Harbour Mini Workflow Engine

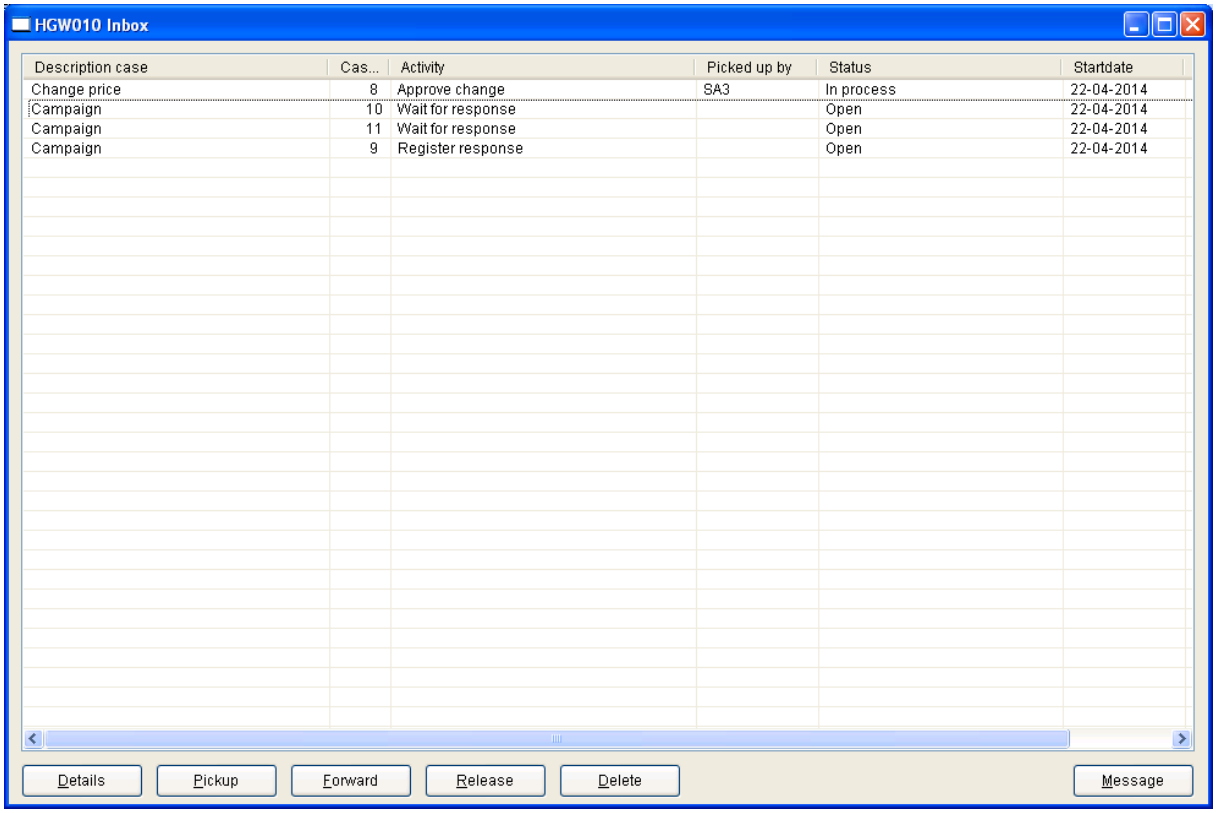
1. Introduction

1.1 What is a workflow engine?

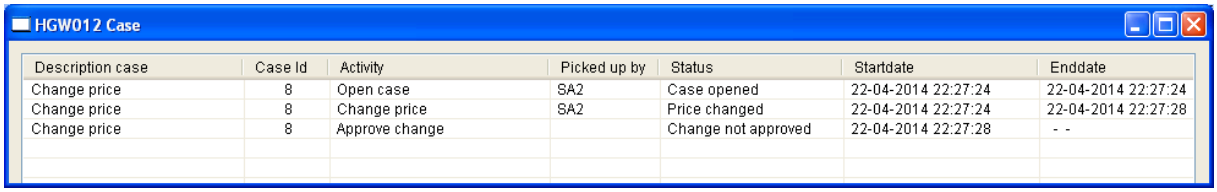
Within one business process different business units and different application systems can be involved. A workflow engine leads an instance of such a business process through the business process.
For each event that triggers an instance of the business process the workflow system creates a *case*. A case has a worklist of activities. An activity can be performed manually or automatically.
For an automated activity the workflow engine starts the linked use case. When the use case has been completed, the workflow system updates the status of the activity and determines which follow up activity has to be started next.
For a manual activity a team member has to pick up the workitem from their worklist, shown in the inbox. Also a manual activity is performed by the linked use case.
For a fully manual activity only the status must be updated when the activity has been completed. For this purpose the predefined use case Change status must be linked.

1.1 The inbox

The inbox shows the current open activities for a specific team (i.e. usergroup), it shows a worklist with workitems. A team member can pick up and process a workitem (or activity). When the workitem has been processed, the workflow engine updates the status of the activity and decides what to do next. Possibly the case is forwarded directly to the team or teams that has to perform the follow up activities. Another possibility is that the case must be forwarded manually by the team member who picked up the workitem.



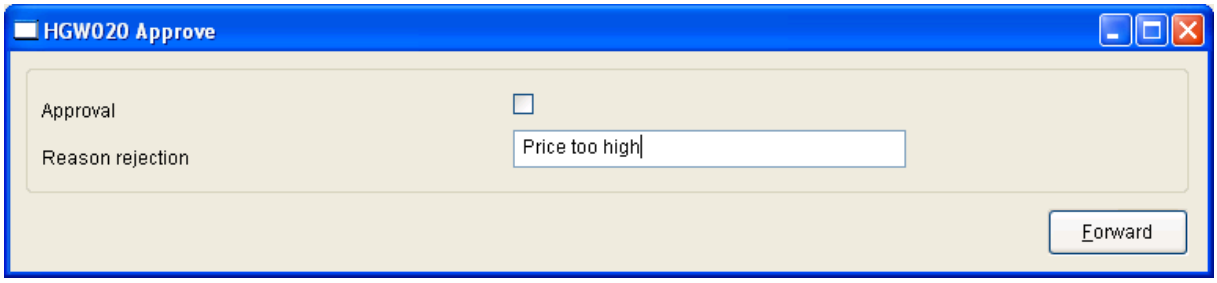
Clicking on Details gives an overview of the performed activities so far.



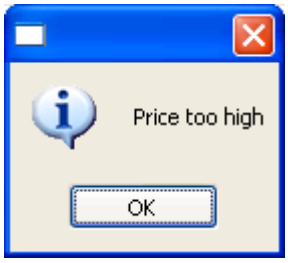
With the Release button the picking up of a workitem can be made undone, so another team member can pick up that item.

When a workitem is deleted, the case is closed. When you use this button intentionally, beware the case has been closed and you can't rely on the workflow system anymore.

When the 4 eyes principle applies for an activity, that activity has to be approved by a colleague. For the approval the workflow systems shows the following form:



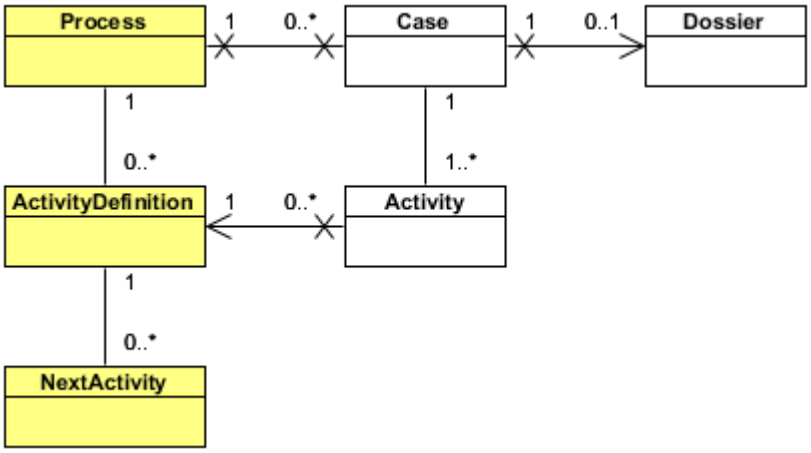
The approver has to give in a message when he doesn't check the approval checkbox.
Then the case is send back to the one who initiated the approval activity and that person can retrieve the message by selecting the workitem and clicking the Message button.



The user who initiated the approval activity can decide to pick up the workitem and make changes, or double check his work and forward the case without making changes.

1.3 Structure workflow engine

The following figure depicts the objectmodel of the workflow system.



On the left side, in yellow, we talk about *definitions* on meta level. Hmggen is used to define these definitions. So a process is a process definition, the definition (or specification) of a certain process, The class Case is a definition (or specification) of an instance of a process. A case object is a real case, with one or more real activities.
A case encapsulates a dossier with specific transactional data in xml format.

Note that an association without arrows is bidirectional.

2. Setting up a business process

In this chapter we set up a business process in hmggen named Process order. Since workflow support starts in the back end, the focus is on *full* back end processing of an order, without a web front end.

Process Order (backoffice)

Actor:
Sales representative

Trigger:
Incoming telephone call

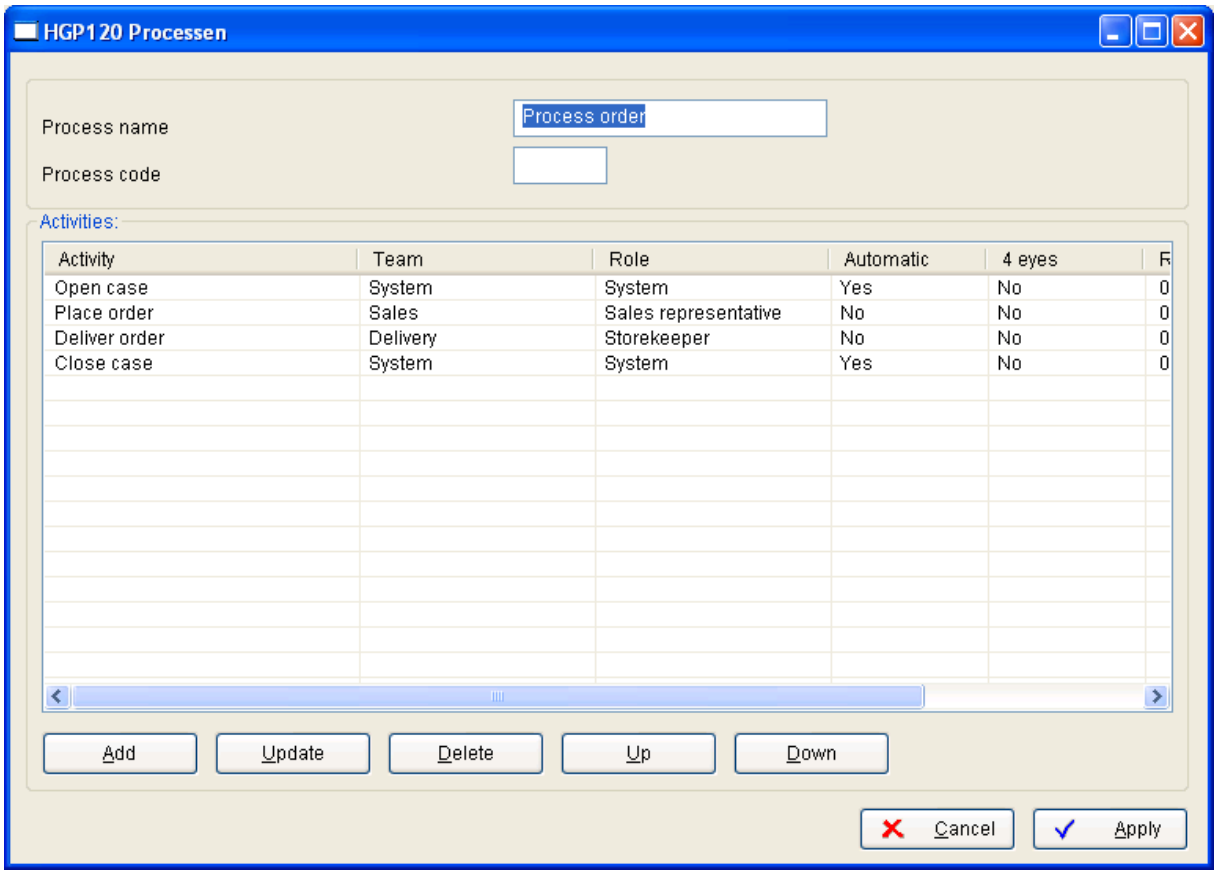
Basic flow:
1. Sales rep. selects customer
2. Sales rep. places a new order
3. Sales rep. passes order to Delivery
4. Storekeeper prepares delivery

5. Storekeeper delivers order on delivery date (customer pays at delivery)

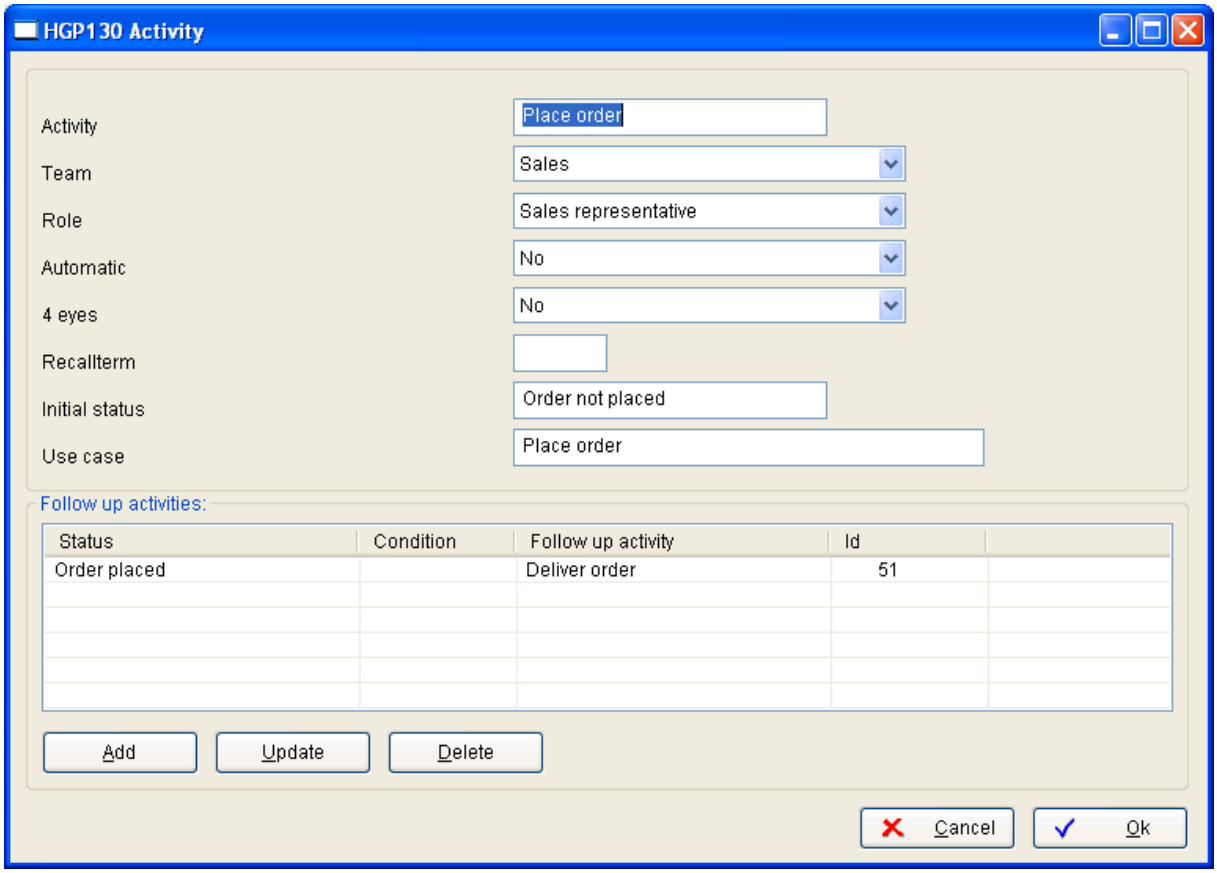
Alternative flows:
New customer
1a. When the customer has not been found
1a1. Sales rep. registers new customer
1a2. Use case resumes with step 2 of the basic flow

In this process two business units are involved, Sales and Delivery. An assumption is that both business units have their own automated systems.
The workflow system handles the communication between the business units, so the sales rep. doesn't have to pass an order to the business unit Delivery.

Each workflow process has two standard activities, Open case and Close case. When the sales rep. has placed an order, the workflow system creates a case, i.e. an instance of the process Process order. Using hmggen you can define activities and conditions under which the activities have to take place. An activity can be an automated activity or a manual activity. An automated activity is started by the workflow system self, a manual activity is started by picking up a workitem from the inbox.
For Process order the following activities haven been defined:



When you select the activity Place order, via Update, hmggen shows the properties of the activity.



An activity must be assigned to a team. This implies you have to set up the organization and the teams (usergroups) first.
An activity must be assigned to a role. Only team users with the requested role can pick up and perform this activity.

The activity can be performed manually or automatically. When you chose “automatic”, the specified use case is started by the workflow system at runtime, otherwise by the user.

You can apply the 4 eyes principle by setting the 4 eyes option to “Yes”. If you do so, a colleague has to approve the outcome of he activity first, before the workflow process resumes.

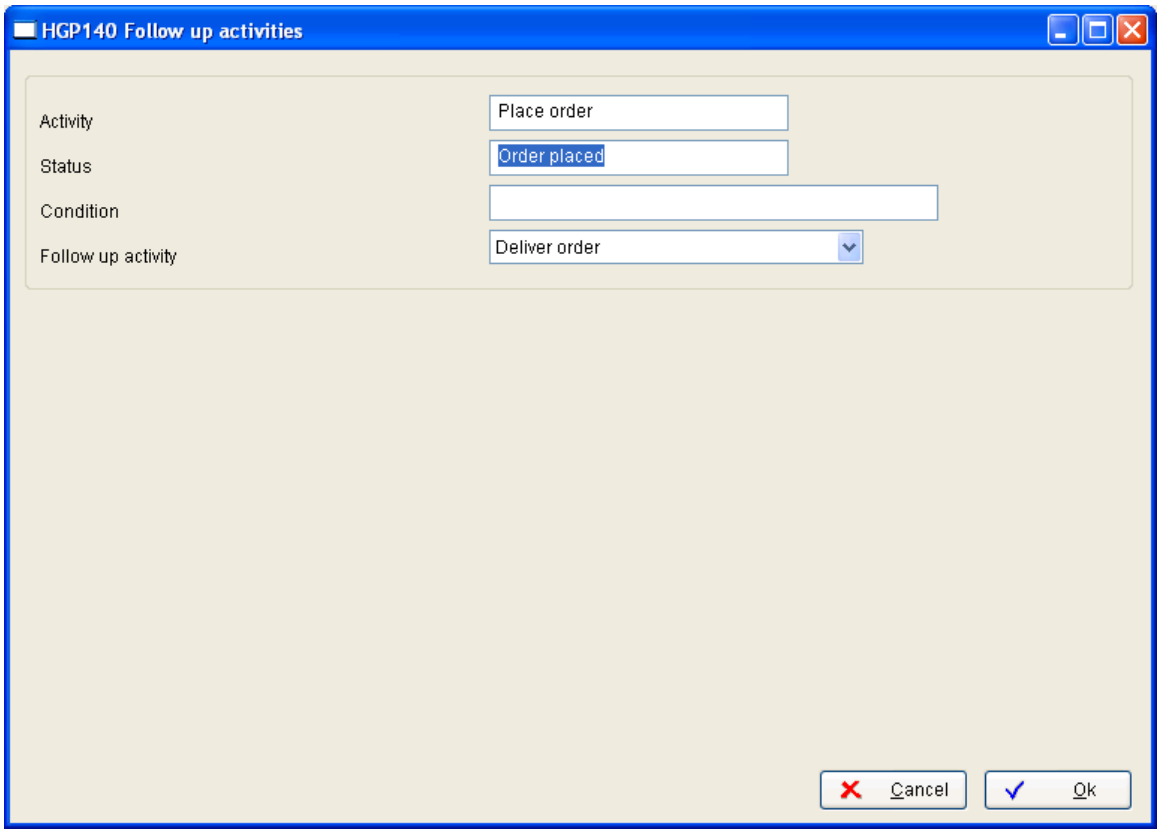
When the workflow system has to take action when the response period has expired, you can set the recallItem (in days).

Each activity needs an initial status. When an activity must be performed manually, the workflow system shows a choicelist with the current status, this initial status, plus the statuses under which follow up activities take place.

An activity delegates the work to be done to a use case. The name of a use case consists of a verb plus a noun, e.g. Place order, Adjust order or Select customer.
At runtime the workflow system composes the name of the function to be called from “On” plus the use case name, Place order becomes OnPlaceOrder().

When the activity must be performed manually completely, the linked use case must be the predefined use case Change status. This use case shows a popup with a list of statuses.

When you select the follow up activity Deliver order, hmggen shows the status and condition under which the follow up activity will be started.



In this example the follow up activity Deliver order is started when the status of the current activity has been set to Order placed.
The condition is optionally, you might specify an extra condition when needed. Such a condition may contain harbour functions or user defined function, they will be interpreted at runtime.

In fact you always have to define an activity, before hmggen enables you to select that activity as follow up activity. So it is recommended to define all activities first, before defining follow up activities.

2. Setting up use cases

A use case can be specified on different levels. A business use case is a high level use case, a use case is a *system* level use case. A business use case can be considered as the IT description of a business *process*.
It's always a challenge to specify uses case steps on the right level. If a use case has too many steps, ask "why questions" to reduce the number of steps. Remember each step must contribute to the goal of the use case. If a use case has only a few steps, ask "how questions", so you find more detailed steps.

For prototyping start with so called use case outlines. The examples in this chapter are outlines.
In each iteration you can add more detail and more "rainy day" scenario's to your use cases.
When a use case has been realized (i.e. build), you might replace storyboards by real screenshots.

Use case Place order

Trigger:
Incoming telephone call

Actor:
Sales representative

Basic flow:
1. The actor selects a customer
2. The system enables the actor to place an order
3. The actor enters order and orderline data
4. The system validates the data
5. The system stores the data

Alternative flows:
Invalid deliverydate
4a. When the actor enters a invalid delivery date
4a1. The system shows the message "Deliverydate must be at least one day ahead"
4a2. The use case resumes with step 3 of the basic flow

Order data:
ordernumber, customer, deliverydate

Orderline data:
order, product, quantity

Usability requirements:
- The system must enable the actor to select a department and depict a product from a list

Refer to the appendix for more use case examples.

2.8 Tracking and tracing

You might measure the throughput of transactions by querying the database.
Each activity has a start date/time and a end date/time, and each activity has an initiator and a performer.

Also the logtable can be examined to determine the number of updates on a certain object in a certain period.

APPENDIX

Business Use Case Change price
Actor:
Sr. Sales representative

Trigger:
Incoming prices change

Basic flow:
1. Sr. Sales rep. selects a product
2. Sr. Sales rep. changes the price
3. Sr. Sales rep. forwards workitem to colleague
4. Colleague approves the price change
5. The system effectuates the price change on the startdate

Alternative flows:
Incorrect price change
4a. When the price change is incorrect
4a1. Colleague sends workitem back to sr. Sales rep. who posted the price change
4a2. Sr. Sales rep. adjusts price change
4a3. Use case resumes with step 3 of the basic flow

Use case Change price
Actor:
Sr. Sales representative

Trigger:
Needed prices change

Basic flow:
1. Actor selects a product
2. Actor changes the price and selects the startdate
3. Workflow system makes a new case
4. Workflow system forwards workitem to colleague for approval

Use case Approve price change
Actor:
Sr. Sales representative (colleague)

Trigger:
Workitem in inbox

Basic flow:
1. Actor selects workitem
2. Actor checks price change
3. Actor approves price change
4. Workflow system creates follow up activity to effectuate prices change on startdate

Alternative flows:
Incorrect price change
2a. When price change incorrect
2a1. Actor enters reason rejection
2a2. Workflow system sends workitem back to sr. Sales rep. who posted the price change

Use case Adjust price change
Actor:
Sr. Sales representative

Trigger:

Rejected workitem in inbox

Basic flow:
1. Actor selects workitem
2. Actor checks message (reason rejection)
3. Actor adjusts price change when needed
4. Workflow system forwards workitem to colleague for approval

Use case Effectuate price change
Actor:
System

Trigger:
Timer

Frequency:
Daily

Basic flow:
1. System selects open workitems for price changes with expired startdate
2. For each selected workitem system effectuates price change and closes the case

Business Use Case Running a campaign
Actor:
Sales representative

Trigger:
Marketing plan

Basic flow:
1. Sales rep. starts campaign
2. System sends out a personalized letter to each customer
3. Customers send back response form
4. Sales rep. measures response

Alternative flows:
No response
3a. When a customer doesn't respond
3a1. System sends out a recall letter to that customer
3a2. use case resumes

Use case Start campaign
Actor:
Sales representative

Trigger:
Marketing plan

Basic flow:
1. Sales rep. starts campaign
2. System sends out a personalized letter to each customer

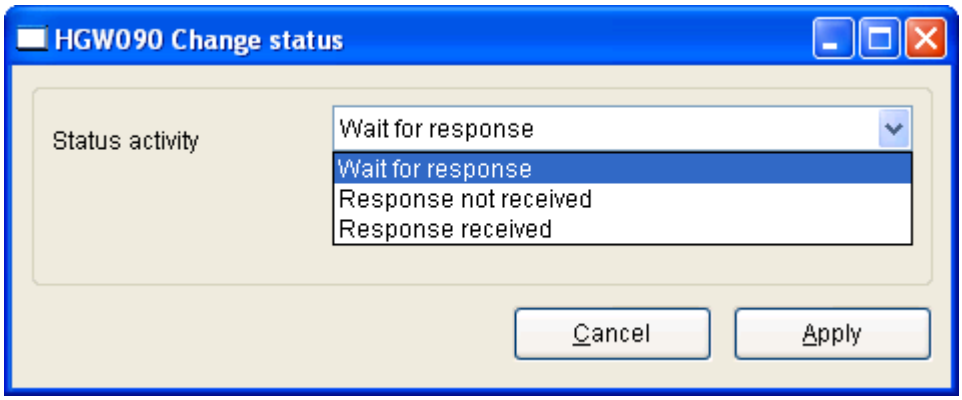
Use case Register reception response
Actor:
Sales representative

Trigger:
Customer has send in response form

Basic flow:
1. Sales rep. selects workitem from inbox
2. Sales rep. updates state activity to “Response received”
3. Workflow system creates workitem to register response

Alternative flows:
Response not received within recall term and max recall terms not exceeded
1a. When response form not received within recall term and max recall terms not exceeded
1a1. System sends out recall letter to customer
1a2. System prolongates recall term
1a2. Use case resumes

Response not received within recall term and max recall terms exceeded
1b. When response form not received within recall term and max recall terms exceeded
1b1. Workflow system updates status to “Response not received” and closes case



Use case Register response
Actor:
Sales representative

Trigger:
Workitem in inbox

Basic flow:
1. Sales rep. selects workitem from inbox
2. System updates state activity to “Response registered”
3. Workflow system closes case