Free University of Bozen-Bolzano – Faculty of Computer Science
Data and Process Modelling – A.Y. 2013/2014
Exam – 26/06/2014 ***Solutions***

This is a closed book exam: the only resources allowed are blank paper, pens, and your head. Explain your reasoning. Write clearly, in the sense of logic, language and legibility. The clarity of your explanations affects your grade. Write your name and ID on every solution sheet. Good luck!

1 Data Modelling

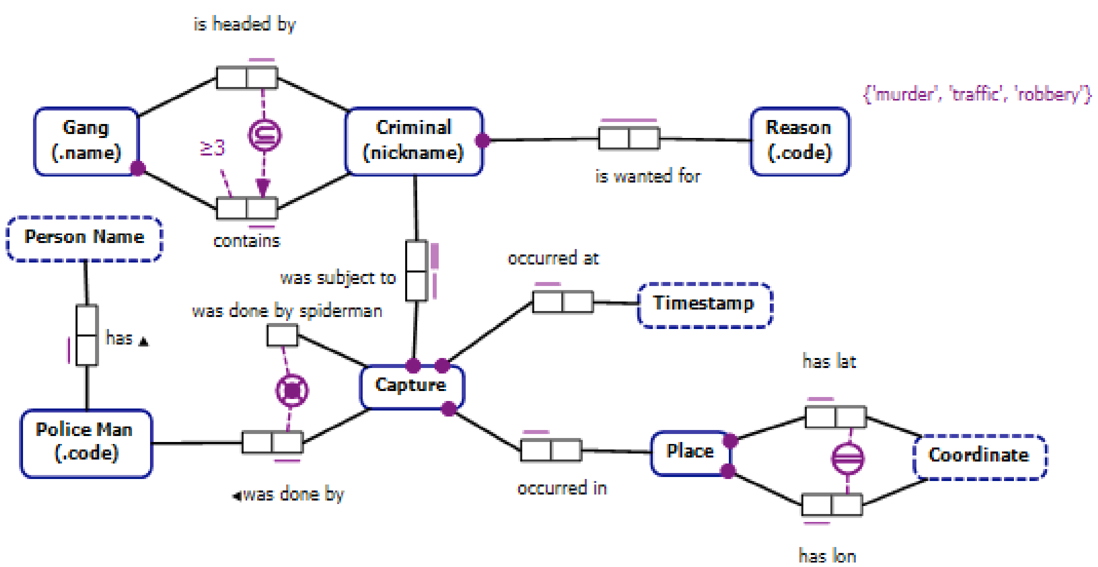
1.1 Design

The *Amazing Spider-man* is interested in building an information system to keep track of criminals acting in the city, and the arrest record of himself and the police.

A criminal is identified by a nickname, and is wanted for one or more reasons, which can be either "murder", "traffic", or "robbery". A criminal can be part of a gang, which is identified by a name. A gang contains three or more criminals, one of which is the boss of the gang. Each criminal may have been captured at a certain timestamp, in a given place (identified by a pair of coordinates: latitude and longitude). Two possibilities hold for a capture: either it was done by spider-man, or by one or more policemen. A policeman is identified by a code and could have a name.

Problem 1.1 [7 points] Design an ORM conceptual schema that represents the fact types, object types and constraints related to the domain described above. Remember to specify all required constraints, documenting the assumptions made when the text is not explicit.

Solution:



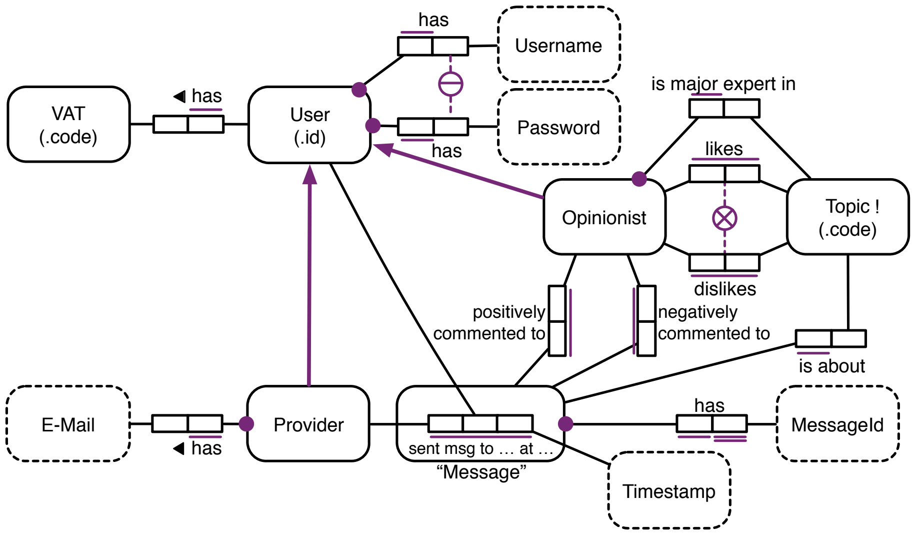
Problem 1.2 [2 points] We want to introduce in the conceptual schema a specific type of gang, i.e., a "dangerous gang". A dangerous gang is a gang whose boss is free. Argue whether this entity type is primitive or derived and, in the latter case, provide a derivation rule that defines it.

Solution:

`A Dangerous Gang is a Gang that is headed by a Criminal who was not subject to a Capture`

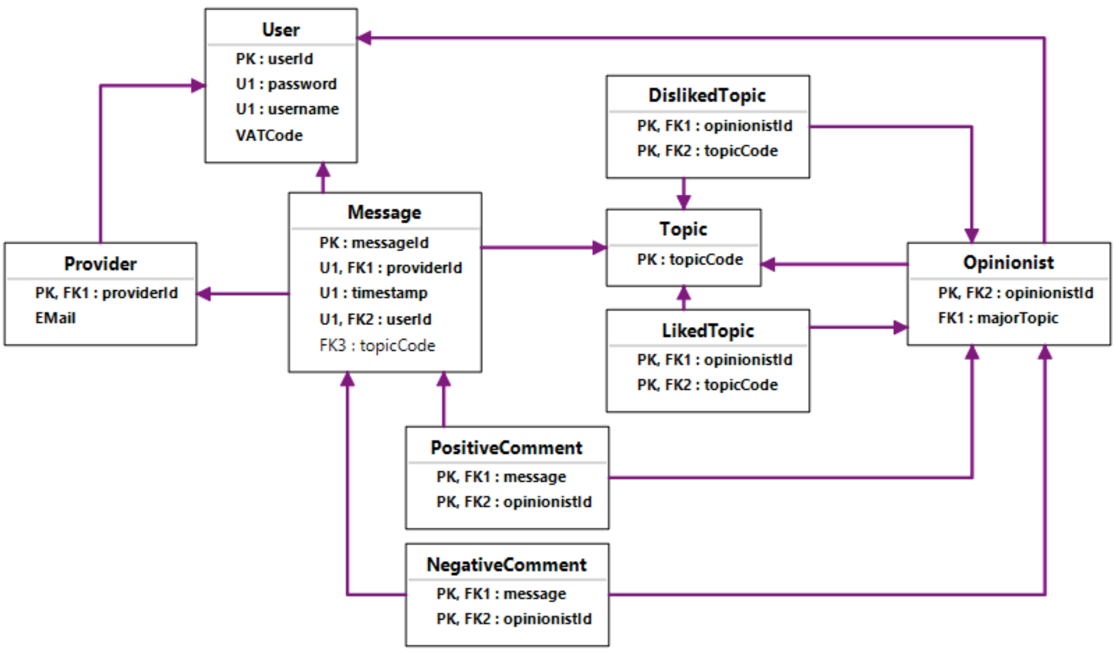
1.2 Relational Mapping

Consider the following ORM schema, modelling a fragment of the information system used for the analysis of messages and comments between web site users.



Problem 1.3 [7 points] Build a relational schema corresponding to the ORM schema, following the `Rmap` procedure and dealing with the User hierarchy following the **separation** strategy. For each relation schema that you produce, highlight the primary key, alternative keys, mandatory and optional attributes. Depict relevant constraints, in particular all foreign key constraints.

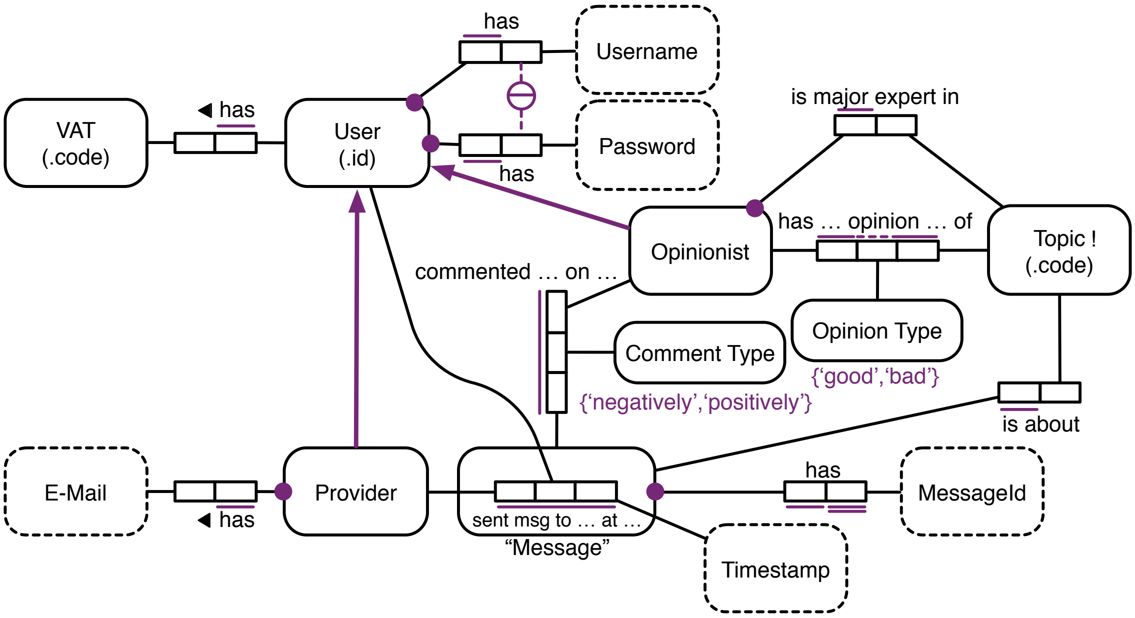
Solution:



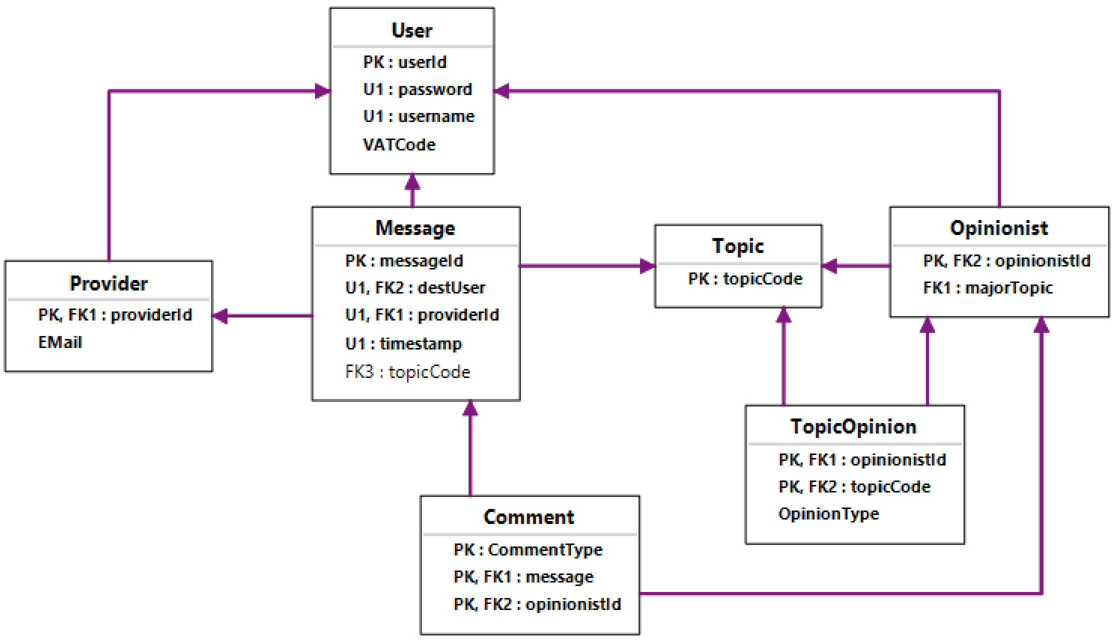
(An exclusion constraint must be added between the `DislikedTopic` and `LikedTopic` tables)

Problem 1.4 [3 points] By focusing on the Opinionist object type, show how its many-to-many relations can be transformed so as to reduce the number of tables produced by `Rmap`.

Solution: The `likes` and `dislikes` fact types are unified, adding a discriminator object type. A similar approach is followed for the two fact types `positively commented to` and `negatively commented to`. Notice the different internal UCs due to the presence of the exclusion constraint between `likes` and `dislikes`.



The resulting relational schema is:



(The corresponding value type contraints must be attached to columns `OpinionType` and `CommentType`)

2 Process Modelling

2.1 Design

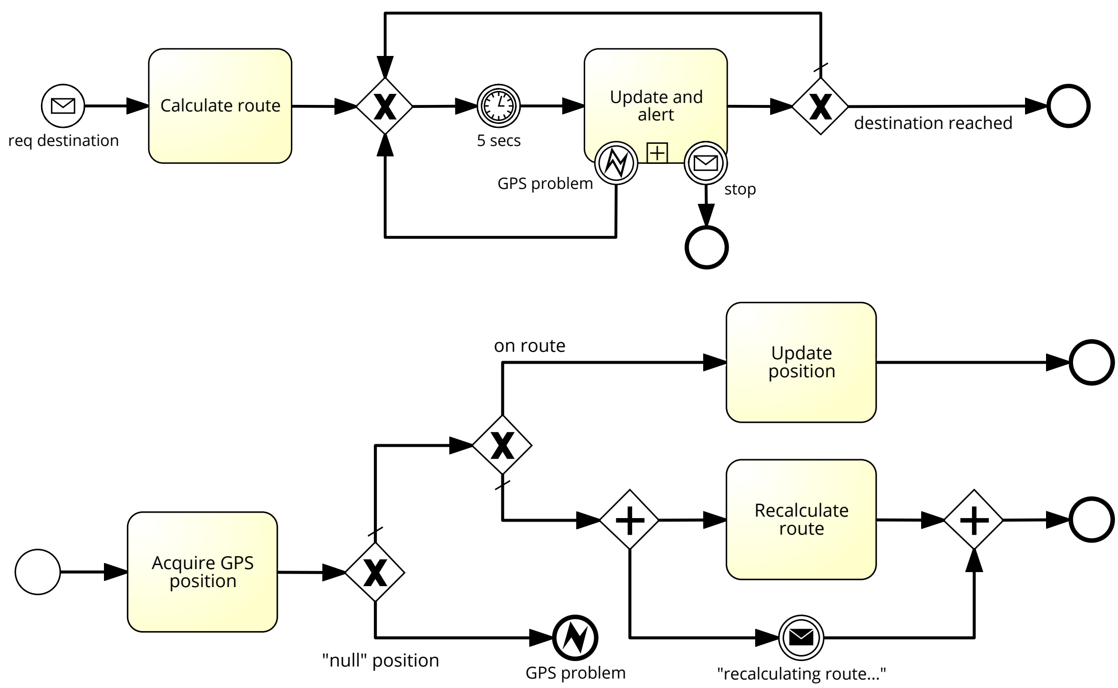
Consider the following simplified behavior of an automotive navigation system (e.g., TomTom). The process starts when the user communicates a destination to the system. The system react by calculating the route, then executing a loop, where each iteration is structured as follows:

1. The system waits for 5 seconds.
2. The systems runs a complex activity called *update and alert*. Two events must be caught during the execution of such complex activity: a *GPS problem* error, which leads the system to re-enter the loop; a *stop* message, which leads the system to terminate the process.
3. The system checks whether the destination has been reached; if so, then it concludes the process, otherwise it re-enters the loop.

The *update and alert* activity is structured as follows. The system first acquires the GPS position of the car. It then checks the obtained result: if it is "null", then the system triggers a *GPS problem* error. If instead it corresponds to a valid position, the system checks whether the car is on route or not. If so, then the system invokes an *update position* task. If not, the system runs a *recalculate route* task, and at the same time alerts the user that the route is being recalculated.

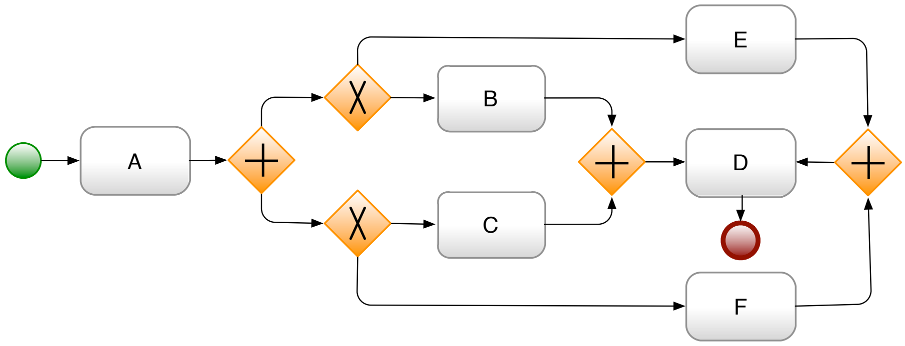
Problem 2.1 [7 points] Model the private process of the automotive navigation system using BPMN.

Solution:



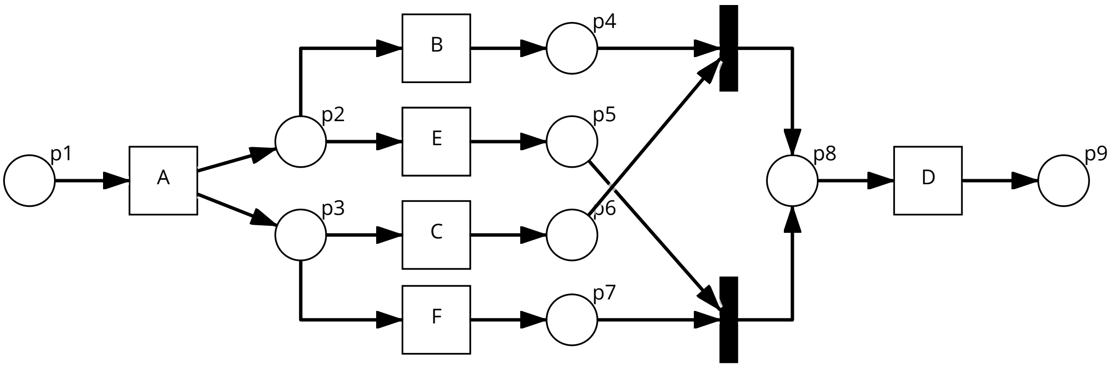
2.2 Analysis

Consider the following BPMN process:

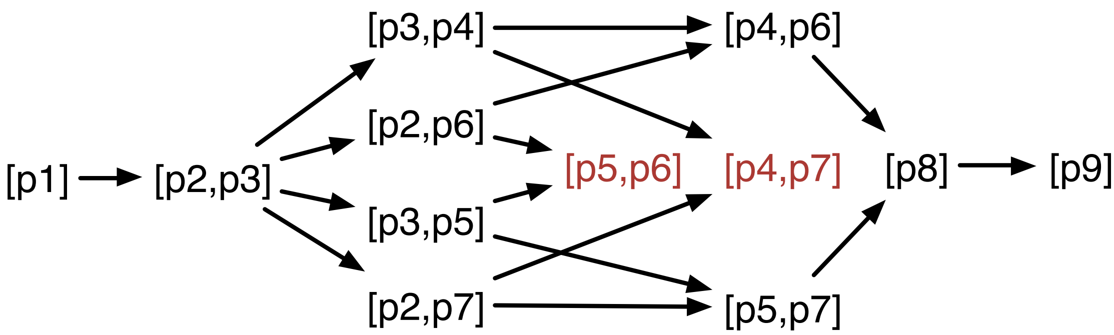


Problem 2.2 [4 points] Show how the process can be encoded in a workflow net.
Calculate the corresponding coverability graph, and use it to discuss whether the resulting net is *sound*.

Solution:



The reachability graph starting from the input state [$p1$] is as follows (we do not consider the short-circuited net, just the original workflow net shown before):



The two markings [$p5, p6$] and [$p4, p7$] represent two deadlock situations: they are reachable from the input state, but do not have any successor. In particular, they do not allow one to reach the output state [$p9$]. Therefore, the net is unsound.

2.3 Mining

Consider the following event log:

- $\langle a, b, c, d \rangle$
- $\langle a, e, c \rangle$

Problem 2.3 [4 points] Apply the α-algorithm on the log, defining the different sets introduced by the algorithm, and showing the resulting Petri net.

Solution: $L = \{\langle a, b, c, d \rangle, \langle a, e, c \rangle\}$.

	a	**b**	**c**	**d**	**e**
a	$\#$	$\rightarrow$	$\#$	$\#$	$\rightarrow$
b	$\leftarrow$	$\#$	$\rightarrow$	$\#$	$\#$
c	$\#$	$\leftarrow$	$\#$	$\rightarrow$	$\leftarrow$
d	$\#$	$\#$	$\leftarrow$	$\#$	$\#$
e	$\leftarrow$	$\#$	$\rightarrow$	$\#$	$\#$

- $T_L = \{a, b, c, d, e\}$
- $T_I = \{a\}$
- $T_O = \{c, d\}$
- $X_L = \big\{(\{a\}, \{b\}), (\{a\}, \{e\}), (\{b\}, \{c\}), (\{c\}, \{d\}), (\{e\}, \{c\}), (\{a\}, \{b, e\}), (\{b, e\}, \{c\})\big\}$
- $Y_L = \big\{(\{c\}, \{d\}), (\{a\}, \{b, e\}), (\{b, e\}, \{c\})\big\}$

$\alpha(L)$: