Contents

[1. Details on GDPR & DCR 1](#_Toc469243043)

[2. Our process – DCR model, DCR requirements - Modeling the GDPR requirements for data protection within Rejsekort A/S 5](#_Toc469243044)

[3. Reflection – the plusses of DCR, the lacks 9](#_Toc469243045)

[4. Conclusion 11](#_Toc469243046)

[Table 1 Graphical representation of activities 4](#_Toc469243047)

[Table 2 Graphical representation of relations 5](#_Toc469243048)

[Figure 1 DCR model of giving the consent for main purposes 7](#_Toc469243053)

[Figure 2 DCR model of getting assistance from the controller 8](#_Toc469243054)

[Figure 3 DCR model of termination of relationship 8](#_Toc469243055)

[Figure 4 DCR model of blocking request 9](#_Toc469243056)

[Figure 5 DCR model of giving consent for main purposes. The activities in group Mandatory are available 10](#_Toc469243057)

[Figure 6 DCR model of giving consent for main purposes. The activities in group Mandatory are blocked 10](#_Toc469243058)

[Figure 7 DCR model of Data breach entry. Two sub-process are spawned 11](#_Toc469243059)

[Figure 8 DCR model of the entity Data breach. Two sub-processes are spawned with StartNotification added in the figure 11](#_Toc469243060)

# Details on GDPR & DCR

* 1. GDPR
     1. Companies which want to process personal data should follow principles

One of the main points in GDPR is that the controllers are required to get consent for processing of the personal data per purpose. The purpose should be explicit and unambiguous and the data for which must be collected given purpose should be clearly specified (7). The aim idea is to give the data subject clear overview of what kind of data for what purposes is collected. On the other hand, getting consent per purpose gives the data subject flexibility not to give the consent for purposes, for which there is no legal basis and would not be an obstacle for the services provided by the controller. An example for the latter is the opinion research institutes which can get access to personal data for opinion surveys. The consent can also be withdrawn later.

Another focus in GDPR is that the collected data should be used only for the purposes, for which they are collected. Processing of data for other purposes is possible only if they are compatible, which would require further analysis. Using data for any other non-specified and non-compatible purposes would mean violation of the regulation law. (7, p.9 (50))

With the new regulation the controller is expected to require only a minimum set of personal data that are necessary for achievement of the purpose. Collecting unnecessary ones should be avoided.

Collected data should be stored as long as it has a purpose. If the purposes are achieved and part/all of the collected data is no longer necessary, then they no longer should be kept.

(Malik) The right to be erased (“the right to be forgotten”) known from the Data Protection Directive 95/46/EC will be expanded with the new regulation. This law requires deletion of personal data without undue delay. This new law covers besides the cases when the data was processed unlawfully, also the situations where the controller has no longer need of the personal data for the original purposes and no legitimate reason the process them.

(known) Before processing all the personal information should be accurate and up-to-date. (p.5)

Each personal data breach which can lead to material or/and non-material damages to natural persons should be notified to the supervisory authority. When the processor of personal data becomes aware of such a data breach, it is expected to inform the controller immediately. The controller is required to inform the supervisory authority within 72 hours. This notification should include details about the kind of the breach and where it occurred, how many persons are affected, the possible consequences and measures which will be taken to address the security gap. If the data breach poses risks for the rights and freedoms of the natural persons, there appears a necessity for controller to notify the data subjects as well.

1. Rights of the data subjects

(Malik)GPDR introduces a new right concerning the data portability. This is that each person should be able to get structured, machine-readable copy of the data for the purposes pointed by controller, so that the data subject can take and transfer the collected personal data to other service providers. This also makes it easier he or she to have more control over the processed data. The new law could lead to some extra burden in the companies because of the additional expenses to implement a new system, dealing with the requirement. (Article 20)

Another area related to the rights of the data subjects extended in GDPR is the right to get support from the controller to exercises his/her rights. With the new change the recommendations of the Directive to help the data subject becomes an obligation, which the controller should comply with. The controller should take into account all the new rights given to the data subject and implement appropriate measures to reflect their expectations. This could include training of the employees to respond as fast as possible to data subjects’ requests. (8)

Right not to be profiled – “1. The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarly significantly affects him or her. … Paragraph 1 shall not apply if the decision … is based on the data subject's explicit consent.” (7, Article 22, paragraph 1,2.). As it is stated, the data subject can object to any automatic processing, including profiling, except the case, when the contest is given. GDPR has the same requirements about profiling like Directive 94/45/EC with some minor amendment and the addition that profiling is possible if the consent is extant.

* 1. DCR

Here, we will represent and describe in more details DCR graphs that we used to model our process. DCR (Dynamic Condition Response) graph is a declarative, constraint-based business process modeling language. In the declarative languages like DCR all the flows in a process are implicitly defined, based on the constraints between the events. The number of the flows is restricted only by these constraints. (3)

A DCR graph is a directed graph, where the nodes, drawn like boxes, represent the events, or activities, and the arrows - the relations between the events. The representation of any process using DCR notations is simply called graph. Each DCR graph represents a process, which includes at least one activity.

The activities play main role in DCR graphs that represent an action to be done. They are connected by one or more edges, which represent the constraints between them. Each activity allows nesting, explained later in this section. An activity can be in one of three states – executed, pending or included/excluded and it could be only in one state at a time. Details and graphical representation of the activities and their states is given in table X.

There are 4 *relations* between activities defined in the first version of DCR graph – condition, response, inclusion, exclusion (2). The relation starts from an activity and ends in another activity or in the same one. Later the graph is extended with notion of milestone. To explain the meaning of the relations we will use the notion of DCR Workbench for simplicity. The graphical representation of the relations can be seen in table X:

* A condition A -[k]->\* B represents the constraint that for event B to happen, A should be executed at least k steps before B (5). When there is no requirement for numbers of the steps, then this condition can be represented in the following format: A -->\* B. If A is excluded, then A is no longer requirement for B, neither the requirement for the steps is valid. After re-inclusion of A, the number of the steps will be reset to k, no matter of the value of k before the exclusion.
* A response A \*-[d]-> B sets an effect that when A happens, B becomes obligated(pending) and should happen at last d steps after the execution of A (5). Here *d* imposes fixed deadline, but in its absence the event B is obligated to happened eventually, without any concrete deadline. It the case of the latter the relation could represented as follow: A \*--> B. The deadline can’t be zero i.e. can’t happen immediately.
* An exclusion A --% B is an effect that when A happens, B is excluded and can’t be executed (5). When B is excluded, it is no longer condition. If B was a pending obligation before exclusion, it need not happen after exclusion, unless not re-included.
* An inclusion A--+ B is an effect that when A happens, B is re-included. Now B can be executed. If B is condition for another activity, after re-inclusion it becomes valid.
* A milestone A --<> B is a constraint that for event B to happen A should be not pending or excluded (5). So If A is included, but not pending, B can also happen.

There is one more relation called “s*pawn*”, but unlike the other five relations it specifies creation of sub-process than setting a constraint. The spawn relation is only possible between an activity and a sub-process, but not between two activities or to the same activity.

Graphs allow assigning a *role* to the events, so that the actors, responsible for execution of an event, can be noted. [16]

*Nesting/Grouping* of the activities is added in one of the later extensions. The nesting represents grouping a set of related events, such that the relation from an event to the nesting node is valid to all grouped events.

The notion of *spawned sub-processes* is one of the other extensions to the DCR graph. A DCR graph represents a business process, but each process can include sub-processes. Every sub-process can be either:

- single-instance sub process, meaning a process embedded in the main process, or

- multi-instance sub process, that can be spawned to run independently of the main process (4).

The graphical representation of the activities in the two tools - the visual tool and DCR workbench - and more details are available in Table 1 below.

Table 1 Graphical representation of activities

|  |  |  |  |
| --- | --- | --- | --- |
| DCR workbench | DCR.net visual tool | DCR workbench, code | Description |
| Activity; included | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Activity_itu.PNG | D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Activity.PNG | “Activity” | Activity can be executed only if it is included and the constraints like condition and milestone are fulfilled. When added the activity is included by default. In the visual tool, this state can set by check “included”. |
| Activity; excluded | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Activity_excl_itu.PNG | D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Activity_excluded.PNG | % “Activity” | After exclusion, an activity will be again available to be executed, when it is included. Exclusion of an activity affects the relation going out from that activity. If the activity is a condition for another, after exclusion it is no longer valid. The same is valid for milestone. |
| Activity; Pending | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Activity_pending_itu.PNG | D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Activity_pend.png | !”Activity” | An activity will occur in pending state after another activity has executed and there is response relation between them. The pending activity is expected at some point to be executed, but the execution is no longer required if it is excluded in meanwhile. |
| Activity; executed | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Activity_executed_itu.png | D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Activity_executed.png | :”Activity” | An activity could be executed several times except it has an exclusion relation to itself. It this case it will be excluded after the first execution. |
| Nesting/Grouping | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Group_itu.PNG | D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Nesting.PNG | Group “Group name” {  “Activity”} | Grouping in DCR Benchmark has the functionality of Nesting in the visual tool. A relation from the group / nesting is equal to putting relation to and from each of the compounding activities of the group. |

In Table 2 is shown the graphical representation of the relations between the activities and sub-processes.

Table Graphical representation of relations

|  |  |  |  |
| --- | --- | --- | --- |
| Relation; Condition | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Condition.PNG |  | -->\*  -[k]->\* | “A condition between two activities ensures that the second activity cannot be executed unless the first is excluded or has been executed at least once.” Timed version is also included. |
| Relation; Response | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Response.PNG |  | \*-->  \*-[k]-> | “A response, or goal, ensures that once the first activity has been executed the other activity becomes a goal, that must eventually be executed or excluded”. Timed version is also included. |
| Relation; Include | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Includes.PNG | D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Include_itu.PNG | -->+ | “The include relation includes other activities upon execution” |
| Relation; Exclude | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Exclude.PNG | D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Exclude_itu.PNG | -->% | “The exclude relation excludes other activities upon execution” |
| Relation; Milestone | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Milestone.PNG | D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Milestone_itu.PNG | --<> | “The milestone relations block the second activity if the first is currently a goal (response) and included.” |
| Relation; Spawn | | | |
| D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Spawn.PNG | D:\ITU_edu\2sem\Critical Systems Project\DCR\pictures\Spawn_itu.PNG | “Activity” {  } | “The spawn relation spawns a new sub-process” (4) |
|  |  |  |  |

# Our process – DCR model, DCR requirements - Modeling the GDPR requirements for data protection within Rejsekort A/S

In this section, we will model how GDPR requirements are expected to be applied in our case - ...

GDPR requires the controller to have lawful basis to process personal data. One of the lawful basis is the consent given by data subject. The consent should cover all the purposes of the processing. This requires Rejsekort A/S to specify and document the purposes of the processing of its customers’ personal data as well. As GPRD underlines, the personal data should be collected for explicit purposes and the consent should be explicit (7), it must require the consent for the services provided by Rejsekort A/S to be distinguished from the consent for marketing purposes. In the section 2 of (6) it is declared that opinion research institutes can have access to the following information about the customer: “name, address and e-mail address”. The given personal information is sufficient to identify natural persons and since the processing of the data for opinion research institutes is separate from the processing to provide services, we decide to split the consent in two different consents: consent for main purposes and consent for opinion research institutes.

Consent for main purposes – going through (6) we considered to put the following purposes in one group, called by us “main purposes” – manufacture the card, reload operations of the balance, send data to partner/affiliated companies. They form the core of the services, provided by Rejsekort A/S, and are closely related to each other.

Each customer using the Rejsekort system should get a traveling card. To produce a card, the manufacturer gets the name and possibly the photo of the customer, depending on the type of the card (6). On the other hand, the customer needs to reload his or her balance, using the reload automats on the train stations or the company’s website. All payment operations are carried out in the system of Nets A/S which receives the data from Rejsekort A/S to finish the operation of paying. In (6) is stated that “Employees in Rejsekort A/S and the affiliated transport companies, whose job it is to serve you as a customer and process your personal data, have access to the collected personal data.” From this statement it is clear that not only the employees of Rejsekort need access to personal data, but the affiliated companies could also get the personal data. We decide to put all the above activities in a group, called main purposes. Any other purposes, which are part of the core functionality of the Rejsekort A/S system, could be specified and added to this group. The DCR model of giving consent for main purposes is shown in Figure 1.

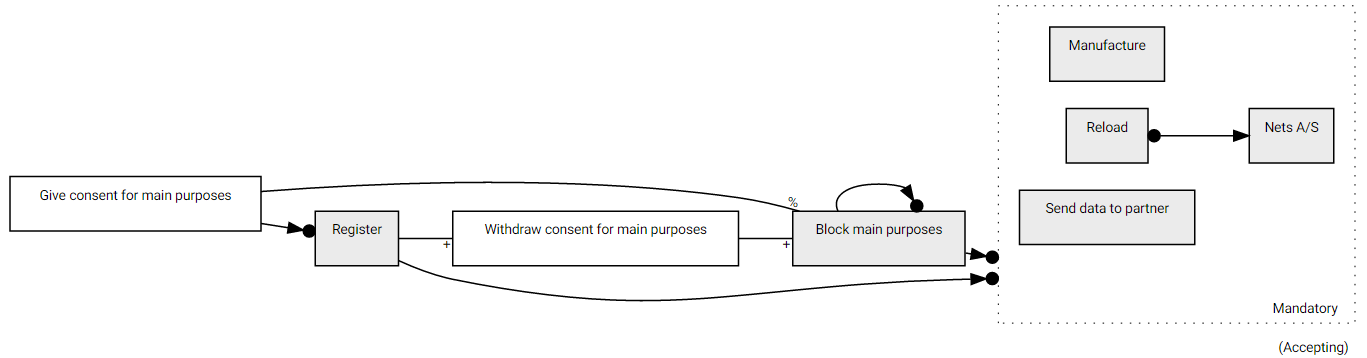


Figure DCR model of giving the consent for main purposes

In Figure 1 the following activities are illustrated – “Give consent for main purposes”, “Block main purposes”, “Register” and group item “Mandatory”. The activity “Give or for main purposes” models giving a consent for the main purposes by the data subject, which is a requirement before to start processing. “Register” represents the act of getting registered in the system. Here the customer fills in his/her personal data in the system of Rejsekort A/S, after which Rejsekort, as a controller, can begin processing with the given personal data. “Block main purposes” and “Withdraw consent for main purposes” are related to withdrawing a consent that will be discussed later in this section.

The group entry “Mandatory” encapsulates the main purposes mentioned above. “Mandatory” group includes the activities “Manufacture", “Send data to the partner” and “Reload” with response relation to “Nets A/S”. The activity “Manufacture” describes the production of the traveling card. “Reload” represents loading the traveling card with desired amount. Since all the payments in Rejsekort A/S is proceed through the system of Nets A/S, the event “Nets A/S” represents the usage of that system. “Send data to the partner” depicts sending the data to the affiliated companies.

In the beginning neither “Register”, nor activities in “Mandatory” can be executed, because they are blocked - “Register” is a condition for “Mandatory”, and “Give consent for main purposes” is a condition for “Register”. The graph in the figure above is in that initial state. The execution of “Give consent for main purposes” excludes “Block main purposes”, because of exclusion relation between them, and so “Block main purposes” is no longer a condition for “Mandatory”. There lefts the condition “Register”, where the customer fills in his personal data. After the execution of “Register”, the activities in “Mandatory” are allowed to be executed. The initial state of the graph above complies with the expectation that a customer should give his or her consent before to be able to “Register”.

(Rejsekort complies with this) GDPR, like the previous directive, guarantees the right of the data subjects to withdraw their consents (7). The withdrawing of a consent make it is impossible for the controller to use the personal data later for the purposes, for which they are collected. To illustrate this, we will use the same model in Figure 1. As mentioned above, the two activities “Withdraw consent for main purposes” and “Block main purposes” are part of the withdrawing of the consent. The activity “Withdraw consent for main purposes” illustrates withdrawing the consent given to process the personal data. “Block main purposes” doesn’t depict some activity of the data subject or the controller and has helper function. The “Register” activity has *include* relation with “Withdraw consent for main purposes”. The latter has the same relation with “Block main purposes”. “Block main purposes” excludes itself when the contest is given, and gets included (becomes again a condition for “Main purposes”) after withdrawing the consent. The initial state of “Withdraw consent for main purposes” is excluded – it cannot be executed before giving the consent. When “Withdraw consent for main purposes” is executed, it re-includes the event “Block main purposes”. Since the latter is a condition for “Mandatory” group, after the its re-inclusion, “Block main purposes” again becomes a condition for “Mandatory” group and the group gets blocked.

Fine The general data protection regulation expands the right of the data subjects to get help from the controller for exercising their rights. The Directive 95/46/EC doesn’t obligate the controllers to give effect to the rights of the data subjects, but in GDPR it becomes mandatory and the controller should comply with it (7). This is reflected in our model and can be seen in Figure 2.

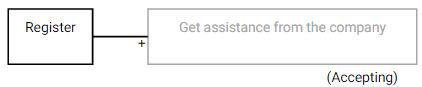


Figure DCR model of getting assistance from the controller

The Figure 2 includes two events – “Register”, which is explained before, and “Get assistance from the company”, which depicts the right of the customer to get assistance to exercise his/her rights. It is excluded by default and can be included when the customer is registered. After the inclusion “Get assistance from the company” can be executed.

Fine Article 13 of (7) requires the controller inform how long the personal data will be kept. Rejsekort A/S already complies with this rule and in (6) it discloses that keeps this information for five years after the termination of the customer relationship. This is modeled and illustrated in Figure 3.

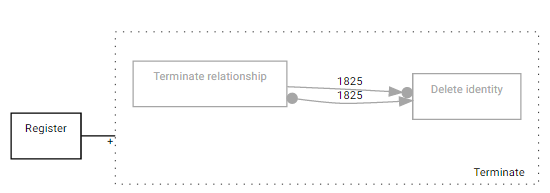


Figure DCR model of termination of relationship

In Figure 3 there are three activities – “Register”, “Terminate relationship” and “Delete identity”. “Terminate” and “Delete identity” are in a group, called “Terminate”. “Terminate relationship” represents stopping to be a customer of Rejsekort A/S. “Delete identity” describes the erasure of all the collected data. “Terminate” group is by default excluded, because it can’t be executed before “Register”. The activities “Terminate” and “Delete identity” are also excluded. The execution of “Register” re-includes the group “Terminate” that on the other hand leads to re-inclusion of both activities in the group, because a relation with a group is valid for all the activities in the group, as it is written before. The activity “Terminate relationship” has response and condition constraints to “Delete identity”. The execution of “Terminate relationship” will require a pending response within 1825 days. The number 1825 days is equal to five years. Since none of the tools allow specifying time units, here and in the other figures, where the constrains are timed, we will use days. Beside the response, there is also a condition constraint that requires “Delete identity” to be executed at least 1825 days after “Terminate relationship” is triggered. The combination of these two relations has the effect that “Delete identity” should be executed exactly after 1825 days - neither earlier, not later, as we expect. So as the model above also behaves, the personal data can’t be deleted before 5 years has passed after the termination of the relation.

(Malik; nothing to do with GDPR) Here we will show the blocking process, described in (6). A travel card can be blocked either by its owner, or by the staff of Rejsekort A/S. The owner can block it in case of lost to prevent further misuse with the card. The reasons of blocking the card by the staff is pointed in (?). One of the reasons could be that the customer has unpaid debts. In this case beside blocking the card, the customer also would be transferred to Customer Register. In both of blockings the card becomes no longer valid and can’t be used. We model both of the cases with one DCR model that is shown in Figure 4.

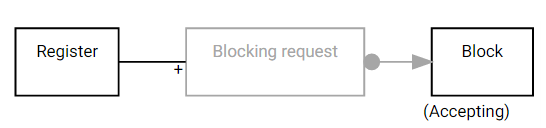


Figure DCR model of blocking request

In Figure 4 there are 3 activities – “Register”, “Blocking request” and “Block”. “Register” is the same activity from the previous graphs. “Blocking request” depicts the request of the owner for blocking, and “Block” - the process of blocking, which could be executed directly by the employees of the controller because of an unpaid debt or as a result of the request of the customer. “Blocking request” is excluded by default and can’t execute before “Register”. There is an *include* relation between “Register” and “Blocking request”, which means that when “Register” is executed “Blocking request” will be re-included and will be available to be triggered. The relation between “Blocking request” and “Block” is of type response. This means that after the execution of “Blocking request”, “Block” becomes a pending response. The deadline is not specified, so it could be triggered at any time. When the card is blocked the user shouldn’t be able to use the system of Rejsekort A/S. This would mean that some of the activities are no longer valid e.g. “Reload” (“Reload” is no longer valid, since the card is blocked). But we decide not to extend the graph with more constraints to keep it simple.

# Reflection – the plusses of DCR, the lacks

Pluses – add a new constraint (no need for changes); loops (example); the graph

Lacks – resets the tick for each sub-process, timing – days, hours, years

When an imperative approach is used for modeling a business process, we should define all the legal steps to end in another legal step. At each step we specify which are the next possible ones. A deviation from the predefined path from a step to another step is not allowed. "However, while imperative approaches are a strong concept when it comes to well-defined processes, they lack clarity once an observed behavior allows for flexible execution." (9)

In the declarative approach, on the other hand, each transition from a state to another state is allowed by default, except there is no violation of any of the constraints. Instead of defining the allowed sequences of events, the workflow is described using rules between the events (10). The flexibility of the declarative way, like DCR, comes from the fact that, when a new constrain between the events arises, it is enough to define it in the graph without explicitly rearranging the flow in the diagram.

(Malik; with DCR we have smaller graph … ???)

In a DCR graph there is no need to put a loop, if some actions would be executed in a cyclic order. The approach here in DCR is to put the appropriate constrains in a way that they allow cyclic execution.

The graph in Figur 1 above shows that if the consent for mandatory is not given, the activities in the group “Mandatory” are blocked. This is the initial state of the graph. When the consent is given, the events in the group become allowed - the graph is shown in Figure 5.

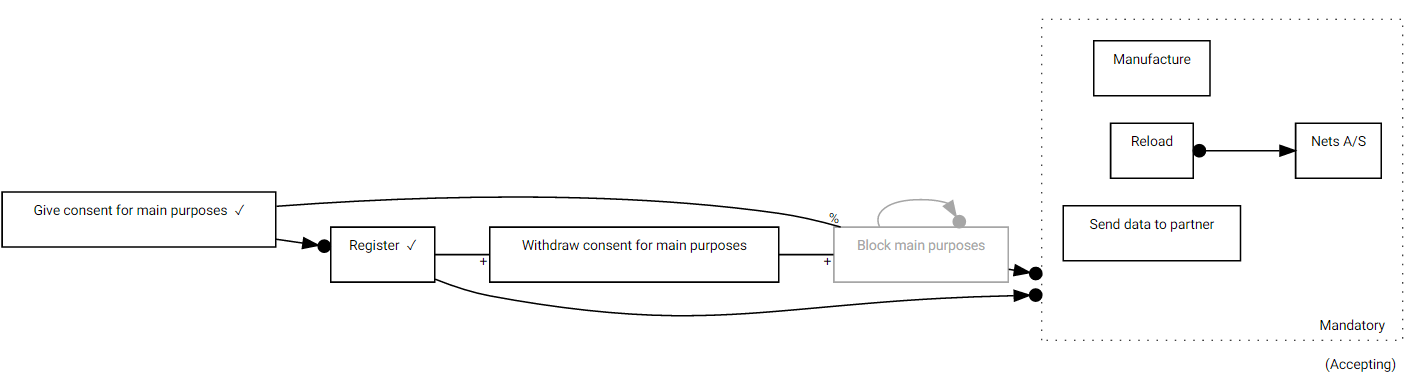


Figure DCR model of giving consent for main purposes. The activities in group Mandatory are available

And now if we withdraw our consent, the activities in the group would be blocked and the graph would look like this:

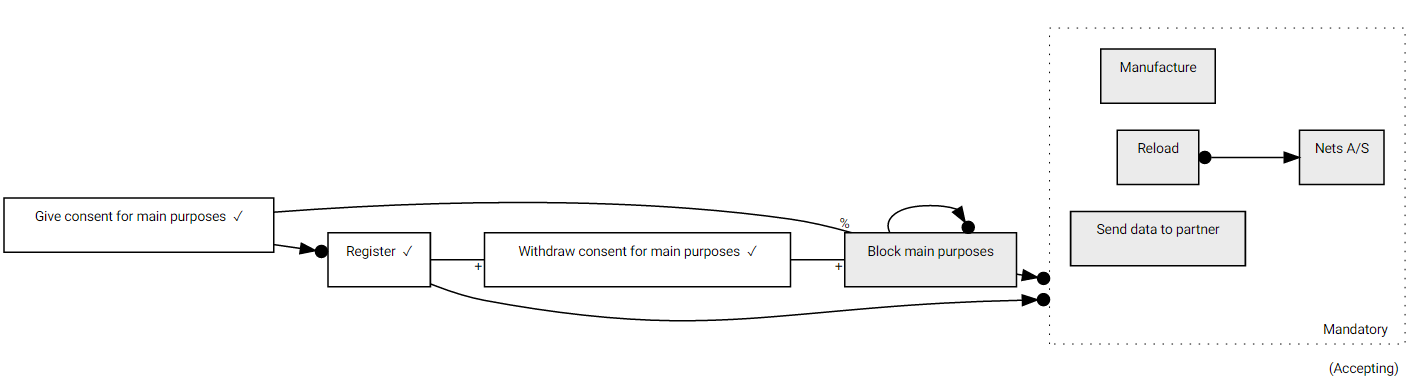


Figure DCR model of giving consent for main purposes. The activities in group Mandatory are blocked

In figure Figure 6 we can see that the activities in “Mandatory” are again not allowed because of the existing condition constraint with “Mandatory” group. If we decide to give the consent again, then the only thing to be done is to execute the activity “Give consent for main purposes”. There is no need to define any loops and we don’t change the flow explicitly. The only thing is to add constraints.

We encountered a problem with the sub-process activities using DCR Workbench. When a new sub-process is spawn after an existing one, the deadlines of the pending responses in the existing sub-process are reset to their initial values, which is not correct. We will demonstrate this with an example to the sub-process “Data breach” from our graph. The Figure 7 shows the sub-process item “Data breach” with two spawned sub-processes.

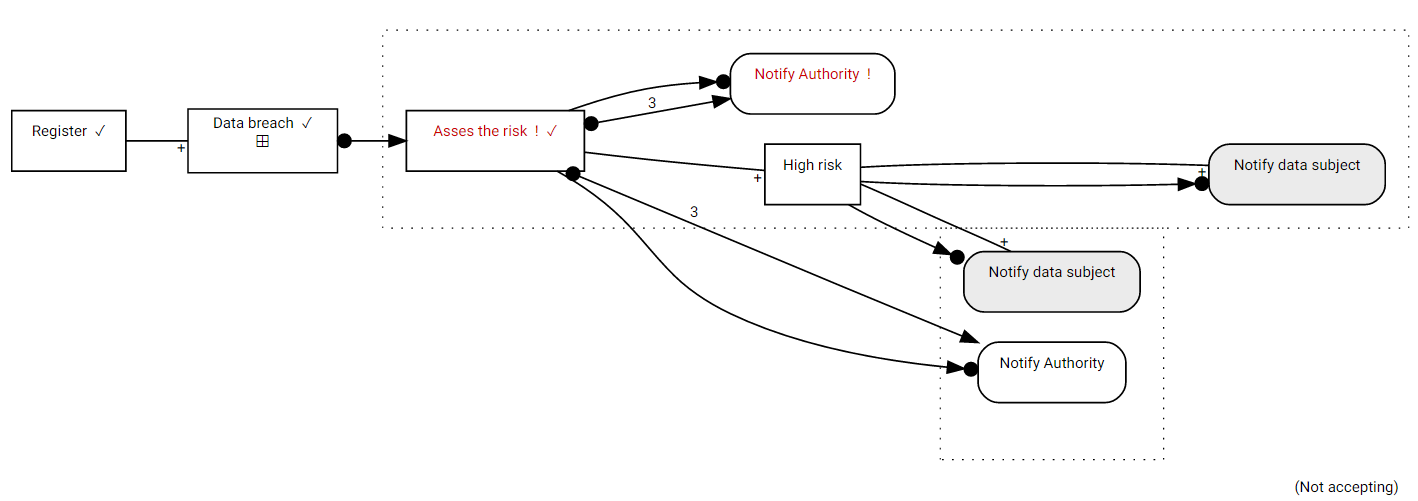


Figure DCR model of Data breach entry. Two sub-process are spawned

As shown in Figure 7 , the activities “Asses the risk” and “High risk” is shared visually between the two sub-processes. During the simulation, the execution of “Asses the risk” for the second sub-process leads to resetting the deadline of “Notify authority” for the first sub-process. For example, if the deadline of the first process is 1, after the execution of the activity “Notify authority” for the second process, both deadlines are set to 3. This is not a correct behavior.

An additional helper event is added to fix this error. The new graph is shown in Figure 8. The event added is “StartNotification” that is put between the activities “Asses the risk” and “Notify Authority”. “StartNotification” has *exclude* relations to itself. After the execution of “StartNotification” it will exclude itself and will be no longer a condition for “Notify authority”, but it will require a pending response with deadline 3 days. When “Asses the risk” is executed for the second or the other subsequent activity, it will not reset the deadlines of the previous activities, where “StartNotification” has already executed. “Asses the risk” will not affect the deadline of “Notify authority”, because “StartNotification” “cuts” the relations between these two events.

To fix this error we recommend not to share the common activities between the sub-processes. Thus in our graph “Asses the risk” will not be shared and such will not reset the deadlines that are already advanced.

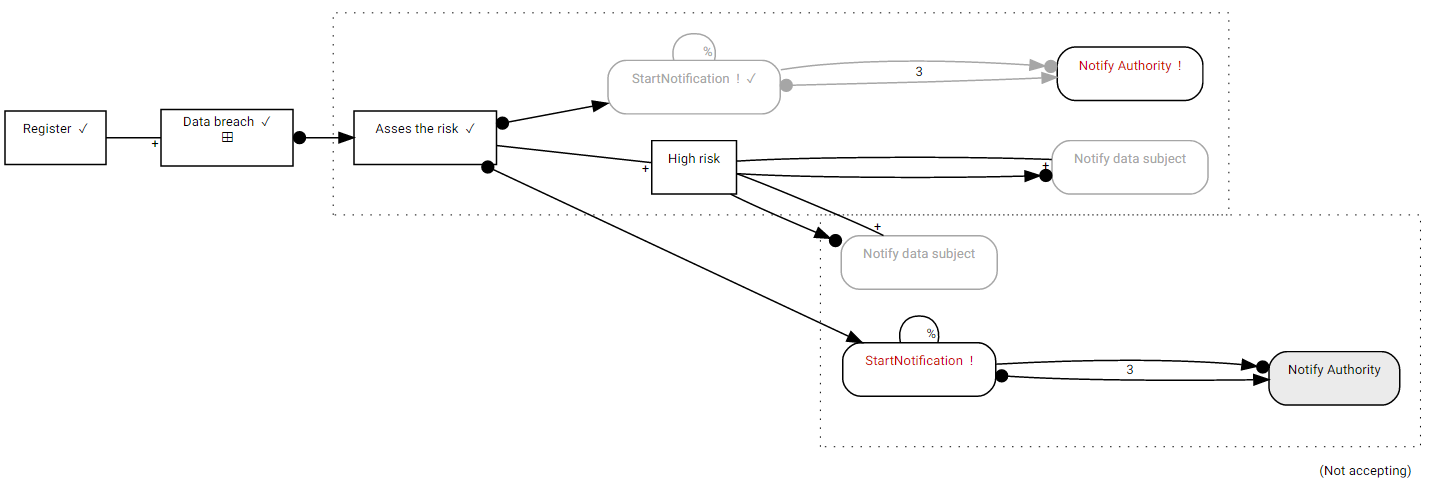


Figure DCR model of the entity Data breach. Two sub-processes are spawned with StartNotification added in the figure

# Conclusion

======

References:

1. T. Hildebrandt, and R. R. Mukkamala. „Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs”, IT University of Copenhagen, year ?
2. G. D. Giacomo, M. Dumas, F. M. Maggi and M. Montali. "Declarative Process Modeling in BPMN", Sapienza Universita di Roma, Italy, University of Tartu, Estonia, Free University of Bozen-Bolzano, Italy
3. http://wiki.dcrgraphs.net/, [Accessed: 11-Dec-2016]
4. In the nick of time …
5. Rejsekort A/S, PROTECTION OF PERSONAL DATA – REJSEKORT A/S, Søborg, 2015
6. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016
7. http://www.whitecase.com/publications/article/chapter-1-introduction-unlocking-eu-general-data-protection-regulation, [Accessed: 12-Dec-2016]
8. J. Prescher, C. D. Ciccio, and J. Mendling, "From Declarative Processes to Imperative Models", Vienna University of Economics and Business, Vienna, Austria, year
9. Pesic, M., Schonenberg, M.H., Sidorova, N., van der Aalst, W.M.P.: Constraint-based workflow models: Change made easy. In: OTM Conferences (1). pp. 77–94 (2007)