

Editors: Dr. Mathew Amisshah¹, Dr. Abbas Zaidi¹, Dr. Ali K. Raz¹, Dr. John Shortle¹, Dr. Azad M. Madni², Dr. Dinesh Verma³
¹George Mason University, ²University of Southern California, ³Stevens Institute of Technology

Arlington, VA, USA April 6-9, 2026

Technical Strategies for Semantic Aggregation of Interoperable Digital Twins

John Morris^{a,*}, Duncan W. Gibbons^b, Joe Gregory^c, Gregory Mocko^a

^aDept. of Mechanical Engineering, Clemson University, Clemson, SC, USA

^bDept. Of Industrial Engineering, Stellenbosch University, Stellenbosch, WC, RSA

^cDept. of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, USA

*Corresponding Author. Email: jhmrs@clemson.edu

Abstract

Digital twin aggregates are a synchronized representation of a real system formed by the composition of multiple subsystem models. When formed in an interoperable manner, the integrated models should reveal the emergent behavior of the greater system. The ability to combine independently developed digital twins is a critical aspect of democratized modeling, however, no methods currently enable universal aggregation. In this paper, we propose a two-pronged approach to solving model interoperability that addresses both behavioral and interpretive aspects of describing supersystems. The first prong employs a constraint hypergraph to capture the relationships between system information; we show that these hypergraphs can be autonomously extended given the satisfaction of certain properties. Extended relationships between the joined hypergraphs reveal emergent behaviors, while non-modeled phenomena are contraindicated. The interpretation of the twin aggregate is given descriptively using SysML, which uses external frames of reference to contextualize the system properties. Limitations include having to trade generality in representing the specific aggregated system. Exploring strategies such as these permit models to be connected autonomously even without a prespecified context, allowing modeling experts to focus on specific subdomains of a problem with guaranteed model compatibility.

Keywords: digital twins, digital twin aggregates, interoperability, system of systems, modeling

Nomenclature

A	(As a subscript) denotes the upper, tethered, driven pendulum arm
α	Angular acceleration (rad/s ²)
B	(As a subscript) denotes the lower, freely swinging pendulum arm
g	Gravitational acceleration (m/s ²)
l	Length of the pendulum arm (m)
$l_{unit?}$	Denotes if the lengths of both pendulum arms are 1 meter (Boolean)
θ	Angular position (radians)
ω	Angular velocity (rad/s)
\ddot{x}, \ddot{y}	Horizontal, vertical acceleration of pendulum pivot (m/s ²)

1. Introduction

Though much has been said regarding the need for interoperability with Digital Twins (DTs) [1–4], the ambiguity around a DT's definition has made it difficult to precisely interpret what this means. In this paper, we specify interoperability as akin to system composition. This is based on the definition of a DT as a representation of some real system [5]. We consequently interpret the connection of two or more DTs as

representing the connection between the respective real systems, which in turn is contrived as a single, greater system [6]. For example, a DT of a car and a DT of a road combine to form a single DT of driving system. The ability for a system to be understood after being combined from distinct subsystems has been called interoperability [2], aggregation [7], extensibility [8], compositionality/composability [9], coupling [10], redeployability [3], modularity [11], and plug and play functionality [12, 13]. The authors make use of the first two terms primarily due to their prevalence in the domain of DTs. When discussing DT aggregation, we refer to the individual systems of the component DTs as **subsystems**, while the greater system formed by considering the interaction of all subsystems is termed a **supersystem**. This is preferred to the alternate term of DT ecosystem [14], which is not used here to avoid referring to signals between agentic DTs versus what we see as the more comprehensive view of DTs combining to represent an aggregate system.

The motivation for providing methods for creating DT aggregates is based on the theoretical foundations of systems modeling [2]; namely, that a system is a way of describing some portion of the real world, and that that portion is arbitrarily scoped. Because of this, all models can be assumed to be lacking in comprehensiveness. The predator-prey dynamics captured in the Lotka-Volterra models [15], for instance, capture consumption, reproduction, and decay, but do not account for disease, weather, terrain, social dynamics, etc.; factors that may be relevant to users in different domains. Understanding the behavior of the system according to these alternative factors requires expanding the system scope of the original predator-prey models [16].

We take this notion as support for the claim that a DT is only as useful as the extent of contexts in which it is able to describe system behaviors. In other words, a DT might be correctly prepared, but it is only useful if it can provide correct information for a specific situation. Here contexts and situations are pseudonyms for system scope. Efficacious DTs are ones that can operate meaningfully across a variety of system scopes. Consequently, authors have often described aggregate interoperability as a core requirement of DTs [1, 6, 17]. Despite the variety of frameworks proposed for DTs, DT interoperability remains largely unfulfilled [18–20]. The rest of this paper details methods that address problems with DT interoperability for both data and model aggregation. We specifically show how different behaviors can be combined mathematically as well as semantically by a case study that demonstrates how such aggregated models can be automatically created. Our expectation is that such methods will lead to modular, plug and play DTs that are useful in any context they are employed in.

2. Review

2.1 *Definition of Interoperability*

To demonstrate interoperability, a precise definition of the subject is required. A characteristic interpretation is given as the ability for a group of actors to make use of exchanged information [21–23]. The various ways that information should be exchanged and used results in different types of interoperability, some of which are defined in the ISO/IEC 21823 series of standards [24], such as interoperability between physical affecters or data formats. In this paper, our specific usage of the word interoperability refers to DT aggregation, motivated by the significance of enabling aggregation for DT development. We define aggregate interoperability as the ability to describe the behavior of a supersystem through aggregating subsystem models.

This restricted definition reflects the use cases of DTs. Morris et al. [25] described the purpose of a DT being the provision of the states of one real system through experimentation on another, distinct system. Provisioning state information can be described as interrogation, with DTs enabling indirect interrogation of some specific system. For example, a user might want to know the current level of charge for an electric vehicle. Rather than measuring the vehicle’s battery pack, the user interrogates the car’s DT. The DT can provide the value for the remaining charge either by direct observation (a synchronized datum) or through simulation (a predicted datum). Note that because the user only experiments on the DT, the state of the physical twin is known only through association with the DT. This association is what must be preserved by the DT in order for the DT to be operable. Useful DTs maintain the ability to expose states of the twinned system in a variety of different contexts and scopes.

Further specification should be given on how interrogated states are understood. Following the premise given in [23], a provisioned state must be interpreted both syntactically and semantically. The former gives how a system is described, while the latter concerns the meaning the system’s behavior. For syntactic

aggregate interoperability, labels of the supersystem must be understood by the interrogating agent; for instance, DTs that are interoperable with each other will all identify the same system state referred to by the label “vehicle range.” The semantics of aggregate interoperability, on the other hand, is related to the validity of the value assigned to the state. The values predicted for the vehicle’s range should remain valid in the context of the hilly terrain of the supersystem for the DTs to have semantic interoperability.

2.2 *State of the Art of Digital Twin Interoperability*

The importance of DT interoperability, as given above, has been repeated throughout literature [1, 2, 18, 19, 26]. While many specific aspects of interoperability have been addressed, general aggregation interoperability remains unfulfilled for DTs, though many solutions have been proposed that might eventually change this. High level frameworks, including standards such as ISO 23247 [27], establish interfaces that structure DT construction and enable information exchange, though adhering to such standards often requires Herculean efforts [28]. An alternate approach is to employ a knowledge graph that associates some meaning to the data exchanged between coupled DTs, such as implemented in [4, 29–31]. Knowledge graphs are often paired with ontologies that define semantics for the relationships in the knowledge graph [32, 33]. Other researchers have handled the uncertainty inherent in coupling independent systems together using stochastic frameworks such as Bayesian networks [34, 35] or Large Language Models (LLMs) trained on knowledge graphs or other data representations of the DT [31, 36].

If the models in a DT can be defined beforehand, an alternate approach is to use a federated architecture that manipulates models to form the aggregate system. Many of these [37, 38] follow the high-level architectures defined by IEEE 1516 [39]. High-level architectures rely on signaling at the expense of ignoring the mathematical definitions of system behavior. Other platforms such as ModelingToolkit.jl [40], Decapodes [41], and Modelica [42] employ algorithms that can numerically solve generalized systems. DTs built on these platforms can be immediately combined, allowing full synchronization of the aggregated supersystem. The latter language supports the Functional Mockup Interface (FMI) [43], a widely accepted standard for dynamic simulation. Each of these is typically limited to a certain class of systems that can be expressed to adhere to a specific interface.

2.3 *Review of Modeling Frameworks*

We propose that aggregate interoperability must be provided across two dimensions: semantics and behavior. The first reflects whether a model can describe the system it represents, while the second dimension corresponds to whether those descriptions are accurate to the aggregated system. The definition of a system is always based on external references; in this paper we use SysML to describe the system semantics. The operation of SysML is the scope of this paper, unfamiliar readers are referred to previously published guides such as [44]. The framework used to provide aggregate interoperability of the system behaviors is based on constraint hypergraphs (CHGs), a newer modeling framework formalized in [45] and proposed as a mathematical foundation for DTs in [25]. In a CHG, the states of a system are represented by a set of nodes. System behavior is given by functions that constrain the system states. These functions are represented as directed hyperedges, whose source nodes are the domain of the function and whose target node is the function’s codomain. The motivation for using CHGs is their ability to represent any system, whether process-based, dynamic, discrete/continuous, or multiphysics. This allows a DT’s models to be expressed in a single, cohesive framework. Simulation of a CHG is conducted by forming paths through the hypergraph. DTs built on CHGs can be interrogated declaratively, where system information is retrieved either by observation or an automatically generated simulation [46].

CHGs also provide a simplified platform for understanding model interoperability. A single CHG represents all the states and relationships for some system. Merging that CHG with another provides a larger graph that represents some aggregate system possessing the collective states of each subsystem. Merging CHGs is done by performing a union operation, so that all nodes are singularly represented between the two graphs. Relationships remain distinct between the two graphs, so that any edge pointing to a node in the subgraph continues to point to that node in the supergraph.

3. Methods

The well-defined system at the center of the case study is a simple planar pendulum, with a fixed length l and angular position θ , as shown in Fig. 2A. The most basic description of the pendulum's oscillations is given by Eq. 1, where α is the pendulum's angular acceleration and g is the gravitational acceleration. The structure and configuration of the system are shown in the SysML diagram in Fig. 1.

$$\alpha = \frac{g}{l} \sin \theta \quad (1)$$

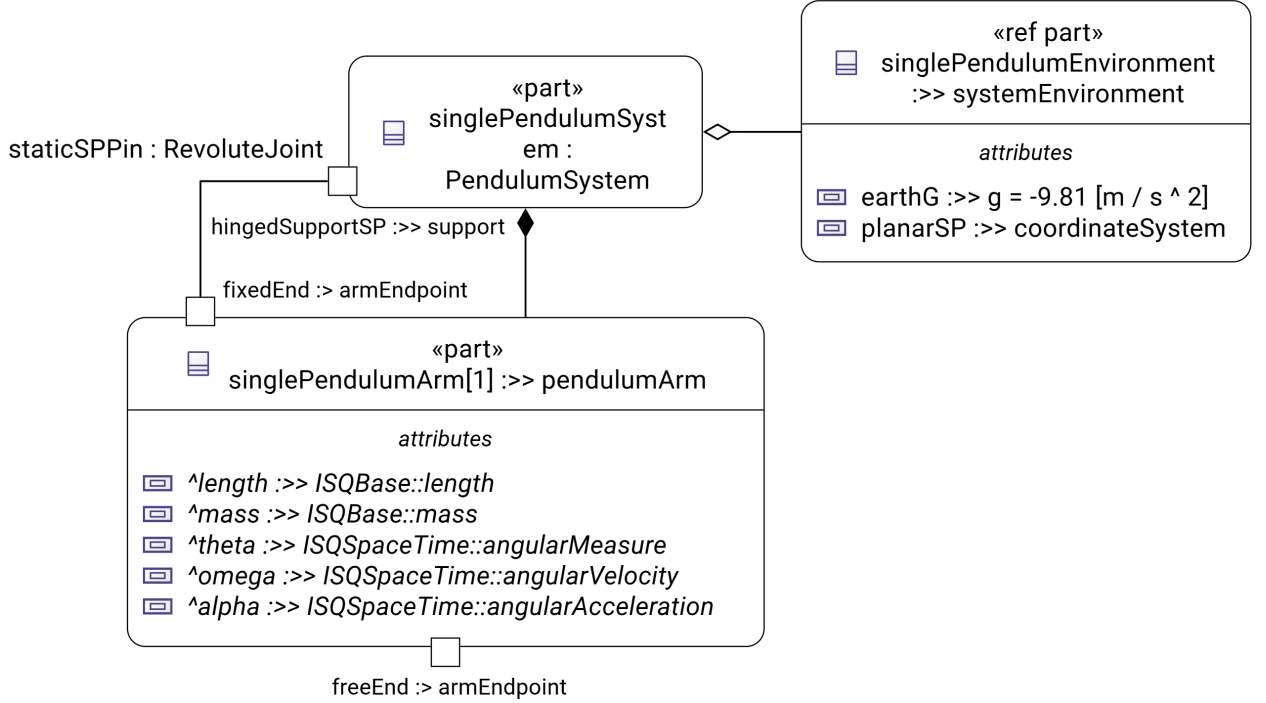


Fig. 1: Descriptive model of simple planar pendulum.

Aggregation is shown by combining two pendulum subsystems into a single supersystem. We show two cases for aggregation. The supersystem in Case I is a trivial aggregation, where the pendulums are considered within the same system scope, but do not otherwise interact, as shown in Fig. 2B. In Case II the two simple pendulums are alternatively combined into a double pendulum, where the pivot of one pendulum is collocated with the free end of the other pendulum as shown in Fig. 2C. In addition to assuming a fixed pivot for the tethered pendulum, to reduce the complexity of the system we constrain the angular velocity (ω) of the tethered pendulum to be a given by a driving function. This has the effect of making interactions one-directional (from the top, driving, arm to the lower swinging pendulum), further isolating the effects of aggregation. The nomenclature will use subscripts X_A and X_B to refer to states for the top (tethered) and bottom (free) pendulum arms respectively, so that θ_A refers to the angular position of the first pendulum.

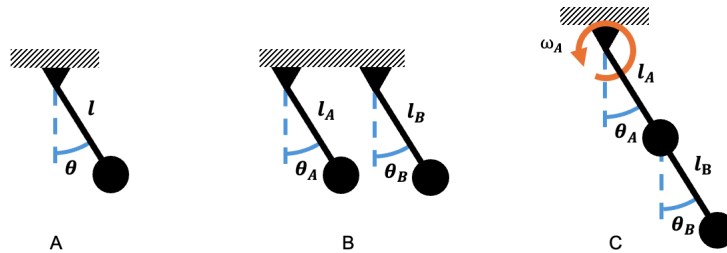


Fig. 2: Overview of pendulum systems used in the case study, from left: (A) the simple pendulum; (B) trivially aggregated

pendulums with no interaction (Case I); and (C) the double driven pendulum (Case II).

The goal with the case study is to examine what strategies can be used by a modeler to provide aggregate interoperability. As described above, aggregate interoperability requires the behavior of a supersystem to be described by the aggregation of its constituent subsystems. Robust interoperability should further occur without manual updating from a modeler. To test this, in each case we present the subsystem models and evaluate the correctness of the aggregated supersystem model. The simulation for each case is of θ_B , capturing the effects of integrating with the driving pendulum. Integration of the equations of motion is performed using basic first-order Eulerian integrators, as in Equations 2 and 3. These integrations rely on a constant timestep Δt . This constant is not visualized in the models (such as in Fig. 3) as it will not vary throughout the cases or be used as a connection point between systems and can consequently be abstracted without affecting system behavior.

$$\omega^{i+1} = \omega^i + \alpha \Delta t \quad (2)$$

$$\theta^{i+1} = \theta^i + \omega \Delta t \quad (3)$$

Note that each of the models in this case study (such as Eq. 1) are only virtual representations, they do not qualify as digital twins. However, the states of each system (expressed as nodes in the corresponding CHG) indicate the potential ports across which the virtual representations can be synchronized with a physical system, forming a digital twin [47]. Consequently, the CHGs can be turned into DTs by passing observed data to the corresponding nodes. We treat synchronization as independent from aggregate interoperability, so that proving interoperability with each model extends to the synchronized DT based upon the model.

4. Results

4.1 Case I: Trivial Aggregation

The trivial case results from a lack of observed interactions between aggregating systems. The angular position for each subsystem, a simple planar pendulum, can be simulated using Equations 1-3. The behavior of these systems does not change whether considering them individually or in the same scope, as in Fig. 2B. The simulation of θ_B can be discovered by tracing a path through the CHG shown in Fig. 3A. Because this process will be repeated in latter cases, it is spelled out more explicitly here. Each simulation in a CHG is represented as a path. These paths are trees, whose leaf nodes are the inputs to the simulation and whose root is the simulation's output. Paths that resolve to a node that is already realized in the path are referred to as cycles. Cycles are solved by unraveling, where the tree is formed by iterating through the cycle repeatedly until some breaking condition is reached. For example, the simulation for the second instance of θ_B is given by the tree in Fig. 3B, where superscripts indicate which iteration of the cycle is being unraveled.

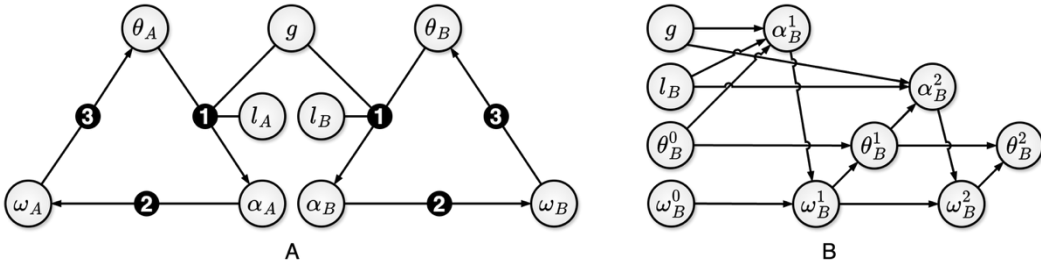


Fig. 3: Models for Case I, from left: (A) the aggregated CHG of the system, where large nodes indicate state variables and numbers on smaller black nodes refer to the specific equation constraining the variables; and (B) the simulation tree for calculating the second value of θ_B by unravelling the cycle in A. The inputs to the simulation are on the left, while the output (θ_B^2) is on the right.

A critical point of aggregation interoperability that is shown in Case I is the implicit specification of the represented system context. For instance, the system modeled in Fig. 3A is an idealization where the pendulum's do not collide or otherwise interact. Any physical system will not perfectly correspond to this idealized model. Adopting this model as a DT requires the assumption that the unmodeled behaviors have

negligible impact on the system; for instance, any DT of Fig. 3A operates under the assumption that there is no resonant interference between the pendulums, that the pivot points are perfectly rigid, and that there is no damping. We point out that these and other assumptions arise from the contextual disagreement between the system of the model and the system represented by the model (the latter of which is usually the physical twin of the DT).

This turns out to be a critical factor in providing aggregating interoperability: the supersystems created through aggregation are incongruent with the systems they are used to represent. Exacerbating the problem is that these incongruencies are by default given implicitly, as it is impossible to explicitly define every dimension along which the modeled system differs from the system of interest. The default result of aggregating the subsystem models (without explicitly specifying the context of that aggregation) is supersystem with minimal interactions. The only shared node between the two models is the gravitational acceleration g . Because no relations are given that solve for g , there are no paths that connect across it, only those that stem from it.

4.2 Case II: Double Driven Pendulum

To discover interaction, a modeler needs a way of specifying the context in which the interactions occur. The context for Case II is a driven double pendulum, in which the two subsystem pendulums should be physically connected rather than merely considered together, as described in the SysML diagram in Fig. 4. The non-trivial part of the double pendulum is capturing the interaction between the two swinging arms. The semantics of this must be established *a priori*, as given by the interface in Fig. 4. However, there is no way to specify the dynamic behavior from the context of the simple models unless more relations are provided.

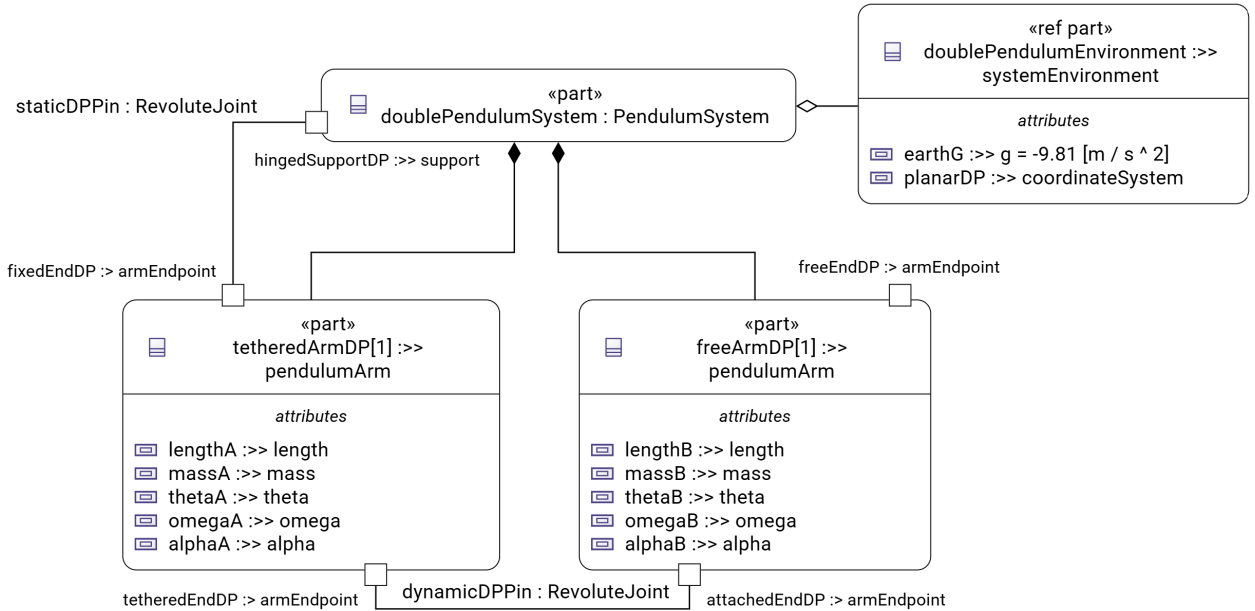


Fig. 4: Descriptive SysML model of aggregated supersystem, describing two pendulum arms and their interface.

In this case, aggregation is performed by the same union operation as in Case I, but with an additional relation provided between θ_A and θ_B . This relation, given in Eq. 4, encodes the constraint between the two pendulum arms. Because this relation would need to be manually added by a modeler at the time of aggregation, these two subsystems do not demonstrate autonomous aggregation interoperability.

$$\alpha_B = -\cos \theta_B (\alpha_A \cos \theta_A - \omega_A^2 \sin \theta_A) - \sin \theta_B (\alpha_A \sin \theta_A + \omega_A^2 \cos \theta_A + g) \quad (4)$$

This constraint competes with Eq. 1 in that both functions solve for α_B . To resolve this, a modeler would need to remove the previous relation and add Eq. 4. These actions do not invalidate the model due to its functional nature, which enables relation modification without side effects [48]. As a result, the CHG for the aggregate system now appears as in Fig. 5B. Note that the assumption of the moving pivot point in Eq. 4

is implicit rather than explicit. This is an imperative connection, which directs how the system should be connected without describing the factors that enable that connection. One consequence of imperative modeling is that the model becomes far less modifiable. If another modeler—who was not involved in the original formulation of the model—wishes to make a change to the model, they will have no recourse to understand what assumptions were made for Eq. 4. One strategy for addressing this limitation is defining new states—given in the CHG as nodes—to the aggregate system. These specify the system's context; in this case that the correct system of a driven double pendulum can be autonomously reconstructed through merger of the subsystem models. We specify the context here across two dimensions by explicitly including an assumption in the model. The two assumptions modeled are that the pendulum tethers are both of equal and unit length, and that the pendulum's pivots are rigid. By deriving states that represent these assumptions, we will show how the corresponding models become increasingly interoperable.

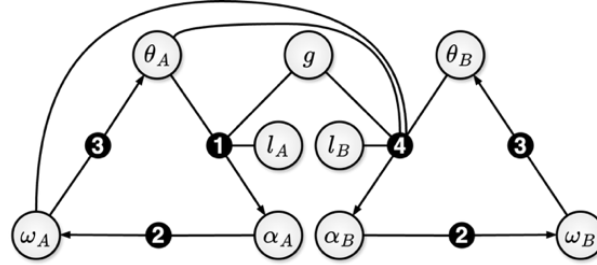


Fig. 5: CHG model of the double driven pendulum for Case I implementing Eq. 4 in place of Eq. 3 for pendulum B , with the shared interaction coming along Eq. 4.

The first new variable introduced is $l_{unit?}$, which acts as a check for pendulum tether length. The variable is *true* if both l_A and l_B are of unit length and *false* otherwise, as given in Eq. 5. This is implemented as a validity frame on Eq. 4, so that the relation is only valid if $l_{unit?}$ is *true*. The mathematics of implementing validity frames onto functional relations is described in [45]. This has the effect of making the edge calculating α_B using Eq. 4 as computable only if the assumption of unit lengths is met—otherwise the edge will be inactive. The CHG from Case II is shown with this additional node in Fig. 7. The inclusion of $l_{unit?}$ allows the Eq. 4 to be incorporated in the model without risking violation of the unit length assumption.

The other implicit assumption preventing context specification is that the pivots of the pendulums are not accelerating, as in the case of the rigidly fixed pivot implied in Eq. 1. To form the context of the double pendulum this assumption is obviously violated, as the pivot for pendulum B must remain collocated with the rotating bob of pendulum A . However, because the assumption is implicit in Eq. 1, it is not possible to express the contradiction, nor the effects of doing so on the rest of the model. To make this model interoperable the assumptions that specify the double driven pendulum need to be made explicit, specifically by adding ports (nodes) along which the assumptions can be characterized.

To accomplish this, we specify the subsystem model for pendulum B to include nodes for a moving pivot. The necessary nodes are \ddot{x}_B and \ddot{y}_B for the horizontal and vertical acceleration respectively. Other candidates for new nodes, such as the position and velocity of the pivot point, are not included for simplicity as they do not affect the pendulum's dynamics. The fully derived equation of motion incorporating \ddot{x}_B and \ddot{y}_B is given in Equations 6-8, and incorporated in the CHG in Fig. 6.

$$l_{unit?} \Leftarrow (l_A = 1 \wedge l_B = 1) \quad (5)$$

$$\alpha_B = -\ddot{x}_B \cos \theta_B - \sin \theta_B (\ddot{y}_B + g) \quad (6)$$

$$\ddot{x}_B = \alpha_A \cos \theta_A - \omega_A^2 \sin \theta_A \quad (7)$$

$$\ddot{y}_B = \alpha_A \sin \theta_A + \omega_A^2 \cos \theta_A \quad (8)$$

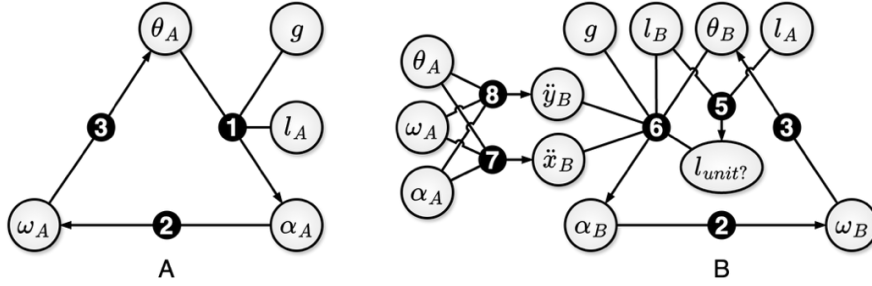


Fig. 6: Subsystem CHGs of the top, tethered pendulum (A) and the bottom, freely hanging pendulum (B).

The two subsystems, so defined, demonstrate full aggregation interoperability, in that once merged they will correctly and automatically reveal the behavior of the driven double pendulum. They also correctly demonstrate the behavior of the subsystems. In the case of the CHG for the lower pendulum in Fig. 6B, this is because inputs for \ddot{x}_B and \ddot{y}_B are known to be zero in the subsystem case, reducing Eq. 6 back to Eq. 1. The systems can similarly be merged to reveal the aggregate system behavior, with the joined CHG given in Fig. 7. Although the CHG in Fig. 7 reveals all modeled interactions in the supersystem, the achievement of aggregation interoperability is less impressive when recognizing that the nodes in the graph of the tethered driven pendulum CHG in Fig. 6A are a subset of the nodes for the graph of the free arm in Fig. 6B—that is, no new nodes were added to Fig. 6B when joining it with Fig. 6A, only new edges. This reinforces that aggregation interoperability is only guaranteed when edges are modified, not when new nodes are introduced.

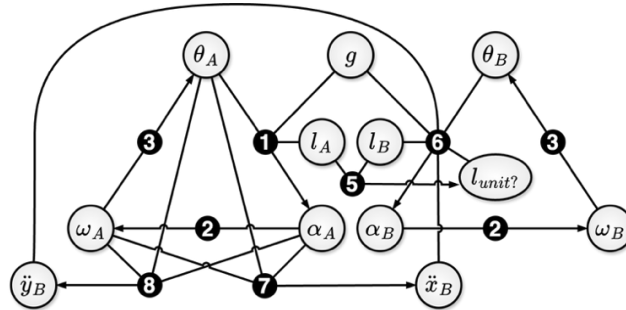


Fig. 7: CHG of double driven pendulum formed by merging the two subsystems in Fig. 6, with acceleration of the lower pendulum's pivot given by \ddot{x}_B and \ddot{y}_B and the assumption of the pendulum tethers' length checked by $l_{unit?}$.

5. Discussion

These exercises in model interoperability demonstrate three observations: 1) systems can be aggregated through the use of a CHG by merging the models along common variables; 2) problems with aggregation interoperability arise from the a lack of specificity in the aggregated supersystem model; and 3) defining the aggregation *a priori* such as with a descriptive model resolves the lack of specificity. Specificity is related to the assumptions made by a modeler. Considering the degree to which assumptions are included in the model reveals an underlying dilemma affecting aggregational interoperability: highly specific models, which explicitly declare the model assumptions, are highly interoperable for singular cases. Omitting the assumptions makes models more general, but at a cost of losing the ability to represent specific supersystems. While both models are still computable, in the generalized case it may not be apparent what is being simulated. In a CHG, adding model specificity (such as assumptions) appears as adding nodes to the graph, while in a descriptive model these are given by providing a limiting frame on the system attributes.

5.1 Future Work

In this example, the CHG has been manually produced by a domain expert based on the description contained in the SysML model. In future work, we will focus on how we might use data interoperability techniques to automate this transformation. Ontologies provide a means of formalizing domain knowledge. The description logic subset of the Web Ontology Language (OWL2DL), for example, extends the Resource Description Framework (RDF) which represents data as directed graphs of subject-predicate-object triples

[49]. As OWL2DL uses only a decidable fragment of first-order logic, automated reasoners can be used to reason across our data and evaluate statements made using the language. This is a crucial capability with respect to interoperability – the models we combine must at least conform to the same worldview, and ontologies enable users to define that worldview explicitly and determine whether our datasets are conformant.

OWL2DL can also be combined with the Semantic Web Rule Language (SWRL) to enable horn-clause-style inference rules. Together, they can be used to formally state that if our system description contains certain patterns, then a corresponding (portion of a) CHG should be generated. This capability is powerful as we are interested in generating appropriate behavioral models from descriptive models as part of this research effort. This has the potential to serve two purposes: 1) it would enable users to generate behavioral models from descriptive models as discussed, and 2) it would enable domain experts to formally capture expert knowledge as ontological axioms and inference rules.

6. Conclusion

The perfect case for aggregation interoperability would be models that could be intelligently composed to represent any supersystem—that models of a car and a road could be combined to describe every type of driving situation. However, such idealistic aggregation implies that the work of modeling is done by the combination of models rather than by an expert modeler. In this paper we have instead shown how models can be extended to aggregated supersystems using two different modeling paradigms: a descriptive model given in SysML and a behavioral model given as a CHG. The first allows the parameters of a system to be concisely described in a way that extends to the supersystem, while the second captures the effect that local interactions have on the global system. Using the case of the double pendulum, we have demonstrated how we can combine two DTs to produce a DT of the resulting supersystem and thus predict the behavior of that supersystem. The description of the double pendulum was given by adding externally derived relationships between the pendulum ports, while the behavior of the supersystem was given by functional chains that extended between the two CHG models along common state variables. The ability for CHGs to be merged allows system behavior to be expressed automatically without violating the model's semantics. However, we show that this interoperability is limited by conditions of specificity, such that the merged CHG must be constrained to only represent the specific supersystem intended by the aggregation. Applying these constraints increases the aggregation interoperability at the expense of the model's generality.

References

- [1] V. Piroumian, "Digital Twins: Universal Interoperability for the Digital Age," *Computer*, vol. 54, no. 1, pp. 61–69, Jan. 2021, doi: 10.1109/MC.2020.3032148.
- [2] A. Budiardjo and D. Migliori, "Digital Twin System Interoperability Framework," Digital Twin Consortium, Boston, MA, Dec. 2021. Accessed: Dec. 07, 2021. [Online]. Available: <https://www.digitaltwinconsortium.org/wp-content/uploads/sites/3/2022/06/Digital-Twin-System-Interoperability-Framework-12072021.pdf>
- [3] V. Zambrano, J. Mueller-Roemer, M. Sandberg, *et al.*, "Industrial digitalization in the industry 4.0 era: Classification, reuse and authoring of digital models on Digital Twin platforms," *Array*, vol. 14, p. 100176, 2022, doi: 10.1016/j.array.2022.100176.
- [4] A. Ricci, A. Croatti, S. Mariani, *et al.*, "Web of Digital Twins," *ACM Trans. Internet Technol.*, vol. 22, no. 4, p. 101:1-101:30, Nov. 2022, doi: 10.1145/3507909.
- [5] Digital Twin Consortium, "Definition of a Digital Twin," Object Management Group, Inc. Accessed: Oct. 18, 2021. [Online]. Available: <https://www.digitaltwinconsortium.org/initiatives/the-definition-of-a-digital-twin.htm>
- [6] J. Michael, J. Pfeiffer, B. Rumpe, *et al.*, "Integration challenges for digital twin systems-of-systems," in *Proceedings of the 10th IEEE/ACM International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems*, in SESoS '22. New York, NY, USA: Association for Computing Machinery, Nov. 2022, pp. 9–12. doi: 10.1145/3528229.3529384.
- [7] C. Human, A. H. Basson, and K. Kruger, "A Design Framework for a System of Digital Twins and Services," *Comput. Ind.*, vol. 144, p. 103796, Jan. 2023, doi: 10.1016/j.compind.2022.103796.
- [8] M. Zenger, "Programming language abstractions for extensible software components," Dissertation, EPFL, Lausanne, Switzerland, 2004. doi: 10.5075/epfl-thesis-2930.
- [9] R. Minerva and N. Crespi, "Digital Twins: Properties, Software Frameworks, and Application Scenarios," *IT Prof.*, vol. 23, no. 1, pp. 51–55, Jan. 2021, doi: 10.1109/MITP.2020.2982896.
- [10] A. W. Wymore, *Model-Based Systems Engineering: An Introduction to the Mathematical Theory of Discrete Systems and to the Tricotomy Design Theory of System Design*. in Systems engineering series. Boca Raton, Fla.: CRC Press, 1993.
- [11] S. Mihai, M. Yaqoob, D. V. Hung, *et al.*, "Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects," *IEEE Commun. Surv. Tutor.*, vol. 24, no. 4, pp. 2255–2291, 2022, doi: 10.1109/COMST.2022.3208773.
- [12] A. Hakiri, A. Gokhale, S. B. Yahia, *et al.*, "A Comprehensive Survey on Digital Twin for Future Networks and Emerging Internet

- of Things Industry,” *Comput. Netw.*, vol. 244, p. 110350, May 2024, doi: 10.1016/j.comnet.2024.110350.
- [13] J. Osho, A. Hyre, M. Pantelidakis, *et al.*, “Four Rs Framework for the development of a digital twin: The implementation of Representation with a FDM manufacturing machine,” *J. Manuf. Syst.*, vol. 63, pp. 370–380, Apr. 2022, doi: 10.1016/j.jmsy.2022.04.014.
- [14] H. Bennett, M. Birkin, J. Ding, *et al.*, “Towards Ecosystems of Connected Digital Twins to Address Global Challenges,” The Alan Turing Institute, London, May 2023. doi: 10.5281/zenodo.7840266.
- [15] A. J. Lotka, “Analytical Note on Certain Rhythmic Relations in Organic Systems,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 6, no. 7, pp. 410–415, July 1920.
- [16] B. P. Zeigler, C.-J. Luh, and T.-G. Kim, “Model Base Management for Multifaceted Systems,” *ACM Trans. Model. Comput. Simul.*, vol. 1, no. 3, pp. 195–218, July 1991, doi: 10.1145/140765.140791.
- [17] S.-W. Lin, K. Watson, G. Shao, *et al.*, “Digital Twin Core Conceptual Models and Services,” Industry IoT Consortium, Nov. 2023. [Online]. Available: <https://www.iiconsortium.org/iic-download-digital-twin-core-paper/>
- [18] National Academies of Sciences, Engineering, and Medicine, “Foundational Research Gaps and Future Directions for Digital Twins,” The National Academies Press, Washington, D.C., Consensus Study Report, Mar. 2024. doi: 10.17226/26894.
- [19] D. L. Van Bossuyt, D. Allaire, J. Bickford, *et al.*, “The Future of Digital Twin Research and Development,” *J. Comput. Inf. Sci. Eng.*, vol. 25, no. 8, p. 080801, Apr. 2025, doi: 10.1115/1.4068082.
- [20] INCOSE, “Systems Engineering Vision 2035,” INCOSE, West Lafayette, IN, 2021. Accessed: Feb. 18, 2025. [Online]. Available: <https://www.incose.org/2021-redesign/load-test/systems-engineering-vision-2035>
- [21] International Organization for Standardization, *Automation systems and integration—Interoperability of capability units for manufacturing application solutions*, International Standard ISO 16300-1:2018, Oct. 2018. Accessed: Dec. 08, 2025. [Online]. Available: <https://www.iso.org/standard/64684.html>
- [22] International Organization for Standardization, *Information Technology—Cloud Computing*, International Standard ISO/IEC 22123-1:2023, Feb. 2023. Accessed: Dec. 08, 2025. [Online]. Available: <https://www.iso.org/standard/82758.html>
- [23] International Organization for Standardization, *Information Technology-Cloud Computing-Interoperability and Portability*, International Standard ISO/IEC 19941, Dec. 2017. doi: 10.3403/30313036U.
- [24] International Organization for Standardization, *Internet of Things (IoT) — Interoperability for IoT Systems*, International Standard ISO/IEC 21823-1:2019, Feb. 2019. Accessed: Dec. 08, 2025. [Online]. Available: <https://www.iso.org/standard/71885.html>
- [25] J. Morris, E. Louis, D. L. Van Bossuyt, *et al.*, “Constraint Hypergraphs as a Unifying Framework for Digital Twins,” *Rev. Syst. J.*, July 2025, doi: 10.48550/arXiv.2507.05494.
- [26] C. Lehner, A. Padovano, C. Zehetner, *et al.*, “Digital Twin and Digital Thread Within the Product Lifecycle Management,” *Procedia Comput. Sci.*, vol. 232, pp. 2875–2886, Jan. 2024, doi: 10.1016/j.procs.2024.02.104.
- [27] International Organization for Standardization, *Automation systems and integration — Digital twin framework for manufacturing*, ISO Standard No. 23247:2021, Switzerland., Oct. 2021. Accessed: Feb. 02, 2024. [Online]. Available: <https://www.iso.org/obp/ui/en/#iso:std:iso:23247:-1:ed-1:v1:en>
- [28] C. Schmidt, F. Volz, L. Stojanovic, *et al.*, “Increasing Interoperability between Digital Twin Standards and Specifications: Transformation of DTDL to AAS,” *Sensors*, vol. 23, no. 18, p. 7742, Jan. 2023, doi: 10.3390/s23187742.
- [29] J. Akroyd, S. Mosbach, A. Bhave, *et al.*, “National Digital Twin of the UK – a Knowledge-Graph Approach,” Dec. 18, 2020, Cambridge, UK. Accessed: May 23, 2024. [Online]. Available: <https://como.ceb.cam.ac.uk/media/preprints/c4e-preprint-264.pdf>
- [30] S. Gil, E. Kamburjan, P. Talasila, *et al.*, “An Architecture for Coupled Digital Twins with Semantic Lifting,” *Softw. Syst. Model.*, vol. 24, no. 5, pp. 1379–1404, Oct. 2025, doi: 10.1007/s10270-024-01221-w.
- [31] X. Shen, D. J. Wagg, M. Tipuric, *et al.*, “Digital Twins as Self-Models for Intelligent Structures,” *Sci. Rep.*, vol. 15, no. 1, p. 30327, Aug. 2025, doi: 10.1038/s41598-025-14347-8.
- [32] M. Drobňakovic, G. Shao, A. Nikolov, *et al.*, “Towards Ontologizing a Digital Twin Framework for Manufacturing,” in *Advances in Production Management Systems. Production Management Systems for Responsible Manufacturing, Service, and Logistics Futures*, in IFIPAICT, vol. 689. Trondheim, NO: Springer, Cham, Sept. 2023, pp. 317–329. doi: 10.1007/978-3-031-43666-6_22.
- [33] Y. Huang, S. Dhoub, L. P. Medinacelli, *et al.*, “Semantic Interoperability of Digital Twins: Ontology-based Capability Checking in AAS Modeling Framework,” in *2023 IEEE 6th International Conference on Industrial Cyber-Physical Systems (ICPS)*, May 2023, pp. 1–8. doi: 10.1109/ICPS58381.2023.10128003.
- [34] M. G. Kapteyn, J. V. R. Pretorius, and K. E. Willcox, “A probabilistic graphical model foundation for enabling predictive digital twins at scale,” *Nat. Comput. Sci.*, vol. 1, no. 5, pp. 337–347, May 2021, doi: 10.1038/s43588-021-00069-0.
- [35] J. Gregory and A. Salado, “Model-Based Verification Strategies Using SysML and Bayesian Networks,” in *The Proceedings of the 2023 Conference on Systems Engineering Research*, D. Verma, A. M. Madni, S. Hoffenson, and L. Xiao, Eds., Cham: Springer Nature Switzerland, 2024, pp. 19–33. doi: 10.1007/978-3-031-49179-5_2.
- [36] U. Memon, W. Mayer, M. Selway, *et al.*, “Interoperability of AI-enhanced digital twins,” *J. Ind. Inf. Integr.*, vol. 48, p. 100961, Nov. 2025, doi: 10.1016/j.jii.2025.100961.
- [37] E. Y. Song, M. Burns, A. Pandey, *et al.*, “IEEE 1451 Smart Sensor Digital Twin Federation for IoT/CPS Research,” in *2019 IEEE Sensors Applications Symposium (SAS)*, Mar. 2019, pp. 1–6. doi: 10.1109/SAS.2019.8706111.
- [38] F. Longo, A. Padovano, L. Caputi, *et al.*, “Distributed Simulation for Digital Twins: an Application to Support the Autonomous Robotics for the Extended Ship,” in *2022 IEEE/ACM 26th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, Sept. 2022, pp. 179–186. doi: 10.1109/DS-RT55542.2022.9932057.
- [39] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Framework and Rules*, 1516–2010, Aug. 2010. doi: 10.1109/IEEESTD.2010.5553440.
- [40] Y. Ma, S. Gowda, R. Anantharaman, *et al.*, “ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling,” Feb. 09, 2022, *arXiv: arXiv:2103.05244*. doi: 10.48550/arXiv.2103.05244.

- [41] L. Morris, A. Baas, J. Arias, *et al.*, "Decapodes: A diagrammatic tool for representing, composing, and computing spatialized partial differential equations," *J. Comput. Sci.*, vol. 81, p. 102345, Sept. 2024, doi: 10.1016/j.jocs.2024.102345.
- [42] M. Otter and H. Elmqvist, "Modelica," *Simul. News Eur.*, vol. 10, no. 29/30, pp. 3–8, Dec. 2000.
- [43] Modelica Association Project FMI, *Functional Mock-up Interface Specification*, Specification, Nov. 27, 2024. Accessed: Sept. 05, 2025. [Online]. Available: <https://fmi-standard.org/docs/3.0.2/>
- [44] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*, 3rd ed. Waltham: Elsevier, 2015.
- [45] J. Morris, G. Mocko, and J. Wagner, "Unified System Modeling and Simulation via Constraint Hypergraphs," *J. Comput. Inf. Sci. Eng.*, vol. 25, no. 6, p. 061005, Apr. 2025, doi: 10.1115/1.4068375.
- [46] J. Morris, G. Mocko, and J. Wagner, "Effects of Functional and Declarative Modeling Frameworks on System Simulation," presented at the The 5th Modeling, Estimation and Control Conference (MECC 2025), Pittsburgh, PA: IFAC, Oct. 2025.
- [47] J. Lussier, E. Y. Hua, A. Ledford, *et al.*, "Differentiating Between Digital Twins and Control Systems for Complex Systems," *J. Comput. Inf. Sci. Eng.*, vol. 25, no. 12, Dec. 2025, doi: 10.1115/1.4070329.
- [48] J. C. Mitchell, *Concepts in programming languages*. Cambridge: Cambridge University Press, 2003.
- [49] J. Beverley, J. Logan, and B. Smith, "About the Unreal," Nov. 12, 2025, *arXiv*: arXiv:2506.06284. doi: 10.48550/arXiv.2506.06284.