UNIVERSITY OF APPLIED SCIENCES

THM
TECHNISCHE HOCHSCHULE MITTELHESSEN

Susanne Müller

The Role of Modelling in Organization and Business Informatics

THM-Hochschulschriften Band 30

Susanne Müller

The Role of Modelling in Organization and Business Informatics

THM-Hochschulschriften Band 30

THM-Hochschulschriften Band 30
© 2024 Susanne Müller
Technische Hochschule Mittelhessen
Fachbereich Wirtschaft - THM Business School

Herausgeber der THM-Hochschulschriften:
Der Präsident der Technischen Hochschule Mittelhessen

Alle Rechte vorbehalten, Nachdruck, auch auszugsweise, nur mit schriftlicher Genehmigung und Quellenangabe.

Die Hochschulschriften sind online abrufbar:
www.thm.de/bibliothek/thm-hochschulschriften

ISSN (Print) 2568-0846
ISSN (Online) 2568-3020

The Role of Modelling in Organization and Business Informatics

Prof. Dr. Susanne Müller
Technische Hochschule Mittelhessen - University of Applied Sciences
THM Business School
Gießen, 20.06.2024

Abstract: In this paper the role of visual formal modelling in the subjects of Organization (i.e. Business Process Management) and Business Informatics (i.e. Software Development) is stated. Based on this we discuss which impact actual developments, i.e. agile work practices and generative AI, have on this role. For generative AI the actual capabilities on visual formal modelling of the best-known AI-tools experimentally are examined. Synthesizing, an outlook on the future application of visual formal modelling by humans is given. The consequences for the future adaption of teaching in the field of visual formal modelling in Business and Informatics curricula (and in related interdisciplinary subjects) are pointed out.

Contents

1 Introduction 4
2 Actual developments and their influence on the usage of Modelling 6
  2.1 Agility 6
  2.2 Low Code and Smart Software Development 7
  2.3 NoSQL databases 7
  2.4 Process Mining 8
  2.5 Generative Artificial Intelligence 9
3 Results of an experiment: Generative AI solving modelling tasks 10
  3.1 Entity relationship modelling task of low difficulty. 10
  3.2 Process modelling task of low difficulty 12
  3.3 Entity relationship modelling and process modelling tasks of medium difficulty 13
  3.4 Submit ideas for “improving” the generated models 15
  3.5 Consideration of learning and ongoing development of generative AI 17
4 Prediction of the future use of Modelling in Organization and Business Informatics 21
5 Implications for teaching Modelling in Business and Informatics curricula 21
References 23
Table of figures 25

1 Introduction

Visual formal modelling techniques cannot be thought away in the subjects Organization and Business Informatics. Modelling techniques in these subjects are known to be used for the creation of formal graphical models that represent the reality in the subject areas of certain projects, i.e.

- software development projects,
- business process management projects and
- projects to create or evaluate enterprise architectures[1].

The advantages of using visual formal modelling techniques especially when representing requirements in software development (including database design) have been highlighted for a long time already. The graphical representation of the models contributes to their cognitive effectiveness[2]. It makes them understandable, also by people with a background from the operational business that know IT from the perspective of an end user. On the other hand, when using a suitable tool, the created conceptual models are formal enough to automatically translate them into code fragments or database design and from there into executable code and an operational database. To manage complexity with the help of models was part of the CASE movement from the dawn of this idea in the late 80ies[3].

In Business Process Management (BPM) projects models are used to visualize the way processes flow through organizations. This gives hints on improvement potential. Once a model of a process is there, detailed information can be added in a way that makes simulations possible. Workflow Software Packages use business process models of different kinds to first configure and then directly support the modelled processes. Process engines simulate modelled processes and identify problems in process execution.

Additionally, in enterprise architecture projects the creation of company-wide models (or business unit models) is expected to give enterprises hints on the question if it is compliant to certain standards. Also, an enterprise can conclude from models if it is easy for them to introduce certain practices or standard software by comparing the own models to reference models, i.e. of standard software packages or industry sectors.

The expectations to the benefits of modelling on the efficiency of the software development process and on quality of the produced software were huge. The intention of MDE (Model Driven Engineering) is to use models as main software development artefacts beginning with the requirements engineering[4]. The term Model Driven Development (MDD) is narrower, as it comprises the part of Software Engineering falling upon the pure development process. Expectations reached up to using models as sole artefacts and to update them automatically, also if modifications are made directly in the code (Roundtrip Engineering). Some of the expectations were

---

[1] (Fettke, How Conceptual Modeling Is Used, 2009, p. 575)
[2] (Moyano, Pufahl, Weber, & Mendling, 2022, p. 2)
[3] (Bucaioni, Cicchetti, & Ciccozzi, 2022:21, p. 1959)
[4] (Bucaioni, Cicchetti, & Ciccozzi, 2022:21, p. 1959)

disappointed. In the course of time many experts have adopted the opinion that Roundtrip Engineering does not work[5]. Also, the CASE trend was subject to negative publicity. The American government in the beginning of the 1990ies spent millions in CASE-projects. Soon, the US government institutions became well recognized critics of CASE-tools. In a report of 1993 the Government Accountability Office (GAO) expressed severe doubts that CASE-tools improve software development productivity or lead to improved software quality[6]. Additionally, the report contained criticism to the procurement process for I-Case-tools at Defense and recommended a more restrictive process based on assessments and pilots for the purchasing process of I-Case environments.[7]

As the ability of abstraction by creating and working with models is considered being indispensable for graduates of Business and Informatics degree courses, which is also true for any interdisciplinary degree courses in these subjects (i.e. Business Informatics), modelling has become a steady component of teaching contents in the corresponding curricula. This is especially true for Germany[8]. Students are guided to create models that ignore irrelevant technical details during the first development phases[9] by using the standardized and market leading modelling languages[10] during their studies at least once, i.e. in software engineering or business process modelling teaching modules, but also in basic modules in business courses. This is true for graduate degree courses as well as for post-graduate degree courses. The main standards taught in Germany are Unified Modelling Language (UML) for software development[11], Entity Relationship Modelling (ERM, different representations) for information/data structures and Business Process Model and Notation (BPMN) or Event Driven Process Chains (EPCs) in modules teaching Business Process Management.

In the light of actual developments, the question comes up, if the role of modelling will change. Modelling by humans may become less important

- if it can also be done by generative AI,
- if the creation of models is viewed to be an unnecessary intermediate step that serves mainly for documentation and slows work processes down or
- if models can be created automatically by corresponding software tools.

Already in year 2009 in a survey on the practical use of techniques for conceptual modelling it struck the eye that there was no technique that was going to be used significantly more in the future[12]. One possible explanation at that time was the maturation of modelling techniques, another explanation is that conceptual modelling simply since then is practiced less.

---

[5] (Siegl, 2015)

[6] (GAO United States General Accounting Office - Information Management and Technology Division, 1993, pp. 4-5)

[7] (GAO United States General Accounting Office - Information Management and Technology Division, 1993, p. 11)

[8] (Fettke, How Conceptual Modeling Is Used, 2009, p. 572)

[9] (Cabot & Clarisó, Low Code for Smart Software Development, 2023, p. 89)

[10] (Fettke, How Conceptual Modeling Is Used, 2009, pp. 572-573)

[11] (Kazi, 2019, p. 15)

[12] (Fettke, How Conceptual Modeling Is Used, 2009, p. 574)

This paper discusses implications of actual trends on the use of modelling and the consequences the findings have on how and how much teaching of modelling is needed in graduate and post-graduate degree curricula.

2 Actual developments and their influence on the usage of Modelling

We consider the following developments as having a potential influence on the usage of visual formal modelling:

- Agility
- Low Code and Smart Software Development
- NoSQL-Databases
- Process Mining
- Generative Artificial Intelligence

2.1 Agility

Agility came up with the publication of the agile manifesto in 2001[13] and is practiced in software development projects as a principle for accelerated customer-oriented software delivery processes and in project management as organization principle for fast progress and product owner driven requirements orientation. Agility is presented as being the opposite of the so-called waterfall-model. The waterfall-model with its phases since the emergence of agility often is figured as being cumbersome and outdated. Many publications leave aside the fact, that in practice waterfall-model projects were carried out in smaller chunks, i.e. called early deliverables, compensating thereby the pitfalls of the waterfall-model already before the term agility was brought up. There are approaches to combine agility and the waterfall model into a “Disciplined Agile Delivery “-framework for software development[14]. Apart from the general critics to agile principles expressed right from the beginning[15] there may be also a different way agility is practiced in large firms and in SMEs. There is no doubt that organizational agility in SMEs is of high value because of the ability to rapidly react to changes in an enterprise’s business environment and to recognize opportunities[16]. However, we see the danger that in smaller enterprises agility in project work may be interpreted as an approach meaning “let us just start and have a look what comes out in the end”. This practice of agility can be viewed critically, amongst others because non-conceptual working may be fostered. There is only few research on the difference of adoption of agile methodologies in large and in small enterprises. There are hints that smaller enterprises assign roles when using SCRUM less formally, i.e. by assigning the roles of Product Owner or Scrum Master as additional roles to team members whereas in larger companies these roles are performed by dedicated persons. Additionally, the team availability seems to be more important in small companies than in large companies[17].

The use of modelling corresponds a lot with putting emphasis on thorough conceptual work. Modelling can be viewed as contradictory to the agile principles and as being a sluggish approach that impedes achieving the objectives of agility, i.e. speed,

---

[13] (Beck, 2001)

[14] (Kazi, 2019, p. 15)

[15] (Misra, Kumar, Kumar, Fantazy, & Akhter, 2011, pp. 977-978)

[16] (Claudhary & Trzcielinski, 2021, p. 50)

[17] (Petrescu & Motogna, April 2023)

responding to change and waiving on documentation[18]. A strong argument for minimizing the usage of models is the fact that software in the agile world changes constantly[19]. So, if modelling is to be accepted in the agile world it needs to be practiced in an adaptive and adaptable manner[20]. Otherwise, practitioners of agility leave visual formal Modelling aside and use more non-formal modelling techniques instead, i.e. cooperative modelling[21]. Still, actual studies reveal that the advantages of modelling, particularly of business process modelling, make practitioners of agile software projects appreciate this approach and use it mainly at the beginning of the agile software development process[22].

2.2 Low Code and Smart Software Development

The Low Code term stands for platforms that allow the creation of applications by visual modelling with only a minimum of writing lines of program code. One of the objectives is to have persons from the domain context who are technically experienced but have no programming skills develop applications by using a building block principle. Employees like that are denoted as citizen developers. Some “new” platforms in this market emerged from rebranding platforms that were known as Model Driven Engineering (MDE) platforms before[23]. Websites promoting Low Code platforms actively refer to the term of Model Driven Development (MDD)[24]. Statements there sound very similar to the ones stated about MDE and even CASE before[25]. Low Code can be considered a continuation of model-based and model-driven approaches, which again are the continuation of Computer Aided Software Engineering (CASE)[26]. A challenge is now the easy integration of Low Code Platforms with AI components that are based on Machine Learning (ML) techniques to achieve a so-called Smart Software Development[27] within Low Code platforms. Summarizing, the Low Code movement is considered being a maturation step of visual formal modelling like practiced in Model Driven Engineering (MDE)[28].

2.3 NoSQL databases

NoSQL databases are databases that follow a non-relational approach in storing data. The emergence of NoSQL databases is linked to the term Big Data. The visual modelling of information structures was triggered by the introduction of the Entity Relationship Model (ERM) in 1976 by Chen[29]. Since then, diverse variants and slight improvements of this modelling technique for data structures have spread rapidly. ERM modelling is tightly linked to the relational database model, for ERMs easily can automatically be transformed into relational structures. As relational database models

---

[18] (Moyano, Pufahl, Weber, & Mendling, 2022, p. 2)
[19] (Kazi, 2019, p. 22)
[20] (Kazi, 2019, p. 22)
[21] (WPS - Workplace Solutions GmbH)
[22] (Moyano, Pufahl, Weber, & Mendling, 2022, p. 9)
[23] (Cabot, Low-code platforms, the new buzzword, 2016)
[24] (Mendix Technology BV A, 2024)
[25] (Mendix Technology BV B, 2024)
[26] (Cabot, Positioning of the low-code movement within the field of model-driven engineering, 2020, pp. 1-2)
[27] (Cabot & Clarisó, Low Code for Smart Software Development, 2023, p. 89)
[28] (Bucaioni, Cicchetti, & Ciccozzi, 2022:21, p. 1975)
[29] (Chen, 1976)

go on playing an important role in storing data[30], the ERM modelling technique still is the one most applied when doing database architecture design. A huge number of tools support ERM modelling and the succeeding transformation of the models into a relational design and in Data Definition Language (DDL). Class diagrams as used by UML have not gained acceptance as basis for database architecture design because object-oriented databases did not prevail over relational databases. NoSQL databases are heterogenous in their database models. For relational databases, document databases and graph databases there are clear transformation rules from an Entity Relationship Model (ERM) into a database design[31]. Specific visual modelling techniques and tools for NoSQL-databases are more database-design oriented, more proprietary and in the consequence less widely spread than the broadly accepted ERM-technique. A widespread use of NoSQL databases instead of relational databases for app-development was feared to draw back the use of visual modelling of data structures. Visual data modelling is not a must as first step in designing this kind of databases. A practitioner's first step in designing NoSQL databases can be the analysis of the data access in the concerning application domain in order to make sound design decisions[32]. After this, ER-Modelling (including enhancement of the original ER-Modelling) and UML-Modelling are used a lot on conceptual and logical modelling levels when developing NoSQL-Databases[33]. These two steps also may be used the other way round: First apply ER-Modelling or UML-class modelling and apply certain transformation rules into a graph or document store. Afterwards, the functional requirements are considered by specifying common queries and their usage frequency to change the database schema in a way that makes the database more efficient in terms of execution time[34].

There are tools coming up that support visual data modelling also for NoSQL databases using other visual techniques than ERM, extensions of ERM or UML class-models. Relational and non-relational databases now are used in combination by applications that work with different kinds of data.

The development regarding to NoSQL databases is considered being neutral on the usage of visual formal modelling. Current studies reveal that the existing modelling methods like UML and ER-models go on being practiced for NoSQL databases.

2.4 Process Mining

In the domain of Business Process Management (BPM) modelling is used for the description of actual and future (i.e. optimized, revised) Business Processes. The most widely spread modelling technique here is Business Process Model and Notation (BPMN) which has been standardized by the Object Management Group (OMG, BPMN 2.0). Also, eEPCs (extended Event-driven Process Chains) are well known for business process modelling, especially in Europe. However, in the recent years BPMN is more widely spread than eEPCs[35]. Process Mining is a field which aims to dig descriptions on current processes up from systems that support them or parts of them. The information on process flows then is collected from log files, transaction protocols etc. So, the first target of process mining is to automatically

---

[30] (solid IT consulting & software development gmbh, Vienna, Austria, n.d.)

[31] (Kaufmann & Meier, 2023, p. 42 et seqq)

[32] (Wildgruber, 2024)

[33] (Vera-Olivera, et al., 2021, p. 116:16)

[34] (Roy-Hubara, Sturm, & Shoval, 2023)

[35] (Harmon & Wolf, 2016, p. 29)

create precise descriptions of current processes, assuming, that models created by humans do not always represent the true facts. Depending on the scenario of the individual processes process mining is more or less viable:

- Process mining works well for integrated digitized processes, i.e. processes that are fully carried out in ERP-systems.
- Processes that contain a relevant part of individual human work, i.e. coordination work by telephone calls, are only partially digitized. Process models for such processes cannot fully be derived automatically from information contained in IT-systems. Human parts of the process then have to be modelled manually.
- For processes whose execution is fully digitized but spread over many systems, the process mining tools must be fed from all these systems to automatically create a comprehensive model of a process's control flow.

We conclude that process mining, so automatic creation of models of current processes, is promoted by ongoing digitization and integration of processes. Process mining tools then offer functionality to simulate and analyze the processes. The problem of process complexity, visually documented by so-called spaghetti diagrams[36], can be tackled by different approaches, i.e. clustering, abstraction, noise filtration and pattern mining[37]. By using these approaches a complex process can be differentiated in actual process mining tools into a basic process flow and process variants. For the basic flow and its variants, the frequencies can be determined. Causes can be found by drill down if those frequencies reveal unexpected facts. This way it can be found out if a certain process always runs in another way than expected if certain objects participate in it. This gives a hint on where to start to achieve more process standardization. Also, there are approaches to automatically improve processes by automated process planning. The combination of Process Mining, AI Planning and Machine Learning experimentally is investigated[38]. The more the preconditions mentioned above for process mining are given the lower is the inclination to let processes be modelled by humans. Probably, this will be the case first in large companies with a high level of process complexity and process digitization.

2.5 Generative Artificial Intelligence

Generative AI is developing as alternative for human work in creating texts and other objects (arts, music). The generated objects can be created targeted for a certain purpose depending on the prompt used for initiating the generation. Lots of sources of knowledge are brought in by the generators. As generative AI in the daily press is reported to being used for creating texts, arts, composing pieces of music it can be suspected that visual formal models can also be created by using the corresponding platforms. In the following an experiment is described, letting generative AI fulfill modelling tasks that normally have to be performed by students in practices to teaching sessions on visual formal modelling.

---

[36] (Imram, Ismail, Hamid, & Nasir, 2022, p. 101516)
[37] (Imram, Ismail, Hamid, & Nasir, 2022, p. 101525)
[38] (Fettke & Rombach, Towards Automated Process Planning and Mining, 2022, p. 98)

3 Results of an experiment: Generative AI solving modelling tasks

The generation of a model based on textual descriptions is a task that on the one hand includes the analysis and abstraction of the contents of the text, on the other hand the production of graphical representations that stick to the rules of a modelling technique. The second step, the production of the graphical representations, differs from the tasks that normally are performed by so-called image generators.

The first step, the analysis of the textual content and its abstraction into models appeared to be a task that most likely could be performed by Generative AI that focuses on text creation depending on a given task. Generators that are focusing on the creation of Blog, Website and Social Media Contents have a marketing-oriented focus but are not targeted on performing analytical tasks. Therefore, the currently available best-known generative AI tool ChatGPT of Open AI and its direct competitor Gemini of Google were selected for conducting the experiment.

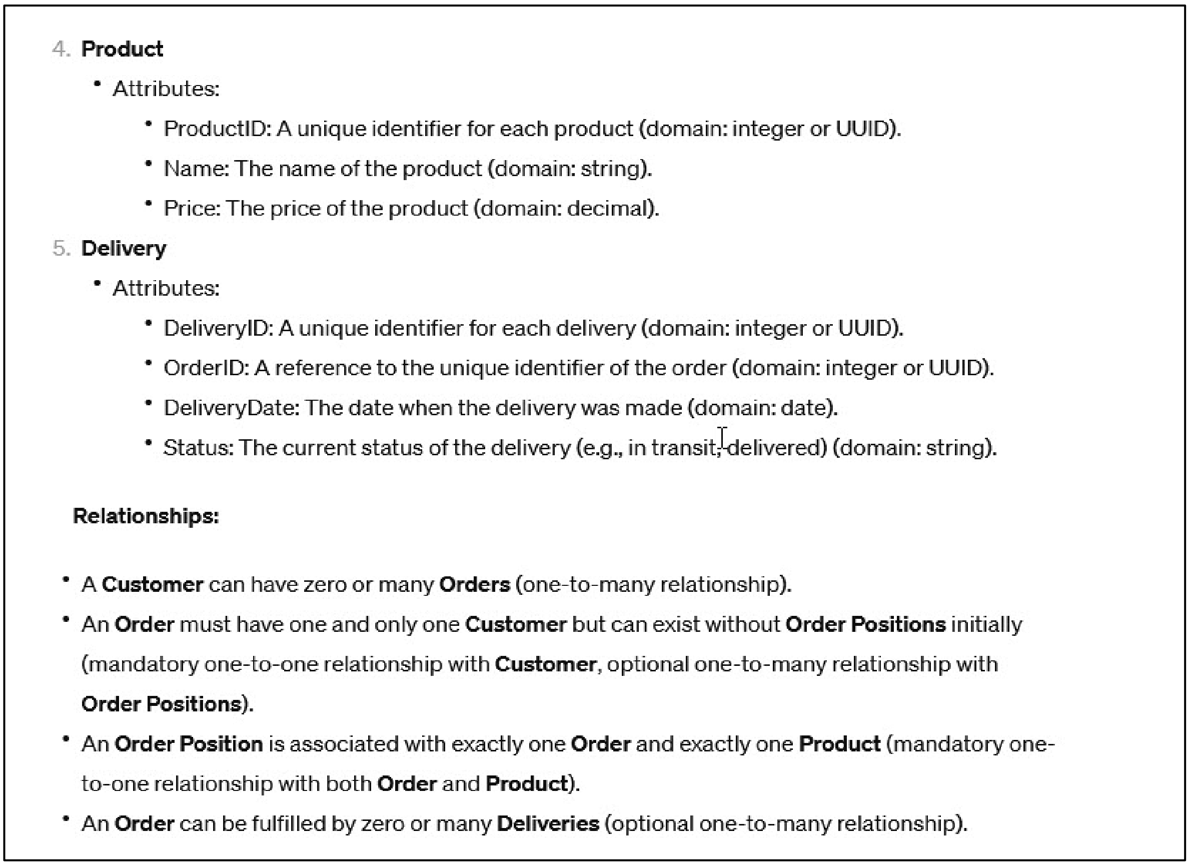
The experiment included the following steps, performed with both tools:

1. Submission of an existing definition of task for entity relationship modelling of low difficulty.
2. Submission of an existing definition of task for process modelling using informal swim lane diagrams of low difficulty.
3. Submission of an existing definition of task for entity relationship modelling of medium difficulty.
4. Submission of an existing definition of task for process modelling with BPMN of medium difficulty.

The task definitions were taken directly from lecture material used for instructing students. They were copied as prompts into the generative AI tool. Only instructions referring to the working environment or the formation of student work groups were left out. The tasks had been used for several years already and had been updated from time to time to avoid misunderstanding them. Therefore, they were formulated in a clear manner. This is especially true for the tasks of low difficulty. These tasks aim at facilitating the identification of modelling objects for students. They make it easy for students to identify the objects and relate them to each other the way the text specifies it. In the first step students achieve a solution that sticks to the text.

3.1 Entity relationship modelling task of low difficulty.

Both AI tools first achieved a textual solution on entity relationship modelling that sticked to the task of low difficulty. Both listed entities, attributes and relationships.

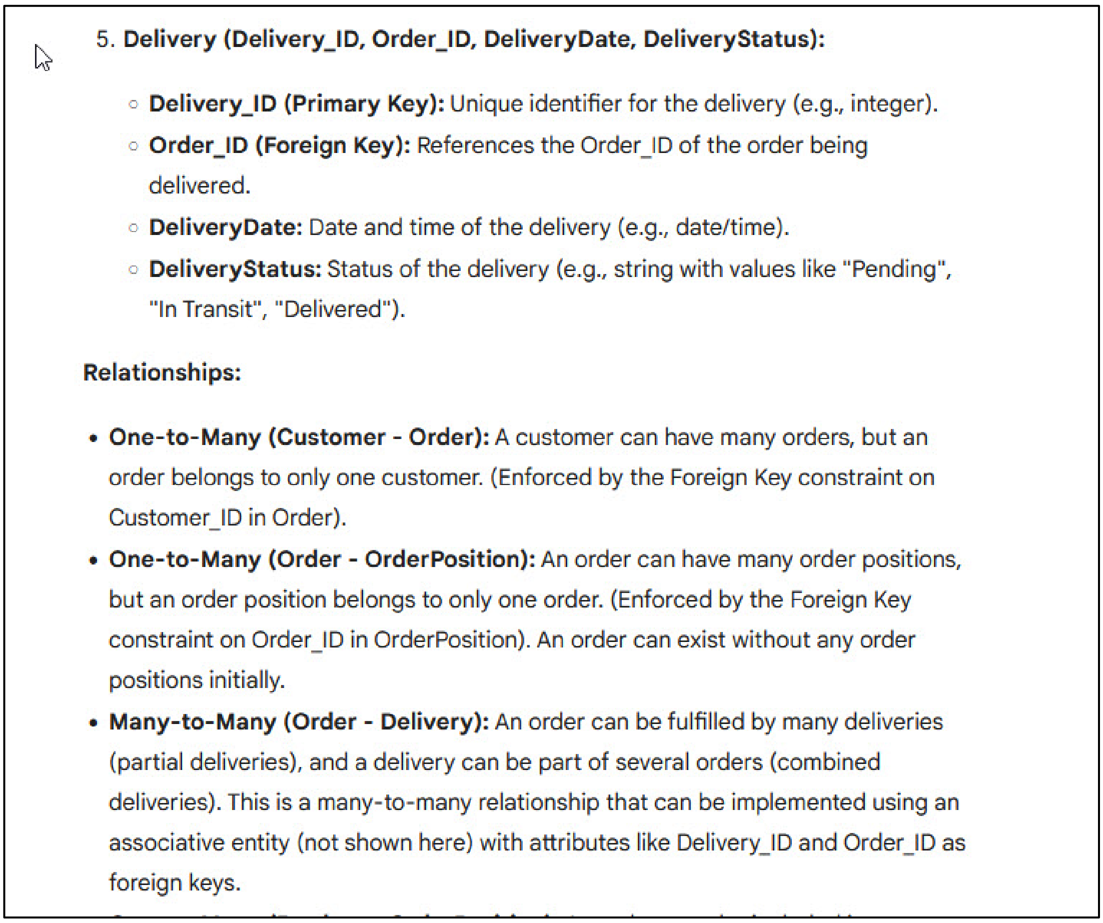


4. **Product**
   - Attributes:
     - ProductID: A unique identifier for each product (domain: integer or UUID).
     - Name: The name of the product (domain: string).
     - Price: The price of the product (domain: decimal).
5. **Delivery**
   - Attributes:
     - DeliveryID: A unique identifier for each delivery (domain: integer or UUID).
     - OrderID: A reference to the unique identifier of the order (domain: integer or UUID).
     - DeliveryDate: The date when the delivery was made (domain: date).
     - Status: The current status of the delivery (e.g., in transit, delivered) (domain: string).

Relationships:

- A **Customer** can have zero or many **Orders** (one-to-many relationship).
- An **Order** must have one and only one **Customer** but can exist without **Order Positions** initially (mandatory one-to-one relationship with **Customer**, optional one-to-many relationship with **Order Positions**).
- An **Order Position** is associated with exactly one **Order** and exactly one **Product** (mandatory one-to-one relationship with both **Order** and **Product**).
- An **Order** can be fulfilled by zero or many **Deliveries** (optional one-to-many relationship).

Figure 1 ChatGPT trying to solve a simple ER Modelling task



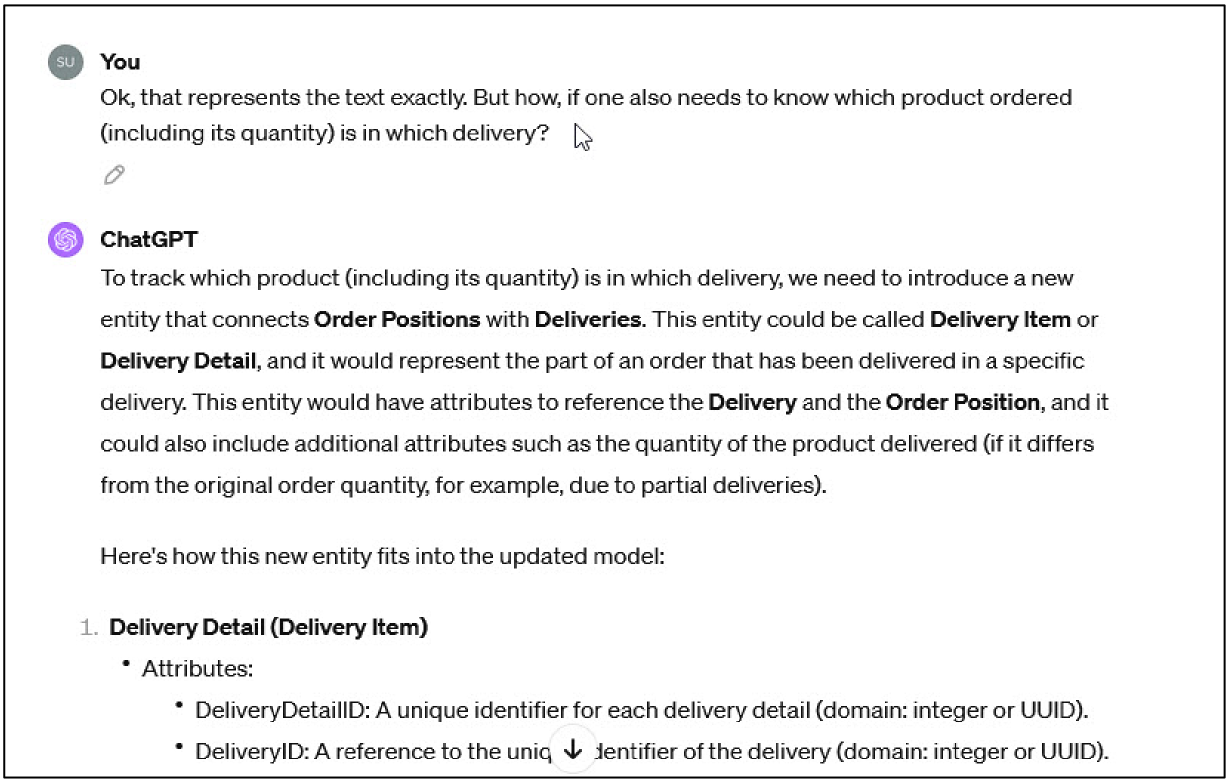
5. **Delivery (Delivery_ID, Order_ID, DeliveryDate, DeliveryStatus):**
   - **Delivery_ID (Primary Key):** Unique identifier for the delivery (e.g., integer).
   - **Order_ID (Foreign Key):** References the Order_ID of the order being delivered.
   - **DeliveryDate:** Date and time of the delivery (e.g., date/time).
   - **DeliveryStatus:** Status of the delivery (e.g., string with values like "Pending", "In Transit", "Delivered").

Relationships:

- **One-to-Many (Customer - Order):** A customer can have many orders, but an order belongs to only one customer. (Enforced by the Foreign Key constraint on Customer_ID in Order).
- **One-to-Many (Order - OrderPosition):** An order can have many order positions, but an order position belongs to only one order. (Enforced by the Foreign Key constraint on Order_ID in OrderPosition). An order can exist without any order positions initially.
- **Many-to-Many (Order - Delivery):** An order can be fulfilled by many deliveries (partial deliveries), and a delivery can be part of several orders (combined deliveries). This is a many-to-many relationship that can be implemented using an associative entity (not shown here) with attributes like Delivery_ID and Order_ID as foreign keys.

Figure 2 Gemini solving the same simple ER Modelling task

After one or two additional prompts that drew the attention to business requirements that were not obvious from the text, both tools finally came to the desired solution, with ChatGPT needing one prompt less than Gemini although prompts were kept to be the same as far as possible in the context of the delivered interim-solution. These additional business requirements represent knowledge of business practitioners. The AI tools in the first step of the experiment did not recognize from own experience the necessity to include this knowledge into the models. They caught up on this after having been reminded of certain business practices. The process of achieving the final solution by first solving only the textual statements and then being reminded of specific business requirements by asking questions was very similar to the way students achieve the desired solution.



You

Ok, that represents the text exactly. But how, if one also needs to know which product ordered (including its quantity) is in which delivery?

ChatGPT

To track which product (including its quantity) is in which delivery, we need to introduce a new entity that connects **Order Positions** with **Deliveries.** This entity could be called **Delivery Item** or **Delivery Detail**, and it would represent the part of an order that has been delivered in a specific delivery. This entity would have attributes to reference the **Delivery** and the **Order Position**, and it could also include additional attributes such as the quantity of the product delivered (if it differs from the original order quantity, for example, due to partial deliveries).

Here's how this new entity fits into the updated model:

1. **Delivery Detail (Delivery Item)**
   - Attributes:
     - DeliveryDetailID: A unique identifier for each delivery detail (domain: integer or UUID).
     - DeliveryID: A reference to the uniq dentifier of the delivery (domain: integer or UUID).

Figure 3 ChatGPT reacting on suggestions for improving the ERM

3.2 Process modelling task of low difficulty

For the process modelling task of low difficulty Gemini in the first try did not allocate all process steps to the correct organization units. It had to be reminded on what was written in the task. On process modelling both tools first produced a simple sequence of process steps and no graphics at all or simple ASCII-graphics. They did not try to include any more sophisticated control flow, i.e. parallelisms or reaction on external events. This differs from the way students normally solve this task. The usage of the ChatGPT image generator for producing a graphical model led to chaotic results:

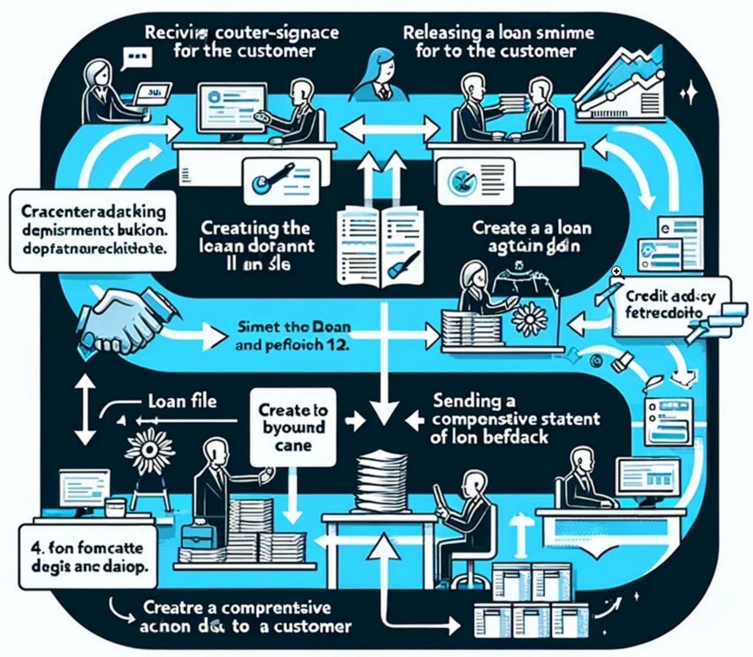

Recivig couter-signace for the customer
Releasing a loan smime for to the customer
Cracenteradatking depmsrments bukion. dopfatnareckidate.
Creating the leaan dorannt ll an sle
Create a a loan agtain gdin
Credit acd.cy fetrecdoito
Simet the Doan and pefioich 12.
Loan file
Create to byound cane
Sending a componstive statent of lon befdack
4. fon fomcatte degis anc daiop.
Creatre a comprentsive acnon dɑ to a customer

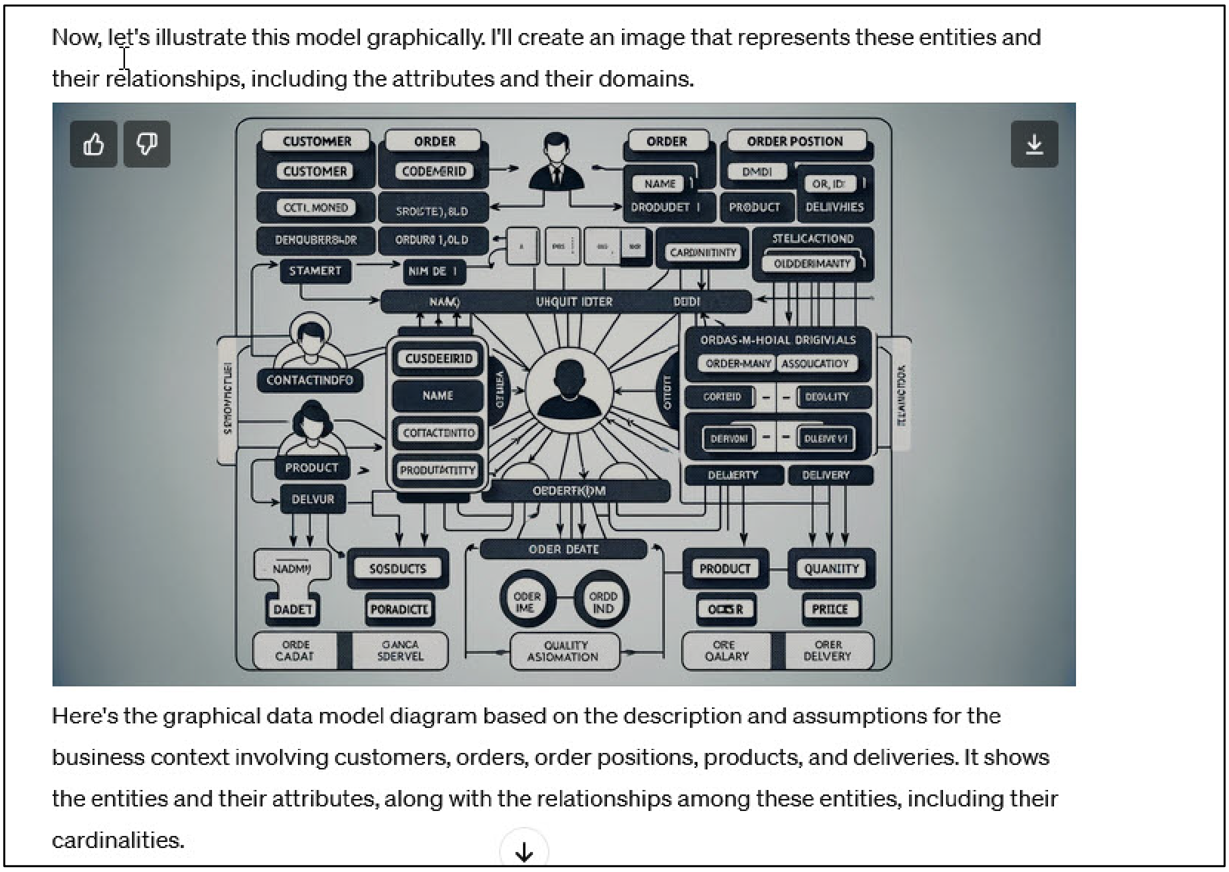
Figure 4 ChatGPT Image Generator visualizing a simple loan origination process

3.3 Entity relationship modelling and process modelling tasks of medium difficulty

When passing the entity relationship modelling task of medium difficulty to both KI-tools the need for additional instruction and discussion raised significantly. The discussion process until finally an adequate solution was achieved comprised in ChatGPT four additional prompts, in Gemini five. The points in which the tools needed guidance were similar to the ones when students need guidance.

The process modelling task of medium difficulty to be solved with BPMN could not be solved by both tools in a satisfying manner. ChatGPT kept the model very rough and left away many details. It justified this by writing that the details cannot be communicated well in this format. Gemini created the model in a more detailed manner but failed in ensuring consistency between the main process and subprocesses. Students in many cases have similar difficulties like Gemini. They tend to model very detailed but loose the overview and the connections between the different subprocesses.

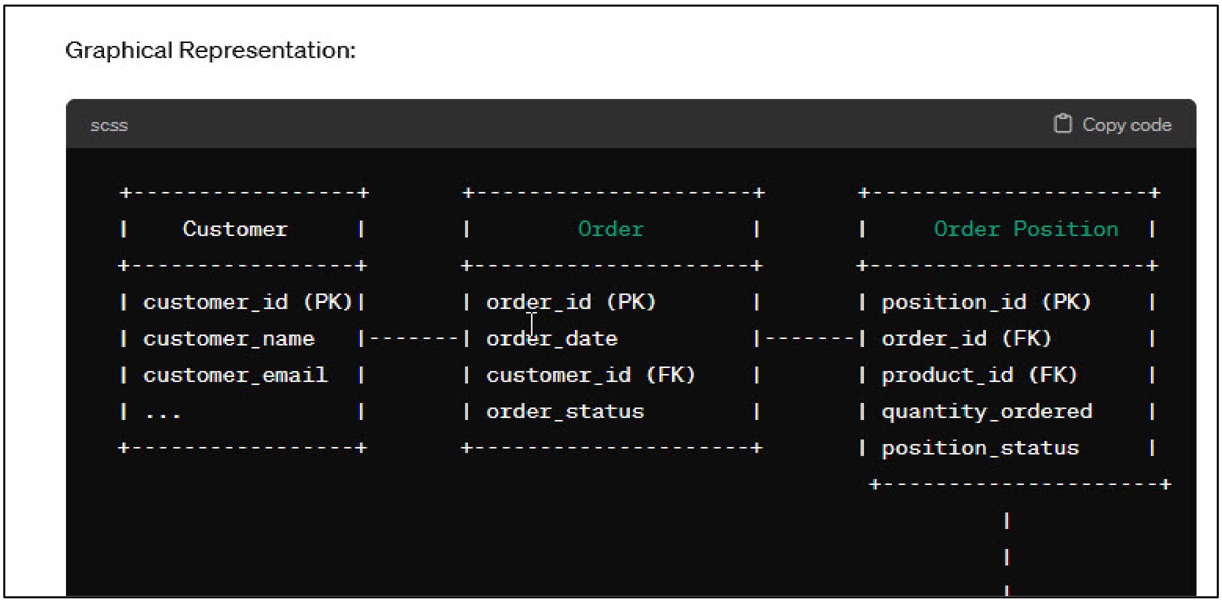
A correct graphical representation could not be generated, not for the ERDs and also not for the process models. This is true for the modelling tasks with low level of difficulty and also for the ones with medium level of difficulty. ChatGPT 4 tried to generate a diagram for the low difficulty ERD-modelling task.


Now, let's illustrate this model graphically. I'll create an image that represents these entities and their relationships, including the attributes and their domains.

Here's the graphical data model diagram based on the description and assumptions for the business context involving customers, orders, order positions, products, and deliveries. It shows the entities and their attributes, along with the relationships among these entities, including their cardinalities.

Figure 5 ChatGPT 4 trying to produce a picture for the simple ERM

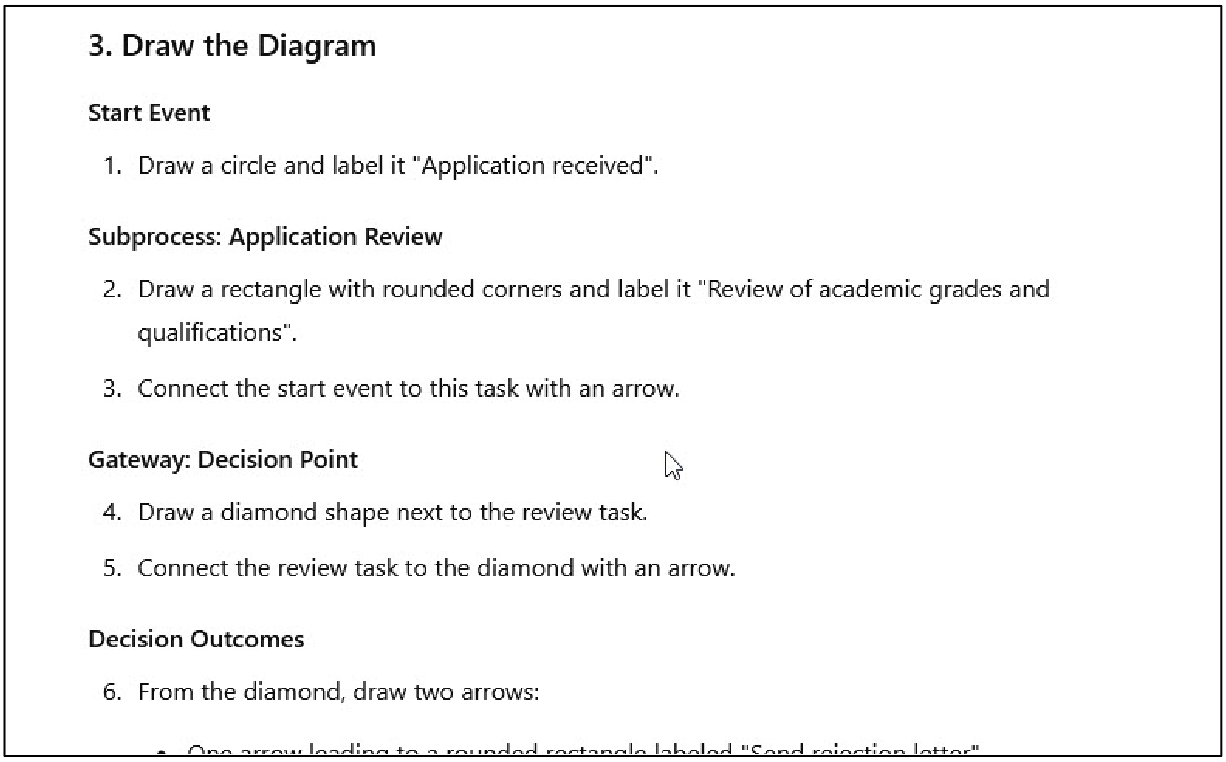
Going back to ChatGPT 3.5 for the task of producing a simple graphical ERD a partially correct but incomprehensive diagram made up of characters was shown:


Graphical Representation:

```
+-----------------+       +---------------------+       +---------------------+
|    Customer     |       |        Order        |       |   Order Position    |
+-----------------+       +---------------------+       +---------------------+
| customer_id (PK)|       | order_id (PK)       |       | position_id (PK)    |
| customer_name   |-------| order_date          |-------| order_id (FK)       |
| customer_email  |       | customer_id (FK)    |       | product_id (FK)     |
| ...             |       | order_status        |       | quantity_ordered    |
+-----------------+       +---------------------+       | position_status     |
                                                         +---------------------+
                                                                    |
                                                                    |
                                                                    |
```

Figure 6 ChatGPT 3.5 showing an ER Model as ASCII graphic

ChatGPT 4 if asked again tried the generation of an image and failed, either producing an error or a new chaotic picture that had no similarity with the correct solution. Also, the additional “Image Generator GPT” available in ChatGPT 4 could not generate a graphical model, not for data structures and not for processes. The “Data Analyst GPT”, also available in ChatGPT 4, however, gave detailed instructions to the user on how to draw a graphical model based on the textual description received from ChatGPT. This was true for the ERM-Diagram and later for the BPMN-diagram for modelling processes as well.



3. Draw the Diagram

Start Event

1. Draw a circle and label it "Application received".

Subprocess: Application Review

2. Draw a rectangle with rounded corners and label it "Review of academic grades and qualifications".
3. Connect the start event to this task with an arrow.

Gateway: Decision Point

4. Draw a diamond shape next to the review task.
5. Connect the review task to the diamond with an arrow.

Decision Outcomes

6. From the diamond, draw two arrows:
   - One arrow leading to a rounded rectangle labeled "Send rejection letter"

Figure 7 ChatGPT Data Analyst giving instructions on how to draw a BPMN diagram

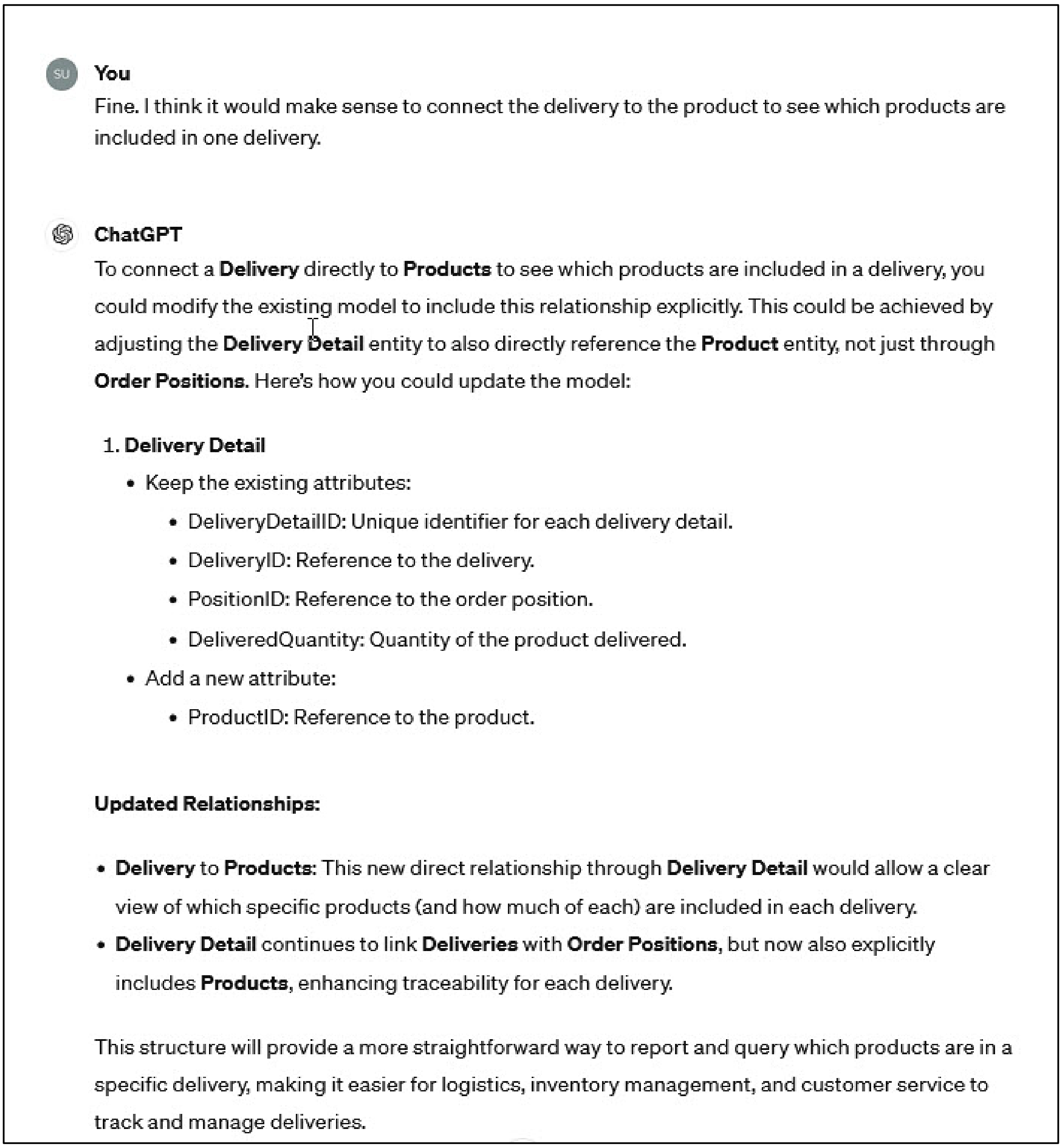
Gemini did not try to generate any graphics in that stage at all.

3.4 Submit ideas for “improving” the generated models

The experiment so far was guided by dealing with the generative AI in a similar manner like one would act when instructing students: If the generative AI has generated a basic solution that needs improvement or extension the instructor helps by asking targeted questions that lead the process of solving the task into the right direction. However, the question is open, how ChatGPT and Gemini would have reacted if the ideas for refining the basic solution would have directed the matter into a wrong direction. A student using the generative AI may issue ideas for the refinement of the first model that rather lead to a deterioration than to an improvement of it. This question was pursued in a subsequent step 5 of the experiment, using the Entity Relationship Modeling (ERM) task of low difficulty of step 1, but suggesting ideas for modification of the basic solution delivered, whose implementation in fact would diminish the validity of the model:

5. Submit a “bad idea” for improvement of the basic solution for the entity relationship modelling task of low difficulty from step 1.

To verify this, in the example used on Entity Relationship Modelling (ERM) the suggestion to link the delivery with the product-entity was made to the AI. This suggestion neglected the fact that the delivery had been made based on an order and on its order positions that were linked to the product entity already. The suggestion was meant to represent an idea a student might have. ChatGPT followed the suggestion by creating a delivery item entity and connecting it to the product entity. It did not provide a counterproposal that would use the existing relationship between the order position and the product.



You

Fine. I think it would make sense to connect the delivery to the product to see which products are included in one delivery.

ChatGPT

To connect a **Delivery** directly to **Products** to see which products are included in a delivery, you could modify the existing model to include this relationship explicitly. This could be achieved by adjusting the **Delivery Detail** entity to also directly reference the **Product** entity, not just through **Order Positions**. Here's how you could update the model:

1. **Delivery Detail**
   - Keep the existing attributes:
     - DeliveryDetailID: Unique identifier for each delivery detail.
     - DeliveryID: Reference to the delivery.
     - PositionID: Reference to the order position.
     - DeliveredQuantity: Quantity of the product delivered.
   - Add a new attribute:
     - ProductID: Reference to the product.

Updated Relationships:

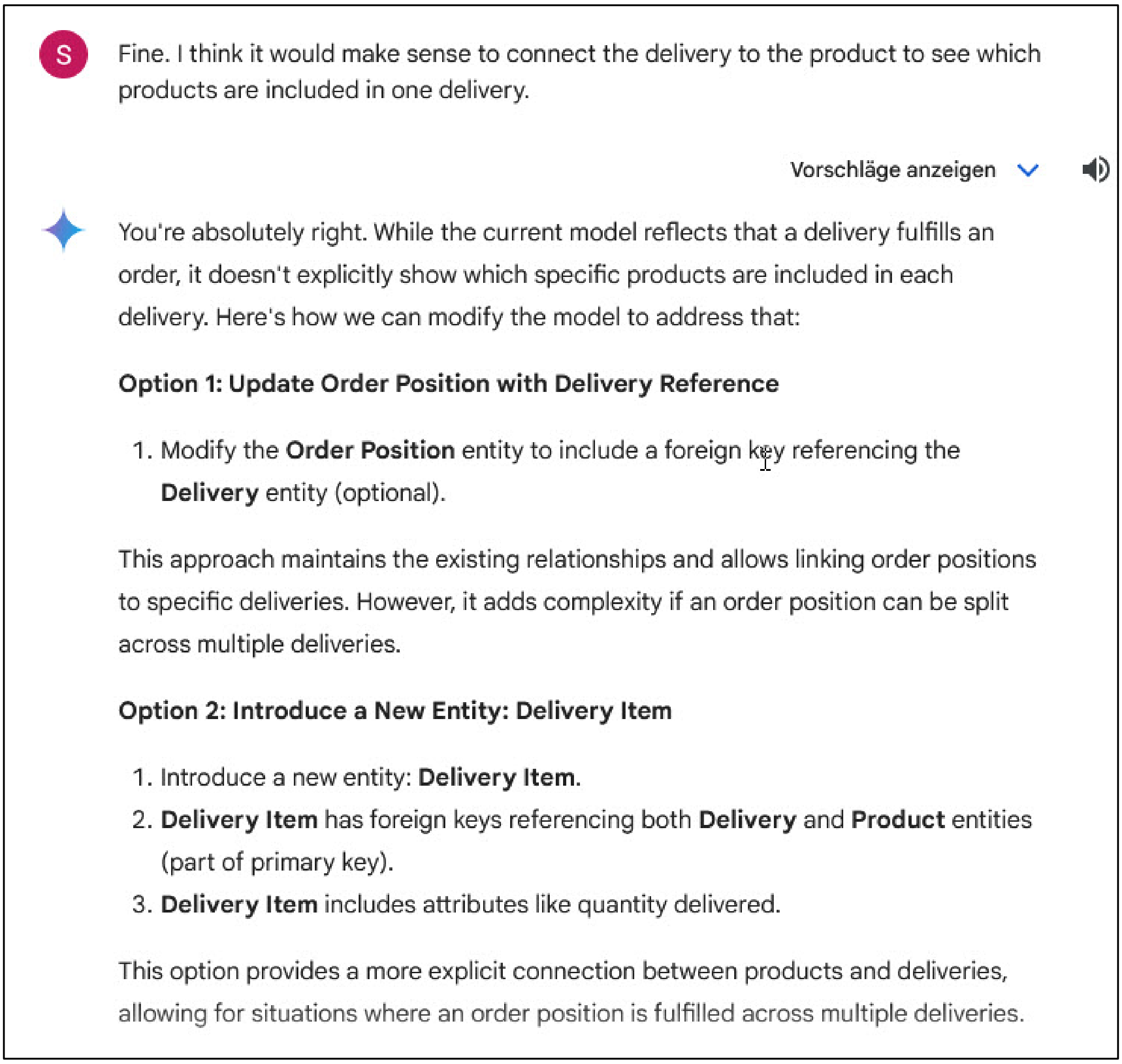
- **Delivery** to **Products**: This new direct relationship through **Delivery Detail** would allow a clear view of which specific products (and how much of each) are included in each delivery.
- **Delivery Detail** continues to link **Deliveries** with **Order Positions**, but now also explicitly includes **Products**, enhancing traceability for each delivery.

This structure will provide a more straightforward way to report and query which products are in a specific delivery, making it easier for logistics, inventory management, and customer service to track and manage deliveries.

Figure 8 ChatGPT reacting on a bad improvement suggestion for the simple ERM

Gemini also followed the instruction that was meant to mislead it. However, it delivered two options for implementing the user's suggestion. Option two in fact was the same solution suggestion as Chat GPT 4.0 had delivered. Option one included the creation of a new relationship between the entities Order Position and Delivery.

However, Gemini did not reflect afterwards, if this would cause the redundancy of the existing relationship between Order and Delivery.



S

Fine. I think it would make sense to connect the delivery to the product to see which products are included in one delivery.

You're absolutely right. While the current model reflects that a delivery fulfills an order, it doesn't explicitly show which specific products are included in each delivery. Here's how we can modify the model to address that:

Option 1: Update Order Position with Delivery Reference

1. Modify the **Order Position** entity to include a foreign key referencing the **Delivery** entity (optional).

This approach maintains the existing relationships and allows linking order positions to specific deliveries. However, it adds complexity if an order position can be split across multiple deliveries.

Option 2: Introduce a New Entity: Delivery Item

1. Introduce a new entity: **Delivery Item**.
2. **Delivery Item** has foreign keys referencing both **Delivery** and **Product** entities (part of primary key).
3. **Delivery Item** includes attributes like quantity delivered.

This option provides a more explicit connection between products and deliveries, allowing for situations where an order position is fulfilled across multiple deliveries.

Figure 9 Gemini reacting on a bad improvement suggestion for the simple ERM

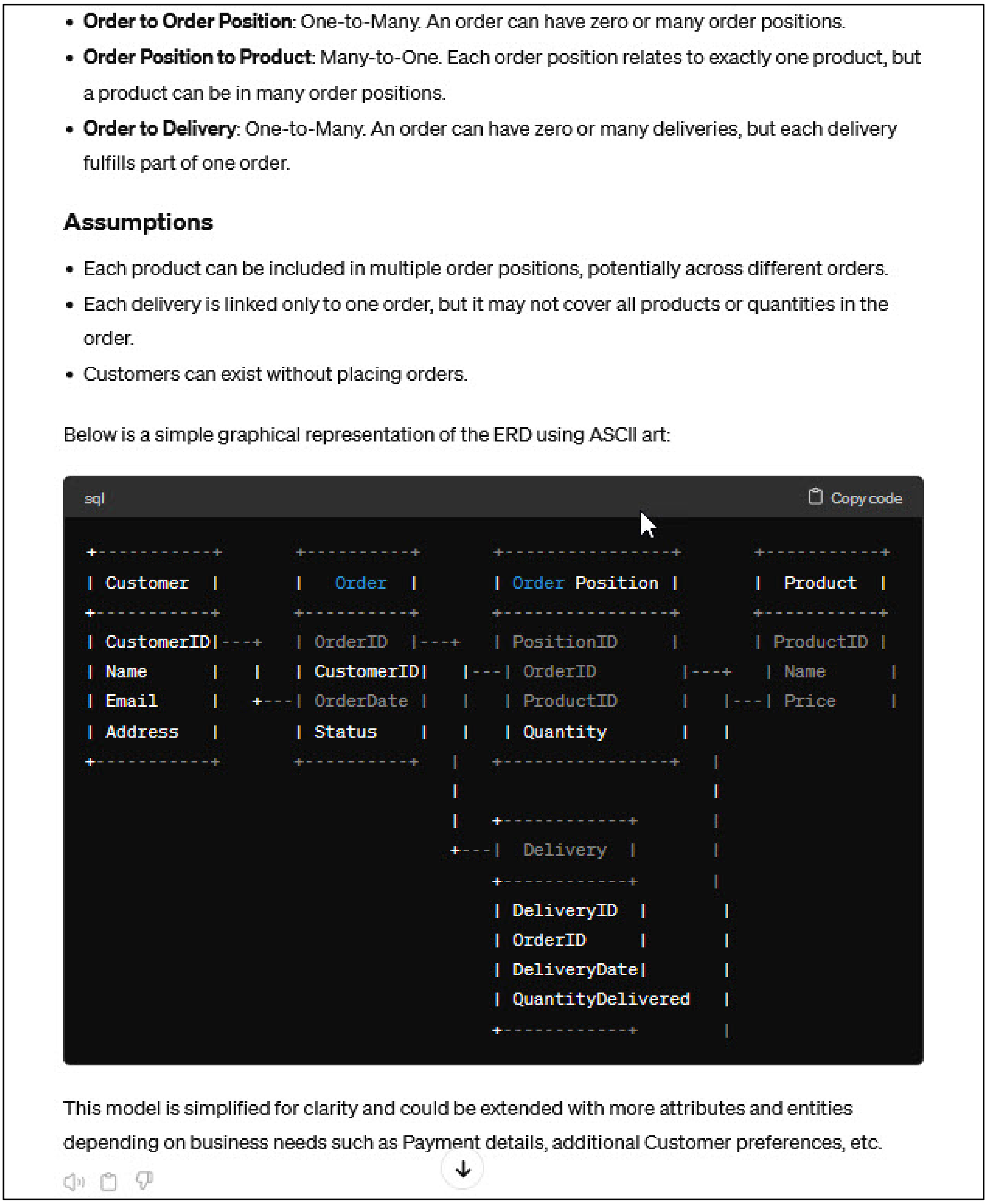
3.5 Consideration of learning and ongoing development of generative AI

After the first five steps of the experiment still the question remained how long it takes the generative AI tools to incorporate the results of the improvement dialogue from the basic solution to the desired final solution directly into a future first solution suggestion without any need of a discussion/instruction process. This question is identical to asking if the experience from the improvement process afterwards is converted into knowledge that the tools apply. This would require that the tools after having learned either directly come to the final solution or first again present a basic solution, but additionally give hints on the fact that there might be the need for extensions. Alternatively, they could after having presented the basic solution supplementally present the final solution together with its advantages compared to the basic one.

To verify this, ChatGPT 4 as well as Gemini after a waiting time of three weeks were asked to solve the same Entity Relationship Modelling task of low difficulty again. The sixth and final step of the experiment was:

6. Submission of the same definition of task for entity relationship modelling of low difficulty like in step 1 again after a waiting time

When doing so, ChatGPT 4 incorporated corrections of minor errors into the new basic solution, i.e. relationship cardinalities it had had wrong the first time. Also, it did not try to generate a real picture but stayed with a simple graphic created by ASCII characters.



- **Order to Order Position**: One-to-Many. An order can have zero or many order positions.
- **Order Position to Product**: Many-to-One. Each order position relates to exactly one product, but a product can be in many order positions.
- **Order to Delivery**: One-to-Many. An order can have zero or many deliveries, but each delivery fulfills part of one order.

Assumptions

- Each product can be included in multiple order positions, potentially across different orders.
- Each delivery is linked only to one order, but it may not cover all products or quantities in the order.
- Customers can exist without placing orders.

Below is a simple graphical representation of the ERD using ASCII art:

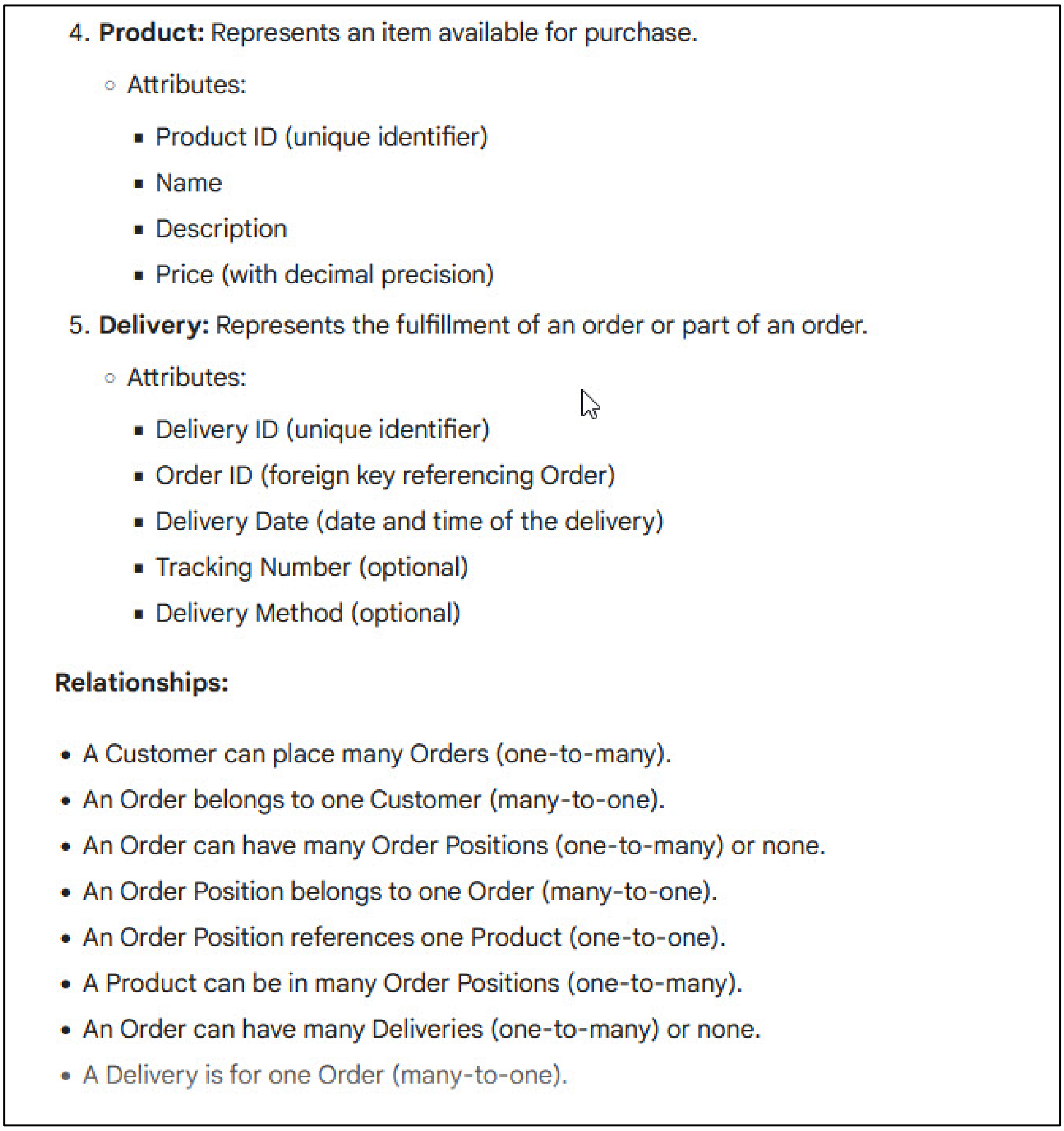
```
+-----------+       +-----------+      +-----------------+       +-----------+
| Customer  |       |   Order   |      | Order Position |       |  Product  |
+-----------+       +-----------+      +-----------------+       +-----------+
| CustomerID|---+   | OrderID   |---+  | PositionID      |       | ProductID |
| Name      |   |   | CustomerID|   |---| OrderID         |---+  | Name      |
| Email     |   +---| OrderDate |   |   | ProductID       |   |---| Price     |
| Address   |       | Status    |   |   | Quantity        |   |
+-----------+       +-----------+  |   +-----------------+  |
                                   |                        |
                                   |   +-----------+        |
                                   +---|  Delivery  |       |
                                       +-----------+        |
                                      | DeliveryID  |        |
                                      | OrderID     |        |
                                      | DeliveryDate|        |
                                      | QuantityDelivered    |
                                       +-----------+        |
```

This model is simplified for clarity and could be extended with more attributes and entities depending on business needs such as Payment details, additional Customer preferences, etc.

Figure 10 ChatGPT solving the same low difficulty ERM task again

The ASCII graphic is very spartan, but it is more correct than the graphic ChatGPT had generated in the first step. This led to the impression that ChatGPT 4 would refrain from trying to generate a real graphic again. But after having been given the task a third time a wrong graphic representation like in the first step was delivered. Not generating a real graphic therefore probably is a result of the non-deterministic nature of generative AI, not of a learning process. The sentence below the ASCII graphic giving the hint that there might be the need for extensions, first left the impression of being the result of a learning process. However, a hint like this was missing in the third try again, leaving the question if there really had been a learning process unanswered.

Gemini after three weeks presented the same solution as in the first step. When comparing the new solution to the first one only the shorter verbal formulation of the relationships strikes the eye.



4. **Product:** Represents an item available for purchase.
   - Attributes:
     - Product ID (unique identifier)
     - Name
     - Description
     - Price (with decimal precision)
5. **Delivery:** Represents the fulfillment of an order or part of an order.
   - Attributes:
     - Delivery ID (unique identifier)
     - Order ID (foreign key referencing Order)
     - Delivery Date (date and time of the delivery)
     - Tracking Number (optional)
     - Delivery Method (optional)

Relationships:

- A Customer can place many Orders (one-to-many).
- An Order belongs to one Customer (many-to-one).
- An Order can have many Order Positions (one-to-many) or none.
- An Order Position belongs to one Order (many-to-one).
- An Order Position references one Product (one-to-one).
- A Product can be in many Order Positions (one-to-many).
- An Order can have many Deliveries (one-to-many) or none.
- A Delivery is for one Order (many-to-one).

Figure 11 Gemini solving the same low difficulty ERM task again

The attempt to submit the process modelling task of low difficulty again to ChatGPT and Gemini after a waiting period of three months led to improved results in both tools. The results showed that a reflection of the first fulfillment of task had taken place and led to improved results when submitting the same task again. Also, an improvement of the graphical representations of process models had taken place. However, the overall graphical representation stayed incomplete and incorrect.

We conclude that a learning process and thereby relevant improvement of results for a repeatedly submitted task takes place but needs time. An instant or short-term learning cannot be expected from generative AI.

After a waiting time of another month ChatGPT (in Release 4o) presented the new tool “Diagrams: Show Me | charts, presentations, code”. The attempt to have the Entity Relationship Diagram of low difficulty be generated by this new tool led to an improved graphical representation, however the new tool made mistakes that ChatGPT had made in the first try about six to seven weeks before but that had been fixed in the try three weeks later. A generation of the data model with ChatGPT 4o showed again that these mistakes really did not turn up any more. In the new Diagrams-Tool assumed cardinalities of the relationships were wrong like it had been during the first try with ChatGPT 3.5 or ChatGPT 4. The new diagramming technique also offered no flexibility in the choice of notation, only Mermaid.js could be used. The new diagram looked as follows:

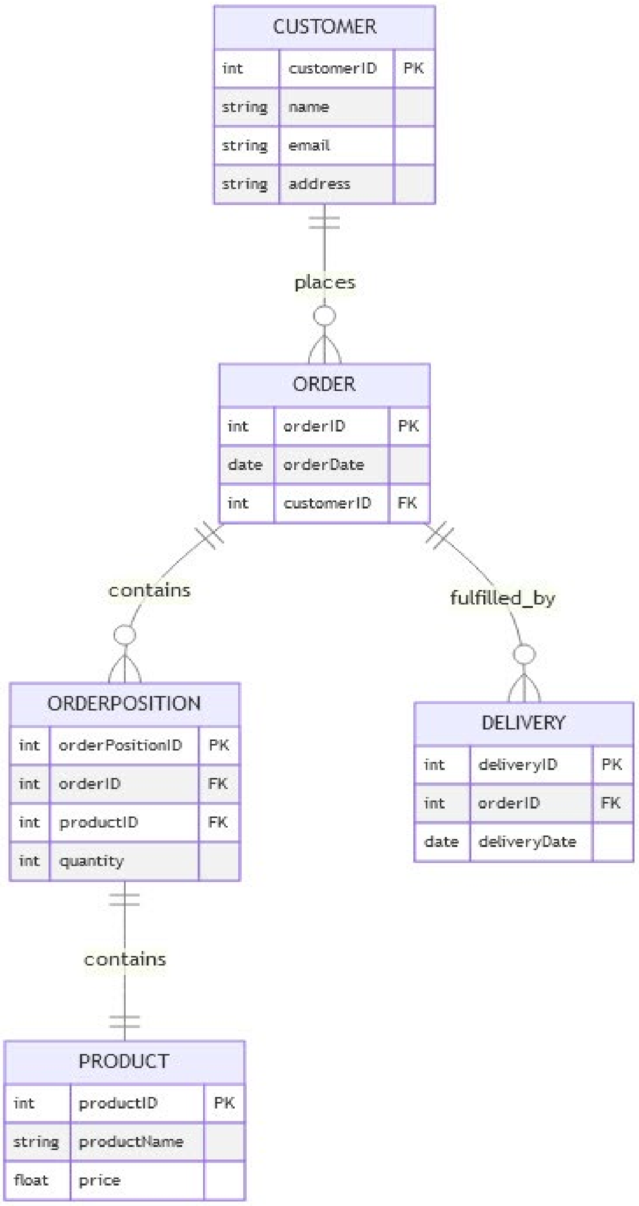


Figure 12 Using the ChatGPT 4o "Diagrams" tool for the ERM task of low difficulty

A verbal representation of the relationships like ChatGPT had generated some weeks before was not given when using the Diagrams tool. We conclude that an improvement process concerning the graphical representation of models is in progress but there is an issue concerning the integration of training that has already taken place in ChatGPT into the new tool.

4 Prediction of the future use of Modelling in Organization and Business Informatics

From the reflection of current developments and their influence on using Modelling together with the results of the experiment it can be predicted, how the usage of modelling by humans in Organization and Business Informatics presumably will develop. The following hypotheses were derived from the analysis and the experiment:

1. Modelling will go on playing an important role when developing software.
2. The trend to low code platforms will induce that the modelling task more often will be given to personnel with a non-IT-Background, i.e. so-called citizen developers.
3. The more complex a modelling task is, the more business analysts and developers will use the capabilities of generative AI for creating basic models that afterwards are candidates for validation.
4. The development to agile work processes brings along a trend of applying visual formal modelling less. However, hybrid approaches find more attention, combining the advantages of both worlds, agile and classic approaches. In consequence the influence of agility to performing less visual formal modelling will weaken as well. However, for agility is highly appreciated in the business world modelling practices need to be applied in a flexible, iterative and adaptive manner to find acceptance.
5. The technique of Process Mining will go on spreading in Business Process Management Projects in large companies with complex system environments and a high degree of digitization. So, process models of actual processes in an increasing number of cases will be derived automatically from systems and be presented in a condensed understandable form that is suitable for analysis.

The task of creating models from scratch in most cases will be taken over by systems, i.e. generative AI or process mining tools. This leads to a change of the human role in performing modelling: Human role in modelling will get more demanding for it will concentrate on validating the meeting of requirements and on quality assurance of pre-generated models.

5 Implications for teaching Modelling in Business and Informatics curricula

The findings from reviewing developments and their implications on how modelling will be practiced in future have to be taken respect of in the way modelling skills are imparted to students. It must be proceeded on the assumption that business practitioners will base their modelling activities on digitally pre-generated models. The requirement to validate pre-generated models raises the standard of modelling skills required, for checking models for their comprehensiveness and correctness for several standard business cases but also for a lot of variants of the standard is a challenging task.

Traditionally, practices for students are built up in a way to first understand the modelling technique, i.e. the meaning of the symbols and connectors. Maybe, students are given the opportunity to check these skills by using an online test for self-assessment. In the next step students have to solve a simple modelling task. If more modelling activities are part of the teaching sessions the difficulty of the modelling tasks will raise step by step.

In future the first step of making students familiar with the modelling notation used and maybe check their knowledge by an online test remains the same. However, already from the second step on instructors should actively lead students to using generative AI and being able to instruct these tools in the right way. This may mean that the tasks should be formulated less deterministic than before to leave more room for variants in the solutions the generative AI will present. The task of the students would be to check the pre-generated models against requirements that are only made available to the students and the AI once the proposed solution has been generated. This way of processing can be also applied for more complex models. Students on the one hand will learn to check the models that have been pre-generated and to critically apply a generative AI. Not all suggestions from the AI to overcome modelling shortcomings will be correct. Students will have problems to recognize mistakes in the solutions suggested by the AI. It is likely that students tend to accept a modelling solution proposed by the AI more quickly than appropriate. So, their ability to validate generated modelling contents against given requirements specifically needs to be trained. On the one hand this is even more difficult than creating models from scratch. The creation of models of low complexity and low difficulty already is a challenge for students. On the other hand, students may have less problems with this new task than with applying themselves to the task of creating new models from given textual descriptions. This part of the task in future mainly will be performed by digital model generators.

Lecturers need to have cases for models to be created by the AI-tools at hand. Additionally, specific requirements to be covered by the models created need to be available. These additional requirements should not be obvious for the AI in the beginning. They also may include basic rules for ensuring the quality of models. The requirements step by step need to be built into the AI-generated basic solutions by the students by using dialogues with the AI tools and by mentally validating the AI-proposed solutions.

This new way of working when creating models will shift the emphasis of required skills away from detailed technical modelling skills towards rather interdisciplinary skills. The task of human modelers will concentrate on ensuring the requirements fulfillment.

References

Beck, K. e. (2001). *Manifesto for Agile Software Development*. Retrieved 05 27, 2024, from https://agilemanifesto.org/iso/en/manifesto.html

Bucaioni, A., Cicchetti, A., & Ciccozzi, F. (2022:21). Modelling in low-code development: a multi-vocal systematic review. *Software and Systems Modeling*, 1959-1981. doi:10.1007/s10270-021-00964-0

Cabot, J. (2016, 9 1). *Low-code platforms, the new buzzword*. Retrieved 05 17, 2024, from https://modeling-languages.com/low-code-platforms-new-buzzword/

Cabot, J. (2020). Positioning of the low-code movement within the field of model-driven engineering. *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, (pp. Article 76, 1.3). doi:10.1145/3417990.3420210

Cabot, J., & Clarisó, R. (2023). Low Code for Smart Software Development. *IEEE Software Volume 40 Issue 1*, pp. 89-93. doi:10.1109/MS.2022.3211352

Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems, 1*(1), 9-36. doi:10.1145/320434.320440

Claudhary, T., & Trzcielinski, S. (2021). *European Research Studies Journal, XXIV*(5), 48-63.

Fettke, P. (2009). How Conceptual Modeling Is Used. *Communications of the Association for Information Systems, 25*, 571-592. doi:10.17705/1CAIS.02543

Fettke, P., & Rombach, A. (2022). Towards Automated Process Planning and Mining. *Proceedings of the Workshop on Process Management in the AI Era 2022 (PMAI 2022). 3310*, pp. 97-100. Vienna, Austria: CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-3310/paper15.pdf

GAO United States General Accounting Office - Information Management and Technology Division. (1993). *Software Tools - Defense is not ready to implement I-Case departmentwide.* Retrieved 05 24, 2024, from https://www.gao.gov/products/imtec-93-27

Harmon, P., & Wolf, C. (2016). *The State of Business Process Management.* BPTrends. Retrieved from https://bptrends.info/bptrends-surveys/

Imram, M., Ismail, M., Hamid, S., & Nasir, M. (2022). Complex Process Modeling in Process Mining: A Systematic Review. *IEEE Access, 10*, 101515-101536. doi:10.1109/ACCESS.2022.3208231

Kaufmann, M., & Meier, A. (2023). *SQL & NoSQL Databases.* Switzerland: Springer. doi:10.1007/978-3-031-27908-9

Kazi, L. (2019). The Role of Modelling in Business Software Development: Case Study of Teaching and Industrial Practice in Zrenjanin, Serbia. *Keynote Speech VIII: The Role of Modelling in Business Software Development: Case Study of Teaching and Industrial Practice in Zrenjanin, Serbia,".* Alexandria, Egypt. doi:10.1109/ICCTA48790.2019.9478791

Mendix Technology BV A. (2024). Retrieved 04 02, 2024, from What is model-driven development?: https://www.mendix.com/platform/model-driven-development/

Mendix Technology BV B. (2024). *What is low-code?* Retrieved 04 02, 2024, from https://www.mendix.com/low-code-guide/

Misra, S., Kumar, V., Kumar, U., Fantazy, K., & Akhter, M. (2011). Agile software development practices: evolution, principles and criticisms. *International Journal of Quality & Reliability Management, 29*(9), 972-980. doi:10.1108/02656711211272863

Moyano, C. G., Pufahl, L., Weber, I., & Mendling, J. (2022, December). Uses of business process modeling in agile software development projects. *Information and Software Technology, 152*. doi:10.1016/j.infsof.2022.107028

Petrescu, M., & Motogna, S. (April 2023). A Perspective from Large vs Small Companies Adoption of Agile Methodologies. *Proceedings of the 18th International Conference on Evaluation of Novel Approaches to Software Engineering.* Prague. doi:10.5220/0011716800003464

Roy-Hubara, N., Sturm, A., & Shoval, P. (2023). Designing NoSQL databases based on multiple requirement views. *Data & Knowledge Engineering, 145*. Retrieved from www.elsevier.com/locate/datak

Siegl, D. (2015, 09 05). *Why Round-Trip-Engineering doies not work*. Retrieved from LieberLieberBlog: https://blog.lieberlieber.com/2015/09/05/why-round-trip-engineering-does-not-work/

solid IT consulting & software development gmbh, Vienna, Austria. (n.d.). *DB Engines Ranking*. Retrieved 05 24, 2024, from DB Engines: https://db-engines.com/en/ranking

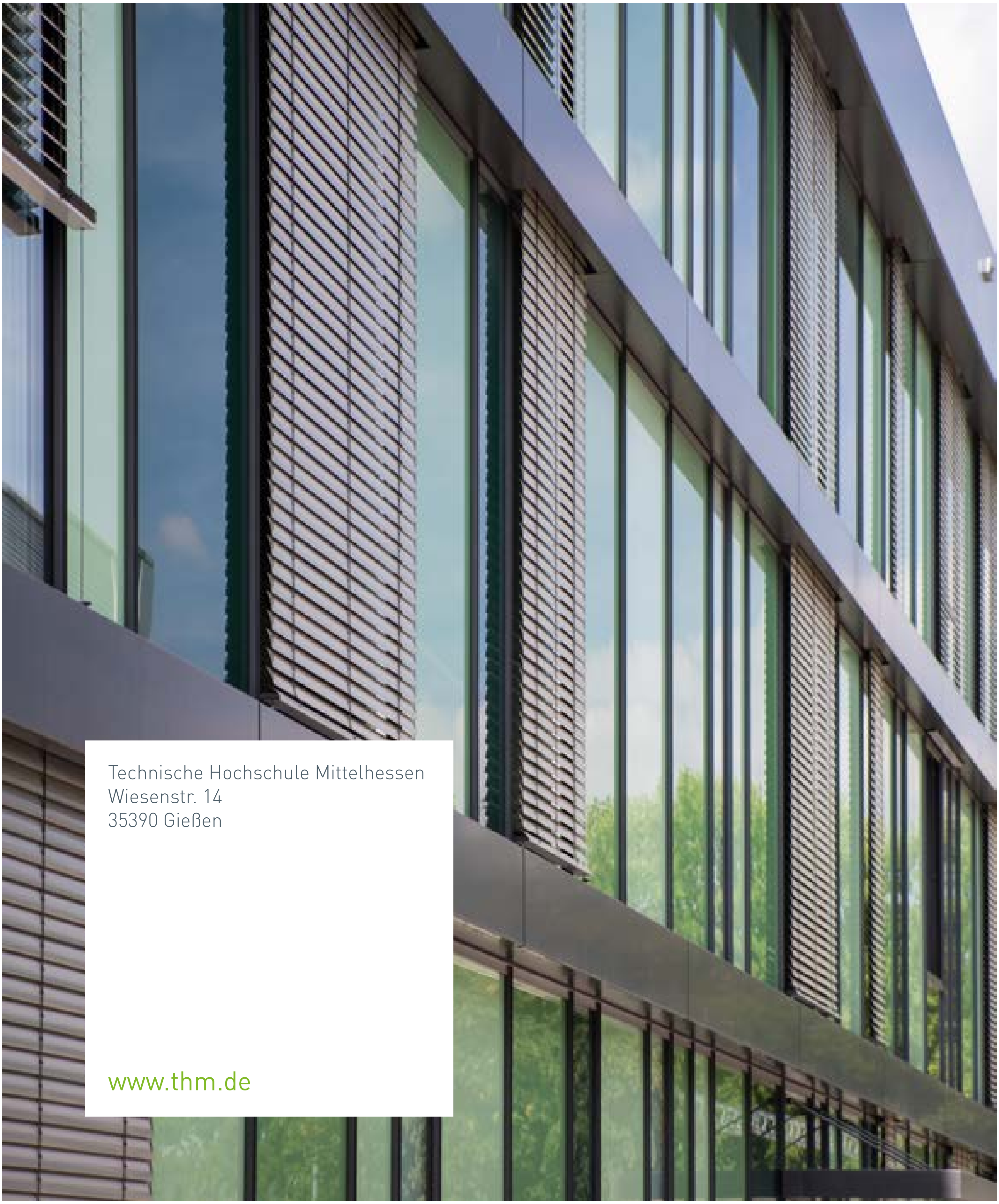
Vera-Olivera, H., Ruizhe, G., Holanda, M., Huacarpuma, R. C., Bernardi da Silva, A. P., & Mariano, A. M. (2021, July). Data Modeling and NoSQL Databases - A Systematic Mapping Review. *ACM Computing Surveys, 54*(6), 116:1-116:26. doi:10.1145/3457608

Wildgruber, M. (2024, 01 10). *Haben ist besser als brauchen - wie Datenvorbereitung NoSQL-Applikationen beschleunigt*. (sevantage software GmbH) Retrieved 05 03, 2024, from https://www.produktiver-entwickeln.de/blog/2024/01/10/nosql-daten-vorbereitung/

WPS - Workplace Solutions GmbH. (n.d.). *Collaborative Modelling - Understanding, shaping, communicating together.* WPS - Workplace Solutions GmbH. Retrieved from https://www.wps.de/en/news/collaborative-modelling

Table of figures

Figure 1 ChatGPT trying to solve a simple ER Modelling task 11
Figure 2 Gemini solving the same simple ER Modelling task 11
Figure 3 ChatGPT reacting on suggestions for improving the ERM 12
Figure 4 ChatGPT Image Generator visualizing a simple loan origination process . 13
Figure 5 ChatGPT 4 trying to produce a picture for the simple ERM 14
Figure 6 ChatGPT 3.5 showing an ER Model as ASCII graphic 14
Figure 7 ChatGPT Data Analyst giving instructions on how to draw a BPMN diagram 15
Figure 8 ChatGPT reacting on a bad improvement suggestion for the simple ERM 16
Figure 9 Gemini reacting on a bad improvement suggestion for the simple ERM ... 17
Figure 10 ChatGPT solving the same low difficulty ERM task again 18
Figure 11 Gemini solving the same low difficulty ERM task again 19
Figure 12 Using the ChatGPT 4o "Diagrams" tool for the ERM task of low difficulty 20



Technische Hochschule Mittelhessen
Wiesenstr. 14
35390 Gießen

www.thm.de