

White Paper

BPMN 2.0 Task Types Explained

WP0093 | August 2013



Gregor Polančič

Gregor is an assistant professor at the University of Maribor and has a decade of experience in BPMN since its first version in 2004.

He participated in the development of one of the first BPMN modeling utilities - a package of plugins for Visio, which were introduced in 2004 and is the main author of the first BPMN poster.

In 2008, he was one of the first authors who published an article dedicated to the experiences and practical use of BPMN. The article was published in "BPM and Workflow Handbook" in association with the Workflow Management Coalition (WfMC).

He is currently researching BPMN from different technological and user aspects.

Tasks represent the most fundamental process elements, which define units of work in a process. In BPMN, a Task represents an atomic Activity that is included within a Process. So, a Task is used when the work in the Process is not broken down to a finer level of process model detail.

In cases where an Activity can be further decomposed, Sub-Processes should be used. A Task object shares the same shape as the Sub-Process, which is a rectangle that has rounded corners. Both the Sub-Process and Task share the same shape outline but a Sub-Process is differentiated with a '+' symbol, illustrating it can be expanded (Figure 1).

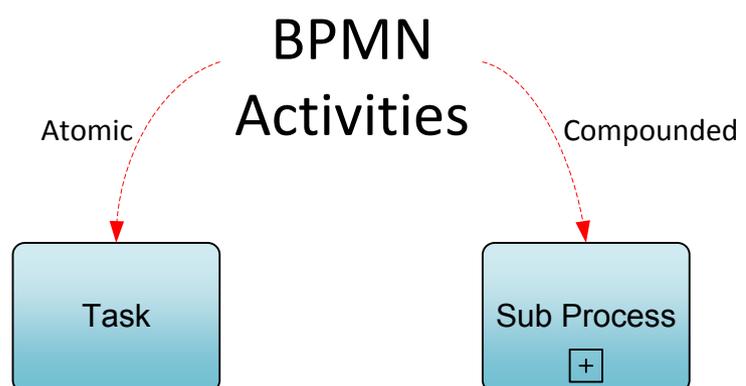


Figure 1: atomic (left) and compounded (right) BPMN Activity

Access our **free**, extensive library at
www.orbussoftware.com/community

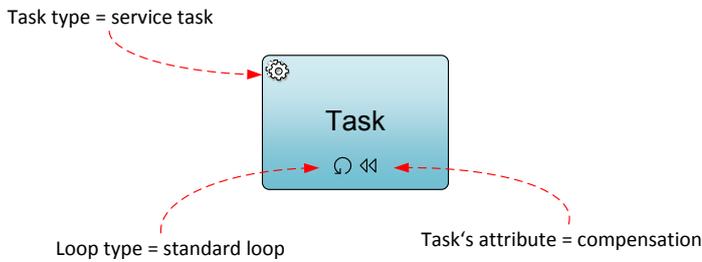


Figure 2: Task types specializations

BPMN 2.0 [1] defines several types of Tasks, which are differentiated by their attributes, behaviours and performers. These Task's specifics are graphically annotated with symbols within the Task element (Figure 2).

The top left corner of a Task element can include several different symbols which represent the Task type or the way a Task is performed. Generally, an end-user, an application, or both will perform the Task. The most common Task is an Abstract Task, which represents a Task with no specialized behaviour.

The bottom side of a Task element can include symbols which represent Task's looping behaviour (normal looping, parallel multiple instance, and sequential multiple instance) as well as the graphical representation of a Compensation Task. A Task can also have a thick border, which would indicate a (calling) Global Task.

BPMN 2.0 Task Types

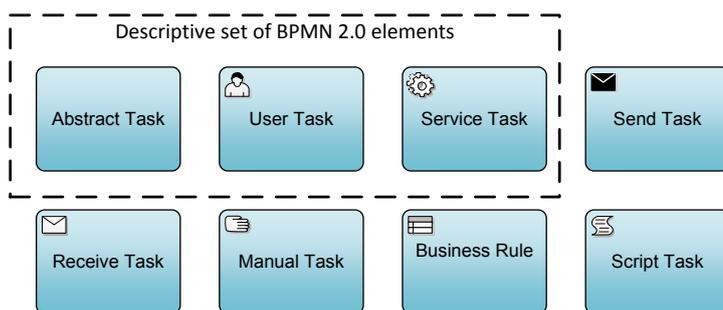


Figure 3: BPMN 2.0 Task Types

Different types of BPMN Tasks separate the inherent behaviors that a Task might possess. The list of Task types may also be extended along with any corresponding indicators. As already mentioned, a Task which is not further specified is called an Abstract Task.

As evident from Figure 3, the descriptive set of BPMN 2.0 elements supports only three Tasks types: Abstract Task, User Task and Service

Task. The remaining Task types are defined in the analytical set of BPMN 2.0 elements. The next sub-sections, will discuss the specific Task types along with examples of their use.

User Task

A User Task is commonly used, since it represents a typical “workflow” Task. A User Task is executed by a person, with the assistance of a “process-aware” application (e.g. a Business Process Management System – BPMS). This means that the lifecycle of a User Task is managed by a software component (a Task Manager) and is typically executed in the context of a Process. User Tasks are assumed to be performed via an application's user interface, as in a BPMS. A User Task symbol must include a human figure marker on the top left side to

distinguish it from other Task types (Figure 4). Examples of User Tasks are the following:

- A customer fulfils an online registration form,
- A help-desk employee marks an open issue as resolved,
- An editor approves an article for publishing.

Note that the above examples of User Tasks should be performed within the context of a “workflow system” or “process-aware” application. In the opposite case, if a Task is supported by a standalone IT system (in a “non-BPMS automated manner”), it shouldn’t be typified as a User Task. In this case a Manual or Abstract Task should be used.

Manual Task

A Manual Task is another type of BPMN Task which requires human involvement to complete. In contrast to a User Task, a Manual Task is a Task that is expected to be performed without the aid of a business process execution engine or application (Figure 4). A Manual Task provides no system interface to the performer, so it is inherently not executable in a BPMS. Examples of Manual Tasks are as follows:

- A telephone technician installs a telephone at a customer location,
- A help desk issue is resolved via telephone call,
- A software engineer implements an open issue.

As evident from Figure 4, the User Task is performed with the assistance of a “process-aware” application, where the Manual Task is performed without the aid of a “process-aware” application.

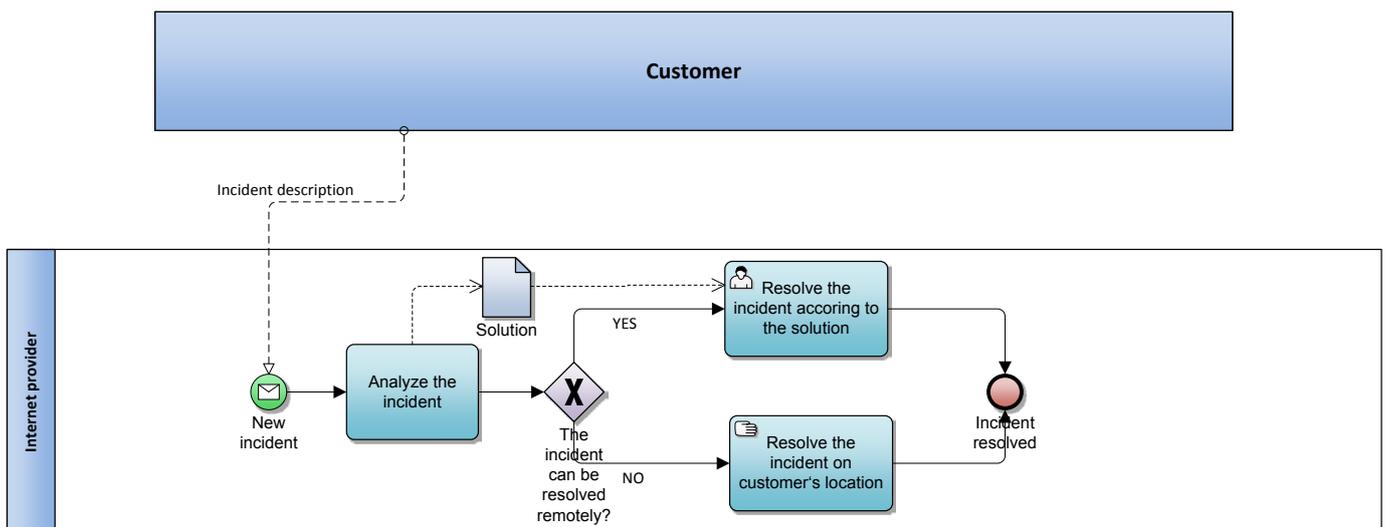


Figure 4: Human interaction Tasks (user Task and manual Task) used in a simplified help-desk process

Service Task

As opposed to the User Task and Manual Task, a Service Task does not require any human interaction. It is completed automatically, by using some sort of an external service (outside the corresponding “process execution environment”), which could be a Web service or an automated application (Figure 5). Examples of a Service Task are as follows:

- A payment processed by PayPal services,
- Storing an image in an online archive,
- Converting a price into a specific currency using an online currency converter.

There is a difference when “Manual Tasks” (e.g. a User Task) and “Automated Tasks” (e.g. a Service Task) are performed. A Service Task is performed immediately - as soon as the sequence flow into it arrives, and the service provider is assumed to begin performing the business function immediately upon invocation. When the business function provided by the service is done, the Service Task is complete and the process continues.

A User Task is assumed to wait for some time in the assigned performer’s work list before task performance begins. Technically, when a process instance reaches a User Task, an entry (work item) is created immediately in the performer’s work list, and the task becomes immediately available to be performed. This means that an activity has already started, even though the corresponding performer has not begun to work on it yet.

Script Task

Another type of automated Task is the Script Task (Figure 5). In contrast to the Service Task, a Script Task is executed by a business process engine. Technically, a Script Task represents code, which can be executed on a process engine. The modeler or implementer defines the Task’s script in a language that the engine can interpret. When the Task is ready to start, the engine will execute the script. When the script is completed, the Task will also be completed. Examples of a Script Task are as follows:

- Receive a new open help-desk issue and send to the operator,
- Calculate the total cost by summing product cost and shipping cost,
- Add a unique ID to a help-desk ticket.

According to Business Process Management Systems’ philosophy, the majority of process Tasks, should be performed in a decentralized way by using existing or external systems (Service Tasks) where new implementations (Script Tasks) should only “glue” together existing functions.

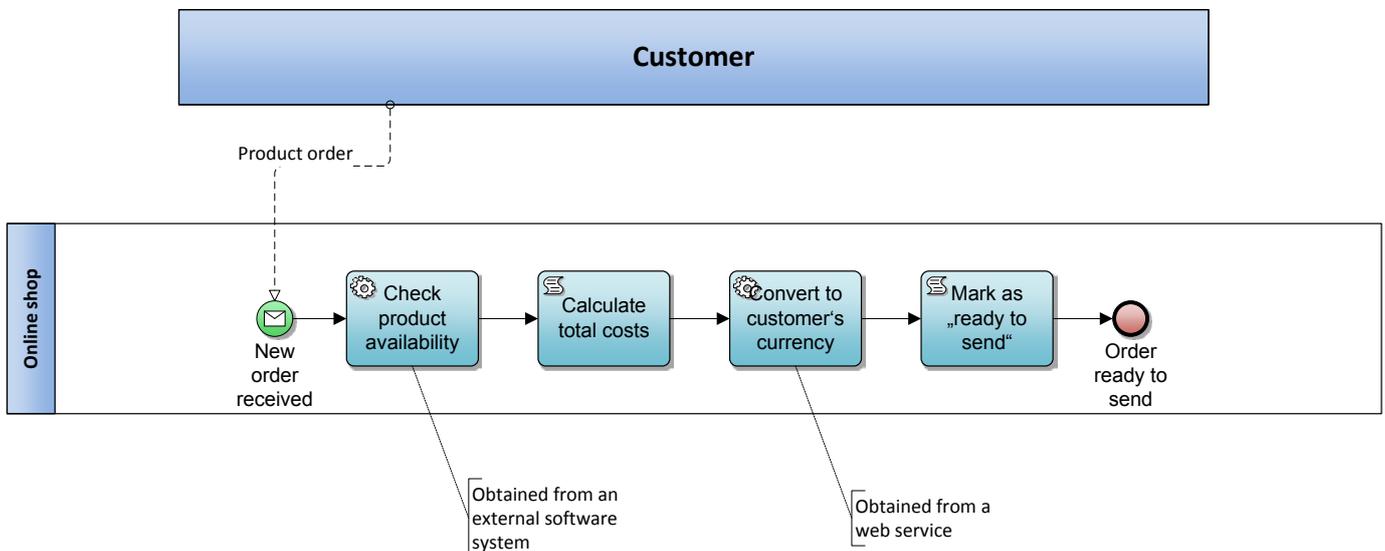


Figure 5: Automated Tasks (service Task and script Task) used in a simplified online shop process



Send Task

A Send Task (Figure 6) is a simple BPMN Task that is designed to send a Message to an external Participant (usually another BPMN Pool). Once the Message has been sent, the Task is completed. The actual Participant who sends the Message can be identified by

connecting the Send Task to a Participant by using a Message Flow. Examples of a Send Task are as follows:

- Inform the collaborator that the process is finished,
- When the purchase is complete, send the invoice to the customer,
- When the article is published, inform the author.

Send Task has the same behaviour as a throwing intermediate message event (Figure 6). However, when using a Task, one is able to attach a boundary event to capture the Task's exception conditions.

Receive Task

The Receive Task (Figure 6) is the opposite of the Send Task. It represents a simple Task that is designed to wait for a Message to arrive from an external Participant (usually another BPMN Pool). Once the Message has been received, the Task is completed. The actual Participant from which the Message is received can be identified by connecting the Receive Task to a Participant using a Message Flow. Examples of a Receive Task are as follows:

- Wait for author's approval before continuing with the publishing process,
- Wait for customer information to resolve an open issue,
- Begin with the delivery process as soon the shipping address is approved.

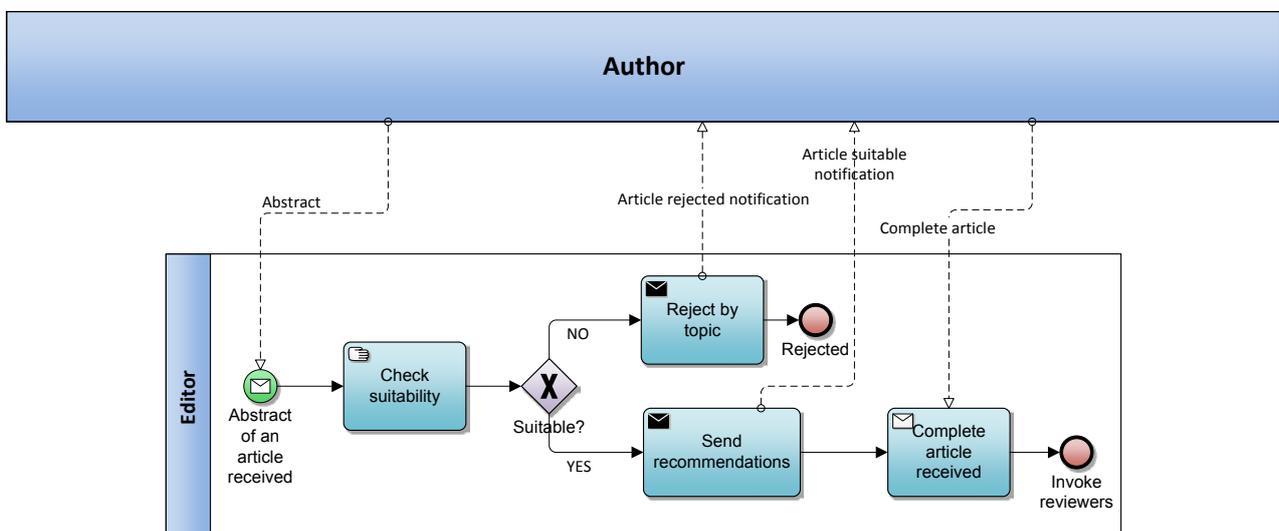


Figure 7: Send and receive Tasks in a simplified authoring process



The analogue to Send Task, a Receive Task has the same behaviour as a catching intermediate message event (Figure 8) [2]. However, a boundary event can be attached to a Receive Event to capture a Task's exceptions.

Figure 8: Sent Task and Intermediate throwing message event

Since Send and Receive Tasks have the same behaviour as the corresponding intermediate message events, they do not have any performer, meaning that they are executed immediately when they are enabled. This means that they are suited for automated message exchange. If a message flow represents some form of human communication (e.g. phone, fax and verbal communication), a User Task should be used instead.

A business rule represents a set of rules and conditions which can be used independently of a specific business process. This means that processes and rules should be considered as complementary rather than alternative ways of representing a business.

Business Rule Tasks

A Business Rule Task provides a mechanism for a Process to provide input to a Business Rules Engine, and to get the output of calculations that the Business Rules Engine might provide (Figure 7). Examples of a Business Rule Task are as follows:

- Define the priority of an open help-desk issue according to the user type,
- Calculate the insurance cost according to the user's profile,
- Send the appropriate product offer according to the user's behaviour.

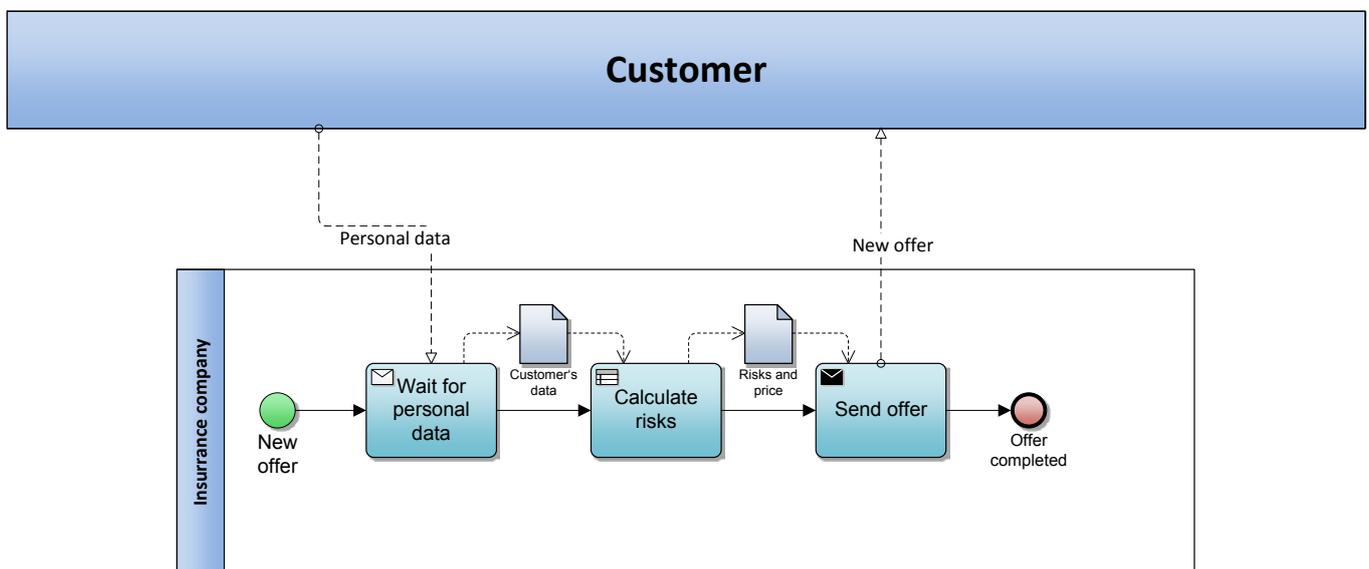


Figure 9: Business Rule Task in a simplified personal insurance process

Note that in case of Business Rule Tasks, the corresponding business rules aren't calculated by the process engine. Instead they are calculated by the business rule engine, which can represent an integral part of a BPMS (Figure 10).

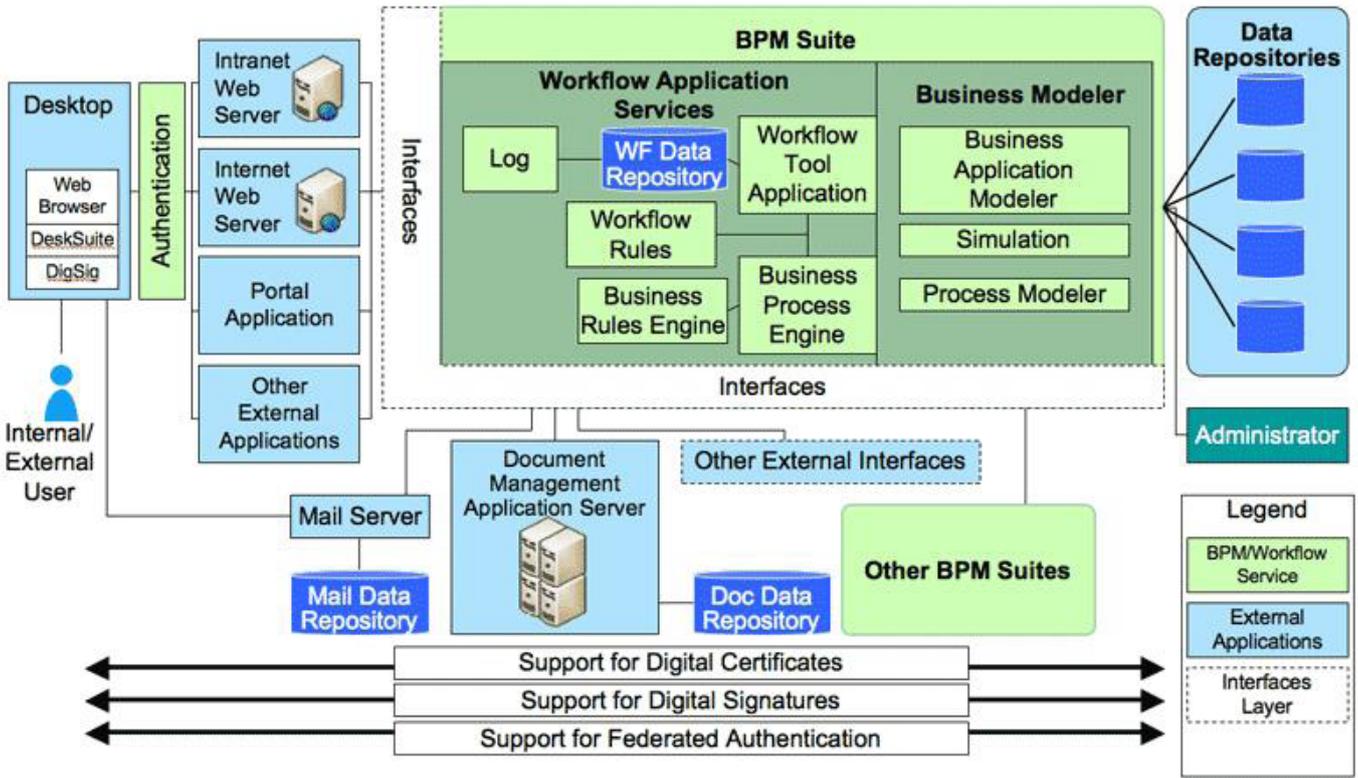


Figure 10: Business process engine and business rule engine in a business process management system (BPM) [Wikipedia.org]

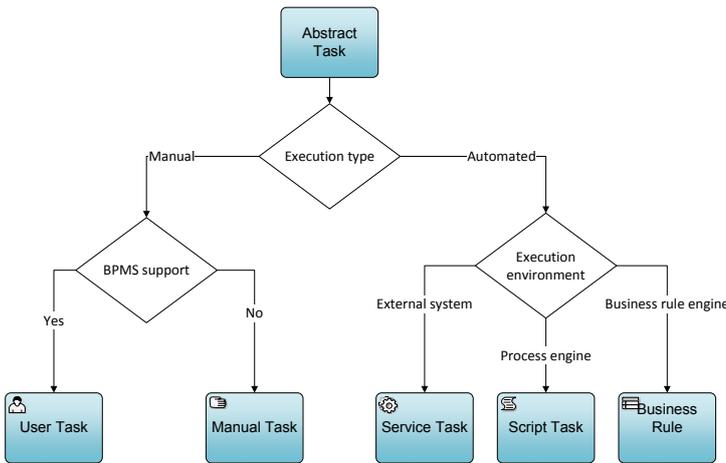


Figure 11: A decision tree for selecting appropriate Task Type

Conclusion

This paper has presented BPMN 2.0 Task Types, as defined in the BPMN 2.0 specification, and has illustrated these through examples. BPMN 2.0 defines seven special Task types and an Abstract Task, which can be used in all other cases. Each Task Type is annotated by a specific graphic symbol on the top left side of the Task element. Figure 11 represents a decision tree which simplifies the selection of an appropriate Task Type.

Beside the Task types presented in Figure 11, BPMN 2.0 also specifies the Send Task and Receive Task, which can be used in case of (automated) message exchange.

References

- [1] OMG, "Business Process Model and Notation (BPMN) Version 2.0." Jun-2010.
- [2] B. Silver, BPMN method and style. Aptos Calif.: Cody-Cassidy Press, 2009.

© Copyright 2013 Orbus Software. All rights reserved.

No part of this publication may be reproduced, resold, stored in a retrieval system, or distributed in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

Such requests for permission or any other comments relating to the material contained in this document may be submitted to: marketing@orbussoftware.com

Orbus Software

3rd Floor
111 Buckingham Palace Road
London
SW1W 0SR
United Kingdom

+44 (0) 870 991 1851
enquiries@orbussoftware.com
www.orbussoftware.com

