Building Digital Twins on Existing Infrastructure: Evaluating The Effectiveness of IDAES for Live Data Processing

Bert Downs
Ahuora Research Group
University of Waikato
Hamilton, New Zealand
bd65@students.waikato.ac.nz

October 2024

Abstract

Digital Twin technology promises improvements in factory performance by optimising based on a digital model that updates to follow physical conditions. This research evaluates the IDAES Process Simulation Environment for use in developing Digital Twins, to show that it can provide for many of the core requirements of a Digital Twin. A theoretical framework is presented to generalise the concept of a Digital Twin. The framework shows that a Digital Twin can be viewed as a software system built on top of an existing simulation platform and live data processing system.

Keywords: Digital Twin; IDAES-PSE; Process Simulation; Hybrid Modelling

List of Abbreviations

IDAES-PSE	Institute for the Design of Advanced Energy Systems - Process Simulation Environment
SCADA	Supervisory Data Aquisition and Control
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
RBF	Radial Basis Function
PYSMO	Python-based Surrogate Modelling Objects
OMLT	Optimization and Machine Learning Toolkit

1 Introduction

Traditional factory control systems are based on feedback loops that use sensor data to adjust the factory's state. However, these systems are limited in their ability to predict future states and optimise factory performance. Digital Twinning is a new approach that combines live factory data, historical state, mathematical modelling, and data-driven modelling to create a digital replica of the factory. Yet in existing literature, most examples of digital twinning focus on specific situations rather than providing a general framework for creating digital twins. Many remain at the proof-of-concept stage, and have not yet been deployed in industrial settings.

In order to implement digital twinning in industry, simulation platforms must be able to work with and process live data, but most simulation environments are designed for design and analysis, rather than real-time processing [Agi et al., 2024]. The objective of this research is to evaluate how effectively an existing simulation platform can be used or extended to process live data.

One such simulation platform is the IDAES-PSE modelling framework, which is designed for the analysis of chemical processes [Lee et al., 2021]. It supports a variety of modelling techniques, including steady-state chemical modelling, and dynamic modelling of time-dependent processes. It has also been extended to support data-driven modelling techniques, using libraries such as PySMO and OMLT [Ceccon et al., 2022]

This research evaluates the extent to which the IDAES-PSE modelling framework can be used to implement emerging chemical modelling techniques for live data processing, to assess its suitability as a platform on which to develop such tools. The results of this research are used to propose a general framework for implementing digital twinning in industry.

2 Theoretical Framework

Conventional simulation platforms do not have built-in support for live data processing. Likewise, conventional factory SCADA[1] systems do not have built-in support for complex simulation. To integrate a simulation platform with live data, a software system must be created that can merge the two systems. This can be considered as an intermediate layer between the simulation platform and the factory SCADA system.

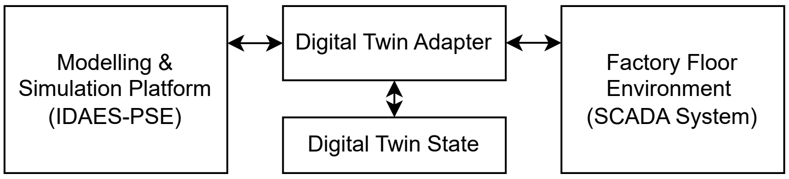
Modelling & Simulation Platform (IDAES-PSE)
Digital Twin Adapter
Digital Twin State
Factory Floor Environment (SCADA System)

Figure 1: Theoretical framework for evaluating IDAES-PSE in a live data processing context.

The concept of a "Digital Twin" refers to a simulation of something in the physical world, which is kept up to date with a physical system using real-time data [Yu et al., 2022]. In Figure 1, this intermediate layer is what converts the simulation platform, and the data from the factory, into a digital twin. It is broken into two parts: a "Digital Twin Adapter" which is responsible for converting the data from the factory into a format that the simulation platform can understand, and to provide results from simulations back the factory. Also specified is a "Digital Twin State", which represents an estimate of the current state of the factory, based on the simulation results and the live data. This represents the ability of a Digital Twin to "learn" how the physical system behaves, and adjust the inputs to the simulation accordingly.

The results of evaluating the IDAES-PSE platform will be considered in the context of this theoretical framework, to determine what the high-level requirements are for a Digital Twin that is built on top of existing simulation and live data processing platforms.

2.1 Limitations of Live Data Processing

The Digital Twin intermediate layer will be working with live data, which adds additional engineering constraints [Hoi et al., 2021]. There is no access to any future data, so the system must only use data that is available at the time of prediction. Live data can also be noisy, delayed, or missing, so the

[1]SCADA systems: Supervisory Data Aquisition and Control systems.

system must be robust enough to handle these situations. Additionally, there is a time constraint on processing the data, as the system must be able to respond to changes in the factory in real-time. Practically, this means that there is only a window of recent data that can be used at any point in time, unless aggregation or smoothing techniques are used.

3 Method

IDAES has many components, so an iterative validation process is used. The evaluation is broken down into parts, each focusing on a different piece of functionality.

Functionality	**Description**
Steady State Modelling	Analysis of systems in a stable equilibrium where variables do not change over time.
Dynamics	Modelling of time-dependent processes to understand how systems evolve over time.
Optimisation/Control	Techniques to find the best operating conditions or control strategies for a system.
Hybrid Modelling	Combining different modelling approaches, such as data-driven and first-principles models, to improve accuracy and robustness.

Table 1: Different pieces of functionality evaluated from the IDAES-PSE framework.

For each of piece of functionality listed in Table 1, a simple prototype flowsheet is developed to evaluate the functionality of IDAES-PSE in a live data processing context. This flowsheet consists of a Heater, that is heating steam from an inlet temperature of approximately 600 K at atmospheric pressure. This approach allows to focus on the functionality of the IDAES-PSE framework, rather than the complexity of the flowsheet, as the techniques used generalise to larger and more complex systems.

3.1 Evaluation Criteria

Each functionality is tested using by developing and evaluating a simple prototype in the IDAES-PSE framework.

The prototypes are used to evaluate the functionality of IDAES-PSE against three broad categories of use in a factory floor environment: As a standalone tool, when connected with process data, and when connected to control systems.

Category	**Evaluation Criteria**
As standalone tools	How well does IDAES-PSE perform the functionality on its own?
With process data	Are the tools useful to better process data from a factory? What additions or modifications are needed to make them useful?
For control systems	Can the tools be used to improve control systems in a factory? What is required to do so?

Table 2: Evaluation criteria for IDAES-PSE functionality.

Evaluating the functionality of IDAES-PSE as a standalone tool gives an indication of the core functionality. This acts as a control, to compare what value is added when connected with live data systems. It acts as an indication of what IDAES functionality is useful for conventional process modelling.

Analysing how the functionality can be used with process data and control systems gives an indication of the potential value of simulation software in a factory environment. This is used to systematically explore a range of potential use cases for simulation software. To do this, some sample data is generated, and then different techniques are used to process the sample data, emulating how live data would be used in a factory environment.

The evaluation criteria focus on the ability of the IDAES platform to be used for live data processing, and the ease of integration with other systems. They intentionally do not focus on the accuracy of the IDAES platform in performing chemical simulations, which depends on the exact chemical equations used to model a specific system. Likewise, they do not focus on the user-friendliness of the interfaces, or the performance of the solving process itself. Those characteristics are considered out of scope of this study, and are evaluated elsewhere [Hart et al., 2011] [Myhre, 2022].

4 Steady State Modelling

Property	**Value**
Inlet Flow (mol/s)	100 ,
Inlet Temperature (K)	380 ,
Inlet Vapor Fraction	1 ,
Inlet Pressure (Pa)	101325 ,
Heat Duty (J)	100,000 ,

Table 3: Specified properties of the heater.

A simple heater model is developed in IDAES, with the properties specified in Table 3. The model is then solved to find the outlet temperature.

```python
m.fs.heater = Heater(property_package=m.fs.properties)

m.fs.heater.inlet.flow_mol[0].fix(100)
m.fs.heater.inlet.temperature[0].fix(380)
m.fs.heater.inlet.vapor_frac[0].fix(1)
m.fs.heater.inlet.pressure[0].fix(101325)
m.fs.heater.heat_duty[0].fix(100_000)

solver = pyo.SolverFactory("ipopt")
solver.solve(m)
```

Listing 1: Defining a simple heater model in IDAES

As shown in Listing 1, IDAES-PSE enables a declarative approach to defining models. IDAES is built on top of Pyomo [Bynum et al., 2021], an algebraic modelling language, which allows for the use of Pyomo's solver interfaces to solve the model. IDAES provides the equations for the thermodynamic and chemical properties of materials and unit operations. IDAES provides a time set for simulation, and the user can specify at what time steps the model will be solved. However, without enabling dynamic modelling, these time steps are independent. In Listing 1, the model is solved at only a single steady state.

This model can be easily extended to solve multiple steady-state systems, however, they would be all solved as a single batch by IPOPT. In a live data processing context, this would not be useful, because the model needs to be solved for each new set of sensor data when the data is received.

Because IDAES is built in Python, a model can easily be incorporated into a script that reads new data from a stream, and then solves the model with the new data.

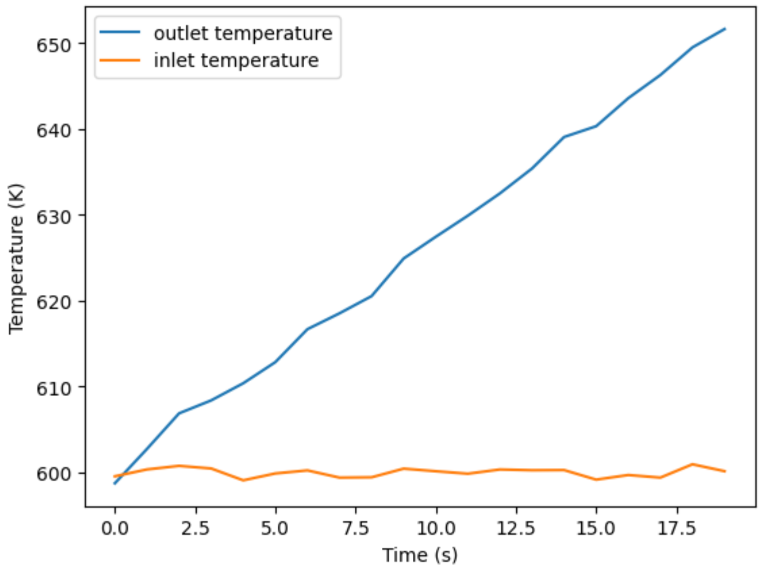

Figure 2: Solving a steady-state model with mock sensor data.

To assess this, a simple script is developed that generates mock sensor data for the heater model. The mock data includes inlet temperatures, which is varied randomly between 599 and 601 K, and inlet pressures, which is set to be approximately 101325 Pa, with a random noise of 5 Pa. A ramp in power from 0W to 20,000W is applied over 20 seconds. The model is then solved with this data, and the outlet temperature is plotted over time. This is shown in Figure 2. At each time step, the model is solved with the most recent sensor data, to provide an estimate of the conditions.

4.1 Evaluation

Steady-state modelling is the core functionality of IDAES. Developing this type of model is well-documented and supported, and solving is robust as long as the model is built with physically feasible conditions. It would be possible to use a steady-state idaes model in a live data processing context, if it was integrated into a process toolchain. Because IDAES has a standardised way of presenting unit models, a data processing framework for multi-steady state can be built in Python that is seperate to the IDAES model itself, but can be linked with any IDAES model.

This would enable modelling live conditions in a factory that are not directly measurable, such as the state of a chemical reaction, or the state of a material in a process. Control systems could use this data to estimate how close the system is to the ideal state, and adjust the system to maintain optimal conditions.

5 Dynamics

To evaluate Dynamic modelling in IDAES-PSE, the heater model is extended to include holdup. Holdup is a property that represents how long it takes for changes to propogate through the system. The heater is modelled over 1 second, with the heat duty instentaneously changing from 0 to 10,000 W half a second through the simulation. As shown in Figure 3, the outlet temperature change is not modelled as an instantaneous change, but instead propogates through the system over time.

This technique could be used in a live system to predict how the system will respond to a change in input conditions. This is particularly valuable in a factory environment, where changes in input conditions can have a significant impact on the system, and may take a long time to propogate through the entire system.

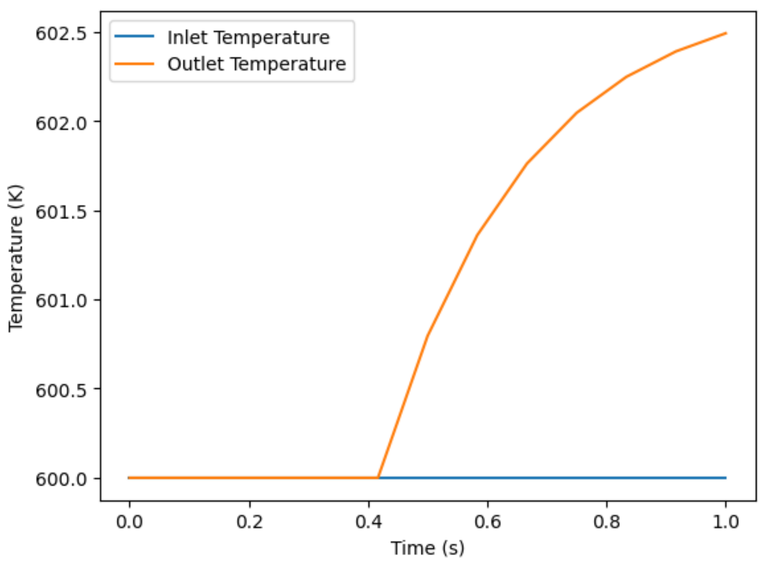

Figure 3: Dynamic response to a step change in heat duty.

To show how a dynamic model could be used in a live data processing context, the same sample data used in the steady state simulation (Section 4) is used in a dynamic model. In steady-state, there is no time dimension, so each timestep can be solved seperately. In dynamic modelling, all timesteps must be solved together. The inlet conditions are specified exactly the same as in the steady state example, but are interpolated between time steps.

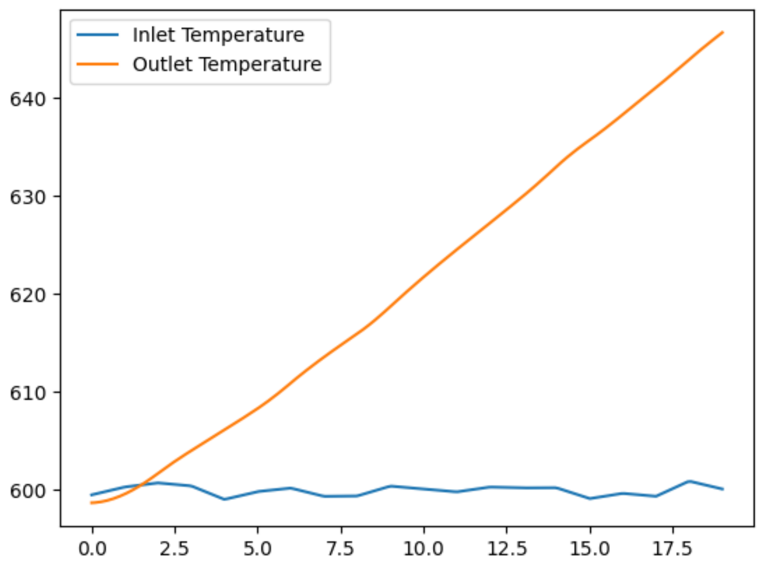

Figure 4: Solving a dynamic model with mock sensor data.

In Figure 4, a smoothing effect is observed in comparison to the steady-state model in Figure 2. This is because it takes time for changes to propogate through, providing a more accurate representation of how a system will respond to change.

5.1 Evaluation

The IDAES framework is well-suited to dynamic modelling, as it provides tools for creating and solving differential equations. It can easily model the same system at different time scales. Dynamic models certainly are much more complex than steady-state models, but IDAES and Pyomo provide the tools to handle this complexity.

Mathematically, dynamic modelling is solving an ODE or PDE system, breaking it into a set of finite elements, and then solving the resulting algebraic equations. This means that each time step cannot be calculated completely independently of the previous time step.

In a live data processing context, this makes is more complex to run dynamic models. However, the problem can still be formulated using only past data, dynamic models can still be used. The final conditions of one time step (variables, and their derivatives) must be used as initial conditions of the next time step. This will contribute to the “State” of the Digital Twin as outlined in Section 2.

Additionally, the model may need to be discretised between time steps to provide a more accurate continuous solution. This will involve interpolating betweeen data points from the factory, or estimating missing values.

Dynamic models can also be used for predicting future states. However, new data will require that all subsequent time steps be recalculated. If the model and initial conditions are specified correctly, from a software development perspective this is no more complex to implement than a steady state model, but it is much more computationally expensive to solve. .

6 Optimisation

Optimising a system involves adding an objective the model using standard Pyomo utilities, and then solving the model to find the optimal conditions. In the heater model, a cost function is added as a test objective, to find the ideal balance between heat duty and outlet temperature. The model is then solved to find the optimal heat duty that minimises the cost function.

```
def cost_objective(h):
return 3**(h.heat_duty[0]/5000) - (h.outlet.temperature[0]-350) * 33000
m.fs.heater.cost_objective = pyo.Objective(rule=cost_objective,
                                            sense=pyo.minimize)
```

Listing 2: Optimising the heater model in IDAES

This is shown in Listing 2. The model must be solved with degrees of freedom, i.e variables that the solver can adjust to find the optimal solution. In this case, the heat duty is the degree of freedom, but there can be multiple degrees of freedom in a model. In Figure 5, a setpoint function for the outlet temperature is added, and optimisation is used to find the heat duty for each timestep that best follows the setpoint. This has one degree of freedom for each discrete time step that must be optimised: the heat duty at that time step. When this technique is used to control a system based on the optimal way to reach a setpoint, it is referred to as Model Predictive Control. Because the setpoint change is instantaneous, it is unable to perfectly follow the setpoint, but is able to minimise the error on either side of the setpoint.

6.1 Evaluation

Because optimisation has first-party support from Pyomo, it is well-supported as a standalone tool, and well integrated with IDAES. It can be used for control to find the optimal setpoints to use. The optimising functionality is generic, so it can also be used for other types of optimisation, such as predicting the most likely conditions from uncertain measurements.

IDAES also supports multi-level modelling, and because it is built on an algebraic modelling language it is possible to include other variables in the optimisation, such current energy costing figures. More advanced models could include these factors, without sigificant rework of the Digital Twin system.

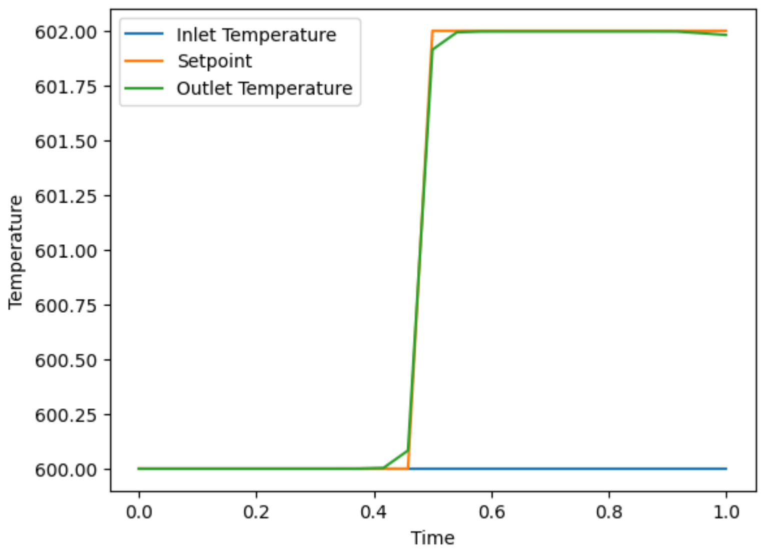

Figure 5: Optimising a dynamic model to follow a setpoint.

7 Hybrid Modelling

To test the workflow for hybrid modelling, a simple surrogate model is created using PySMO. First, a set of data points is generated by solving a heater model at different pressures, temperatures, and flow rates. Then, this data is used to train a surrogate model, which can predict the outlet pressure and enthalpy from the valve based on the inlet conditions. This generated data for a steady-state simulation, predicting the outlet pressures and temperatures. An RBF network is trained to predict these values from the inlet conditions. The weights of the trained model are then saved to disk.

To use the surrogate model in a flowsheet, the weights of the model are loaded and the structure of the model is recreated as a set of algebraic constraints, relating the input conditions to the predicted output conditions. This can be combined with IDAES's UnitModel class and StateBlocks to create a new unit operation that can be used in a flowsheet.

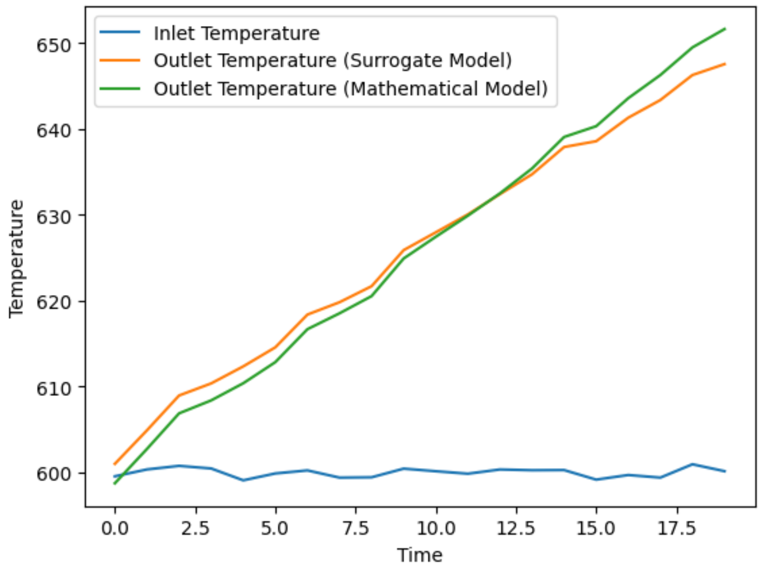

Figure 6: Comparison of the output of the surrogate model and the IDAES mathematical model.

In Figure 6, the output of the surrogate model is compared to the output of the IDAES model. The surrogate model is able to closely follow the IDAES mathematical model, but it does not match exactly. For the purposes of this demonstration, the accuracy is not important; with careful tuning of the surrogate model, it is possible to get a very close match to the IDAES model. However, in a live data processing context, the surrogate model could be retrained using recent process conditions, deviating from the mathematical solution to better reflect real-world nuances in the system.

This demonstration uses a simple RBF network, but other machine learning techniques can also be used, as long as there is support for creating algebraic constraints to represent them in Pyomo. In theory, it is possible to model a dynamic system in the same way that this steady-state system is modelled, but there is less evidence of this being done in IDAES.

7.1 Evaluation

Surrogate modelling may be achieved using IDAES's built-in PySMO libraries, or other similar libraries such as OMLT. It is reasonably straightforward to train a surrogate model to represent a non-dynamic unit operation, but dynamic unit operations get significantly more complex - instead of modelling a single value, the surrogate model must be able to model the entire time system. There are some methods of doing this, such as using neural ODEs, Residual Networks, Operator Networks, or convolutional neural networks. There is little research into applying these methods in the field of chemical and process simulation, especially in the context of mathematical modelling such as the IDAES framework.

The exact same process for surrogate modelling can also be used to model unit operations from historical data. This is useful when there is no mathematical model of the unit operation, but there is historical data available. Online Learning techniques could be used to update the surrogate model in real-time, as new data becomes available. This is a key step in turning a simulation into a Digital Twin, as it allows the model to self-adapt to real-world conditions. The current machine learning libraries in IDAES do not support this, so this functionality would need to be added.

8 Results & Discussion

8.1 Results

The morphological matrix in Table 4 shows the discussed ways of where the IDAES-PSE functionality could be used in a factory environment. While IDAES-PSE provides the core simulation backend to support these use cases, the research into each functionality leads to the conclusion that a custom software solution will be required to integrate it with live data systems.

Functionality	As Standalone Tools	With Process Data	For Control Systems
Steady State Modelling	Basic analysis of systems in equilibrium	Model live conditions	Estimate target conditions for control strategies
Dynamics	Simulate how changes propogate across a system	Predict effect of changing input conditions	Develop dynamic control strategies
Optimisation	Find optimal operating conditions	Multi-Scale Global Optimisation	Implementing Model Predictive control
Hybrid Modelling	Combine modelling approaches for better accuracy	Online learning so digital twin matches physical state	Self-Adaptive control systems based on past performance

Table 4: Morphological matrix showing overlaps and use cases of IDAES-PSE functionality.

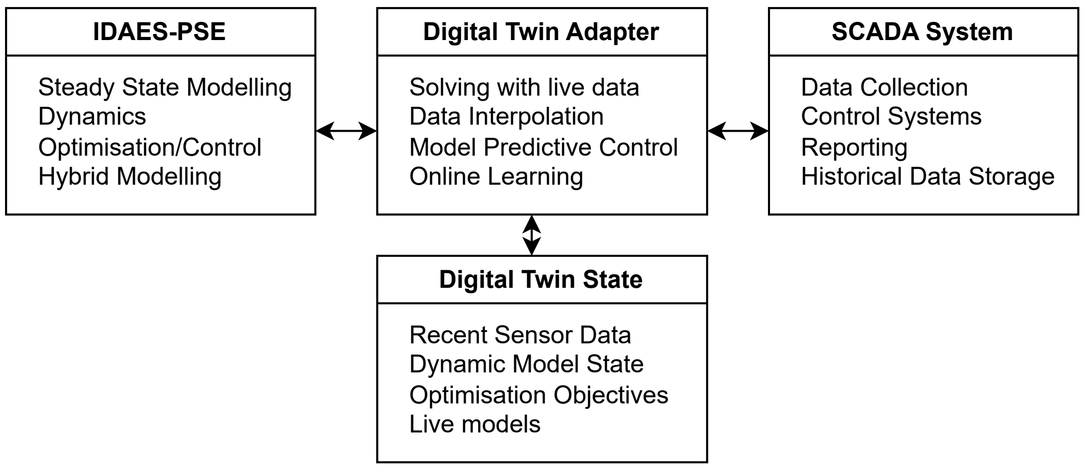

Figure 7: Key Components of a Digital Twin system built on top of IDAES-PSE.

Figure 7 summarises what additional functionality would be required to communicate with a factory system. These can be viewed as the “Digital Twin Adapter” and “Digital Twin State” as outlined in Section 2.

8.2 Discussion

This analysis is unique among the field of Digital Twinning, as it looks at the problem from a software engineering perspective, and does not focus on a custom-built solution.

These results assess the effectiveness of IDAES-PSE as a platform for developing Digital Twins. It outlines how a general solution can be implemented on top of a simulation platform to turn it into a digital twin. The theoretical framework outlined can be used to evaluate other simulation platforms, and factory SCADA systems, to compare them to IDAES-PSE for a broader understanding of the problem space.

In a software context, this provides a set of requirements for such a software system to be developed on top of IDAES-PSE. Development of such a general-purpose digital twin platform for process engineering would be a significant contribution to the field, as it would enable factories of all sizes to take advantage of the benefits of digital twinning.

9 Conclusions

By seperating the simulation platform from the live data processing system, it is possible to create a general-purpose Digital Twin platform that can be used in a variety of factory environments. The IDAES-PSE modelling framework is well-suited to developing Digital Twins for chemical processes. This is because it is supported by an algebraic modelling language, which easily lends it to extensibility and integration with other systems. Additionally, it’s simulation techniques are relevant to live data processing. With appropriately designed software, techniques from IDAES-PSE such as steady-state modelling, dynamic modelling, optimisation, and hybrid modelling can be used to analyse live data from a factory process. A software system that is built on top of this functionality would enable a much wider audience to take advantage of the benefits of digital twinning.

Acknowledgements

This research has been conducted as part of the development of the Ahuora Digital Twin Platform, a key project of the Ahuora Research Group. A sincere thanks to all those who have provided feedback and support during the writing of this paper, particularly many members of the Ahuora Research Group: Tim Walmsley, Mark Apperley, Stephen Burroughs, Ben Lincoln, and many others.

References

[Agi et al., 2024] Agi, D. T., Jones, K. D., Watson, M. J., Lynch, H. G., Dougher, M., Chen, X., Carlozo, M. N., and Dowling, A. W. (2024). Computational toolkits for model-based design and optimization. *Current Opinion in Chemical Engineering*, 43:100994.

[Bynum et al., 2021] Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J.-P., and Woodruff, D. L. (2021). *Pyomo - Optimization Modeling in Python, 3rd Edition*. Springer.

[Ceccon et al., 2022] Ceccon, F., Jalving, J., Haddad, J., Thebelt, A., Tsay, C., Laird, C. D., and Misener, R. (2022). OMLT: Optimization & Machine Learning Toolkit. *Journal of Machine Learning Research*, 23(349):1–8.

[Hart et al., 2011] Hart, W. E., Watson, J.-P., and Woodruff, D. L. (2011). Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3:219–260.

[Hoi et al., 2021] Hoi, S. C., Sahoo, D., Lu, J., and Zhao, P. (2021). Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289.

[Lee et al., 2021] Lee, A., Ghouse, J. H., Eslick, J. C., Laird, C. D., Siirola, J. D., Zamarripa, M. A., Gunter, D., Shinn, J. H., Dowling, A. W., Bhattacharyya, D., et al. (2021). The idaes process modeling framework and model library—flexibility for process simulation and optimization. *Journal of advanced manufacturing and processing*, 3(3):e10095.

[Myhre, 2022] Myhre, M. N. (2022). Investigation of idaes and optimization for a smr process. Master's thesis, NTNU.

[Yu et al., 2022] Yu, W., Patros, P., Young, B., Klinac, E., and Walmsley, T. G. (2022). Energy digital twin technology for industrial energy management: Classification, challenges and future. *Renewable and Sustainable Energy Reviews*, 161:112407.