Software and Systems Modeling (2022) 21:1047–1069
https://doi.org/10.1007/s10270-022-00982-6

THEME SECTION PAPER

AI-driven streamlined modeling: experiences and lessons learned from multiple domains

Sagar Sunkle[1] · **Krati Saxena**[1] · **Ashwini Patil**[1] · **Vinay Kulkarni**[1]

Received: 17 March 2021 / Revised: 5 December 2021 / Accepted: 24 January 2022 / Published online: 19 February 2022

Abstract
Model-driven technologies (MD*), considered beneficial through abstraction and automation, have not enjoyed widespread adoption in the industry. In keeping with the recent trends, using AI techniques might help the benefits of MD* outweigh their costs. Although the modeling community has started using AI techniques, it is, in our opinion, quite limited and requires a change in perspective. We provide such a perspective through five industrial case studies where we use AI techniques in different modeling activities. We discuss our experiences and lessons learned, in some cases evolving purely modeling solutions with AI techniques, and in others considering the AI aids from the beginning. We believe that these case studies can help the researchers and practitioners make sense of various artifacts and data available to them and use applicable AI techniques to enhance suitable modeling activities.

Keywords AI-driven · Domain modeling · Natural language processing · Information extraction · Knowledge graphs

1 Introduction

The recent advent of Deep Learning (DL), first in images and videos [45] and then in text (with language models and transformers [27]), marked the beginning of a new era in Machine Learning (ML), Natural Language Processing (NLP), and more generally in Artificial Intelligence (AI)[1]. Model-driven Engineering, or MDE, where models drive software and systems engineering, is not alien to using AI techniques either. Still, there is a long way to go.

Models and associated practices (MD*) have been studied and used in practice for two qualities they bring to the table, namely *abstraction* and *automation*, the former enabling building similar systems and the latter enabling the desired implementation(s) via code generation. However, it has been argued that MD* has seen limited adoption because its benefits do not outweigh its costs and that cognification or use of AI techniques can drastically improve the benefits and reduce the cost of adoption [24].

The modeling community has recently started using AI techniques. However, the use of AI techniques is quite limited, and AI techniques are generally applied to one of the many activities in the modeling lifecycle prompting calls for broader application of AI techniques in modeling—cognifying modeling [24], intelligent modeling assistants [51], and new frameworks like data-driven modeling [25].

These new developments point to two main concerns- the need to leverage new and upcoming AI techniques in modeling activities [24,51] and embracing different kinds of models working with different kinds of data [25].

In our organization's modeling journey, we have had the opportunity to address both these concerns to some extent. Starting from using models purely to generate code with which we delivered 70+ business-critical enterprise applications, we shifted the gears to using models to analyze and aid in enterprise problem-solving [43]. This transition involved several industrial case studies where we have gradually introduced and increased the use of AI techniques, driven partly by customer ask and partly by realities of modern enterprises, including the increasing variety of data that enterprises seek to make functional.

[1] The two papers have massive citation counts, more than 36K for [45] published in 2015, and more than 16K for [27] published in 2018, indicating an extensive outreach and increasing use of DL and, in general, AI techniques in every human endeavor.

Communicated by L. Burgueño, J. Cabot, M. Wimmer & S. Zschaler.

✉ Sagar Sunkle
sagar.sunkle@tcs.com

[1] Tata Consultancy Services Research, Pune 411013, India

In this paper, we present selected case studies. These case studies have already been published, so it is not the specific problems we are interested in discussing. We present the description that abstracts out the details of how we have applied and continue to apply various AI techniques to enhance individual solutions. We call our perspective *AI-driven Streamlined Modeling*. Referring back to the two concerns pointed out earlier, our perspective on the case studies shows examples of:

1. How to use AI techniques across various modeling activities.
2. How to use a variety of data and artifacts in purposive modeling.

We use the following case studies to elaborate *AI-driven Streamlined Modeling*:

- *Regulatory compliance* providing a generic model-driven framework for regulatory compliance with many AI enhancements [62–64,74,75,77,78]
- *Document generation and checking* based on several industry standards using NLP and image processing [54,61]
- *Formulated product design* framework enabling human-in-the-loop generation of formulation recipes for *any* type of formulated products using information extraction and knowledge graphs [66,79,81,82]
- *Information to insights* framework for legal cases in any sub-domains such as parental alienation and divorce [68, 80]
- *What-if analyses for enterprises* that enable modeling and conducting what-if scenarios for a given situation in an enterprise [60,73]

The key benefit of the *AI-driven Streamlined Modeling* case studies is the examples of a diverse set of artifacts and AI techniques, along with the discussion of their industrial application contexts.

We arrange the paper as follows. We discuss the theme of *AI-driven Streamlined Modeling* by visiting each term in this phrase in Sect. 2. In Sect. 3, we organize and review the related work across various modeling activities. We present the five case studies in a structured manner by describing the problem context, challenges, modeling solutions, AI enhancements and discuss the lessons learned in Sects. 4.1–4.5. In Sect. 5, we discuss different ways to use AI techniques for specific artifacts and present our case studies as data-driven modeling [25] instances. In addition, we discuss the generalizability of *AI-driven Streamlined Modeling* and the lessons learned from the use of specific AI techniques. Section 6 concludes the paper.

2 Background

In the following, we begin by discussing what we mean by our perspective of *AI-driven Streamlined Modeling*. To discuss the background and the context, we split the phrase into three terms—*modeling* to explain kinds of models and modeling activities under consideration and *AI-driven* and *Streamlined* to discuss the role of AI techniques in modeling.

2.1 Modeling

The modeling community and modeling research are unique in terms of a plethora of different acronyms [20]. The very existence of so many acronyms (model-driven/-based *) indicates varied perspectives on what models are and how to use them. Instead of calling our approach architecture or framework, we choose to call it *AI-driven Streamlined Modeling*. In this phrase, we refer to the term *modeling* with the broadest applicability. In so far as the models under consideration possess *reduction* features and *pragmatic* features [42], i.e., the models selectively project the original system (reduction features) to achieve a purpose (pragmatic features) in place of the system, we refer to activities associated with these models collectively as *modeling*.

To establish the relevance of various AI techniques in modeling activities, we suitably divide the modeling activities from the creation of models to their specification, transformation, and population.

Accordingly, we refer to the activity of building a model of core concepts and relations of the target business domain as *domain modeling*. The specific format and formalisms are irrelevant to this activity. They come into the picture when specifying the model or in *model specification*. These may be UML class diagrams, OWL ontologies, RDF graphs, programs in general or domain-specific (modeling) languages, and even speech dictations and many other formats and formalisms.

Within the activity of *domain modeling*, we take the use of the phrases *conceptual models*, *concept models*, *ontologies*, and even *knowledge graphs* to convey the same intent—arriving at the core concepts and relations of the domain under consideration. For a finer distinction between models and ontologies in terms of *descriptiveness* or *prescriptiveness*, we request the reader to refer to [9]. For an additional property of *predictiveness* relevant especially concerning AI techniques applied to different kinds of data/artifacts, we refer the reader to [25]. For variations on the theme of knowledge graphs, we refer the reader to [87].

Depending on the purpose at hand, models may need to be transformed into other models or executable languages, i.e., to carry out *model transformation*. Model transformation may not always be necessary; the model specified at the

model specification activity may be used as-is to achieve the intended purpose [22].

In both cases, models may need to be populated with data on top of the generated code. If *model transformation* is not used, the models specified at the *model specification* activity need to be populated directly. The *model population* can take place by integration with a relational database, or it might need to be carried out by extracting data from (un-/ semi-) structured sources. The specified models could be in the form of knowledge graphs and the entire set of activities; from building the ontology underlying the knowledge graph and populating and maintaining the knowledge graph with relevant information may be referred to as *information modeling*.

Our objective is to present our case studies as not just modeling case studies; instead, we emphasize using AI techniques across different modeling activities. This takes us to the term *AI-driven*.

2.2 AI-driven

Over the last decade, the newfound success in ML, first in deep neural networks for vision (images and videos) and then in NLP with language models and transformers for text (including the latest Open GPT-3[2]), has resulted in a Cambrian explosion in the implementation ecosystems supporting these techniques and their applications across every domain and every possible system. The existence of massive data that every large enterprise routinely collected but not necessarily knew what to do with it has made it now quite customary to think of making something AI-driven.

Due to the enormous interest generated in AI, many AI techniques have been discovered to deal with every kind of data, including texts, images, videos, and sounds. We take the terms like AI-aided, AI-assisted, AI-enhanced to mean the same as AI-driven. Embedding/imbibing human knowledge and intelligence into machine processes, i.e., cognification and providing intelligent assistance, applied to modeling (resp. [24] and [51]) also points in the same direction.

In keeping with the discussion on *modeling*, our focus is to discuss AI techniques in each modeling activity. While a detailed discussion of specific AI techniques is out of this paper's scope, we are interested in elaborating how available AI techniques aid in processing artifacts in various modeling activities.

That leads us to discuss the artifacts available for each activity since specific AI techniques apply to specific artifacts. Such artifacts include a wide variety of textual artifacts with (un-/ semi-) structured nature (including the texts of Domain-specific Languages (DSLs)) and with and without embedded images, visual artifacts such as images (including images of visual models) and videos, and auditory artifacts such as voice commands and voice recordings.

One of the key lessons from customer interaction in our case studies is that customers always believe that complete automation of some modeling activity is possible using AI techniques but eventually become convinced that no modeling activity can be automated entirely. Human intervention is always required, and the aid provided is to the *human-in-the-loop*.

The combination of *modeling* activities with access to/availability of specific artifacts and correspondingly applicable AI techniques to make these activities *AI-driven* brings us to the term *streamlined*.

2.3 Streamlined

The term *streamlined* means *make (... system) more efficient and effective by employing faster or simpler working methods* [30]. Particularly one aspect of the definition points at being *effectively integrated* or *organized*. It is with this particular meaning that we approach the description of *streamlined* in *AI-driven streamlined modeling*. As such, we define *AI-driven Streamlined Modeling* as *an integrated and organized application of AI techniques relevant to artifacts/data available in and across modeling activities to model a system*.

It is easy to observe in the related work covered in Sect. 3 that most of the state of the art in using AI techniques in *modeling* focuses on a single modeling activity. We believe this is for a reason. Examples from academic research are primarily *modeling* examples. Examples from industry tend to evaluate first what the AI techniques can do for their problem situation, and then, often in secondary status, whether it can also benefit from modeling.

Early efforts to approach the solutions from both modeling and AI perspectives have begun recently with the Models and Data framework (MODA) [25], and Reference Framework for Intelligent Model Assistants (RF-IMA) [51]. Both frameworks refer to modeling the *socio-technical* systems compared to purely technical systems to support a data-centric model-driven approach for the entire life cycle of system development. Note that the use of term *socio-technical* indicates the acknowledgment of interaction between people and technology, especially concerning *modeling*. RF-IMA even involves the notion of an *actor* or a human (such as modelers and domain experts) with intention or purpose and envisages the actor's interaction with *multiple* assistants. On the other hand, MODA presents the interaction between data and purposively descriptive, prescriptive, and predictive models.

Our experiences are consistent with the vision laid out by these frameworks. Three out of five case studies that we present involve human-in-the-loop interactions (in more than

[2] Language Models are Few-Shot Learners https://github.com/openai/gpt-3.

one of the modeling activities in two case studies). As we will show, the key to ensuring that AI techniques are used where they are merited is to explore AI techniques for *each* modeling activity but in concert.

The following section covers the current landscape of AI-driven modeling spread across the modeling activities discussed above.

3 Related work

Interestingly, given the early stages in which AI-driven modeling research finds itself, no surveys or systematic literature reviews are available when writing this paper. Instead, most publications that do review the related work in AI-driven modeling have covered it as a part of long-term vision [24,25,51]. We attempt to cover most of the works that apply AI techniques to different modeling activities.

3.1 AI-driven domain modeling

Based on the idea that artifacts available and under consideration guide the application of specific AI techniques for domain modeling, we classify the domain modeling approaches into the following categories that use—(a) repositories of existing and related (meta-) models available, perhaps in a multitude of formats [5,69], (b) the statement of requirements or requirement document(s) or problem description document(s) [7,8,65], and (c) external knowledge source(s) to guide the selection of model elements [2,3], as a starting point. We review each of these briefly next.

3.1.1 Using repositories

The rationale behind such approaches is that properly maintained repositories are likely to contain relevant metamodels, class diagrams, ontologies, XML schema definitions, Resource Description Framework (RDF) documents in standardized specification styles [5]. Such approaches provide mechanisms to search repositories on the basis of synonyms and word senses by integrating with lexical databases like WordNet[3] as in [5,69] or ConceptNet[4] or by using cosine similarity measures based on various vector space models [65].

Solutions relying on repositories of (meta-) models tend to suffer from *cold start* problem. To build a domain model by reusing existing (meta-) models, a sufficient number of these need to be available to be useful [3]. Additionally, industry participation is needed in building and maintaining repositories, which is often lacking [24,51], thereby reducing the import of such approaches in industry settings.

3.1.2 Using requirements/problem description documents

These approaches rely on varied AI techniques such as parsing requirements documents using syntactic parsing with extraction rules [7,90] amended with classifiers for distinguishing between model elements such as classes or attributes [65].

Depending on the scope or length of the documents under consideration, approaches relying primarily on syntactic processing for domain modeling are liable to produce many false/superfluous candidates [7]. Such application may require further steps of pruning using additional classifiers with the domain experts' active participation to indicate positive and negative instances [8] or using reinforcement learning [14].

3.1.3 Using external knowledge sources

The semantic network approach SemNet[5] [2] enables mediator-based knowledge base querying in the *DoMoRe* tool [3]. *SemNet* is obtained by processing the Google Books n-grams corpus [32] to enable computing semantic relatedness of single and multi-word terms [3] using distributional semantics hypothesis [33]. SemNet contains binary noun-noun relationships, verbal relationships (how often noun terms cooccur with verbs), and ternary relationships (simultaneous occurrence of three technical terms) and, based on this, is capable of suggesting *contextually* similar terms for the varied combination of model elements.

Although other online sources such as Wikipedia and related sources Wikidata[6], DBpedia[7], and Yago[8] as well as WordNet and ConceptNet can also be used for getting suggestions with regard model elements, these lack precisely the context precomputation provided by SemNet.

An additional category of approaches, not yet prominent in the modeling community, enable building knowledge graphs by using above- mentioned online sources, especially related to Wikipedia, as well as other domain-specific sources, by leveraging category structure, infoboxes, and content available in Wikipedia [18,67,87,89].

In our experience, the availability of a specific kind of input material for building domain models matters significantly. In early exchanges between the solution provider and a customer in an industrial setting, requirement documents

[3] WordNet https://wordnet.princeton.edu/.

[4] ConceptNet https://conceptnet.io/.

[5] SemNet http://semnet.henning-agt.de/.

[6] Wikidata https://query.wikidata.org/.

[7] DBpedia https://www.dbpedia.org/.

[8] Yago https://yago-knowledge.org/.

or problem description documents are often not available; in many cases, the customer has an idea around an existing problem that they want to vet out. To proceed further in a meaningful manner, a domain model is often a necessary artifact that needs to be created despite the lack of any requirements or problem description documents and refined throughout the engagement. These realities often make a *combination* of approaches capable of using external sources of knowledge and capable of extracting domain models from available texts quite desirable.

3.2 AI-driven model specification

We classify and review AI-driven model specification approaches by which artifacts are available to conduct the specification and activities involved in the specification (including domain-specific languages and domain-specific modeling languages) rather than specific formats or formalisms arrived at during the activity. Accordingly, we classify AI-driven approaches for model specification as - (a) voice-driven specification approaches [17,47] and (b) collaborative specification approaches using social networks and chatbots [55]. We review these next.

3.2.1 Model specification using voice

Voice-driven modeling proposes to use speech processing and NLP to achieve context-specific modeling [17]. Primarily motivated to increase the efficiency of modelers and to enable modeling by persons with disabilities [47], it builds on similar efforts in voice-driven or vocal/spoken programming [86].

The critical target benefit of such efforts is that the corresponding tool is not bound to a particular modeling language if it is mappable to a specific metamodel. As speech recognition is itself evolving, voice-driven modeling efforts are in the early stage of development.

3.2.2 Collaborative model specification using chatbots

A prototype implementation of modeling chatbots called *SOCIO* works on social networks like Twitter and Telegram. The modeling chatbots can interact with users and interpret their chat/text messages to specify metamodels and models [55]. While the bots' ability to interpret model construction and update commands is currently limited, the approach convincingly takes a step toward exploiting the collaborative and ubiquitous nature of social networks by enabling assisted modeling by many participants, perhaps simultaneously.

The Xatkit chatbot development framework enables defining chatbots as well as voicebots in a platform-independent way [26]. The modeling infrastructure of Xatkit contains an intent package to describe user intentions, training sentences, information extraction, and matching conditions, and an execution package to define chatbot behavior [26]. Internally, it contains chatbot DSLs that provide primitives for design and deployment in addition to user intentions and execution logic. The construction of voicebots in Xatkit currently supports Alexa as a voice platform to capture voice input as text.

Note that specification using chatbots differs from crowdsourcing the specification elements as in [19]. Presumably, while the chatbots are likely to be made available to a close-knit community of people already aware of the specification language, crowdsourcing, by definition, involves anyone willing to contribute. It is additionally possible to combine the two, use chatbots to show specification elements and the related questions to the volunteers, and proceed similarly.

3.2.3 Other AI-driven model specification

Some other promising work in AI-driven model specification includes Optical Character Recognition (OCR) of DSLs [56]. Although the proposed approach faces challenges such as error-free recognition and the addition of domain-specific vocabularies to the pre-trained models, the approach hints at the possibility of benefitting from recognizing snippets of DSLs from conference proceedings, books, and in the long run from online presentation videos.

In our experience, such a requirement also exists for graphical modeling languages used in other domains such as refinery flow charts in materials engineering or hazard pictograms in the manufacturing domains.

Translating natural language requirements to a domain-specific language specification is a relatively unexplored approach in modeling but used in other communities such as NLP [85].

3.3 AI-driven model transformation

Nontrivial model transformations require a deep understanding of the source and target language metamodels and the model transformation language if one is used. Approaches such as [23,44] propose to use NLP and DL techniques to improve the model transformation process.

In an industry setting, our experience suggests whenever transformation is required, it is mainly carried out in a general-purpose language rather than a model transformation language; an observation also recorded in [22].

3.4 AI-driven model population

As indicated earlier, several industry scenarios require populating models from a wide variety of unstructured to semi- and structured documents; in many of these scenarios, both domain modeling and model population stages use such sources of information. This activity is also referred to

as *information modeling* especially, in the manufacturing domain, and in cyber-physical systems, and IoT systems and smart digital factories created for any business domain [37,40,57,58].

It is, of course, not restricted to these domains. Information modeling is equally prevalent in the medical domain—to answer controlled natural language questions over RDF clinical data [39], classification of biomedical documents using Wikipedia [31], and so on.

Information modeling is the choice of modeling for large enterprises looking to make the existing information functional and obtain and act on insights. In that sense, information modeling and its downstream usage could also be referred to as *model-driven information analysis*.

In such cases, the domain model is built manually, albeit with structured participation of domain experts [36]. The choice of specification in recent times has been *knowledge graph* [28,37,57,59]. Several challenges remain to be addressed in terms of extraction of facts, including semantic annotations [31], checking facts encoded in the knowledge graph for correctness and contextualization, and in terms of actual deployment [37].

The stored information is usually queried using graph queries which are manually entered. In some cases, (controlled) natural language queries can be generated and translated over the graph using a finite state machine over the underlying ontology [39].

3.5 Model-driven AI

Apart from using AI techniques in modeling, it is possible to apply modeling to AI techniques.

This set of works can be referred to as model-driven AI rather than AI-driven modeling. Examples include (1) decomposing ML into chainable microlearning units targeting cyber-physical systems and Internet of things (IoT) applications [34], (2) a proposal to extend the domain-specific language to be able to generate ML code, also in the domain of IoT [50], and (3) a metamodeling framework for meta-learning to enable the integration of ML into modeling frameworks to be used as black boxes (along the lines of AutoML [35]) thus mainly doing away with the need of expertise in AI techniques. Model-driven AI is out of the scope of this paper.

As we can observe from the related work, AI-techniques have penetrated every modeling activity but are seldom used in multiple modeling activities. Additionally, the use of diverse data prevalent in the industry seems somewhat restricted, save a few exceptions.

We present our case studies in the next section to demonstrate *AI-driven Streamlined Modeling* where we show that depending on the artifacts/data at the disposal, the application of AI techniques need not remain restricted in a single modeling activity. Also, depending on the purpose at hand, artifacts/data can be processed and represented to leverage AI techniques appropriately.

4 AI-driven streamlined modeling case studies

We present our case studies in a structured manner as follows:

- *Problem Context and Challenges* We describe the problem statement and challenges in the prevailing situation for which we created either a modeling solution that we enhanced with AI techniques or adopted an AI-driven approach in our modeling solution from the beginning.
- *Modeling solution and AI enhancements* We describe the modeling activities involved in the modeling solution followed by the AI enhancements as applicable to each activity with specific artifacts and AI techniques. We also show later in Sect. 5 how the cognizance of which AI techniques are applicable at a specific modeling activity can help the reader choose such techniques in modeling their problem context.
- *Comparison with AI-driven modeling work* Here, we compare and contrast our solution approach with the related work presented in Sect. 3. Later in Sect. 5, we show depending on the purpose at hand and the available artifacts, it may be possible to use AI techniques in concert for a given modeling activity.
- *Experiences and lessons learned* Here, we talk about our experiences with customers and lessons learned in enhancing the modeling solution with AI techniques.
- *Applicability and customer buy-in* In this section, we describe the applicability of the solution and customer buy-in received for it. We believe that recognizing specific domains where the given case study found buy-in can help the reader apply a similar solution to a similar problem in the same domain.

We use the words enterprises and companies interchangeably. We begin with the first case study next.

4.1 Regulatory compliance

Problem Context and Challenges Modern enterprises operate in a tightly regulated environment with rapidly changing regulatory requirements originating from emerging standards for transparency reasons and face hefty penalties for non-compliance. Compliance is, therefore, a top priority for enterprises and needs to be a swift response.

The prevailing state of the art and practice faced the following challenges:

– *Semantic disparity* The state of the art focused on using various formal languages to check process compliance. The state of the practice used manually operated Governance, Risk, and Compliance (GRC) frameworks. The key challenge for both was the semantic disparity or matching concepts and labels from regulations with concepts and labels in enterprises' operations and data.
– *Explanation of proof of compliance* pointed at the ability to prove and explain (non-) compliance.
– *Managing changes* in the regulations, as various standards continue to expand categories of regulated items and regulations thereupon.

Modeling Solution and AI Enhancements Our first baseline modeling solution consisted of manual modeling of the regulatory domain and constructing the Semantics of Business Vocabulary and Rules (SBVR) model for enterprise operations. We used DR-Prolog as the formal language of choice for the implementation of defeasible reasoning (rule-based approach to reasoning with incomplete and inconsistent information [15]). SBVR acted as a bridge between the regulatory domain model and the enterprise operations to tackle semantic disparity [76].

The first AI enhancement consisted of generating a natural language explanation of proofs of compliance as shown on the right of Fig. 1. First, we would obtain proof of compliance. For this, we came up with an algorithm to process a representation of the procedure box abstraction created using a Prolog meta interpreter that emitted a trace of rule invocations. This gave us the facts (operational details) that led to a specific regulation's success or failure. To obtain the explanation for the success or failure of facts, we created a lookup for each term/keyword in the SBVR model's *Business Vocabulary* body of concepts and its corresponding terminological representation in the *Terminological Dictionary*. For rules, we fetched the logical formulation of rules from *Business Rules Vocabulary* and obtained its natural language representation from its corresponding mappings in the *Terminological Dictionary*. For an elaborate discussion, we request the reader refer to [74] and [76].

In addition to the explanation of proofs, we created a model versioning mechanism for both regulations and operational details to tackle regulatory changes [75].

The second AI enhancement consisted of aided domain modeling as shown on the left of Fig. 1. We used a combination of relation extraction (based on fact orientation) using open information extraction (Open IE) [76], and clustering to enable the tool called *Concept Model Generator* (CMG) to aid the expert in building the domain model from the regulatory text(s). The expert could also provide feedback to the tool for extracting rule sentences from the regulations using active learning [78].

While the original version of the CMG needed seed concepts to start suggesting the next set of concepts and/or relations, we use a text ranking [49] implementation to compute top(k) terms in the most recent version to also suggest seed concepts. The text ranking implementation creates a text graph to compute top(k) terms. The expert then refines the baseline domain model like the previous version by modifying (adding/deleting) concepts and relations [78].

Prior to the third enhancement, we also created a specification language for the experts to encode regulations in Structured English (SE). The SE specification translated to the SBVR model (which was thus far being created manually). We also provided transformation of SBVR to Drools and Java in addition to DR-Prolog code generation. At this point, the expert could author the regulations in SE without worrying about other representations (SBVR and DR-Prolog/Drools) and manually map regulations to enterprise data.

The third AI enhancement consisted of using the domain model and rule sentence extractor (using active learning as noted above) and transforming sentences in English to Structured English suggestions. This process again used open IE and rule-based translation along with dictionary of domain concepts (in CMG) to obtain suggestions as shown in the middle of Fig. 1. For a detailed discussion, see [63].

Comparison with AI-driven Modeling Work Our domain modeling aid is similar to the category of domain modeling aids visited in Sect. 3.1.2. In comparison to those approaches which use syntactic parsing, we rely on open IE implementation (an open-source ML model by AllenAI bootstrapped from other relation extractors[9]), which we have found to be more robust for various kinds of syntactically complex sentences (as in regulatory documents). We use the CMG whenever aided domain modeling is necessary (as in Sect. 4.4).

Our aids in proof explanation generation using SBVR and Prolog meta interpreter and transforming English sentences to SE are unique to modeling literature covered in Sect. 3. However, some works in the regulatory compliance community have investigated approaches along similar lines (for comparison with these, see [63,74,76,77]).

Experiences and Lessons Learned Our key customers were not sold on a purely model-driven solution since they were using GRC frameworks anyway. However, with the three AI aids provided—(a) aided domain modeling that could potentially replace the expert-driven regulatory taxonomy construction, (b) easily learnable SE for the specification of regulations with suggestions, and (c) explanation and traceability of compliance, it was possible to draw interest from multiple customers.

[9] Open IE by AllenAI https://github.com/allenai/openie-standalone.

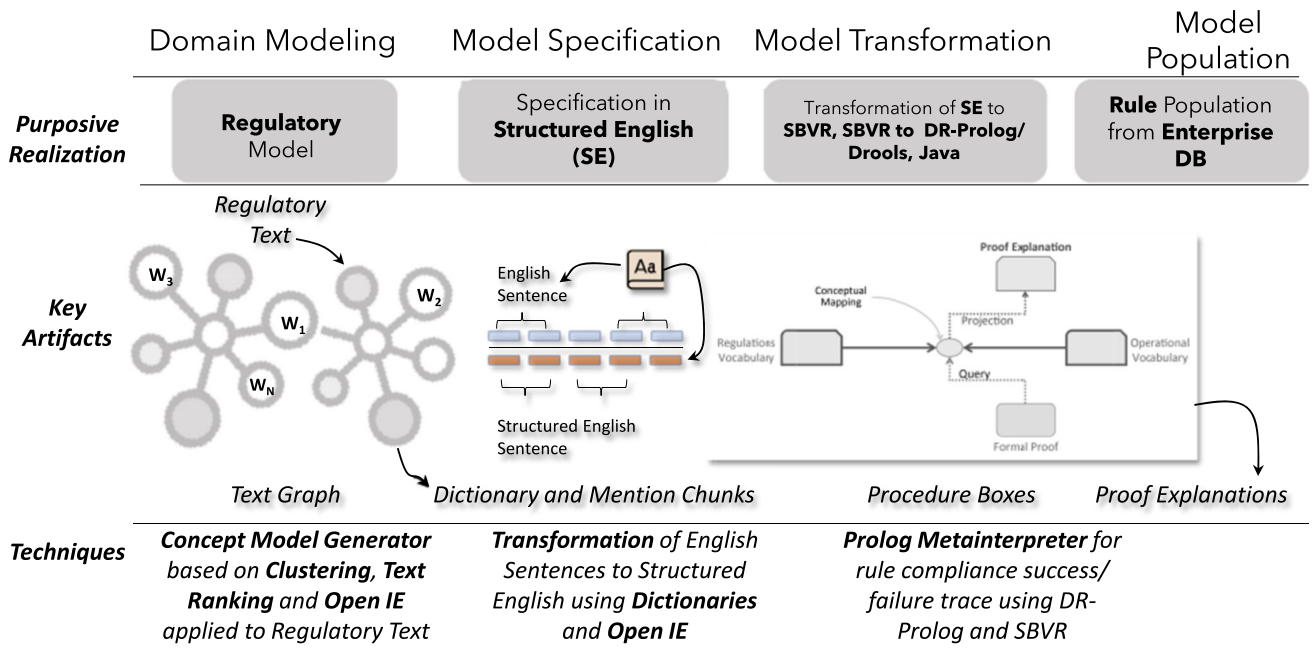

Fig. 1 Case Study 1: Regulatory Compliance

Customers found additional uses for the CMG in mapping regulation statements, company policy statements, and control implementation based on common mentions of concepts. Primarily started to address the banking and financial services domain, the AI-aided model-driven solution also found acceptance by medical equipment manufacturing companies on privacy regulations for data stored in medical equipment.

In general, we train the developers/modelers in the interested business unit in all three AI aids. This is also true of AI aids provided in other case studies.

Applicability and Customer Buy-in We have applied the AI-aided model-driven regulatory compliance framework to regulations such as KYC regulations[10] [75,76], MiFID[11] [41] and MMSR[12] [62–64]. Our customer interactions and buy-ins involve a Fortune 500 US insurance company, a large bank in Western Europe, a US multinational investment bank and financial services company, and a US medical equipment manufacturer.

4.2 Document generation and checking

Problem Context and Challenges Document intensive enterprise ecosystems rely on the generation and checking of documents governed by industry standards. Two examples of standards governing their respective enterprise ecosystem are *Uniform Customs and Practice (UCP) for Documentary Credits* and *Globally Harmonized System (GHS) for Classification and Labeling of Chemicals*. In more than 175 countries, banks and commercial parties use UCP in international trade finance [48,61]. GHS is adopted by all major countries and companies operating in handling, transport, and usage of especially hazardous chemicals have to adhere by it [54].

The key documents related to UCP are Letters of Credit (LOC), Bills of Lading (BOL), commercial invoices, and inspection certificates. Key documents for GHS are safety data sheets (SDSs) that contain both text and pictograms. Given the dynamic nature of international trade and chemical domains, the rules governing the generation and checking of documents change (in addition to their geography-specific constraints), requiring more efficient ways of handling changes.

The *document generation and checking* problem is different from the compliance checking problem. While compliance checking applies to business processes and data generated in the processes, the document generation and checking problem is purely document-oriented [54].

Modeling Solution and AI Enhancements Both in the case of UCP and GHS, we modeled the domains of international trade and relevant document structure as well as GHS and corresponding safety data sheets manually.

For UCP, we specified the domain using international trade ontology. Since the problem required language pro-

[10] RBI KYC- Know Your Customer Direction https://www.rbi.org.in/CommonPerson/english/scripts/notification.aspx?id=2607.

[11] MiFID- Markets in Financial Instruments Directive (2004/39/EC) https://www.esma.europa.eu/policy-rules/mifid-ii-and-mifir.

[12] MMSR- Money Market Statistical Reporting https://www.ecb.europa.eu/stats/money/mmss/shared/files/MMSR-Reporting_instructions.pdf.

cessing, in the case of UCP, we provided a way of extracting rule content from rules sentences governing LOC and BOL generation using syntactic processing [61] as an AI enhancement. Additionally, we provided a mechanism to transform the extracted rules to Semantic Web Rule Language (SWRL) rules to enable the checking.

In the case of GHS, which is a more recent case study, the rule statements are clearly stated in GHS governing document, and language processing is not needed for extracting rules. Instead, the challenge lies in extracting tables from the GHS documents in PDF format and processing both text and images (pictograms) to both generate and check the safety data sheets against GHS rules. This is illustrated in Fig. 2 (only GHS case is shown).

We parse the rules and store them in a graph database with the GHS domain model as its schema. SDS text is parsed similarly, and rule generation and checking take the form of rule graph traversal [54]. For change management, we use versioned graph representation [54].

The AI enhancement is realized in the form of a pictogram processing algorithm that extracts pixel representation and checks against existing SDSs to be checked. When generating SDSs for new chemicals, it inserts required pictograms based on the pixel representation matching [54].

For both table extraction and manipulation and pictogram processing, we use state-of-the-art APIs based on computer vision. We request the reader to refer to [54] for further details.

Comparison with AI-driven Modeling Work The work closest in comparison is using OCR to recognize DSLs [56] except that while it proposes to recognize textual DSLs, pictograms denote a visual DSL-like nature with specific meaning associated with the specific pictograms.

Experiences and Lessons Learned The need to generate, check and revise documents per standards or rules is quite prevalent in many domains. Apart from banking and financial services and chemical manufacturing domains discussed above, other domains like airlines use document generation extensively. For instance, the generation of flight tickets and cancellation and refund receipts are governed by geography-specific rules and concerns such as economy and business class tickets.

In all these domains, the modeling part of the solution helps scale the document generation, while the AI enhancements improve checking and revising documents.

In addition to generating and checking generated documents, we have also found that customers are interested in (semi-) automated optical recognition of blueprint-like documents specific to a given domain, for instance, a refinery flow chart that guides the next set of actions for a chemical plant company. Such documents contain graphical icons, directed arrows, and instructional text requiring both image and text processing.

Applicability and Customer Buy-in Customer interactions and buy-ins include a large US multinational investment bank and financial services corporation and a large US multinational chemical corporation.

The next two case studies use *information modeling* or model-driven information analysis for generating insights and recommendations as discussed earlier in Sect. 3.4. In both cases, the information stored in the knowledge graph (Sect. 4.3) and data frames (Sect. 4.4) is verified by the domain experts before using it.

4.3 Formulated product design

Problem Context and Challenges Formulated product industry is a multi-billion Euro industry with products ubiquitous in use (cosmetics, paints and coatings, pharmaceutical drugs, etc.). A formulation is a recipe of chemical ingredients processed through step-by-step processes. State of the art and practice rely heavily on experts to form new recipes. The details required for new formulation recipes reside in offline and online textual resources. Current manual formulation recipe generation incurs hefty costs to the industry and a lengthy time to market. A (semi-) automated aid to the expert is needed to generate new recipes much quicker.

Modeling Solution and AI Enhancements Our modeling solution uses a conceptual/domain model for formulated products prepared based on the structure of historical formulations [81]. We do not need to use CMG because the structure of a formulation (which is the same for all historical formulations) reveals the core concepts. This domain model forms the schema of the knowledge graph in which we store details of individual formulations. Sub-domains such as cosmetics, paints, and coatings, etc., can be easily accommodated.

Given that the key challenge was the processing of vast information available in offline sources such as handbooks of formulations and online sources such as specialized chemical websites, we took the AI-driven approach from the beginning as illustrated in Fig. 3. Recipes are instructional or imperative texts (meaning that grammatical subject is implicit) and use references to previously stated facts (problem known as *ellipsis* in NLP). We created an algorithm that uses open IE (or dependency parsing) along with the stack data structure to tackle both the above problems [81].

While handbooks contain recipes, they do not contain information on chemical ingredients, such as why they were used/ what their intended functionality in the formulation was. Information such as ingredient synonyms is also crucial to indicate the kinds of products where it is used (but appears using different names). We created web crawlers that accumulated such information from specialized websites by first starting on Wikipedia and then fanning out [66]. As an example, the reader is invited to refer to the links on *Cetyl Alcohol*, an *emulsifier* or a moisturizing chemical used

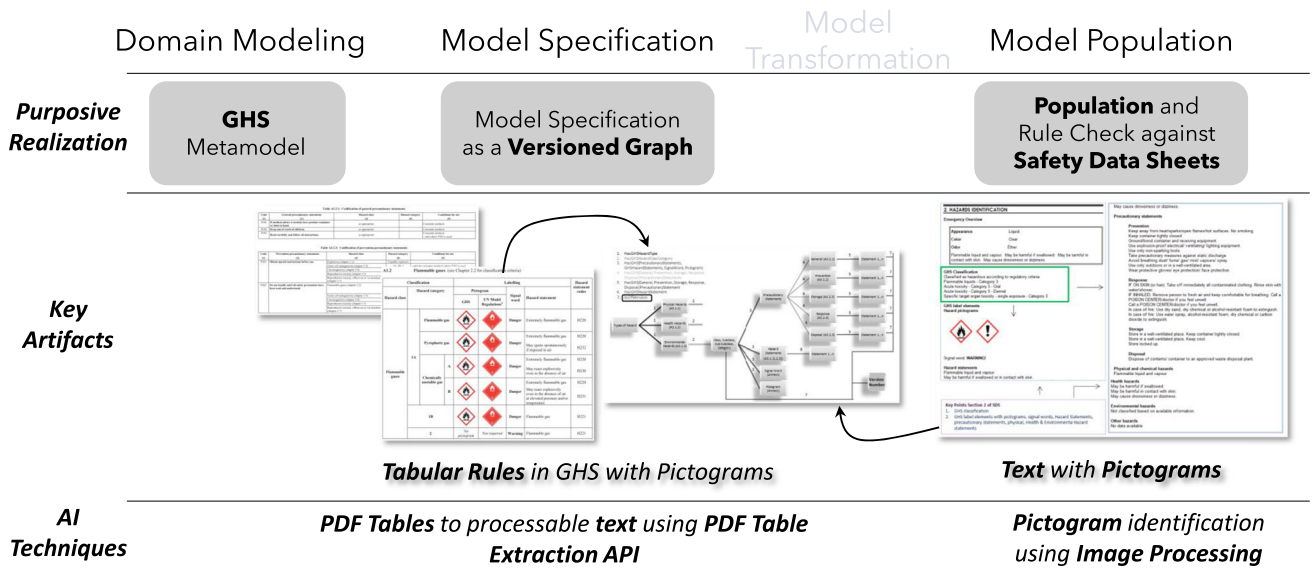

Fig. 2 Case Study 2: Document Generation and Checking

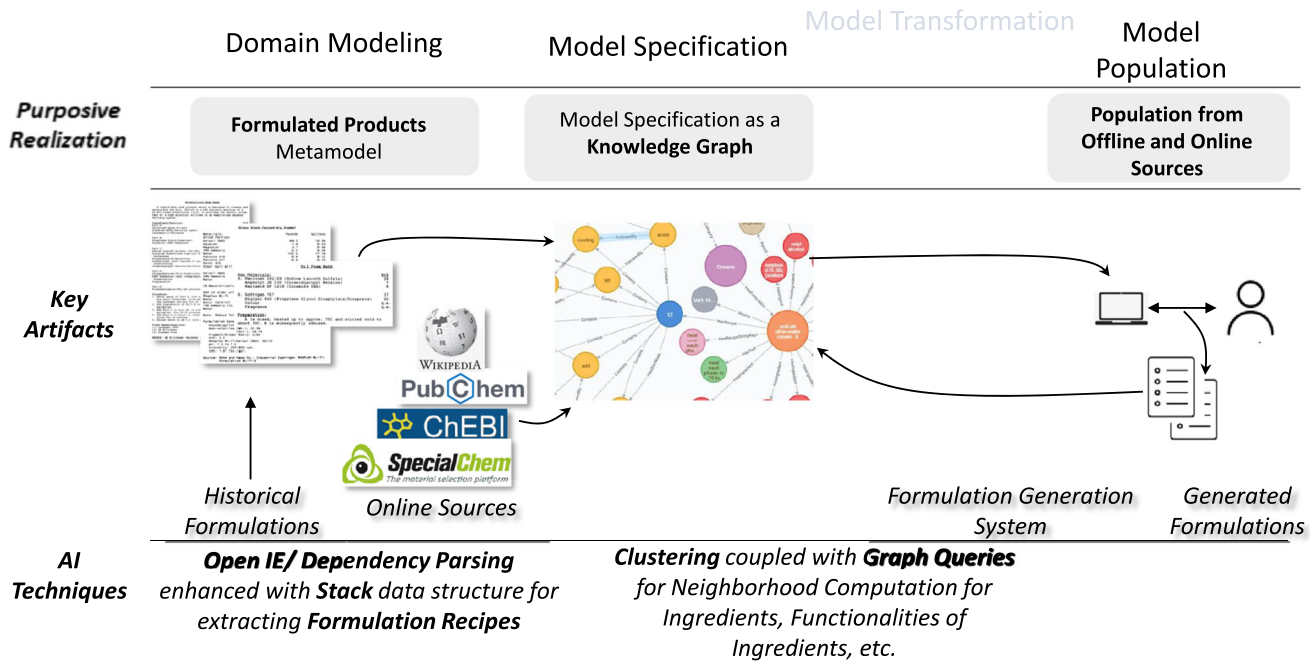

Fig. 3 Case Study 3: Formulation Recipe Generation

in cosmetic products at Wikipedia[13], PubChem[14], Chebi[15], and SpecialChem[16]. This is shown in Fig. 3.

Once we extracted and stored such details in the knowledge graph, we created several clustering analyses implemented as graph queries to aid the new recipe generation by expert. These analyses are executed by the recipe building/formulation generation system during the human-in-the-loop generation of new recipes. For a detailed discussion on how the expert interacts with the system and the system's analyses, see [81].

Comparison with AI-driven modeling Work This work falls in the area of information modeling or model-driven

[13] *Cetyl Alcohol* at Wikipedia https://en.wikipedia.org/wiki/Cetyl_alcohol.

[14] at PubChem https://pubchem.ncbi.nlm.nih.gov/compound/1-Hexadecanol.

[15] at Chebi https://www.ebi.ac.uk/chebi/searchId.do?chebiId=16125.

[16] at SpecialChem https://cosmetics.specialchem.com/inci/cetyl-alcohol.

information analysis. Compared to works in information modeling such as [31,39], we model the formulated products information. Traditionally, new formulation recipes are formulated by chemists, then made and tested in labs and factories. This approach is known as *generate, make,* and *test* [79]. Incidentally, in the continuation of current work on AI-aided formulation generation or *generate*, we are also working on transforming the currently manually operated *make*, and *test* steps to *digital make* and *digital test*, respectively. The use of robotically controlled labs and IoT-driven tests will lead to smart digital factory realization for formulated products with similar information modeling applied to digital *make* and *test* steps.

Experiences and lessons learned Formulated product industry has traditionally relied on experts. Only specially trained people get accepted for new formulation generation roles in products such as perfumes and flavors. Except for such products, there is a recent rush in digitalization in companies manufacturing other formulated products. Inundated partly by siloed data and partly due to substantial turnaround times, these companies are looking forward to solutions, such as ours, as discussed above.

Additionally, there is increasing demand to digitalize not just the *generate* step, but also *make* and *test* steps as discussed earlier. In all, such a realization would constitute a digital twin, with a long-term vision utilizing predictive models with regards to interaction between ingredients with specific functionalities. We have started working on an early design of a model-driven framework for this purpose [83].

Applicability and Customer Buy-in As indicated earlier, the system is extensible to any formulated product (individual types such as cosmetic products have more than ten subtypes), such as paints and coatings, food flavors, fuels, fuel additives, construction materials, and medicine and pharmaceutical products.

Customer interactions and buy-ins include a large US company involved in manufacturing of consumer goods, including cosmetic products, a Fortune 500 US company and global supplier of paints, coatings, specialty materials, and an Indian multinational steel-making company.

4.4 Information to insights framework for legal cases

Problem context and challenges Many countries maintain the judicial data for legal cases. Consider that people find themselves involved in a legal case as appellants or defendants say in civil law cases such as parental alienation or divorce. Also, assume that they attempt to find out what happens in such cases by looking at the case data. It is most likely that they are better off seeking legal advice rather than deciphering the data. The legal professionals also depend on their exposure to such cases or try to get an idea from available sources. Data exists, but it is hard to get insights from it and make recommendations.

Modeling solution and AI enhancements We approached this problem by manually creating a metamodel of legal cases that contains abstract concepts like parties, facts of the case, statement of appeal, verdicts, and reasons for verdicts. Any legal case is likely to have specific instances of these concepts [80]. For cases such as parental alienation (a major concern in Western Europe), we obtain the geography-specific open case data[17]. We apply the CMG (discussed in Sect. 4.1) to obtain the concept/domain model of the legal sub-domain such that it conforms to the legal case metamodel. Using CMG, the domain expert can relate mentions (synonyms, instances, indicator words or patterns, etc.[18]) to each concept and create what we refer to as *pattern dictionaries*.

At the same time, using the legal case metamodel and the concept model of the legal sub-domain, we create a set of *wh* questions [80], along with answer options for the users (people involved in the case, including the legal counsel) to construct user profiles as shown in Fig. 4.

When the user selects specific options for various questions, we generate statistical counts across the case data for specific situations covered in the options using sentence parsing and pattern matching using sentences and patterns stored in data frames. A data frame represents a two-dimensional data structure with the ability to store and analyze heterogeneous tabular data[19]. These insights are filled into a recommendation template per the user choices leading to user profile-specific recommendations. The recommendations convey the kinds of actions/precautions that the user can take to maximize their chances of favorable results [68].

For results and validation on parental alienation cases and divorce cases, we request the reader to refer [68].

Comparison with AI-driven Modeling Work We use the CMG for creating a conceptual model of the legal sub-domain under consideration. Note that the data we had at our disposal was in the Dutch language. Our CMG implementation uses the latest pipeline that is capable of dealing with various language models, including the Dutch language model[20]. We experimented with the original Dutch text as well as Dutch text translated to English using Google translate (several APIs are available to use Google Translate[21]) and found comparable results. While dealing with

[17] An example of a case file in Dutch Civil Court https://uitspraken.rechtspraak.nl/inziendocument?id=ECLI:NL:GHAMS:2019:44.

[18] CMG deliberately takes such a broad-based approach as explained in [77,78].

[19] Pandas Data Frame https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html.

[20] See Spacy Dutch language model https://spacy.io/models/nl.

[21] Example Google Translate API https://github.com/matheuss/google-translate-api.

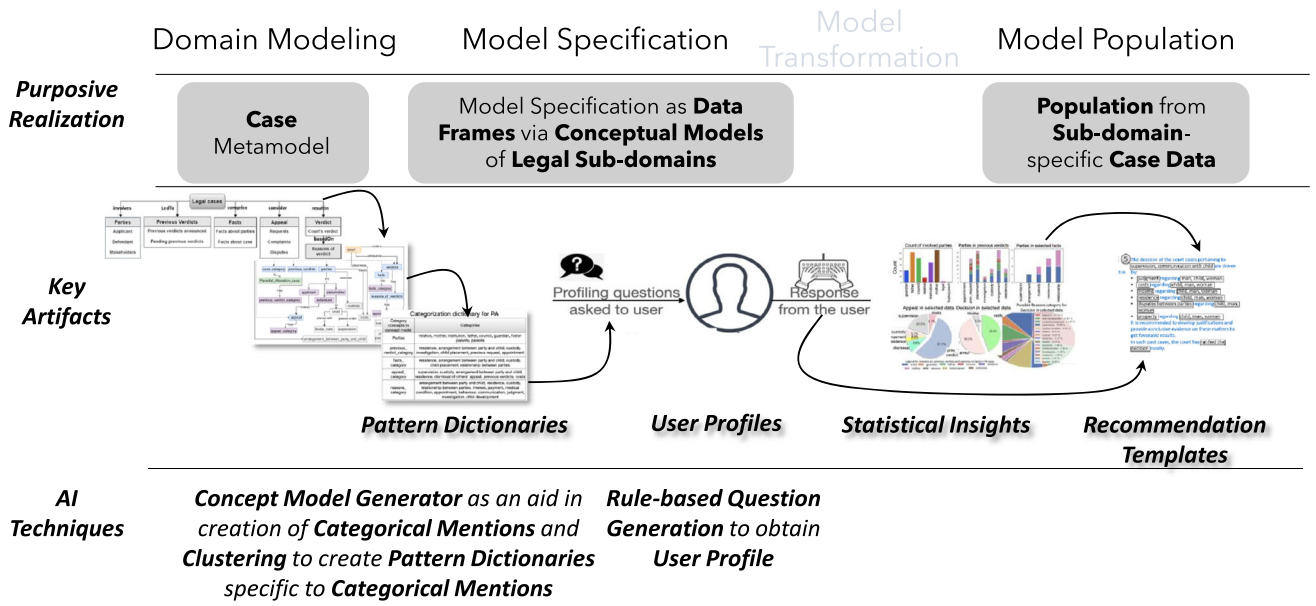

Fig. 4 Case Study 4: Insights and Recommendations for Legal Cases

non-English languages is quite common in the ML community, it seems to be less explored in the modeling community, especially by the domain modeling works.

Experiences and lessons learned We have seen the need for insight generation and recommendation from multiple domains in recent years. Examples include financial insights from annual and quarterly reports, and drug efficiency insights from the clinical study reports on clinical trials, to name a few.

We have observed some common elements in these situations as follows-a) these documents contain (un-/semi-) structured text, and companies often employ experts to derive insights and recommendations b) there is a set of stakeholders (user profiles) for whom recommendations need to be put together c) the complexity of the domain and availability of data determine the granularity in recommendations. With the case study we presented here, we continue to explore the AI-aided model-driven way to tackle such problems.

Applicability and customer buy-in This case study points to a class of problems in which company-, domain-, and geography-specific information is available in the form of (un-/ semi-) structured text from which insights and recommendations are needed. Traditionally, human experts have addressed such problems due to multi-level comprehension and classification of the subject matter. Accordingly, our work sees buy-in from several interested companies, in most cases, as an aid to the experts already employed.

4.5 What-if analyses for enterprises

Problem context and challenges Modern enterprises are essentially *system-of-systems* [71,73]. The cost of an incorrect decision in a sub-system can be prohibitively high for the enterprise. It is possible to represent enterprise goals and directives atop a model of enterprise and conduct various what-if analyses to determine which course of action is most suitable.

Modeling solution and AI enhancements Our modeling solutions for what-if analyses for enterprise evolved from impact analyses over Enterprise Architecture (EA) models [71], to extending the EA models with *intentional* elements [72] for static what-if analyses and mapping to *system dynamics* elements for dynamic what-if analyses [60].

As shown in Fig. 5, given the distinction between macro/aggregate behavior and micro/local and emergent behavior and the need to learn from alternate courses of action at the micro-level, another group of researchers from our organization built an actor-based enterprise simulation approach [12]. In this realization of enterprise simulation, the actual or real enterprise provides the data for virtual enterprise simulation based on key performance indicators to achieve the real enterprise's goals.

The enterprise simulation can use AI techniques like Reinforcement Learning (RL) for problems such as supply chain replenishment for a grocery retailer under varying constraints [10]. The retailer has a network of warehouses served by a fleet of trucks for moving products. The retailer needs to regulate the availability of the entire product range in each store subject to the constraints imposed by available stocks, labor

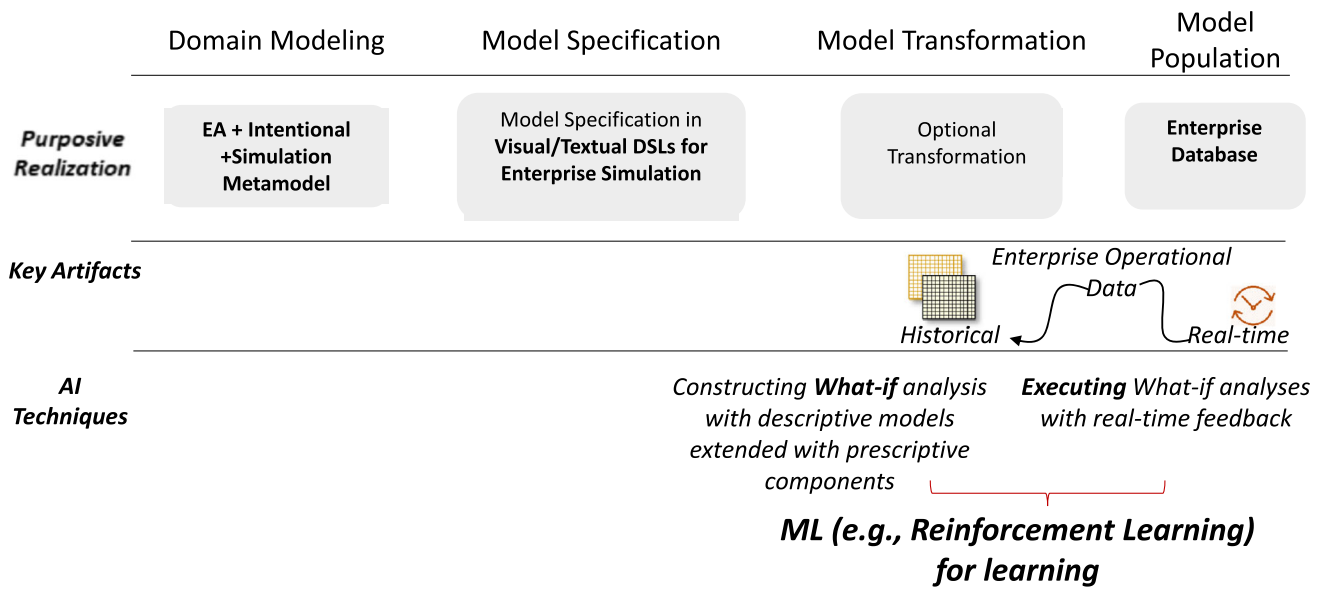

Fig. 5 What-if analyses for enterprises with predictive component

capacity, truck capacity, transportation times, and available shelf space for each product in each store [10].

The actor model of the retailer was constructed manually via interactions with the client stakeholders. The cognification applies to model population activity via policies learned by the RL agent as shown in Fig. 5.

The RL-based model-driven approach contains an RL agent (controller) and two control loops. The model-driven simulation loop helps train the RL agent and evaluate new policies before implementing them in the real system (retailer). The real-time control loop controls the real system using the trained RL agent. Essentially, *the RL agent learns how the environment operates and conducts what-if scenarios to maximize the discounted long-term reward*. The reward is a function of actions (defined in terms of replenishment quantities), and the inventory status [10]. The inventory status comprises the number of products that remain available throughout the stipulated time and the wastage of any products within the same time.

The model-driven simulator consumes an action that the controller produces as an external event and derives its impact by computing the state and rewards when a specific action gets executed in the actual system. To evaluate the performance of the RL agent, the reward is compared with a simplified version of an industry-standard replenishment heuristic, which aims to maintain the inventory levels of all products at a constant level [10]. Once the RL agent is trained, the actions rolled out to the actual system show substantial performance gains [12].

Regarding the earlier discussion on micro and macro behaviors, in the RL-based simulation used in the retailer case study, the retailer's micro-behaviors are used to compute emerging macro behaviors. We request the reader to refer to [12] for the metamodel of a complex system of actors as well as the formal representation of the RL agent's interaction with the real system. Additional application of actionable what-if analyses enabled using RL includes scenario playing for research ranking improvement for a university [11], and a nontrivial exploration of localized non-pharmaceutical interventions to control COVID-19 [13].

Comparison with AI-driven Modeling Work In our opinion, the AI enhancement discussed above using RL is different compared to modeling related work such as [14] covered in Sect. 3, mainly because the current modeling research has less exposure to enterprise simulation, control systems, and applications. However, given the rise of modeling in cyber-physical systems and IoT, we think that investigation along similar lines as described above may prove quite helpful.

Experiences and Lessons Learned In this case study and the problem it tackles, often the fidelity of the virtual enterprise is questioned. By using reinforcement learning, the agent-based simulation serves to provide the reward function. Along with the rest of the architecture as presented by Barat et al. [12] that includes capturing uncertain events probabilistically, it is possible to improve the realism of the objective (in the specific example, supply chain replenishment).

Applicability and Customer Buy-in For the actor-based enterprise simulation with reinforcement learning, customer interactions and buy-ins involve a supermarket chain, a large postal company, and a telecommunications company in Western Europe.

Table 1 Artifacts and AI Techniques in Modeling Activities; RW—Related Work in Sect. 3, CS—Case Studies in Sect. 4

		Domain modeling	Model specification	Model transformation	Model population
Artifacts (+ intermediate representations)	RW	requirement documents/ problem description documents [7,65], repositories of (Meta-) models/ ontologies [5,69], external knowledge sources [2], dependency graphs [7], Tagged text [65], relational and graph databases [2]	chat/text messages [55], voice commands [17,47], textual and visual DSL scans [56]	repositories of (Meta-) models [23,44]	sensor data [37], process data [59], clinical data [39], multi-lingual bio-medical data [31] graph databases [37,57]
	CS	regulatory documents [75,77], standards and generated documents [54], legal case texts [80]	rule sentences [63]	SBVR models [75]	historical formulations, Wikipedia + specialty sites [66,81]
AI Techniques	RW	lexico-syntactic pattern matching [2,7], co-occurrence analysis [2], syntactic parsing and rule-based processing [7] classification with text vectors [65]	speech recognition and synthesis [47], NLP/NLU [17]	classification of (meta-) models using deep NN [23]	semantic annotations+ cross-lingual concept matching [31], finite state machine over ontology [39]
	CS	clustering [77] open IE [77], classification using active learning [78]	NL to SE transformation using domain dictionaries and open IE [63], NL to SWRL using ontology and dependency parsing [61]	NL explanation of proofs of compliance using Prolog meta interpreter and SBVR [74]	open IE [81], dependency parsing [81], pattern matching [80] clustering analysis [81], image processing [54], rule-based question generation [80]

5 Discussion

5.1 Artifacts and AI techniques in modeling activities

Table 1 presents a consolidated view of artifacts and AI techniques seen in various modeling activities, both in our case studies described in the previous section and the related work discussed earlier in Sect. 3. The header row shows the modeling activities. The artifacts row lists artifacts and intermediate representation and divides them into the modeling activities used in the case studies and the related work. The AI techniques row similarly lists and divides AI techniques into modeling activities.

Such clustering of artifacts and AI techniques immediately reveals alternate AI techniques that might apply to artifacts listed in the same modeling activity. For instance, the lexico-syntactic pattern matching technique in [7] has been used for processing requirement/problem description documents for domain modeling. But it is likely applicable to external knowledge sources like Wikipedia as well (listed in the domain modeling column and artifacts row in Table 1). It turns out to be the case since there exists work in AI literature (but not in the related work or our case studies) that uses this technique to acquire hypernym-hyponym relations from Wikipedia (such as [53]).

By checking the column specific to a modeling activity in Table 1 and depending on the available artifacts, the reader may refer to the corresponding AI techniques and set up experiments to determine if any techniques referred can be used for their problem-solving. We list some such possible ways per modeling activity from Table 1 based on our experiences.

Domain modeling If requirement documents or problem description documents available to the user use a predefined structure or a set of syntactical patterns, then dependency and pattern-based syntactic analyses as in [7] are appropriate to discover candidate concepts from such documents.

On the other hand, suppose the available data includes texts with complex sentences (containing multiple clauses, bulleted lists, cross-references to various sections, definitions, annexures, etc., such as seen in legal text, user manuals, knowledge guides, and handbooks). In such cases, clustering-based approaches [77] or text ranking approaches are more suitable for discovering candidate concepts compared to purely syntactic approaches.

If only a problem statement is available but no documents to work with, then one may use an approach like SemNet [1,2]. Such approaches that leverage open-domain data such as Google Books corpus and Wikipedia are exploratory by nature and enable growing a domain model from seed concepts by following the related terms from the knowledge bases.

A relatively unexplored area of cognification in domain modeling is discovering candidate concepts from informal texts and colloquial language artifacts (such as speech-to-text transcripts, tweets). Such an effort may require exploring different techniques not covered in the related work or case studies, such as processing lexical and acoustic-prosodic information in the speech [84], and using extended dependency parsing techniques for tweets [46].

Model specification and transformation It can be observed in Table 1 that the nature of artifacts used and created as well as the applicable techniques in these activities differ substantially from the domain modeling activity. There is a natural shift towards alternate ways of model specification such as chat messages [55] and voice commands [17,47] and techniques such as speech recognition [17], natural language understanding and collaborative specification [55], optical character and image recognition [56], etc.

Although we have not used such techniques in the case studies, our interactions with the clients indicate a substantial buy-in for the ease and facility such techniques introduce. For instance, a Telegram chatbot is available on any device/platform and enables collaborative input [55]. In nontrivial use cases, such implementations would necessitate proper authorization and model update mechanisms but still present exciting possibilities.

Some additional ways to explore these activities could be using a combination of artifacts and techniques. For instance, instead of typing chat messages in a chat window (such as shown using Telegram [55]), it might be possible to use voice commands as the input mechanism. Using speech recognition and natural language understanding as proposed in [17], voice commands can be captured and sent as a text to the chatbots. The Xatkit chatbot development framework indeed provides support for such manner of interaction [26] via voice platforms like Alexa. One important concern here is that Alexa's skill interaction models and voice and intent recognition services are available only via the Amazon cloud [70]. Our suggestion applies to text-to-speech technologies in general in addition to using voice platforms like Alexa.

Another possibility is to use the ability to write/draw by hand on touch screens using a stylus or digital pen and capture the input as part of the specification activity. Using a stylus to write or draw is referred to as inking, and the research area that explores such activity is known as *pen computing* [6]. Pen computing research suggests that a stylus allows users to ink with more fluidity and naturalness than a mouse or keyboard, which utilizes indirect input interaction. Like OCR and voice recognition, pen computing has its own challenges, such as palm rejection [6], but still presents an exciting avenue for (model) specification activity.

Interestingly, it seems possible to use these techniques in the domain modeling activity as well. For instance, it may be possible to use a DoMoRe-like domain modeling recom-

mender system [3] that the user interacts with via a chatbot interface and/or voice commands. Although such a combination has not been explored in the AI-driven modeling literature, it does present an exciting new avenue.

Some recent works explore model-driven optimization, a combination of search-based and model-driven engineering for solving optimization problems. These works enable domain-specific formulation and specification of optimization problems via DSLs and search space exploration via model transformation rules [38]. The encoding of the solutions is either model-based, i.e., using models to represent candidate solutions [21], or rule-based, where the solutions are represented as sequences of transformation rule applications [16]. Objectives to be optimized may be defined using evolutionary algorithms such as genetic algorithms or reinforcement learning techniques [29]. These approaches investigate how model-driven design, specification, and transformation can make optimization more easily accessible to a wider audience.

Model population Table 1 shows that the widest variety of NLP and ML techniques are present in the model population activity owing to the widely varying nature of data available, both domain-specific and open-domain. Other techniques such as named entity recognition, entity linking, entity disambiguation, and relation filtering may be relevant in this step [87].

Entity linking means linking an entity/mention in the model to an entity in knowledge bases. We demonstrate a form of entity linking, in case study 3 (in Sect. 4.3) and detailed in [81], when we connect ingredient names to the information present on the respective specialty sites, which can be considered as knowledge bases. Entity disambiguation, distinguishing the same mentions of different entities, and filtering non-relevant relations are beneficial when open-domain knowledge bases such as Wikipedia are used as the only/primary source of information [87].

The basic ideas in such domain modeling techniques as SemNet [2] apply to the model population activity also. For instance, if extensive text material is available in a given problem setting, one may create a domain-specific semantic network. Instead of using the Google Books corpus as in [2], one may create a corpus of relevant books, papers, articles, etc., and apply the same steps as in [2] to the n-grams of this dataset and obtain a semantic network for the given domain[22]. Such a network can help in modeling information of the given domain to create and maintain a domain model and populate it as a knowledge graph.

[22] An example of such an alternate dataset is the COVID-19 Open Research Dataset at https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge, which has been prepared using over 400,000 scholarly articles, including over 150,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses.

5.2 Models and data

As discussed earlier in Sect. 2, the recent frameworks denote a shift in focus from modeling—in RF-IMA, it is signified by the shift from modeling to modeling assistants [51] and in MODA, by the shift from models to models *and* data [25]. In particular, we believe that MODA's data-centric approach is consistent with our experience (also in all the case studies discussed).

Interestingly, the discussion and examples in MODA focus on representing the *descriptive*, *prescriptive*, and *predictive* nature of models and the interaction of such models with the data but do not discuss specific modeling activities where such interactions may take place. In such cases, to aid the reader make sense of in which modeling activities do the *descriptive*, *prescriptive*, and *predictive* models interact with data, we present the MODA representations of all the case studies in Fig. 6.

Figure 6a shows the MODA framework itself. In Fig. 6a, arrows labeled A and B indicate input and output processing. The C arrow represents the collection of metadata or metrics about the running system [25].

The arrows D and E in Fig. 6a indicate generalization to yield a descriptive model (such as a domain model) and predictive model building, respectively. The F arrow represents decision support activities like what-if analyses (and updates to the prescriptive model). The arrows G and H indicate software development activities and deployment, respectively. The I arrow represents the enactment of actions in the system [25].

In addition to the arrows described [25], we also show human interactions in individual case studies in Fig. 6b–f.

In the following, we interpret Fig. 6b–f as representations of case studies 1—respectively.

1. Figure 6b shows that the data under consideration is regulatory texts and enterprise database. The regulatory compliance system also contains the CMG. We show two distinct human interactions in Fig .6b—(1) the interaction by the domain expert building the domain model with CMG to create the regulatory domain model, (2) the interaction by the rule author who uses the SE editor to author regulations in SE. The regulatory domain model, the SE rules, and the corresponding SBVR model represent the *descriptive* model. The rule language code generated using these represents the *prescriptive* model reporting the enterprise of compliance successes and failures.
2. Figure 6c shows that the data under consideration is a set of documents (both the standards and the generated documents). The versioned graph acts as both a *descriptive* and a *prescriptive* model to achieve document generation, checking, and revision.

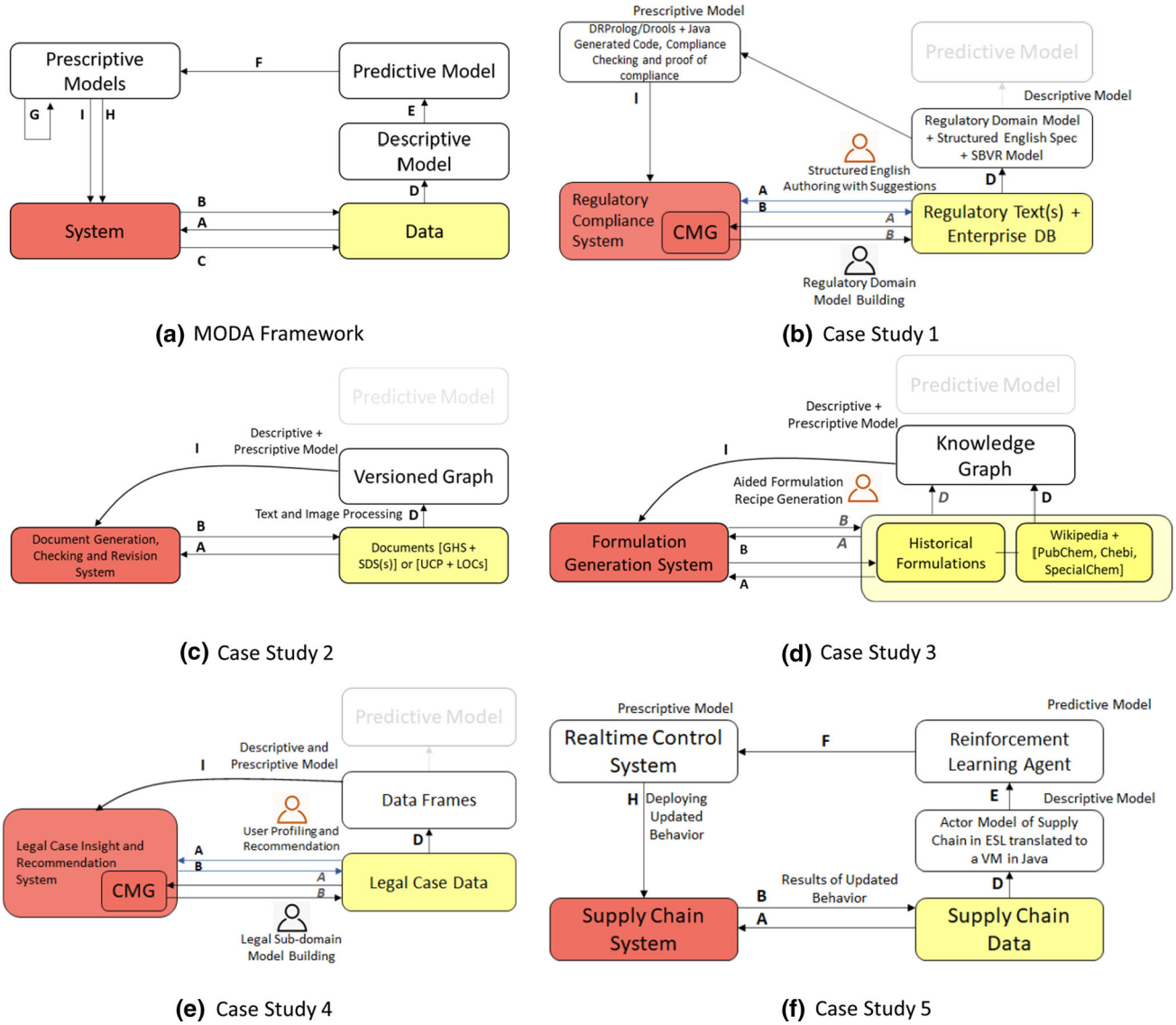

Fig. 6 **a** MODA Framework [25] (**b**–**f**) Case Studies 1–5 as Instantiations of MODA Framework; MODA Arrows: A- Inputs, B, C-Measurements, D-Descriptive Model, E-Predictive Model, F-Decision Support Activities, G-Generation, H-Deployment, I-Enactment

3. Figure 6d shows that historical formulations and associated information online is the data under consideration. A knowledge graph is constructed as a *descriptive* model. The interaction of the expert with the formulation generation system to generate new formulations treats the same knowledge graph as a *prescriptive* model.
4. Like Fig. 6b, Fig 6e shows two distinct user interactions with the legal case insights and recommendation system. The first interaction is that of the legal expert with the CMG to generate legal sub-domain concept models. The second interaction is in user profiling and recommendations to the user by the system based on insights computed using data frames. The *data frames* perform the roles of *descriptive* and *prescriptive* models.
5. In Fig. 6f, we show the use of a *predictive* model such as using a reinforcement learning agent in enterprise simulation. ESL is an actor-based simulation language targeted at a virtual machine in Java. As discussed earlier, the trained agent provides policies or behavioral predictions which the control system deploys in the real enterprise [12].

For modeling to transition to MODA-like interpretation and implementation, the community needs to adopt AI techniques in activities where they are relevant with data/artifacts that are available. Our case studies and the representation of the case studies as MODA instantiations can help corroborate the traditional modeling activities enhanced with AI

techniques and performing roles of *prescriptive*, *descriptive*, and *predictive* models.

5.3 Generalizability

Our case studies describe patterns of how various modeling activities can be amenable to specific AI techniques depending on the available data. As mentioned earlier, ours is a perspective, and an account of the successful use of AI techniques in modeling lifecycle applied to varied business domains rather than an approach or a method. As such, we argue that the generalizability of our account applies to individual case studies.

For instance, the answer to the question "*does the use of various AI techniques and models of compliance help intelligently automate various compliances in a given business domain such as banking and financial services?*" is empirically positive [63,64,75]. As a matter of fact, this question is essentially two questions rolled in one. The first question is, "does the use of *modeling* help automate compliances in a business domain such as banking and financial services?", to which we have a positive answer presented in [75]. The second question is, "if compliance modeling were to be enhanced with AI techniques, would it help *intelligently automate* compliances in a business domain such as banking and financial services?", to which also we have a positive response as detailed in [64,74,77,78]. Similarly, for other case studies, we have positive responses to these questions described in the related work for each case study.

As noted in [24], cognification of modeling is the use of knowledge (AI techniques, as well as techniques such as crowd-sourcing [24]) to enhance and boost the performance of modeling. Therefore, regarding all the case studies, one may ask questions of generalizability regarding modeling and AI enhancement, i.e., whether modeling improved the initially/predominantly *manual* problem-solving? And whether the use of AI techniques in modeling activities improved the solution compared to using modeling alone?

Beyond the applicability to other similar problems within a specific business domain, our case studies, as presented in this paper, also contribute uniquely compared to the recent efforts of aligning AI techniques in modeling lifecycle. For instance, while works on RF-IMA focus on better understanding, comparison, and selection of existing and future IMAs by creating a reference framework and discussing a likely set of properties to be considered [51,52], it does so without offering concrete instances of the complete framework. The work on MODA aptly shows several instances of the MODA framework (see Fig. 2 in [25]). Still, it discusses these examples at a level of descriptive, prescriptive, and predictive models rather than modeling activities and AI techniques. In our view, our presentation of the case studies (i.e., concrete examples) can help new and experienced practitioners of modeling and AI (in the context of this paper, AI-driven modeling) piece together modeling concepts with AI techniques via artifacts/data. It can also help to make sense of promising new integration views such as MODA, as shown in Sect. 5.2.

5.4 On the use of specific AI techniques

In our case studies, we have attempted to show the transition from purely manual to model-driven to AI-driven modeling solutions. Regarding specific AI techniques we used in the cased studies, the related publications cited in the previous section for each case study described in detail why we used those techniques (as opposed to other possible techniques).

The more valuable and critical lessons that we learned are the necessity to analyze data before constructing a solution and using several relevant AI techniques and technologies before adopting any of them to production. AI literature refers to such analysis as Exploratory Data Analysis (EDA). Common goals of such analysis are profiling of artifacts/data, formation of hypotheses, and testing [88].

In *AI-driven streamlined modeling*, such analysis would also consider the modeling activities. One key aspect of implementing EDA is to avoid the ad hoc application of AI techniques. A classification of artifacts and AI techniques in various modeling activities, such as shown in Table 1, can help to stay on the course and benefit from such analysis as described below.

- *Profiling artifacts/data* In this step, the practitioner can get a feel for the structure of the data, the kinds of constructs that it contains, and the complexity of constructs. Our experiences in the presented case studies suggest that it is better to get acquainted with the available data and its characteristics before forming and testing any problem-specific hypothesis [4]. In the context of *AI-driven Streamlined Modeling*, such analysis and interaction would form the notion of the models required for the given problem and the AI techniques available to enhance various modeling activities. Such analysis often leads to discovering interesting aspects of the artifacts followed by checking this understanding with the stakeholders to generate relevant hypotheses/questions/possible directions of investigation for a modeling solution enhanced with AI techniques.
- *Using and comparing various AI techniques* In the second step, the practitioner may attempt several AI techniques relevant to the nature of the problem in various modeling activities to see which techniques perform well concerning agreed-upon metrics in the exchange with the client/stakeholders in the previous step.

The implementation technologies of AI techniques may be discovered as a significant concern during such analysis. The licensing, versioning, interoperability, hardware requirements, and scalability of the implementation technologies are critical concerns, especially in industry settings, and may rule out using popular or standard AI techniques due to such limitations.

The nature of available data, AI techniques and implementation technologies available at the time, and business requirements for modeling, together tend to drive the exploration, hypothesis formalization, and testing in *AI-driven Streamlined Modeling*. Knowing the specific class of AI techniques relevant to artifacts available in a modeling activity, in our view, can significantly benefit the practitioners.

6 Conclusion

Recent research suggests that MD* should adopt AI- and data-driven approach to find better adoption in the industry. We presented five case studies with a perspective called *AI-driven Streamlined Modeling*, in which we described the modeling solutions and AI enhancements along with comparative applicability. Our case studies, from multiple domains, show examples of prevalent artifacts and data available at specific modeling activities and which AI techniques to apply to them. We believe that the modeling community and practitioners can benefit from the examples by relating their problem context with our perspective and improving the use of AI techniques and thereby helping better adoption of modeling.

References

1. Agt, H., Kutsche, RD.: Automated construction of a large semantic network of related terms for domain-specific modeling. In: International Conference on Advanced Information Systems Engineering, Springer, pp 610–625 (2013)
2. Agt-Rickauer, H., Kutsche, RD., Sack, H.: Automated recommendation of related model elements for domain models. In: International Conference on Model-Driven Engineering and Software Development, Springer, pp 134–158 (2018a)
3. Agt-Rickauer, H., Kutsche, RD., Sack, H.: DoMoRe? a recommender system for domain modeling. In: Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development, pp 71–82 (2018b)
4. Alspaugh, S., Zokaei, N., Liu, A., Jin, C., Hearst, M.A.: Futzing and moseying: interviews with professional data analysts on exploration practices. IEEE Transact. Vis. Comput. Gr. **25**(1), 22–31 (2019). https://doi.org/10.1109/TVCG.2018.2865040
5. Ángel, M.S., de Lara, J., Neubauer, P., Wimmer, M.: Automated modelling assistance by integrating heterogeneous information sources. Comput. Lang. Syst. Struct. **53**, 90–120 (2018)
6. Annett, M.: (digitally) inking in the 21st century. IEEE Comput. Gr. Appl. **37**(1), 92–99 (2017). https://doi.org/10.1109/MCG.2017.1
7. Arora, C., Sabetzadeh, M., Briand, L., Zimmer, F.: Extracting domain models from natural-language requirements: approach and industrial evaluation. In: Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, pp 250–260 (2016)
8. Arora, C., Sabetzadeh, M., Nejati, S., Briand, L.: An active learning approach for improving the accuracy of automated domain model extraction. ACM Transact. Softw. Eng. Methodol. (TOSEM) **28**(1), 1–34 (2019)
9. Aßmann, U., Zschaler, S., Wagner, G.: Ontologies, meta-models, and the model-driven paradigm. In: Ontologies for software engineering and software technology, Springer, pp 249–273 (2006)
10. Barat, S., Khadilkar, H., Meisheri, H., Kulkarni, V., Baniwal, V., Kumar, P., Gajrani, M.: Actor based simulation for closed loop control of supply chain using reinforcement learning. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pp 1802–1804 (2019a)
11. Barat, S., Kulkarni, V., Clark, T., Barn, B.: An actor based simulation driven digital twin for analyzing complex business systems. In: 2019 Winter Simulation Conference (WSC), IEEE, pp 157–168 (2019b)
12. Barat, S., Kumar, P., Gajrani, M., Khadilkar, H., Meisheri, H., Baniwal, V., Kulkarni, V.: Reinforcement learning of supply chain control policy using closed loop multi-agent simulation. In: Paolucci, M., Sichman, J.S., Verhagen, H. (eds.) Multi-Agent-Based Simulation XX, pp. 26–38. Springer International Publishing, Cham (2020)
13. Barat, S., Parchure, R., Darak, S., Kulkarni, V., Paranjape, A., Gajrani, M., Yadav, A.: An agent-based digital twin for exploring localized non-pharmaceutical interventions to control covid-19 pandemic. Transact. Indian Natl. Acad. Eng. (2021). https://doi.org/10.1007/s41403-020-00197-5
14. Barriga, A., Rutle, A., Heldal, R.: Personalized and automatic model repairing using reinforcement learning. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), IEEE, pp 175–181 (2019)
15. Bikakis, A., Papatheodorou, C., Antoniou, G.: The DR-Prolog tool suite for defeasible reasoning and proof explanation in the semantic web. In: Darzentas J, Vouros GA, Vosinakis S, Arnellos A (eds) Artificial Intelligence: Theories, Models and Applications, 5th Hellenic Conference on AI, SETN 2008, Syros, Greece, October 2-4, 2008. Proceedings, Springer, Lecture Notes in Computer Science, vol 5138, p 345–351, https://doi.org/10.1007/978-3-540-87881-0_31, (2008)
16. Bill, R., Fleck, M., Troya, J., Mayerhofer, T., Wimmer, M.: A local and global tour on momot. Softw. Syst. Model. **18**(2), 1017–1046 (2019). https://doi.org/10.1007/s10270-017-0644-3
17. Black, D., Rapos, EJ., Stephan, M.: Voice-driven modeling: Software modeling using automated speech recognition. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), IEEE, pp 252–258 (2019)
18. Bordea, G., Faralli, S., Mougin, F., Buitelaar, P., Diallo, G.: Evaluation dataset and methodology for extracting application-specific taxonomies from the Wikipedia knowledge graph. In: Proceedings of the 12th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, pp 2341–2347, https://www.aclweb.org/anthology/2020.lrec-1.285 (2020)
19. Brambilla, M., Cabot, J., Cánovas Izquierdo, JL., Mauri, A.: Better call the crowd: using crowdsourcing to shape the notation of domain-specific languages. In: Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, pp 129–138 (2017a)

20. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice: Second Edition, 2nd edn. Morgan &; Claypool Publishers (2017b)
21. Burdusel, A., Zschaler, S., Strüber, D.: Mdeoptimiser: a search based model engineering tool. In: Babur Ö, Strüber D, Abrahão S, Burgueño L, Gogolla M, Greenyer J, Kokaly S, Kolovos DS, Mayerhofer T, Zahedi M (eds) Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS 2018, Copenhagen, Denmark, October 14-19, 2018, ACM, pp 12–16, https://doi.org/10.1145/3270112.3270130, (2018)
22. Burgueño, L., Cabot, J., Gérard, S.: The future of model transformation languages: an open community discussion. J. Object Technol. (2019). https://doi.org/10.5381/jot.2019.18.3.a7
23. Burgueño, L., Cabot, J., Gérard, S.: An LSTM-based neural network architecture for model transformations. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS), IEEE, pp 294–299 (2019b)
24. Cabot, J., Clarisó, R., Brambilla, M., Gérard, S.: Cognifying model-driven software engineering. In: Federation of International Conferences on Software Technologies: Applications and Foundations, Springer, pp 154–160 (2017)
25. Combemale, B., Kienzle, J., Mussbacher, G., Ali, H., Amyot, D., Bagherzadeh, M., Batot, E., Bencomo, N., Benni, B., Bruel, J., Cabot, J., Cheng, B.C., Collet, P., Engels, G., Heinrich, R., Jezequel, J., Koziolek, A., Mosser, S., Reussner, R., Sahraoui, H., Saini, R., Sallou, J., Stinckwich, S., Syriani, E., Wimmer, M.: A hitchhiker's guide to model-driven engineering for data-centric systems. IEEE Software **01**,(2020). https://doi.org/10.1109/MS.2020.2995125
26. Daniel, G., Cabot, J., Deruelle, L., Derras, M.: Xatkit: a multimodal low-code chatbot development framework. IEEE Access **8**, 15332–15346 (2020)
27. Devlin, J., Chang, MW., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint http://arxiv.org/abs/1810.04805 (2018)
28. Duan, Y., Shao, L., Hu, G., Zhou, Z., Zou, Q., Lin, Z.: Specifying architecture of knowledge graph with data graph, information graph, knowledge graph and wisdom graph. In: 2017 IEEE 15th International Conference on Software Engineering Research, pp. 327–332. Management and Applications (SERA), IEEE (2017)
29. Eisenberg, M., Pichler, H., Garmendia, A., Wimmer, M.: Towards reinforcement learning for in-place model transformations. In: 24th International Conference on Model Driven Engineering Languages and Systems, MODELS 2021, Fukuoka, Japan, October 10-15, 2021, IEEE, pp 82–88, https://doi.org/10.1109/MODELS50736.2021.00017, (2021)
30. *streamlined* (2021) In: The Merriam-Webster Dictionary, Based on Merriam-Webster's Collegiate ® Dictionary 11th edn, Merriam-Webster Inc., https://www.merriam-webster.com/dictionary/streamlined
31. García, M.A.M., Rodríguez, R.P., Rifón, L.A.: Leveraging Wikipedia knowledge to classify multilingual biomedical documents. Artif. Intell. Med. **88**, 37–57 (2018)
32. Goldberg, Y., Orwant, J.: A dataset of syntactic-ngrams over time from a very large corpus of English books. In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp 241–247 (2013)
33. Harris, Z.: Distributional structure. Word **10**(23), 146–162 (1954)
34. Hartmann, T., Moawad, A., Fouquet, F., Le Traon, Y.: The next evolution of MDE: a seamless integration of machine learning into domain modeling. Softw. Syst. Model. **18**(2), 1285–1304 (2019)
35. He, X., Zhao, K., Chu, X.: AutoML: a survey of the state-of-the-art. Knowl.-Based Syst. **212**(106), 622 (2019)
36. Hildebrandt, C., Törsleff, S., Caesar, B., Fay, A.: Ontology building for cyber-physical systems: A domain expert-centric approach. In: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), IEEE, pp 1079–1086 (2018)
37. Hossayni, H., Khan, I., Aazam, M., Taleghani-Isfahani, A., Crespi, N.: SemKoRe: improving machine maintenance in industrial iot with semantic knowledge graphs. Appl. Sci. **10**(18), 6325 (2020)
38. John, S., Burdusel, A., Bill, R., Strüber, D., Taentzer, G., Zschaler, S., Wimmer, M.: Searching for optimal models: Comparing two encoding approaches. J. Object Technol. **18**(3), 1–22 (2019). https://doi.org/10.5381/jot.2019.18.3.a6
39. Karam, N., Streibel, O., Karjauv, A., Coskun, G., Paschke, A.: Answering controlled natural language questions over RDF clinical data. In: European Semantic Web Conference, Springer, pp 129–134 (2020)
40. Kharlamov, E., Grau, BC., Jiménez-Ruiz, E., Lamparter, S., Mehdi, G., Ringsquandl, M., Nenov, Y., Grimm, S., Roshchin, M., Horrocks, I.: Capturing industrial information models with ontologies and constraints. In: International Semantic Web Conference, Springer, pp 325–343 (2016)
41. Kholkar, D., Sunkle, S., Kulkarni, V.: From natural-language regulations to enterprise data using knowledge representation and model transformations. In: Maciaszek LA, Cardoso JS, Ludwig A, van Sinderen M, Cabello E (eds) ICSOFT-PT, Lisbon, Portugal, July 24 - 26, 2016., SciTePress, p 60–71, https://doi.org/10.5220/0006002600600071, (2016)
42. Kühne, T.: What is a model? In: Bézivin J, Heckel R (eds) Language Engineering for Model-Driven Software Development, 29. February - 5. March 2004, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl Seminar Proceedings, vol 04101, http://drops.dagstuhl.de/opus/volltexte/2005/23 (2004)
43. Kulkarni, V.: Model driven software development- a practitioner takes stock and looks into future. In: European Conference on Modelling Foundations and Applications, Springer, p 220–235 (2013)
44. Lano, K., Fang, S., Umar, M., Yassipour-Tehrani, S.: Enhancing model transformation synthesis using natural language processing. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, pp 1–10 (2020)
45. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
46. Liu, Y., Zhu, Y., Che, W., Qin, B., Schneider, N., Smith, NA.: Parsing tweets into Universal Dependencies. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, pp 965–975, https://doi.org/10.18653/v1/N18-1088, (2018)
47. Lopes, J., Cambeiro, J., Amaral, V.: ModelByVoice-towards a general purpose model editor for blind people. In: MODELS Workshops, pp 762–769 (2018)
48. Matthes, F., Mendling, J., Rinderle-Ma, S.: (eds) 20th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2016, Vienna, Austria, September 5-9, 2016, IEEE Computer Society, http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7578983 (2016)
49. Mihalcea, R., Tarau, P.:TextRank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Barcelona, Spain, pp 404–411, https://aclanthology.org/W04-3252 (2004)
50. Moin, A., Rössler, S., Sayih, M., Günnemann, S.: From things' modeling language (ThingML) to things' machine learning (ThingML2). In: Proceedings of the 23rd ACM/IEEE International

Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, pp 1–2 (2020)
51. Mussbacher, G., Combemale, B., Abrahão, S., Bencomo, N., Burgueño, L., Engels, G., Kienzle, J., Kühn, T., Mosser, S., Sahraoui, H., et al.: Towards an assessment grid for intelligent modeling assistance. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, pp 1–10 (2020a)
52. Mussbacher, G., Combemale, B., Kienzle, J., Abrahão, S., Ali, H., Bencomo, N., Búr, M., Burgueño, L., Engels, G., Jeanjean, P., et al.: Opportunities in intelligent modeling assistance. Softw. Syst. Model. **19**(5), 1045–1053 (2020)
53. Nityasya, MN., Mahendra, R., Adriani, M.: Hypernym-hyponym relation extraction from indonesian wikipedia text. In: 2018 International Conference on Asian Language Processing (IALP), IEEE, pp 285–289 (2018)
54. Patil, A., Sunkle, S., Kulkarni, V.: Checking, generating, and revising safety data sheets using globally harmonized system standards. In: 24th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2020, Eindhoven, The Netherlands, October 5-8, 2020, IEEE, p 165–170, https://doi.org/10.1109/EDOC49727.2020.00028, https://www.researchgate.net/publication/346173384_Checking_Generating_and_Revising_Safety_Data_Sheets_using_Globally_Harmonized_System_Standards(2020)
55. Pérez-Soler, S., González-Jiménez, M., Guerra, E., de Lara, J.: Towards conversational syntax for domain-specific languages using chatbots. J Object Technol **18**(2), 5–1 (2019)
56. Perianez-Pascual J, Rodriguez-Echeverria R, Burgueño L, Cabot J (2020) Towards the optical character recognition of DSLs. In: Proceedings of the 13th ACM SIGPLAN International Conference on Software Language Engineering, pp 126–132
57. Perzylo, A., Kessler, I., Profanter, S., Rickert, M.: Toward a knowledge-based data backbone for seamless digital engineering in smart factories. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, vol 1, pp 164–171 (2020)
58. Petersen, N., Halilaj, L., Grangel-González, I., Lohmann, S., Lange, C., Auer, S .:Realizing an RDF-based information model for a manufacturing company–a case study. In: International semantic web conference, Springer, pp 350–366 (2017)
59. Ringsquandl, M., Kharlamov, E., Stepanova, D., Lamparter, S., Lepratti, R., Horrocks, I., Kröger, P.: On event-driven knowledge graph completion in digital factories. In: 2017 IEEE International Conference on Big Data (Big Data), IEEE, pp 1676–1681 (2017)
60. Roychoudhury, S., Sunkle, S., Rathod, H., Kulkarni, V.: Toward structured simulation of enterprise models. In: Grossmann G, Hallé S, Karastoyanova D, Reichert M, Rinderle-Ma S (eds) 18th IEEE International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, EDOC Workshops 2014, Ulm, Germany, September 1-2, 2014, IEEE, p 72–76, https://doi.org/10.1109/EDOCW.2014.19, (2014)
61. Roychoudhury, S., Bellarykar, N., Kulkarni, V.: A NLP based framework to support document verification-as-a-service. In: [48], p 1–10, https://doi.org/10.1109/EDOC.2016.7579376, (2016)
62. Roychoudhury, S., Sunkle, S., Kholkar, D., Kulkarni, V.: A domain-specific controlled english language for automated regulatory compliance (industrial paper). In: Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2017, Vancouver, BC, Canada, October 23-24, 2017, p 175–181, https://doi.org/10.1145/3136014.3136018, (2017a)
63. Roychoudhury, S., Sunkle, S., Kholkar, D., Kulkarni, V.: From natural language to SBVR model authoring using structured english for compliance checking. In: Hallé S, Villemaire R, Lagerström R (eds) 21st IEEE EDOC 2017, Quebec City, QC, Canada, October 10-13, 2017, IEEE Computer Society, p 73–78, https://doi.org/10.1109/EDOC.2017.19, (2017b)
64. Roychoudhury, S., Sunkle, S., Choudhary, N., Kholkar, D., Kulkarni, V.: A case study on modeling and validating financial regulations using (semi-) automated compliance framework. In: 11th IFIP WG 8.1. Working Conference, PoEM 2018, Vienna, Austria, p 288–302, https://doi.org/10.1007/978-3-030-02302-7_18, (2018)
65. Saini, R., Mussbacher, G., Guo, JL., Kienzle, J.: Towards queryable and traceable domain models. In: 2020 IEEE 28th International Requirements Engineering Conference (RE), IEEE, pp 334–339 (2020)
66. Saxena, K., Patil, A., Sunkle, S., Kulkarni, V.: Mining heterogeneous data for formulation design. In: Di Fatta G, Sheng VS, Cuzzocrea A, Zaniolo C, Wu X (eds) 20th International Conference on Data Mining Workshops, ICDM Workshops 2020, Sorrento, Italy, November 17-20, 2020, IEEE, p 589–596, https://doi.org/10.1109/ICDMW51313.2020.00084, (2020)
67. Saxena, K., Singh, T., Patil, A., Sunkle, S., Kulkarni, V.: Leveraging Wikipedia navigational templates for curating domain-specific fuzzy conceptual bases. In: Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances, Association for Computational Linguistics, Online, pp 1–7, https://doi.org/10.18653/v1/2021.dash-1.1, https://aclanthology.org/2021.dash-1.1 (2021a)
68. Saxena, K., Sunkle, S., Kulkarni, V.: Towards recommendations from user-specific insights based on historical legal cases. In: Jain S, Gupta A, Lo D, Saha D, Sharma R (eds) ISEC 2021: 14th Innovations in Software Engineering Conference, India, February 25-27, 2021, https://www.researchgate.net/publication/350007811_Towards_Recommendations_from_User-specific_Insights_based_on_Historical_Legal_Cases (2021b)
69. Segura, ÁM., Pescador, A., de Lara, J., Wimmer, M.: An extensible meta-modelling assistant. In: 2016 IEEE 20th International Enterprise Distributed Object Computing Conference (EDOC), IEEE, pp 1–10 (2016)
70. Steinberger, C., Kop, C.: A domain specific modeling language for model-based design of voice user interfaces. In: Michael J, Torres V (eds) ER Forum, Demo and Posters 2020 co-located with 39th International Conference on Conceptual Modeling (ER 2020), Vienna, Austria, November 3-6, 2020, CEUR-WS.org, CEUR Workshop Proceedings, vol 2716, pp 3–16, http://ceur-ws.org/Vol-2716/paper1.pdf (2020)
71. Sunkle, S., Kulkarni, V., Roychoudhury, S.: Analyzing Enterprise Models Using Enterprise Architecture-Based Ontology. In: Moreira A, Schätz B, Gray J, Vallecillo A, Clarke PJ (eds) MoDELS, Springer, Lecture Notes in Computer Science, vol 8107, p 622–638 (2013a)
72. Sunkle, S., Kulkarni, V., Roychoudhury, S.: Intentional Modeling for Problem Solving in Enterprise Architecture. In: Hammoudi S, Maciaszek LA, Cordeiro J, Dietz JLG (eds) ICEIS (3), SciTePress, p 267–274 (2013b)
73. Sunkle, S., Roychoudhury, S., Kulkarni, V.: Using intentional and system dynamics modeling to address WHYs in enterprise architecture. In: Marca, D.A., van Sinderen, M., Cordeiro, J. (eds.) ICSOFT. SciTePress, Setúbal (2013)
74. Sunkle, S., Kholkar, D., Kulkarni, V.: Explanation of Proofs of Regulatory (Non-) Compliance Using Semantic Vocabularies. In: Bassiliades N, Gottlob G, Sadri F, Paschke A, Roman D (eds) Rule Technologies: Foundations, Tools, and Applications - 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, Springer, LNCS, vol 9202, p 388–403, https://doi.org/10.1007/978-3-319-21542-6_25, (2015a)
75. Sunkle, S., Kholkar, D., Kulkarni, V.: Model-driven regulatory compliance: A case study of ”know your customer” regulations. In: Lethbridge T, Cabot J, Egyed A (eds) 18th ACM/IEEE International Conference on Model Driven Engineering Languages and

Systems, MoDELS 2015, Ottawa, ON, Canada, September 30 - October 2, 2015, IEEE Computer Society, p 436–445, https://doi.org/10.1109/MODELS.2015.7338275, http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7328153 (2015b)
76. Sunkle, S., Kholkar, D., Kulkarni, V.: Toward better mapping between regulations and operations of enterprises using vocabularies and semantic similarity. CSIMQ **5**, 39–60 (2015). https://doi.org/10.7250/csimq.2015-5.04
77. Sunkle, S., Kholkar, D., Kulkarni, V.: Comparison and synergy between fact-orientation and relation extraction for domain model generation in regulatory compliance. In: Conceptual Modeling - 35th International Conference, ER 2016, Gifu, Japan, November 14-17, 2016, Proceedings, Lecture Notes in Computer Science, vol 9974, p 381–395, https://doi.org/10.1007/978-3-319-46397-1_29, (2016a)
78. Sunkle, S., Kholkar, D., Kulkarni, V.: Informed active learning to aid domain experts in modeling compliance. In: [48], p 1–10, https://doi.org/10.1109/EDOC.2016.7579382, (2016b)
79. Sunkle, S., Jain, D., Saxena, K., Patil, A., Chacko, R., Rai, B.: Generate and test for formulated product variants with information extraction and an in-silico model. In: Advanced Digital Architectures for Model-Driven Adaptive Enterprises, IGI Global, p 223–250 (2020a)
80. Sunkle, S., Saxena, K., Kulkarni, V.: Conceptual modeling of legal case insights for stakeholder decision making. In: Michael J, Torres V (eds) ER Forum, Demo and Posters 2020 co-located with 39th International Conference on Conceptual Modeling (ER 2020), Vienna, Austria, November 3-6, 2020, CEUR-WS.org, CEUR Workshop Proceedings, vol 2716, p 31–44, http://ceur-ws.org/Vol-2716/paper3.pdf (2020b)
81. Sunkle, S., Saxena, K., Patil, A., Kulkarni, V., Jain, D., Chacko, R., Rai, B.: Information extraction and graph representation for the design of formulated products. In: Dustdar S, Yu E, Salinesi C, Rieu D, Pant V (eds) CAiSE 2020, Grenoble, France, June 8-12, 2020, Proceedings, Springer, Lecture Notes in Computer Science, vol 12127, p 433–448, https://doi.org/10.1007/978-3-030-49435-3_27, (2020c)
82. Sunkle, S., Saxena, K., Patil, A., Kulkarni, V., Jain, D., Chacko, R., Rai, B.: Knowledge graph for formulated product design. In: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining Workshops, KDD Workshops, Virtual Event, CA, USA, August 23-27, 2020, https://suitclub.ischool.utexas.edu/IWKG_KDD2020/index.html (2020d)
83. Sunkle, S., Jain, D., Saxena, K., Patil, A., Singh, T., Rai, B., Kulkarni, V.: Integrated "Generate, Make, and Test" for Formulated Products using Knowledge Graphs. Data Intelligence 3(3):340–375, https://doi.org/10.1162/dint_a_00096, https://direct.mit.edu/dint/article-pdf/3/3/340/1963445/dint_a_00096.pdf (2021)
84. Tran, T., Toshniwal, S., Bansal, M., Gimpel, K., Livescu, K., Ostendorf, M.: Joint modeling of text and acoustic-prosodic cues for neural parsing. CoRR abs/1704.07287, http://arxiv.org/abs/1704.07287, 1704.07287 (2017)
85. Vo, N., Mitra, A., Baral, C.: The NL2KR platform for building natural language translation systems. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp 899–908 (2015)
86. Weigelt, S., Steurer, V., Hey, T., Tichy, WF.: Programming in natural language with fuse: Synthesizing methods from spoken utterances using deep natural language understanding. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp 4280–4295 (2020)
87. Weikum, G., Hoffart, J., Suchanek, F.: Knowledge harvesting: achievements and challenges. In: Computing and Software Science, Springer, pp 217–235 (2019)
88. Wongsuphasawat, K., Liu, Y., Heer, J.: Goals, process, and challenges of exploratory data analysis: An interview study. 1911.00568 (2019)
89. Yu, H., Li, H., Mao, D., Cai, Q.: A domain knowledge graph construction method based on wikipedia. Journal of Information Science p 0165551520932510 (2020)
90. Yue, T., Briand, L.C., Labiche, Y.: A systematic review of transformation approaches between user requirements and analysis models. Requir. Eng. **16**(2), 75–99 (2011)

Sagar Sunkle is a senior scientist at Tata Consultancy Services (TCS). Sagar has a Ph.D. in Software Engineering and two masters, one in Computer Science specializing in Soft Computing technologies and another in Data and Knowledge Engineering from prestigious universities in Germany and India. Sagar has authored over 40 publications, including conferences, journals, book chapters, and patents, and serves as a program committee member for premier ACM and IEEE conferences. His current research interests include using natural language processing and machine learning to derive insights and transform them into recommendations with applications to material informatics and other business domains.

Krati Saxena is a Scientist at Tata Consultancy Services (TCS). Previously, she completed a Bachelor's degree in System Science from the Indian Institute of Technology Jodhpur and received a Master's degree in Global Advanced Assistive Robotics course from Kyushu Institute of Technology, Japan. Her current research interests include content mining, natural language processing, textual AI, data-driven decision-making, and knowledge discovery.

Ashwini Patil is a researcher at Tata Consultancy Services (TCS). Previously, she completed a Master's degree in Computer Science from Visvesvaraya National Institute of Technology (VNIT), Nagpur, India. She currently uses natural language processing to generate executable specifications from natural language text for document computing. Her current research interests include graph databases, semantic Similarly, and deep learning for information and knowledge discovery.

Vinay Kulkarni is a Chief Scientist and Head of Software Systems Research at Tata Consultancy Services (TCS) and a Fellow of the Indian National Academy of Engineering. His research interests include model-driven software engineering, enterprise modeling, and software engineering for an uncertain world. His work has led to a toolset used to deliver several large business-critical systems over the past 20 years. Much of this work has found a way into OMG standards. He has several patents to his credit and has authored more than 100 papers in journals and conferences worldwide. He has served as the chairperson and program committee member for the premier ACM and IEEE international conferences. An alumnus of the Indian Institute of Technology Madras, Vinay also serves as Visiting Professor at Middlesex University, London.