Business Process Intelligence

Project 2025

Crisis management: Flooding

By

Hritik Bikram Rawal

Moeez Ahmed

Mintesnot Yimer

Table of Contents

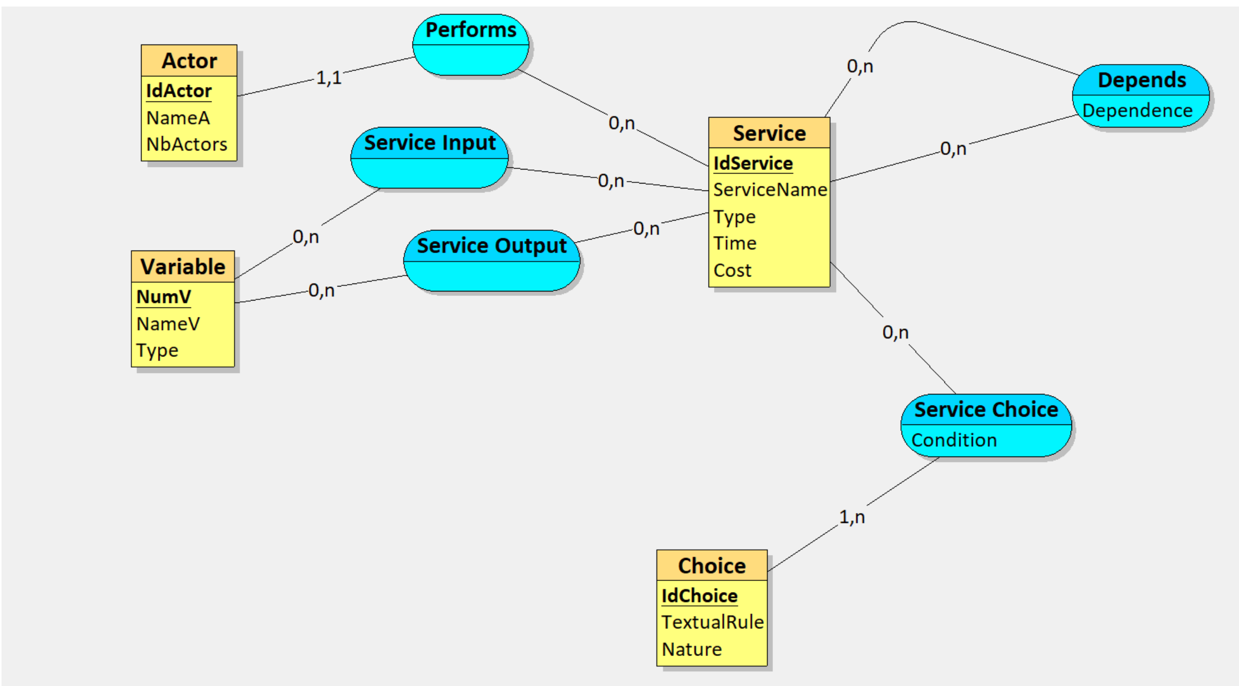
1) Introduction

Coordinating various stakeholders, quick decisions, and disseminating information on time are all necessary for flood crisis management. Business process intelligence (BPI) provides a powerful way to analyze, model, and optimize these essential procedures. During this project, we identified an inspired scenario process based on the real world, extract behavioral knowledge, automate it, and simulate performance in various organizational settings.

We utilized several softwares such as SQL, Python, ProM, Iterop, and the BIMP simulator to investigate how process mining and data-driven modeling can be used to improve public crisis response .

2) Data Representation of the universe of discourse

ER Diagram



This diagram was designed via. Looping and it was based on the table provided under (**IV. Appendix**) section of the project information.

The implementation of the innovative form was done with the use of React.Js for frontend and MongoDB for database. We used Node.Js, ExpressJs and mongoDB for the backend side (Server-side programming). This project has been hosted locally as the database has been used as a localhost (MongoDB Compass).

Flood Crisis Management

Resource Event All

Resource Form

ID

Name

Submit Resource

All Resources

ID	Name
1	Mayor

Flood Crisis Management

Resource Event All

Event Form

ID

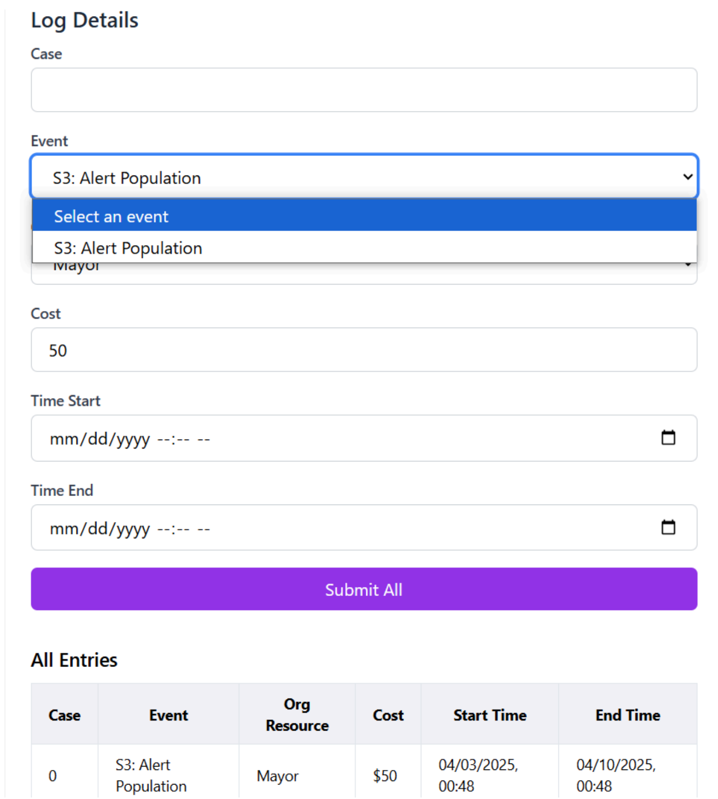
Name

Cost

Submit Event

All Events

ID	Name	Cost
S3	Alert Population	€50




```
_id: ObjectId('67f84ab3605c9aab7916c254')
case : 0
event : "S3: Alert Population"
orgresource : "Mayor"
cost : 50
TimeStart : 2025-04-02T22:48:00.000+00:00
TimeEnd : 2025-04-09T22:48:00.000+00:00
__v : 0
```


3) Discovering disaster management process behavior (Process Mining Task)

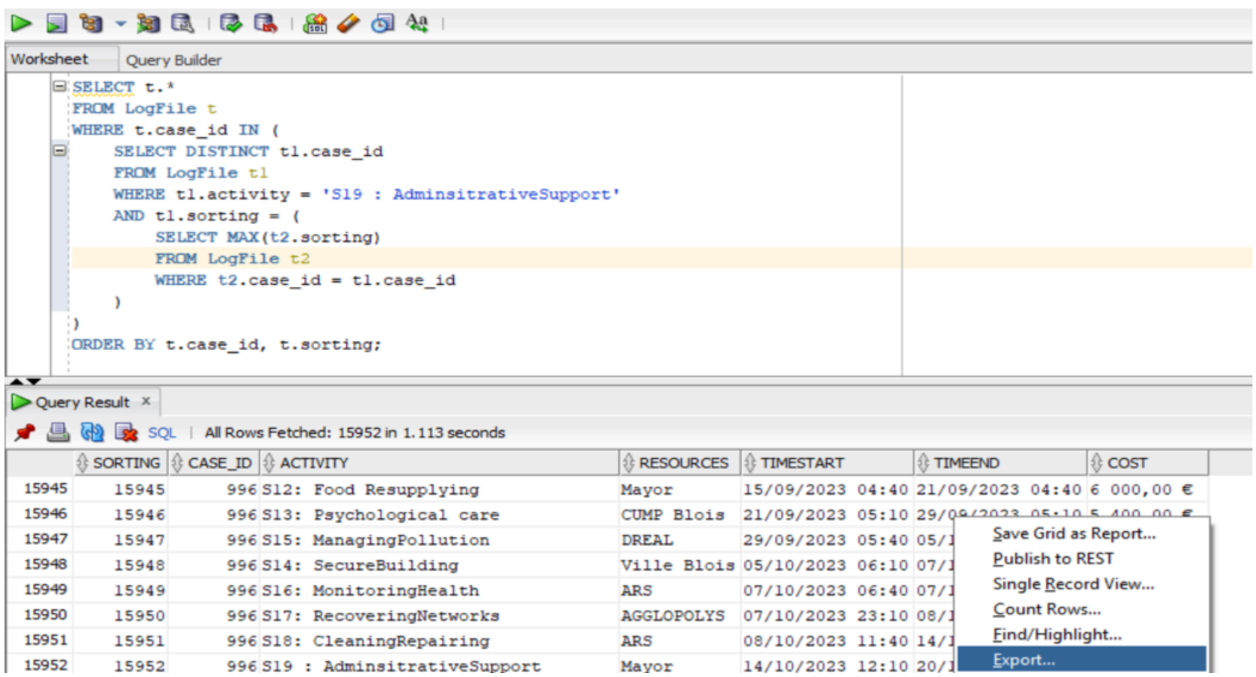
3.1) LogFile Cleaning.

We started our analysis by cleaning the log file to remove uncompleted cases using PROM and SQL. For sql first we imported the given log file to oracle database management system by using the insert import method and assigning appropriate data types for each column after that we write an sql query To extract the complete log details for all cases that conclude with the 'S19 : Administrative Support' activity, allowing further analysis or reporting on those specific cases. As a result the query returns 1592 rows.

We also tried to filter on simple heuristics in the prom inspector and the result is the same but we found that SQL is a better option in terms of data formatting and handling a special character like euro sign.

After that we exported the log files to do further analysis

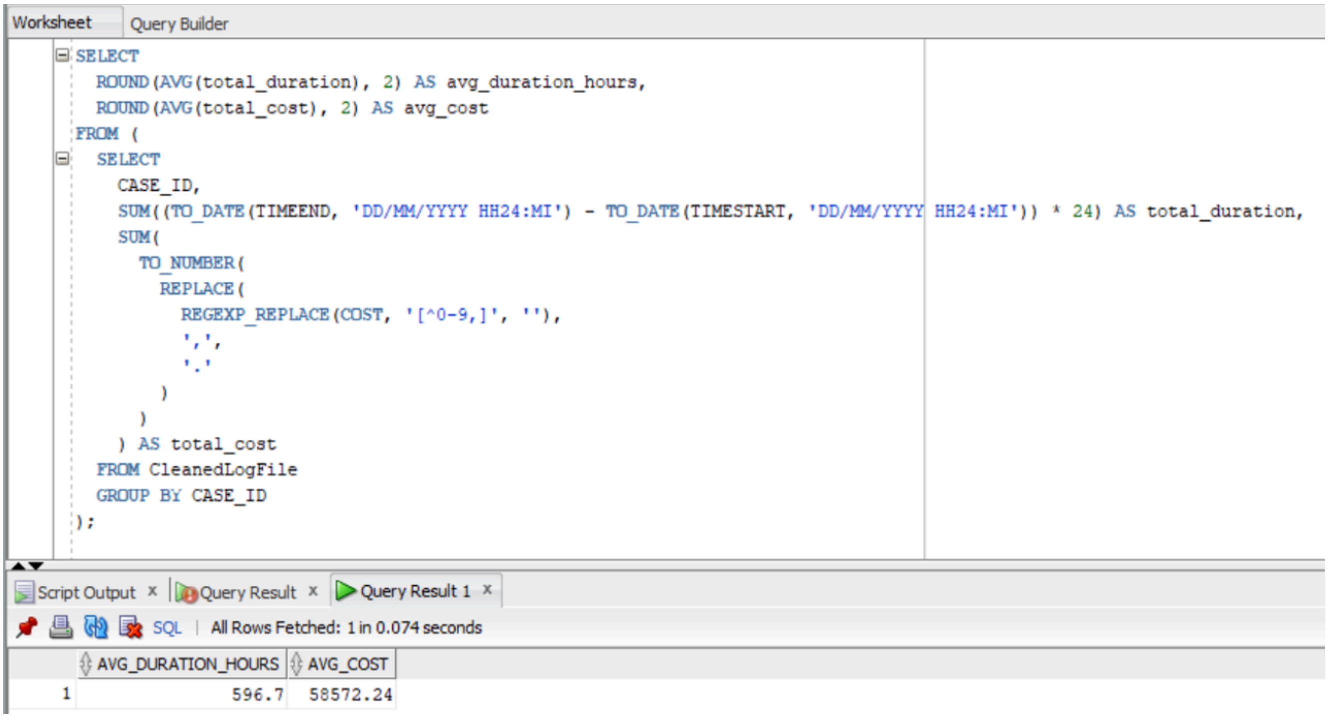
- Screenshot of the Query to clean the file and the export process of the cleaned log file.

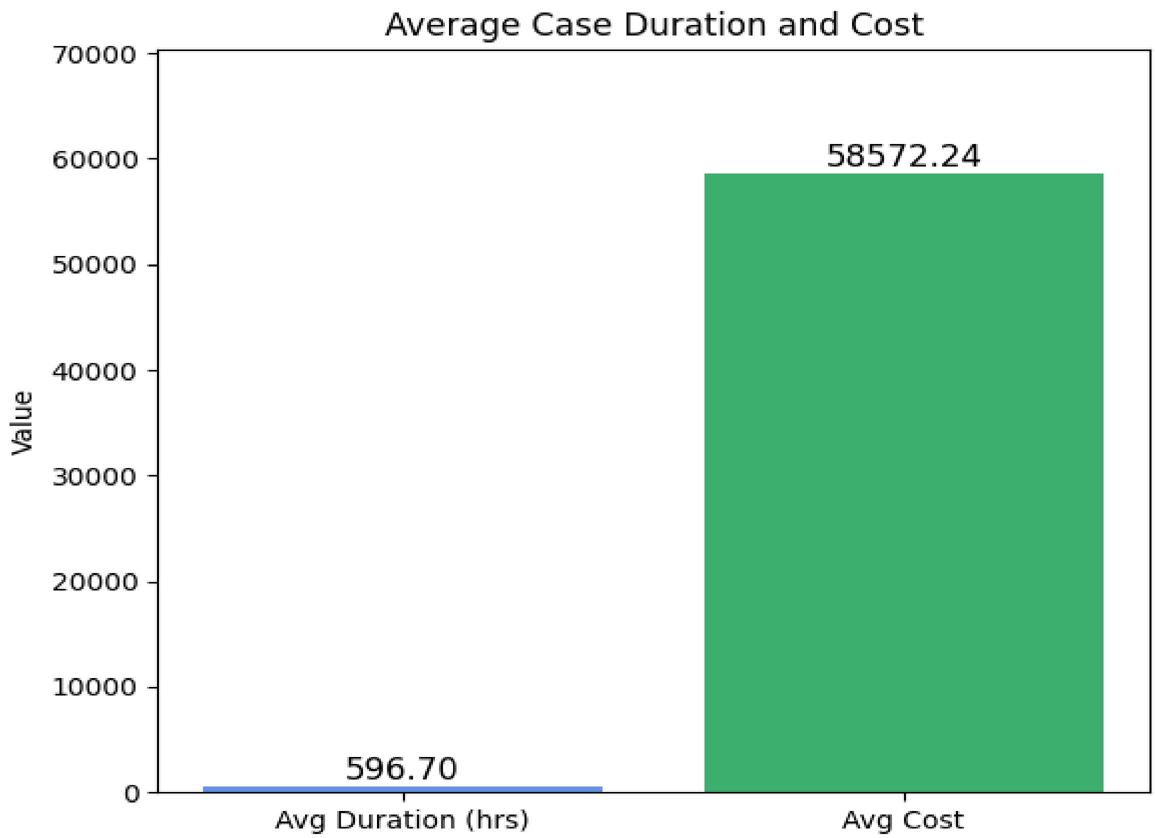


3.2) Calculate indicators to better understand the log file (SQL, Excel, R, etc.) and represent them with graphics (compare when possible, with Prom inspector)

a. Average Duration and Cost of a Case

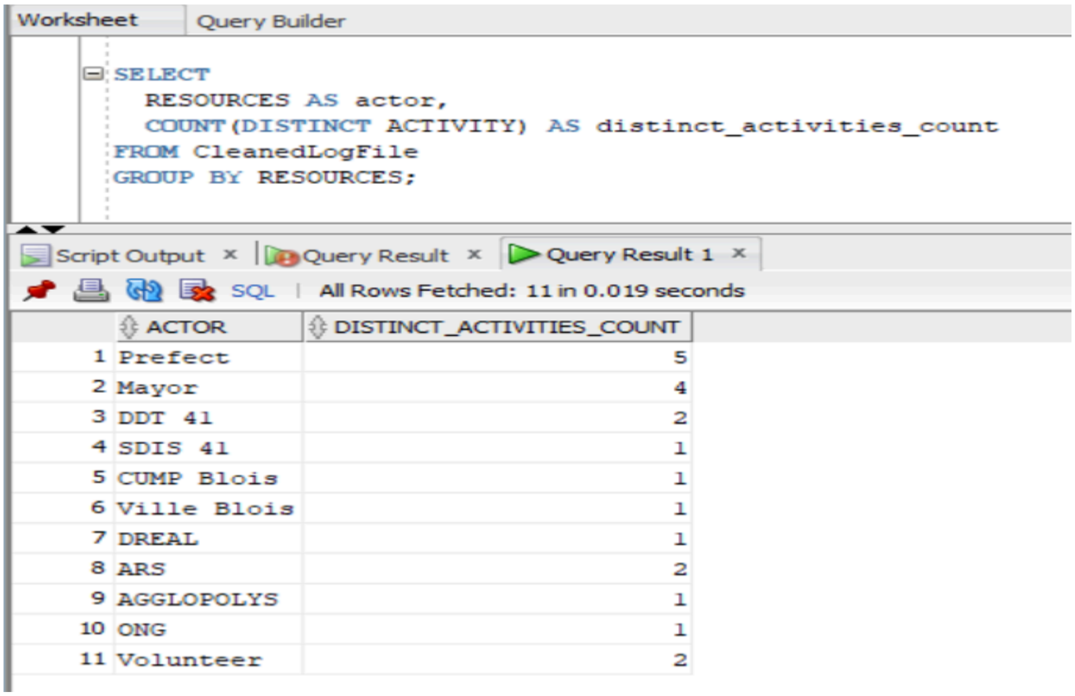
To calculate the average duration (in hours) and cost per case, first we aggregate per case (summing up the duration and cost of all activities), then compute the averages across cases. The expression (TO_DATE(TIMEEND, 'DD/MM/YYYY HH24:MI') - TO_DATE(TIMESTART, 'DD/MM/YYYY HH24:MI')) gives the difference in days; multiplying by 24 converts it to hours. The REGEXP_REPLACE(COST, '[^0-9,]', '') removes any non-numeric characters except the comma, and then REPLACE(..., ',', '.') converts the comma to a period. Finally, TO_NUMBER converts the cleaned string to a number. The outer ROUND functions ensure that the averages are rounded to 2 decimal places.





b. Profile of each actor: number of different activities he/she performed

This query lists each actor (from the RESOURCES column) with the count of unique activities they have performed. The query groups records by actor (using the RESOURCES column) then counts the unique activities for each actor.



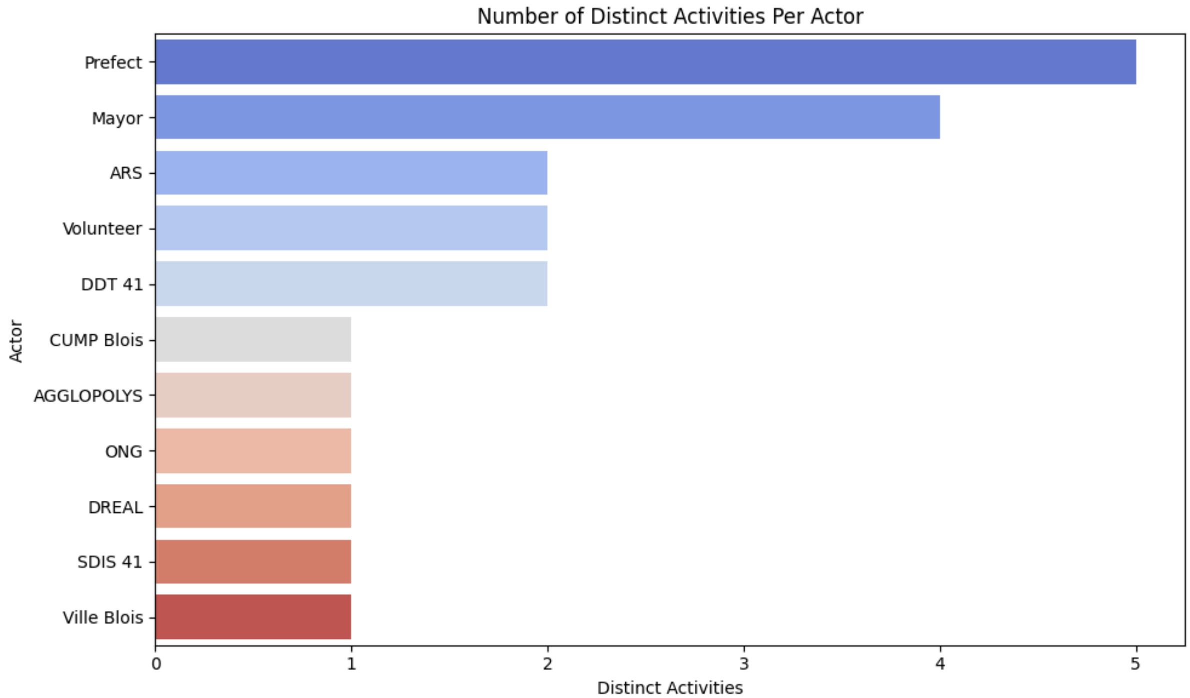
Worksheet | Query Builder

```sql
SELECT
  RESOURCES AS actor,
  COUNT(DISTINCT ACTIVITY) AS distinct_activities_count
FROM CleanedLogFile
GROUP BY RESOURCES;
```

Script Output × | Query Result × | Query Result 1 ×

SQL | All Rows Fetched: 11 in 0.019 seconds

	ACTOR	DISTINCT_ACTIVITIES_COUNT
1	Prefect	5
2	Mayor	4
3	DDT 41	2
4	SDIS 41	1
5	CUMP Blois	1
6	Ville Blois	1
7	DREAL	1
8	ARS	2
9	AGGLOPOLYS	1
10	ONG	1
11	Volunteer	2



c. Actor with the widest profile: actor who performs the maximum number of different activities.

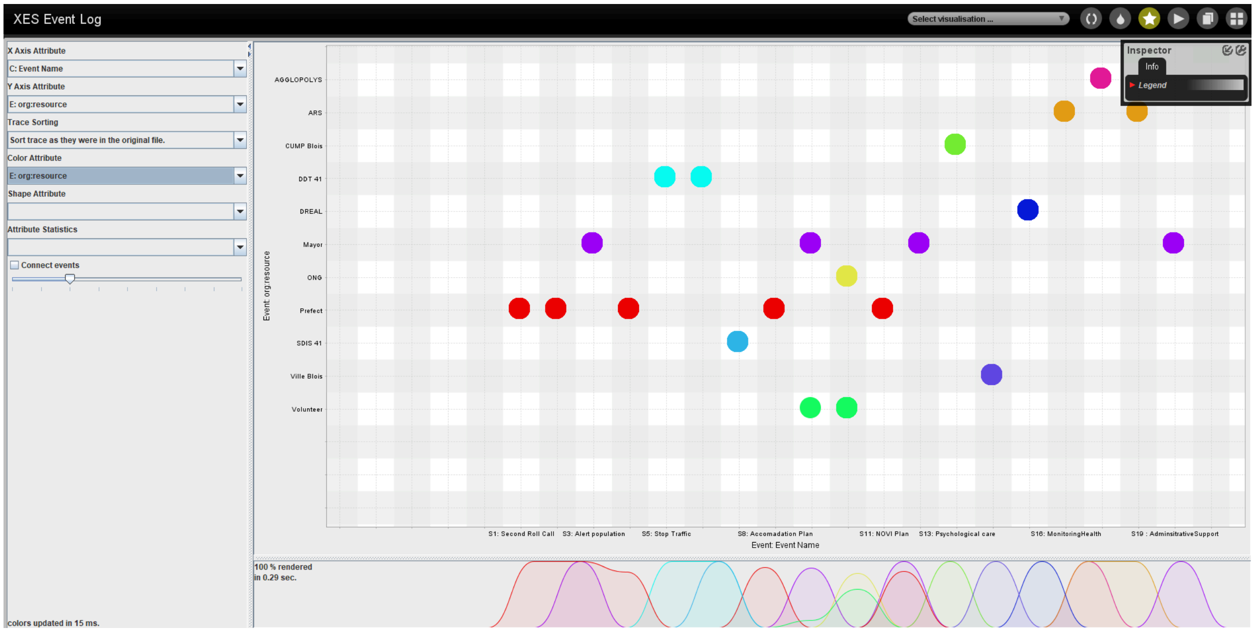
Worksheet Query Builder

```
SELECT *
FROM (
  SELECT
    RESOURCES AS actor,
    COUNT(DISTINCT ACTIVITY) AS distinct_activities_count
  FROM CleanedLogFile
  GROUP BY RESOURCES
  ORDER BY distinct_activities_count DESC
)
WHERE ROWNUM = 1;
```

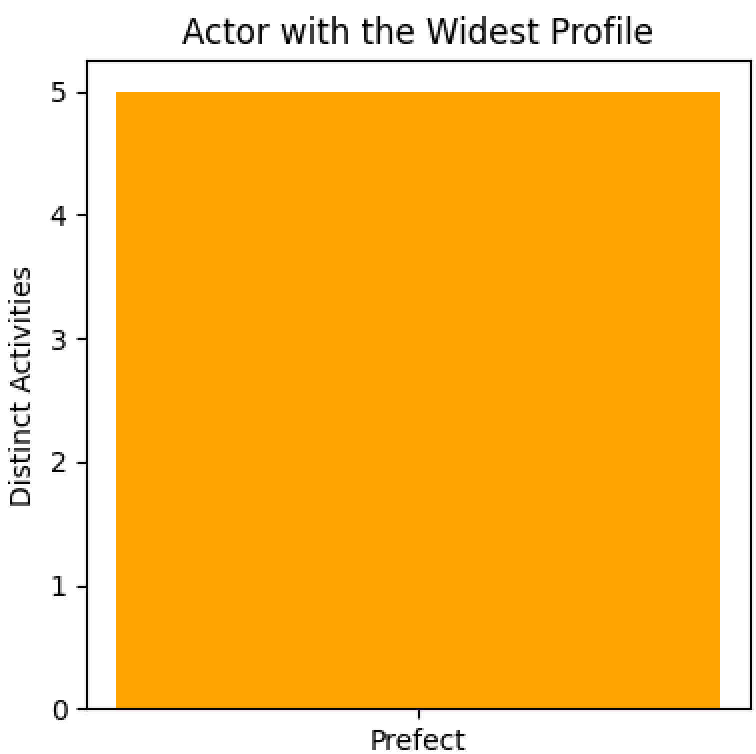
Script Output × | Query Result × | Query Result 1 ×

SQL | All Rows Fetched: 1 in 0.01 seconds

	ACTOR	DISTINCT_ACTIVITIES_COUNT
1	Prefect	5




```
Actor with the widest profile:
     Actor  Distinct Activities
7  Prefect                    5
```



By using these queries, we are able to extract meaningful indicators from our log file:

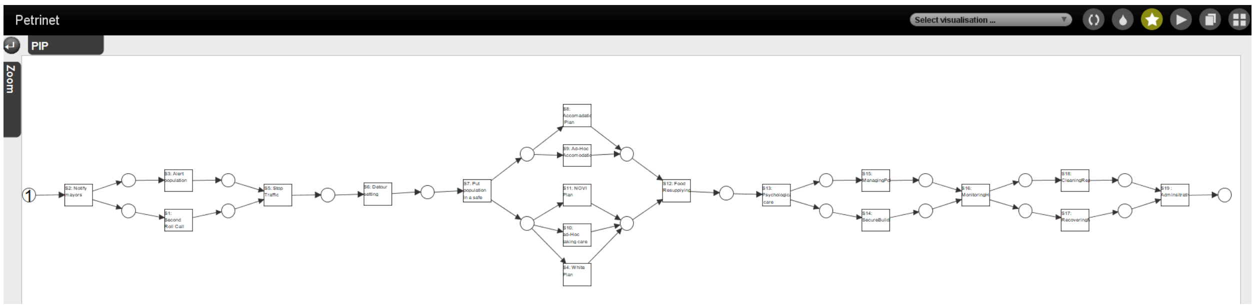
- The overall efficiency and cost per case (a).
- A clear profile of each actor's involvement (b).

- Identification of the most versatile actor (c).

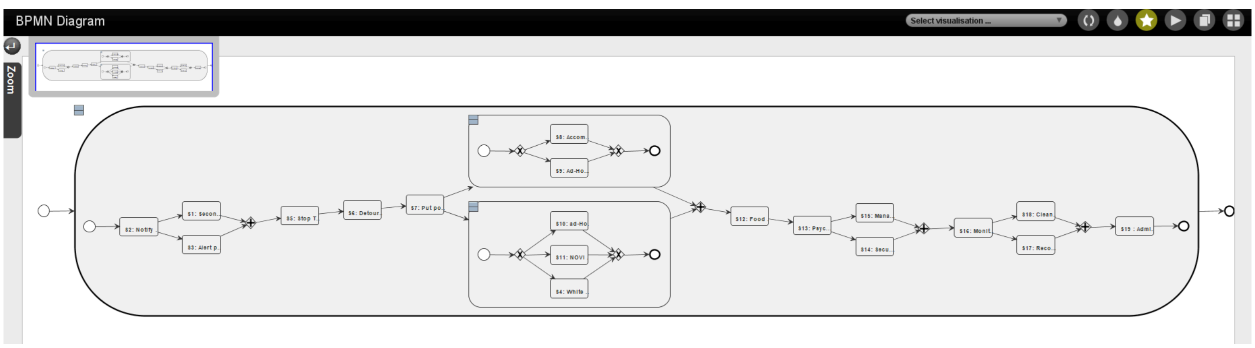
These indicators can also be visualized to offer a clearer picture for decision-makers, and comparing these results with Prom inspector can further validate and enrich our analysis.

3.3) Process Mining with PRoM

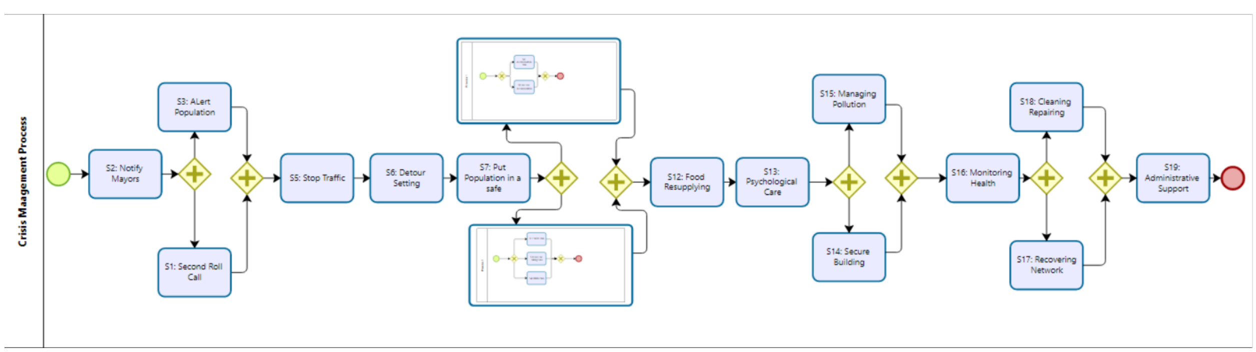
A) Process synthesized with the completed case with pertrinets

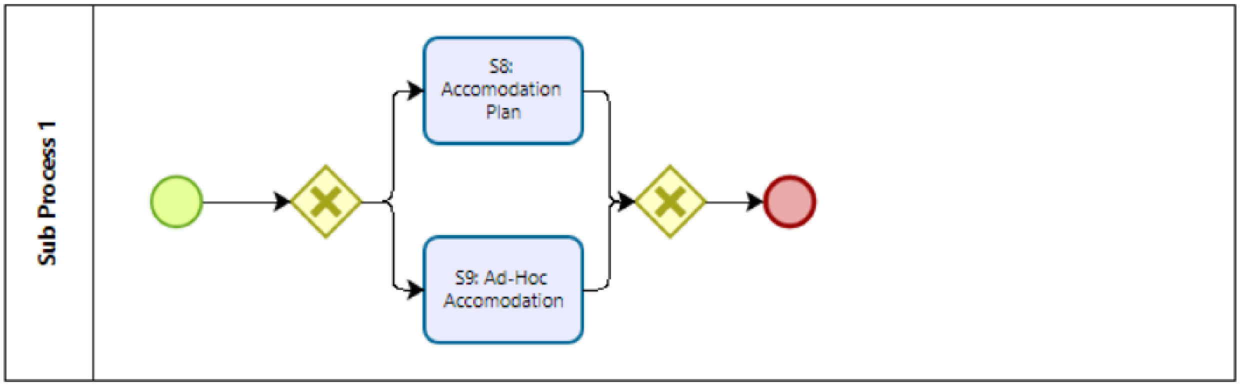


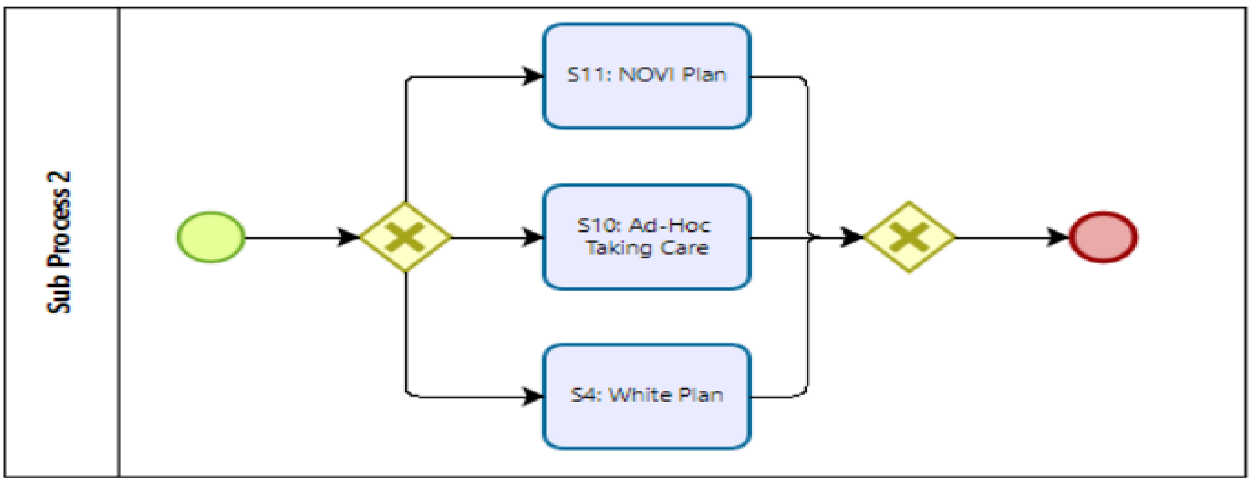
Transforming petri net into BPMN using PRoM



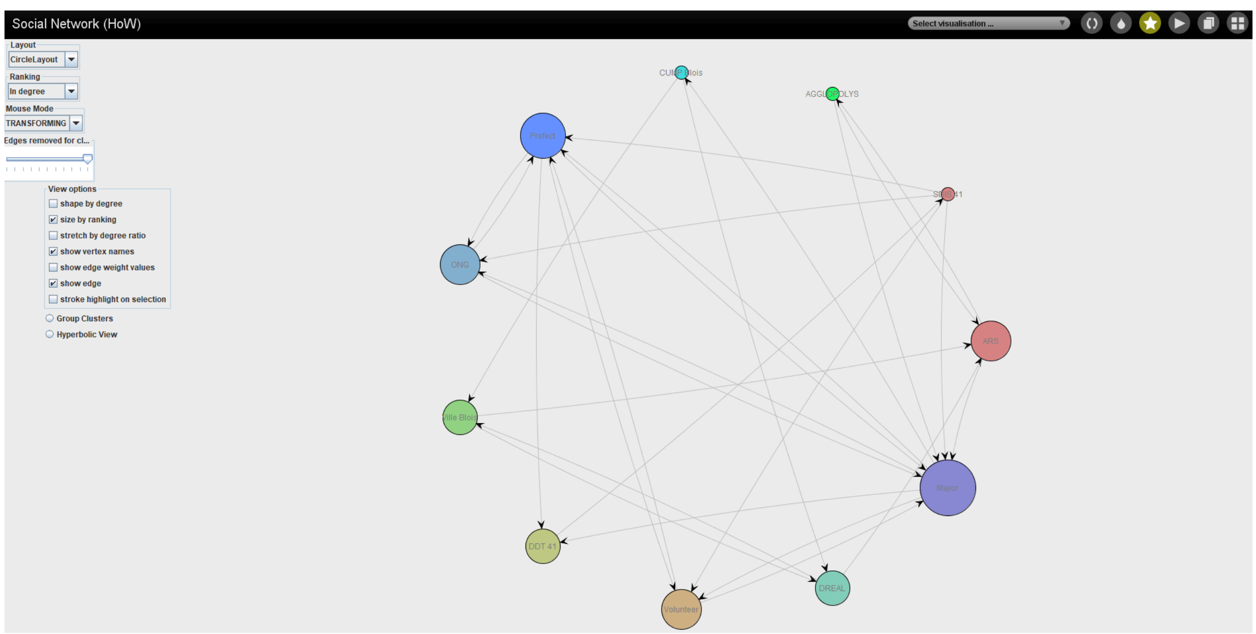
Cleaned process with Bizagi



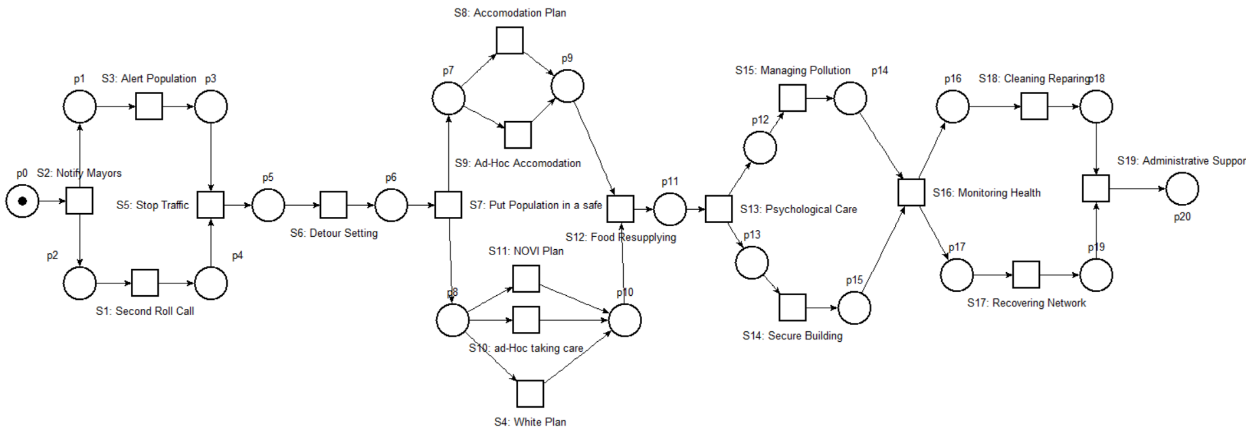




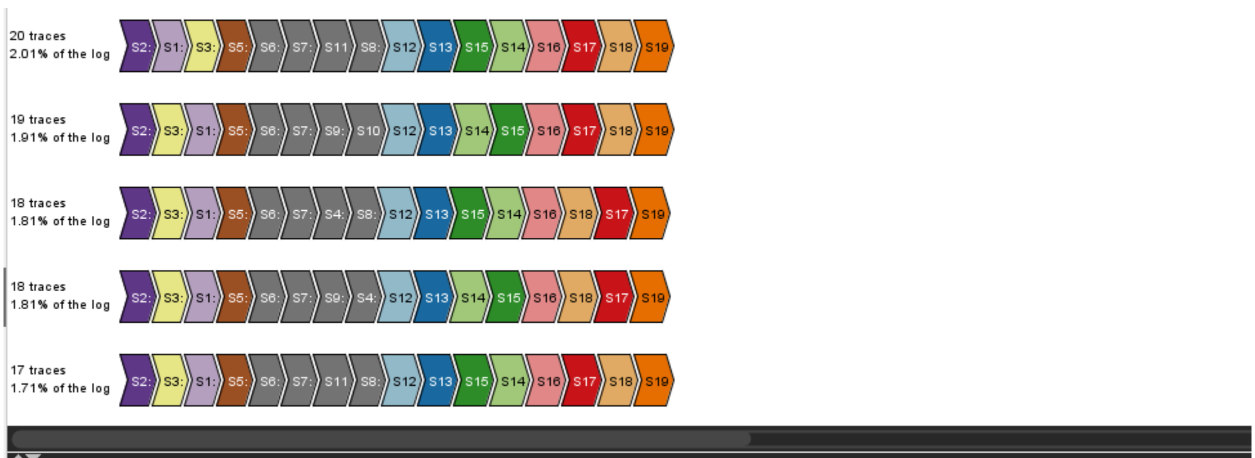
B) Social Network (Handover Relation)



C) Petri Net with Tina Software



Below is the screenshot for the number of traces:-



Traces	997
Events	15,952
Event Classes	19
Attributes	7
Variants	96
Events per Trace	16
First Event	292278994-08-17T08:12:55Z
Last Event	292269055-12-02T17:47:04Z

4) Analysis of conformity and deviations in Disaster Management Process (SQL and/or Programming Language)

a) Should we be worried?

Yes, the deviant trace represents a significant and concerning deviation from standard crisis management protocols.

There are key missing activities in the deviant trace (`BCEFGKLNOPQR` = `S2, S3, S5, S6, S7, S11, S12, S14, S15, S16, S17, S18`) compared to normal cases:

S1: Second Roll Call - Ensures accountability of municipalities.
S13: Psychological Care - Critical for victim mental health support.
S19: Administrative Support - Formal closure of the crisis.

Some Risks of this deviation are:
- Lack of coordination (no roll call).
- Delayed or missing psychological aid for victims.
- Legal/financial issues due to incomplete administrative closure.

b) Conformance Checking

The Levenshtein distance quantifies deviations by counting the minimum edits (insertions, deletions, substitutions) needed to align traces.

Results:

Case	Distance	Interpretation
162	5	Missing 5 critical activities (e.g., `S1`, `S13`, `S19`).
196	4	Similar deviation; lacks victim support steps.
202	6	Highest deviation due to alternate activity order.

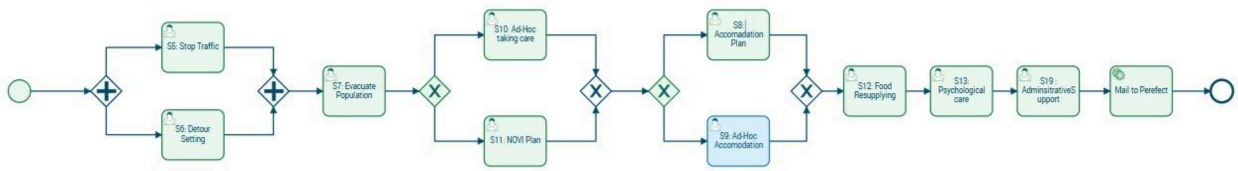
Limitations of Levenshtein Distance:
- Pros: Simple, quantitative.
- Cons: Ignores activity semantics (e.g., missing `S13` is worse than missing `S6`).

Alternative Methods:
1. Token-based similarity: Compare activity sets (not order).
2. Alignment-based conformance (Prom): Advanced process mining to pinpoint mismatches.
3. Rule-based checks: Flag missing mandatory steps (e.g., `S19` must always occur last).

The deviant trace violates core crisis management requirements. While Levenshtein quantifies deviation, combining it with semantic rules (e.g., "`S19` is mandatory") would improve analysis.

```
Levenshtein Distance for case 162: 5
Levenshtein Distance for case 196: 4
Levenshtein Distance for case 202: 6
```

5) Automating of the disaster management process with Iterop



1. Objective

To streamline and automate the emergency services segment of the flood management process, reducing manual coordination efforts and ensuring timely and efficient operations. The system concludes with notifying the Prefet via email upon process completion.

2. Assumptions

- Emergency services encompass tasks such as traffic control, population evacuation, accommodation planning, food resupply, psychological care, and administrative support.
- Relevant data points include:
  - Number of serious and light victims.
  - Availability of housing volunteers.
  - Safe locations and evacuation routes.
- Email notifications are mandatory for process conclusion.

3. Steps for Automation

3.1 Process Inputs

- Forms for data collection:
    - Serious Victim count
    - Light Victim count
    - Housing Volunteers
    - Cities affected
- Displayed Information:
    - Safe locations
    - Detour locations
    - Cleared areas

3.2 Task Automation

1. **Stop Traffic and Set Detours**
   Trigger tasks (`S5: Stop Traffic` and `S6: Detour Setting`) based on the location-specific forms for traffic control.
   Automated Logic: Affected areas are flagged for detour setup.

2. **Evacuation of Population**
   Execute `S7: Evacuate Population` using safe location inputs.
   Logic Gateway: Exclusive OR to decide on further actions:
    - `S11: NOVI Plan` (if Serious Victims ≥ 20).
    - `S10: Ad-Hoc Taking Care` (if Serious Victims < 20).

3. **Accommodation and Food Resupply**
   Assess housing volunteers and victim numbers:
    - Trigger `S8: Accommodation Plan` (if Light Victims > Housing Volunteers).

    - Trigger `S9: Ad-Hoc Accommodation` (if Light Victims ≤ Housing Volunteers).
      Manage food resupply under `S12`.

4. **Psychological Care and Administrative Support**
   Allocate victims to `S13: Psychological Care` and finalize with `S19: Administrative Support`.

3.3 Notification to Prefet

- **SMTP Configuration**:
  Send a notification email to the Prefet upon process completion.
  Email content includes:
    - Summary of actions completed.
    - Data on victim counts and resource utilization.

4. Execution and Testing

- **Execution**:
    - Deploy the automated process for real-time operations.
    - Ensure data inputs trigger corresponding workflows without manual intervention.
- **Testing Scenarios**:
    - Scenario 1: High serious victim count (triggers NOVI Plan).
    - Scenario 2: Low serious victim count (ad-hoc care is sufficient).
    - Scenario 3: Volunteers more or equal to the number of light victims (ad-hoc housing functions).
    - Scenario 3: Light victims exceed the number of volunteers (ad-hoc housing functions).

5. Innovations in Iterop Usage

- Implementation of Exclusive OR gateways for dynamic task allocation based on real-time data.
- Comprehensive email integration using SMTP for automated reporting.

- Consolidated decision-making by embedding logic conditions in the workflow.

6. Outcomes

- Reduced time and cost for emergency service coordination.
- Improved operational accuracy and resource allocation.
- Transparent communication with Prefect through automated notifications.

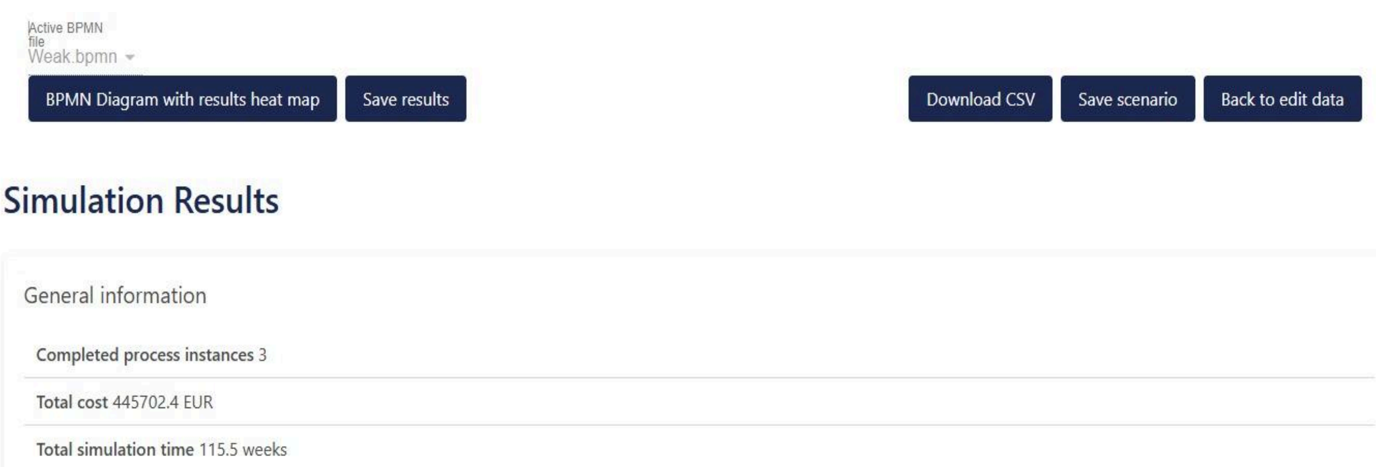
6) Simulation and resource calibration of the disaster management process with BIMP

To verify the performance during crisis response, we have executed it in the tool (BIMP) with 4 overlapping crisis cases.

Scenario 1 - Weak Organization (Baseline)

Inputs Taken:

- Minimal resources (e.g., 1 Mayor, 1 Prefect).
- Long activity durations (e.g., 30 days for administrative support, 25 days for psychological care).



- Limited task optimization (e.g., "Evacuate Population" with a mean duration of 2 days).

Results:
Couldn't simulate 4 crises because of *Arrival rate too long - process instances started over maximum allowed period (1000).*

- Completed Instances: 3 crises.
- Total Simulation Time: 115.5 weeks (~2.2 years).
- Total Cost: 445,702.4 EUR.

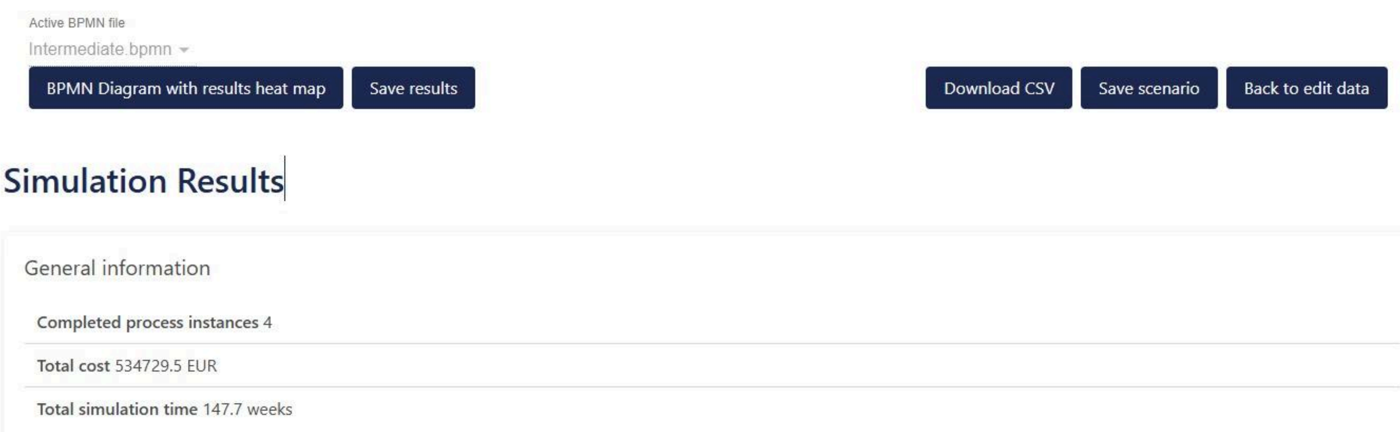
Analysis:
The baseline setup fails to handle overlapping crises due to insufficient resources and lengthy critical activities.

Scenario 2 - Intermediate Configuration

Input Adjustments:
- Increased Resources:
  - Mayors: 3 (from 1).
  - CUMPBlois: 3 (from 1).
  - DDT41: 2 (from 1).

- Reduced Durations:
  - Administrative support (S19): 28 days (from 30).
  - Evacuate Population: Mean duration reduced to 1 day.

- Parallel Task Optimization:
  - Food resupplying (S12) duration reduced to 1 day.



Results:
- Completed Instances: 4 crises.
- Total Simulation Time: 147.7 weeks (over 2.8 years).
- Total Cost: 534,729.5 EUR.

Analysis:
Improved resource allocation allowed simulation of 4 crisises so it reduced time and cost, but overlapping crises still exceed one year.

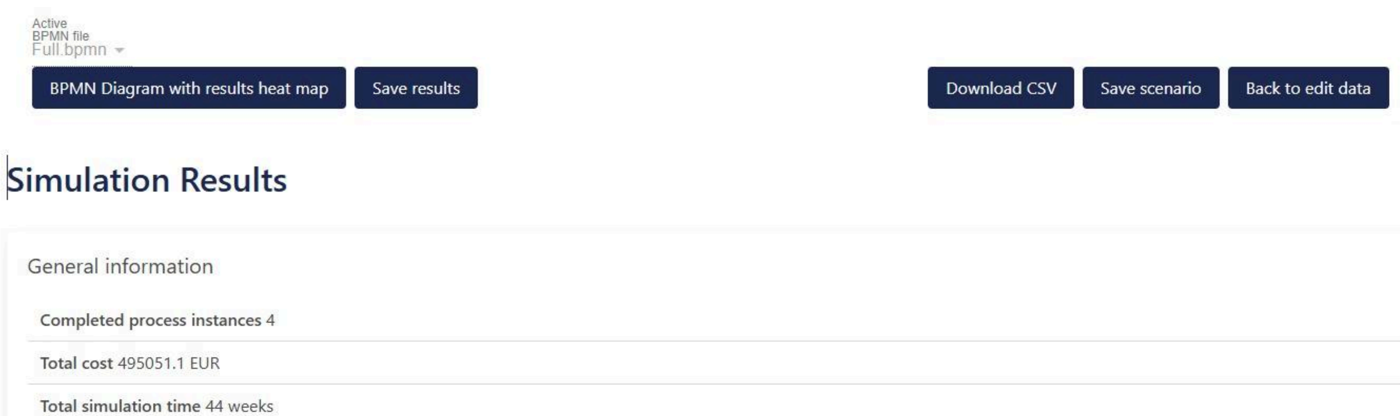
Scenario 3 - Optimized Configuration (Target)

Input Adjustments:

- Significant Resource Boosts:
  - Mayors: 8 (from 3).
  - Prefects: 4 (from 1).
  - CUMPBlois: 6 (from 3).
  - SDIS41: 2 (from 1).

- Critical Activity Streamlining:
  - Administrative support (S19): 25 days (from 28).
  - Psychological care (S13): Fixed at 25 days.
  - Detour Setting and Evacuate Population: Durations halved (1 day each).

- Increased Work Hours: Resources operate more than usual.



Results:

- Completed Instances: 4 crises.
- Total Simulation Time: 44 weeks (~10 months).
- Total Cost: 495,051.1 EUR.

Analysis:
This configuration achieves the goal of resolving 4 crises in under a year. Increased resources and reduced critical path durations enable parallel crisis management.

Key Takeaways:

1. *Resource Allocation:*
   1.1. Maintain at least 8 Mayors, 4 Prefects, and 6 CUMPBlois teams to handle overlapping tasks.
2. *Activity Duration Reductions:*
   2.1. Limit administrative/psychological activities to 25 days (use automated tools for paperwork).
   2.2. Streamline evacuation and detour tasks to 1 day via pre planned protocols.
3. *Cost Efficiency:*
   3.1. Despite higher resource costs, the reduced timeline minimizes long term economic impacts of prolonged crises.

7) Research Questions : How LLM could help BPI?

LLMs can analyze the data, automate routine tasks, and assist in decision making which allows the process to be faster, smarter and efficient. It can process large volumes of unstructured data to detect the patterns, inefficient information and insights in improvements. Automating repetitive tasks, suggestions for process optimization, and simulating outcomes can reduce the human workload and error. And embedding LLMs in BPM tools while offering real-time data can help those involved in the process to improve strategy and implementation. It will certainly make the process faster, accurate, and reduce cost. At the end it depends upon the user and how it is utilized to complement human decision making. With the innovation of LLMs, it is possible that there might be a self-optimizing business system in the future.

8) Conclusion

This project provided us insights of BPI in terms of crisis management. From cleaning and analyzing to modeling, automation and simulation, each phase offered practical insights to improve the efficiency. Using Python and SQL was effective at extracting and visualizing whereas PROM helped to validate the control flow and interaction among the actors in process mining. Implementing the process via. Iterop showed how automation can reduce the delay in communication and enable faster solutions.

The simulation in BIMP helped us evaluate the process under varying conditions: weak, intermediate and optimized organizational setup. The optimized setup confirmed that the overlapping crisis could be resolved within 44 weeks.

Overall, this project demonstrated how BPI tools and methods can be applied to real scenarios during the crisis. The combination of data-driven decision making, automation and scenario simulation can be beneficial to be better prepared for the crisis. With the dynamic simulation on real time data, integration of AI in decision making and user friendly tools for staff could be explored in future for better experience.