

13.1 To Go

We're making great progress using the method. I think the suite of tools actually helped us very well to describe the models, to try them out, and to implement them.



Using the specialized tools, it is much simpler to write down what we actually mean. It is so much more tedious with standard applications for writing and drawing. I especially enjoyed being able to immediately interactively test my specifications.



The validation in particular was so much easier. We could interactively evaluate the various models without endless interviews and theoretical discussions, and without traveling around.



Yes, that does make everything easier and faster. I think that the direct implementation has saved us a lot of time. With the help of the simulation, we were also able to better assess how many resources we need to execute the processes.

I think it is important for us to respond quickly to changes in our business environment, and the tools allow us to do so.



However, only after we have not only defined the targets, but also clarified further activities for moving forward. For Actors, this would mean that their interventions along the process sequence have been determined.



Speaking for management, we are reassured that everyone knows how they need to work. Compliance is virtually no longer a problem, because the process models are documented seamlessly, and the resulting key performance indicators provide a good overview of the company situation.



In the following sections, we provide insights into jBOOK, jSIM, and the Metasonic Suite, exemplifying a set of tools for each activity bundle in the development process for business process applications. jBOOK is a documentation tool to support subject-oriented analysis. jSIM can be used by Actors to simulate processes based on subject-oriented models on the computer.

The Metasonic Suite consists of a number of elements: the module “Build” supports the modeling of the subjects, their behavior, their interactions, and the thereby exchanged messages and business objects. “Proof” enables distributed, computer-aided validation and “Flow” as a process engine controls the execution of instances for all subjects involved in the process. The base module includes the “Usermanager”, which allows those responsible for organization-specific implementation the assignment of users to roles and subjects.

The subsequent content is illustrated mainly with screen shots, but should not be understood as a step-by-step guide of how to use the tools. It should give rather an impression of the practical work with the tools in each activity bundle of the S-BPM process model, ranging from the analysis of a process, modeling activities, validation, optimization, and implementation as executable workflow to monitoring during operation.

13.2 Process Analysis

For analysis activities, jBOOK provides appropriate checklists and form templates supporting the documentation of results. Figure 13.1 lists, as a practical guide, the activities within the activity bundle for analysis. We explain these in more detail below.

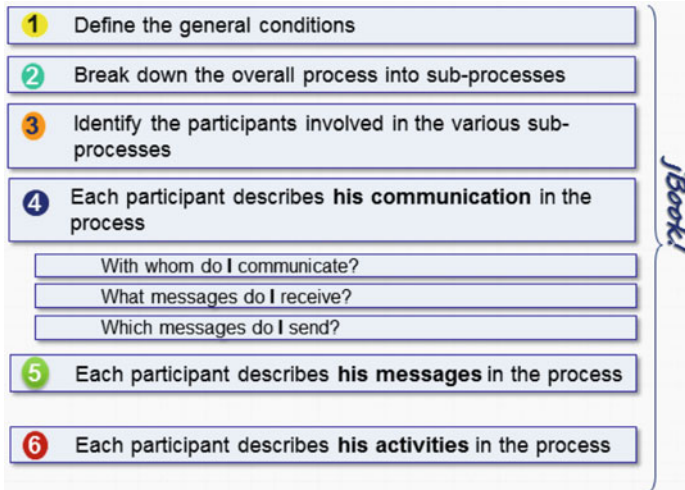


Fig. 13.1 Activities of analysis

Depending on the intensity and level of detail when performing analysis steps, the results can already include many elements of modeling. The team working on a process decides by itself, depending on the situation, to what extent details are already explored in the course of analysis, or instead should be considered later on.

The first step is to define the general conditions for accomplishing tasks in the appropriate form (see Fig. 13.2). This includes information such as name, objectives, tasks, success criteria, contribution to organizational success, and participants of the process. Furthermore, any risks are identified, described, and evaluated. These conditions should provide a brief overview of the position of an observed process in the organizational environment.

The process objectives can be refined on the basis of an overview. jBook provides a separate template to this respect, in particular, to establish criteria to measure and evaluate the achievement of objectives (see Fig. 13.3).

In the process map, it has already been specified that the process of applying for business trips poses no risk to the organization. Figure 13.4 shows a form designed to capture potential risks in detail.

Once the general conditions of the selected process are defined, the analyst can, in the second step, detail its structure. In this case, he splits the business trip handling into two subprocesses, the application process and the booking process, the latter of which is run at the travel agency.

Figure 13.5 shows the resulting structure and the messages exchanged between the subjects in the subprocesses, namely as a process network diagram (see Sect. 5.5.2).

Based on the process structure, in a third analysis step the subjects of subprocesses need to be identified and their essential activities specified (see Fig. 13.6).

1

Define the general conditions

Process name:	Business Trip Application
Process owner:	Mr. Ernst Haftig
Author:	J. Doe, E. Haftig
Date:	November 05, 2010
Company:	
Organization:	

Initial situation

Currently, applications for a business trip will be delivered in paper form.

The approved trips are also given in paper form to the travel office which carries out the administrative tasks and sends the appropriate booking orders by mail to the travel agency.

This returns the documents by mail to the company.

Objectives and success criteria

The throughput times of business trips are to be reduced, and the processing status should always be detectable for all concerned persons.

Risks and risk reduction

As a result of the process, there are no special risks for the company.

Process participants

Employee

Manager

Travel office

Travel agency

Fig. 13.2 General conditions form of a process

In the next step, the analyst describes the communication between subjects. To do so, he collects and documents which messages a subject receives from others, or sends to others, respectively (see Fig. 13.7).

For a more detailed specification of the messages, jBOOK provides a template for parameters as shown in Fig. 13.8.

The description of the communication between the involved parties can be included in the analysis, but this is not mandatory. It is advisable to include it, if at the time of the analysis the respective information is already available and can easily be complemented to the specification. The resulting documentation,

1 b Objective definition	
Objective	
	Objective of process definition and introduction is to process business trip requests more quickly, and to get an overview of the processing status of each application at any time.
Success criterion:	
	Measuring the throughput times for business trip requests
	Report on processing status at the push of a button

Fig. 13.3 Form for detailing objectives

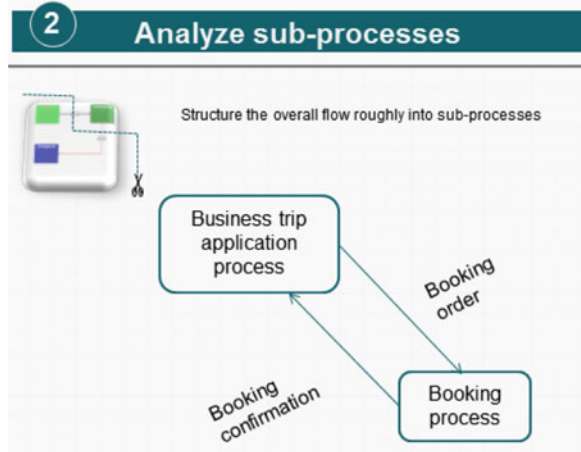
Risk
No specific risks have been identified
Detection criterion (occurs when):
Measures:

Fig. 13.4 Form for detailing risks

however, will be complete only in exceptional cases. Therefore, traditionally, information about the message exchange and the parameters of the messages is added in the course of modeling, or even right from the beginning, first collected and described in detail there.

The same statement holds analogously for the sixth step of the analysis, the description of the subject activities. The analysis usually only leads to a rough outline that needs to be refined when modeling, i.e., describing the subject behavior. A typical basic behavior description for the subject “employee” is shown in Fig. 13.9.

Fig. 13.5 Subprocesses of the process “business trip application”



13.3 Process Modeling

The results of subject-oriented analysis are complemented and accurately detailed in the context of modeling activities. In the following sections, we show how to use the module “Build” of the Metasonic Suite to enrich models with further details.

13.3.1 Process Overview

The starting point of modeling is the process map, which is based on step two of analysis. The tool allows the modeler to structure subprocesses in the form of process network diagrams (PND) (see Sect. 5.6.2). These shows how the subprocesses “business trip application” and “booking” are mutually related, and the possible interactions between the concerned subjects (see Fig. 13.10).

The interactions in the overview do not yet need to correspond to individual messages. Thus, an interaction can be refined if needed into multiple messages in the communication view (see Sect. 13.3.2). In our example, this is not the case. The interactions between the processes consist of single messages, the booking order, and the booking confirmation, respectively.

13.3.2 Communication View

The refinement of the process overview leads to the communication view, which is represented in the modeling phase by subject interaction and communication structure diagrams (SID, CSD) (see Sect. 5.5.3). As an input the modeler can use the information from the completed jBOOK templates gained in the analysis steps two, three, and four (see Sect. 13.2).

3

Define participants

Sub-process	Sub-process
<div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #e0f2f1; margin-bottom: 5px;">Bt-application process</div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Subject Employee</div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Role</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Requests business trip</div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div>	<div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #e0f2f1; margin-bottom: 5px;">Booking process</div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Subject Travel agent</div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Role</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Accepts booking order and confirms booking</div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div>
<div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Subject Manager</div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Role</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Checks Bt-request</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Justifies rejection</div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div>	<div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Subject</div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Role</div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div>
<div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Subject Travel office</div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Role</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Initiates bookings at the travel agency</div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div>	<div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Subject</div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #bbdefb; margin-bottom: 5px;">Role</div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 20px; margin-bottom: 5px;"></div>

Fig. 13.6 Form for naming the subjects of subprocesses and identifying their main activities

Figure 13.11 shows how the “Build” tool displays an interaction diagram containing the subjects of the process “business trip application”. The subject “travel office” is an external subject, representing a corresponding interface as part of the process “booking” (see Chap. 5.6.2).

Figure 13.12 shows the process “booking”. From here, the external subject “travel office”, as an interface subject, refers to the process “business trip

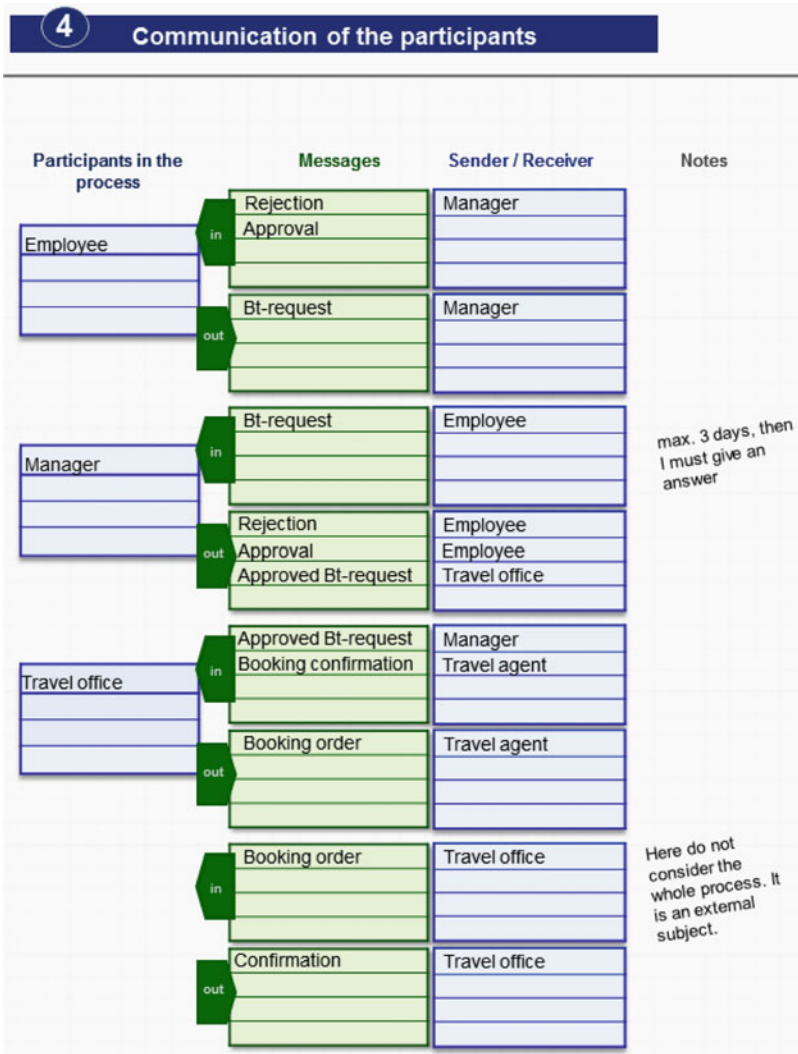


Fig. 13.7 Form for documenting the communication between subjects

application” in which it resides. The communication partner to the travel office in the process “booking” is the internal subject “travel agent”.

Looking back again to the lower part of Fig. 13.11, it can be noted that the interface subject “travel agent” in the process “booking” [*Properties* tab, *Link (relative)* selection list] is termed “travel agent” (*Related Subject* selection list) rather than “travel agent”. This means that the interface subject and the corresponding internal subject (in this case, in the “booking” process) need not be named identically.

Name	Bt-request			
	Parameter	Type	Mand.	Medium/Remarks
First name		<input type="checkbox"/>		
Name		<input type="checkbox"/>		
Personnel number		<input type="checkbox"/>		
Start of trip		<input type="checkbox"/>		
End of trip		<input type="checkbox"/>		
Reason for trip		<input type="checkbox"/>		
		<input type="checkbox"/>		
		<input type="checkbox"/>		
		<input type="checkbox"/>		

Fig. 13.8 Form for defining messages and message parameters

6

Outline the sequence

What activities do I run in what order?

Subjekt: Employee

Requirements		Tools
<p>1. Submit Bt-request in a timely manner</p>	<ul style="list-style-type: none"> Fill in Bt-request and then send it to the manager. Wait for answer of the manager. If rejected, possibly submit changed Bt-request. Otherwise, make travel arrangements. 	<p>1. Intranet with form</p> <p>2. Documents</p>

Fig. 13.9 Form for describing a subject's behavior

For reasons of clarity however, identical identifiers are recommended, as with the travel office (see bottom part of Fig. 13.12). However, in practice, this is not always possible, especially in cases in which the subprocesses are located in different organizations that need to be connected via interface subjects, and there are already historically defined names for organizational units or roles.

13.3.3 Subject Behavior

The next step in modeling is the definition of subject behaviors. The methodology provides the subject behavior diagram (SBD) for this purpose (see Sect. 5.5.5). Starting point is the data collected in step six of process analysis (see Sect. 13.2).

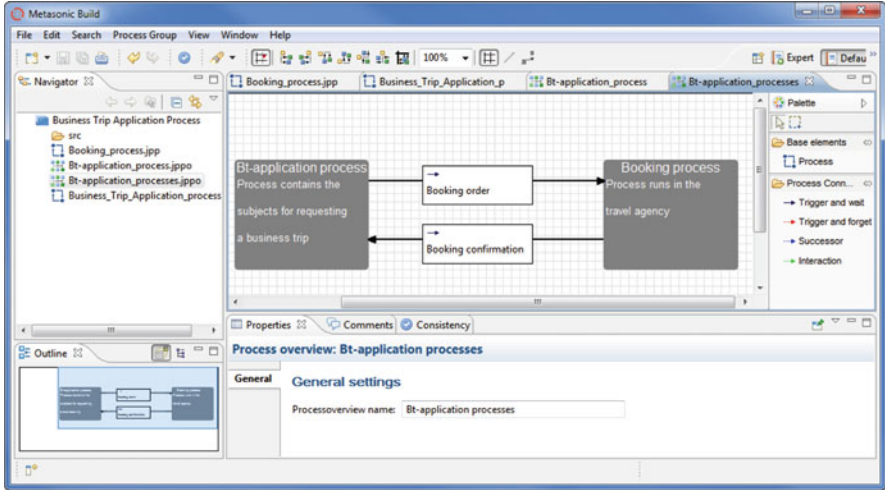


Fig. 13.10 Process network diagram “business trip application process”

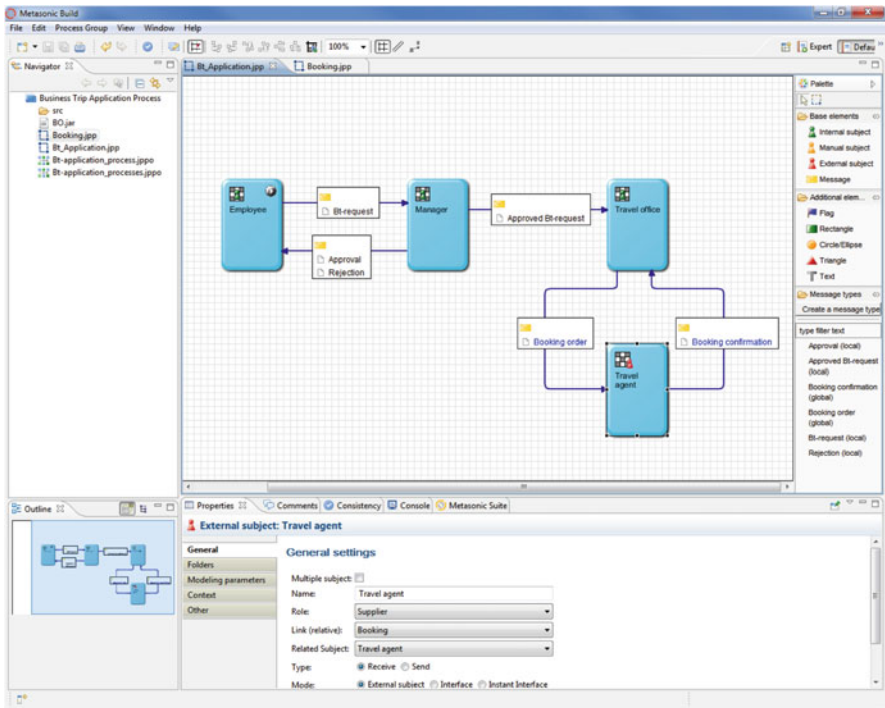


Fig. 13.11 Subject interaction diagram of the subprocess “business trip application”

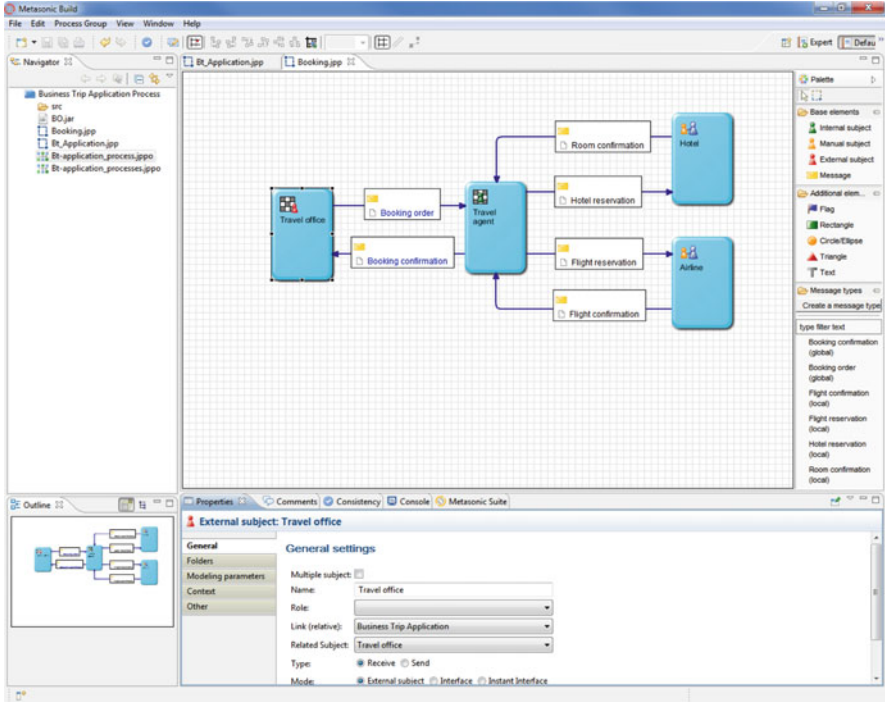


Fig. 13.12 Subject interaction diagram of the subprocess “booking”

Figure 13.13 shows the sample SBD for the subject “employee” of the “business trip application” process created using the modeling tool. Function states are characterized as rounded rectangles with a small clock icon, send and receive states with a small envelope icon with incoming or outgoing triangles, respectively. The transitions are specified using conventional rectangles with a horizontal arrow and a standardized verbal description.

This behavior reveals that the employee fills out the application form first and then sends it via “provide business trip request” to the manager. Then he waits for the response of his manager. This can be “approval” or “rejection”. In the first case, the employee takes the trip and then reaches the end state. In the case of “rejection”, the subject will immediately proceed to the end state.

So far, we have considered only the logical flow of the process. However, the model specification can already be executed at this stage. This means that participants are able to test the business logic in a distributed role play. Before discussing this further in the context of the validation process described in Sect. 13.4, we explain the required data modeling activities. The modeler needs

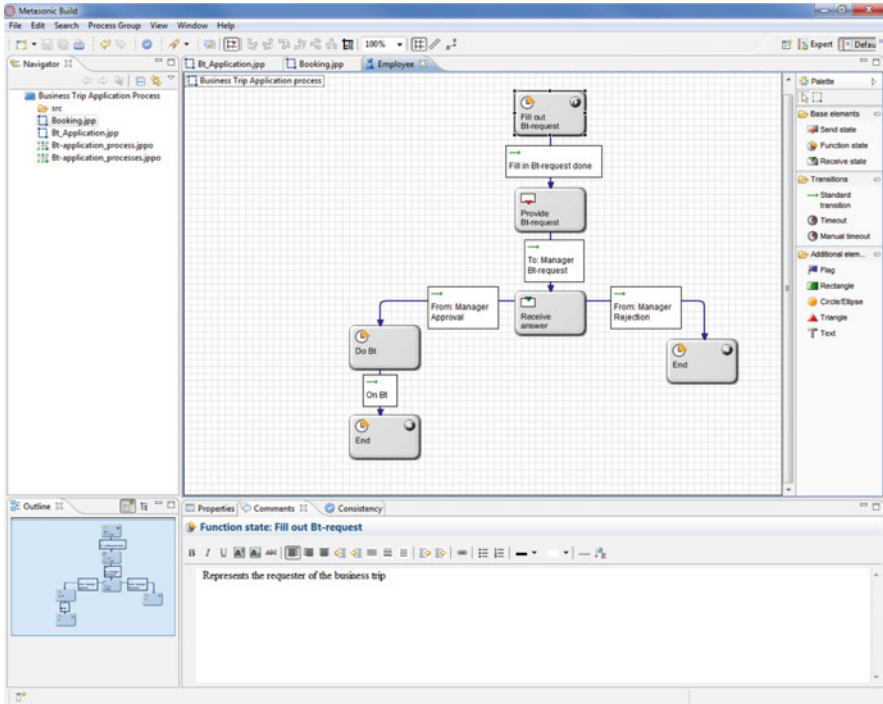


Fig. 13.13 Behavior description of the subject “employee”

to specify what data exists in each subject state, and what messages transfer it between subjects.

The S-BPM method provides business objects for this purpose. They can be a complex structure, with different statuses, views, and access rights (see Sect. 5.5.7). For their manipulation, appealing user interfaces should exist (see Sect. 10.5.1.1). The functions provided by the modeling tool for the detailed definition of business objects are discussed in Sect. 13.6.

Here, we show instead how to quickly and easily define data, in order to test in the subsequent validation, whether they are even the right data, before refining their definition. Using this straightforward approach, complex business structures with object data types, plausibility rules for entry, etc. are not yet created, but rather simple data elements, which are initially sent as parameters using messages. The definition of such primitive business objects occurs on the level of business processes. The required information may stem from the jBOOK form for describing messages, completed in step four of the analysis (see Sect. 13.2).

Figure 13.14 shows the data (parameters) required for the process “business trip application”. Not all of this data is used in all subjects. However, each subject has its own set of variables for these parameters. Hence, a change of name in the subject “employee” is not visible to the other subjects. Instead, the value of this variable “name” needs to be transferred with a message containing the parameter “name” to another subject that should know the value. When accepting the message, the value of this message parameter is transmitted to the variable “name” of the receiving subject “manager”. Thus, the variable “name” in the subjects “employee” and “manager” has the same content.

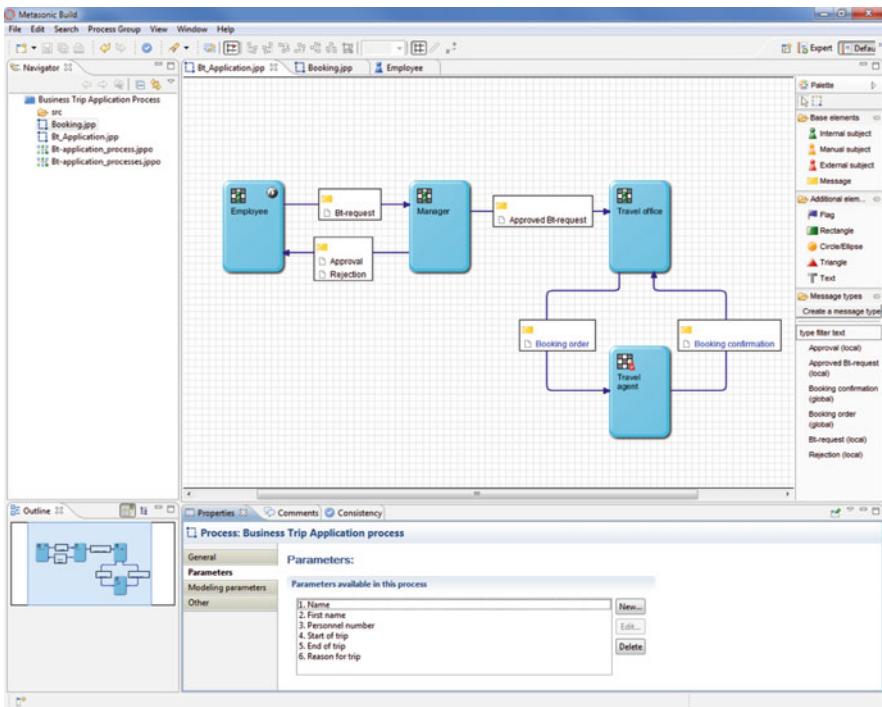


Fig. 13.14 Modeling the data of the process “business trip application”

Each subject can potentially access all process parameters, which can be filled with values by internal functions in the subject behavior. Figure 13.15 shows this for the assignment of values to the variables “name”, “first name”, “personnel number”, “start of trip”, “end of trip”, and “reason for trip” in the function state “fill out business trip request”.

For the transmission of parameter values between subjects, they need to be assigned to appropriate messages. Figure 13.16 shows this assignment for the message type “business trip request”, sent by the employee to his manager.

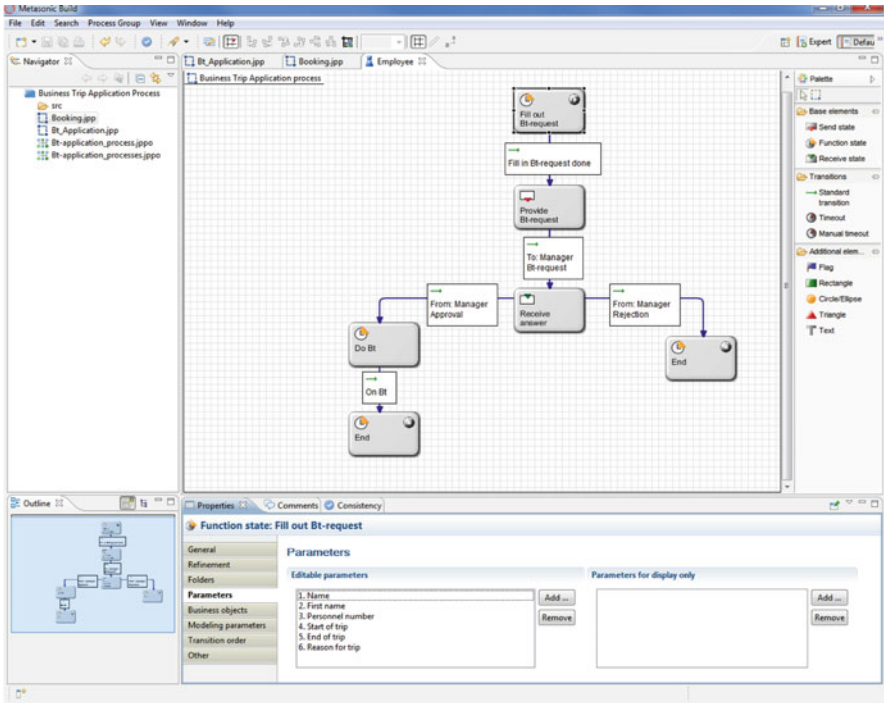


Fig. 13.15 Parameter assignment in the internal function “fill out business trip request”

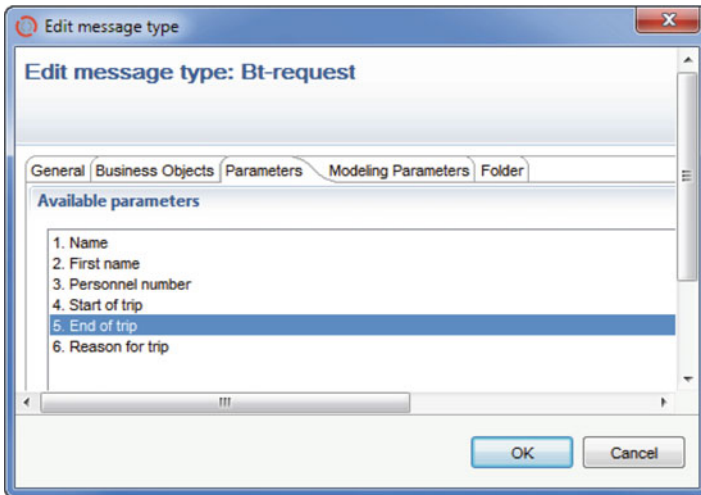


Fig. 13.16 Modeling of the parameters of the message type “business trip request”

When receiving a message, the values are transferred from the message parameters to the subject's local variables with the same name. Thus, the business trip request data is available after the receipt of the message "business trip request" by the subject "manager" for use in its internal checking function. Hereby, the data is delivered by which the supervisor in his check function decides whether the transition to "reject" or "approve" will subsequently be executed (see Fig. 13.17).

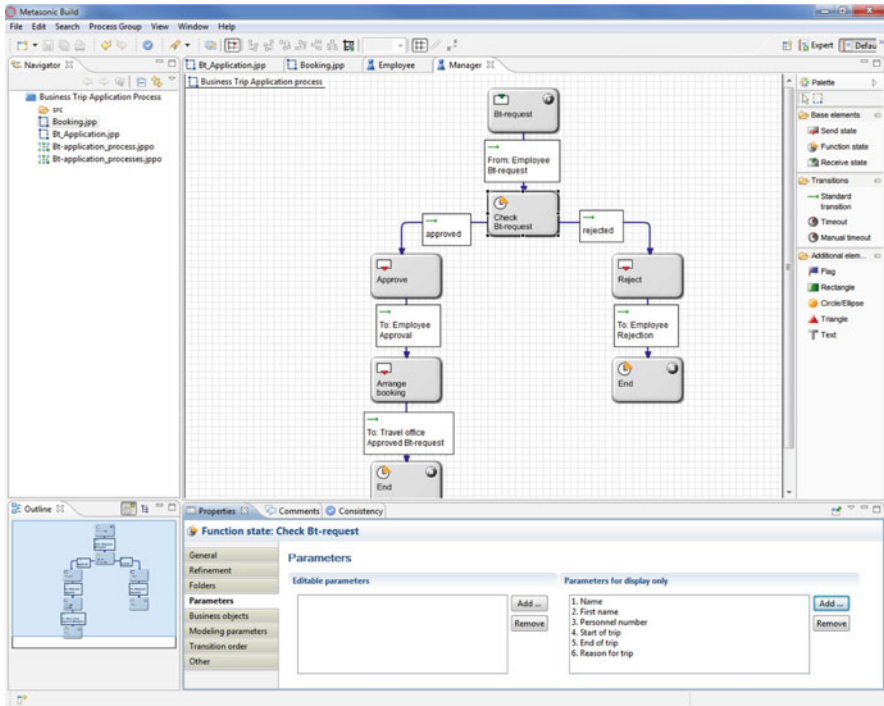


Fig. 13.17 Modeling the receipt and use of parameter values

13.4 Process Validation

The process model contains in the current status all information regarding the logical flow of the process, the data required in the process, and the data variables either being used by the subjects or being exchanged between them by sending and receiving messages. Although the business objects are currently defined only in the previously introduced primitive form, i.e., without data types, value domains, origin of values, etc., the existing model can already be tested in a role play. This involves reviewing the following two questions:

- Does the described process logic correspond to the desired way of working?
- Do the data variables meet the process objectives?

For implementing an IT-based role playing (see Sect. 7.5.2.2), the process model is transferred by the click of a button into the appropriate execution environment in

the module “Proof” of the Metasonic Suite. This environment is available via Internet or Intranet and can be accessed by a browser via its address (URL). Employees who are involved in a process can now use the subject, as it represents their share in the process: an employee applying for business trips uses the subject “employee”, a manager the subject “manager”, and an employee of the travel office the subject “travel office”.

These individuals can validate the process from their respective workstations. Each of them sees the behavior of the subject which he represents, and for which he will later be responsible in process execution. Each participant enters the necessary values of variables for his respective behavior states, i.e., works on the primitive form of business objects occurring in the process. By exchanging this information in accordance with the process flow, they quickly notice whether parameters for task accomplishment are missing, or redundant, etc. The participants can immediately overcome such deficiencies by using the “Build” module, then restarting the test environment “Proof” with the modified model, and examining the effects of the modification.

Figure 13.18 gives an overview of the control windows of the validation environment for the subject “employee”. The left window shows in which state the subject currently is (function state “fill out business trip request”). By clicking on the “parameter” icon, the middle window will be displayed to enter values. In the example, this has already happened. Closing of the input leads to delivering the message “business trip request” to the manager in the right window of the screenshot.



Fig. 13.18 Validation user interface of the subject “employee”

Figure 13.19 shows the interactive window for the subject “manager” indicating the receipt of a business trip request (left window). The manager accepts, by clicking on the icon with a right arrow, and changes from the receive state to the state “check business trip request” of his behavior, where he can then decide between the options “approved” or “rejected” (middle window). For decision making, he can display the trip data by clicking on the parameter icon (right window).

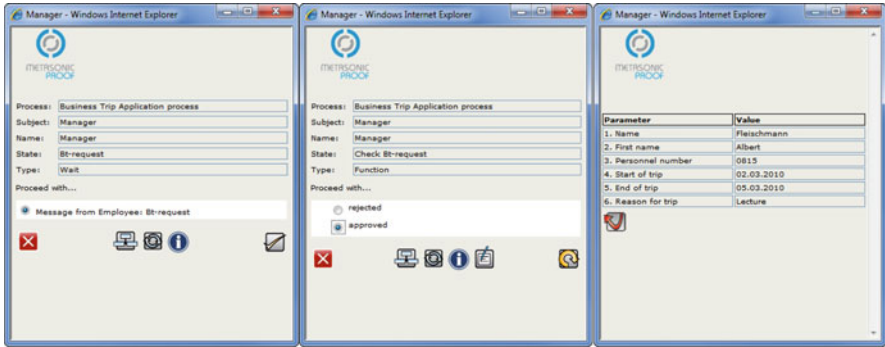


Fig. 13.19 Validation user interface of the subject “manager”

An iteration in such a validation session corresponds to the execution of a process instance. A recorder documents each step of a validation session. The steps can be displayed in a swim lane diagram (see Fig. 13.20).

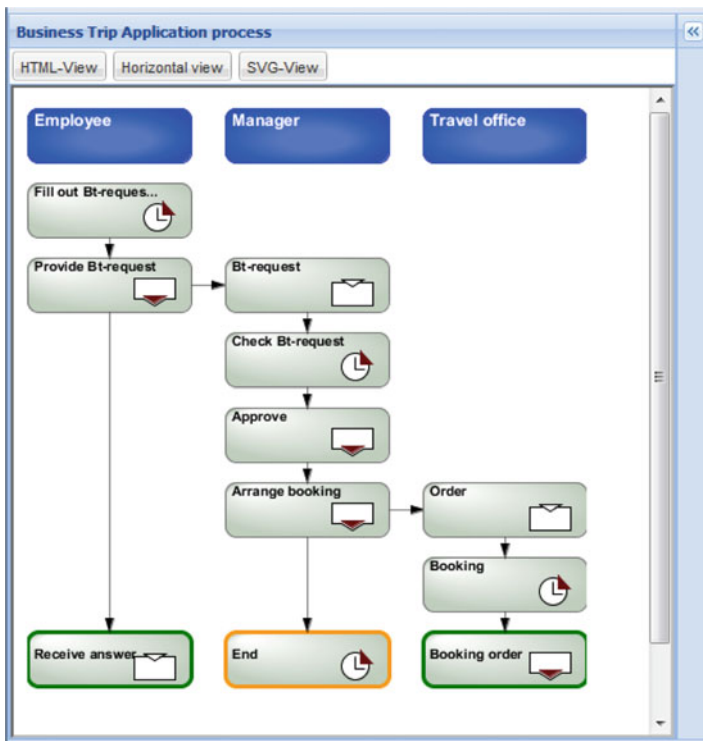


Fig. 13.20 Swim lane protocol of one iteration of a validation session

In this way, an arbitrary number of validation sessions addressing different variants of process iterations for a process model can be performed, potentially with changing participants. This allows the parties to review whether the process corresponds to the desired way of working. Through the recording of the validation iterations, the test coverage can also be estimated.

13.5 Process Optimization

The validation checks whether the described process corresponds to the intended way of working, i.e., whether the right action is taken. Optimization is on the other hand about checking whether the validated process can be performed with minimal effort (see Sect. 8.2). For an associated simulation, it is necessary to determine, or at least estimate, the time required for each activity within a subject. In addition, it needs to be known how often per time unit a corresponding process instance is created and put into execution. Since such information is usually enriched with probabilities, the parameters for probability densities need to be known. Finally, resources need to be assigned to the subjects before starting a simulation run. Figure 13.21 shows the main screen of the jSIM tool for a subject-oriented simulation determining resource requirements and costs incurred in the execution of a process.

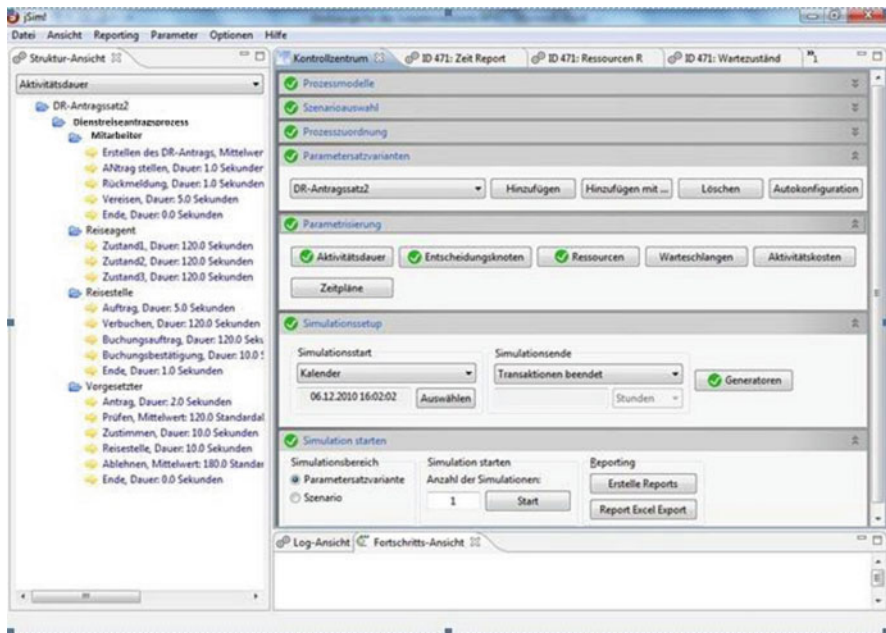


Fig. 13.21 Main screen for entering simulation parameters

In Fig. 13.22, the times required for the accomplishment of individual actions in the respective subjects are shown. Thus, the duration of creating the business trip request is distributed normally with an expected value of 180 s and a standard deviation of 40 s. For reasons of simplicity, in this example the other time parameters are assumed to be constant.

Business trip application process						
Subject/Activity	Distribution	Value1	Value2	Unit	Activity cost	Cost category
Employee						
Fill out Bt-request	normal distribution	180	40	Seconds		
Send Bt-request	constant	1		Seconds		
Receive answer	constant	1		Seconds		
Do Bt	constant	5		Seconds		
End	constant	0		Seconds		
Travel agent						
State 1	constant	120		Seconds		
State 2	constant	120		Seconds		
State 3	constant	120		Seconds		
Travel office						
Order	constant	5		Seconds		
Booking	constant	120		Seconds		
Booking order	constant	120		Seconds		
Booking confirmation	constant	10		Seconds		
End	constant	1		Seconds		
Manager						
Bt-request	constant	2		Seconds		
Check Bt-request	normal distribution	120	60	Seconds		
Approve	constant	10		Seconds		
Arrange booking	constant	10		Seconds		
Reject	normal distribution	180	60	Seconds		
End	constant	0		Seconds		

Fig. 13.22 Excerpt of simulation parameters

It becomes apparent that the determination of the parameters for the simulation is not trivial and requires extensive experience. Even after this hurdle has been taken, the interpretation of simulation results requires advanced skills. In Fig. 13.23, an excerpt of the simulation results is presented. The graph shows the minimum and maximum activity and waiting time, and the minimum and maximum resource requirements for a given instantiation of processes.

13.6 Modeling Business Objects and Integrating in Behavior Descriptions

So far, only simple parameters have been used as business objects in process models. They merely served to verify in validation that all required data was included in the model.

In Sect. 13.3.3, we have already mentioned that business objects occurring in a process subsequently require a more detailed and precise modeling specification in

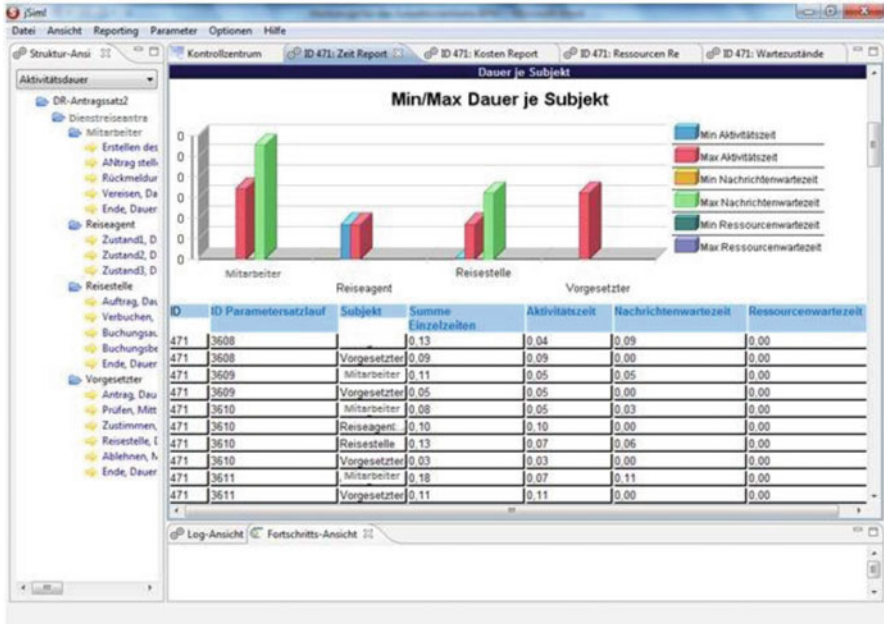


Fig. 13.23 Excerpt of simulation results

order to comply with the requirements of a workflow system being used in practice. This detailed description includes aspects that we have presented in Sect. 5.5.7. Examples here are hierarchical structuring; the definition of states, views including access rights, look-and-feel, value ranges for user input; and the coupling of programs to manipulate data elements.

In the following, we show an excerpt of the potential tool support for detailed modeling of business objects in terms of their subsequent use when executing the process in a workflow engine. The result of this detailed modeling of business objects can also be tested in the validation environment, before implementing a process in a workflow.

Figure 13.24 shows the structure of the business object “business trip request” and its defined views. The application consists of three parts: “personal data”, “information on the business trip”, and “processing status”. Each of these three sections contains respective elements. The modeler can therefore organize a business object with the tool across various hierarchy levels, in each of which data structures and/or individual data elements can occur. For each element, different attributes can be specified, e.g., whether an element could occur multiple times (like a position of an order), whether it is a mandatory field users need to fill in, its specific data type, etc.

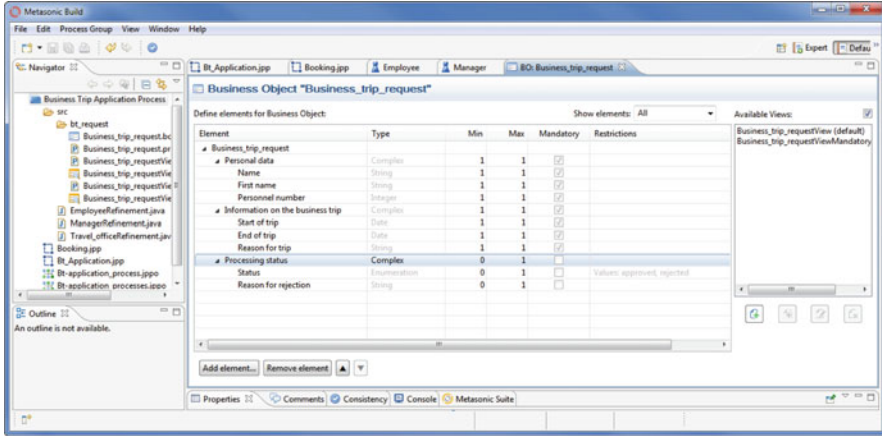


Fig. 13.24 Business object definition using the business object editor

For each business object, any number of views can be specified, each containing subsets of the elements of the object. In this way, the modeler can determine that during execution only an excerpt of a business object is displayed, processed, or transmitted in specific states. Figure 13.25 shows the view “no decision” on the business object “business trip request”, which contains only the personal data and the information on the business trip. The processing status containing the approval notice is not displayed in this view.

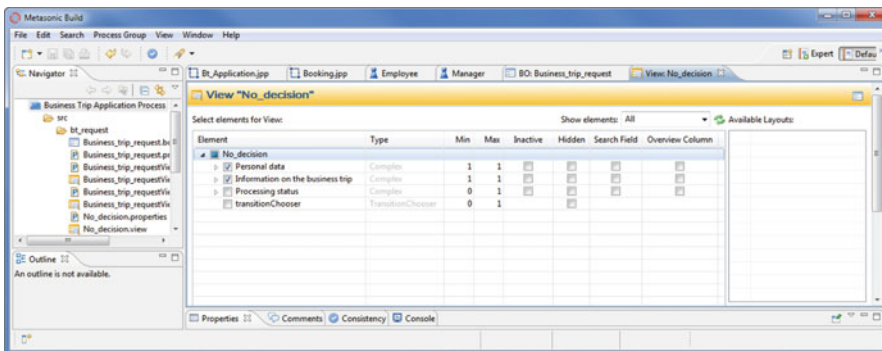


Fig. 13.25 Specification of a view using the business object editor

After having defined the structure, views, and rules (not illustrated), it has to be determined how the business object is to be displayed on the screen. Figure 13.26 shows the editor for specifying the layout.

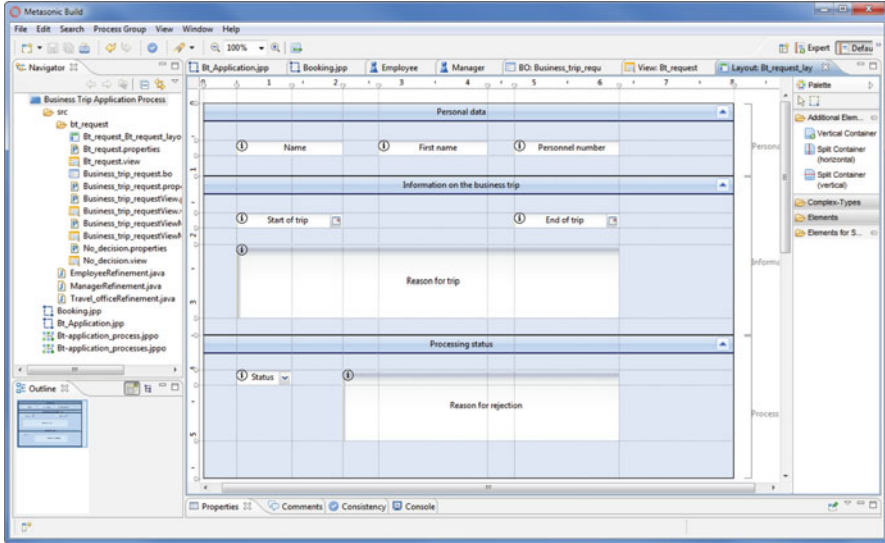


Fig. 13.26 Specification of a form using the business object editor

After completing their definition, business objects need to be inserted at appropriate positions into the behavior description of a process. To do so, the user selects in the modeling tool the state in which the business object is used, e.g., displayed and/or filled out. For this purpose, there are so-called folders. In each state, it is defined what business object types are allowed in a specific folder, and what types of operations can be executed in this state.

Figure 13.27 shows this information for the state “fill out business trip request”, in which the “business trip request” can be created, displayed, and edited.

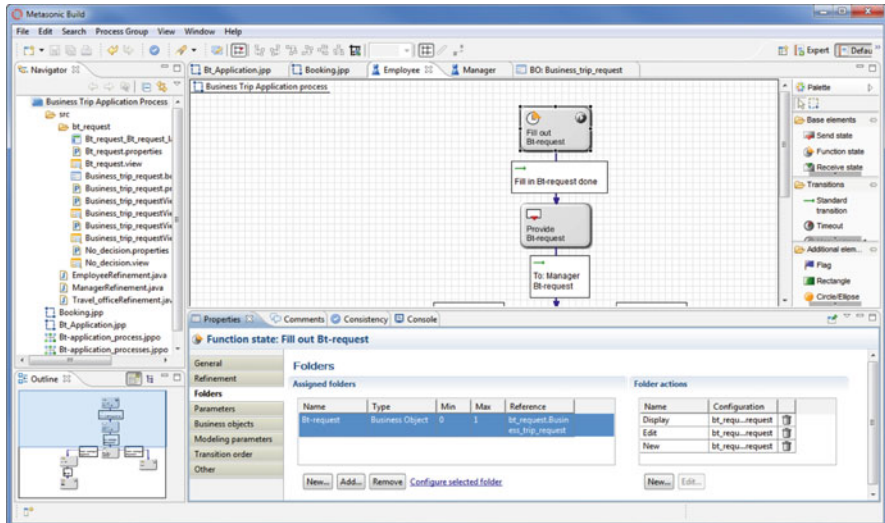


Fig. 13.27 Integration of the business object “business trip request” in the state “fill out business trip request” as part of the behavior specification of the employee

13.7 Organization-Specific Implementation

After describing the process behavior and the business objects, an active agent (subject carrier) needs to be assigned to each subject. This carrier performs the actions of the subject according to the modeled behavior (see Chap. 9).

The assignment of an active agent to a subject is performed using the tool “Usermanager” on several levels. A person (subject carrier) is part of one or more groups (subject carrier groups). One or more of these groups are assigned to a role, and a role to one or more subjects.

Figure 13.28 shows how Mr. Schulz is assigned to the group “employee group” using the “Usermanager”.

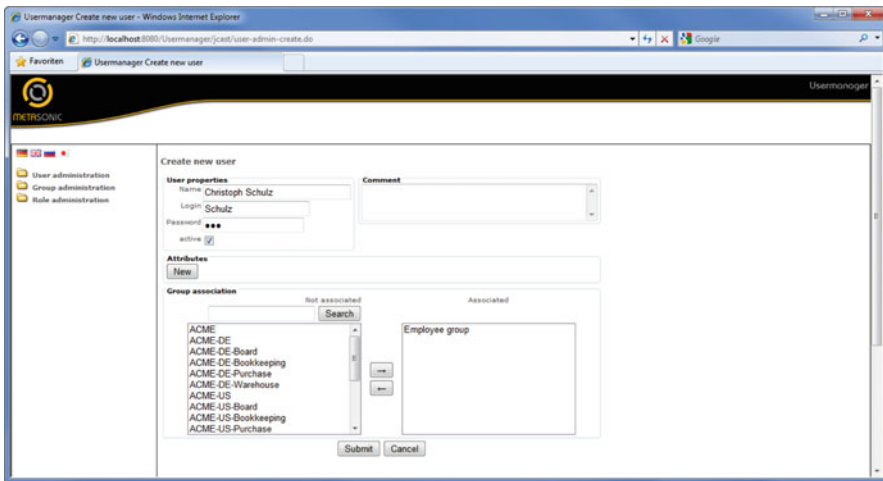


Fig. 13.28 Screenshot of the tool for managing users, groups, and roles

Analogous to the assignment of Mr. Schulz to a group, a role is assigned to a group. The assignment of roles to subjects is performed using the modeling tool. Figure 13.29 shows how the role “employee” is assigned to the subject “employee”.

At the end of the outlined multiple steps, Mr. Schulz is able to submit a business trip request, since he has been assigned to the subject “employee” as subject carrier.

13.8 IT-Specific Implementation

After embedding a process in the organization, the integration of applications needs to be performed. Applications are used to retrieve business object content, to manipulate it, to store it, etc. (see Sect. 10.5.1).

The integration is realized by so-called refinements. They denote software invoked in function states within the subject behavior. Whenever a process enters a

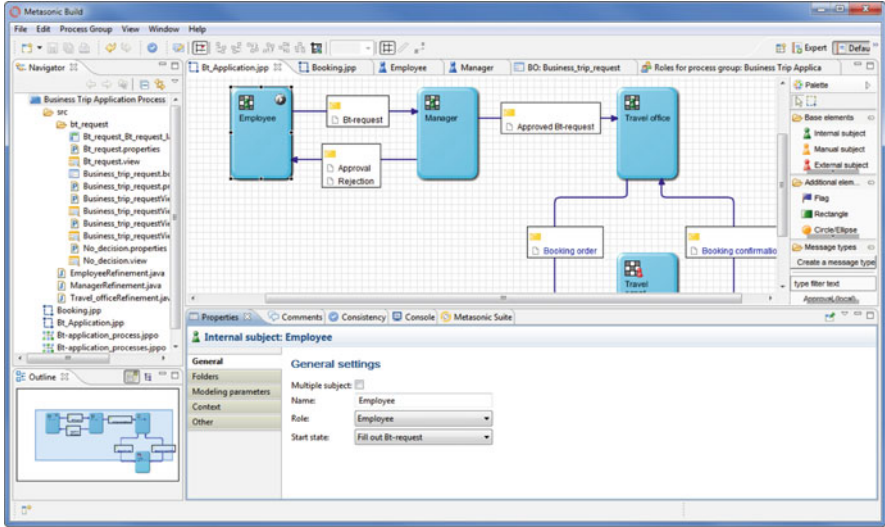


Fig. 13.29 Assignment of a role to a subject using the modeling tool

state with a refinement, the stored program is executed. Such a program may initially serve only to call an existing application having a specific user interface for editing a business object (e.g., an SAP transaction). A refinement could also be code which itself accesses business object content and manipulates it in a dialog with the user.

Figure 13.30 shows the storing of a refinement in the state “check business trip request”. The implementer uses the option “Execute own refinement” to insert a specific refinement to this state.

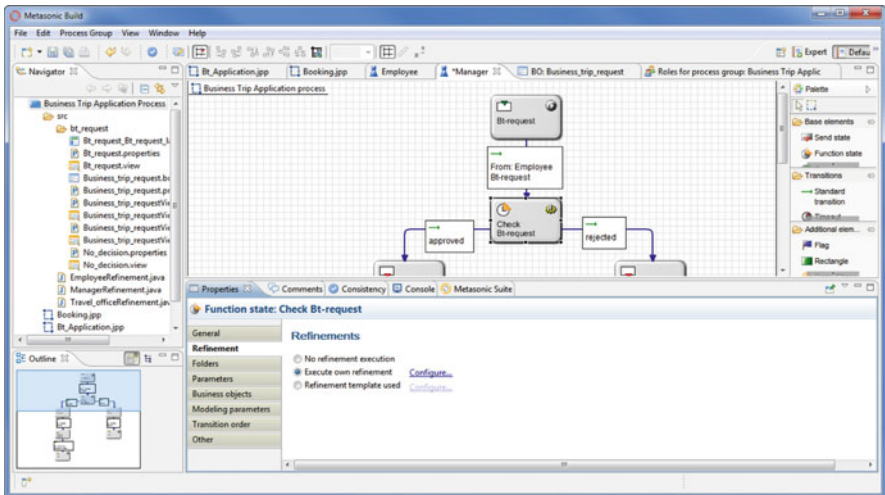


Fig. 13.30 Insert “Execute own refinement”

Figure 13.31 shows the respective potential code body.

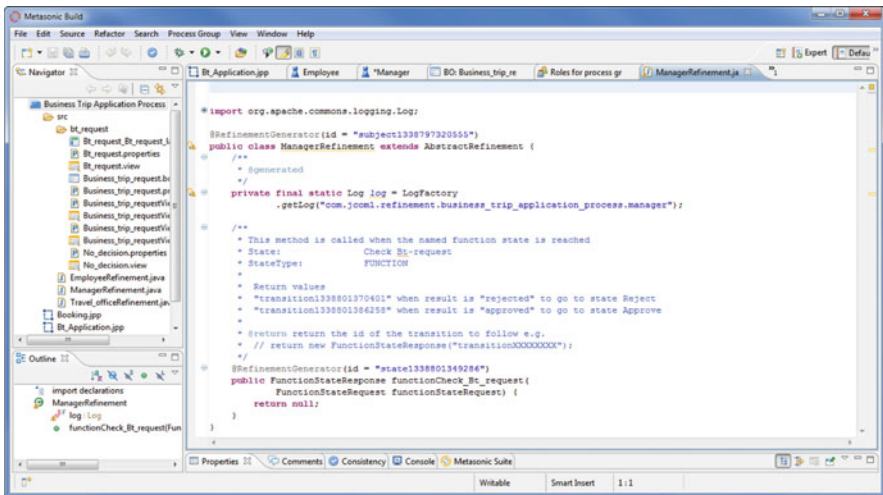


Fig. 13.31 Body of code for a refinement method

13.9 Process Execution

Once the applications running in a process have been integrated, the process can be used productively after extensive testing.

A suitable workflow engine, in our case, the module “Flow” from the Metasonic Suite, interprets the structured process model at runtime and controls the operations from its instantiation to its termination. The engine ensures that the subject carriers perform those actions in each processing step that are expected of them according to the behavioral description of their assigned subjects (internal function, send, and receive). At the designated positions in the model, the engine supplies them with the business objects to be processed and invokes the designated applications.

For instance, Mr. Schulz could log on to the workflow system and create a process instance for applying for a business trip. Figure 13.32 shows the workflow system in the initial state in which the employee submits the request. In the upper part, the respective state of the process is displayed to the user, and in the lower part the business object to be filled out.

After filling out the business object, the user triggers the transition to the next state (top right). After that, the business trip request of Mr. Schulz is transmitted by an appropriate interaction to his manager Mr. Schmid. Mr. Schmid accepts the message with the request and checks it. Figure 13.33 shows the corresponding user interaction.

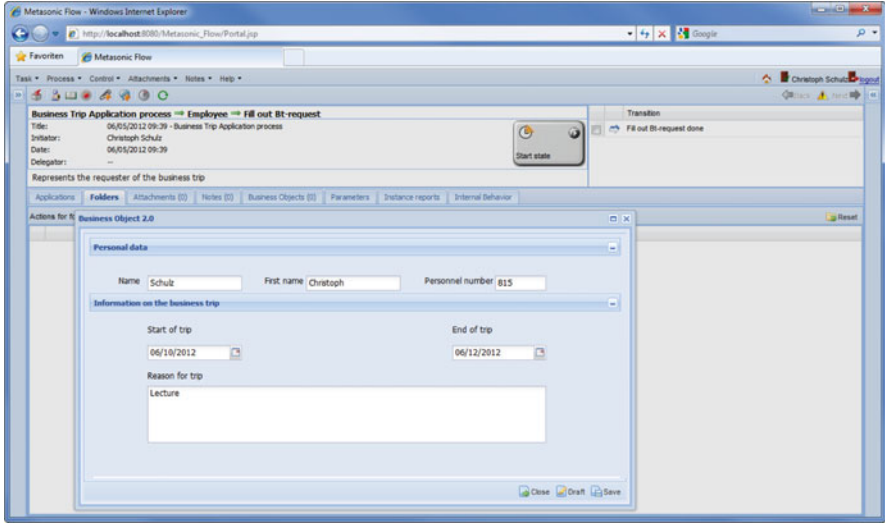


Fig. 13.32 The workflow system in the state “fill out business trip request” of the subject “employee”

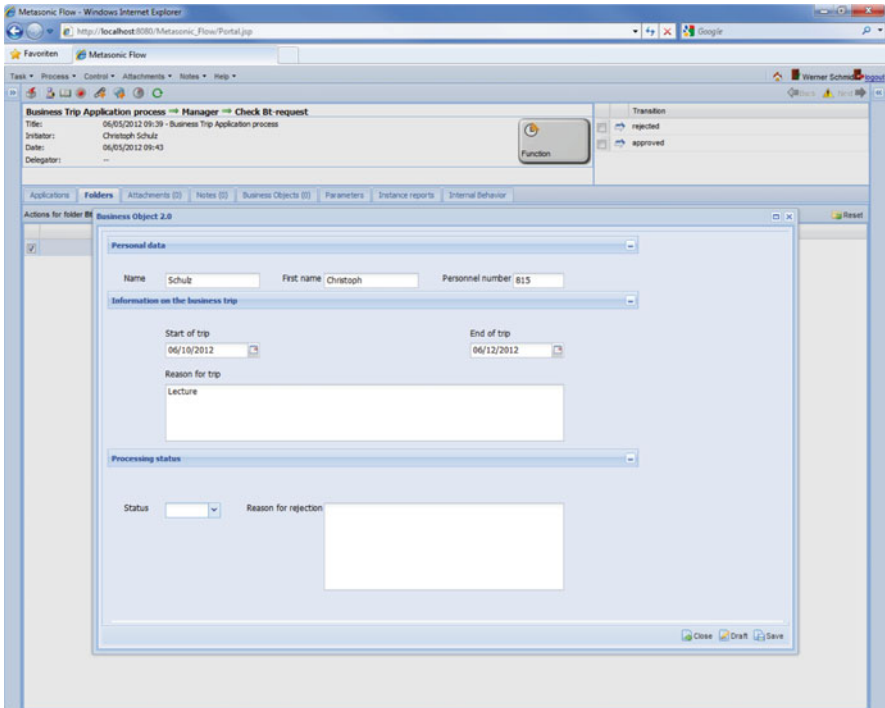


Fig. 13.33 The workflow system in the state “check business trip request” of the subject “manager”

13.10 Process Monitoring

During execution of each process instance, the workflow engine records numerous data. Examples include the state for each process instance, the point in time at which this state is reached, and much more. Such data about instances can be used to observe process executions in an organization (see Chap. 11). Executives can, for example, receive information about how many crucial process instances are currently being executed, or how each process progresses in each process instance.

Figure 13.34 exemplifies a simple list including details of running process instances. It contains the name of the process, its priority, the name of the person who created the instance, the time stamp when it was created, etc. The table includes only a small part of the recorded, and therefore available, data.

Process name	Priority	Title	Creator	Start time	Running time	Duration state	SessionId
Business Trip Application process	Normal	06/05/2012 09:49 - Business Trip Application proc...	Christoph Schulz (Schulz)	05.06.2012	0 day(s)	🛡️	sid64e6c1-see2-1
Business Trip Application process	Normal	06/05/2012 09:48 - Business Trip Application proc...	John Doe (John)	05.06.2012	0 day(s)	🛡️	sid642c660-see2-1
Business Trip Application process	Normal	06/05/2012 09:48 - Business Trip Application proc...	John Doe (John)	05.06.2012	0 day(s)	🛡️	sid68805f41-see2-1
Business Trip Application process	Normal	06/05/2012 09:48 - Business Trip Application proc...	Max Mustermann (max)	05.06.2012	0 day(s)	🛡️	sid6294a80-see2-1
Business Trip Application process	Normal	06/05/2012 09:47 - Business Trip Application proc...	Max Mustermann (max)	05.06.2012	0 day(s)	🛡️	sid6e8f231-see2-1
Business Trip Application process	Normal	06/05/2012 09:39 - Business Trip Application proc...	Christoph Schulz (Schulz)	05.06.2012	0 day(s)	🛡️	sid9765e361-see1-1

Fig. 13.34 List of process instances in the monitoring tool

Such a list representation of the processes running in an organization can quickly become overloaded once further parameters are included. Then, it can become necessary to implement a process cockpit with intelligible indicators and traffic light representations (see Sect. 11.6).

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution Non-commercial License, which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.