ArchiMate Modeling Mistakes: A Comparative Analysis of Student Assignments and Prior Research on EA Modeling Mistakes

Ville Seppänen
University of Jyvaskyla
ville.r.seppanen@jyu.fi

Jarkko Nurmi
University of Jyvaskyla
jarkko.s.nurmi@jyu.fi

Abstract

Enterprise Architecture is one of the core competencies of higher education IS programs and is widely regarded as one the most common ways to produce valuable and usable information for decision-makers regarding business-IT alignment. Prior research notes the limited perceived usefulness of EA visualizations, which are often characterized by their complexity, lack of focus, and inappropriate level of abstraction, which inhibits their effective use for decision-making. Despite this, research on teaching enterprise architecture modeling is scarce, and understanding the problems students face and the solutions to overcome these are lacking. This study reports findings from the analysis of roughly 300 student assignments, collected from an undergraduate course on EA. Our findings indicate that the mistakes made by the students are in line with the prior research, as the student's modeling errors aligned with limitations commonly associated with EA models, such as poor readability, unfit level of abstraction, and either lack of or excessive information in the model.

Keywords: enterprise architecture, ArchiMate, modeling, teaching

1. Introduction

Understanding the interplay between information technology (IT) and business has become critical for the success of any company, serving as a pivotal strategy facilitator. As IT complexity within organizations continues to escalate, successful efforts on business-IT alignment are increasingly reliant on enterprise architecture (EA). EA is widely regarded as a central concept in harmonizing an organization's strategies, business processes, information systems, and technologies (e.g., Kaisler et al., 2005; Robl and Bork, 2022; Seppänen et al., 2009).

The concept of EA can be understood in two ways: either as the physical structure composed of an organization's assets and other resources or as a collection of models and other descriptions that depict this structure. According to the latter view, these artifacts are developed to support the enterprises' planning, development, and management. In this paper, we follow the definition by Kaisler et al. (2005, p. 1): "An enterprise architecture identifies the main components of the organization, its information systems, the ways in which these components work together in order to achieve defined business objectives, and the way in which information systems support the business processes of the organization."

Thus, EA offers a holistic approach to managing different dimensions of an organization. This necessitates that the organizational assets and their interconnections are made visible by modeling. For this purpose, an open ArchiMate language, a technical standard from The Open Group, is extensively used in both public and private organizations. Consequently, ArchiMate is the most used modeling language in higher education EAM education (Robl and Bork, 2022). Due to various reasons, more than half of organizations practicing EA report unsatisfactory quality of EA documentation as a major challenge for their EA practices (Kotusev et al., 2023), making modeling mistakes education a topical issue for research.

In this paper, the research question: *What kind of modeling mistakes are typical for students and how do these align with prior research on EA modeling mistakes by professionals?* is answered by collecting and analyzing data from the EA models the students of information systems and computer science created as learning assignments on an undergraduate university course on EA. We also discuss the potential consequences of these mistakes in actual EA practice and suggest how to reduce the occurrence of these mistakes among students.

According to Kudryavtsev et al. (2018), no universally recognized enterprise architecture teaching materials exist in the field of higher education. We hope this analysis offers insights and guidelines for educators

URI: https://hdl.handle.net/10125/107003
978-0-9981331-7-1

teaching EA modeling. The results should also help students and professionals of information systems, enterprise engineering, and organization modeling to identify the typical EA modeling pitfalls and steer clear of them.

The rest of this paper is structured as follows. The next section covers the theoretical background of our study, discussing the modeling of enterprise architecture, including the industry standard EA modeling language ArchiMate and prior research on teaching EA modeling. Section three introduces the research setting of this study and the course from which the data were collected for this research. After that, the study results are presented in section four and discussed in section five. Finally, section six gives concluding remarks.

2. Theoretical Background

2.1. Enterprise Architecture Modeling

Enterprise architecture has a myriad of potential benefits, including improved understanding of complex organizational structures and processes, enhanced business-IT alignment and change capacity, better IT and business capabilities, and improved decision-making and project management (e.g., Shanks et al., 2018; Tamm et al., 2022). Enterprise architecture modeling can be perceived as a knowledge-intensive and modeling-focused expert profession in which the perceived value of EA comes from ”a comprehensive blueprint of an enterprise covering its business, data, applications and technology domains and consisting of individual EA artifacts” (Kotusev et al., 2022, p. 1). These artifacts are regularly different types of models used to convey information to different stakeholder groups for decision-making (Franke et al., 2018). Although the actual usage of enterprise architecture artifacts is still not understood comprehensively, and the frameworks and modeling methods used by different organizations somewhat differ (Kotusev et al., 2022), the most commonly used framework is The Open Group Architecture Framework (TOGAF), together with the TOGAF compliant modeling language ArchiMate.

ArchiMate (Lankhorst et al., 2010) is a semi-formal modeling language with which different components, along with their interdependences, can be visually represented and communicated to different stakeholders. These components are represented by dozens of different modeling elements and relationship types between the elements, such as structural, dependency, dynamic, and other relationships. The elements can be viewed from different viewpoints (Strategy, Business, Application, Technology, Physical, Motivation, Implementation & Migration) and structural aspects (Active, Behavior, Passive). ArchiMate comprises two levels of formality: the abstract syntax defining the modeling concepts and their relationships (i.e., the graphical representation of the concepts) and the concrete syntax, which is specific to the notation and semantics of the ArchiMate (i.e., the meaning of the graphically represented concepts) (Bastidas et al., 2021).

2.2. Teaching Enterprise Architecture Modeling

Enterprise architecture is one of the nine core areas of the global competency model for IS curricula (Topi et al., 2017, p. 74), according to which the students should be able to ”design an enterprise architecture (EA). This involves identifying and applying a formal approach to EA development, performing the multistage process of developing an EA, identifying the EA change needs, and applying them to the EA.” Although these competencies concern graduate degree programs, the minimum competency level required from the students concerning EA should be ”novice”, the second level of four altogether (ibid, p. 20). Further, the recent competency model for undergraduate curricula (Leidig and Salmela, 2020) does not include EA as a separate competency area (the 2010 model curriculum does). Yet, it is considered (ibid, p. 29) ”mandatory for graduates to understand the basic concepts of Enterprise Architecture”, and EA appears included in the competency areas of Systems Analysis and Design, IT Infrastructure, IS Management, and Strategy.

Prior research on teaching enterprise architecture is limited. A comprehensive literature review on EA by Kotusev (2017, p. 20) found only 9 articles discussing EA education, and noted that education is among topics that are ”on their peaks of attention”, where researchers ”should understand current problems, questions and controversial points in these established topics”. Seppänen et al. (2020) reported experiences from an undergraduate course on EA modeling, concluding that modeling tools that are strict in EA standard conformance are perceived as easier to learn and use by students than merely illustrative tools with lenient or nonexistent conformance checks, while Gamble (2011) discussed the design and delivery of a course on enterprise integration and enterprise architecture. Simplified EA methods (Kudryavtsev et al., 2018) and mini-projects and cases (Buckl et al., 2010; Lankhorst et al., 2013; Wegmann et al., 2007) for teaching purposes have been proposed, and the essence of teaching EA to graduate (G., 2006) and undergraduate students (Araya-Guzmán et al., 2018) has been discussed. Still, as Kudryavtsev et al. (2018) noted, no universally recognized EA teaching materials exist in higher education, and EA has been among the professions with the lowest levels of specialist training. However, the number of TOGAF 9 certified professionals has steadily grown and reached 100 000 in 2020 (The Open Group, 2020b), which further implies the need for higher education in this field.

3. Research Method and Data

We build our analysis on the errors, mistakes, and shortcomings in the ArchiMate models. While taking an enterprise architecture course, 117 bachelor students created the models that were investigated for this study. The study focuses on two relatively simple learning assignments. In the first assignment, the students were asked to create a model representing an actual learning environment system (later System) that allows teachers to create interactive learning materials and exams, grade the coursework etc., and students to study the materials and perform learning activities of several sorts. The models had to contain five elements and their relationships for each of the three ArchiMate core layers: Technology, Application, and Business. The students had access to the System's technical documentation, showing some details of its container-based implementation on a logical level. This documentation served as the basis for modeling the Technology layer. As all the students were familiar with the System from the end-user perspective, for the Application and Business layers, they were asked to draw from their experiences of the System usage: the application services they had personally used and the corresponding study processes, for example. In the second assignment, the students were asked to create a transition roadmap using the ArchiMate elements from the Implementation & Migration category. The teachers did not define the required contents of this model. The students were only given some examples to feed the ideas, such as an organization migrating from one information system to another, an unspecified business development project, or a student's project in their personal life. We analyzed around 300 ArchiMate models, including those that were returned revised after students were asked to make some corrections to their initial works. Most of our findings, discussed in the next section, are based on the first assignment, which appeared more fruitful for identifying common and generalizable mistakes.

Before the exercises, the students were taught the ArchiMate language version 3.1. These 12 video lectures covered the syntax, semantics, and use cases for each ArchiMate modeling element. The lectures also contained several examples of how the ArchiMate elements are used in different contexts to represent different layers and domains of EA and different categories of models. Most students created their models using the Archi tool, which offers comprehensive built-in documentation of the ArchiMate language (c.f., Seppänen et al., 2020).

The first author logged all the mistakes in the students' models. First, these included syntactic errors, which were identified by comparing the models against the ArchiMate language specification. Then, the semantic mistakes were identified by evaluating each model as a whole and checking if its information contents were correct and meaningful in relation to the modeling assignment. This data were then further analyzed. First, one-off mistakes and random anomalies were removed, and then the remaining mistakes were classified into the categories discussed in the next section. The analysis was data-driven, i.e., the categories were identified from the data, and no preconceived categories were used to guide the analysis. Next, the validity of the analysis was assessed by the second author, who re-evaluated approximately half of the models and categorized the mistakes into the categories created during the previous phase of the analysis. Finally, the authors compared their notes and agreed on the analysis results.

4. Findings

This section discusses the typical mistakes found in the students' ArchiMate models. We also propose some practices for avoiding these mistakes and, where applicable, comment on the mistakes' practical implications.

4.1. Readability

Some of the models were disorganized and visually unclear (Figure 1). In these cases, the modeling elements were placed on the canvas randomly and not aligned on either X or Y axes. A good practice for creating layered ArchiMate models is to follow the horizontal order of the layers of the ArchiMate framework: e.g., the Business elements are on top, the Application layer elements are in the middle, and the Technology layer elements are at the bottom.

Relationship connectors often crossed each other and passed over modeling elements, making models visually unpleasant and challenging to read. While this is a minor issue in models with only a few

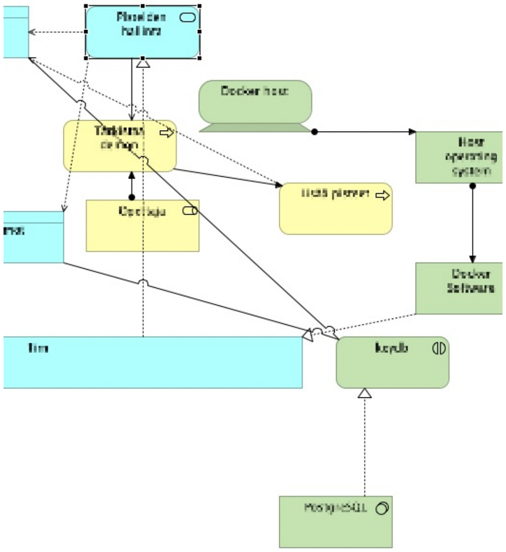

Figure 1. Poor visual readability.

modeling elements, it effectively impairs the readability of the more complicated and information-heavy models. Most modeling tools provide manual or automated functionality for arranging and curving connectors to avoid the above.

The above issues seem to be in line with previous research. Blumenthal (2007, p. 63) notes that the ”problem is EA information often is unintelligible. The necessary data might be there, but the presentation is so poor that the decision-maker’s ability to use it is impaired”. As discussed by Arnautu and Degenais (2021), the previous literature on decision-making indicates that decision-makers favor understandable, user-friendly, easy-to-access, visually engaging documents that can be quickly examined and interpreted. Enterprise architects should not overlook the visual readability of the models, as it seems to be a significant issue for decision-makers and somewhat easy to fix.

4.2. Stakeholder Information Need

Several models were overly information-rich and complicated. They contained elements not needed to fulfill the given assignment, i.e., they were answering the questions that were not asked. This possibly reflects students’ aspirations to showcase what they have learned. Still, in real-life cases, such models would not target the exact information needs of the stakeholders but could instead appear confusing. Previous studies Kotusev et al. (2023) have also criticized EA models and artifacts for containing irrelevant informational content.

On the other hand, some models denoted that either the assignment or its background material was wrongly understood. Therefore, the resulting models either lacked some relevant information or contained erroneous information. This would be problematic as an accurate understanding of data can be considered a prerequisite for decision-making (Dy et al., 2021). As improved decision-making is among the most often-mentioned benefits of EA (Kurnia et al., 2021) and EA is supposed to produce valuable and usable information for decision-makers, poor readability and either lack of or excessive information in the models is a considerable problem. Prior research notes the limited perceived usefulness of EA visualizations, which are often characterized by their complexity, lack of focus, and inappropriate level of abstraction, which inhibits their effective use for decision-making and leads to a low added value perceived by stakeholders (Rehring et al., 2019). These issues can be due to either modeling mistakes or other factors, such as the inherent qualities of the modeler, model, or the organization modeled.

4.3. Level of Abstraction and Granularity

The elements used in the models were often inconsistent in granularity, i.e., how large an entirety one modeling element represents. For example, there were Behavioral elements corresponding to a large UML use case. In contrast, in the same models, other elements were rather comparable to a single activity in a UML activity diagram. (Figure 2.) While this may not be a considerable problem regarding the understandability of a single model, should we assume that the elements drawn in a model populate a shared modeling repository, the problem can escalate over time, and the reuse of these elements would continue to feed the confusion. This will deteriorate the repository and the usability of its content. The students had previously studied UML, and it appeared that, in some cases, this negatively affected their fluency in ArchiMate modeling. The students mixed and matched the different levels of abstraction, granularity, and purposes of different UML diagramming techniques while working with the ArchiMate language. Kotusev et al. (2023) reviewed prior research and concluded that EA artifacts are often deemed excessively complex, have the wrong level of detail or are overly conceptual. Consequently, over half of the organizations practicing EA report unsatisfactory quality of EA documentation as a significant challenge for their EA practices (ibid).

While not as diverse as UML, there are different purposes for different ArchiMate models. It is advisable to hide unnecessary details for some purposes while representing details is necessary in other cases. In

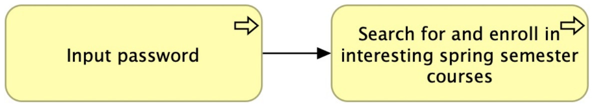

Figure 2. Imprudent and inconsistent granularity.

the ArchiMate, nesting elements allow alternative graphical notation to express structural relationships (e.g., Composition and Aggregation) or, for example, Specialization. The nesting can be used to vary the level of detail per purpose while still retaining consistent granularity of the elements stored in the repository. By this means, for example, one model can contain only an overall business process flow with details hidden, while for other use cases and information needs, another model can represent each of its sub-processes in detail. However, some students used element nesting that did not follow the ArchiMate language specification. For example, an Application Component was nested inside a Business Actor or a Business Process. Such a practice may appear visually reasonable and carry some intuitively understandable semantics, such as the business process ”Enroll in a course” uses the application ”Course Management System” (Figure 3). However, this does not follow the ArchiMate specification, and thus the interpretation may vary per reader. There is no right nor wrong way to decode the model's meaning, and such visual nesting would not result in any intelligible and storable information to the model repository about the elements' relationship.

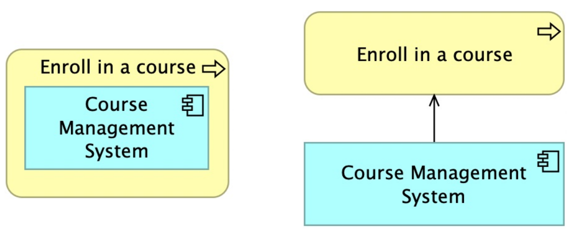

Figure 3. Left: Meaningless visual nesting. Right: A correct way to represent the relation with Serving.

4.4. Naming Conventions

Naming conventions have been discussed in several areas of computer science. Programming style guidelines give recommendations for function and variable naming, and, for example, Oracle naming conventions are commonly used as a standard for database naming (Langer, 2007). The ArchiMate standard also gives explicit recommendations for naming the elements (The Open Group, 2020a). For example, it is advised to name the Service elements using a verb ending with ”ing” or, alternatively, to use a name explicitly containing the word ”service”, the Business Actors elements should be named using a noun, and so forth. These recommendations were quite often overlooked. As the graphical ArchiMate notation indicates which elements represent services and which are business actors, it is understandable that modelers easily neglect to follow the naming conventions.

The second form of neglecting good naming conventions was to use the names of the element types as the names of the elements. For example, there were several Application Interface elements named as ”Application Interface” or “API”, and Data Objects named ”Data”. Again, these names may be decipherable in the context of a single model, but that would not be the case if these elements were stored in a multi-user repository. This is similar to naming a table in a relational database as ”Table”. A related mistake was using overly generic names that cannot be understood independently of the context. These, for example, might have been Active Structure elements named as ”Received” or ”Checking”. While such vague names may serve their purpose in the context of a specific model, they offer little information for another modeler reusing the contents of a shared modeling repository.

The third type of naming-related mistake was using plural nouns as element names. Once again, plural names, such as ”Standard APIs”, can serve purposes of simplified visualization by hiding details, but they convey very little usable information. Instead, it would be recommendable to first model each type of these ”standard APIs” and then, if necessary for presentational purposes, to aggregate them into a collection with the Aggregation relationship. (Figure 4.)

According to Gustafsson (2007), groups develop a group-specific verbal and graphic discourse that is not easily understood by outsiders and have a ”tendency to match each other in a choice of words, syntax, and semantics during verbal dialogue” (Healey et al., 2007, p. 286), creating language-specific communities of knowing. This can be interpreted in several ways in the context of this study, one of which would be to interpret ArchiMate itself as a group-specific verbal and graphical discourse, which is not easy for the students to grasp, and, therefore, for example, naming conventions are violated.

Finally, only some of the students' models contained names or annotations for the relationship connectors that would significantly improve the understandability of a model. More importantly, when using the Flow relationship, it is often necessary to indicate the content of the information flow. (Figure 5.)

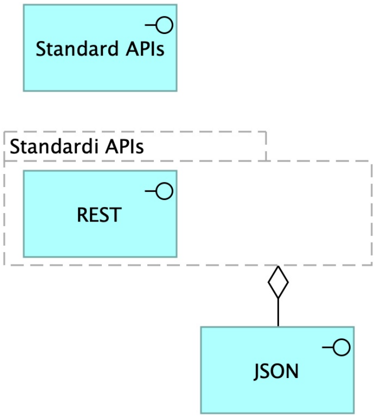

Figure 4. Top: Element name in plural. Bottom: A correct way to present groups with Aggregation relationship.

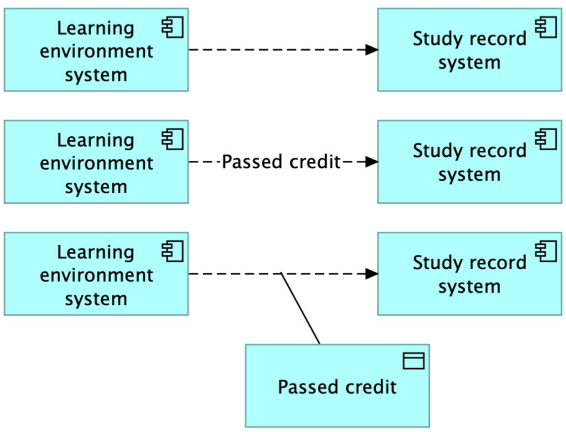

Figure 5. Top: Ambiguous Flow relationship. Bottom: Correct ways to represent information flow content.

4.5. Core Framework Aspects, Relationships and Element Types

For a layered architecture model, such as the one created in our first learning assignment, it is advisable to consistently follow the active-behavior-passive structure (referred to as Aspects in the ArchiMate framework). Several models featured ArchiMate's behavior aspect elements without expressing the Realization relationship from the active structure element. Sometimes it is feasible to omit the aspects that do not provide immediate value in the model. A typical example is a services catalog. It gives the audience a quick look at, for example, the key business services without obscuring the presentation with unnecessary details, such as the Business Processes and the Business Actors that realize the services. Another example is modeling Application Services without providing information on what Application Components realize these services. Even though this information would not be of interest in every EA visualization, it must not be completely omitted. Also, several wrong or unfit relationship connections appeared in the models (e.g., Application Service serving directly Business Actor, instead of Business Process), and some missed representing the required relationships altogether.

Visual similarities of different ArchiMate elements seemed to cause some confusion among the students. For example, some students used the light green three-dimensional box type representation of the Plateau element when they should have used Node, which can be represented with the darker but also green three-dimensional box element. While the Plateau and Node elements are visually reasonably similar in the notation, their semantics are completely different. It sometimes appears necessary to remind the learning modelers, and why not also more advanced practitioners, about the syntactic and semantic formalities. The ArchiMate language is more than just a graphical notation to support the verbal communication between stakeholder groups, as hinted by Bastidas et al. (2021).

Finally, some modeling elements were used for apparently wrong purposes with no relation to the abstract syntax (c.f., Bastidas et al., 2021). These mistakes mostly took place with the concepts of technology architecture. For example, Docker containers might have been modeled using Technology Process elements. Although it is far from obvious how a container technology is correctly modeled with ArchiMate (this topic has been discussed, for example, in the Open Group's ArchiMate User Community), the ArchiMate specification might indicate that the Technology Process is not the correct element: "A technology process represents a sequence of technology behaviors that achieves a specific result." However, this can also be regarded as a limitation of the ArchiMate language. The intentionally high level of abstraction cannot directly and explicitly address, for example, all the possible details of current technology architectures. Therefore, it is necessary for enterprise architects and modelers to either keep a close eye on the recent discussions of the professional community or to define the modeling rules for their organizations to ensure the consistency of the models.

4.6. Summary

Figure 6 depicts the categorization and distribution of more than 600 mistakes found in the students' models, with repeated mistakes within the same model excluded. The predominant category, comprising 36% of all errors, pertained to incorrect or incomplete utilization of the ArchiMate language. These manifested through elements being employed in wrong manners or inappropriate contexts, erroneous use of relationship types, or omission of some required relationship connectors altogether, among others. The second largest category, accounting for 23% of mistakes, encompassed deficiencies in the model's capacity to correctly address the information need expressed in the assignment. The third most large category encompassed 21% of the mistakes, and they related to the consistency and purposeful level of abstraction and elements' granularity. Finally, the issues in element naming and the challenges related to the readability and clarity of models both comprised 10% of the mistakes.

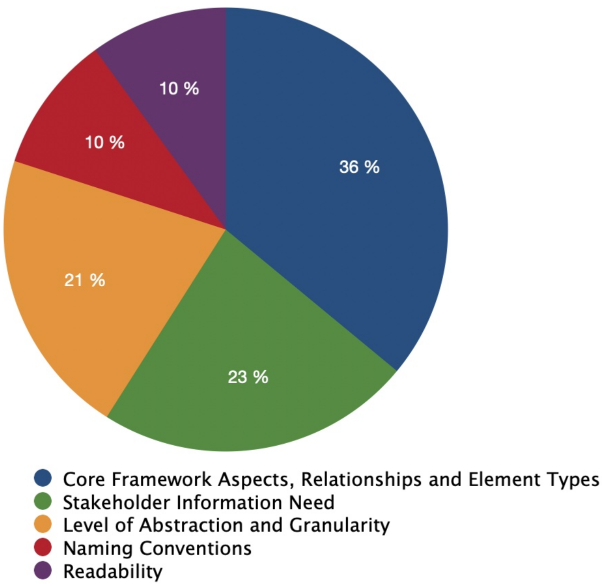

Figure 6. Distribution of mistakes in categories

5. Discussion

Our study revealed several topics to consider when teaching and learning to model and communicate enterprise architecture using the industry-standard ArchiMate language. First, it must be remembered that the EA models and other artifacts have no intrinsic value. They only carry extrinsic, instrumental worth. The models should be created for a need, and they must convey extractable information that supports maintaining and developing organizations' purposeful, effective, and secure architecture. As noted already in the late 1970s, information needs vary per decision-maker (Rockart, 1979), and understanding these needs is essential for any enterprise architect. The models must not be saturated with unnecessary information or ignore any relevant information. EA artifacts are often deemed to be excessively complex. They might have irrelevant informational content, the wrong level of detail, and an overly conceptual nature, hindering their use and leading to major challenges for the organization's EA practices (Kotusev et al., 2023). Similar results have also been found in other areas, as understandable, user-friendly, easy-to-access, and visually engaging, quickly interpreted documents are favored in decision-making (Arnautu and Degenais, 2021). Regarding EA work, prior studies (Nurmi et al., 2019) have indicated that relevant factors include e.g., co-creation and needs-based utilization, which supposedly also benefit from clear and conceptually sound artifacts.

Second, it must be acknowledged that ArchiMate models and other EA artifacts have different stakeholders and different uses. For example, a deliverable created for software development project stakeholders requires more detailed information content and strict adherence to the syntactic specification of the language than those delivered to, for example, a company's strategy workshop. In either case, the models should be expressive, communicative, and easy to interpret, yet syntactically and semantically correct. When using semi-formal languages, such as ArchiMate, it is often easier to abstract and hide unnecessary details from a model afterward than to add those details and correct the mistakes of a sketchy model.

Third, although it is possible to extend ArchiMate's basic properties with, for example, new modeling elements, use the language in "creative" manners, and adapt it with other means to fit the group-specific needs (c.f., Healey et al., 2007), it is first necessary to master the basics of the language and meticulously study the need for a specific extension or adaptation before taking such endeavors. ArchiMate is a standard and, as such, a universal language of enterprise architecture. Therefore, any organization-specific deviations from the standard must be communicated to the stakeholders, especially external vendors, subcontractors, and consultants, to ensure that all are still speaking "the same language".

Overall, our analysis of the typical mistakes in models is in line with some prior research, as the students' modeling errors aligned with weaknesses associated with the EA models in critical reviews (c.f., Kotusev et al., 2023), such as poor readability, unfit level of abstraction and either lack of or excessive information in the models. Although some of these issues (i.e., poor readability) were due to students' mistakes, there seems to be a potential overlap between the students' modeling errors and the inherent characteristics of EA models. Thus, future studies should form a deeper understanding of the extent to which the identified issues reflect errors and the portion that can be attributed to the natural characteristics of the modeling process and the modeling language used. This could facilitate the development of improved modeling practices and enhance the effectiveness of enterprise architecture and its usage in organizational development. For example, how do these mistakes affect the use of the models in different situations and by different stakeholders, in terms of, e.g., the rhetorical effects and the communicative aspects of the models? Is there a linkage between the presentation forms of the information content and the use of the information? Are the models with typical mistakes still comprehensible to other architects or even a part of the group-specific verbal and graphic discourse?

There are some limitations to this study. First, the data collection was done in a single Finnish university with a single cohort of students. The curriculum of our university is based on the IS model curricula, and EA has been a part of our teaching for many years. Although the modeling mistakes presented in this study were similar to the ones noted by the prior literature, the contents of the courses and the specific teaching methods used may vary between universities, and thus, the results of the modeling exercises may vary. For example, depending on the modeling tool used by the students and the specific assignments given, the results may vary if the study is repeated in different circumstances.

6. Conclusions

Enterprise architecture professionals strive to produce valuable and usable content for different purposes, including decision-making and communication with stakeholders. Still, these goals are hindered, among others, by poor modeling. Despite this, studies on teaching EA modeling are limited, and standardized teaching practices are lacking.

In this paper, we examined *what kind of modeling mistakes are typical for students and how do these align with prior research on EA modeling mistakes by professionals?* Our data indicate that the typical mistakes concern (1) the readability of the models, (2) stakeholder information needs, i.e., models being overly information-rich and complicated or lacking relevant information, (3) level of abstraction and granularity, i.e., models showing a poor understanding of the entities modeled, (4) naming conventions, i.e., either lacking names or annotations, or naming elements in a confusing manner, and (5) core framework aspects, relationships and element types, i.e., the abstract and the concrete syntax of the models were misunderstood. As discussed earlier, these mistakes seem to be in line with the prior research on enterprise architecture artifacts and modeling mistakes of professional architects. The potential consequences of the mistakes were discussed, and these implications as a whole can be summarized by the notion that over half of the organizations practicing EA report unsatisfactory quality of EA documentation as a significant challenge for their EA practices (Kotusev et al., 2023).

Prior research on teaching enterprise architecture modeling in higher education (Seppänen et al., 2020) suggests that tool support is vital for learning EA modeling, and while simpler tools seem to facilitate learning better, more complex tools might, in turn, facilitate real work environments more comprehensively. This would foster deeper learning experiences and, as stated by Seppänen et al. (2020), help students understand a phenomenon rather than a tool. As previously discussed, EA models and other artifacts have no intrinsic value, and their form and use vary from context to context. For teaching, along with our results, this might have several implications. First, as almost half of the mistakes made by the students were related to core framework aspects, relationships, element types, and naming conventions, using simpler tools that enforce the proper use of the notation might be favorable. Second, as the other half of the mistakes were related to the stakeholder information needs and the informational content itself, i.e., to the phenomenon rather than the tool or the notation, teaching emphasis should be put on the wider understanding of enterprise architecture itself. This resonates with the changes made in the IS2020 curricula Leidig and Salmela (2020), where EA content is included in several competency areas, such as Systems Analysis and Design, IT Infrastructure, IS Management, and Strategy. Undergraduate EA courses should, therefore, probably include lessons on modeling as well as knowledge and know-how on how to integrate the contents of courses on, e.g., systems analysis and IS management. Further, although educators can use our findings of typical modeling mistakes when planning their teaching and guiding their students on modeling, the emphasis should be put on a wider ”philosophical” understanding of what kind of mistakes are usually made during EA work and why it is important to avoid them. This would include communications and rhetorical aspects, an understanding of psychological aspects, such as the cognitive fit of different kinds of informational content, as well as a managerial understanding of the actual aims of EA work beyond the modeling itself.

EA is currently one of the most common ways to produce valuable and usable information for decision-makers regarding, e.g., business-IT alignment. Therefore, we argue the role of EA should be more prominent in IS curricula. This could include general courses on EA, such as the one reported in this study, as well as specific courses on EA modeling, applied courses including real-life projects, and courses that would discuss relevant themes from other perspectives, such as management, cognitive sciences, and psychology. In practice, this could mean applying different methods of teaching and a wider array of assignments. As an example, poor visual readability and excess information were both issues that were present in our data and that are often mentioned in the prior literature. From an educational perspective, this might be due to the fact that the exercises that the students pursue are often quite simple. That was also the case in our course. As previously noted, simple models with only a few elements do not necessarily show why poor readability and excess information are problems. We would thus suggest including assignments such as ones related to comprehending information from complicated and information-heavy models, assignments where the students should convert overly information-rich documents into models that are relevant to different contexts and stakeholder groups, as well as assignments that would include real-life projects, where the effects of different modeling mistakes are easily seen. While more complex models might be used to illustrate the need for clear and stakeholder-specific models, it is also important to note that they might have different pedagogical uses than simpler models. Complex models might have negative effects on students learning (c.f., Taipalus, 2020) if they are used to teach basic skills such as modeling notations and their use. Consequently, complex models should be used to apply these skills to more advanced situations, as well as to teach skills like interpreting information for specific contexts. These pedagogical changes could further motivate students to comprehend the positives of good modeling (c.f., Franke et al., 2018; Arnautu and Degenais, 2021) as well as the negative sides of lousy models (c.f., Rehring et al., 2019) and poor visualization (c.f., Eberhard, 2023).

References

Araya-Guzmán, S., Cares-Monsalves, L., Ramirez-Correa, P., Grandón, E. E., & Alfaro-Perez, J. (2018). Enterprise architecture proposal for undergraduate teaching in higher education institutions. *Journal of Information Systems Engineering and Management*, (3(3)), 21.

Arnautu, D., & Degenais, C. (2021). Use and effectiveness of policy briefs as a knowledge transfer tool: A scoping review. *Humanities and Social Sciences Communications*, (8(1)), 1–14.

Bastidas, V., Reychav, I., Ofir, A., Bezhradica, M., & Helfert, M. (2021). Concepts for modeling smart cities: An archimate extension. *Business Information Systems Engineering*, 1–15.

Blumenthal, A. N. (2007). *The long view: Enterprise architecture plans are useless without clear, relevant information*. https://www.govexec.com/management/2007/05/the-long-view/24486/ (accessed: 06.10.2023)

Buckl, S., Poorthuis, A., & Joyce, S. (2010). Teaching enterprise architecture management with student mini-projects. *Software Engineering 2010 Workshopband (inkl. Doktorandensymposium)*.

Dy, B., Ibrahim, N., Poorthuis, A., & Joyce, S. (2021). Improving visualization design for effective multi-objective decision making. *IEEE Transactions on Visualization and Computer Graphics*, (28(10)), 3405–3416.

Eberhard, K. (2023). The effects of visualization on judgment and decision-making: A systematic literature review. *Management Review Quarterly*, *73*(1), 167–214.

Franke, U., Cohen, M., & Sigholm, J. (2018). What can we learn from enterprise architecture models? an experiment comparing models and documents for capability development. *Software & Systems Modeling*, *17*, 695–711.

G., S. (2006). Reflections on teaching enterprise architecture to graduate students. *MCIS 2006 Proceedings, 296*.

Gamble, M. T. (2011). Teaching enterprise integration and architecture – tools, patterns, and model problems. *AMCIS 2011 Proceedings - All Submissions. 417*.

Gustafsson, R. (2007). Strategy work with artifacts: Neurocognitive advantages of visual sense-building. *Cognitive Aids in Strategy. Emerald.*

Healey, P. G., Swoboda, N., Umata, I., & King, J. (2007). Graphical language games: Interactional constraints on representational form. *Cognitive Science*, (31(2)), 285–309.

Kaisler, S. H., Armour, F., & Valivullah, M. (2005). Enterprise architecting: Critical problems. *Proceedings of the 38th Annual Hawaii*

International Conference on System Sciences. IEEE., 224b–224b.

Kotusev, S. (2017). Enterprise architecture: What did we study? *International Journal of Cooperative Information Systems*, (26(04)).

Kotusev, S., Kurnia, S., & Dilnutt, R. (2022). The practical roles of enterprise architecture artifacts: A classification and relationship. *Information and Software Technology*, (147).

Kotusev, S., Kurnia, S., & Dilnutt, R. (2023). Enterprise architecture artifacts as boundary objects: An empirical analysis. *Information and Software Technology*, (155).

Kudryavtsev, D., Zaramenskikh, E., & Arzumanyan, M. (2018). The simplified enterprise architecture management methodology for teaching purposes. *Enterprise and Organizational Modeling and Simulation: 14th International Workshop, EOMAS 2018, Held at CAiSE 2018, Tallinn, Estonia, June 11–12. Selected Papers 14. Springer International Publishing.*, 76–90.

Kurnia, S., Kotusev, S., Shanks, G., Dilnutt, R., & Milton, S. (2021). Stakeholder engagement in enterprise architecture practice: What inhibitors are there? *Information and Software Technology*, (134).

Langer, A. M. (2007). *Analysis and design of information systems*. Springer Science Business Media.

Lankhorst, M. M., Proper, H. A., & Jonkers, H. (2010). The anatomy of the archimate language. *International Journal of Information System Modeling and Design (IJISMD).*, (1(1)), 1–32.

Lankhorst, M. M., Proper, H. A., & Jonkers, H. (2013). Enterprise architecture specification case study. *Journal of Information Systems Education*, (24(2)), 105–120.

Leidig, P., & Salmela, H. (2020). Is2020 a competency model for undergraduate programs in information systems. *The Joint ACM/AIS IS2020 Task Force.*

Nurmi, J., Penttinen, K., & Seppänen, V. (2019). Towards ecosystemic stance in finnish public sector enterprise architecture. *Perspectives in Business Informatics Research: 18th International Conference, BIR 2019, Katowice, Poland, September 23–25, 2019, Proceedings 18*, 89–103.

Rehring, K., Greulich, M., Bredenfeld, L., & Ahlemann, F. (2019). Let's get in touch-decision making about enterprise architecture using 3d visualization in augmented reality. *Proceedings of the 52nd Hawaii International Conference on System Sciences 2019.*, 1769–1778.

Robl, M., & Bork, D. (2022). Enterprise architecture management education in academia: An international comparative analysis. *Complex Systems Informatics and Modeling Quarterly.*, (31), 29–50.

Rockart, J. F. (1979). Chief executives define their own data needs. *Harvard Business Review*, (57(2)), 81–91.

Seppänen, V., Heikkilä, J., & Liimatainen, K. (2009). Key issues in ea-implementation: Case study of two finnish government agencies. *IEEE Conference on Commerce and Enterprise Computing. IEEE.*, 114–220.

Seppänen, V., Pulkkinen, M., Taipalus, T., & Nurmi, J. (2020). Information systems students' impressions on learning modeling enterprise architectures. *2020 IEEE Frontiers in Education Conference (FIE). IEEE.*, 1–8.

Shanks, G., Gloet, M., Someh, I. A., Frampton, K., & Tamm, T. (2018). Achieving benefits with enterprise architecture. *The Journal of Strategic Information Systems*, *27*(2), 139–156.

Taipalus, T. (2020). The effects of database complexity on sql query formulation (journal-first). *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 185–185.

Tamm, T., Seddon, P. B., & Shanks, G. (2022). How enterprise architecture leads to organisational benefits. *International journal of information management*, *67*, 102554.

The Open Group. (2020a). *Archimate® 3.2 specification*. https://pubs.opengroup.org/architecture/archimate32-doc.singlepage/ (accessed: 06.10.2023)

The Open Group. (2020b). *Togaf® 9 certification passes 100,000 milestone!* https://blog.opengroup.org/2020/06/23/togaf-9-certification-passes-100000-milestone/ (accessed: 06.10.2023)

Topi, H., Karsten, H., Brown, S. A., Carvalho, J. A., Donnellan, B., Shen, J., Tan, B. C. Y., & Thouin, M. F. (2017). Msis 2016 global competency model for graduate degree programs in information systems. *Communications of the Association for Information Systems*, (40(18)).

Wegmann, A., Regev, J. D., G.and De la Cruz, Lê, S., L., & Rychkova, I. (2007). Teaching enterprise and service-oriented architecture in practice. *Journal of Enterprise Architecture*, (4), 15–24.