Quaestio User Manual

Marcello La Rosa

BPM Group, Queensland University of Technology, Australia
m.larosa@qut.edu.au

1 Introduction

This document provides a guide through the configuration of reference process models via the *Quaestio* tool. The document has been adapted from [2].

Quaestio is a Java application that guides users through a set of questions given a questionnaire model as input. The graphical interface comprises a main window showing a list of Valid Questions, a list of Answered Questions and a Question Inspector.

When a question is picked from one of these lists, the Question Inspector shows the question's details: the list of facts for the question, the dependencies on other questions, and guidelines in natural language to configure the question. In a separate window, a Fact Inspector shows detailed information for each fact: its default value, whether it is mandatory, the constraints that bind the fact, the dependencies on other facts, the level of impact on the system or model to be configured, and specific guidelines to configure the fact. A screenshot of the tool with the Fact Inspector is shown in Figure 1.

The input format is described by an XML schema. Figure 2 shows the representation of the schema as UML class diagram.[1] Here each class maps to a complex type in the schema, the aggregation relation maps to the composition of complex types, and the occurrence constraints map to the element cardinality. The schema encodes some well-formedness rules. This way, non-well-formed models are ruled out (e.g., a model with a fact that is not associated to any question or a set of questions not covering all the facts). For the details of the schema, please refer to [2].

Once a model is loaded, Quaestio shows the set of initial valid questions. Next, for each answer given, the tool dynamically calculates the next valid state and updates the lists of valid and answered questions. The configuration process completes when all the questions have been answered, or at least all the mandatory facts have been set and the remaining ones can take their defaults without violating the constraints. A (partial) configuration can be exported to XML as a list of facts, keeping track of the values that have been set and whether they deviate from the defaults.
The main features of Quaestio are:

– decision support: by means of guidelines, constraints and impact-level;

[1] The diagram has been generated with the Eclipse Modeling Framework (`http://www.eclipse.org/modeling/emf`)

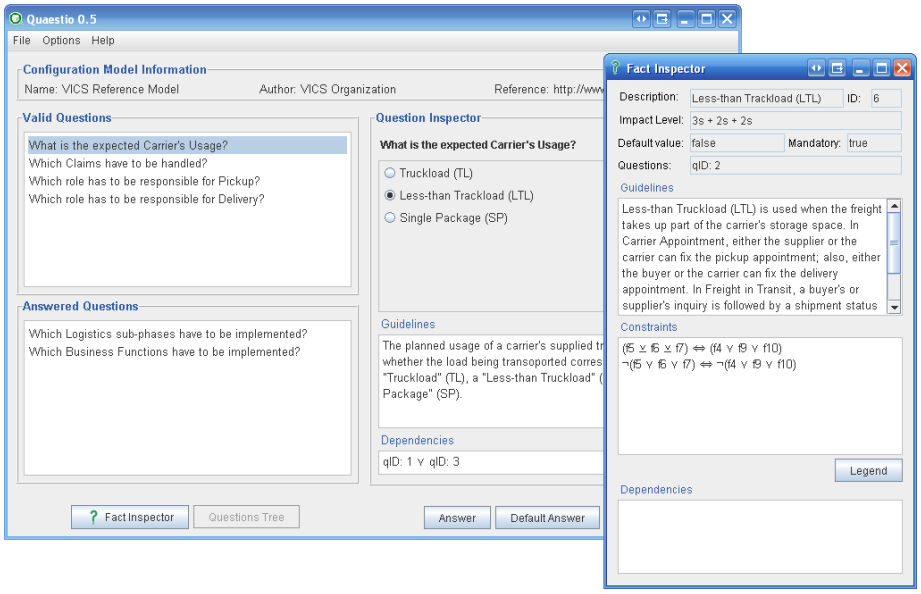

Fig. 1. A screenshot of Quaestio with the Fact Inspector.

- dynamic checking of answers: answers can be given only if they comply with the constraints;
- default answer: default values can be given to all the facts of a question if:
  - the value of those facts that have already been set or that are forceable, does not deviate from the default, and
  - the resulting valuation is valid given the current state;
- fact's value preservation: facts that occur in more than one question are set the first time and then preserve their value in subsequent questions they appear in;
- forceable facts: such facts are disabled and show their forced value;
- skippable questions: such questions are automatically answered;
- automatic completion: upon request the system can automatically complete the configuration process whenever all the mandatory facts have been answered and default values can be used for the remaining ones.
- question rollback: each answered question can be rolled back to the state before the answer.

The rollback operation was implemented with the purpose of preserving the answering order. When a question is rolled back, the current state is set to the one before answering the question, and hence, all the questions that were answered thereafter are rolled back too. Moreover, a fact occurring in multiple questions is kept forced to the value it was set the first time, until all its questions are rolled back.

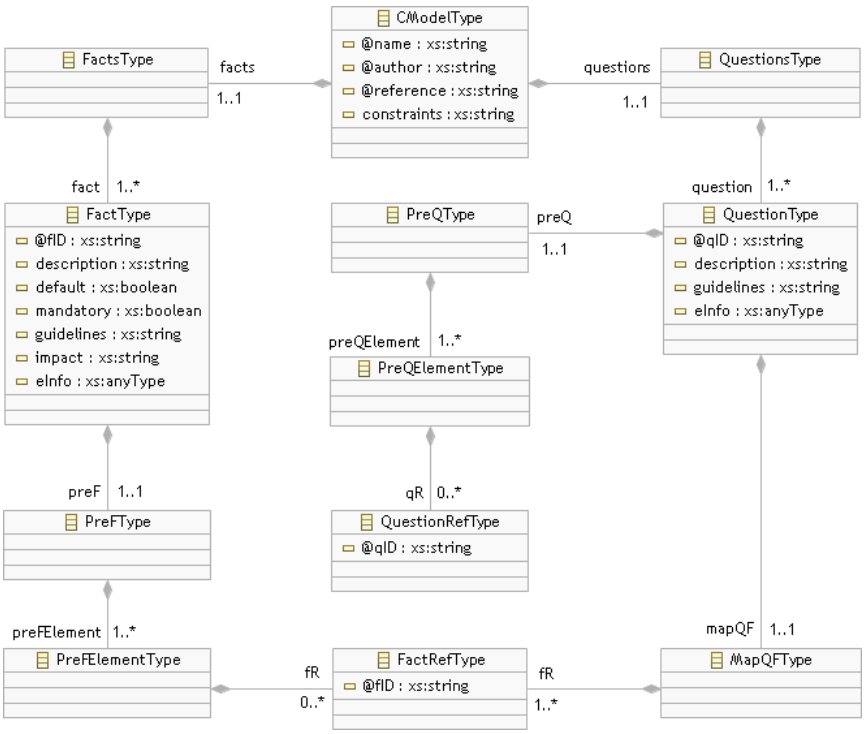

Fig. 2. The UML representation of the XML schema for the input file.

1.1 Configuration Example

This section shows a sample configuration process for an order fulfilment process model. The model is extracted from the Voluntary Inter-industry Commerce Standard (VICS) reference process model. VICS is an industry standard endorsed and used by various large companies that interact with suppliers and logistics providers by means of Electronic Data Interchange (EDI) transactions. This process model includes a number of variability points since it is intended to be adapted by user organizations in order to fit their individual requirements.

1.2 The Order Fulfilment Process Model

The order fulfillment process involves three roles: Supplier, Buyer and Carrier, and may support one or more business functions among Product Merchandising, Ordering, Logistics and Payment. Logistics may comprise one or more sub-phases among Freight Tender, Carrier Appointment, Freight in Transit and Freight Delivered. These phases range over the whole logistics sub-process, from making an offer to a Carrier (Freight Tender), through agreeing on the freight pick-up and delivery details (Carrier Appointment) and on the messages to be

exchanged during the shipment (Freight in Transit), to the types of claims to be supported after the delivery (Freight Delivered).

The planned usage of a Carrier's supplied trailer can also be decided upon, and thus configured, based on the size of the freight being shipped. It can be "Truckload" (TL) for full usage, "Less-than Truckload" (LTL) for partial usage, or "Small Package" (SP) when just single packages are to be shipped. This choice has a strong influence on subsequent decisions. For TL- or LTL-shipments, the roles responsible for fixing the Pickup and the Delivery appointments can be decided, provided Carrier Appointment is included in Logistics. For the pickup, this role can be played by either the Supplier or the Carrier; for the delivery, by either the Buyer or the Carrier. The appointment negotiation is not allowed in case of SP shipments, as the dates of pickup and delivery are imposed by the Carrier.

The Carrier's usage also affects the type of notifications to be sent during the transit, if Freight in Transit is included in Logistics. For TL or LTL, a Supplier's or Buyer's inquiry to the Carrier is followed by a shipment-status message for each parcel of the freight, whilst for SP the inquiry is followed only by one package-status message. Also, only in case of TL or LTL, and if Payment is selected, the Carrier can support a module for charging incidental costs that may be incurred during the transit.

Finally, in Freight Delivered, Claims support can be configured, in order to handle a Merchandise Return and/or cases of Freight Lost or Damaged. If the latter type of claim has been selected, then the Claim Manager is to be chosen between the Supplier and the Buyer.

The questionnaire model for the order fulfilment reference process model is depicted in Figure 3.

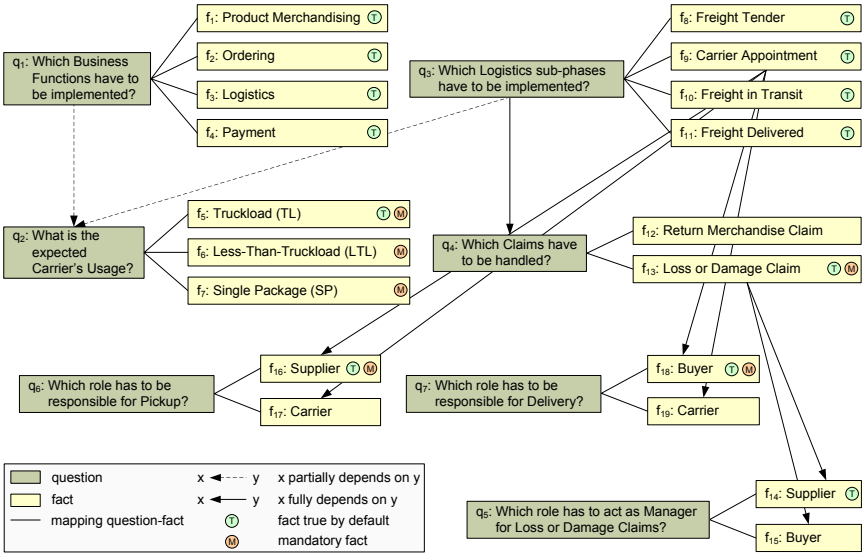

Fig. 3. The questionnaire model for the order fulfilment reference process model.

For example, facts f_1 to f_4 refer to the four business functions the process can implement. These facts are grouped in question q_1 that asks for the business functions to be implemented. Question q_2 groups the facts relating to the expected Carrier's usage. Since this choice is rather important as it affects the process overall, these facts are mandatory (labeled with a Ⓜ in the picture), so that they have to be explicitly set to $true$ or $false$ when answering q_2. Other questions would allow users to choose the roles responsible for Pickup and Delivery (q_6, q_7), the Claims to be handled (q_4) and the Manager for Loss or Damage Claims (q_5).

Default values have been assigned to the facts of Figure 3 (a Ⓣ indicates a fact whose default=$true$, while no symbol means that default=$false$). Selecting the default values leads to a VICS process that implements all the business functions ($f_1, f_2, f_3, f_4 = true$) and all the Logistics' sub-phases ($f_8, f_9, f_{10}, f_{11} = true$), and that supports TL shipments ($f_5 = true$, $f_6, f_7 = false$). In this type of shipment, the Supplier is usually responsible for organizing and scheduling the Pickup (so $f_{16} = true$ and $f_{17} = false$) while the Buyer is responsible for organizing and scheduling the Delivery ($f_{18} = true$, $f_{19} = false$). The process handles only Loss or Damage Claims (thus $f_{12} = false$ and $f_{13} = true$), managed by the Supplier which acts as intermediary between the Buyer and the Carrier ($f_{14} = true$, $f_{15} = false$).

The following boolean expressions have been used to constrain the values of the facts for the order fulfilment process model:[2]

C_1: $f_1 \vee f_2 \vee f_3 \vee f_4$
C_2: $f_3 \Leftrightarrow (f_8 \vee f_9 \vee f_{10} \vee f_{11})$
C_3: $(f_5 \veebar f_6 \veebar f_7) \Leftrightarrow (f_4 \vee f_9 \vee f_{10})$
C_4: $(f_{12} \vee f_{13}) \Rightarrow f_{11}$
C_5: $\neg(f_5 \vee f_6 \vee f_7) \Leftrightarrow \neg(f_4 \vee f_9 \vee f_{10})$
C_6: $f_{13} \Leftrightarrow (f_{14} \veebar f_{15})$
C_7: $(f_9 \wedge \neg f_7) \Leftrightarrow ((f_{16} \veebar f_{17}) \wedge (f_{18} \veebar f_{19}))$
C_8: $\neg f_{13} \Leftrightarrow \neg(f_{14} \vee f_{15})$
C_9: $\neg(f_9 \wedge \neg f_7) \Leftrightarrow \neg(f_{16} \vee f_{17} \vee f_{18} \vee f_{19})$.

C_1 ensures that at least one business function is chosen in q_1. C_3 and C_5 state that exactly one type of shipment is to be selected as Carrier's usage in q_2, if and only if at least one phase among Payment, Carrier Appointment and Freight in Transit is selected in q_3, otherwise no shipment type can be chosen. Indeed, as mentioned before, TL, LTL and SP affect the above process phases, so it makes no sense to decide on the shipment type unless a phase that is affected by the Carrier's Usage is selected. Likewise, as per C_7 and C_9, exactly one role between Supplier and Carrier is to be responsible for Pickup (q_6), and exactly one role between Buyer and Carrier is to be responsible for Delivery (q_7), if and only if Carrier Appointment is selected and one of TL and LTL is $true$. This is because the Pickup and Delivery appointments are handled during the Carrier Appointment phase of the VICS process and only in case of TL- or LTL-shipments.

[2] $\veebar$ indicates the exclusive disjunction (XOR), i.e., $f_1 \veebar f_2 \Leftrightarrow (f_1 \vee f_2) \wedge (f_1 \neq f_2)$.

Figure 4 shows an overview of the order fulfillment process model.[3] The model has been divided into a set of configurable process fragments, where fragments are delimited by dashed boxes and identified by the facts of Figure 3.

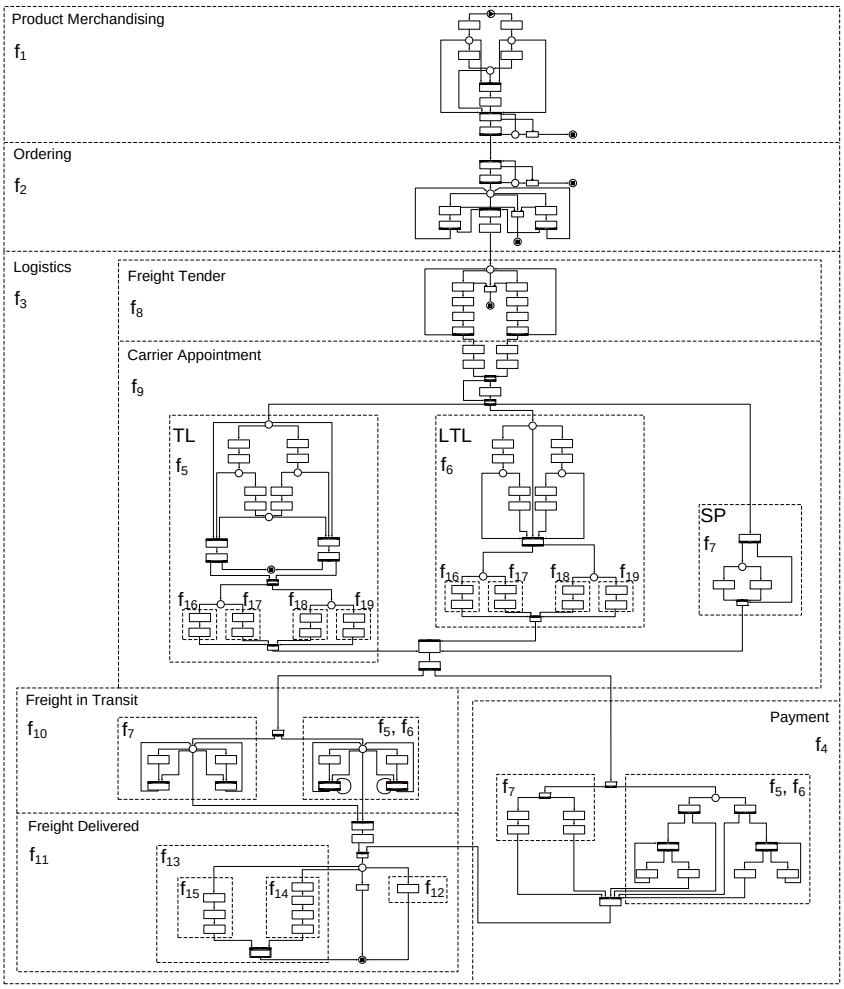

Fig. 4. The order fulfillment process model associated to the facts of Figure 3.

The four main process fragments refer to the Business Functions – Product Merchandise, Ordering, Logistics and Payment – that the process can implement. As such, their boxes encompass all the other configurable fragments. For example, Logistics (box "f_3") contains the

[3] A full representation of this process using the YAWL notation [1] can be found at `http://www.fit.qut.edu.au/~dumas/ConfigurationTool.zip`

fragments for its sub-phases, i.e. Freight Tender ("f_8"), Carrier Appointment ("f_9"), Freight in Transit ("f_{10}") and Freight Delivered ("f_{11}"). By associating to each of these facts an action that corresponds to the removal of the affected process fragments, setting f_3 to $false$ would imply to remove Logistics as well as all the fragments therein. This complies with C_2, which has been built to reflect this 'parent-child' relation that Logistics holds with its sub-phases.

Carrier Appointment, in turn, includes a fragment for handling each type of shipment ("f_5","f_6","f_7") and each role that can be responsible for Pickup (boxes "f_{16}" and "f_{17}") and for Delivery (boxes "f_{18}" and "f_{19}"). The last four fragments occur only within the boxes for "f_5" and "f_6", as only for TL- or LTL-shipments the Pickup and Delivery details can be decided. Since all the above facts are mapped to fragments within Logistics, if at least one of them is chosen in the configuration process, then Logistics cannot be removed anymore (i.e. f_3 must be set to $true$). At the level of facts, these interactions are described by constraints C_3, C_5, C_7 and C_9. Similar considerations hold for the remaining process fragments and constraints.

1.3 Configuring the Order fulfilment process model with Quaestio

For convenience, we introduce the notation $a_m^{q_n}$ to indicate the valuation that is given by the answer m to the facts of question n (where the remaining facts that are not set by the answer are left out). Also, the symbols T and F are shorts for a *true*, resp. for a *false*, valuation.

Assume, for example, that we want to configure the model to handle SP shipments and to support only Loss or Damage Claims managed by the Supplier, and that we are not interested in the Payment phase of the process as it will be outsourced. These can be common choices among the stakeholders of a supply-chain management company interested in implementing the VICS EDI Framework.

Once the corresponding configuration model has been loaded into Quaestio, the valid questions are shown in the Valid Questions list. These are q_1 and q_3, since they have no dependencies (Figure 5). The initial state is s_1 where no answers have been given, i.e. $qs(s_1) = \varnothing$. We decide, for example, to answer q_3 – *Which Logistics phases have to be implemented?* with its default answer. This corresponds to giving answer $a_1^{q_3} = \{(f_8, \mathsf{T}), (f_9, \mathsf{T}), (f_{10}, \mathsf{T}), (f_{11}, \mathsf{T})\}$, since all the facts of q_3 are $true$ by default (shown by a green Ⓣ next to the fact's description).

With a_1 we reach state s_2 with $qs(s_2) = \{q_3\}$. q_2 is added to the valid questions due to its partial dependency on q_1 or q_3. Assume we choose q_1 from the Valid Questions. From the Question Inspector we can see that f_3 has been forced to $true$ and has been grayed out (Figure 6). The system has reacted to a_1 by setting f_3 in order to comply with C_2. We answer q_1 with $a_2^{q_1} = \{(f_1, \mathsf{T}), (f_2, \mathsf{T}), (f_3, \mathsf{T}), (f_4, \mathsf{F})\}$ so as to exclude Payment.

After a_2, we reach s_3 with $qs(s_3) = \{q_3, q_1\}$. Questions q_4, q_6 and q_7 are added to the valid ones as they depend on q_3. Assume we pick q_2 – *What is the expected Carrier's Usage?*. Due to C_3 and to the answers given so far, this question can only be answered if exactly one

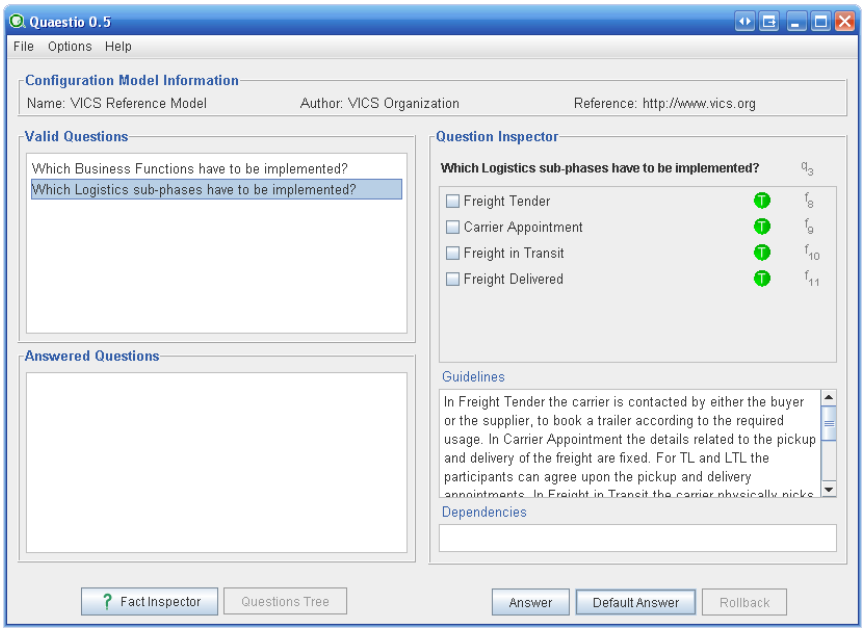

Fig. 5. State s_1: the only valid questions are q_1 and q_3.

of its facts is set to *true* (the answer button is disabled). Also, this question needs to be explicitly answered as all its facts are mandatory (indicated by a red Ⓜ next to the fact's description). We select Single Package and $a_3^{q_2} = \{(f_5, \mathsf{F}), (f_6, \mathsf{F}), (f_7, \mathsf{T})\}$ is given.

The next state is s_4 with $qs(s_4) = \{q_3, q_1, q_2\}$. Although no questions depend on q_2, after answering a_3 both q_6 and q_7 become skippable, since all their facts can take only value *false* due to C_9. Thus $a_4^{q_6} = \{(f_{16}, \mathsf{F}), (f_{17}, \mathsf{F})\}$ and $a_5^{q_7} = \{(f_{18}, \mathsf{F}), (f_{19}, \mathsf{F})\}$ are automatically given by the system, which moves from s_4 to s_5 with a_5, and from s_5 to s_6 with a_6. q_6 and q_7 are added to the set of answered ones (shown in gray in Figure 7) and $qs(s_6) = \{q_3, q_1, q_2, q_6, q_7\}$. Next we answer the only valid question remaining, q_4 – *Which claims have to be handled?*, with its default answer $a_6^{q_4} = \{(f_{12}, \mathsf{F}), (f_{13}, \mathsf{T})\}$ as it complies with our requirements.

After a_6 we reach s_7 with $qs(s_7) = \{q_3, q_1, q_2, q_6, q_7, q_4\}$. q_5 – *Which role has to act as Manager for Loss or Damage Claims?* is now valid as it depends on q_4. s_7 is a final state as all the mandatory facts have already been set and the remaining ones still unset (f_{14} and f_{15}) can take their defaults without violating the constraints. q_5 can thus be answered automatically with defaults. At this point users can decide whether to continue or to complete the configuration automatically. We decide to use the automatic completion and answer $a_7^{q_5} = \{(f_{14}, \mathsf{T}), (f_{15}, \mathsf{F})\}$ is given.

State s_8 is the next state with $qs(s_8) = \{q_3, q_1, q_2, q_6, q_7, q_4, q_5\}$. Assume that now we want to change q_4 in order to support only Return Merchandise Claims. In this case we can

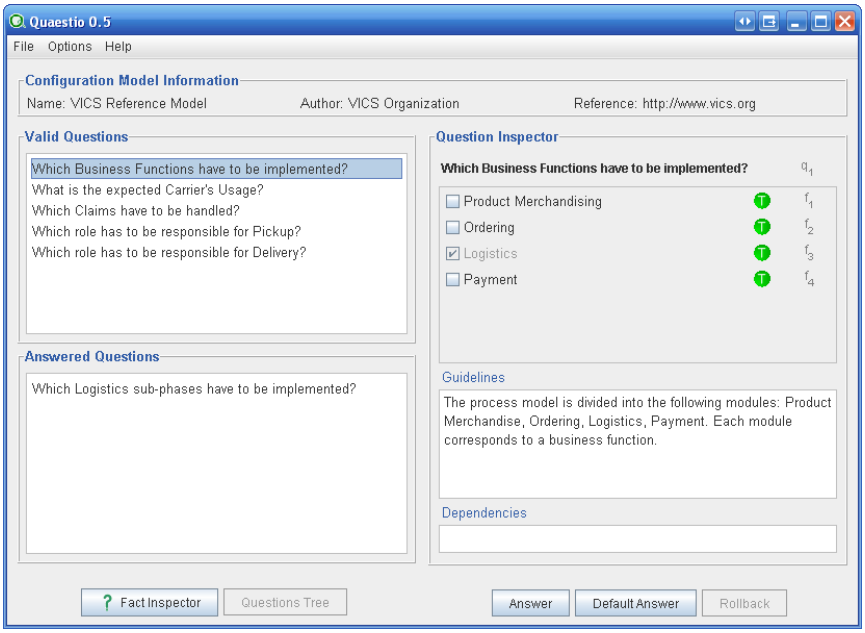

Fig. 6. State s_2: f_3 has been forced to $true$ in q_1 in order not to violate C_2.

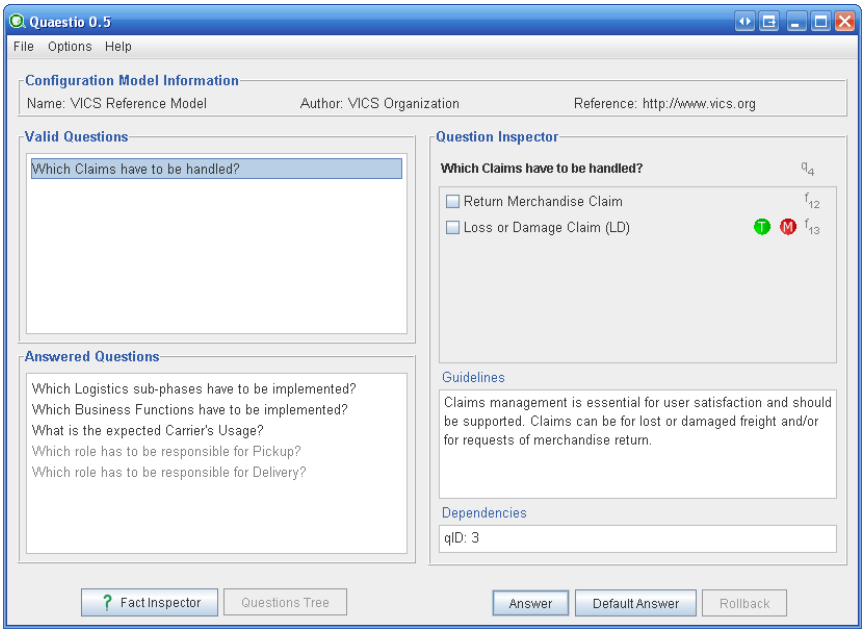

Fig. 7. State s_6: q_6 and q_7 have been skipped as their facts can only be negated.

rollback q_4 and re-answer it. The system restores the current state to s_6, i.e. the state before answering q_4. We then answer $a_6^{q_4} = \{(f_{12}, \mathsf{T}), (f_{13}, \mathsf{F})\}$ and reach s_7 again. This time, though, q_5 is skippable since a Manager can be chosen only for Loss or Damage Claims. The only valid answer is $a_7^{q_5} = \{(f_{14}, \mathsf{F}), (f_{15}, \mathsf{F})\}$. With this we reach s_8 and complete.

The corresponding configuration trace is $\sigma = \{(s_1, (a_1, q_3), s_2), (s_2, (a_2, q_1), s_3), (s_3, (a_3, q_2), s_4), (s_4, (a_4, q_6), s_5), (s_5, (a_5, q_7), s_6), (s_6, (a_6, q_4), s_7), (s_7, (a_7, q_5), s_8)\}$, and the configuration is $cf_\sigma = \{(f_1, \mathsf{T}), (f_2, \mathsf{T}), (f_3, \mathsf{T}), (f_4, \mathsf{F}), (f_5, \mathsf{F}), (f_6, \mathsf{F}), (f_7, \mathsf{T}), (f_8, \mathsf{T}), (f_9, \mathsf{T}), (f_{10}, \mathsf{T}), (f_{11}, \mathsf{T}), (f_{12}, \mathsf{T}), (f_{13}, \mathsf{F}), (f_{14}, \mathsf{F}), (f_{15}, \mathsf{F}), (f_{16}, \mathsf{F}), (f_{17}, \mathsf{F}), (f_{18}, \mathsf{F}), (f_{19}, \mathsf{F})\}$.

The above configuration leads to the configured order fulfillment process model pictured in Figure 8. This model is the result of performing certain actions on the initial model to remove the irrelevant process fragments, i.e. those process fragments whose facts have been set to false.

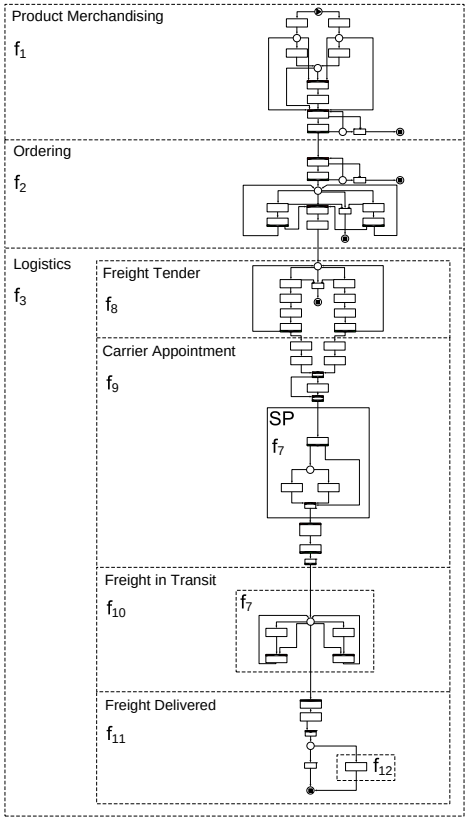

Fig. 8. The configured order fulfillment process model.

References

1. W. M. P. van der Aalst and A. H. M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.
2. M. La Rosa, W. M. P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede. Questionnaire-based Variability Modeling for System Configuration. 2008. Available at QUT ePrints, `http://eprints.qut.edu.au/archive/7992`.