

Business Process Management

SS 2021

Exercise 6

Formalities for submitting your solution:

- Please submit your solution in OLAT
- The solution is due on **23.06.21 at 23:59 (UTC+2)**
- Please provide one single PDF file per group
- Please include the names of all group members into your solution
- You can reach up to 10 points in this exercise sheet

Consider the following EPC model data:

- $T_V = \{\text{Function, Event, XOR, OR, AND}\}$ is the set of vertex types.
- $T_E = \{\text{Function_Event, Function_XOR, Function_OR, Function_AND, Event_Function, Event_XOR, Event_OR, Event_AND, XOR_Function, XOR_Event, XOR_XOR, XOR_OR, XOR_AND, OR_Function, OR_Event, OR_XOR, OR_OR, OR_AND, AND_Function, AND_Event, AND_XOR, AND_OR, AND_AND}\}$ is the set of edge types, which are all directed (thus: $\text{Function_Event} = (\text{Function, Event, 1})$ and so on..)
- $T = T_V \cup T_E$ is the set of element types.
- $Z = V \cup E$ is the set of elements.

In the following, please provide the respective model queries in GMQL based on this data!

Task 1 (3 points)

Figure 1 shows an EPC pattern, which is syntactically correct but is a weakness. The pattern is called “AND-entry to loop” and depicts an AND join triggered by an external event and by a loop originating from the AND connector itself. The problem of this construct is, that the AND connector can never fire after we go through the loop (e.g., f_0 happens, then both e_1 AND e_2 happen, we follow the loop, e_1 happens again, we wait at the AND Connector but are stuck, because e_2 cannot happen again).

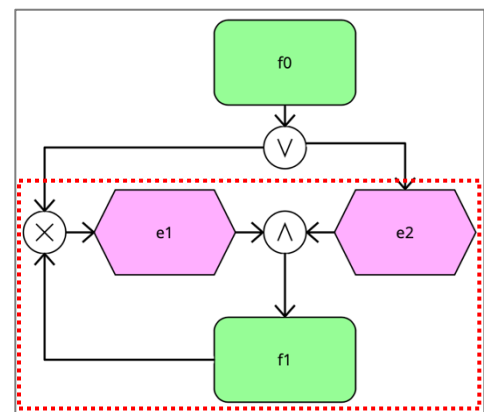


Figure 1: AND-entry to loop

Design a **GMQL** pattern that represents a joining AND connector that is an entry to a directed loop.

Note: This is an actual exam question from previous years.

Task 2 (3 points)

Design a **DMQL** pattern that is equivalent to your pattern from Task 1, i.e., that models a joining AND connector that is an entry to a directed loop.

Task 3 (4 points)

Figure 2 shows an organizational chart. As there is a directed edge from Anna to Bob, this means that Anna is the boss (superior) of Bob. This is the same for Anna and Andreas.

Figure 3 shows a short excerpt of a credit loan business process. As can be seen, each function is connected to a document (i.e. the loan application that is checked) and to a person (i.e. the person who performs this task).

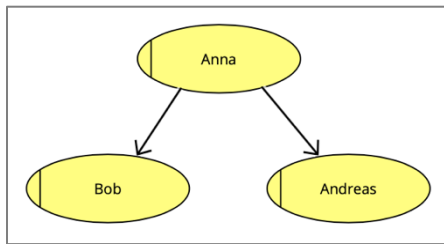


Figure 2: Organizational chart

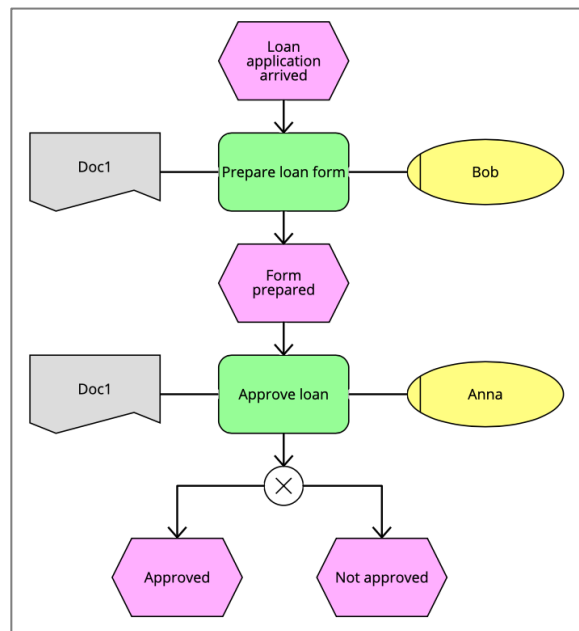


Figure 3: Loan approval process

The company wants to implement the so-called 4-eye principle. This means, that **approving** the loan **MUST** be performed by a person that satisfies **both** following conditions

1. a different person than the person who prepared the loan form
2. a person who is superior (i.e., the boss of the person who prepared the form)

This is a very common practice in banks, to ensure that important decisions, like approving a loan, are double-checked.

Design a DMQL pattern that can detect the 4-eye principle.

- You can assume that the business process is exactly as shown, i.e., the document is called Doc1 and the organizational units are called Anna and Bob.
- You can also assume that there are no loops in the process (as mentioned, just assume the process is “as-is”).

As a result, the pattern should be able to find paths in the EPC, where the four-eye principle can be detected.

Also, please explain why a DMQL query with your pattern would not find any results when ANNA was replaced with ANDREAS in your EPC (in the process).