





# Generating Formal Process Models and Decisions from Examination Rules in Natural Text with ChatGPT

Andreas Speck<sup>1</sup><sup>a</sup>, Melanie Windrich<sup>1</sup><sup>b</sup>, Jan Hesse<sup>1</sup>, Melina Sentz<sup>1</sup>, David Kuhlen<sup>2</sup><sup>c</sup>,  
Thomas Stucht<sup>3</sup> and Elke Pulvermüller<sup>4</sup><sup>d</sup>

<sup>1</sup>Department of Computer Science, Christian-Albrechts-University, Kiel, Germany

<sup>2</sup>IU International University of Applied Sciences, Hamburg, Germany

<sup>3</sup>PPI AG, Hamburg, Germany

<sup>4</sup>School of Mathematics/Computer Science/Physics, University of Osnabrueck, Germany

**Keywords:** ChatGPT, Generative AI for Requirements Retrieval, Business Process Models, Decision Diagrams.

**Abstract:** Many business and administrative systems are modeled with business process models in a notation like Business Process Model and Notation (BPMN) supported by tools like Camunda (BPMN, 2025). When such systems are to be build in most cases the requirements and rules are recorded in plain text. This raises the desire using AI tools (artificial intelligence) for generating business process models from the text. The question is to which extend AI techniques may support the development of formal process models. We apply ChatGPT to analyze judicial regulations written natural text (examination regulation) and request transforming the text to process models and decision diagrams as an XML exchange file which may be displayed in the Camunda Modeler.

## 1 INTRODUCTION

Administrative rules and regulations in text form are challenging for digitalization. In many cases texts, which tend to be written in a judicial manner, are hard to transform in a formal requirements specification. A major problem with these regulations is their ambiguous wording. From a process modeling point of view the processes in these texts are not always complete. Conditions for decisions are partially implicit as well as the matter of tasks in the processes is also implicit or kept vague.


Due to this deficits a straightforward transformation of administrative regulations in written in plain text is not possible. Currently human consultants in discussion with the stakeholders are manually extracting the processes and build process models in notations like BPMN (Business Process Model and Notation (Object Management Group, 2013)) to use them in business process management systems (BPMS).


These administrative systems are mostly process-


based and BPMN is frequently used to define the specifications systems.


The question is how to support this type of requirements engineering by automated tools. Besides other concepts AI-based approaches are currently emerging (cf. Section 2).

In this paper we have a look at examination rules in the computer science department at Kiel University. Although these have been revised and improved many times there are still ambiguous regulations. This set of regulations is the base for a transformation into formal business process models (modeled in BPMN). Furthermore we take the decisions into account which are at the branches or gateways in the process. Those can explicitly be modeled in DMN (Decision Model and Notation) (Object Management Group, 2024). Although not fully covered in this paper, the general goal is to automatically generate administrative software systems from such regulations. In the paper we focus on the possibility to just transform the textually described rules into a formal process model.

<sup>a</sup> <https://orcid.org/0000-0002-7603-2493>

<sup>b</sup> <https://orcid.org/0000-0003-3285-4708>

<sup>c</sup> <https://orcid.org/0000-0001-8338-7527>

<sup>d</sup> <https://orcid.org/0009-0000-8225-7261>

## 2 RELATED WORK

In general natural texts are in many cases the starting point of the development of business process models. These business process models may be designed manually or more desirably automatically. (Sholiq et al., 2022) describes these general challenges and possible solutions beyond the usage of AI.

Natural Language Processing (NLP) is the base of many AI-based approaches for analyzing text in general and administrative regulations in particular. Other approaches for automated business process generation are rule-based (like (Neuberger et al., 2023)) or using speech pattern like shown in this survey (Schüler and Alpers, 2023).

An older approach for generating BPMN processes from texts is (Friedrich et al., 2011). The text are plain descriptions no judicial rules with less ambiguity and imprecision. Different NLP-concepts are used.

An alternative are indirect approaches by transforming text in other models which are then the base for business process models generation. An example is (Honkisz et al., 2018) where technical documentation is first transformed to spreadsheets.

Many approaches for generating business process models are using specific language models through libraries of programming languages like Python. In these cases a specific program interacts via library with the AI language model. An example of these approaches is (Möblang et al., 2024). In the conclusion of this specific paper it is encouraged to investigate alternative approaches by using Large Language Models (LLMs) like provided by ChatGPT and others.

A first response on the request to LLMs is (Kourani et al., 2024). In this paper ChatGPT is successfully used for generating BPMN processes. However, the textual descriptions are kept well structured an explicit. Nevertheless, this promising approach is a motivation to have a look at the more ambiguous and vague natural texts of judicial rules and regulations.

The generation of business process models may be an important challenge. However, there are further issues like the usage of chatbots (as a mining system for processes) that are promising to generate business value, including explanation of process mining outcomes and preparation of input data (Klietsova et al., 2024).

## 3 BPMN AND DMN NOTATION

In this paper we use BPMN (Business Process Model and Notation (BPMN, 2025)) and DMN (Decision

Model and Notation (BPMN, 2021)) for modeling business processes and decisions (Freund and Rücker, 2019). However, as we are in the requirements phase of a project we use only basic elements of both notations which we introduce here. We introduce the notation as provided by the Camunda Modeler.

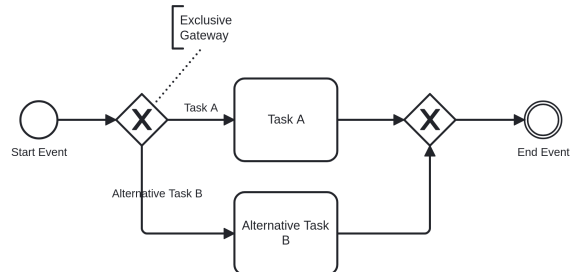


Figure 1: Example of a graphical BPMN process model.

### 3.1 BPMN Notation

The main elements of BPMN process models (as shown in Figure 1) comprise *tasks* (representing the basic functionality of the process), workflow objects (flow objects such as different gateway types or events or connection objects representing the sequence flow). In this paper we use exclusive gateways only which are very common in BPMN process models. The processes start with *start events* and terminate with *end events*.

BPMN process models as well as DMN models can be exchanged in an XML based format which includes semantic and graphical information.

### 3.2 DMN Notation

The Decision Model and Notation (Object Management Group, 2024) provides decision requirements diagrams depicting the *decisions* and *input data* of this decisions (see Figure 2). Decisions may be cascading which, however, we do not use in this paper.

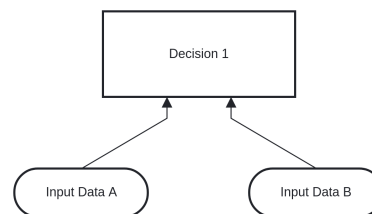


Figure 2: Example of a graphical Decision Requirements Diagram.

The decisions itself are represented in the decision tables like the example in Figure 3.

Decision 1		Hit policy: Unique				
When	Input Data A	And	Input Data B	Then	Output: Paper to be Reviewed	Annotations
	"Paper"		NumberofReceivedPapers	<=	true	

Figure 3: Example of a Decision Table.

In the upper part of the decision table the *input data* (left) and the *output* (middle). Below there are the rules and their results. In our example, however, there is only one rule: If Input Data A is "Paper" and if this (in Input Data B) has a number less or equal the NumberOfReceivedPapers received then the resulting output Paper to be Reviewed is true.

## 4 APPROACH

As already mentioned the general goal is analyzing a regulatory text with a publicly accessible AI LLM in order to identify the structures and processes in these texts and to formalize these.

To be concrete we take the examination rules in the computer science department at Kiel University and analyze this text by ChatGPT with the GPT-4o model. The ChatGPT interface we use does not allow to train the ChatGPT LLM.

We are investigating the following questions:

- Which processes are described in the text?
- Can the processes be extracted and formalized?
- Is it possible to transform the natural text representations of the processes into a BPMN model?
- Which decisions are described in the text?
- Can these decisions be extracted and represented using DMN (Decision Model Notation), and furthermore can decision tables be constructed?
- Are there decisions which are not sound described?

### 4.1 Exemplary Paragraphs for the Paper

In this paper we focus on two paragraphs of the examination rules (Christian-Albrechts-Universität zu Kiel, 2023)<sup>1</sup>:

#### § 7 Missing academic achievements and admission to modules in the 1-subject Bachelor's degree programmes

<sup>1</sup>Please note that in this text the numbering of the sections starts with 4 which is not correct. In the compulsory German version the numbering is correct.

(4) A student is missing a module in a semester if he or she has not yet passed it and the module is intended for a programme schedule that is earlier than his/her current semester (see the respective annex for the subject). Minor subject modules and compulsory elective modules are excluded from this.

(5) If a student of the 1-subject Bachelor's degree programme Computer Science or Business Information Technology is missing modules, he or she must attend these as a priority, insofar as they are offered in the current semester. In this case, he or she may attend modules worth a maximum of 35 ECTS credit points and complete examinations for them. Here, priority must be given to modules (especially missing modules) from an earlier semester according to the curriculum in the respective annex to the subject. Participation in examinations for modules that were taken and not missing is only permitted if the registrations for all missing modules in the same examination period are present or the missing modules have been passed in the meantime.

(6) The Examination Board may approve exceptions to the rules in (2) in justified exceptional cases at the student's request.

This paragraph though rather short represents a typical mixture of a process and decisions in the examination rules of Kiel University (and other universities). In the paper we first demonstrate the typical findings with this exemplary paragraph.

In second attempt we consider §29 of the examination rules:

#### § 29 Calculation of the final grade

The overall grade is calculated from the arithmetic average of the module grades weighted with ECTS credit points, excluding grades from optional subjects outside the field of computer science. Ungraded modules, such as the research project, are also not included in the final grade.

The process of this rule is rather straightforward. However, the transformation results may be an issue for discussion.

### 4.2 Prompting ChatGPT

At first we face a typical problem of using the publicly accessible ChatGPT model. The results of the prompts are not static but may differ for similar requests (which is a typical problem when working with AI). However, we are also not able to provide a consistent development when teaching the model since our impact on such an AI system is simply too limited.

In order to have at least some reproducibility we use standardized prompts.

1. Generate the process models:

Can you formalize the processes in the following text?  
 Can you build a BPMN diagram file to be imported into the Camunda Modeler?

2. Extraction of the decisions:

Can you formalize the decisions in the following text?  
 Can you build a DMN diagram file to be imported into the Camunda Modeler?

The goal is a set of process models and decision requirements diagrams including decision tables which can be imported into the Camunda Modeler. In this case we do not consider further characteristics of the tasks like forms for human interaction and code for automated execution of the tasks. These issues are important when the tasks are to be executed. However, at the moment we focus on the formalization of the requirements which is the first step in the digitalization.

As the first results (in the following Section 5 presented) are no machine readable BPMN or DMN files in XML exchange format we repeated the prompt request and emphasized the expected result should be imported into the Camunda Modeler be reminding that these first ChatGPT answers are not valid for importing into Camunda Modeler.

## 5 RESULTS

In Subsection 5.1 the results of the first prompts are presented which are textual information and no models in XML exchange format. In the further responses of ChatGPT the machine readable files in XML exchange format could not be imported into the Camunda Modeler. Therefore, in the next prompt we add an exemplary BPMN file or DMN file respectively to our prompt. This led to answers with XML files which can be imported into the Camunda Modeler. There was only one such prompt per BPMN process request and DMN request.

In the following Subsections 5.2 BPMN Process and 5.3 Decisions we describe a first case of ChatGPT prompting resulting in models which we can import in the Camunda Modeler. In a second case in Subsection 5.4 we focus on the difference between a process model generated by ChatGPT and a manually designed model.

### 5.1 First Results

In almost all of the first responses the processes and decisions in the regulations are identified. However, in these first responses there where no formal models in XML exchange format given though the key elements of the processes and decisions are presented. In a second step on a renewed request ChatGPT delivers descriptions how to build the process models and decision requirements diagrams with decision tables in the Camunda Modeler. A formal XML exchange file is not delivered.

Nevertheless, these first results are promising:

1. **Start Event:** "Missing Examinations Check"
2. **Exclusive Gateway:** "Is Student Missing Modules from Previous Semester?"  
**Yes:** "Attend Missing Modules Task"  
**No:** "Continue with Current Modules Task" (Leads directly to End Event: "Complete Current Semester Tasks")
3. **Sub-Process:** "Attend Missing Modules"  
**Task:** "Check Availability of Missing Modules"  
**Exclusive Gateway:** "Are Missing Modules Offered This Semester?"  
**Yes:**  
**Task:** "Attend Offered Missing Modules"  
**Task:** "Register for Examinations"  
**Exclusive Gateway:** "Can Register for Current Modules?"  
**Yes:** "Register for Current Modules Task"  
**No:** Leads to End Event "Registered for Missing Modules Only"  
**No:** Leads directly to End Event: "Missing Modules Not Offered"
4. **Intermediate Event:** "All Missing Modules Registered or Passed"
5. **Conditional Sequence Flow:** "If missing modules registered or passed, participation in current module exams permitted"
6. **End Event:** "Completion of Module Enrollment and Examination Registration"
7. **Exclusive Gateway:** "Justified Exception?"  
**Yes:**  
**Task:** "Request Exception Approval from Examination Board"  
**Exclusive Gateway:** "Approval Granted?"  
**Yes:** Redirect to Task "Register for Current Modules Task" or "Attend Missing Modules Task" as per circumstances.  
**No:** End Event "Standard Process Continues"

In general this first ChatGPT proposal represents the process in Section 5 of §7 of the examination rules. However, there is also a potential process in Section 6 which describes how the examination board

may handle exceptions. A short sub-process Attend Missing Modules is introduced.

Due to the missing second process and the sub-process (we consider the latter as unnecessary overhead) in the next prompt we renew our request and ask for a simplified version of the model covering both processes. As result ChatGPT presents a description without sub-process (including the sequence of the former sub-process in the now only process). Furthermore, the sequence with the approval of an exception by the examination board is added.

#### BPMN Model Elements:

**Start Event:** "Check Missing Modules"

**Task:** "Identify Missing Modules"

Identify if the student has any modules intended for earlier semesters that are not passed.

**Exclusive Gateway:** "Are Modules Missing?"

**Yes:** Proceed to "Check Module Offerings and Register"

**No:** End Event "All Modules on Track for Current Semester"

**Task:** "Check Module Offerings and Register"

Register for missing modules that are offered this semester.

**Exclusive Gateway:** "Are Missing Modules Offered This Semester?"

**Yes:** Proceed to "Register for Missing Modules and Exams"

**No:** End Event "Missing Modules Not Offered, Wait Until Next Semester"

**Task:** "Register for Missing Modules and Exams"

Ensure registration for all missing modules and examinations, prioritizing those from earlier semesters.

**Exclusive Gateway:** "Does Total ECTS Exceed 35?"

**Yes:** Redirect to End Event "Must Reduce ECTS Load"

**No:** Proceed to "Complete Registration Process"

**Parallel Gateway (AND):** "Registration Completion Check"

**Condition:** All missing modules registered, or passed.

**Task:** "Register for Current and Non-Missing Modules" Allow registration if conditions for missing modules are met, ensuring exam registration for these modules only occurs if all conditions are satisfied.

**Exclusive Gateway:** "Require Approval for Exception?"

**Yes:** Proceed to "Request Examination Board Approval"

**No:** End Event "Registration Finalized"

**Task:** "Request Examination Board Approval"

Submit a request for exception to the Examination Board.

**Exclusive Gateway:** "Approval Granted?"

**Yes:** Redirect to "Complete Registration

Process"

**No:** End Event "Exception Not Granted, Follow Standard Rules"

**End Event:** Various end states ("All Modules on Track for Current Semester", "Missing Modules Not Offered, Wait Until Next Semester", "Must Reduce ECTS Load", "Registration Finalized", "Exception Not Granted, Follow Standard Rules")

Although all prompts up to now requested a BPMN model ChatGPT failed to deliver such a model. Such model attempts are given after a pronounced prompt requesting a model to be imported into the Camunda Modeler.

## 5.2 BPMN Process

After the first prompts ChatGPT responds with an file in XML exchange format which represents (partially) a BPMN process model. However, at first information about the positioning of the graphical elements for the graphical model editor are missing:

```
<!-- Diagram positions and edges are
omitted for simplicity -->
```

Another prompt requesting the graphical part of the BPMN process model results in an XML file including the information about the graphical elements and their position. This file can be imported into the Camunda Modeler. Figure 4 depicts the graphical display of this model in the Camunda Modeler (in this case the arrangement of the elements, concrete the arrows representing the process flow, are horizontally optimized by applying the respective functionality of the Camunda Modeler).

This model is not perfectly conform to BPMN. The end event Exception Not Granted, Follow Standard Rules is not connected to any other element although this end event should be connected to the exclusive gateway Approval Granted? as the negative outgoing flow.

Besides the not connected end event Exception Not Granted, Follow Standard Rules the model represents the text of the rule. Nevertheless, looking at this model it is clear that the end events need to be connected to other (sub-) processes. This, however, points to the text itself. One may ask if the text needs improvement.

In general all requests to ChatGPT for generating a BPMN model lead to similar problems: The general intention of the process is covered well. However, specific details are lost.

## 5.3 Decisions

Besides the processes we also have a look at the decisions which also are a important when generating

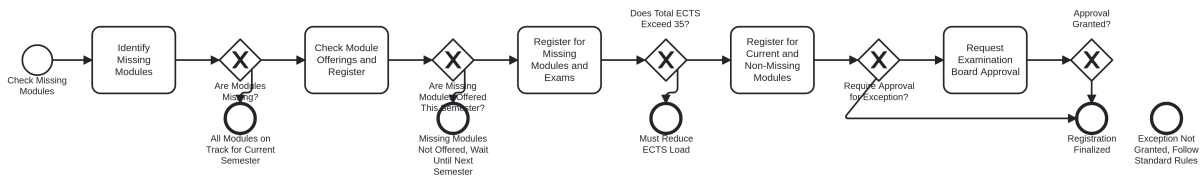


Figure 4: First BPMN diagram based on the BPMN process in XML exchange format presented by ChatGPT.

Determine if Module is Missing						Hit policy: Unique
When	And	And	And	Then	Annotations	
Status of Module (Passed or Not Passed) "Passed", "Not Passed"	Module Intended Semester integer	Student's Current Semester integer	Type of Module "Core", "Minor Subject", "Compu..."	Is Module Missing boolean		
1 "Not Passed"	< CurrentSemester	CurrentSemester	"Core"	true		

Figure 5: Decision Table of Determine Missing Module.

formal specifications from text. It is of interest if all decision points are identified. Then the DMN model and the decision table need to be generated.

ChatGPT identifies the following decisions:

- Determine Missing Module:**  
A module is missing if it is not passed and intended for an earlier semester than the current one, excluding minor and elective modules.
- Prioritize Module Attendance:**  
Students who are missing modules in the specified degree programs must prioritize attending these if they are offered in the current semester, with constraints on ECTS.
- Examination Participation Conditions:**  
Priority for examination participation must be given to missing modules, and participation in other examinations is conditional on completing missing module registrations or passing them.
- Exceptional Case Approvals:**  
The Examination Board can approve exceptions in specific cases.

Indeed, all decisions mentioned in §7 of the examination rules are identified and extracted.

In the following we focus on the first decision Determine Missing Module. This is the most complex decision and therefore of our specific interest.

Figure 6 depicts the DMN model of the decision chosen. This DMN decision requirements diagram contains the correct decision and input data (Status of Module, Student's Current Semester, Module Intended Semester and Type of Module). An output data is not part of the model which is according to the DMN standard.

The arrangement of the arrows representing the dependencies between input data and the decision is not correct as all arrowheads should point to the decision. However, this is only a problem of the graphical layout. In the DMN XML file these dependencies are

expressed right:

```
<decision id="Decision_MissingModule" name="Determine if Module is Missing">
  <informationRequirement id="InformationRequirement_1">
    <requiredInput href="#InputData_ModuleStatus" />
  </informationRequirement>
  ...
</decision>
```

The problem is then solely the layout.

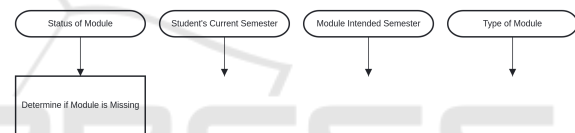


Figure 6: DMN Decision Requirements Diagram (DRD) of Determine Missing Module.

The decision table expresses the how the decisions have to be taken. Figure 5 shows the decision table for the decision Determine Missing Module

The decision table has only one rule determining that a module is missing which means that the module is not passed within the given time range or semester count respectively. All input data are covered and an output data is generated although no such output data is mentioned in the examination rules. Also the rule conditions are expressed correctly.

Remarkably, there is no rule for passing the module in time. Again, in the text there is no such rule described explicitly. This rule is only given implicit.

All other decisions are similarly covered in a correct way.

### 5.4 Case 2 BPMN Process in §29

In the second case we investigate §29 of the examination rules. The result of the generation is depicted in Figure 7.

The BPMN process of Figure 7 may be considered as culmination of various prompting attempts. In some answers only three tasks were in the model. And the task Exclude Ungraded Modules is interpreted

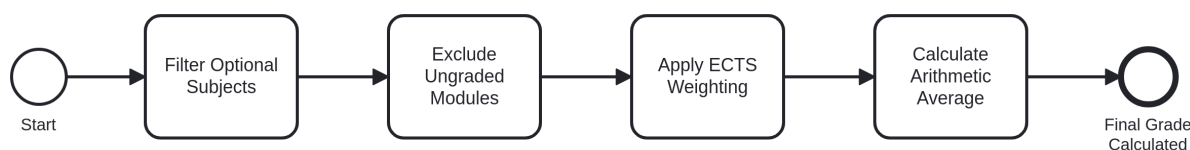


Figure 7: Graphical BPMN representation of the process, generated by ChatGPT.

as exclusive gateway which may be reasonable to a certain point.

We now compare this prompting result with a process model which has been modeled manually. The manually created process model, as depicted in Figure 8, divides the procedure into five steps. First, all modules of the respective student are compiled. This is based on a data source not further specified in §29. The paragraph describes two filtering steps for the modules, whereby ungraded modules and optional modules are excluded under certain conditions according to the examination rules. For the modules remaining after filtering, the final grade is calculated by applying the weighted arithmetic mean based on the ECTS credits of each module. The mathematical procedure for the calculation is described in detail in the process representation in Figure 8.

At first glance, the processes shown in Figure 8 and Figure 7 exhibit a high degree of similarity. For the simple case of §29, the process generated by ChatGPT appears to closely match the manually created process. Both processes depict the application of module filtering through two activities where filtering criteria are applied. The BPMN process generated by ChatGPT is less detailed. The activity for determining all modules, as shown in Figure 8, was not included in the ChatGPT-generated process. Here, the process generated by ChatGPT is close to the original text of §29 examination rules. While describing this first step may provide some guidance to a reader, the activity *Determine all Modules* can be omitted without significant loss of content. Both process versions (Figure 8 and Figure 7) also align in the step *Weight Module Grade with ECTS Credit Points*. However, the process representation in Figure 8 includes slightly more detail, to emphasize the starting point of the calculation and indicating that the step involves a sequential loop. Compared to the activity *Apply ECTS Weighting* generated by ChatGPT, these details appear minor, as the activity in ChatGPT's process diagram is clear and understandable.

A major discrepancy between the processes in Figure 8 and Figure 7 becomes evident in the final activity. In the manually created process diagram, the last step is titled *Divide by the Total Number of ECTS*. In contrast, the last step of the process generated by ChatGPT is *Calculate Arithmetic*

*Average*. Due to the reduced level of detail in the ChatGPT process model, the source used to calculate the average is not explicitly defined. The sequence of activities in the process model in Figure 7 suggests that the third step produces a list of numbers representing the product of the module grades and their respective credits, which serves as the basis for the fourth step to calculate the average. Consequently, the computational result of the process generated by ChatGPT would be  $\frac{1}{n} \sum_{i=0}^n c_i g_i$  where the average of  $n$  modules is calculated,  $c_i$  represents the credits of the module, and  $g_i$  the module grade. Mathematically, this form of calculation is incorrect as it does not correspond to the weighted arithmetic mean required by §29 of the examination regulations. Correctly, the weighted arithmetic mean in this case is calculated using the formula  $\frac{1}{\sum_{i=0}^n c_i} \sum_{i=0}^n c_i g_i$ , as clearly illustrated in Figure 8.

## 6 EVALUATION

The general problem with LLMs like ChatGPT is that their answers to prompts are arbitrary and not predictable. There is no goal to which one can converge to e.g. by teaching the model. So far no consistent methodology for evaluating the generation of process diagrams and decision requirements diagrams by LLMs can be found. Therefore we propose a set of relevant metrics which we then use to evaluate the findings described in Section 5.

### 6.1 Metrics

When defining relevant metrics, two aspects can be considered. The first aspect is the process of generating itself. For example it could be counted how many attempts/prompts are needed before a sufficient result is generated. But this depends highly of the prompt engineer's skills and furthermore is only relevant during the first tries. Later a comprehensive prompt, considering the failed attempts can be created to get sufficient result in only one step. Other aspects like the runtime of each prompt may be measured but only have a slight relevance, as the whole generation process is not used often or for large amounts of text. For



Figure 8: Graphical BPMN model of the process, manually created by Camunda Modeler.

our use case we found no relevant metric in this area and therefore do not consider it in the following.

The other aspect is the quality of the generated diagrams. This is highly relevant for our use case, as the diagrams may be used in practice and represent important regulations. Of course, each generated diagram should be reviewed before actual use. Nevertheless the generated diagrams should meet a high quality standard to be useful. We identified three relevant metrics to evaluate the quality of generated diagrams that are described in the following.

### 6.1.1 Completeness and Accuracy of Generated Diagrams

The generated diagrams should represent the given text as good as possible. That means, that firstly all actors, activities, decisions etc. mentioned in the text should be observable in the diagrams. Secondly the diagrams should not include any additional aspects. And thirdly the process model sequence and the all in all semantics of the diagrams have to comply with the given text.

While this requirements seem to be obvious, they are hard to verify, especially for the process models. Process modeling is a very subjective field of activity. On the one hand, this is because natural language texts are often ambiguous or do not contain all relevant information. On the other hand, often various methods to model a single matter exist.

### 6.1.2 Compliance with Standards

Besides the semantics the syntax of the proposed diagrams need to meet the corresponding standard (i.e. BPMN or DMN). This means the graphical diagram should be valid (e.g. form and arrangement of the model elements) for clear understanding. And moreover the underlying XML file must match the corresponding XML Schema definition so that it can be imported into widespread (standard compliant) modeling tools. The standard conformity is relatively easy to verify by importing the diagram into one of those tools. Of course, the implementation of an own syntax check would be another option beneficial for large sets of generated diagrams.

### 6.1.3 Execution Feasibility

Lastly, the process models in combination with the modeled decisions should be executable in a Business Process Management System with as little efforts as possible. This criterion is not as relevant as the others because not for every use case the automatized execution is necessary. An important part of the execution feasibility is the logical soundness which already should partially be ensured by considering the consistency with the BPMN Standard. But some problems may still occur like deadlocks or infinite loops if gateway conditions are defined inconveniently. The difficulty in really verifying the execution feasibility is, that dependent of the process complexity many different cases have to be tested.

## 6.2 Evaluation of the Results

In this paper we present two process models. Furthermore five other paragraphs of the examination rules have been tested:

- §4 Examinations and examination prerequisites
- §6 Mobility windows
- §11 Master’s thesis and final presentation in the 1-subject Master’s degree programmes in Computer Science and Business Information Technology
- §26 Admission to the Master’s degree programme
- §42 Transitional provisions

Paragraphs without any process descriptions have not been considered. The examined paragraphs in most cases contain no proper process descriptions. Therefore, as goal we try to evaluate how close the natural text is transformed including possible errors in the text.

### 6.2.1 Completeness and Accuracy of the Results

Besides in the first prompts all given natural texts are transformed into some sort of process models whereas the conformity to the standard is sometimes not met (cf. Subsection 6.2.2). These prompts resulted in XML exchange files which can imported in the Camunda Modeler.

An open issue is the identification of the tasks in the processes. In case there are sections in the paragraphs ChatGPT intends to determine these sections



as separate tasks. In most cases of the chosen paragraphs this may be correct.

The sequence of these tasks proves to be a problem. First in the natural text the sequences are not always explicit and easy to identify. In other cases there is no sequence of the tasks and the tasks may be considered as logical parallel. For example in §4 *Examinations and examination prerequisites* there is no (complete) temporal order of the tasks (e.g. no temporal order between examinations, seminars and practical exercises of the project modules is expressed in the rule). If in prompts it is explicitly stated that there is no temporal order then ChatGPT tends to create parallel paths originating from an exclusive gateway. Modeling different processes proved to be impossible in the trials. Obviously the LLM would need some training for handling this request. In the second case we show that the process model offered by ChatGPT differs from the manual modeling. Moreover, the models offered by ChatGPT differ when the same prompt is repeated. However, when humans model such processes the outcomes may also vary.

The decisions are far less problematic than the process model. In all cases the decisions are identified from the natural text. However, consequences or rules only based of implicit assumptions in the natural text are generally missing.

Since the models reflect the text (at least to some degree in the approach), there is the question if the process and decision models may not be used to assess the quality of the written rules. Missing parts or paths in graphical process representation indicate open issues in the text. Problems with identifying logical parallel processes indicate that the concurrency is not mentioned explicitly.

In the decision models rules may be missing. The leads to the assumption that these rules are not given in the text.

In this way the approach may be used to improve the examination rules text.

### 6.2.2 Meeting the Notation Standards

The conformity with the modeling standard is met in most cases. One consistent problem, however, appears in many first prompts to a specific paragraph: The concluding exclusive gateways ending the different paths in a process model have not been considered. When the missing concluding gateways are mentioned in the prompt, these gateways appear at the right place in the process model.

In the semantic part of the XML exchange file description of the processes the connection object are correct. However in the graphical information the coordinates of these arrows are not always the right

ones. Therefore, in the graphical models the connection objects look misplaced.

The decision diagrams as well as the decision tables are all compliant with the standard. However, in the same way as the connection objects of the BPMN processes the connection between the objects in the graphical representation are misplaced.

### 6.2.3 Not Considered in the Paper: Execution Feasibility

In this paper we solely focus on the modeling aspect of the processes and decisions. The reason is that much further information would be required for generating executable models. This information is not provided in the document we use as base. A potential solution might be combining the examination rules document with other sources such as existing software user interfaces which may serve as pattern. In many cases further details of the execution of the tasks would be required. It must be stated explicitly if this task is to be performed by a human and therefore a human interface is required or the task should run automatically with an explicit information about the business logic of this task.

## 7 CONCLUSION

In the paper we use the LLM ChatGPT for analyzing and transforming judicial rules in form of natural written text into formal process models (BPMN) and decisions (DMN). The ChatGPT access we use does not allow to train the model.

We send a prompt including the textual rule and the request to generate a formal XML exchange file which serves as input to the Camunda Modeler.

The result of our trials is (unsurprisingly) not perfect. The very first prompts resulted in a textual description of the rule. Only after two or three repeated prompts requesting the models in XML exchange format such models are provided. The generated process models have issues with parallel processes or concluding gateway elements. In some cases the generated processes are not conform to the standard. In general the process models are more critical than the decision models. The latter are considered as transformed in a correct way.

In contrast to the process models the decision diagrams are quite correct. A reason for this may be that these decisions are much better described in the textual rules. Due to our experience being involved in the development of such rules, the decisions descriptions in the textual rules are comparatively concise,

which enables ChatGPT detecting generating correct decision diagrams.

Besides the pure transformation in formal models another benefit appears: The formal representations may support the quality assurance of the examination rules in natural text. Ambiguous, vague or missing parts in the text are hard to detect by pure reading. These problems are made visible by the graphical models.

A next step may be to take the experiences and use LLMs which may be trained. Such LLMs may learn the specific issues of the examination rules as well as the standards. It is to be expected that the results would be far better.

## REFERENCES

- BPMN (2021). Camunda, DMN Tutorial, <https://camunda.com/dmn/>.
- BPMN (2025). BPMN Camunda, Documentation, <https://camunda.com/bpmn/>.
- Christian-Albrechts-Universität zu Kiel (2023). Examination Regulations (Rules) of the Faculty of Engineering at Kiel University for students of the Bachelor's and Master's degree programmes in Computer Science and Business Information Technology, as well as parts of degree programmes in Computer Science within the scope of the doublesubject Bachelor's and Master's degrees leading to a Bachelor of Arts (B.A.), Bachelor of Science (B.Sc.), Master of Arts (M.A.), Master of Science (M.Sc.) and Master of Education (M.Ed.) - 2021 (Computer Science and Business Information Technology Degree-Specific Examination Regulations - 2021) of 15 July 2021. <https://www.studservice.uni-kiel.de/sta/fachpruefungsordnung-informatik-wirtschaftsinformatik-bachelor-master-1-fach-2-faecher-englisch.pdf>. Accessed 2025-01-22.
- Freund, J. and Rücker, B. (2019). *Real-Life BPMN: Using BPMN and DMN to Analyze, Improve, and Automate Processes in Your Company*. Camunda, Berlin, 4th edition.
- Friedrich, F., Mendling, J., and Puhmann, F. (2011). Process Model Generation from Natural Language Text. In Mouratidis, H. and Rolland, C., editors, *Advanced Information Systems Engineering*, Lecture Notes in Computer Science, pages 482–496, Berlin, Heidelberg. Springer.
- Honkisz, K., Kluza, K., and Wiśniewski, P. (2018). A Concept for Generating Business Process Models from Natural Language Description. In *Knowledge Science, Engineering and Management: 11th International Conference, KSEM 2018, Changchun, China, August 17–19, 2018, Proceedings, Part I*, page 91–103, Berlin, Heidelberg. Springer-Verlag.
- Klievtsova, N., Benzin, J.-V., Kampik, T., Mangler, J., and Rinderle-Ma, S. (2024). Conversational process modeling: Can generative ai empower domain experts in creating and redesigning process models?
- Kourani, H., Berti, A., Schuster, D., and van der Aalst, W. M. P. (2024). Process Modeling with Large Language Models. In van der Aa, H., Bork, D., Schmidt, R., and Sturm, A., editors, *Enterprise, Business-Process and Information Systems Modeling*, pages 229–244, Cham. Springer Nature Switzerland.
- Mößlang, M., Bernsteiner, R., Ploder, C., and Schlögl, S. (2024). Automatic Generation of a Business Process Model Diagram Based on Natural Language Processing. In *Knowledge Management in Organisations. KMO 2024. Communications in Computer and Information Science*, volume 2152, pages 237–247. Springer.
- Neuberger, J., Ackermann, L., and Jablonski, S. (2023). Beyond rule-based named entity recognition and relation extraction for process model generation from natural language text. In *Cooperative Information Systems: 29th International Conference, CoopIS 2023, Groningen, The Netherlands, October 30–November 3, 2023, Proceedings*, page 179–197, Berlin, Heidelberg. Springer-Verlag.
- Object Management Group (2013). Business Process Model and Notation (BPMN), Version 2.0.2.
- Object Management Group (2024). Decision Model and Notation.
- Schüler, S. and Alpers, S. (2023). State of the Art: Automatic Generation of Business Process Models. In *Business Process Management Workshops. BPM 2023*, volume 492, pages 161–173. LNCS, Springer.
- Sholih, S., Sarno, R., and Astuti, E. S. (2022). Generating BPMN diagram from textual requirements. *Journal of King Saud University - Computer and Information Sciences*, 34(10, Part B):10079–10093.