From Textual Requirements to BPMN

NIVON Quentin[1], SALAÜN Gwen[1]

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, Inria, LIG
F-38000 Grenoble France

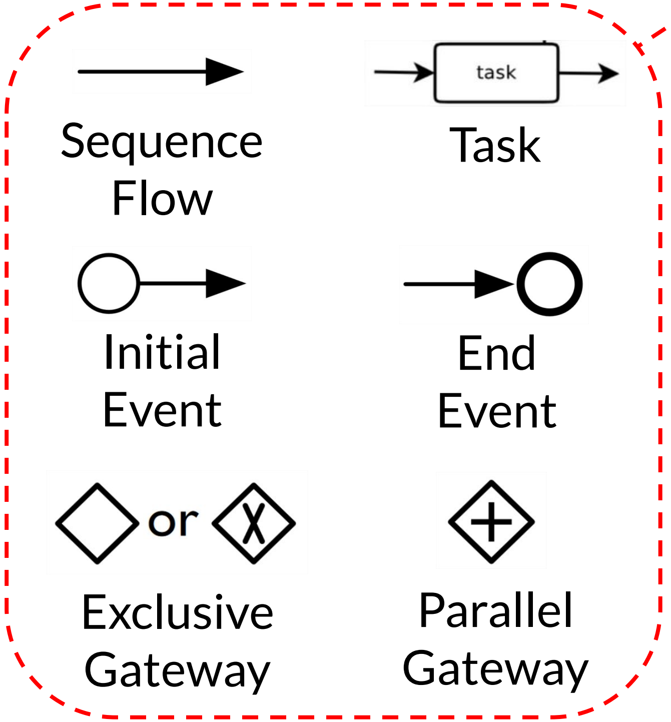
What is BPMN?

BPMN™

- ➢ A **workflow-based notation** created in 2004 by the Business Process Management Initiative (BPMI) and the Object Management Group (OMG).
- ➢ It aims at **representing business processes** in a way that is **understandable for both experienced and novice users.**
- ➢ An **ISO/IEC standard** since version 2.0 in 2013.

BPMN Syntax

Supported



Sequence Flow

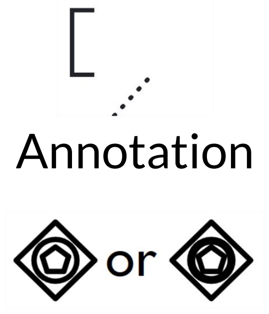
Task

Initial Event

End Event

Exclusive Gateway

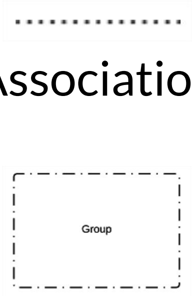
Parallel Gateway



Annotation

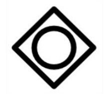
Event-based Gateway

Association

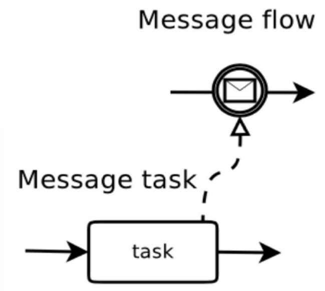


Group

etc.



Inclusive Gateway



Context

- Companies are making use of the BPMN notation to **represent their business processes.**

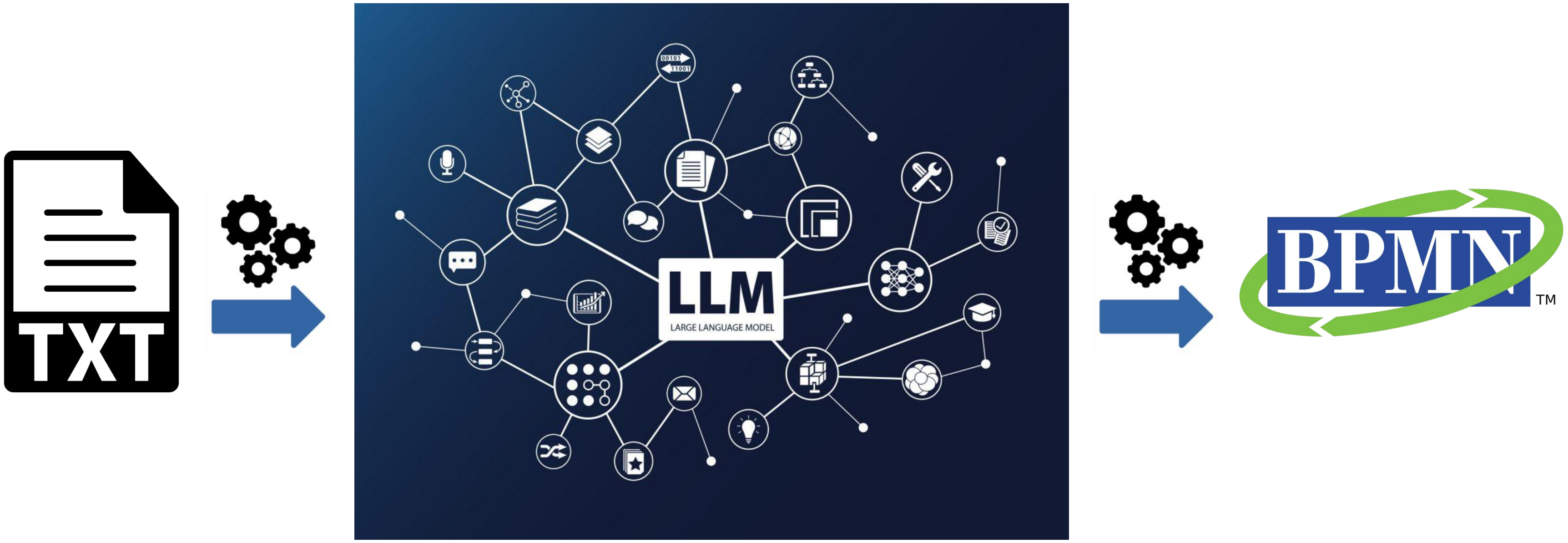
- They **hire experts** to analyse and design the **most adequate BPMN process** according to their needs.

- These processes are often **syntactically/semantically incorrect**.

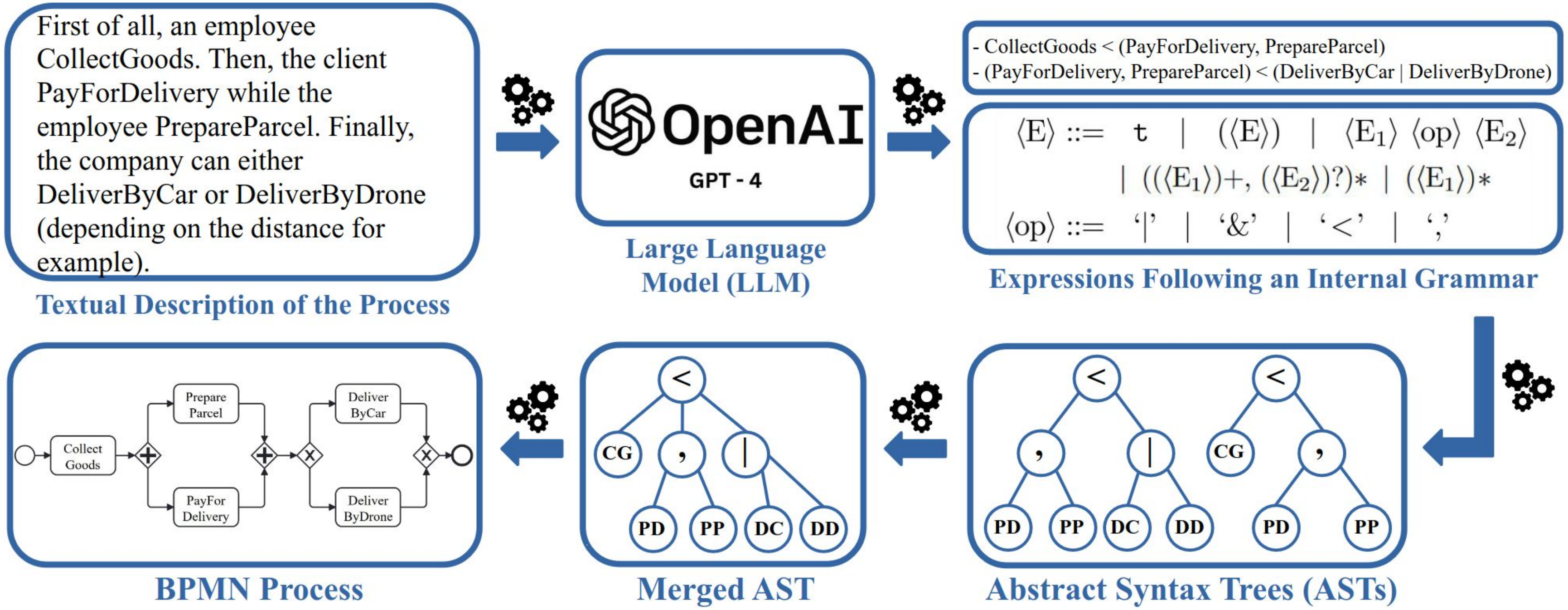
Research Questions

- ➢ What if you do not know **how to write BPMN**?
- ➢ What if you do not want to **spend time designing** your BPMN process graphically?
- ➢ How can you be sure that your BPMN process is **syntactically/semantically correct**?

General Solution



Overview of our Solution



First of all, an employee CollectGoods. Then, the client PayForDelivery while the employee PrepareParcel. Finally, the company can either DeliverByCar or DeliverByDrone (depending on the distance for example).

- CollectGoods < (PayForDelivery, PrepareParcel)
- (PayForDelivery, PrepareParcel) < (DeliverByCar | DeliverByDrone)

$$
\begin{aligned}
\langle E \rangle &::= \ \mathtt{t} \ | \ (\langle E \rangle) \ | \ \langle E_1 \rangle \ \langle op \rangle \ \langle E_2 \rangle \\
&\quad | \ (((\langle E_1 \rangle)+, (\langle E_2 \rangle)?)* \ | \ (\langle E_1 \rangle)* \\
\langle op \rangle &::= \ `|' \ | \ `\&' \ | \ `<' \ | \ `,'
\end{aligned}
$$

Textual Description of the Process
OpenAI
GPT - 4
Large Language Model (LLM)
Expressions Following an Internal Grammar
<
,
|
CG
PD
PP
DC
DD
Abstract Syntax Trees (ASTs)
Merged AST
Collect Goods
Prepare Parcel
PayFor Delivery
Deliver ByCar
Deliver ByDrone
BPMN Process

Detailed Solution – Step 1

The user first has to write a **textual description** of the process-to-be.

First, the system ProcessApplication.
Then, if the client has an account, the system RetrieveCustomerProfile and AnalyseCustomerProfile.
Otherwise, the banker CreateProfile.
Once done, the system IdentifyAccountType, and the banker PrepareAccountOpening.
After that, the user ReceiveSupportDocuments, the banker UpdateInfoRecords, and the system also performs some BackgroundVerification at the same time than ReviewApplication.
Finally, the bank either NotifyRejection, or GenerateAccountNumber, SendStarterKit, and ActivateAccount.

Detailed Solution – Step 2

The textual description is then **given to a (fine-tuned) LLM** (GPT 3.5 atm).

First, the system ProcessApplication.
Then, if the client has an account, the system RetrieveCustomerProfile and AnalyseCustomerProfile.
Otherwise, the banker CreateProfile.
Once done, the system IdentifyAccountType, and the banker PrepareAccountOpening.
After that, the user ReceiveSupportDocuments, the banker UpdateInfoRecords, and the system also performs some BackgroundVerification at the same time than ReviewApplication.
Finally, the bank either NotifyRejection, or GenerateAccountNumber, SendStarterKit, and ActivateAccount.

OpenAI
GPT - 4

The LLM processes the description and returns a **set of expressions** following an **internal grammar**.

$$\langle \mathrm{E} \rangle ::= \ \mathtt{t} \ \mid \ (\langle \mathrm{E} \rangle) \ \mid \ \langle \mathrm{E}_1 \rangle \ \langle \mathrm{op} \rangle \ \langle \mathrm{E}_2 \rangle \ \mid \ ((\langle \mathrm{E}_1 \rangle)+, (\langle \mathrm{E}_2 \rangle)?)* \ \mid \ (\langle \mathrm{E}_1 \rangle)*$$

$$\langle \mathrm{op} \rangle ::= \ `|' \ \mid \ `\&' \ \mid \ `<' \ \mid \ `,'$$

Detailed Solution – Step 3

Given our description, the LLM returns **five expressions**:

ProcessApplication

ProcessApplication < ((RetrieveCustomerProfile, AnalyseCustomerProfile) | CreateProfile)

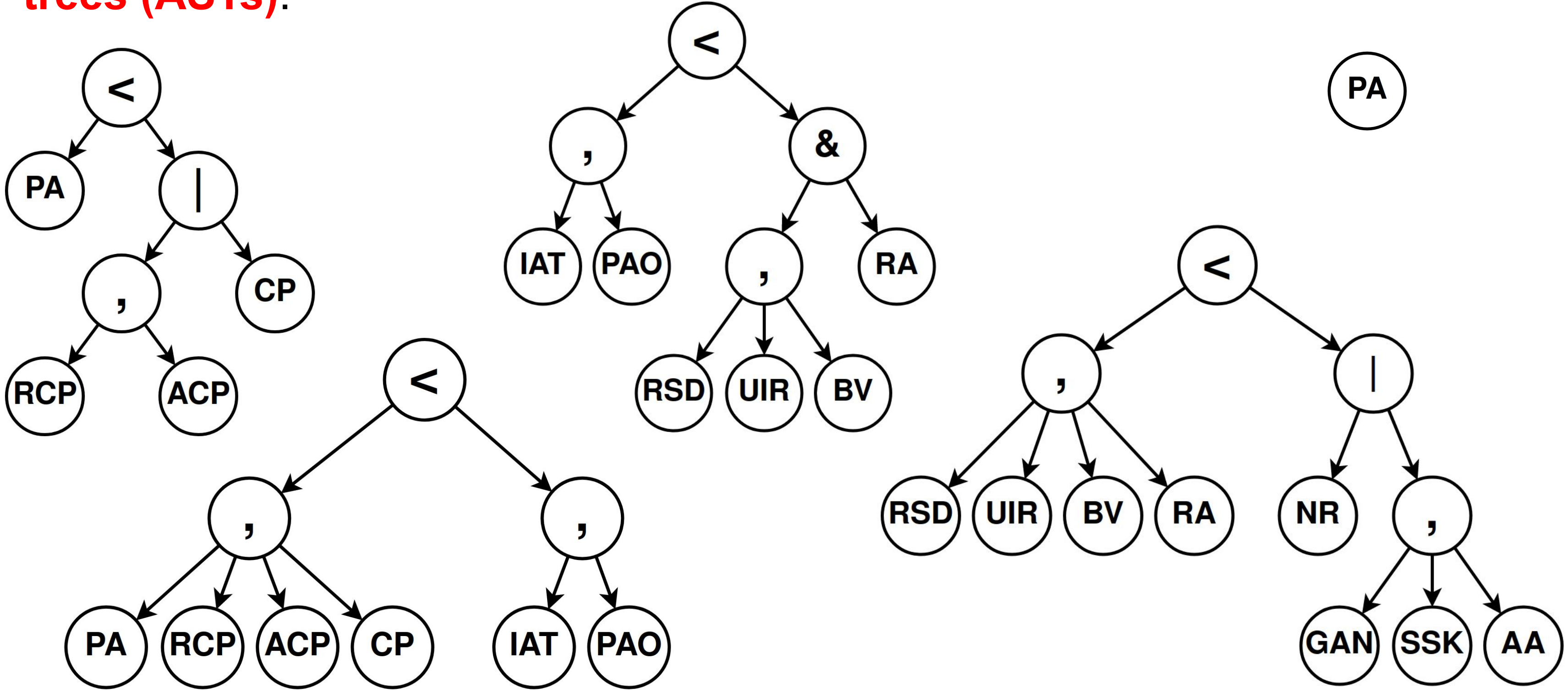
(ProcessApplication, RetrieveCustomerProfile, AnalyseCustomerProfile, CreateProfile) < (IdentifyAccountType, PrepareAccountOpening)

(IdentifyAccountType, PrepareAccountOpening) < (ReceiveSupportDocuments, UpdateInfoRecords, BackgroundVerification & ReviewApplication)

(ReceiveSupportDocuments, UpdateInfoRecords, BackgroundVerification, ReviewApplication) < (NotifyRejection | (GenerateAccountNumber, SendStarterKit, ActivateAccount))

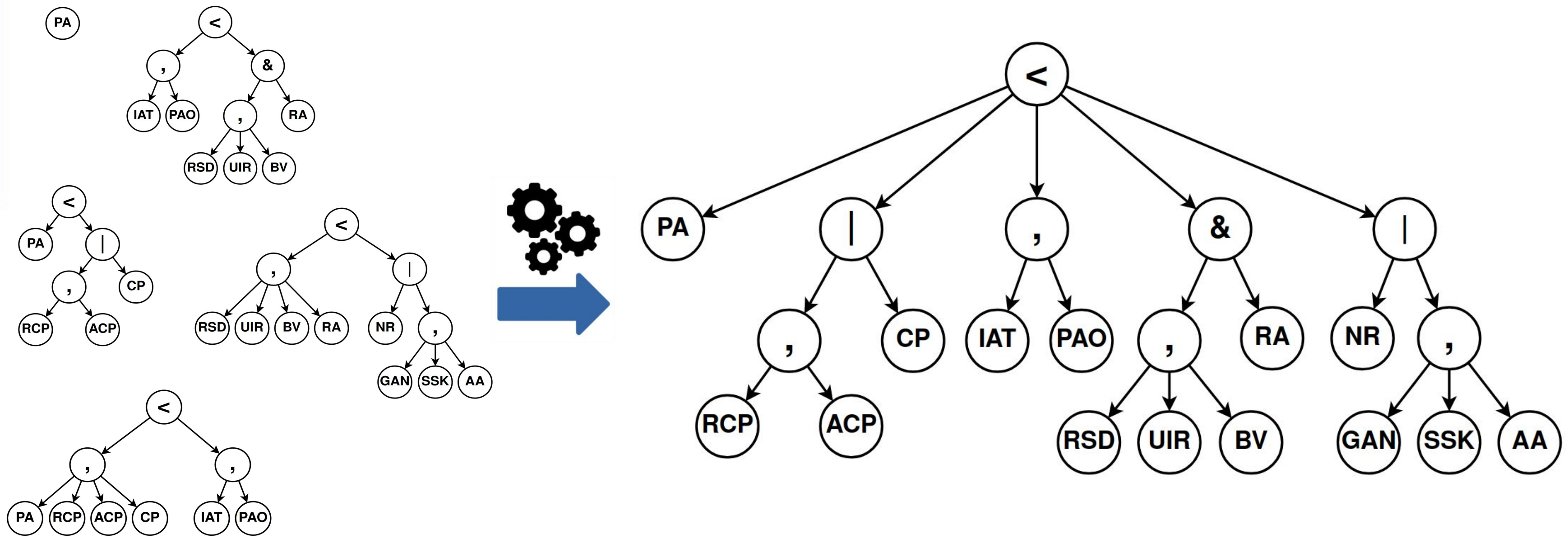
Detailed Solution – Step 4

These expressions are then **mapped to** their corresponding **abstract syntax trees (ASTs)**.



Detailed Solution – Step 5

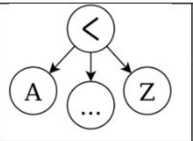
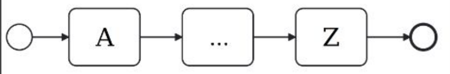
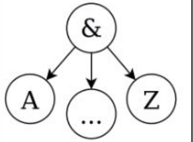
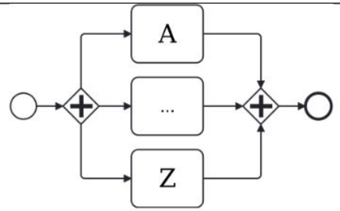
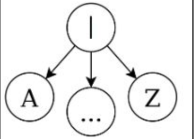
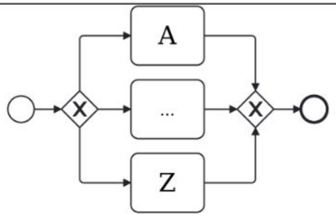
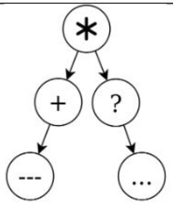
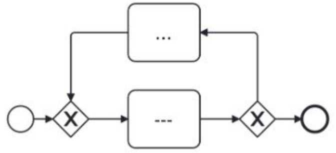
Each AST contains a **fragment of information** about the desired process.
To **gather this information**, the **ASTs are merged** into a single AST.



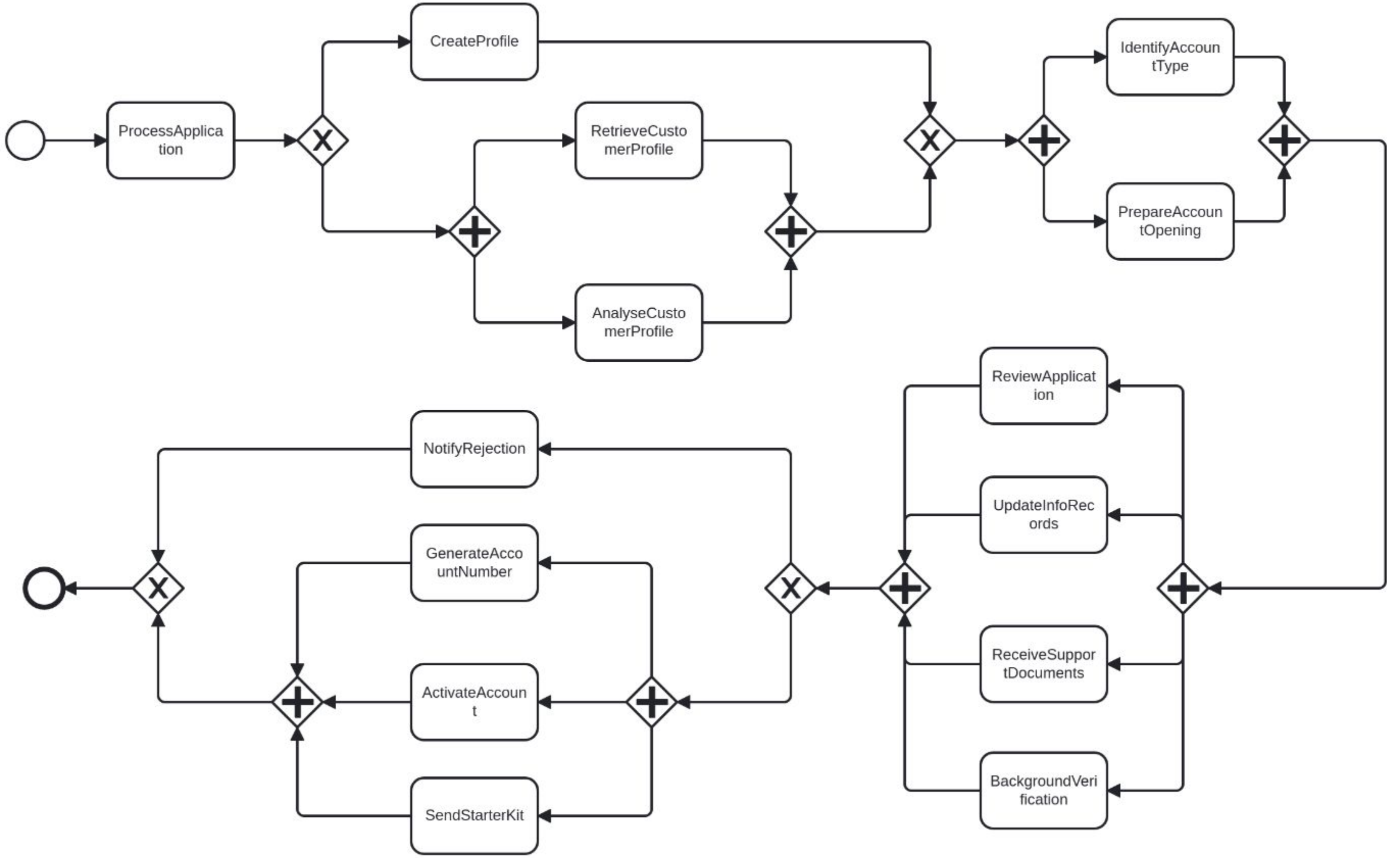
Detailed Solution – Step 6

Once the merged AST has been generated, the **BPMN process is ready to be generated**.

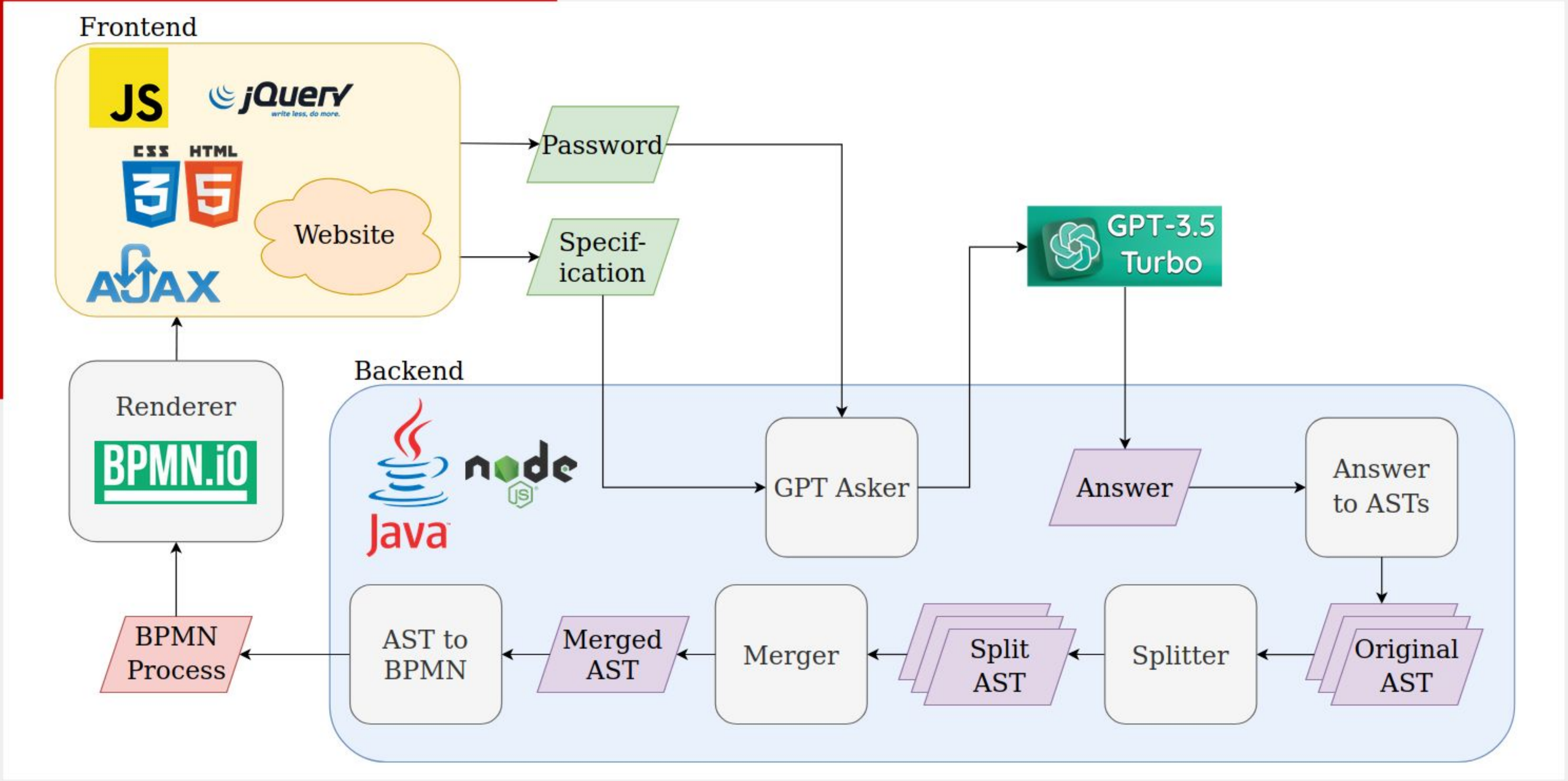
To do so**, patterns** are applied recursively to the merged AST, **starting from the deepest nodes** (leafs).

Pattern	AST	BPMN
(1) Sequential Pattern		
(2) Parallel Pattern		
(3) Choice Pattern		
(4) Loop Pattern		

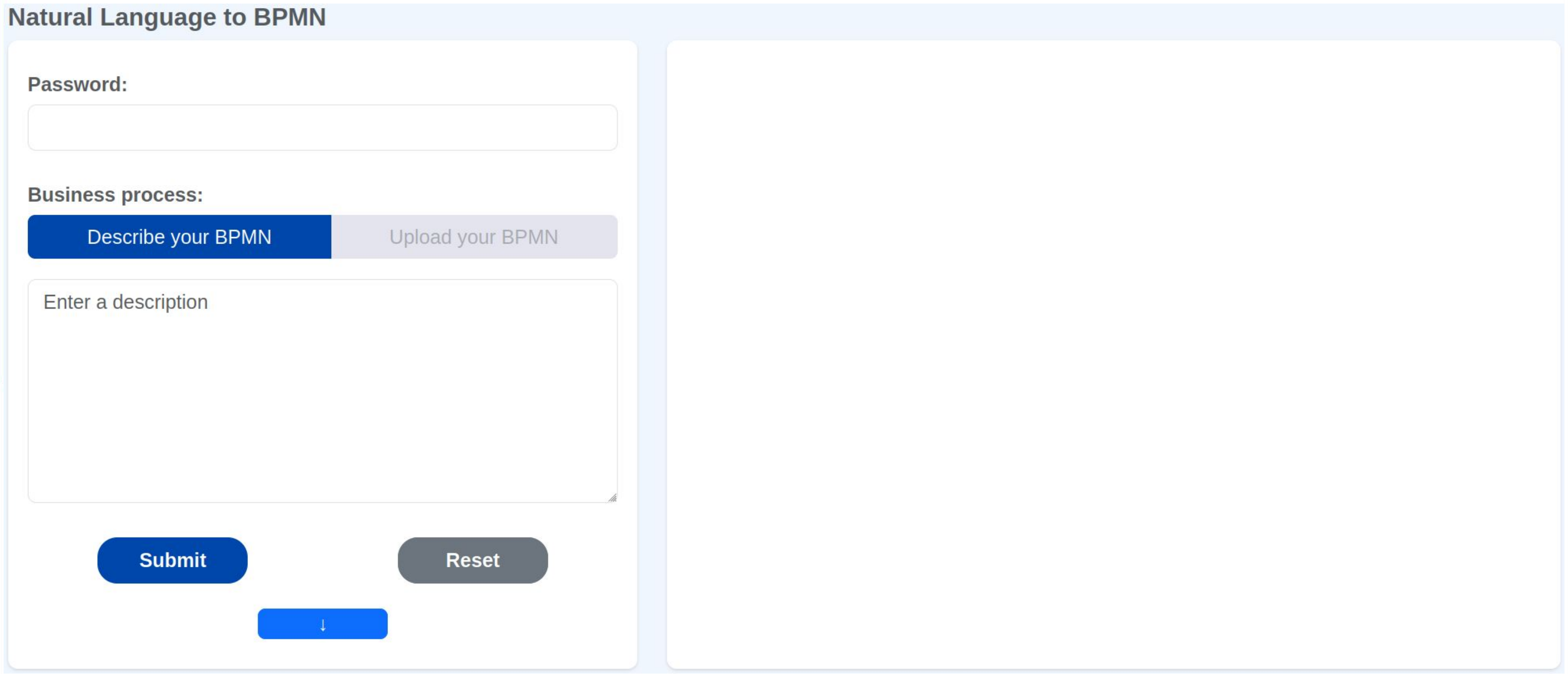
Detailed Solution – Result



Tool – Description



Tool – Online Version


Natural Language to BPMN

Password:

Business process:

Link: http://34.72.213.68:3000/

Conclusion

In this work, we proposed a **6 steps approach** aiming at automatically designing **syntactically and semantically correct BPMN** processes from a **textual description** of the requirements.

The main **perspectives** of this work are:

- Release the task naming constraints
- Improve the loops management
- Enlarge the supported BPMN syntax
- Improve the results