## Contents

AD HOC PATTERN PROCESS - CONSTRUCTION ............................................. 1

1. Ad Hoc Process ......................................................................................... 3

2. PROCESS MODEL ..................................................................................... 4

3. DATA MODEL ............................................................................................ 6

4. DEFINE FORMS ........................................................................................ 7
   4.1 Create Task Event Forms ..................................................................... 7
   4.2 Execute Task (Perform Ad Hoc Task Sub Process) ............................... 8
   4.3 Global Form ......................................................................................... 9

5. BUSINESS RULES .................................................................................... 11
   5.1 Activity Actions .................................................................................... 11
1. Ad Hoc Process

It is fairly common that businesses need to manage unstructured processes (ad hoc) rather than well predefined business processes.

Ad hoc processes consist of a series of activities which cannot be predefined. In such processes, users must be able to decide what to do and when to do it, and also they must be able to assign work (activities) to other people, creating interactions among various users.

To implement Ad hoc processes Bizagi proposes a workflow pattern that allows creating tasks at any time that can be performed in any order during the life of a process. These tasks can be assigned to anyone and be scheduled to be resolved anytime. Users who have been allocated tasks, can in turn create additional tasks for themselves or other process participants.

The Ad hoc process pattern uses an event that is available for all users at any time in the process to create and assign tasks for themselves or somebody else.
2. PROCESS MODEL

The following is a process that seeks to illustrate the use of Ad Hoc Pattern.

Imagine a Committee meeting. At the end of the meeting there is always a series of tasks left to be carried out. Since there is no way to pre-define the outcome of a meeting we will use Bizagi’s ad hoc process pattern. In it, any type of task can be assigned to the appropriate people.

The first activity will record the outcome of the meeting. When the meeting is finished, an event called **Create and review tasks**, is available at any time to allow anyone to schedule tasks and keep track of the tasks previously created.

In the event users include a description that defines what has to be done, an assignee and a due date.

Then, an activity is created for the assignee to carry out the task, through the **Perform Ad Hoc Task** sub process. The due date is set as the date to finish the scheduled activity.
3. DATA MODEL

The data model for our ad hoc pattern process is very simple.

The process entity is Committee Meeting where the information of the Meeting is recorded. The entity is related to the Task Scheduled entity, the relationship is one to many. The Task Scheduled entity contains the history and status of all tasks created.

In the Create and Review Tasks event there are three temporary attributes. These temporarily contain the information of the new task to be created which is then recorded in the Task Scheduled entity where it is recorded permanently.
4. DEFINE FORMS

4.1 Create Ad Hoc Task Event Forms

In the Create Ad Hoc Task Event, users include the information of the new Task that will be created. It is necessary to include some information about the task, for example, the descriptions, due date and the person responsible for it.
4.2 Execute Task (Perform Ad Hoc Task Sub Process)

The activity of the sub process “Perform Ad Hoc Task” will be assigned to the person chosen during the Committee Meeting or Create Ad Hoc Task. It will be scheduled to be performed on the Due Date.

The Form for this activity will contain a field to record any observations and the status of the task. For this example we have included two status options: Pending and Completed. This information will be updated automatically in the Create Ad Hoc Task event described above, as well as the Global form of the process.

There will be as many Perform Ad Hoc Task sub processes as required by the users. As soon as a new task is recorded, a new Perform Ad Hoc Task sub process is activated. Perform Ad Hoc Task sub processes are completely independent from each other.
4.3 Global Form

In order to control the status of all tasks, we recommend creating a Global form for the process so that when a case is searched, the global form will display the updated information.

This is how your process will look when you search for it on the Work Portal

If you would like to highlight the status of any task or note whether it is on time or overdue, change the Behavior of the attribute in the Form designer.
When CommitteeMeeting.TaskScheduled-Status is different to 2

Then set forecolor for CommitteeMeeting.TaskScheduled-Status to Red

Else perform reverse actions
5. BUSINESS RULES

5.1 Activity Actions

An expression is required to record the information from the temporary attributes in the Event to the Task Scheduled entity, where the history of tasks is stored permanently.

In the expression the three temporary attributes in the event are stored in a new record of the **Task Scheduled** entity. Additionally when created they will have the 'pending' status set with Code ‘1’ in the Parameter entity.
AD HOC PATTERN PROCESS-CONSTRUCTION

www.bizagi.com