Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

REUSABLE ONTOLOGY GENERATION AND MATCHING FROM SIMULATION MODELS

Ming-Yu Tu
Hans Ehm
Abdelgafar Ismail
Philipp Ulrich

Infineon Technologies AG
Am Campeon 1-15
Neubiberg, 85579, GERMANY

ABSTRACT

As simulating semiconductor manufacturing grows complex, model reuse becomes appealing since it can reduce the time incurred in developing future models. Also, considering a large network of the semiconductor supply chain, knowledge sharing can enable the efficient development of simulation models in a collaborative organization. Such necessity of reusability and interoperability of simulation models motivates this paper. We will address these challenges through ontological modeling and linking of the simulation components. The first application is generating reusable ontologies from simulation models. Another discussed application is ontology matching for knowledge sharing between simulation components and a meta-model of the semiconductor supply chain. The proposed approach succeeds in automatically transforming simulation into reusable knowledge and identifying interconnection in a semiconductor manufacturing system.

1 INTRODUCTION

Companies continuously improve their operational efficiency in the supply chain to reduce increasing costs caused by global competition or unpredictable crises, such as the global pandemic due to COVID-19 or the Russian invasion of Ukraine (Alper 2023). With its dynamic and uncertain environment, semiconductor manufacturing requires simulations to provide decision support in such disruptive situations (Moench et al. 2011). A simulation is an approach to holistically capture the complexity of causal relationships in the semiconductor supply chain network and the accompanying heterogeneous data. Supply chain simulation is commonly implemented as a single model. With this approach, a company uses simulation software to create individual models for business activities, each representing different supply chain echelons, such as sourcing, manufacturing, or distribution (Fujimoto 2001). The reuse of simulation models becomes appealing, based on the intuition that time and cost for model development can be decreased (Robinson et al. 2004). Semiconductor supply chain activities are generally intertwined, so finding links among these models becomes essential. Divisions in the company expect to exchange data and reuse simulation building blocks among various models so that they can diagnose the problem from a holistic view. The distributed simulation approach was proposed to improve the reuse of simulation models. However, this approach is still facing some difficulties, so its usability is held for questioning (Bell et al. 2007). To achieve reusability and interoperability between simulation models, a common representational framework of these models should be developed (Rathnam and Paredis 2004).

Ontologies are a common approach to capturing and representing scattered pieces of information and therefore play an important role in enabling simulation interoperability and information sharing at an abstract level. This semantic technology supports the creation of machine-understandable, structured data and the introduction of a joint knowledge base across various domains. Thereby, ontologies enable knowledge reusability and extensibility for simulation components (Fayez et al. 2005). For example, the Digital Reference (DR) is a semiconductor supply chain ontology, which provides a shared knowledge framework and allows internal or external stakeholders to interact within mutual taxonomies and relationships (Ehm et al. 2019). The application of ontologies in the field of supply chain simulation has potential because attributes of simulation components can be extracted and then reused to build another model by using ontologies, saving costs and time spent on repetitively acquiring domain knowledge for different use cases that already exist (Ramzy et al. 2020). Furthermore, ontologies support the structured representation of knowledge and explicit definition of relations, simplifying the process of system understanding as well as mapping and connecting objects for information sharing (Benjamin et al. 2006). Hence, channeling information from simulation models into a semantic layer of ontologies is expected to reduce the usage of human resources and thus enable faster decision-making. However, no concrete work has yet been implemented to provide a practical framework for using simulation models to generate ontologies and to further allow interoperability in the context of the semiconductor supply chain.

This paper aims to use ontologies to store knowledge from simulation models and discover their interconnections in an overall system. Its result benefits the reusability and interoperability of the simulation. The structure of this paper is organized as follows: Section 2 provides an overview of existing literature in the field of ontology translation from simulation and the latest contributions to ontology matching. It is followed by the description of the methodology in Section 3. The results of the knowledge transformation will be discussed in Section 4, and the paper concludes with an outlook in Section 5.

2 LITERATURE REVIEW AND RESEARCH BACKGROUND

This section summarizes the approaches in the two fields of literature: ontology translation from simulations in the area of the semiconductor supply chain and ontology matching for simulation-translated ontologies.

2.1 Simulation-translated Ontology

Ontologies are a tool for knowledge management and have been used in industry for the structuring and storage of knowledge since the rising idea of the Semantic Web (Uschold and Gruninger 1996). Ontology translation has received different attention from simulation experts. Some of the contributions consider the use of ontologies to represent specific application domains, others incorporate the general view of simulation modeling through ontologies, and the others address approaches to the integration of both.

Regarding ontologies for domain modeling, Fayez et al. (2005) design an approach based on ontologies integrating heterogeneous supply chain aspects to address the complexity and usability of distributed simulation models for supply chain management. The authors base the ontologies on the Supply Chain Operation Reference model as the core reference to represent widely recognized knowledge within the supply chain community. Their work ends with a description of suitable OWL classes tailored to the supply chain function. Rabe and Gocev (2012) propose a Semantic Web framework for modeling and simulation for information preparation and result evaluation. They base it on a Reference Manufacturing Ontology for the explicit description of the manufacturing domain and then map ontologies generated from manufacturing domain data to the Reference Manufacturing Ontology structure through rule-based meditation. Terkaj et al. (2015) propose an ontological model to structure factory data from different sources or over various time spans in a single representation of the production system. This historical ontology is used to enable the continuity between the physical factory and its virtual model.

Considering ontologies for simulation modeling, Bell et al. (2007) propose an approach to simulation model reuse by using a simulation component ontology and semantic search architecture. The authors transform simulation models into ontologies based on the Discrete Event Simulation Component ontology to preserve the domain concepts of simulation components. Kernan and Sheahan (2010) develop an ontology identifying all the relevant entities, attributes, and activities of simulation in a manufacturing enterprise. The ontology is created for a specific simulation package, em-Plant.

Regarding the approach combining ontologies for both domain and simulation modeling, McGinnis et al. (2011) use a formal modeling language, OMG SysML, to create an ontology named Domain Specific Language for a class of simulation applications and therefore build a conceptual model for a domain problem. Subsequently, they translate a conceptual model into a computational simulation model by applying model transformation technology, i.e. the translation of a formal modeling language to a programming code, and SysML-based conceptual models. The mentioned technologies depend on a meta-modeling architecture as a transformation framework. Ramzy et al. (2020) propose a concept to automatically build simulation models by using ontologies. The concept aims to reuse ontologies to extract desired supply chain components for other models, resulting in saving time in collecting information and building models from scratch. The authors introduce a simulation ontology, which contains simulation entities and their parameters, and a domain ontology, which describes domain-specific requirements on a semantic basis. Next, the authors use a rule-based engine to interpret research objectives, determine required simulation entities, and confirm required domain-specific data for simulation. Finally, it outputs a table of simulation entities and parameter values for a simulation model. Jurasky et al. (2021) develop a Simulation Ontology as a foundation to serve the need for a generic model for any type of simulation model. It includes both the set of all potential model components or building blocks for the class taxonomy, and control logic for the system interrelationships for each type of model. Moreover, the authors create mapping rules to guide how to find and populate the required classes and data properties of a Simulation Ontology and the correct individuals of an existing domain ontology. Finally, the authors design a parser that automatically generates a simulation model from the instantiated Simulation Ontology. Listl et al. (2022) develop an ontological architecture in manufacturing simulation. They build two production ontologies representing both production tasks and resources and two simulation ontologies speaking for both simulation entities and scenarios.

In contrast to the majority of contributions, this paper focuses on a concrete implementation that allows the translation from simulation models to ontologies. Its translative ability is extended to all the simulation techniques. To the authors' knowledge, no contribution has been published in this field in combination with the complex simulation characteristics of the semiconductor supply chain.

2.2 Ontology Matching

To understand state-of-art ontology matching approaches tailored to supply-chain-simulation-related ontologies, different techniques will be reviewed. A correspondence matched is a relation that holds between entities from different ontologies. An entity can be a class or a property of an ontology (Euzenat and Shvaiko 2013, p.25-54). The relation can be, for example, equivalence, disjointness, or less general. This paper focuses on correspondences with the relation of equivalence and subsumption. The objective of ontology matching is to retrieve an alignment for a pair of ontologies. This is completed through a match operator that uses supporting information (e.g. dictionaries) and that is configured with a set of parameters (e.g. thresholds) (Euzenat and Shvaiko 2013, p.25-54). Various kinds and categories of algorithms in matching ontologies have been researched. Euzenat and Shvaiko (2013) presented a framework for classifying matching methods into an element or structure level. An element level considers only ontology entities, regardless of their relations, while a structure level examines the relations of entities and their instances.

Regarding the research on the element level of matching methods, Saba and Mohamed (2013) introduce an ontological mapping approach for transmitting domain information to simulation entities. Its mapping technique is based on pair mapping between ontologies and general process modeling concepts common in the domain and discrete event simulation application ontologies. Fengel (2014) develops a linguistic matcher that automatedly determines the semantic similarity of ontological elements. It identifies equivalence among all the ontology labels and takes stop words and synonyms into account. Moreover, it performs stemming to reduce inflected words to their grammatical stems and then uses a string algorithm to determine similarity. Li et al. (2018) propose an ontology of product knowledge integration and a process of ontology mapping and merging. Similarity scores between two elements are determined by calculating the attribute and concept similarities. Jirkovský et al. (2018) propose a semi-automated approach to integrating semantic information into a supply chain ontology. It starts from text normalization and verifies semantic equivalence between a concept name and an ontology term. Kim et al. (2022) propose a deep-model-based entity alignment framework for the knowledge graph to reconcile the semantic heterogeneity. The authors perform model learning to align concept and instance entities. Two models, BERT for the concept entities and generative adversarial networks for the instance entities are applied. Ocker et al. (2022) introduce a semi-automated framework to merge the semantic elements of production ontologies. The framework focuses on the linguistic analysis of labels and the structural analysis of finding graph isomorphism by identifying the correspondence of classes, data properties, and object properties.

Considering the research on the structure level of matching methods, Ye et al. (2008) propose an architecture to integrate supply chain components using Semantic Web technologies. The architecture bases itself on a Supply Chain Ontology to assure semantic consistency of knowledge mapping with domain ontologies. The authors map semantically similar concepts based on SWRL rules. Lu et al. (2013) align concepts between the Supply Chain Operations Reference ontology and the Product Ontology by identifying linguistically or taxonomically common concepts and then expressing concepts by axioms in descriptive logic. Finally, the authors adopt SWRL to define mapping rules and use the Pellet inference engine to verify aligned concept candidates.

3 METHODOLOGY

This paper aims to derive both translated ontologies from simulation models and matched ontological pairs from translated simulations. This section introduces the simulation-enabled knowledge transformation framework of generating a simulation-translated ontology and its matched pairs with a meta ontology.

3.1 Simulation-enabled Knowledge Transformation Framework

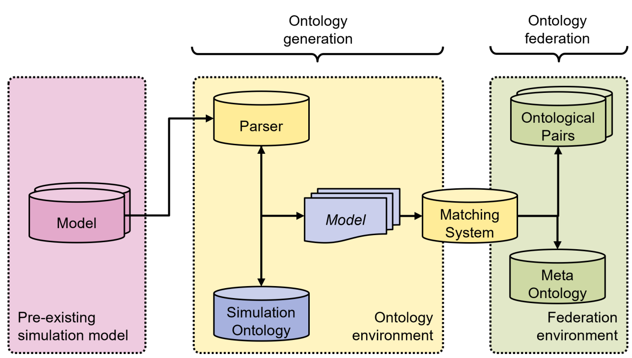

Figure 1: Framework for simulation-enabled knowledge transformation.

As depicted in Figure 1, the proposed framework includes three layers. The first layer is represented by pre-developed simulation models defined by specific simulation software, such as AnyLogic Project (ALP) files. Since a simulation model in this paper serves as a reference to structure and visualize the information flow between all the stakeholders in the context of the semiconductor supply chain, it is used as a data source to develop a semantic model and to summarize components and relationships in the system.

The second layer represents the knowledge extracted from the simulation model by an ontology that acts as a conceptual model for all components and parameters of the simulation and semantic networks. The transformation interface addresses the procedure to transform the information of the simulation to an ontology serialization format representing RDF triples of ontologies, such as Turtle files. From a technical point, the interface is applied for AnyLogic, defined with an application-specific model description language. An auxiliary tool is required to generalize it for improved interpretability. Therefore, an external *Parser* is proposed. The technical script presented in the course of the paper retrieves the simulation ontology by translating from the XML-based simulation models into Turtle files. The purposes of the framework are both to simplify an explanation of models in general and to extend and relate to components of other models with relevant domain knowledge. AnyLogic is chosen as a simulation application for the framework. The layer of ontology generation can also be enabled for other simulation applications on the condition that the specific parts in the parser must be adapted to the corresponding model description language.

The third layer represents the federation of the transformed ontology on the meta ontology. It combines the acquired domain or system knowledge to define common or shared components or interoperability among them. For this purpose, a corresponding matching system provides general guidance for the decision of which two ontological entities equate. This challenge can be tackled by using a natural language processing technique that automatically matches instances of the transformed ontology with the classes of the meta ontology. However, given the hypothesis that this work focuses on the field of the semiconductor supply chain, the solution leverages the existing well-defined semiconductor supply chain ontology, the Digital Reference, as a meta-model to create the matching system and to provide a standardized semantic knowledge of various classes in ontologies (Ehm et al. 2019). Based on ontology, the matching of concepts and the identification of interrelationships can meet the semantic conventions of the semiconductor supply chain.

3.2 Simulation Ontology Extended for Ontology Generation

To facilitate the process of ontology generation, its backbone - the Simulation Ontology - is demonstrated in the following. The Simulation Ontology serves as a generic conceptual model for any type of simulation technique, namely Discrete Event Simulation (DES), Agent-Based Simulation (ABS), and System Dynamics (SD) (Grigoryev 2022, p.23). The ontology extracts both the terminological and theoretical concepts from the practical simulation software to fulfill its applicability. Concerning the reuse of previous works, the extent of suitable starting points is limited to the Simulation Ontology presented by Jurasky et al. (2021). This work extends the building blocks and parameters required for the specific use case that will be validated in Section 4. The extended items are listed in Table 1

Table 1: Extended ontology concepts within the Simulation Ontology.

New item	OWL concept	Modeling purpose/relationship
Enter	Class	Transfer agents created e.g. statecharts
Exit	Class	Implement custom routing of agents
agentType	Object properties	Domain: EntityActivity; Range: Entity
HasNoMaximumCapacity	Data properties	Domain: Queue
IsContinuous	Data properties	Domain: Statistics
IsDiscrete	Data properties	Domain: Statistics
OccurenceTime	Data properties	Domain: Event
Rate	Data properties	Domain: Event

The set of resulting classes representing building blocks of a simulation model lays a foundation of the Simulation Ontology. To organize them according to their respective simulation techniques, an appropriate taxonomy is organized. The highest level of classes covers technique-independent concepts, including `BuildingBlocks` and type-independent `GlobalElements`. The lowest level of classes, such as `Enter` and `Exit`, represents the actual building blocks of a simulation model and is especially important since they will serve the purpose of being instantiated to become the components of the model.

The properties within the Simulation Ontology also play a critical role in model formalization. The object properties with their respective domain and range are organized to support the understanding of the structural model dependencies. For example, different types of `EntityActivity` have a property of `agentType` linking to the `Entity`.

The data properties handle the parametrization of the model by setting the potential parameter values of the building blocks. They are displayed with the corresponding class as a domain. For example, the

Event has a Rate. While the object properties are generally defined to describe the semantic network of the class taxonomy, the data properties are designed depending on the simulation software. AnyLogic offers a wide range of building blocks with the corresponding parameters, covering as many simulation use cases as possible. However, this paper focuses on the data properties required for the model in the use case application presented in Section 4, especially building blocks for the DES modeling technique.

3.3 Parser Automating Ontology Generation

To automatically derive an ontology from a simulation model, the translated ontology must be based on the Simulation Ontology. With this ontological basis, the parsing script aims to translate the relevant simulation elements into an ontology language. The script runs through the following general logic: data collection, parsing, ontology engineering, and finally Turtle-writing. The resulting ontology generation workflow is summarized in Figure 2. It further highlights what is automated by the Parser.

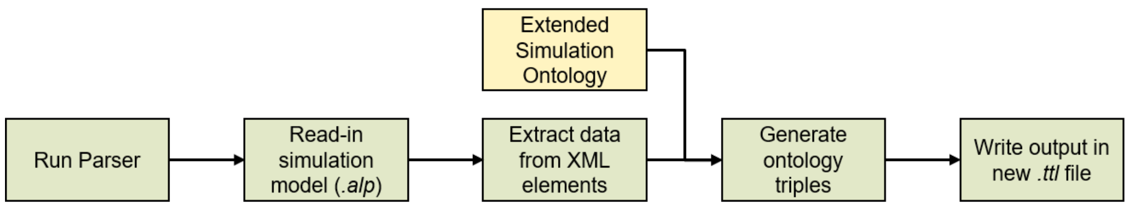

Figure 2: Workflow of the Parser.

1. Data collection: The script imports a pre-developed ALP model. Since an ALP file is an XML-tree following a certain structure to define a simulation model, the Parser encoded in Python uses the ElementTree and Pandas libraries to parse XML and thereafter store the output data in tables.
2. Parsing: An algorithm processes building blocks and extracts relevant elements of the corresponding tag for each instance at the designated position in the XML-tree. After an element is located, the algorithm runs through a list of parameters to extract the data assigned to the respective element, before proceeding to the next building block and repeating the same procedure. The list of building blocks is processed according to the general sequence defined by the ALP schema. In particular, all elements of the Variable class are followed by Dependences, Events, StateChartElements, Events, EmbeddedObjects, AnalysisData, and associated agents.
3. Ontology engineering: The Simulation Ontology serves as a fundamental structure to translate simulation models to ontologies. It defines the classes and object properties for any kind of simulation technique, so it can be referred to and then enables the insertion of the associated individuals. The relevant individuals stored in the tables are looked up to match tuples belonging to the corresponding classes. Their associated properties and corresponding domains or values are also combined to form triples. The matched records are added into a graph instantiated with the RDFLib library to form an ontology. Moreover, the properties introduced in the table are formally labeled and added with their corresponding domains into the graph to further formalize the ontology.
4. Turtle-writing: the concatenated triples in the graph are serialized in a Turtle file to represent the source code of a desired ontology. This completes the ontology with the classes and object properties from the Simulation Ontology and the instances and data from the simulation model.

3.4 Matching System for Ontology Federation

Having generated ontologies from simulation models, the simulation components are represented by concepts or relations in the knowledge base. The representation enables the next step of matching them with elements in the meta ontology. The meta ontology selected is the Digital Reference, a Semantic Web representation of the semiconductor supply chain that provides a structure of a knowledge base readable for both humans and computers (Ramzy et al. 2022). Various techniques can perform the ontology matching operation in an automated fashion (Euzenat and Shvaiko 2013). Textual content, such as labels and descriptions, is one of the strongest signals representing the semantics of an ontological entity. Therefore, matching techniques usually borrow concepts from the Natural Language Processing community to determine the semantic similarity between two entities. The old-fashioned methods, such as string-based comparisons, are simple and thus do not capture the actual semantic meaning of the texts. Comparing pairs of all textual descriptions of concepts in two ontologies is expensive and scales quadratically. This issue escalates when a concept has multiple descriptions. To overcome this problem, this paper presents a Sentence Bidirectional Encoder Representations from Transformers (SBERT) to the ontology matching task (Reimers and Gurevych 2019). The matching system is mainly divided into two parts in the multi-step pipeline, illustrated in Figure 3.

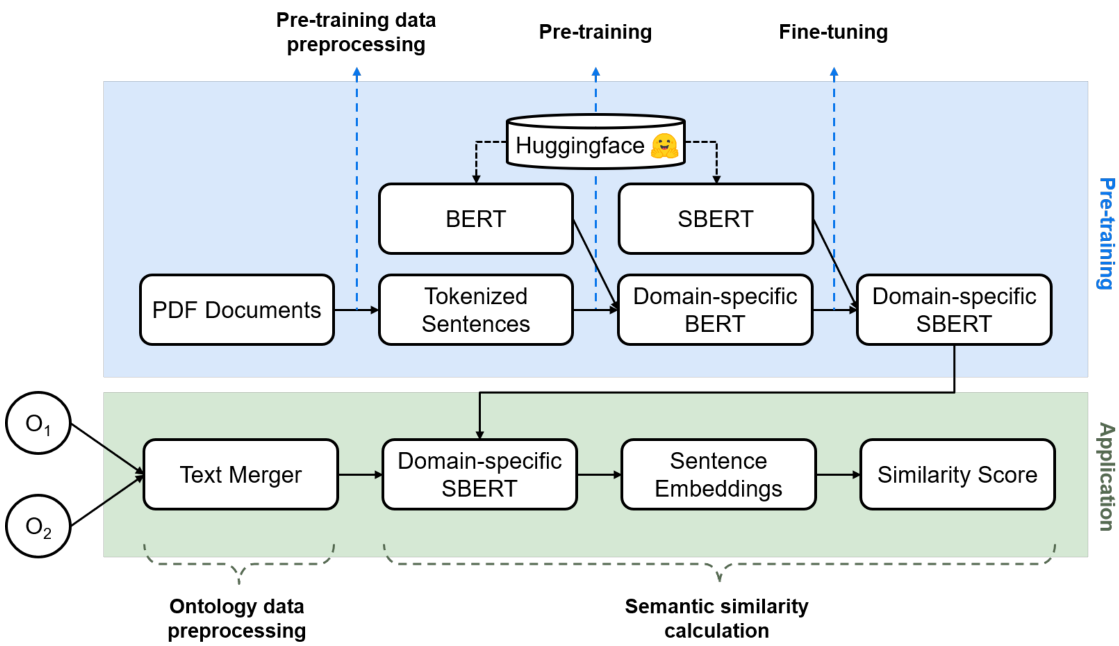

Figure 3: Overview of the matching system.

- Pre-training: In the first step, tokenized sentences are generated from raw data of documents. Subsequently, an SBERT model is designed specifically for the domain of the semiconductor supply chain such that it can understand and analyze the semantics in this field.
- Application: Each instance from the transformed ontology O_1 is to be paired with a class of the meta-ontology O_2 by comparing textual representations. A text can be retrieved, for example, by concatenating the URI fragment and annotation properties. To turn textual descriptions into sentences, they are merged for each of the ontological entities. The SBERT loads the textual input and then derives embeddings such that textual pairs are close in the semantic space of the semiconductor supply chain when they have a similar meaning. Therefore, based on the embeddings, similarity scores can be calculated and used to match ontological pairs.

3.4.1 Pre-training

First, the conversion from PDF to TXT is addressed. Then, the text contained in the documents is tokenized into sentences. Finally, to further pre-train BERT on a domain corpus, the data input is generated in a specific format. To tailor the language model to the domain of the semiconductor supply chain, processing the related input documents is inevitable to create the text corpus on which BERT is pre-trained (Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova 2018). Therefore, Infineon provides PDF documents as the data input for the pre-training. They are scientific articles, Bachelor's, Master's, and PhD's theses, related to the topic of the semiconductor supply chain. The dataset contains a balanced number of documents from all sub-domains, such as artificial intelligence, Semantic Web, and digitalized supply chain, so the model can fairly capture its vocabulary. Since BERT is fed with plain text, the PDFs are converted into TXT files. Once the files are converted into the TXT format, the corpus in the documents is tokenized into sentences by using an open-source NLP library: spaCy (Honnibal and Montani 2017). The sentence segmentation task is performed using a dependency parser that supports a statistical model to tokenize sentence boundaries. Next, all the sentence-tokenized documents are concatenated into a file appending an empty line between documents. The raw text is concatenated until reaching the maximum sequence length to reduce the computation for padding and used for pre-training for a BERT model. Once the domain-specific BERT is built, fine-tuning can start. The idea of fine-tuning is to train an SBERT on the domain-specific BERT. The SBERT generates sentence embeddings by using a Siamese network on top of two BERT instances. On top of each instance, a pooling layer is used to generate a fixed length of the sentence embedding (Reimers and Gurevych 2019). Finally, the SBERT is trained on a combination of MultiNLI and SNLI datasets and evaluated on the STS dataset. On top of the pooling layers, a soft-max function is applied to classify the resulting embeddings as entailment, neutral, or contradiction. This idea is to fine-tune the model over the Natural Language Inference task to obtain the domain-specific SBERT.

3.4.2 Application

The text merger extracts textual descriptions from ontological resources, removes their repetitive parts, and merges them. Matching resources in ontologies requires their texts to be extracted as a set of strings. The specific literals or URI fragments are queried depending on the ontology sources, simulation-translated ontology, and the DR respectively. Regarding simulation-translated ontology, a list of all building blocks, i.e. all defined individuals, corresponding parameters, and respective values, i.e. inherited and instantiated properties per individual, is required. As for the DR, its text extraction focuses on all literals where the URI fragment of the property is either a label, comment, definition, or note. This includes `rdfs:label` and `rdfs:comment`. Optionally, all object properties can be extracted as textual descriptions and concatenated to the extracted literals. Before the language model encodes texts into numeric representations, the text merger also cleans texts by removing whitespaces, removing punctuations, and converting them into lowercase. Next, it reduces the set of texts further by checking if a text is fully contained in another text. In this case, the text is not returned. First, this process investigates if a class resource fragment is included in the label property. Then, it checks if this short text is contained in the long texts, such as comment and definition properties. This reduces the set of literals even more because labels that appear in a comment are also not returned. The merged texts out of these relationships are concatenated into a pool of sentences referencing the classes of the ontology, leading to the formulation of the proper input to the domain-specific SBERT. Once domain-specific SBERT is trained and ontology data is preprocessed, it is ready to create the embedding representing the entity of the ontology and compare the elements from the simulation-translated ontology and those from the DR to evaluate their similarities. Therefore, a pool of sentences belonging to a class of ontology is processed by the model. Then, the output embedding is generated and stored in an array. Once the embeddings of two ontologies are available, meaningful vector operations, such as cosine or Euclidean distance, are applied to determine similarity scores between embeddings from one ontology and those from the other and to identify which ontological pairs are best matched. The ontological pairs with the highest similarity scores are saved to a table and then further labeled if they pass a threshold.

4 RESULTS: CASE STUDY

To test the effectiveness of simulation component transformation and matching performance, the framework presented in the previous section is evaluated with a model simulating energy efficiency in semiconductor manufacturing (Hopf et al. 2022). The first step of knowledge transformation is automated ontology generation by using the presented *Parser*. The entire resulting simulation-translated ontology for the energy efficiency in semiconductor manufacturing is illustrated in Figure 4. The ontology includes multiple individuals of the entity activity to represent the flows of the front-end production and the central infrastructure. The visualization of the activities illustrates how they interact with one another in the context of the simulation system. Their corresponding data properties further illustrate what parameter values they have in the simulation model. For example, the individuals of the entity `Source` are associated with their respective property values of `PushProtocal`. Moreover, the ontology consists of three `Entity` (agents) and their associated `GlobalElements`. The complexity of the transformation is that the agents have data properties shared with an enormous number of variables and parameters since the energy consumption calculation requires both fixed and variable energy factors.

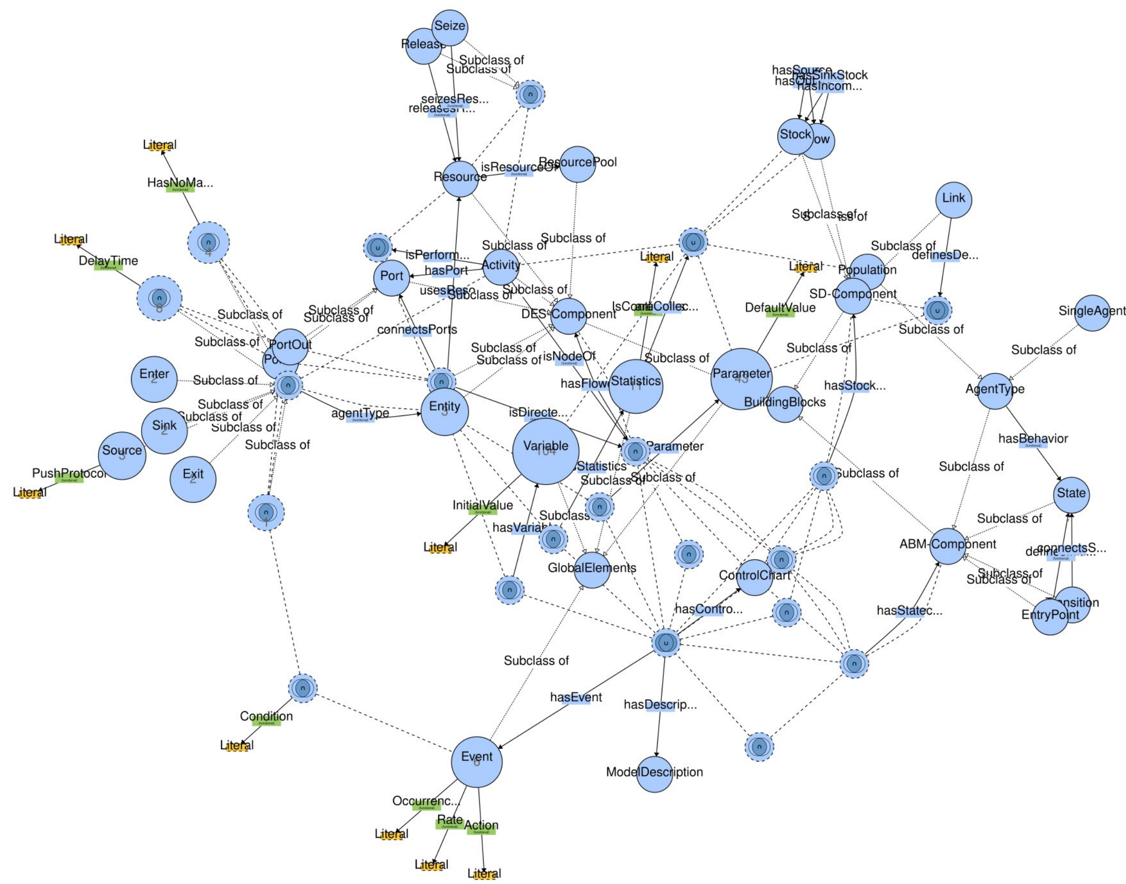

Figure 4: Ontology translated from the simulation model of energy efficiency.

The second step of knowledge transformation is the matching of the simulation-translated ontology with the DR. We begin by providing sufficient textual descriptions for the pre-trained SBERT model. From the ontologies, the concepts, Lithography and Photolithography respectively, are taken as examples. Table 2 demonstrates an implementation of matching the two concepts. First, textual descriptions are extracted and merged into sentences for each ontological entity. For instance, the class Lithography from the DR has two annotation properties, namely label and comment, and one object property Requires. Since the word in the label is already contained in the comment, the former is not concatenated in the final sentences. Next, the object property and its associated classes are combined into a sentence. For instance, Requires bridges between Lithography and Mask. Likewise, the procedure of text extraction and merging is applied to the individual Photolithography from the simulation-translated ontology. The resulting final sentences represent the two concepts respectively. Finally, these sentences are fed into the SBERT model to generate the respective embeddings and then calculate a similarity score of the two concepts.

Table 2: Example of the matching concepts.

Ontology	Label	Comment	Property	Domain/ Range	Sentences
Digital Reference	Lithography	The lithography process is the key process in the frontend…	Requires	Mask	• The lithography process is the key process in frontend… • Mask requires lithography.
Simulation-translated Ontology	Photolithography		Delay Time	Photolit_RPT/OxiLoops	Photolithography delay time is Photolit_rpt/Oxi loops.

As a baseline, the string-based method of Levenstein distance is used (Levenshtein 1966). The SBERT model from the Hugging Face repository is used: paraphrase-TinyBERT-L6-v2 (Reimers and Gurevych 2019). After the calculation of similarity scores by the system, we use domain expert knowledge to classify whether the semantic matches are hit or not. The resulting list of matches considering object properties as textual input is presented in Table 3.

Table 3: Matching results considering object properties.

Simulation-translated ontology	Digital Reference (Levenstein)	Digital Reference (SBERT)	Similarity Score	Hit (SBERT)
ion implantation	implantation	implantation	0.772	true
etching	machine	dry etching	0.675	false
alpha ion implant	actuation range	implantation	0.667	true
counter ion implant	customer plant	implantation	0.659	true
ion implantation queue	implantation	implantation	0.659	true
counter ion implant acc	customer plant data	implantation	0.637	true
ion implant rpt	implantation	implantation	0.635	true
ion implant fix share	constant resistance	implantation	0.605	true
ion implant capa	implantation	implantation	0.603	true
calculation month ion implant	customer plant	implantation	0.600	true
ion implant utiliz statistics	ro hs compliant status	implantation	0.562	true
ion implant total share	ro hs compliant status	implantation	0.561	true

The result proves that the literal-driven matching system can discover the most relevant concept of the DR for each instance of the translated ontology by ranking the highest similarity scores. If a score is above a parameterized threshold of 0.55, the correspondence is deemed as a reasonable pair with the relation of either equivalence or subsumption. Although the SBERT trained in the field of the semiconductor supply chain does not have a comparable universal performance benchmark because of a lack of a labeled reference alignment, the results show that the concepts matched by the SBERT are generally more sensible compared with the baseline. It can be argued that these matching differences are because SBERT can better capture the overall textual meaning of each ontological element, while the baseline can only compare the character differences of the element titles. On the other hand, the resulting pairs matched by the SBERT are further manually evaluated by domain experts to confirm whether they are hit or not. The matched concepts scored above 0.55 are overall superclasses of the concepts from the simulation model. For instance, all of the metric concepts related to ion implantation from the simulation model are categorized under the concept of implantation. However, since the development of the meta ontology is ongoing and might not include all the concepts from its domain, failed matches could occur. For example, because no general concept of etching exists in the DR, etching from the simulation model can only be matched with dry etching from the DR, a match being the reverse subsumption.

5 CONCLUSION AND OUTLOOK

Our framework provides an opportunity to store simulation components in an ontology for efficient knowledge management and to identify their relationships with the overall system. This enables fast knowledge-based decision-making in an uncertain and complex environment. Our framework includes two processes: (a) ontology generation, i.e. the automatic translation from simulation models to ontologies with the support of the extended Simulation ontology and the technical parser, (b) ontology federation, which enables the automatic matching between the simulation-translated ontology and the meta ontology by using the matching system. Our solution is classified as semi-automatic, whereby ontology generation is automated and the steps of ontology federation are supported. It is not limited to a particular simulation type but supports all simulation modeling techniques. Moreover, our solution supports simulation-based ontology matching and enables the relation identification between simulation models and a meta ontology, allowing for information exchange and the further causal inference that simulated objects would affect the overall supply chain system. A use case model studying the impact of energy consumption on the global supply chain of a semiconductor manufacturer is used to prove the concept. In particular, the framework translates the used components of DES modeling techniques and matches them with the DR.

REFERENCES

Alper, A. 2023. “Exclusive: Russia’s Attack on Ukraine Halts Half of World’s Neon Output for Chips”. https://www.reuters.com/technology/exclusive-ukraine-halts-half-worlds-neon-output-chips-clouding-outlook-2022-03-11/, accessed 20.03.2023.

Bell, D., S. de Cesare, M. Lycett, N. Mustafee, and S. J. E. Taylor. 2007. “Semantic Web Service Architecture for Simulation Model Reuse”. In *Proceedings of 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT’07)*, 129–136. Oct. 22th-26th, USA.

Benjamin, P., M. Patki, and R. Mayer. 2006. “Using Ontologies for Simulation Modeling”. In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, B. Lawson, J. Liu, F. Wieland, D. M. Nicol, and R. M. Fujimoto, 1151–1159. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova 2018. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”.

Ehm, H., N. Ramzy, P. Moder, C. Summerer, S. Fetz, and C. Neau. 2019. “ECSEL Digital Reference: A Semantic Web for Semiconductor Manufacturing and Supply Chains Containing Semiconductors”. In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 2409–2418. National Harbor, Maryland: IEEE Press.

Euzenat, Jérôme and Shvaiko, Pavel 2013. “Ontology Matching, Second Edition.”.

Fayez, M., L. Rabelo, and M. Mollaghasemi. 2005. “Ontologies for Supply Chain Simulation Modeling”. In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 2364–2370. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fengel, J. 2014. “Semantic Technologies for Aligning Heterogeneous Business Process Models”. *Business Process Management Journal* 20(4):549 – 570.

Fujimoto, R. M. 2001. “Parallel and Distributed Simulation Systems”. In *Proceeding of the 2001 Winter Simulation Conference*, 147–157 vol.1. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Grigoryev, I. 2022. “AnyLogic in Three Days: A Quick Course in Simulation Modeling”. 6th ed. Chicago: AnyLogic.

Honnibal, Matthew and Montani, Ines 2017. “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing”.

Hopf, A., A. Ismail, H. Ehm, D. Schneider, and G. Reinhart. 2022. “Energy-Efficient Semiconductor Manufacturing: Establishing an Ecological Operating Curve”. In *Proceedings of the 2022 Winter Simulation Conference*, edited by B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C. G. Corlu, L. H. Lee, E. P. Chew, T. Roeder, and P. Lendermann, 3453–3464. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Jirkovský, V., O. Šebek, P. Kadera, and N. Rychtyckyj. 2018. “Heterogeneity Reduction for Data Refining Within Ontology Learning Process”. In *Proceedings of IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 3108–3113. Oct. 22th-23th, Washington, DC.

Jurasky, W., P. Moder, M. Milde, H. Ehm, and G. Reinhart. 2021. “Transformation of Semantic Knowledge into Simulation-based Decision Support”. *Robotics and Computer-Integrated Manufacturing* 71:102174.

Kernan, B., and C. Sheahan. 2010. “Development and Construction of an Ontology to Represent Simulation Data for a Generic Enterprise”. *Applied Ontology* 5(1):29–46.

Kim, J., K. Kim, M. Sohn, and G. Park. 2022. "Deep Model-Based Security-Aware Entity Alignment Method for Edge-Specific Knowledge Graphs". *Sustainability* 14(14):8877.

Levenshtein, V. I. 1966. "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals". *Soviet Physics Doklady* 10(8):707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

Li, X., Z. Wu, M. Goh, and S. Qiu. 2018. "Ontological Knowledge Integration and Sharing for Collaborative Product Development". *International Journal of Computer Integrated Manufacturing* 31(3):275–288.

Listl, F. G., J. Fischer, A. Sohr, S. Grimm, and M. Weyrich. 2022. "Ontological Architecture for Knowledge Graphs in Manufacturing and Simulation". In *Proceedings of 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–8. Sep. 6th-9th, Stuttgart, Germany.

Lu, Y., H. Panetto, Y. Ni, and X. Gu. 2013. "Ontology Alignment for Networked Enterprise Information System Interoperability in Supply Chain Environment". *International Journal of Computer Integrated Manufacturing* 26(1-2):140–151.

McGinnis, L., E. Huang, K. S. Kwon, and V. Ustun. 2011. "Ontologies and Simulation: A Practical Approach". *Journal of Simulation* 5(3):190–201.

Moench, L., J. W. Fowler, S. Dauzère-Pérès, S. J. Mason, and O. Rose. 2011. "A Survey of Problems, Solution Techniques, and Future Challenges in Scheduling Semiconductor Manufacturing Operations". *Journal of Scheduling* 14(6):583–599.

Ocker, F., B. Vogel-Heuser, and C. J. Paredis. 2022. "A Framework for Merging Ontologies in the Context of Smart Factories". *Computers in Industry* 135:103571.

Rabe, M., and P. Gocev. 2012. "Applying Semantic Web Technologies for Efficient Preparation of Simulation Studies in Manufacturing". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ramzy, N., H. Ehm, V. Wiens, and L. Kohnen. 2022. "The Digital Reference: Semantically Connecting Semiconductor Supply Chains to Customers -The Open Online Sales and Marketing Vision". In *Proceedings of the CASE 2021- International Conference on Automation Science and Engineering*. Aug. 23th-27th, Lyon, France.

Ramzy, N., C. James Martens, S. Singh, T. Ponsignon, and H. Ehm. 2020. "First Steps Towards Bridging Simulation and Ontology to Ease the Model Creation on the Example of Semiconductor Industry". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. G. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 1789–1800. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rathnam, T., and C. Paredis. 2004. "Developing Federation Object Models Using Ontologies". In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 1062. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Nils Reimers and Iryna Gurevych 2019. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks".

Robinson, S., R. E. Nance, R. J. Paul, M. Pidd, and S. J. Taylor. 2004. "Simulation Model Reuse: Definitions, Benefits, and Obstacles". *Simulation Modelling Practice and Theory* 12(7):479–494. Simulation in Operational Research.

Saba, F., and Y. Mohamed. 2013. "An Ontology-driven Framework for Enhancing Reusability of Distributed Simulation Modeling of Industrial Construction Processes". *Canadian Journal of Civil Engineering* 40(9):917–926.

Terkaj, W., T. Tolio, and M. Urgo. 2015. "A Virtual Factory Approach for In Situ Simulation to Support Production and Maintenance Planning". *CIRP Annals* 64(1):451–454.

Uschold, M., and M. Gruninger. 1996. "Ontologies: Principles, Methods, and Applications". *The Knowledge Engineering Review* 11(2):93–136.

Ye, Y., D. Yang, Z. Jiang, and L. Tong. 2008. "An Ontology-based Architecture for Implementing Semantic Integration of Supply Chain Management". *International Journal of Computer Integrated Manufacturing* 21(1):1–18.

AUTHOR BIOGRAPHIES

MING-YU TU is a MSc. in Management and Technology from the Technical University of Munich. He was part of the simulation team at Infineon Technologies AG from August 2022 until March 2023. His email address is mingyutu1219@gmail.com.

HANS EHM is Senior Principal Engineer Supply Chain of Infineon Technologies AG. For four decades he has been heading the Supply Chain Innovation department of Infineon Technologies AG. His email address is hans.ehm@infineon.com.

ABDELGAFAR ISMAIL MOHAMMED HAMED is a staff supply chain engineer of Infineon Technologies AG. He leads the Supply Chain Simulation team at Infineon Technologies AG. His email address is Abdelgafar.Ismail@Infineon.com.

PHILIPP ULRICH has a MSc. in Information Systems from the Technical University of Munich. He leads the Supply Chain Innovation Semantic Web team at Infineon Technologies AG. His email address is Philipp.Ulrich@infineon.com.