2018

Tignor & Myrtveit 2000 Tignor, Warren, and Magne Myrtveit: Object Oriented Design Patterns and System Dynamics Components. *Proceedings of the International System Dynamics Society,* 2000

Yeager et al. 2014 Yeager, Larry, Thomas Fiddaman, and David Peterson: Entity-based system dynamics. *Proceedings of the international system dynamics conference,* Delft, 2014