Towards an Assessment Grid for Intelligent Modeling Assistance

Gunter Mussbacher, Benoit Combemale, Silvia Abrahão, Nelly Bencomo, Loli Burgueño, Gregor Engels, Jörg Kienzle, Thomas Kühn, Sébastien Mosser, Houari Sahraoui, et al.

HAL Id: hal-02925142

https://inria.hal.science/hal-02925142

Submitted on 28 Aug 2020

Towards an Assessment Grid
for Intelligent Modeling Assistance

Gunter Mussbacher
McGill University
Montreal, Canada
gunter.mussbacher@mcgill.ca

Benoit Combemale
University of Toulouse & Inria
Rennes, France
benoit.combemale@inria.fr

Silvia Abrahão
Universitat Politècnica de València
Valencia, Spain
sabrahao@dsic.upv.es

Nelly Bencomo
Aston University
Birmingham, UK
n.bencomo@aston.ac.uk

Loli Burgueño
Open Uni. of Catalonia / CEA List
Barcelona / Paris, Spain / France
lburguenoc@uoc.edu

Gregor Engels
Paderborn University
Paderborn, Germany
engels@upb.de

Jörg Kienzle
McGill University
Montreal, Canada
joerg.kienzle@mcgill.ca

Thomas Kühn
Karlsruher Institut für Technologie
Karlsruhe, Germany
thomas.kuehn@kit.edu

Sébastien Mosser
Université du Québec à Montréal
Montreal, Canada
mosser.sebastien@uqam.ca

Houari Sahraoui
DIRO, Université de Montréal
Montreal, Canada
sahraouh@iro.umontreal.ca

Martin Weyssow
DIRO, Université de Montréal
Montreal, Canada
martin.weyssow@umontreal.ca

ABSTRACT

The ever-growing complexity of systems, the growing number of stakeholders, and the corresponding continuous emergence of new domain-specific modeling abstractions has led to significantly higher cognitive load on modelers. There is an urgent need to provide modelers with better, more Intelligent Modeling Assistants (IMAs). An important factor to consider is the ability to assess and compare, to learn from existing and inform future IMAs, while potentially combining them. Recently, a conceptual Reference Framework for Intelligent Modeling Assistance (RF-IMA) was proposed. RF-IMA defines the main required components and high-level properties of IMAs. In this paper, we present a detailed, level-wise definition for the properties of RF-IMA to enable a better understanding, comparison, and selection of existing and future IMAs. The proposed levels are a first step towards a comprehensive assessment grid for intelligent modeling assistance. For an initial validation of the proposed levels, we assess the existing landscape of intelligent modeling assistance and three future scenarios of intelligent modeling assistance against these levels.

CCS CONCEPTS

• **General and reference** → **Evaluation**; • **Software and its engineering** → **Model-driven software engineering**; **Abstraction, modeling and modularity**; **Integrated and visual development environments**; **Application specific development environments**.

KEYWORDS

Model-Based Software Engineering, Intelligent Modeling Assistance, Integrated Development Environment, Artificial Intelligence, Feedback, Assessment Levels

1 INTRODUCTION

Over the last decades, the use of abstractions provided by general-purpose and domain-specific modeling languages has supported a multitude of stakeholders involved in software development. However, modelers are facing high cognitive load for their modeling tasks due to *(i)* the continuous increase of the complexity of the problems where modeling techniques are used nowadays and *(ii)* the increasing number of stakeholders whose needs have to be addressed, and with it the domain-specific modeling abstractions used by these stakeholders. For example, the development of modern data-driven software systems potentially involves many

different interdisciplinary perspectives [22, 28]. As many heterogeneous modeling abstractions are used across the software life-cycle, modelers have to ensure the global consistency of these modeling abstractions and their relationships with other abstractions [39]. The recent use of black-box techniques, such as machine learning, also has an impact on understandability [19, 33]. All the above have increased the demand for support for working with large models [10]. As a result, there is an urgent need to provide modelers with better, more Intelligent Modeling Assistants (IMAs).

At the same time, the availability of data that can be observed from different modeling activities has increased significantly, leading to many opportunities to provide intelligent modeling assistance to modelers, *e.g.,* using previous modeling experiences or historical information in model repositories, and considering the modelers' context or domain-specific knowledge. Data-driven techniques [15] (*e.g.,* data mining and machine learning [20]) enable the automatic derivation of modeling knowledge and the provision of context-aware assistance. It is of utmost importance to make use of this data and associated techniques, combined with the power of abstraction, to assist modelers in their modeling activities.

An important factor to consider is the ability to assess and compare different IMAs, to learn from existing IMAs, inform future IMAs, and possibly combine different IMAs. Unfortunately, most existing IMAs lack transparency to understand their scope as well as adaptability and flexibility to be customized, compared, and combined beyond their original intended use. Recently, a conceptual Reference Framework for Intelligent Modeling Assistance (RF-IMA) [29] defined the main components required for IMAs, the interactions of these components, and high-level properties of IMAs. RF-IMA describes how the framework would help comprehend, develop, and ultimately compare IMAs. RF-IMA serves tool builders when engineering new, improved IMAs and modelers who will eventually use these better IMAs, which may cover a broad range, from basic model auto-completion tools and modeling chatbots to more advanced modeling environments fueled by data-driven techniques. IMAs may help a modeler who may be unaware of knowledge required for an activity or who does not have time to access such knowledge.

In this paper, and as a first step towards a comprehensive assessment grid for intelligent modeling assistance, we present a detailed, level-wise definition of these properties of RF-IMA to enable a better understanding and comparison of existing and future IMAs. For each property, different levels are proposed that exemplify the increasing capabilities of an IMA to support intelligent modeling assistance. An IMA can be more or less intelligent, and it will therefore have a series of capabilities depending on that intrinsic intelligence. As an initial validation of the proposed levels, we assess the existing landscape of intelligent modeling assistance and three future scenarios of intelligent modeling assistance against these levels.

In the remainder of this paper, Section 2 summarizes RF-IMA. Section 3 discusses the details of the properties of RF-IMAs by describing several levels for each property. Section 4 assesses the existing landscape of IMAs against the levels of the properties, and then illustrates three future scenarios for IMAs and assesses them as well. Section 5 concludes the paper and presents future work.

2 BACKGROUND

The conceptual reference framework RF-IMA [29] is the outcome of a week-long workshop on *Data and Models*[1] with an emphasis on intelligent modeling environments. RF-IMA lays the foundation for modeling activities supported by an *Intelligent Modeling Assistant* (IMA). As illustrated in Fig. 1, RF-IMA gives a holistic perspective of how a modeler's context in the *Socio-Technical Modeling System* (STMS, ①) is connected with external sources of data, information, and knowledge or possibly domain experts (④) to realize intelligent modeling assistance. In an STMS, an actor (*i.e.,* modeler) works on models with the help of the technical infrastructure provided by the modeling environment. The activities performed by the actor are driven by the actor's intention, which defines what the actor wants to accomplish and why.

The main component of an IMA (②) is the assistant, which provides assistance to the modeler (*e.g.,* contextualized information, recommendations) with the help of the context shadow (*i.e.,* any information relevant to the modeling activity). The data acquisition/production layer gives access to data/information/knowledge and has the ability to connect a modeler with domain experts. The optional adaptation component uses a modeler's feedback to adapt to changing context (*e.g.,* the modeler's skill level).

The bidirectional communication (③) between the STMS and an IMA is characterized by context information flowing from the STMS to the IMA, which results in assistance being provided by the IMA to the modeler(s) in the STMS and, in turn, feedback being provided by the modeler(s) to the IMA. In general, there might be a federation of IMAs interacting with each other and processing context information. Each actor may interact with multiple IMAs.

Last but not least, RF-IMA highlights several crosscutting quality properties (⑤). For an IMA, the assistant's *quality degree* (*i.e.,* sophistication) and *timeliness* (how often and how quickly is adequate assistance provided?) as well as the *trust* of modelers (*i.e.,* the perception of the quality of the IMA from the modelers' perspective) need to be considered. For the communication, the *degree of autonomy* [34, 38] is important for context information and the modelers' feedback (*e.g.,* see existing levels of autonomy [30]). The assistance may be characterized by *relevance* of and *confidence* in the provided information (how adequate is it for the current modeling context and how sure is the IMA that it is adequate?). Furthermore, *explainability* (i.e., the degree to which an actor understands why a particular assistance is provided to her) is important. For the STMS, the *quality* of the model and modeling activity ensured by the IMA is also an issue (e.g., from a syntactic and semantic point of view). Finally, the *quality* of IMA with respect to its external sources must be considered, too (*e.g.,* scope, transparency, accessibility, accurateness, and data curation concerns).

While the overarching goal of RF-IMA is to create a research momentum to eventually define appropriate, domain-independent, generic modeling interfaces and protocols (e.g., similar to the language server protocol [1] for programming languages) to enable the seamless integration of various IMAs and modeling environments, this paper focuses instead on the more detailed and precise definition of the properties in RF-IMA. Ultimately, the goal of an IMA from a modeler's point of view is to improve the modeler's user

---

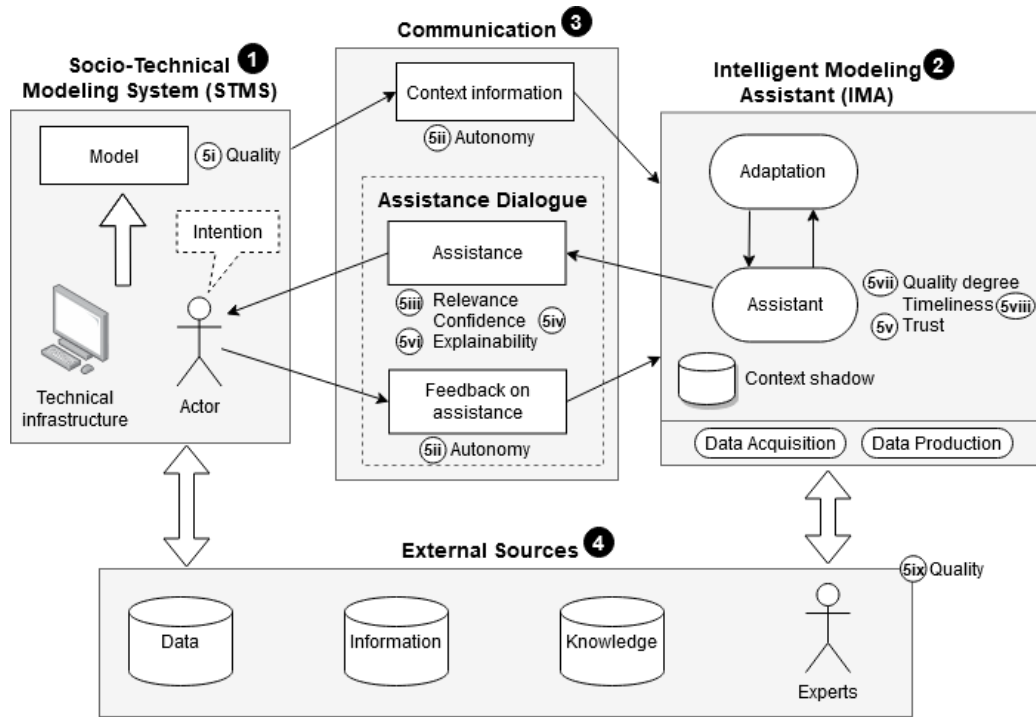[1]http://www.bellairs2020.ece.mcgill.ca/

Figure 1: Conceptual Reference Framework for Intelligent Modeling Assistance (RF-IMA), highlighting four main components (1-4) and the crosscutting concern related to quality properties (5) [29]

experience and to increase both the quality of the model produced and the quality of the modeling process.

3 RF-IMA PROPERTIES

This section defines levels for each property in RF-IMA. All properties are defined and evaluated from the IMA's point of view (and indicated by ⑤ in Figure 1). Each level represents to which degree a property is provided by an IMA: *(i)* assurance of *quality* of models, *(ii) autonomy* with which context information and feedback is gathered, *(iii) relevance* of the assistance, *(iv) confidence* in the assistance, *(v) explainability* of the assistance, *(vi)* the *quality degree* with respect to the degree of excellence of an IMA to address the modeler's needs, *(vii) timeliness* of an IMA, *(viii) trust* a modeler has in an IMA, and *(ix) quality* of an IMA with respect to the external sources used by an IMA. For each property, levels are numbered from 0 to N, with 0 representing the lowest and N representing the highest degree of the increasing capabilities of an IMA. In the remainder of this section, the levels of each property are introduced.

3.1 Quality of IMA Regarding Models

This property represents the ensured level of *quality* of a model from the IMA's and modeler's perspective, which is relative to the specific context and modeling task at hand. This property is assessed by considering the three dimensions in the framework of Lindland et al. [27]: syntactic, semantic, and pragmatic quality.

- *Level 0 – Model quality not ensured.* The IMA does not ensure the quality level of a model at all.

- *Level 1 – Syntactic quality.* The IMA ensures that the model corresponds to the modeling language used in the IMA for the context at hand, *i.e.,* the model conforms to the modeling language. It is assessed by comparing the model (*i.e.,* the set of all statements actually made) to the modeling language (*i.e.,* the set of all statements that are allowed with respect to the modeling language's vocabulary and grammar). The model is syntactically correct, if all the statements in the model are according to the syntax of the language.

- *Level 2 – Semantic quality.* The IMA ensures that the model corresponds to the domain, where the domain is considered the ideal knowledge about the situation to be modeled. A model is semantically correct, if it is valid and complete with respect to the problem domain. Validity means that all information conveyed by the model is correct and relevant to the problem, whereas completeness entails that the model contains all information about the domain that is considered correct and relevant (*i.e.,* a valid model may still be missing some correct and relevant information). Semantic quality can also be assessed subjectively by the modeler. Perceived semantic quality is the correspondence between the modeler's interpretation of a model and her current knowledge of the domain. Krogstie et al. [24] propose several metrics that can be used to assess the modeler's perceived validity and completeness of a model.

- *Level 3 – Pragmatic quality.* The IMA ensures that the model can be understood by the different stakeholders of the IMA. Two aspects are relevant here: social pragmatic quality, which

captures the extent to which the modelers using the IMA completely and accurately understand the statements in the model that are relevant to them, and technical pragmatic quality, which captures the degree to which a modeling environment can interpret the model representation. Social pragmatic quality can be assessed by asking the modeler to rate the pragmatic quality level of a model.

3.2 Autonomy

For the *autonomy* property of context information or feedback, we adapt existing automation levels [30] to modeling and modeling assistance, resulting in the following six levels.

- *Level 0 – No autonomy.* The IMA does not gather context information or feedback independently, *i.e.,* the modeler explicitly provides all information (*e.g.,* the modeler states explicitly that the application domain is avionics).
- *Level 1 – Complete set.* The IMA offers a complete set of alternative choices regarding context or feedback to the modeler (*e.g.,* the modeler is presented with a list of all application domains of which the IMA is aware and selects avionics).
- *Level 2 – Narrowed set.* The IMA narrows the selection down to a few (*e.g.,* the modeler is presented with a list of the most probable application domains and selects avionics).
- *Level 3 – Best suggestion.* The IMA suggests one (best) choice to the modeler (*e.g.,* the modeler is presented with avionics as the best choice possibly in a list of further options and selects avionics).
- *Level 4 – Timed veto.* The IMA allows the modeler a restricted time to veto the suggested choice (*e.g.,* the modeler is presented with a list of application domains where avionics is highlighted as the best, which is considered selected by the IMA after a certain time).
- *Level 5 – Autonomous.* The IMA independently gathers context information or feedback, and then may or may not inform the modeler, possibly on demand (*e.g.,* the IMA identifies avionics as the application domain and possibly informs the modeler about its choice).
- *Level 6 - Fully autonomous.* The IMA gathers all information independently and acts autonomously (*e.g.,* the IMA identifies avionics as the application domain without involving the modeler at all).

3.3 Relevance

Modeling assistance is an iterative process where a human modeler receives support to close the syntactic and semantic gap between the current state of a model and a targeted model which matches with the intentions of the modeler of the ongoing modeling task. A property of such a modeling assistance process is the degree of *relevance* for a modeler. This can be distinguished into the degrees of precision and recall.

Precision means how useful the assistance process and the model extensions and changes proposed by the IMA are for the modeler with respect to closing this syntactic and semantic gap. Recall means how complete the suggested model modifications are with respect to all possible positive recommendations. It depends on the used modeling language and the degree of formality of its definition. It

also depends on whether these measures can be evaluated automatically or are the result of empirical investigations where different modelers engaged with different modeling tasks are observed and questioned about their satisfaction with the used IMA.

An important aspect is that relevance is a human-depending property, as it depends on a modeler's skills and experiences. The same assistance might be helpful for one modeler, while it may be non-relevant or even useless for another modeler. Thus, besides language-based levels (syntactic/semantic relevance), also the pragmatic level, *i.e.,* the human-related level, is of high importance here.

Levels for relevance are given in the range [0,100] and are generally based on individual precision and recall values.

- *Level 0 – No relevance.* The IMA does not provide values for precision and recall at all.
- *Level 1 to 100 – Relevance.* Since the importance of precision and recall may change depending on the system, either both or one of them or another related metric such as accuracy or F-measure may be used to determine these levels. In any case, the result of the metric is normalized to [1,100]. If several metrics are used, the minimum is taken. *Level 100* means the metric value(s) are at the highest possible.

3.4 Confidence

The property *confidence* measures how often the IMA provides a confidence value to the modeler expressing how certain it is whether a given, single assistance or recommendation is useful to the modeler. This is in contrast to relevance which is measured for the IMA as a whole, A confidence value is in the range [0,1], where 0 denotes that the IMA does not know how useful/accurate its recommendation may be, while the other extreme denotes that the IMA is 100% certain. Different values of confidence can exist between 0 and 1 and these values will depend on the application domain, the context, and modeling task at hand. If a technique does not provide a value in the range [0,1], it should be normalized.

The computation of the confidence value will depend on the techniques used by the IMA. For example, if the assistant is implemented as a feed-forward neural network for classification, the value can be given by the probability that the last layer assigns to each category; if the assistant uses Bayesian learning, the value can be computed using the distributions in use, etc. The following levels of the property measure how often confidence is provided:

- *Level 0 – No confidence.* A level of 0 is given, meaning that the IMA only provides recommendations, but it does not associate any confidence value to them. The reason for this may be the lack of data associated to compute the value.
- *Level 1 to 100 – Confidence provided.* These levels represent the percentage of times the IMA makes a recommendation: *Level 100* means that the IMA provides a confidence value each time, and *Level 67* means that the IMA provides a confidence value 67% of the times. The IMA can either provide one recommendation or a list of recommendations sorted by their confidence values.

3.5 Trust

We define *trust* as the perception that the modeler has about the quality of an IMA. The modeler's trust is not built upon a single

suggestion, but it is based on her personal experience after her repeated interactions with the IMA. The modeler's trust can be:

- **Level 0 – Unidentified.** There is no information about the modeler's trust on the IMA.
- **Level 1 – Satisfaction-based.** The IMA obtains information about the modeler's trust by asking the developer herself. To have a better understanding about the modeler's satisfaction and identify possible improvements, the IMA has different alternatives. For example, the IMA can request the modeler to fill in a satisfaction survey, which can be as simple as one single qualitative question about her whole experience with the IMA (*e.g.,* rate your level of trust on the IMA from 1 to 5 being 1 not useful at all and 5 very useful) or more detailed. Another possibility is to ask the modeler to assign levels of confidence to individual suggestions and then composing an overall value for the whole system or its different subsystems (*e.g.,* this could be achieved for (un)certainty, confidence, and belief and their propagation [11, 13]).
- **Level 2 – Feedback-based.** The IMA analyses the modeler's interaction with the IMA and her feedback and infers the modeler's satisfaction. For example, the IMA studies how many suggestions the modeler accepts, ignores, and rejects, how different the solution she finally models is from the suggestions she has received (the closer, the more useful the suggestions were and the more she should trust the IMA).
- **Level 3 – Community-based.** The IMA not only studies the trust of modelers individually but also its own reputation taking into account all modelers as a community. The IMA collects the trust of modelers for individual suggestions to assess their quality in a more generalized manner. The IMA could also use this information when giving recommendations. For example, it could informatively add the percentage of modelers that have chosen a specific recommendation. A modeler might trust more a recommendation which others have indicated as trust-worthy.

3.6 Explainability

We define *explainability* as the degree to which a modeler understands why a particular assistance is provided to her. Especially at the beginning, an IMA's help may be of limited use, if it is not possible to explain how a given assistance was produced, which, in turn, gives insight into whether and why the provided help is relevant.

- **Level 0 – No explanation.** At the lowest level, an IMA does not offer any insight on how its assistance suggestions have been produced.
- **Level 1 – Basic.** IMAs that provide basic explanation report to the modeler what technique was used to produce the assistance. For example, the IMA might report that an assistance was produced by a decision tree, or by pattern-matching the current model to other models from the same domain.
- **Level 2 – Detailed.** Detailed explanation not only discloses to the modeler the technique that was used to produce the assistance, but also the detailed information of the calculation. For example, if a decision tree was used, the IMA could show the modeler the decision path that was taken. If pattern

matching was used, the IMA might show the modeler the models that matched her model.
- **Level 3 – Proven.** At the highest level, an IMA explains the full details of how a suggestion was obtained to a modeler without requiring her to understand the specific technique that was used to produce the assistance. In contrast to Level 2, with a proven explanation, an IMA is also capable of demonstrating to the modeler why the proposed recommendation is the best and not just how it arrived at the conclusion.

3.7 Quality Degree

The *quality degree* measures the degree of excellence of the IMA to address the needs of a modeler. More specifically this level of excellence is identified according to the properties and possibly extra information conveyed for each option provided by the IMA. It ranges from contextualized options, over syntactically or even semantically valid options in that context, to the computation of extra functional properties to allow a modeler to explore a possible design space and perform trade-off analysis. The following levels measure the quality degree of the IMA:

- **Level 1 – Simple lookup.** The IMA offers a set of options after a simple look up, possibly using external sources but not enforcing syntactic or semantic validity. It is called possibly interactively, and displays a list of the possible completions for the activity at hand, before the selected one is applied.
- **Level 2 - Safe lookup.** The IMA leverages the context shadow, external sources, and a syntactic or semantic description of the formalisms employed to provide the modeler with a set of options, which are all syntactically or semantically valid when applied to the current model.
- **Level 3 - Trade-off analysis.** The IMA leverages the context shadow, external sources, and tailored predictive models to compute the trade-off among several options with regards to given properties and metrics. The IMA supports the modeler in the exploration of such a design space and in the choice of a specific option to be applied.

3.8 Timeliness

Timeliness is one of the most important properties for measuring user satisfaction of software systems [6]. Unsurprisingly, this holds also true for recommender systems [4]. Although there are other interpretations of timeliness of IMAs (e.g., to which extend temporal information is considered [14] or how up-to-date the used information is [40]), we here focus on timeliness as a measure of responsiveness of the IMA. As such, it not only depends on the response time of the IMA for a given request, but also on the task, frequency, and modelers' expectations (for the response time of a request). Consequently, when classifying IMAs, measuring the maximum response time or checking timeliness guaranties is of limited use, as these measurements are more suitable for classifying/optimizing distributed embedded and/or real-time systems [9, 36], as the perceived timeliness is greatly context-dependent [26]. Thus, we follow the notion proposed by Fadzlah [17] stating that timeliness is the *"[c]apability in acting at a fitting or advantageous time or performing exactly at the time appointed"* [17, p. 436]. In particular, this description focuses on the modelers' perspective on completing a

task. While perfectly capturing the modeler's perception, it does not account for the timeliness of the underlying software with which the modeler interacts. An IMA's task is to provide a suitable recommendation at an advantageous time or whenever requested, whereas the required timeliness of the IMA directly corresponds to the task, request frequency, and patience of the modeler. While we concede that quantifying to which extent response time satisfies modelers' expectations is possible with an empirical evaluation, we propose that the timeliness of IMAs can be categorized with respect to the acceptable response time and frequency of requests. The following levels outline the categorization of an IMA's timeliness:

- *Level 0 – Uncontrollable and sporadically.* The timeliness of an IMA is uncontrollable, if it defers its request to a third party, *e.g., StackOverflow,*[2] whose response is beyond the reach of the IMA.
- *Level 1 – Long running and periodically.* The IMAs timeliness is long running, if recommendations require a long search/computation time, *e.g.,* for computing semantics preserving model refactorings. Consequently, this recommendation can be triggered manually or periodically (*e.g.,* daily). The IMA should indicate the duration until completion and create a report of recommendations as a result.
- *Level 2 – Short running and regularly.* In contrast, an IMA is considered short running, if deriving recommendations is completed in seconds (at most minutes) upon a request. Similar to a search or build job, the IMA should indicate its progress and provide a dedicated view for the resulting recommendations to be shown upon completion.
- *Level 3 – Iterative and regularly.* Similar to short running, the IMA supports iterative timeliness, if it already displays found recommendations, while still completing its tasks.
- *Level 4 – Immediate upon request and frequently.* If the IMA immediately provides recommendations upon request, it is considered immediate. In particular, the response time expected by modelers should be below a second, *e.g.,* when requesting quick fixes for an ill-formed model, regardless of whether results are provided at once or iteratively. Modelers will issue these requests more frequently and do not expect a large number of results.
- *Level 5 – Immediate autonomous and frequently.* In addition to the previous level, an IMA is considered immediate autonomous if it provides recommendations immediately without waiting for a request from the modeler by observing triggering actions of the modeler. As these actions occur in high frequencies, the IMA must provide/recompute recommendations fast (under 500 milliseconds) as otherwise a modeler's actions might invalidate already provided recommendations.

3.9 Quality of IMA Regarding External Sources

There are four dimensions to the *quality* of the IMA with respect to its external sources, *i.e.,* the IMA is assessed against each of these four dimensions. First, the IMA may be assessed by the scope of its external sources (indicated by the prefix *S* for levels), *i.e.,* characterized by its increasing sophistication. Second to fourth,

the IMA may choose to use external sources with varying accessibility/transparency, up-to-dateness/accuracy, and data curation characteristics, respectively (indicated by the prefixes *A*, *U*, and *C* for levels, respectively).

- *Level 0 – No external sources.* An IMA does not use any external source.
- *Level S1 – Project and infrastructure.* The IMA uses *(i)* current and past development artefacts, *i.e.,* technical data, information, or knowledge about the project, or *(ii)* data, information, or knowledge about employed IDEs and development tools, *i.e.,* the technical infrastructure.
- *Level S2 – Process.* The IMA uses data, information, or knowledge about the development process such as team size and configuration, including technical background of team members from résumés, which may involve the identification of domain experts.
- *Level S3 – Social network.* The IMA uses skill/network profiles of modelers from social networks, which may also involve the identification of domain experts.
- *Level S4 – Runtime and metadata.* The IMA uses *(i)* performance metrics of various kinds from system execution or simulation or *(ii)* feedback from modelers and measured properties about the IMA that are discussed in this paper.
- *Level S5 – Business and stakeholder.* The IMA uses *(i)* data, information, or knowledge about business type and size, customer profiles, and geographic area or *(ii)* data, information, or knowledge about stakeholders such as intention, objectives, and demographics.
- *Level S6 – Science.* The IMA uses scientific data, information, or knowledge such as novel modeling solutions.
- *Level S7 – Environment.* The IMA uses data, information, or knowledge about the environment such as energy costs at geographical location.
- *Level S8 – Law and culture.* The IMA uses *(i)* data, information, or knowledge such as regulations, certification rules, and standards or *(ii)* cultural data, information, or knowledge such as broad values and social rules.
- *Level A1 – Private.* The IMA uses hard to access or often unavailable (*i.e.,* private) external sources.
- *Level A2 – Public.* The IMA uses only readily accessible and available (*i.e.,* public) external sources.
- *Level U1 - No guarantees.* The IMA does not know whether external sources are up-to-date at all.
- *Level U2 - Eventually up-to-date.* The IMA uses external sources that will be up-to-date, but the exact time frame is unknown or variable.
- *Level U3 - Periodically updated.* The IMA uses external sources that are reliably updated within a given time frame.
- *Level U4 - Immediately up-to-date.* The IMA uses external sources that are continuously kept up-to-date.
- *Level C1 – Basic.* The IMA uses external sources without any metadata (*i.e.,* not well curated).
- *Level C2 – Curated.* The IMA uses external sources that are well curated with a wealth of metadata.

---

[2]https://stackoverflow.com

Table 1: Assessment of existing and future IMAs

Levels	Model 0..3	Autonomy 0..6	Relevance 0..100	Confidence 0..100	Trust 0..3	Explainability 0..3	Qu. Degree 1..3	Timeliness 0..5	Ext. Sources 0,S1..8,A1..2,U1..4,C1..2
[37]	1	0	0	0	0	1	2	2	0
[21]	1	0	0	0	0	1	2	2	S5, A2, U3, C2
[16]	1	3	0	0	0	1	2	2	S1, A2, U3, C2
[25]	1	5	0	0	0	1	2	5	S1, A1, U3, C2
[35]	1	0	0	0	0	2	2	2	S1, A1, U1, C2
[31, 32]	1	0	0	0	0	0	2	4	0
[18]	1	1	0	0	0	1	2	2	S2, A2, U3, C2
[5]	1	1	0	0	0	1	2	2	S1, A2, U3, C2
[8]	2	5	0	0	0	1	2	1	S2, A1, U3, C2
[12]	1	3	>90	0	0	0	1	4	S1, A2, U1, C2
[23]	1	1	0	0	0	1	2	2	S1, A1, U3, C1
Scenario 1	2	5	≥ 90	100	3	2	2	5	S7, A1, U3, C2
Scenario 2	3	5	≥ 80	100	2	3	3	1	S7, A1, U3, C2
Scenario 3	2	3	0	0	2	1	2	4	S8, A1, U4, C2

4 PROPERTY-BASED ASSESSMENT OF EXISTING AND FUTURE IMAS

Since the '80s, assistants have helped with specific software engineering activities such as design and development or to support the enactment of the overall software process. In Section 4.1, we give a brief overview of assistants that focus on modeling activities and assess them against the properties defined in Section 3, while in Section 4.2, we describe and assess three future scenarios. The assessments are made by consensus among the authors.

4.1 Assessment of Related Work

Focusing on the field of software and system modeling, existing assistants together support all key aspects of the system life-cycle of increasingly complex systems. Large efforts have gone into supporting common modeling languages like UML, *e.g.*, to support a particular modeling process [37], and to help build diagrams from natural language [21] or through recommendations from similarities [16] or established patterns [25]. However, it is very time-consuming to build similar IMAs for the diverse set of other domain-specific abstractions that are tailored to heterogeneous stakeholders. A lack of understandability and comparability of IMAs contributes to this difficulty. The proposed properties with their levels are a first step towards addressing this issue.

In the context of domain-specific modeling languages, efforts to apply recommendation have been limited. IMAs are either applied to narrow contexts, *e.g.*, the current model state and its possible extension with regards to the metamodel [35], or to specific interactions, *e.g.*, helping modelers build and query domain-specific models using natural language via a chatbot interface [31, 32]. IMAs may also target specific activities, such as system requirements (*e.g.*, focus on variability [7][3] or behavior [18]), domain modeling [5], or model transformations [8, 12]. Finally, leveraging a collective knowledge for a recommender system for the whole modeling process is envisioned to not only infer model transformations but

also recommend model repair or refactoring [23]. Again, it is difficult to understand and compare these existing IMAs without an assessment framework.

Mendix Assist [2] and ServiceStudio [3] are two commercial tools that have recently introduced artificial intelligence-powered assistants to help citizen programmers develop their own applications. However, they focus on the area of low-code platforms and there is also not enough public data available to be able to assess these two tools.

Table 1 summarizes our assessment of existing IMAs discussed in this subsection based on the property levels introduced in Section 3. Furthermore, the table also shows the assessment of three future scenarios introduced in the following subsection. All existing IMAs focus only on syntactic model quality (Level 1 of *Quality of IMA Regarding Model*), *i.e.*, they do not consider semantic or pragmatic quality, except for one case of semantic quality ([8]). The majority of existing IMAs remains at low levels of *autonomy* (Level 0 or 1), but there are four cases with higher autonomy ([8, 12, 16, 25]). None of the existing IMAs report on *relevance* (Level 0) or *confidence* (Level 0), or provide information about *trust* (Level 0), except for one case with high relevance ([12]). Most of the existing IMAs offer a basic level of *explainability* (Level 1), but there is one case ([35]) that reaches the more detailed level (Level 2). All but one existing IMA provide a safe lookup (Level 2 of *Quality Degree*), *i.e.*, none of them reaches the level for trade-off analysis. Most existing IMAs provide recommendations regularly upon request and are able to do so quickly, *i.e.*, at most minutes (Level 2 of *Timeliness*), except for three cases that offer more immediate and/or autonomous recommendations ([12, 25, 31, 32]). In terms of *external sources*, existing IMAs predominantly use a mix of public and private data (Level A1 or A2) that is well curated (Level C2) and periodically updated (Level U3). However, the scope is generally low at the level of project and infrastructure (Level 1) or process (Level 2), without making use of more sophisticated external sources such as scientific data and knowledge about the environment, law, and culture. In summary, existing IMAs score low for most properties, pointing to a large potential to improve intelligent modeling assistance.

---

[3]Not included in assessment, because it is a literature review

4.2 Looking Ahead on Future Scenarios

Considering the assessment discussed in the previous section, we propose three scenarios that shed light on how to push forward the current state of practice and uncover further exciting research opportunities in the context of IMAs. These scenarios specifically cover *(i)* evolution of an employee's expertise inside a company (skill-based support), *(ii)* provisioning of multi-disciplinary expertise required for modeling activities (multi-disciplinary pull requests), and finally *(iii)* improvements of the way modeling is taught, from both students' and professors' points of view.

Collectively, these three scenarios illustrate the need for pragmatic model quality (Level 3 of *Quality of IMA Regarding Model*) and high levels of *autonomy* (Level 5), *relevance* (Level ≥ 90), *confidence* (Level 100), *trust* (Level 3), *explainability* (Level 3), and *timeliness* (Level 5). The *quality degree* needs to include trade-off analysis (Level 3) and the scope of *external sources* needs to reach Level S8.

Scenario 1: Skill-based Support. This scenario shows how an IMA can support Jeronimo, a software architect at SpaceCorp in his work depending on his level of experience. The main task of Jeronimo is to work on structural modeling. Features of the IMA are enabled and disabled according to Jeronimo's level of experience in the company. As a junior architect, the IMA will support Jeronimo by providing just-in-time, context-aware advice and information, *i.e.,* taking into account the actions already performed by him. This support includes information collected from previously developed models in the company, standards, best practices, and certification information, such as "*The identifier you provided should be more informative*". The suggestions also include background domain knowledge, such as "*You are using engine vocabulary. An engine is generally connected to a propulsor. Do you need more information about this connection?*"

As a more experienced architect, the IMA may give more precise and personalized suggestions to Jeronimo. For example, it could suggest identifier formatting or model layout usually used by Jeronimo. It can also recommend concepts related to ones currently modeled, *e.g.,* when the engine consumption system is described, the IMA recommends the concept of fuel with its relation to the existing model fragment. The IMA always explains why and/or where its suggestions come from so that Jeronimo can take or ignore them fully understanding why he received them.

As an experienced architect, the IMA may propose to Jeronimo to import and customize existing models of other application domains related to the one he is modeling. Such recommendations enhance Jeronimo's creativity. At all times, the IMA's suggestions respect SpaceCorp's policies about data privacy, ethics, etc.

In this context of a highly specialized company, the *external sources* of data are mainly private (Level A1) and gathered across the company's lifetime. The nature of the data sources is broad and covers Jeronimo's modeling, environment data, company-wide data, as well as general knowledge data (Level S7). In addition, the company ensures that the data is well curated (Level C2) and periodically updated (Level U3).

The data quality as well as the logic behind the IMA enables the IMA to provide safe (Level 2 of *Quality Degree*) and semantically correct recommendations with respect to the application domain (Level 2 of *Quality of IMA Regarding Models*). With each recommendation, the IMA provides a detailed explanation of why the recommendation is suggested (Level 2 of *Explainability*) as well as a confidence value associated to it (Level 100 of *Confidence*). Given the safety-critical application domain, the *relevance* of the recommendations of the IMA must be high (the level needs to be in the range [90,100]). To minimize intrusiveness, the IMA collects context information and feedback from modelers and sporadically informs the modelers (Level 5 of *Autonomy*). This IMA not only learns about Jeronimo's interaction with the IMA but also from models previously developed at the company. From this information, the IMA infers the level of *trust* modelers have (Level 3) and helps Jeronimo make better choices. Finally, this IMA reaches *timeliness* Level 5 to enhance the productivity of Jeronimo as much as possible.

Scenario 2: Multi-disciplinary Pull Request. In the second scenario, we consider the software development company IoTCorp working on IoT development for smart environments. These systems are by essence multi-disciplinary, including requirements such as maximization of network efficiency, energy consumption minimization, and maximization of data accuracy. Considering a set of such nonfunctional requirements (NFRs) to be satisfied, it is paramount to encourage the best possible trade-offs while designing the system's architecture. The IMA supports this task by suggesting alternative architectural designs, at the global level allowing the architect to study the impact of the design choices on the levels of satisfaction of the NFRs. At a more fine grained level, the architect is guided by the IMA while pursuing modeling activities. As an example, consider that the architect has the goal to minimize energy consumption in the system, while at the same time maximizing performance. First, the architect can select from the most appropriate design patterns made available by the IMA, according to the situation at hand and the NFRs given. The IMA also guides the architect in the correct use of these patterns. Further, the IMA crucially identifies individual design decisions applied in a given local context, which may jeopardize the state of the system at a global level. Locally, an expert is often not able to perform the required trade-off analyses by themselves, because of the deep knowledge required from the other specialists' domains. Therefore, the IMA identifies domain experts affected by a local design decision, contacts them through a multi-disciplinary pull request to make the trade-off options explicit to all, while individually adapting to the vocabulary, skill level, and interaction style of each expert. Corrective actions are suggested by the IMA in case they are needed (*e.g.,* recommending a different architectural design that at first may have been considered not favorable), while feeding the resultant information into the global trade-off analysis.

Stakeholders expect that the architectural designs recommended by the IMA must be both syntactically and semantically correct, and these models must also be well understood by the different stakeholders involved in the architectural design task (Level 3 of *Quality of IMA Regarding Models*). Given the type of systems developed by the company and the domain problems tackled, the *relevance* of the recommendations must be high (*i.e.,* in the range [80,100]). Furthermore, the IMA needs to provide *confidence* values for its trade-off estimation all the time (Level 100), because Service Levels Agreements prescribe desired quality targets (*e.g.,* maximization of network efficiency should be at least 95%) and the confidence value allows the architect to better understand how reliable a trade-off

estimate is. Explanations about the IMA's confidence values need to be provided.

In this scenario, it is assumed that the IMA collects context and feedback information independently and informs the architect from time to time (Level 5 of *Autonomy*). Furthermore, the IMA studies how different the architectural model finally created by the architect is from the recommendations received and infers the architect's satisfaction (Level 2 of *Trust*). In terms of *explainability*, the IMA needs to demonstrate to the architect why a trade-off is the best considering the current situation in addition to disclosing the details to get these results (Level 3).

The IMA performs trade-off analysis (Level 3 of *Quality Degree*). The recommendations require a long search/computation time, are triggered manually by the architect, and are executed periodically (Level 1 of *Timeliness*). With respect to *external sources*, the IMA uses a collection of private data (Level A1) about the architectural modeling context including historical project data, model coherence rules and patterns observed in sets of multi-disciplinary models, the role of domain experts in the architectural design process and their associated profile data. Other operational system data relevant for the trade-off is also used (Level S7). IoTCorp ensures that the data is well curated (Level C2) and periodically updated (Level U3).

Scenario 3: Teaching. For the third scenario, consider the student Bob attending a modeling course by professor Alice. In addition to the course material, Alice sets up an interactive teaching environment offering additional modeling tasks to students. Bob is given as a starting point a set of user stories / use cases and is tasked with some specific modeling activity, *e.g.,* to create a domain model by means of a class diagram based on the use case descriptions.

During the task completion, Bob gets stuck and asks the IMA for hints. The IMA always responds with a hint encompassing *one* recommendation. First, it gives a general hint about what process steps Bob needs to take next, *e.g.,* "*Identify the nouns in the scenario description, and model them as classes, attributes, or roles.*" As Bob still has trouble figuring out what to do, he asks the IMA for a more specific hint. The IMA then highlights a specific sentence and words. Bob gets it, and starts drawing a diagram involving a class Person and a class Student. Bob either clicks "*Done*" or asks for further hints. The IMA then provides him with feedback on the current class diagram, by applying well-formedness rules or anti-pattern detection. Here it identifies that Bob modeled Student as a class. It then provides gradual hints to help Bob correct his mistake: it first tells Bob there is a problem, then that the problem lies around the class Student, then that Student should be a role played by a person, and ultimately gives him the solution.

During the whole modeling activity of Bob, the IMA observes task completion, number of hints, and details of hints to assess Bob's skill level and learning style. Depending on his acquired skills the IMA proposes another task for Bob to complete further training of his skills. This task, however, is adapted to Bob's learning style, *e.g.,* having inferred that Bob is a group learner, the IMA assigns Bob to another student who is also Bob's friend and with whom he can complete tasks collaboratively.

For Alice, the teacher, the IMA compiles a learning report for the entire class. Upon request she can also access the individual learning performance record of each student. To improve the quality of recommendations, she can also evaluate the effectiveness of the given recommendations on task completion and learning outcomes.

To support the above teaching scenario, the IMA needs to ensure high *model quality* (Level 2) as it should guide the student to design a model that is both syntactically and semantically correct wrt. a given task. Although the IMA presents the student with *one* best recommendation on how to complete the task, it autonomously takes the current task, model under development, the number of preceding requests, and the student's skill level into account (Level 3 of *Autonomy*) to determine the confidence score of each recommendation and the overall relevance score. Based on that, the IMA determines the suitable recommendation, yet shows only the recommendation to the student that maximizes the learning experience (Level 2 of *Quality Degree*). The relevance and confidence scores are not displayed to the student to avoid confusion as only one recommendation is shown (Level 0 of *Relevance* and *Confidence*, respectively). From the above it becomes clear that the teaching IMA builds up trust with the student and continuously observes reaction of the student to provided recommendations calculating his/her satisfaction and acquired skill level (Level 2 of *Trust*). In addition, the IMA should be able to immediately provide a recommendation whenever a hint is requested (Level 4 of *Timeliness*).

Although the IMA does not provide explanations for recommendations to students, it provides teachers with basic explanations for the recommendations given (Level 1 of *Explainability*), as they might need to review the recommendations given to students during a course to adjust the recommendations.

To provide effective recommendations, the IMA must know the project, technical infrastructure, the process used by the student, as well as runtime and metadata information to gather feedback from the student to adjust the recommendations to maximize the learning experience. With access to the student's social network, the IMA might identify a group learner and suggest group learning exercises. With information about the student's cultural background, the IMA is able to further tailor its recommendations (Level S8 of *Quality of IMA Regarding External Sources*). As such, it uses private (Level A1), up-to-date (Level U4), and well curated (Level C2) external sources.

5 CONCLUSION AND FUTURE WORK

To enable better understanding, comparison, and selection of existing and future Intelligent Modeling Assistants (IMAs), we present a level-wise definition for the properties of the recently proposed Reference Framework for Intelligent Modeling Assistance (RF-IMA). A level for a property exemplifies particular capabilities of an IMA to support intelligent modeling assistance, which increases the higher the level. The property levels are a first step towards a comprehensive assessment grid for IMAs, which eventually could lead to a certification process for IMAs. In general, existing IMAs score low for most properties, indicating that there is still tremendous potential to improve intelligent modeling assistance. Therefore, we describe three future scenarios of intelligent modeling assistance that push the capabilities of IMAs further into unexplored areas.

While this paper offers a first assessment of a given set of existing modeling assistants as well as the three future scenarios, we plan to validate more thoroughly the property levels for a wider set of existing IMAs based on a survey of the modeling community. In

addition, realizing the future scenarios, *e.g.*, by building prototypes of the described IMAs, would allow us to get further insights.

A future research avenue could be the potential use of the assessment grid as a blueprint for a feature model to configure IMAs as a product line. Along these lines, one could investigate trade-offs and dependencies among the different properties and their levels to get a better understanding of feature interactions in the feature model. Last but not least, IMAs themselves could be self-adaptive systems that can model themselves and learn from the interactions with the modeler to autonomously reach higher property levels.

REFERENCES

[1] [n.d.]. *Language Server Protocol*. https://microsoft.github.io/language-server-protocol/
[2] [n.d.]. *Mendix Assist*. https://www.mendix.com/blog/introducing-ai-assisted-development-to-elevate-low-code-platforms-to-the-next-level
[3] [n.d.]. *ServiceStudio from OutSystems*. https://www.outsystems.com/ai
[4] Adekunle Oluseyi Afolabi and Pekka Toivanen. 2020. Harmonization and Categorization of Metrics and Criteria for Evaluation of Recommender Systems in Healthcare From Dual Perspectives. *Intl Journal of E-Health and Medical Communications (IJEHMC)* 11, 1 (2020), 69–92. https://doi.org/10.4018/IJEHMC.2020010105
[5] Henning Agt-Rickauer, Ralf-Detlef Kutsche, and Harald Sack. 2018. DoMoRe - A Recommender System for Domain Modeling. In *Proc. of the 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD'18)*. 71–82. https://doi.org/10.5220/0006555700710082
[6] James E. Bailey and Sammy W. Pearson. 1983. Development of a Tool for Measuring and Analyzing Computer User Satisfaction. *Management Science* 29, 5 (1983), 530–545. https://doi.org/10.1287/mnsc.29.5.530
[7] Noor Hasrina Bakar, Zarinah M. Kasirun, and Norsaremah Salleh. 2015. Feature Extraction Approaches from Natural Language Requirements for Reuse in Software Product Lines: A Systematic Literature Review. *J. Syst. Softw.* 106, C (Aug. 2015), 132–149. https://doi.org/10.1016/j.jss.2015.05.006
[8] Islem Baki and Houari A. Sahraoui. 2016. Multi-Step Learning and Adaptive Search for Learning Complex Model Transformations from Examples. *ACM Trans. Softw. Eng. Methodol.* 25, 3 (2016), 20:1–20:37.
[9] A. V. Bogatyrev, V. A. Bogatyrev, and S. V. Bogatyrev. 2020. The Probability of Timeliness of a Fully Connected Exchange in a Redundant Real-Time Communication System. In *2020 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)*. IEEE, Saint-Petersburg, Russia, Russia, 1–4. https://doi.org/10.1109/WECONF48837.2020.9131517
[10] Antonio Bucchiarone, Jordi Cabot, Richard Paige, and Alfonso Pierantonio. 2020. Grand challenges in model-driven engineering: an analysis of the state of the research. *Software and Systems Modeling* (01 2020), 1–9. https://doi.org/10.1007/s10270-019-00773-6
[11] Loli Burgueño, Manuel F. Bertoa, Nathalie Moreno, and Antonio Vallecillo. 2018. Expressing Confidence in Models and in Model Transformation Elements. In *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems* (Copenhagen, Denmark) *(MODELS '18)*. Association for Computing Machinery, New York, NY, USA, 57–66. https://doi.org/10.1145/3239372.3239394
[12] Loli Burgueño, Jordi Cabot, and Sébastien Gérard. 2019. 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, 294–299.
[13] Loli Burgueño, Robert Clarisó, Jordi Cabot, Sébastien Gérard, and Antonio Vallecillo. 2019. Belief uncertainty in software models. In *Proceedings of the 11th International Workshop on Modelling in Software Engineerings, MiSE@ICSE 2019, Montreal, QC, Canada, May 26-27, 2019*, Marsha Chechik, Daniel Strüber, and Dániel Varró (Eds.). ACM, 19–26. https://doi.org/10.1109/MiSE.2019.00011
[14] Dongpei Chen, Xingming Zhang, Haoxiang Wang, and Weina Zhang. 2020. TEAN: Timeliness enhanced attention network for session-based recommendation. *Neurocomputing* 411 (2020), 229 – 238. https://doi.org/10.1016/j.neucom.2020.06.063
[15] Benoit Combemale, Jörg Kienzle, Gunter Mussbacher, Hyacinth Ali, Daniel Amyot, and et al. 2020. A Hitchhiker's Guide to Model-Driven Engineering for Data-Centric Systems. *IEEE Software* (2020). https://doi.org/10.1109/MS.2020.2995125
[16] A. Elkamel, M. Gzara, and H. Ben-Abdallah. 2016. An UML class recommender system for software design. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. 1–8. https://doi.org/10.1109/AICCSA.2016.7945659
[17] Amalina Farhi Ahmad Fadzlah. 2012. Timeliness Measurement Model: A mathematical approach for measuring the timeliness of handheld application usage. *International Journal of New Computer Architectures and their Applications (IJNCAA)*

[18] 2, 3 (2012), 431–450. https://go.gale.com/ps/anonymous?id=GALE|A353645948
[18] Fabian Friedrich, Jan Mendling, and Frank Puhlmann. 2011. Process model generation from natural language text. In *International Conference on Advanced Information Systems Engineering (CAISE)*. Springer, 482–496.
[19] Antonio Garcia-Dominguez and Nelly Bencomo. 2018. Non-human Modelers: Challenges and Roadmap for Reusable Self-explanation. In *Software Technologies: Applications and Foundations*, Martina Seidl and Steffen Zschaler (Eds.). Springer International Publishing, Cham, 161–171.
[20] T. Hartmann, A. Moawad, F. Fouquet, and Y. Le Traon. 2017. The Next Evolution of MDE: A Seamless Integration of Machine Learning into Domain Modeling. In *2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS)*. 180–180. https://doi.org/10.1109/MODELS.2017.32
[21] M. Ibrahim and R. Ahmad. 2010. Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques. In *Proc. of the 2nd International Conference on Computer Research and Development*. 200–204. https://doi.org/10.1109/ICCRD.2010.71
[22] Jörg Kienzle, Gunter Mussbacher, Benoit Combemale, Lucy Bastin, Nelly Bencomo, and et al. 2020. Towards Model-Driven Sustainability Evaluation. *Commun. ACM* 63, 3 (1 3 2020), 80–91. https://doi.org/10.1145/3371906
[23] Stefan Kögel. 2017. Recommender System for Model Driven Software Development. In *11th Joint Meeting on Foundations of Software Engineering* (Paderborn, Germany) *(ESEC/FSE 2017)*. Association for Computing Machinery, New York, NY, USA, 1026–1029. https://doi.org/10.1145/3106237.3119874
[24] John Krogstie, Odd Ivar Lindland, and Guttorm Sindre. 1995. Defining quality aspects for conceptual models. In *ISCO*.
[25] Tobias Kuschke and Patrick Mäder. 2014. Pattern-Based Auto-Completion of UML Modeling Activities. In *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering* (Vasteras, Sweden) *(ASE '14)*. Association for Computing Machinery, New York, NY, USA, 551–556. https://doi.org/10.1145/2642937.2642949
[26] Robert N Levine. 2008. *A geography of time: On tempo, culture, and the pace of life*. Basic Books. 288 pages.
[27] O. I. Lindland, G. Sindre, and A. Solvberg. 1994. Understanding quality in conceptual modeling. *IEEE Software* 11, 2 (1994), 42–49.
[28] Gunter Mussbacher, Daniel Amyot, Ruth Breu, Jean-Michel Bruel, Betty Cheng, and et al. 2014. The Relevance of Model-Driven Engineering Thirty Years from Now. https://doi.org/10.1007/978-3-319-11653-2_12
[29] Gunter Mussbacher, Benoit Combemale, Jörg Kienzle, Silvia Abrahão, Hyacinth Ali, Nelly Bencomo, Márton Búr, Loli Burgueño, Gregor Engels, Pierre Jeanjean, Jean-Marc Jézéquel, Thomas Kühn, Sébastien Mosser, Houari Sahraoui, Eugene Syriani, Dániel Varró, and Martin Weyssow. 2020. Expert Voice: Opportunities in Intelligent Modeling Assistance. *Software and Systems Modeling* (2020).
[30] Raja Parasuraman and Victor Riley. 1997. Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors* 39, 2 (1997), 230–253. https://doi.org/10.1518/001872097778543886 arXiv:https://doi.org/10.1518/001872097778543886
[31] S. Pérez-Soler, G. Daniel, J. Cabot, E. Guerra, and J. de Lara. 2020. Towards automating the synthesis of chatbots for conversational model query. In *Proc. of the Int. Conf. on Exploring Modeling Methods for Systems Analysis and Development*. to appear.
[32] Sara Pérez-Soler, Esther Guerra, and Juan de Lara. 2018. Collaborative Modeling and Group Decision Making Using Chatbots in Social Networks. *IEEE Software* 35, 6 (2018), 48–54. https://doi.org/10.1109/MS.2018.290101511
[33] A. Rocha, J. P. Papa, and L. A. A. Meira. 2010. How Far You Can Get Using Machine Learning Black-Boxes. In *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*. 193–200. https://doi.org/10.1109/SIBGRAPI.2010.34
[34] Michael Rovatsos and Gerhard Weiss. [n.d.]. *Autonomous Software*. 63–84. https://doi.org/10.1142/9789812775245_0003
[35] Sagar Sen, Benoit Baudry, and Hans Vangheluwe. 2010. Towards Domain-specific Model Editors with Automatic Model Completion. *Simulation* 86, 2 (2010), 109–126. https://doi.org/10.1177/0037549709340530
[36] Michael Spörk, Carlo Alberto Boano, and Kay Römer. 2020. Improving the Timeliness of Bluetooth Low Energy in Dynamic RF Environments. *ACM Trans. Internet Things* 1, 2, Article 8 (April 2020), 32 pages. https://doi.org/10.1145/3375836
[37] Kalaivani Subramaniam, Dong Liu, Behrouz Homayoun Far, and Armin Eberlein. 2004. UCDA: Use Case Driven Development Assistant Tool for Class Model Generation. In *SEKE*.
[38] Ingo J. Timm, Peter Knirsch, Hans-Jörg Kreowski, and Andreas Timm-Giel. 2007. *Autonomy in Software Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 255–273. https://doi.org/10.1007/978-3-540-47450-0_17
[39] Jon Whittle, John Hutchinson, and Mark Rouncefield. 2014. The State of Practice in Model-Driven Engineering. *Software, IEEE* 31, 3 (2014), 79–85.
[40] Yanlong Zhang, Hong Zhu, and S. Greenwood. 2005. Empirical validation of Website timeliness measures. In *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, Vol. 1. IEEE, 313 – 318 Vol. 2. https://doi.org/10.1109/COMPSAC.2005.72