

1. Examples

1.1 Influenza Vaccine Guideline Documentation

In this documentation, we will briefly describe the process to encode the influenza vaccine guideline developed by Advisory Committee on Immunization Practices (ACIP) (MMWR 2000;49(RR03):1-38) in GLIF. The URL for the guideline is: <http://www.cdc.gov/epo/mmwr/preview/mmwrhtml/rr4903a1.htm>.

Level A – Main Guideline

Data Collection – DOB, Contraindications and Number_Of_Previous_Influenza_Vaccine_Doses

We first need to collect data such as DOB, Contraindications and Number_Of_Previous_Influenza_Vaccine_Doses. These processes are represented as action steps in GLIF. As these action steps can be performed in any order, we enclose them within a branch-synchronization pair, as shown in Figure 1. The first step of the guideline is the Branch 1 step. As all the data are needed to make decisions in later steps, these action steps are synchronized with AND.

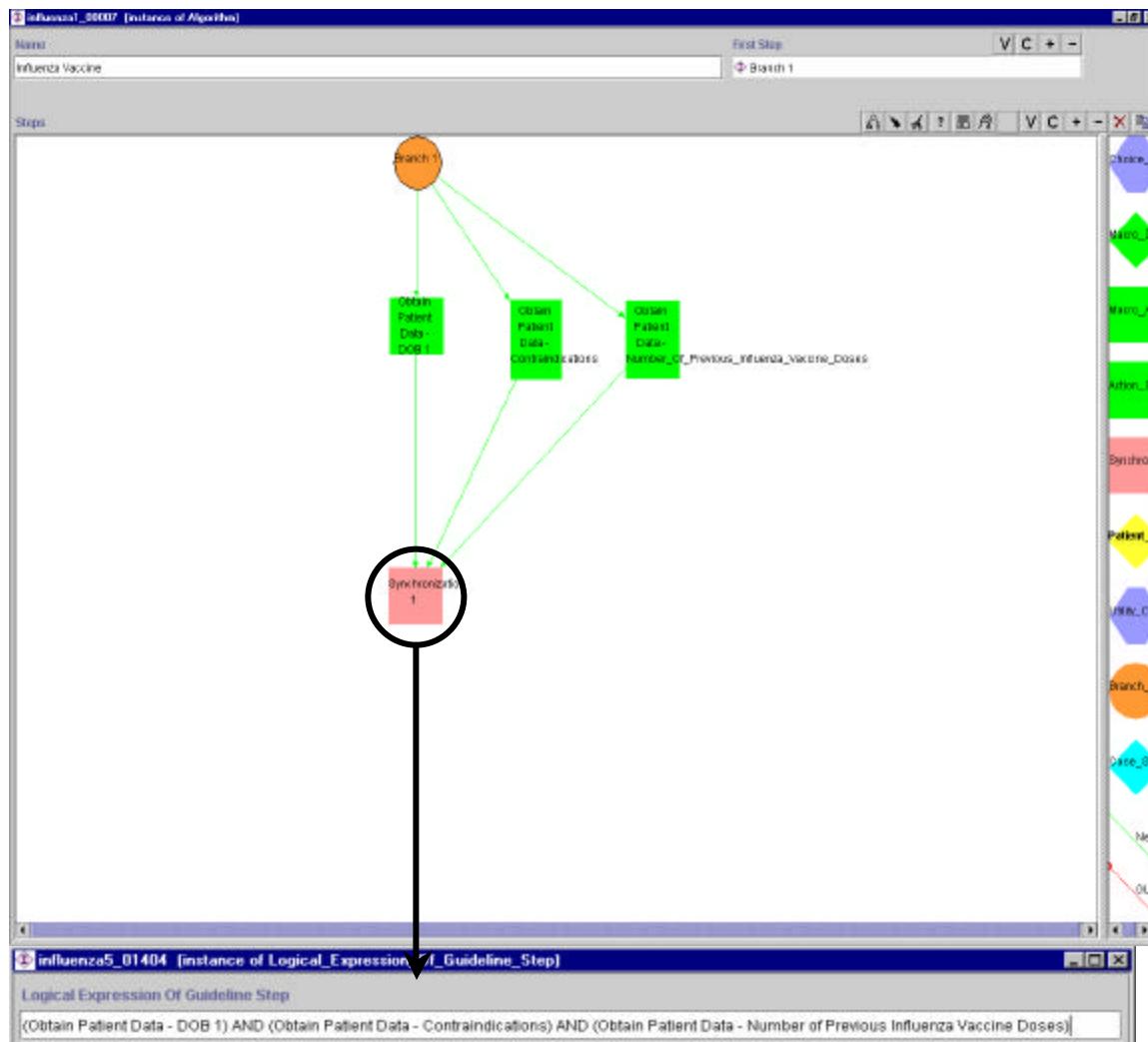


Figure 1. Data Collection Action Steps and Synchronization with AND

If an action step contains a complex process of decision making and other actions, we can encode it as a sub-guideline within that action step, as in the case of collecting contraindication data. We will describe sub-guideline encoding in later sections.

Data Collection – Groups at Increased Risk for Complications, Persons Who Can Transmit Influenza to Those at High Risk and Willing_To_Get_Influenza_Vaccine

There are some other patient data that need to be collected or decided, such as Groups at Increased Risk for Complications, Persons Who Can Transmit Influenza to Those at High Risk and Willing_To_Get_Influenza_Vaccine. Any one of these action steps can decide if a patient is potentially eligible for influenza vaccine and these action steps can be performed in any order. So we add a decision step after each data collection action step, enclose these action and decision steps within a branch-synchronization pair and synchronize them with *OR*, as shown in Figure 2.

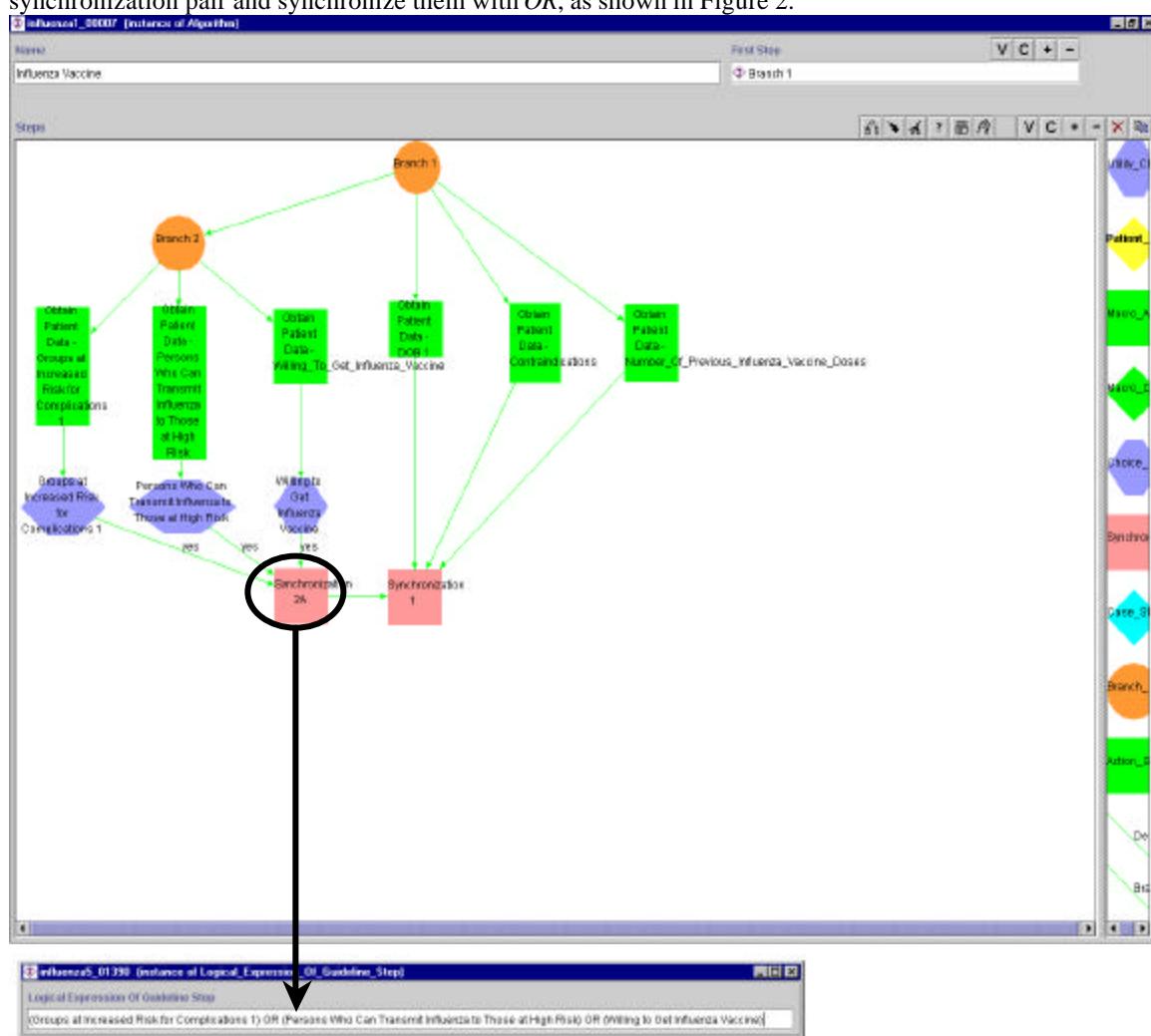


Figure 2. Data Collection Action Steps and Synchronization with OR

Data Collection – Shortcut to the End

In case we can know that a patient is not eligible for influenza vaccine, we may provide shortcuts to the end of the guideline, which is a patient state step. This is the case for patients not in groups at increased risk for complications, patients not persons who can transmit influenza to those at high risk and patients not willing to get influenza vaccine, as shown in Figure 3.

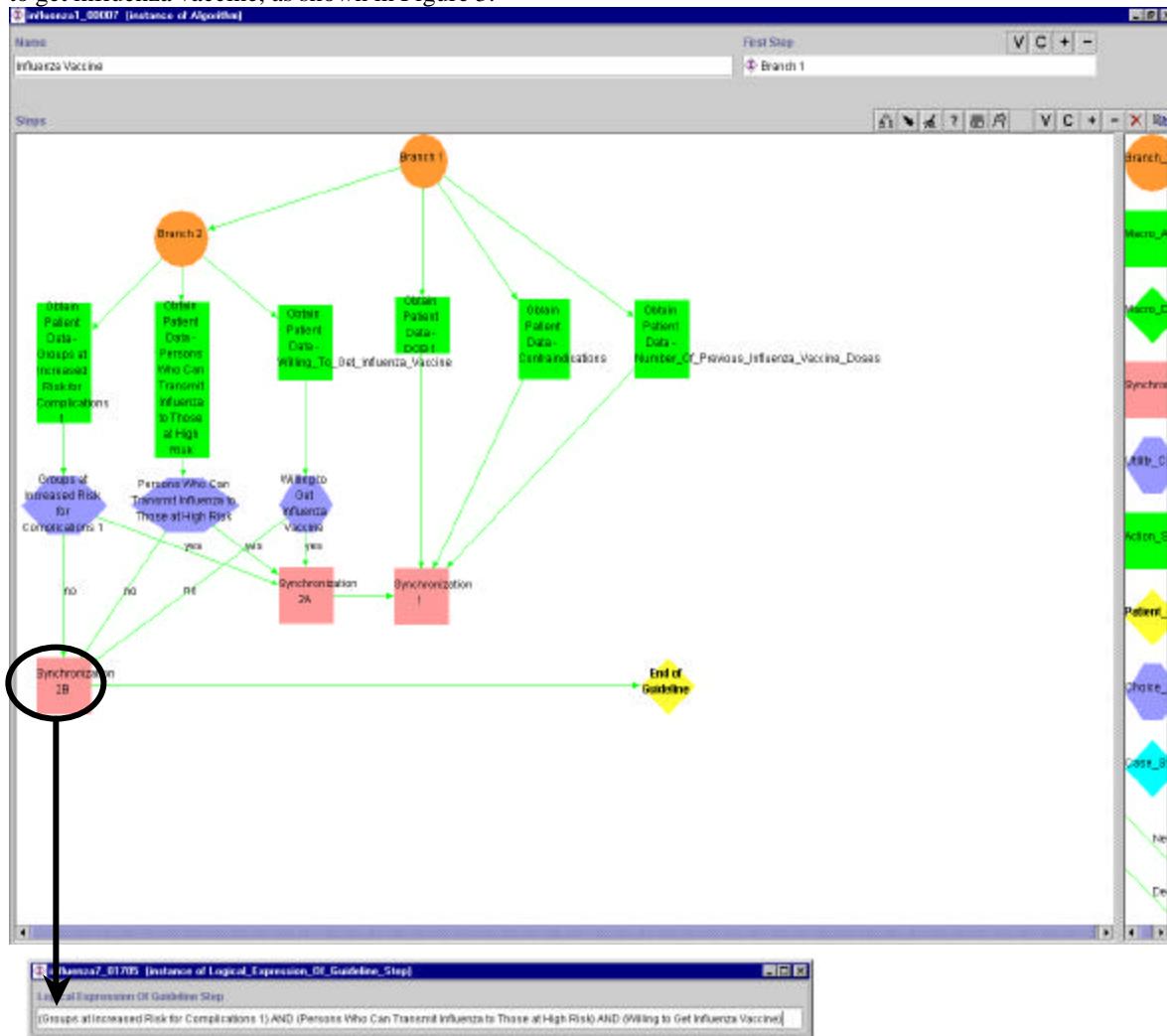


Figure 3. Shortcuts to the End of the Guideline

Now we can see, there are two synchronization steps, Synchronization 2A and Synchronization 2B, corresponding to Branch 2. In general, a branch step may correspond to multiple synchronization steps, as long as these synchronization steps are mutually exclusive.

Decision Making – The First Dose of Influenza Vaccine

After we get the initial patient data, we can make decisions about the first dose of influenza vaccine. There are five possibilities for this decision.

A patient may be eligible for the first dose of influenza vaccine, in which case we distinguish three possibilities, i.e., patient age between 6 months to 35 months, patient age between 3 years to 12 years, and patient age 13+ years, as vaccines for patients in these three age groups are different. The results of these decisions are to give influenza vaccine doses to patients, which are represented as action steps.

A patient may be potentially eligible for the second dose of influenza vaccine, in which case we need to make further decisions. The result of this decision is collect more data for further decision making.

Finally, a patient may be ineligible for any doses of influenza vaccine, in which case we reach the end of the guideline.

The whole process is shown in Figure 4.

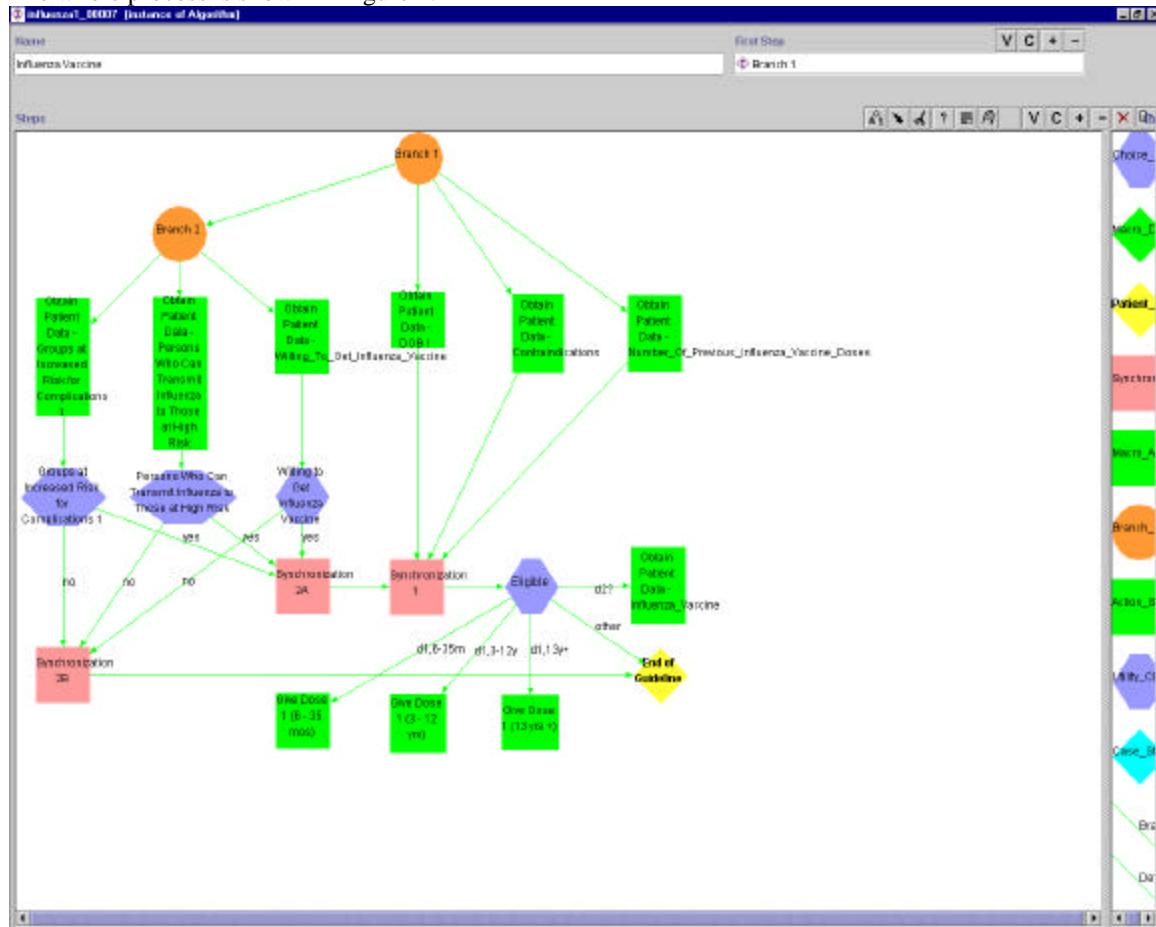


Figure 4. Decision Making – The First Dose of Influenza Vaccine

Decision Making – The Second Dose of Influenza Vaccine

For those patients potentially eligible for the second dose of influenza vaccine, we first need to collect more data – `Influenza_Vaccine`. Based on this and previously collected data, we can make the decision for the second dose of influenza vaccine.

Patients may be eligible for the second dose, in which case we distinguish two possibilities, i.e., patient age between 6 months to 35 months and patient age between 3 years to 8 years. For each possibility, we give correspondent vaccine dose to the patient, which is represented as an action step.

Patients may be ineligible for the second dose, in which case we will reach the end of the guideline. This process is shown in Figure 5.

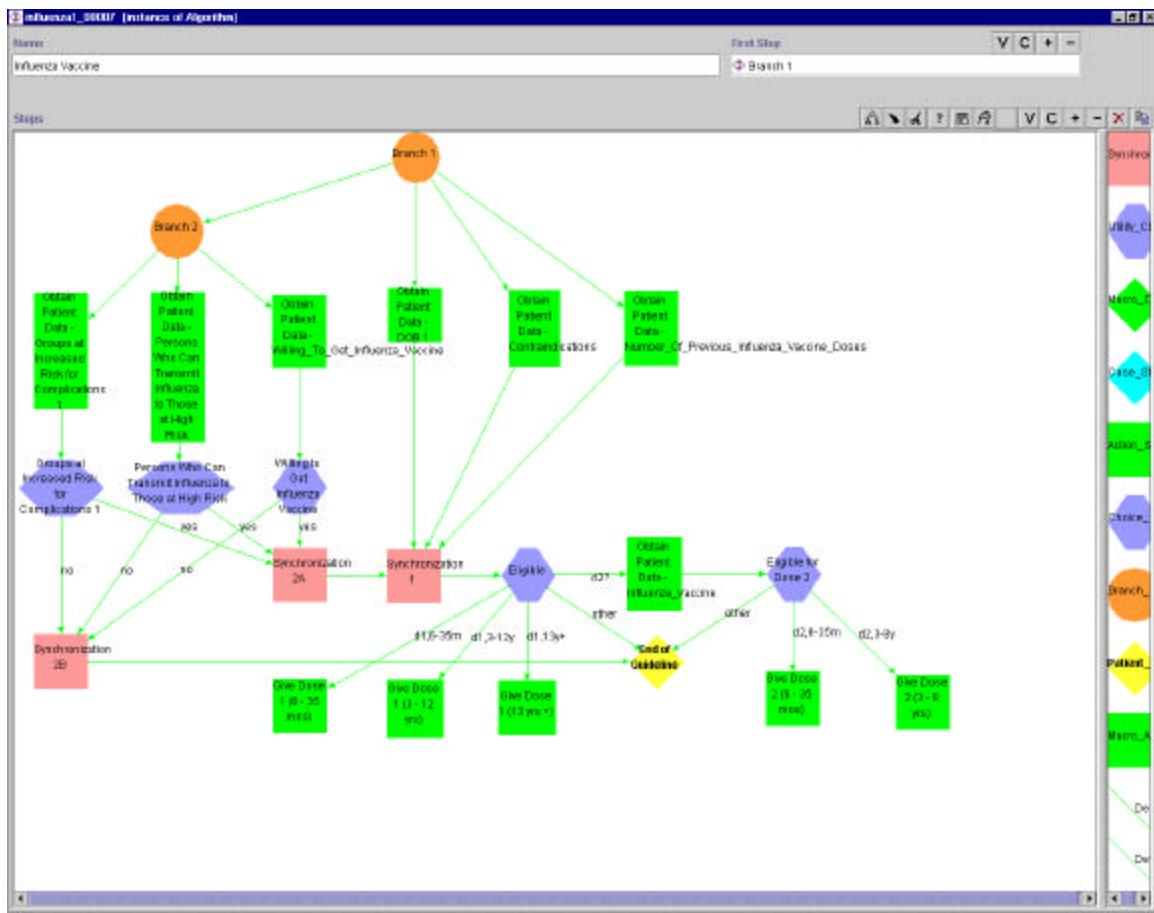


Figure 5. Decision Making – The Second Dose of Influenza Vaccine

Patient Education – Side Effects and Adverse Reactions

After patients are given the influenza vaccine dose, we need to provide patient education about the side effects and adverse reactions, which is represented as an action step.

After this patient education, we will reach the end of the guideline.

This process is shown in Figure 6.

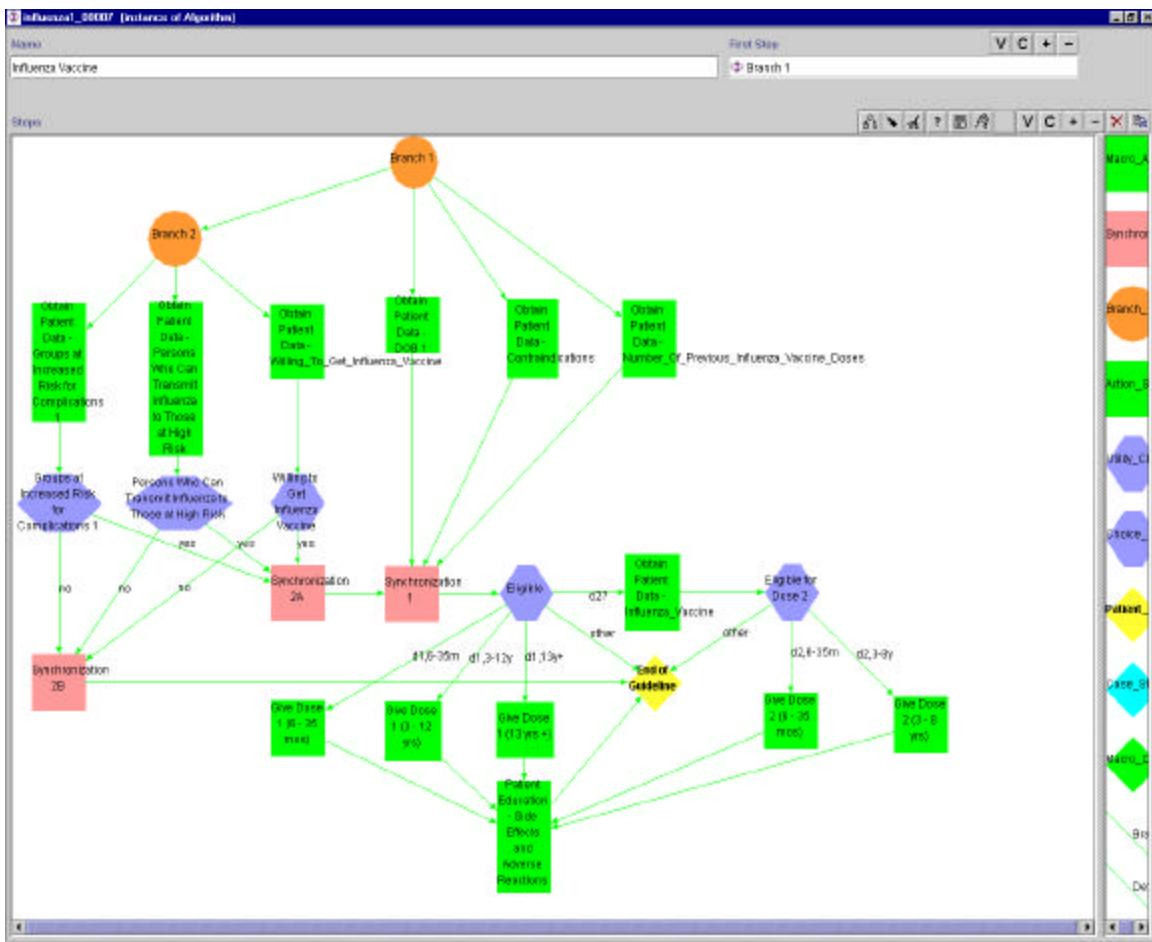


Figure 6. Patient Education – Side Effects and Adverse Reactions

Level A – Subguidelines

Subguideline – Collect Contraindication Data

Collect contraindication data is a complex process that contains a series of data collection, decision making and clinical action processes.

If a patient is in the first trimester of pregnancy, or if a patient is allergic to influenza vaccine and not in groups at increased risk for complications, or if a patient has acute febrile disease, or if a patient has Guillain-Barre Syndrome and not in groups at increased risk for complications, than the patient can be considered as having contraindication to the influenza vaccine. Otherwise, a patient is considered as not having contraindication to the influenza vaccine, with a special case for a patient allergic to influenza vaccine and in groups at increased risk for complications, when this patient should have allergy evaluation and desensitization.

For clarification of understanding and easiness of maintenance, we can encode these processes as a sub-guideline, as shown in Figure 7.

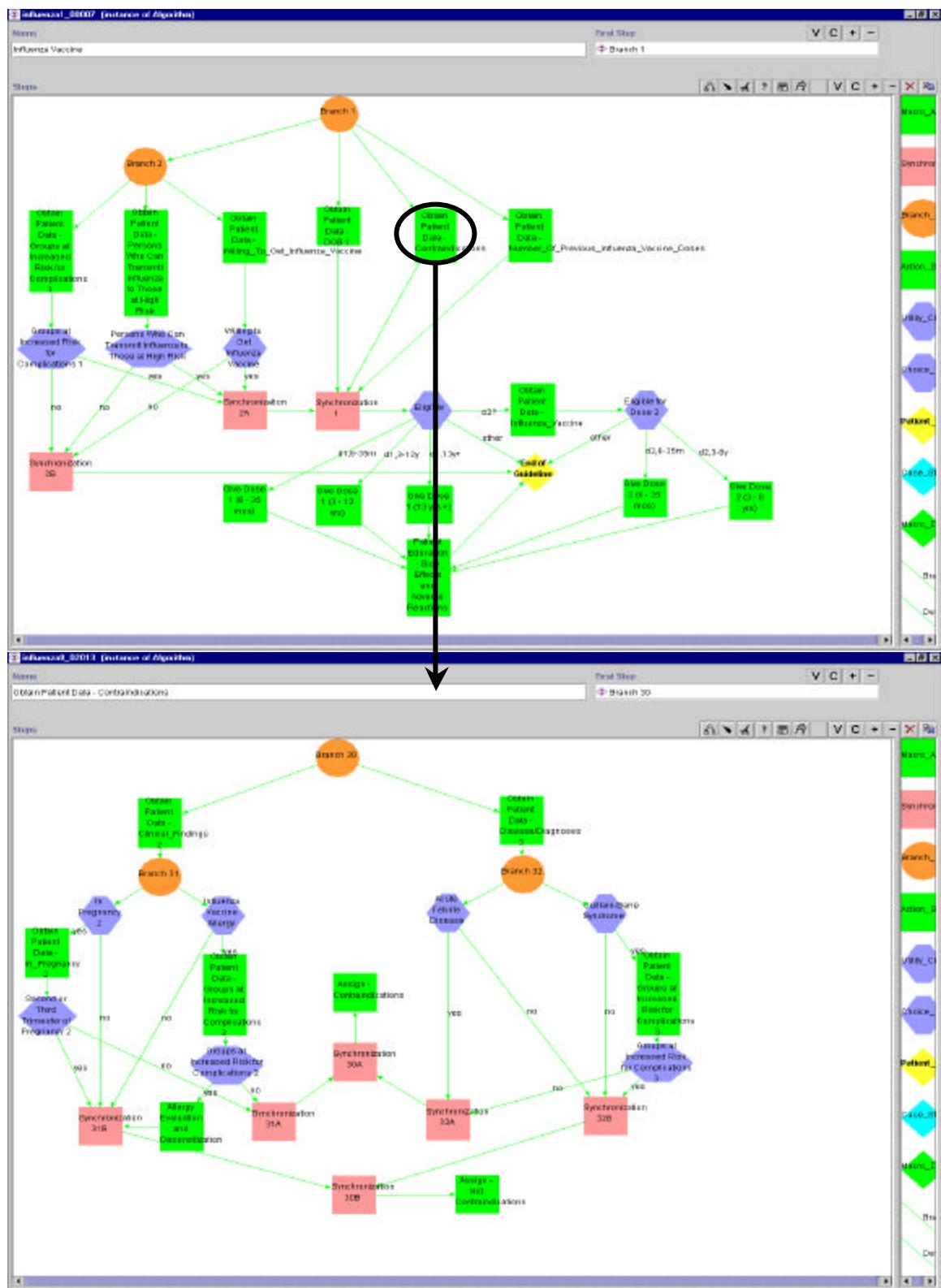


Figure 7. Subguideline – Collect Contraindication Data

Subguideline – Groups at Increased Risk for Complications

As in the case of contraindication data collection, the process to decide if a patient is in groups at increased risk for complications is very complex.

According to the influenza vaccine guideline, if a patient is more than 50 years old, or if a patient is a resident of nursing home, or if a patient has chronic disorder of pulmonary or cardiovascular systems, or if a patient has medical follow-up or hospitalization due to chronic metabolic diseases, renal dysfunction, hemoglobinopathies or immunosuppression, or if a patient is a child or teenager receiving long-term aspirin therapy, or if a patient is in the second or the third trimester of pregnancy, then this patient is considered to be in groups at increased risk for complications. Otherwise, the patient is not in groups at increased risk for complications.

We can encode this process as another subguideline, as shown in Figure 8.

Subguideline – Persons Who Can Transmit Influenza to Those at High Risk

Similarly, the process to decide if a patient is a person who can transmit influenza to those at high risk is rather complex.

If a patient's occupation is physician, nurse or hospital worker, or if a patient is a household member of persons in high risk groups, than that patient is a person who can transmit influenza to those at high risk. Otherwise, the patient is not a person who can transmit influenza to those at high risk.

We can encode this process as a sub-guideline, as shown in Figure 9.

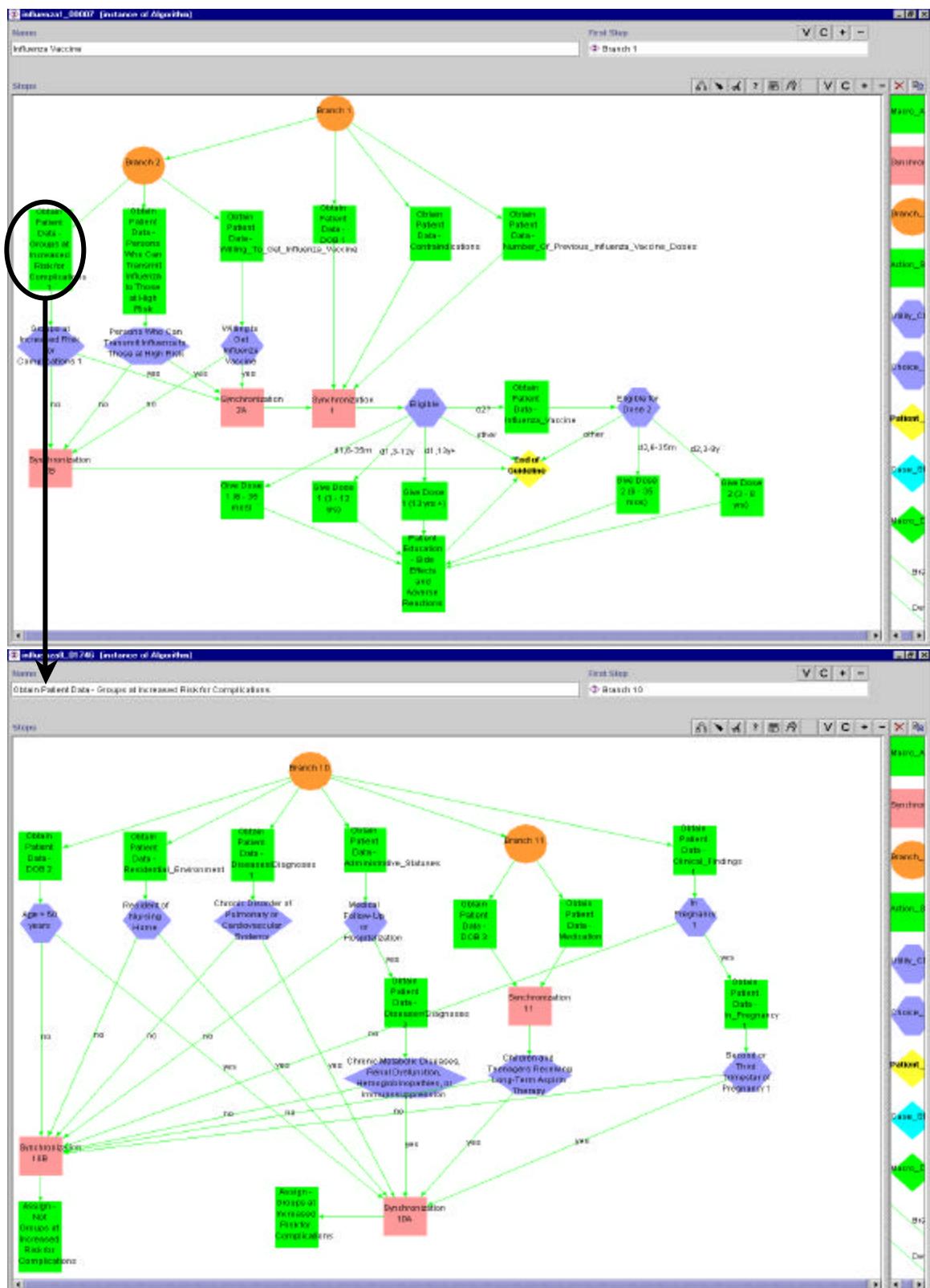


Figure 8. Subguideline – Groups at Increased Risk for Complications

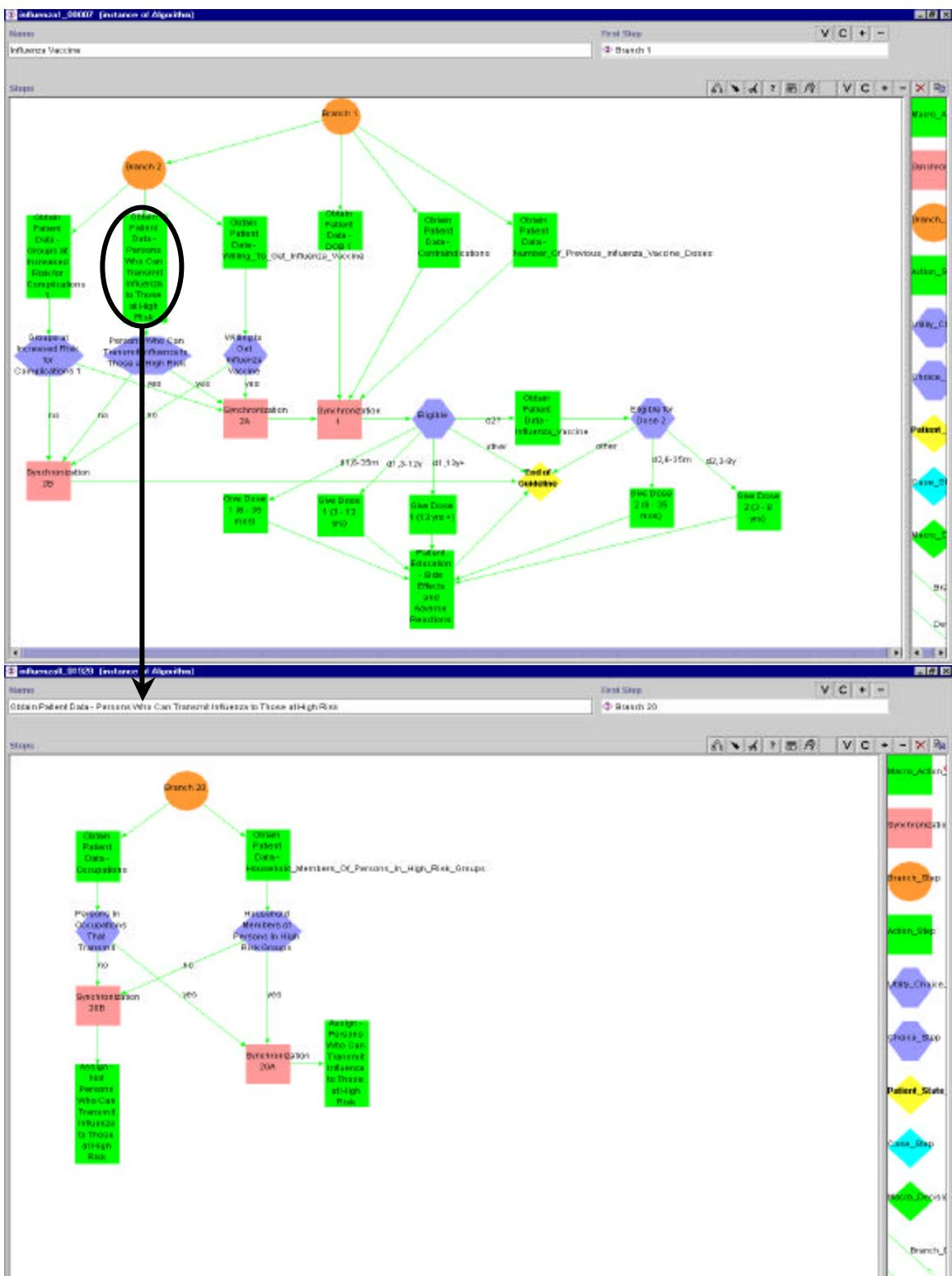


Figure 9. Subguideline – Persons Who Can Transmit Influenza to Those at High Risk

Level B

Branch Step

In a branch step, we need to define Order Constraint, Branches, Strength Of Evidence and Didactics. For example, in the Branch 1 Step, we define the Order Constraint as *any_order* and the Branches as *Branch 2*, *Obtain Patient Data – DOB 1*, *Obtain Patient Data – Contraindications* and *Obtain Patient Data – Number_Of_Previous_Influenza_Vaccine_Doses*.

The class definition of Branch_Step is shown in Figure 10 and the instance of Branch 1 Branch Step is shown in Figure 11.

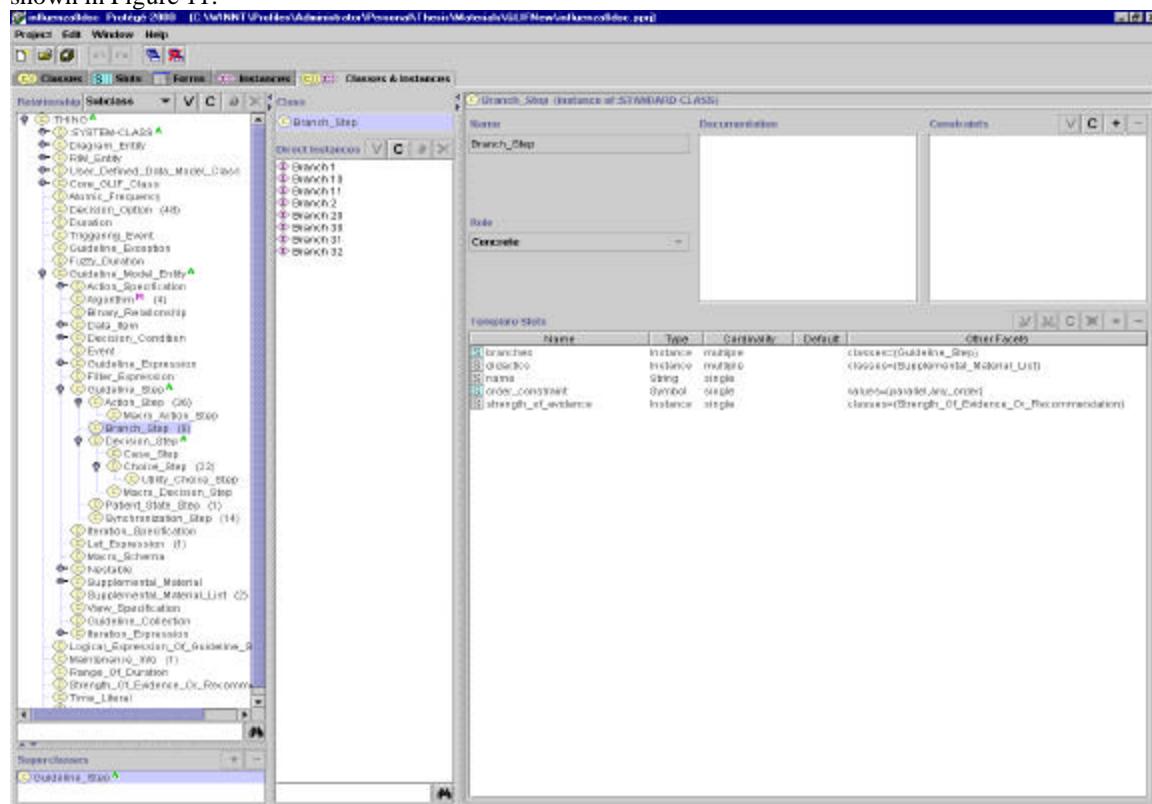


Figure 10. Class Definition – Branch Step

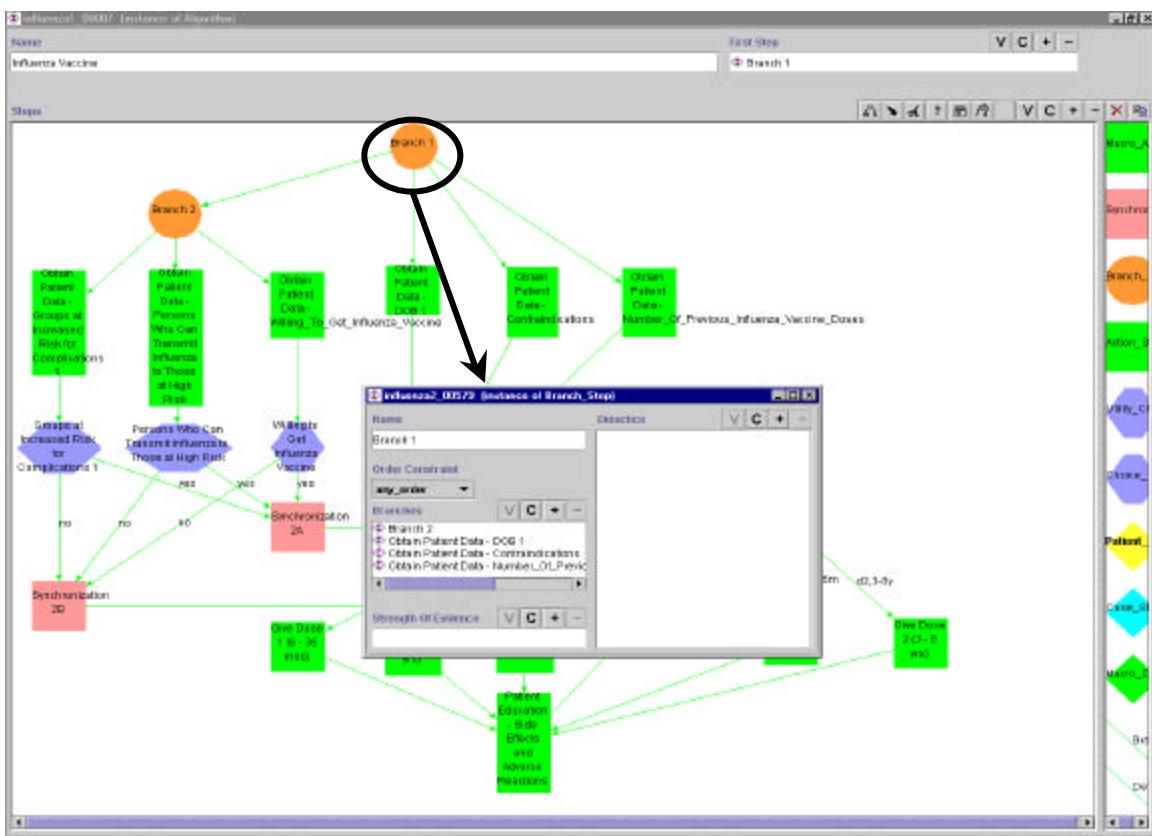


Figure 11. Branch Step Instance – Branch 1

Synchronization Step

In a synchronization step, we need to define Continuation, Next Step, Strength Of Evidence and Didactics. For example, in the Synchronization 1 Step, we define the Continuation as (*Synchronization 2A*) AND (*Obtain Patient Data – DOB 1*) AND (*Obtain Patient Data – Contraindications*) AND (*Obtain Patient Data – Number of Previous Influenza Vaccine Doses*) and the Next Step as *Eligible*.

The class definition of *Synchronization_Step* is shown in Figure 12 and the instance of *Synchronization 1* Synchronization Step is shown in Figure 13.

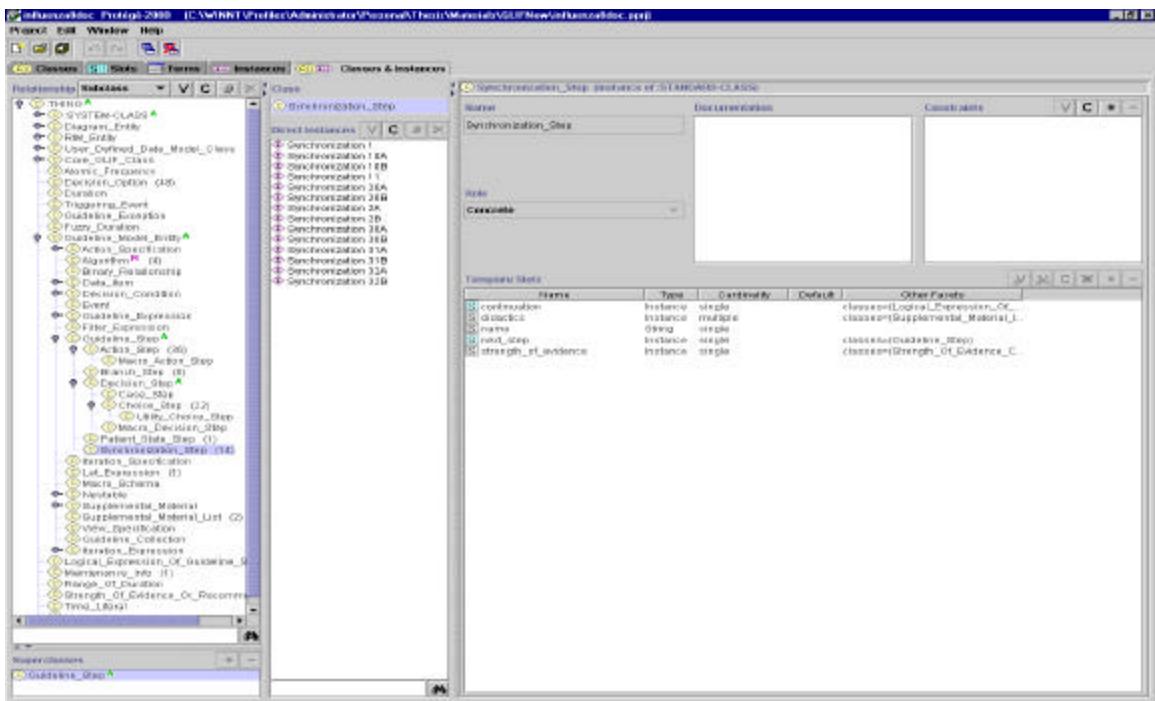


Figure 12. Class Definition – Synchronization Step

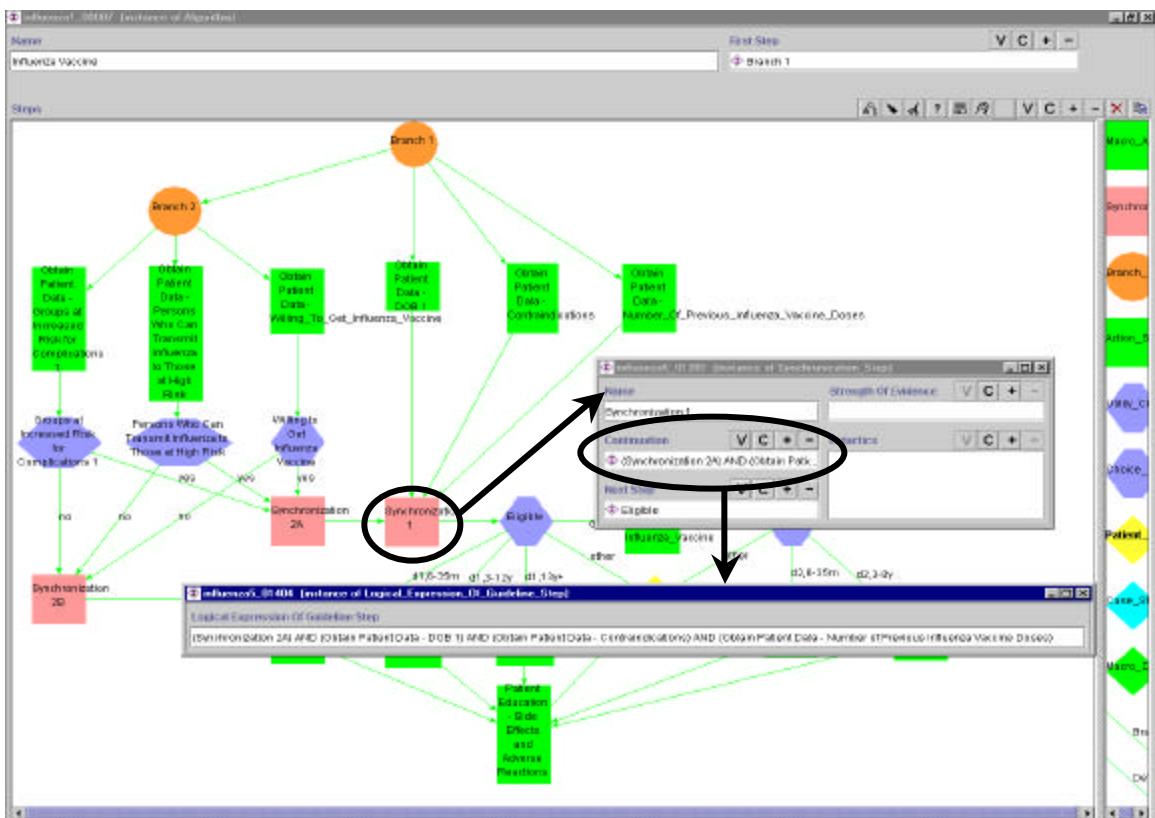


Figure 13. Synchronization Step Instance – Synchronization 1

Action Step

In an action step we need to define Tasks, Next Step, Iteration Info, Triggering Events, etc. For example, in the Obtain Patient Data – DOB 1 Step, we define the Tasks as *Obtain Patient Data – DOB* and the Next Step as *Synchronization 1*.

The values of Tasks are instances of Action_Specification class, for which we need to define Description, Intention, etc. For example, in the Obtain Patient DOB Action_Specification, which is an instance of Get_Data_Action (a subclass of Action_Specification class), we define the Var Data Item, another slot of a Get_Data_Action, as *DOB*.

Similarly, in DOB Variable_Data_Item, we define Concept Id as *C0421451*, Concept Source Id as *UMLS*, Data Model Source Id as *USAM* and Data Value as *DOB*.

The class ontology of Action Step, Action_Specification and Variable_Data_Item, and class definition of Get_Data_Action are shown in Figure 14. The instances of Obtain Patient Data – DOB 1 Action Step, Obtain Patient Data – DOB Get_Data_Action and DOB Var_Data_Item are shown in Figure 15.

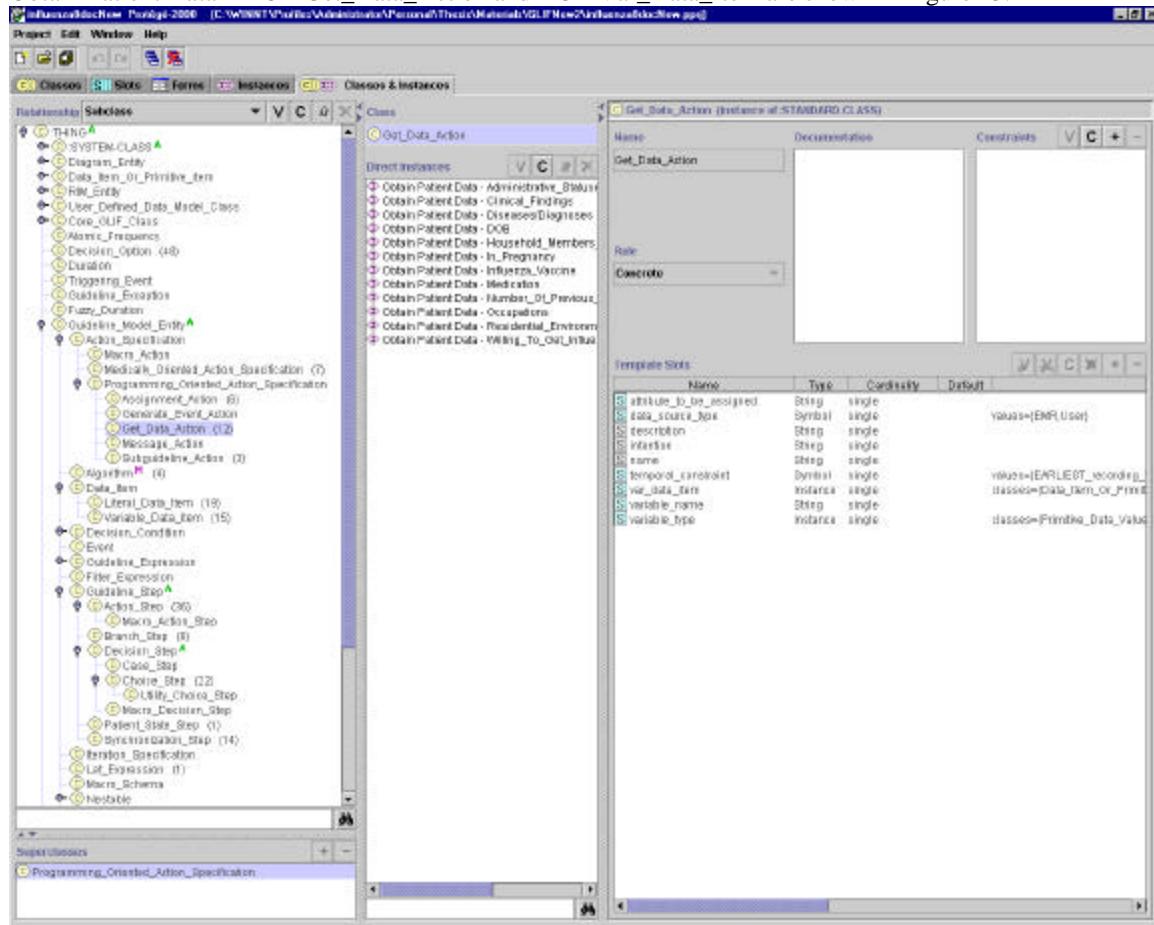


Figure 14. Class Ontology – Action Step, Action_Specification, Variable_Data_Item and Class Definition – Get_Data_Action

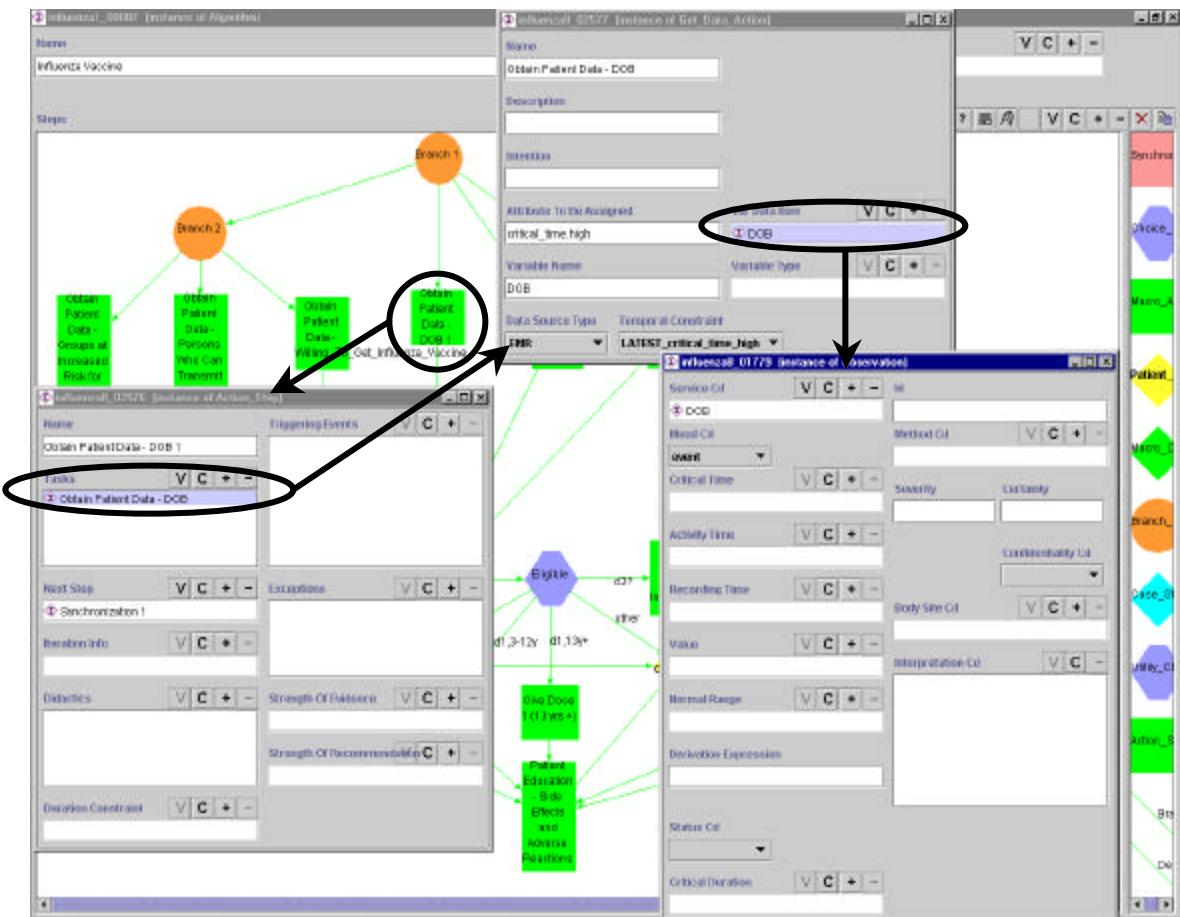


Figure 15. Action Step Instance – Obtain Patient Data – DOB 1, Get_Data_Action Instance – Obtain Patient Data – DOB, and Variable_Data_Item Instance – DOB

From the class ontology we can see there are several types of Action_Specification. One of the subclasses of Action_Specification is Subguideline_Action, which is used for subguideline definition. In Subguideline_Action class, Action Detail slot is used to define the subguideline. For example, in the Obtain Patient Data – Groups at Increased Risk for Complications Action_Specification, we define the Action Detail as *Obtain Patient Data – Groups at Increased Risk for Complications*, which is a subguideline, as shown in Figure 16.

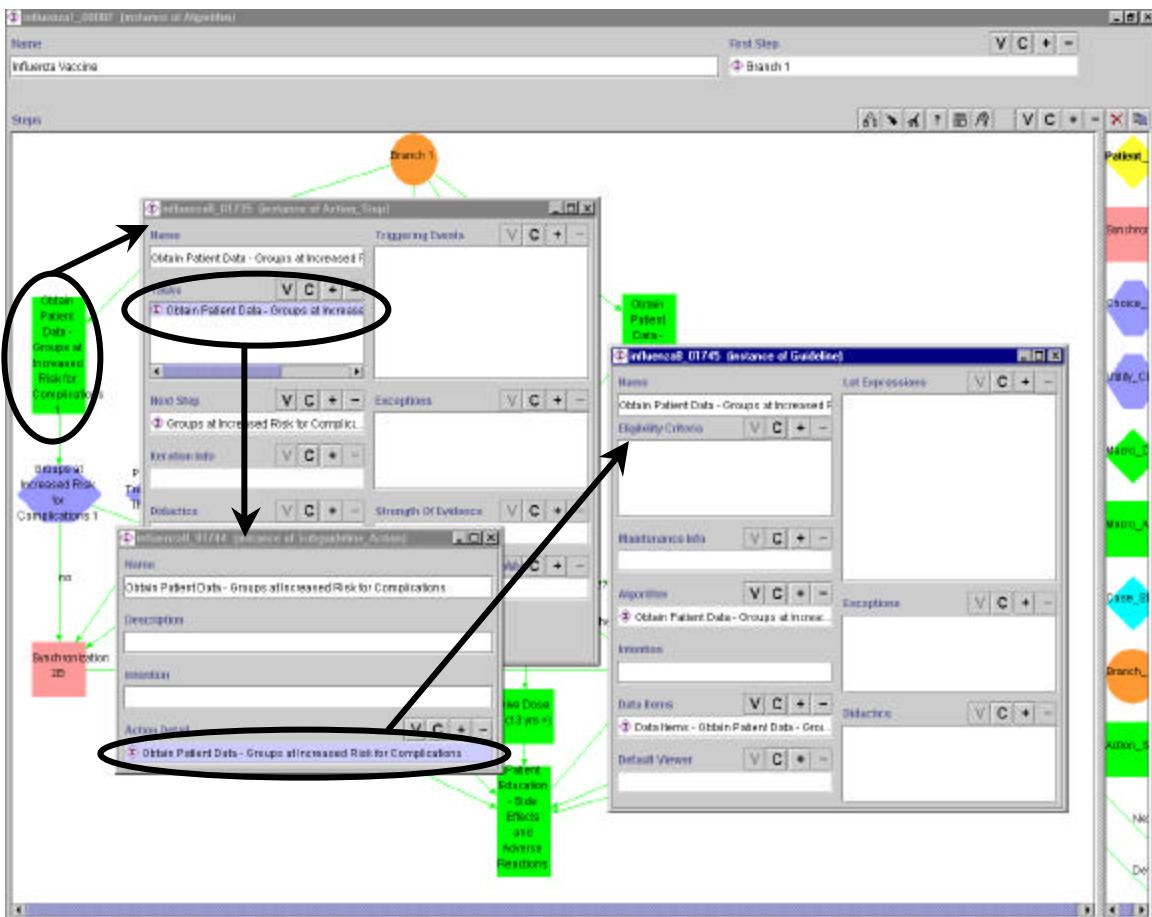


Figure 16. Subguideline_Action Instance – Obtain Patient Data – Groups at Increased Risk for Complications

Decision Step

In a decision step we need to define Options, Decision Details, Triggering Events, etc. For example, in the Willing to Get Influenza Vaccine Step, we define the Options as Willing to Get Influenza Vaccine and Not Willing to Get Influenza Vaccine.

The values of Options are instances of Decision_Option class, for which we need to define Condition_Value, Destination, etc. For example, in the Willing to Get Influenza Vaccine Decision_Option, we define the Condition_Value as Willing to Get Influenza Vaccine and Destination as Synchronization 2A.

Similarly, in Willing to Get Influenza Vaccine RuleInChoice, we define Rule In as *Willing to Get Influenza Vaccine*; in Willing to Get Influenza Vaccine Three_Value_Criterion, we define Specification as *Willing_to_Get_Influenza_Vaccine == true*.

The class ontology of Decision Step, Decision_Option, RuleInChoice and Three_Value_Criterion, and class definition of Decicition_Option are shown in Figure 17. The instances of Willing to Get Influenza Vaccine Decision Step, Willing to Get Influenza Vaccine Decision_Option, Willing to Get Influenza Vaccine RuleInChoice and Willing to Get Influenza Vaccine Three_Value_Criterion are shown in Figure 18.

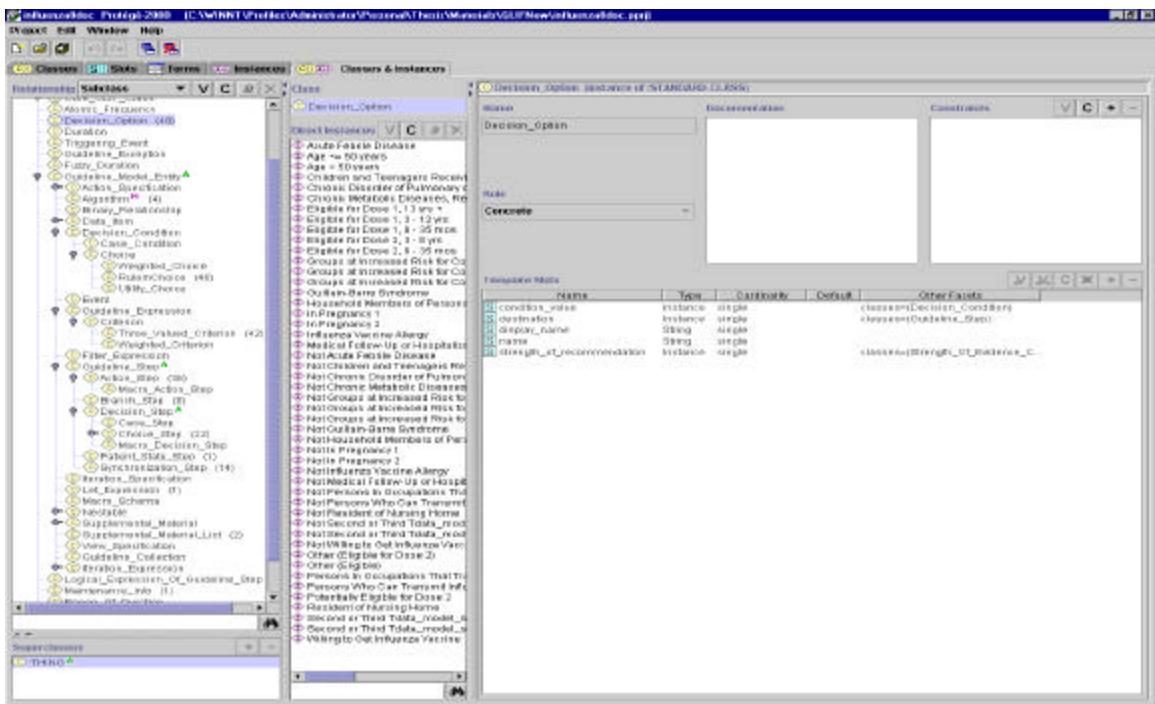


Figure 17. Class Ontology – Decision Step, Decision_Option, RuleInChoice, Three_Value_Criterion and Class Definition – Decision_Option

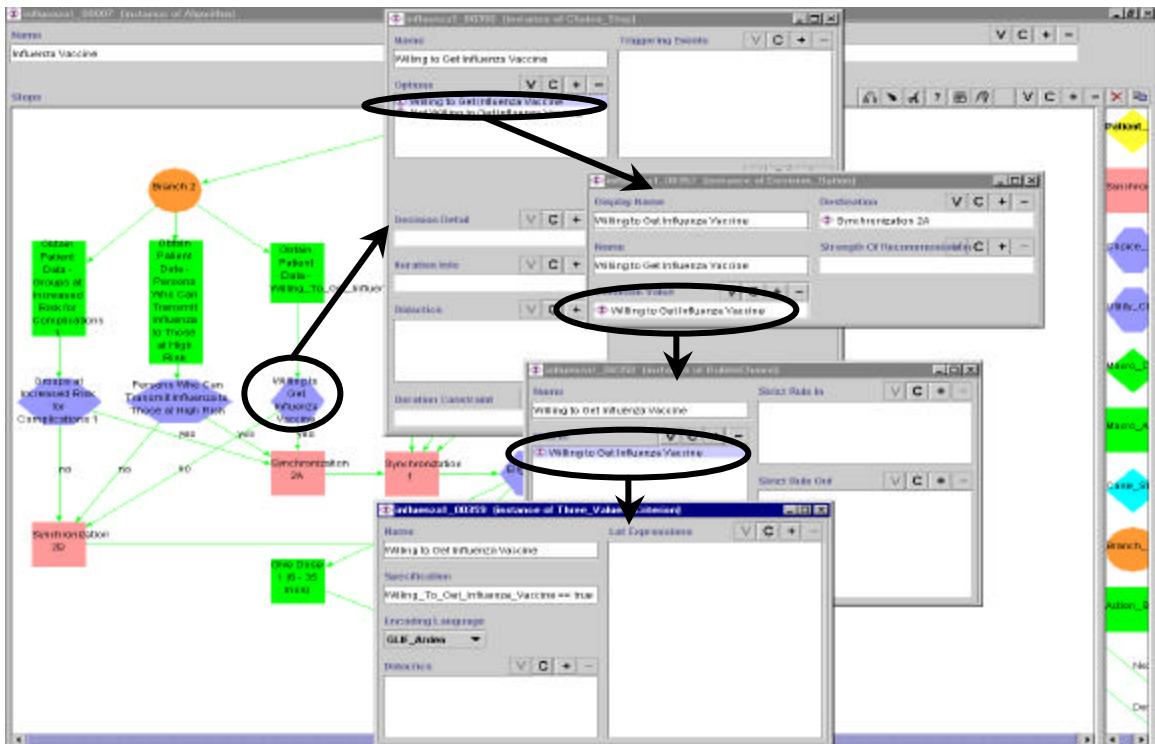


Figure 18. Decision Step Instance – Willing to Get Influenza Vaccine, Decision_Option Instance – Willing to Get Influenza Vaccine, RuleInChoice Instance – Willing to Get Influenza Vaccine, and Three_Value_Criterion Instance – Willing to Get Influenza Vaccine

Patient State Step

In a patient state step we need to define Patient State Description, Next Step, etc. For example, in the End of Guideline Step, we define the Patient State Description as *End of Guideline*.

The class definition of Patient State Step is shown in Figure 19 and the instances of End of Guideline Patient State Step and End of Guideline Three_Value_Criterion are shown in Figure 20.

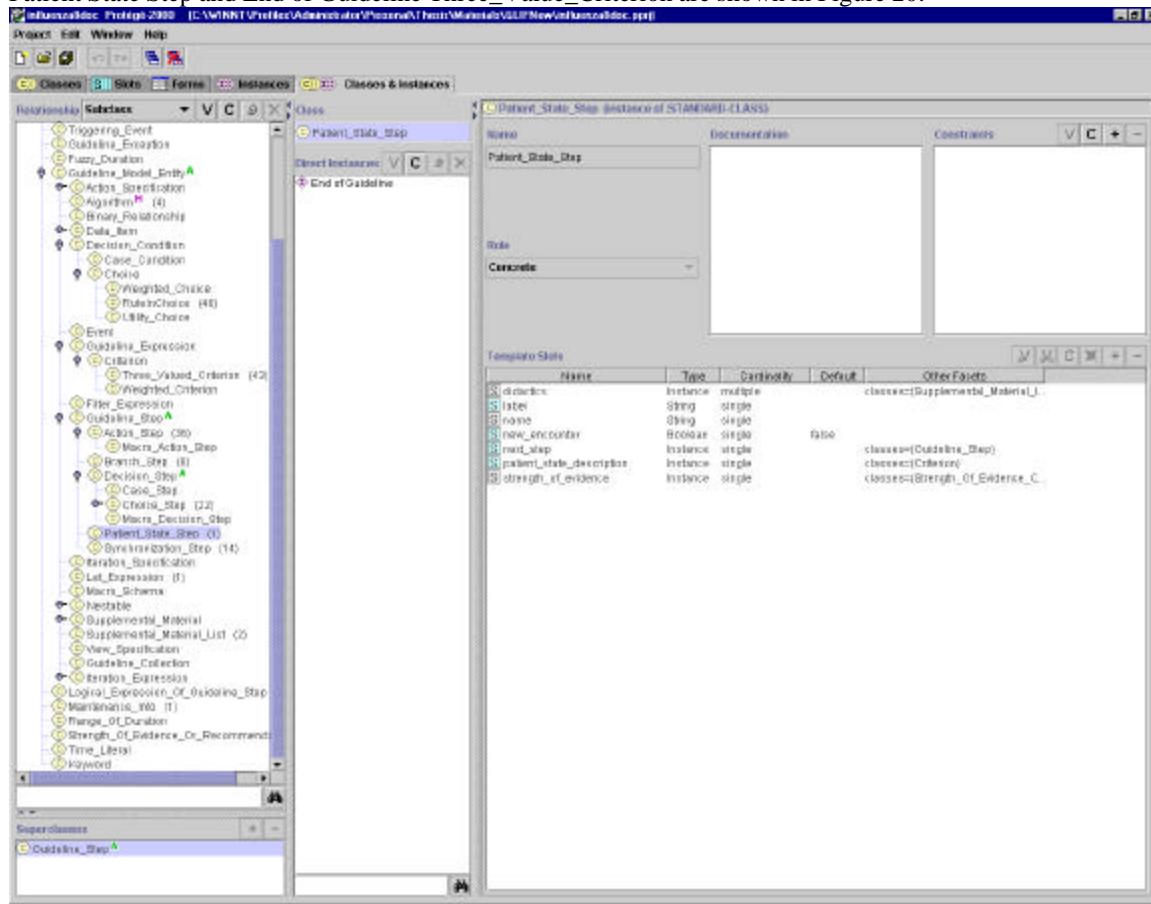


Figure 19. Class Definition – Patient State Step

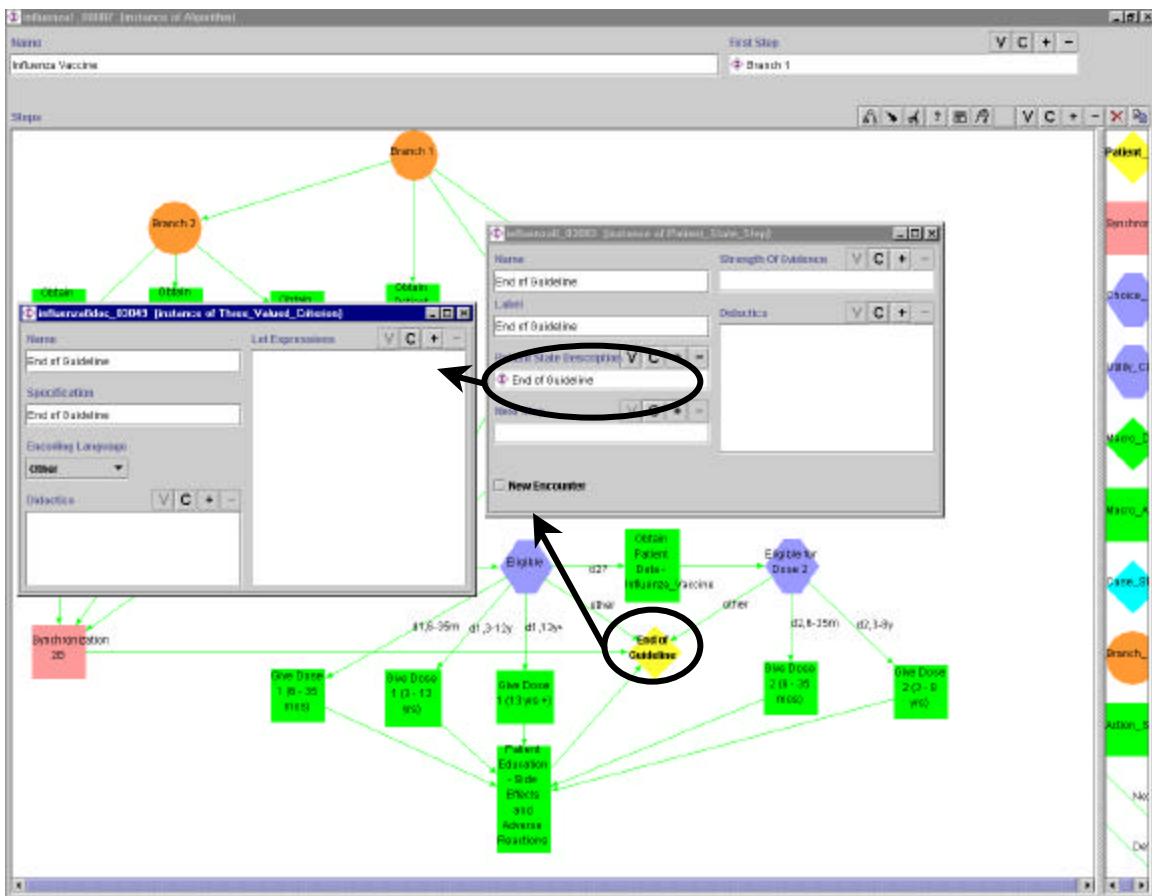


Figure 20. Patient State Step Instance – End of Guideline and Three_Value_Criterion Instance – End of Guideline

Patient Data Definition

Patient data are defined by Data_Item class, in which we need to define Concept Id, Concept Source Id, Data Model Class Id, Data Model Source Id, Data Value, etc. For example, in the Influenza_Vaccine Variable_Data_Item (a subclass of Data_Item class), we define the Concept Id as C0021403, the Concept Source Id as UMLS, the Data Model Source Id as USAM, and the Data Value as Influenza_Vaccine. The values of Data Value are instances of Patient_Data class, for which we need to define Service Cd, Critical Time, etc. For example, in the Influenza_Vaccine Observation (a subclass of Patient_Data), we define the Service Cd as Influenza_Vaccine.

Similarly, in Influenza_Vaccine Vocabulary Code, we define Term Id as C0021403, Term Name as Influenza_Vaccine and Vocabulary Id as UMLS.

The class ontology of Data_Item, Patient_Data and Patient_Data_Code, and class definition of Variable_Data_Item are shown in Figure 21. The instances of Influenza_Vaccine Variable_Data_Item, Influenza_Vaccine Observation and Influenza_Vaccine Vocabulary_Code are shown in Figure 22.

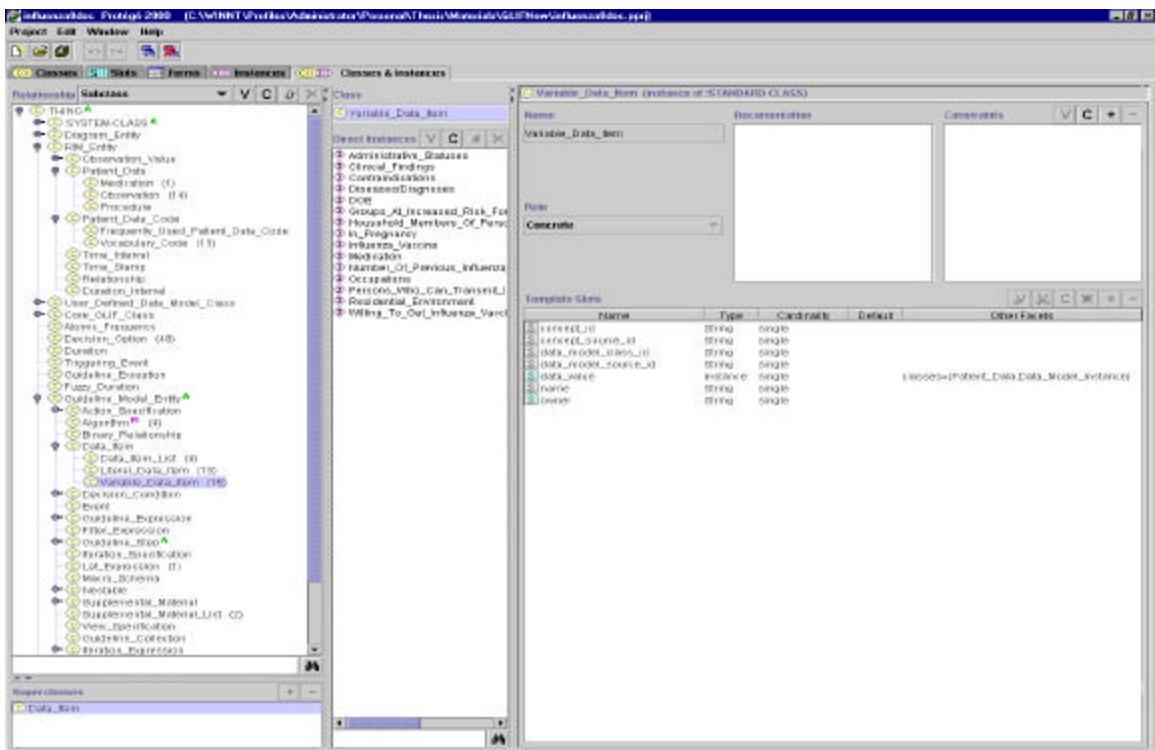


Figure 21. Class Ontology – Data_Item, Patient_Data, Patient_Data_Code and Class Definition – Variable_Data_Item

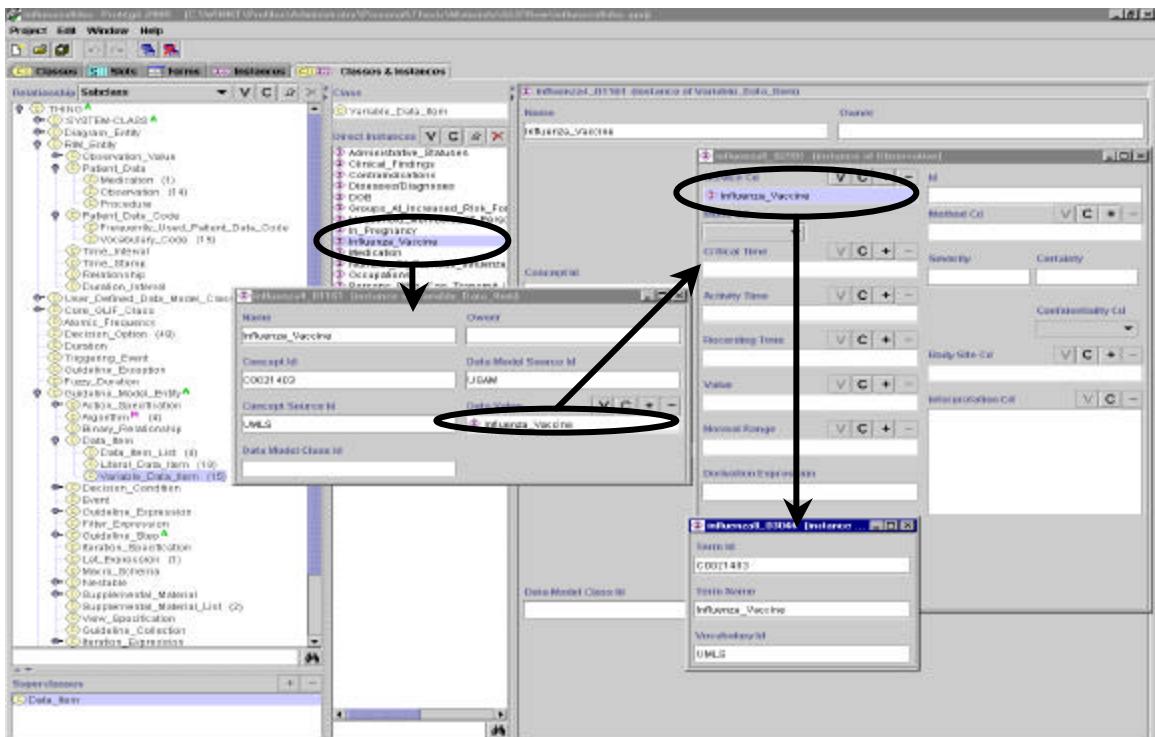


Figure 22. Variable_Data_Item Instance – Influenza_Vaccine, Observation Instance – Influenza_Vaccine and Vocabulary_Code Instance – Influenza_Vaccine

Parameters Passing to Subguidelines

For communication purpose, we may need to pass parameters to subguidelines. This is defined by the Parameters Passed Slot in the Guideline Class. For example, in the Obtain Patient Data – Contraindications Subguideline, parameter Contraindications is passed from the main guideline.

Within the Parameter_Passed Class, we need to define Direction and Variable Data Item. For example, in the Contraindications Parameter_Passed, we define Direction as out and Variable_Data_Item as Contraindications, as shown in Figure 23.

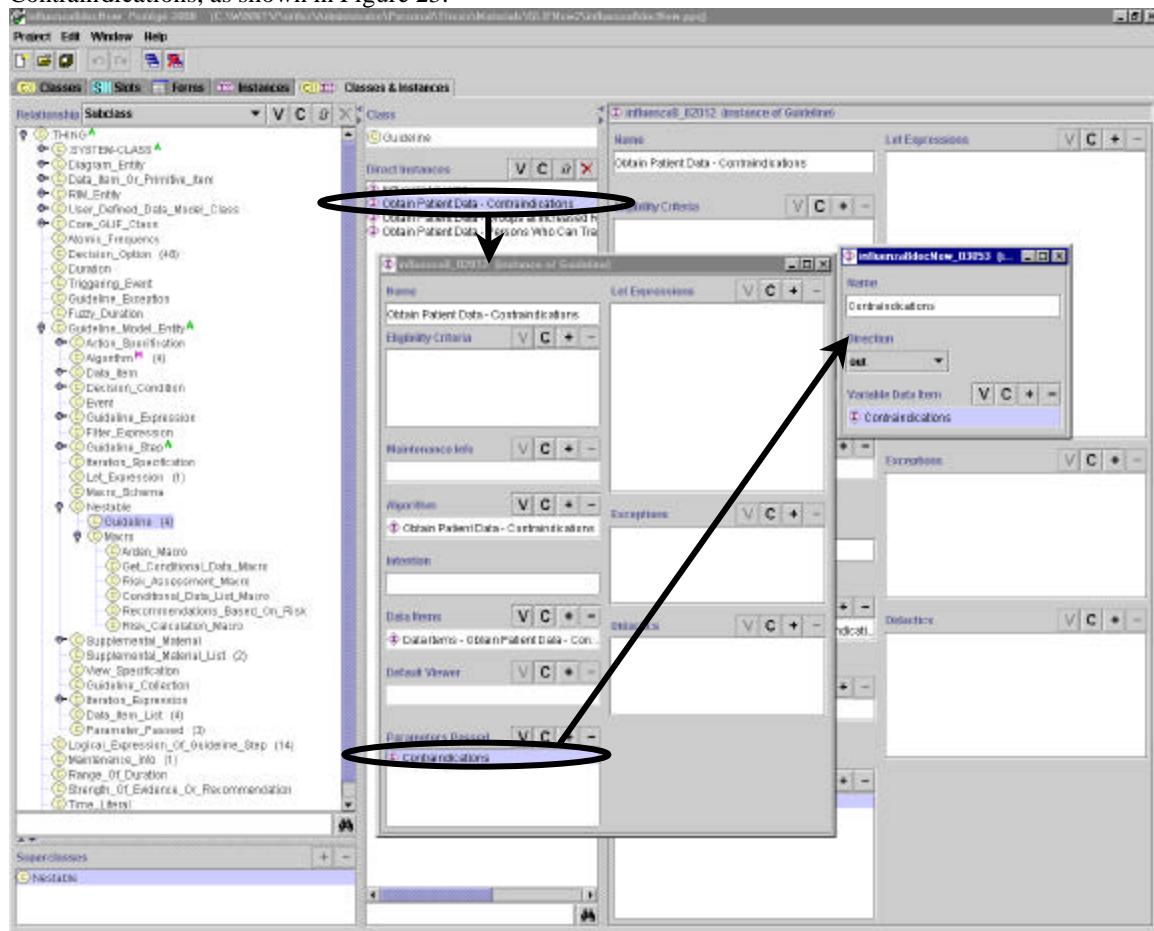


Figure 23. Parameters_Passed Instance - Contraindications

1.2 The cough guideline

In this example, we will show how a guideline is authored in GLIF.

A medical domain expert (a physician) first authors a guideline. The domain expert generates a Level-A specification. Then, an informatician generates the computable, Level-B specification.

Level A specification

The starting point for the guideline encoding is to indicate the top-level guideline, as shown in Figure 1.

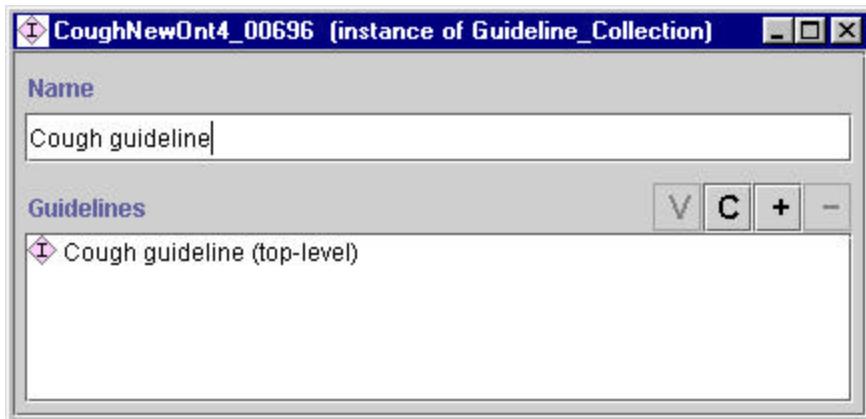


Figure 1. The guideline collection class identifies the top-level guideline object.

Next, the author fills in the guideline intention, maintenance information, eligibility criteria, and didactics, as shown in Figure 2, and goes on to create the top-level algorithm, shown in Figure 3.

Name		Maintenance Info (1 values)		
Cough guideline (top-level)				
Intention		Title		
Manage chronic cough		Managing cough as a defense mechanism and as a symptom		
Eligibility Criteria		Author		
chronic cough in immunocompetent adults		Richard S. Irwin, MD, FCCP Worcester, MA		
Algorithm		Encoder		
Top-level Cough algorithm		Mor Peleg, PhD		
Didactics		Authoring Date		
http://www.chestnet.org/health.science.policy#c		August 1998		
		Encoded Last Modified		
		10/25/2000		
Developing Institution		Adapting Institution		
American College of Chest Physicians American Thoracic Society Canadian Thoracic Society				
Guideline Version		Encoded Guideline Version		
		2.0		

Figure 2. The top-level guideline

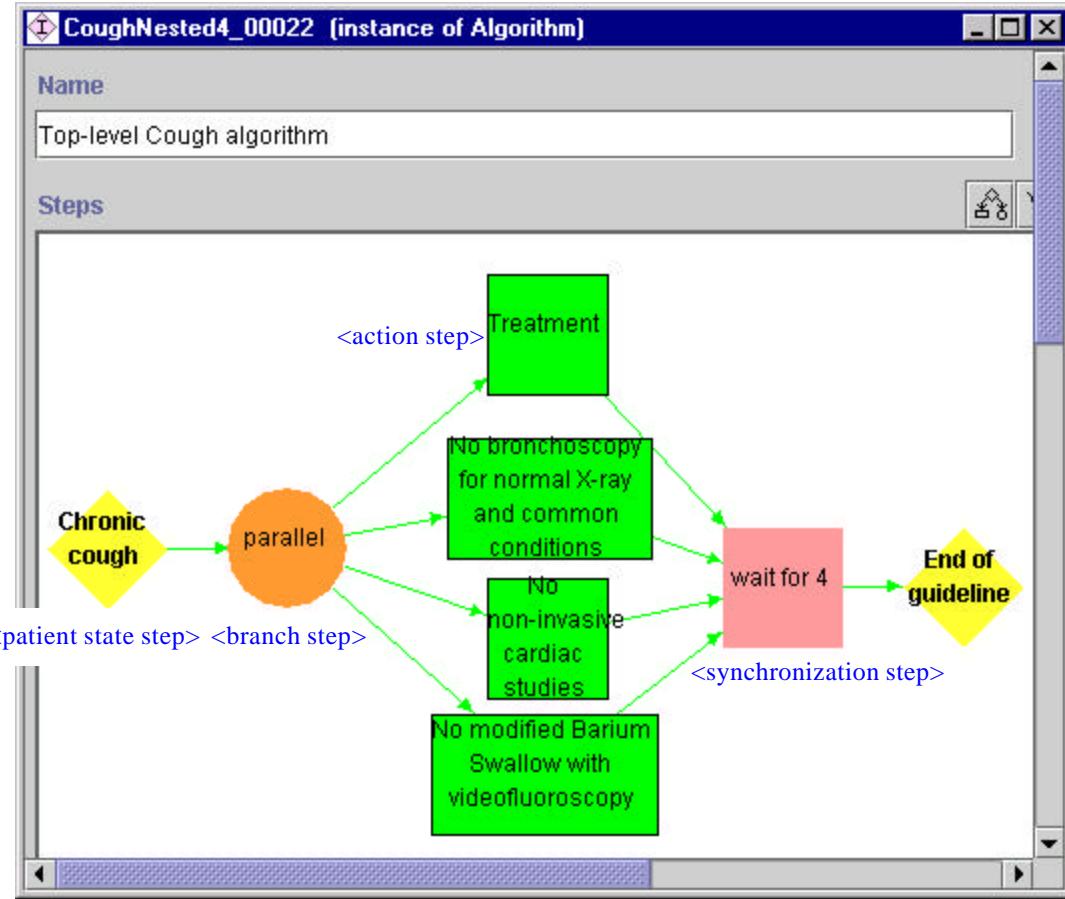


Figure 3. The top-level cough algorithm. Nested steps are shown with a bold contour.

The algorithm shown in Figure 3 specifies that there are four action steps (shown in green boxes) that should be done in parallel. The domain expert creates the algorithm by dragging and dropping the different guideline steps, and connecting them by connectors. The only information that is filled in at this time is the name of each step, didactics, strength of evidence, strength of recommendation, and iteration information (see

Figure 4). Next, the author expands some of the action and/or decision steps in algorithm to represent their details. For example,

Figure 4 shows how the author specifies that the task that should be done in the action step “No bronchoscopy for normal X-ray” action step, shown in Figure 3, should be of type sub-guideline action specification.

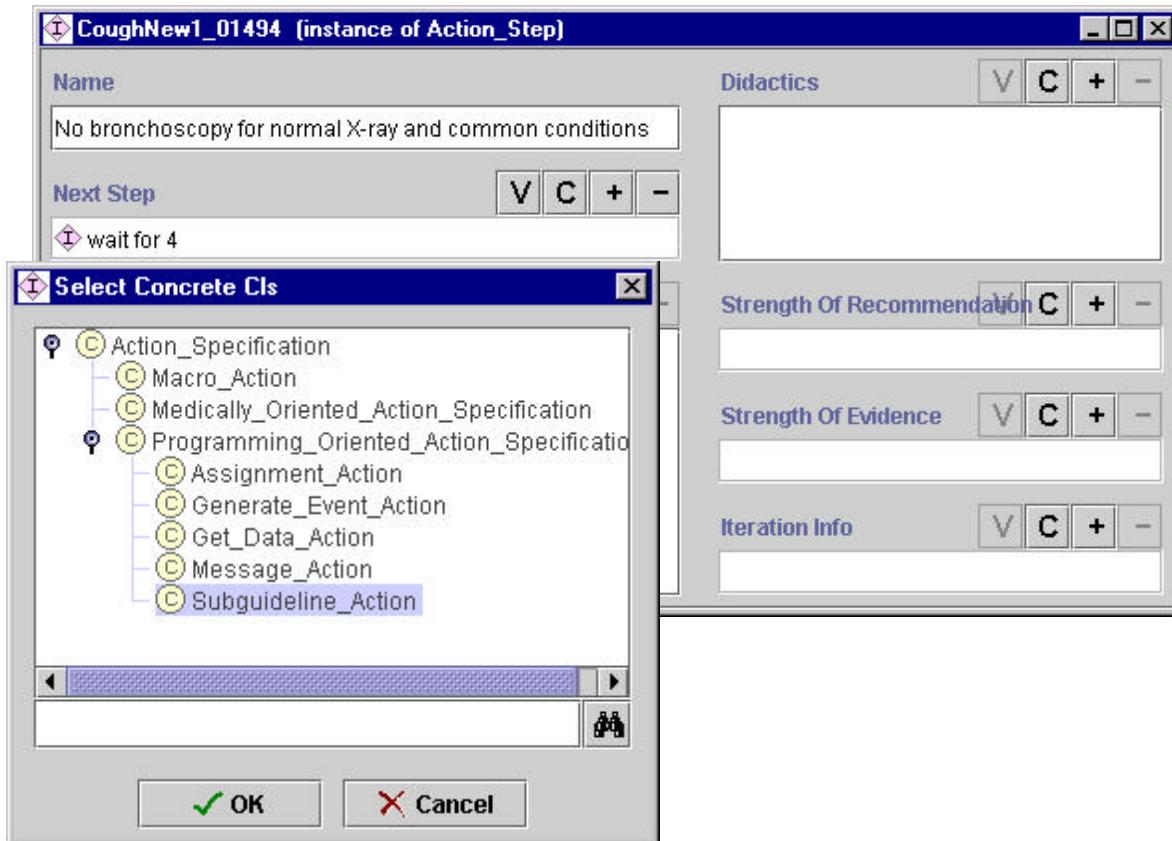


Figure 4. Details of the action step “No bronchoscopy for normal X-ray” action step, shown in Figure 3.

Next, the domain expert specifies the sub-guideline for “No bronchoscopy for normal X-ray”, as shown in Figure 5, and gives the algorithm in Figure 6. Note that “no bronchoscopy” is a negative recommendation, that is modeled here as an action. Later on, the informatician will fill in the details of that action, showing it to be a medically oriented action specification that is an order not to do a procedure to a patient, and indicating the UMLs code of the procedure “bronchoscopy”.

CoughNested4_00030 (instance of Guideline)

Name	Maintenance Info (0 values)
No bronchoscopy for normal X-ray and common condition	V C + - X
Intention	
Eligibility Criteria	V C + -
Algorithm	V C + -
No bronchoscopy for normal X-ray	
Didactics (1 values)	V C + -
Name	
Unless the chest radiograph is abnormal (e.g., suggesting malignancy or inflammatory lung disease), flexible bronchoscopy will have a lower diagnostic utility, in the range of 5%. Therefore, fiberoptic bronchoscopy should not be ordered routinely.	
Purpose	Items
Evidence	V C + -

Figure 5. The “no bronchoscopy” sub-guideline

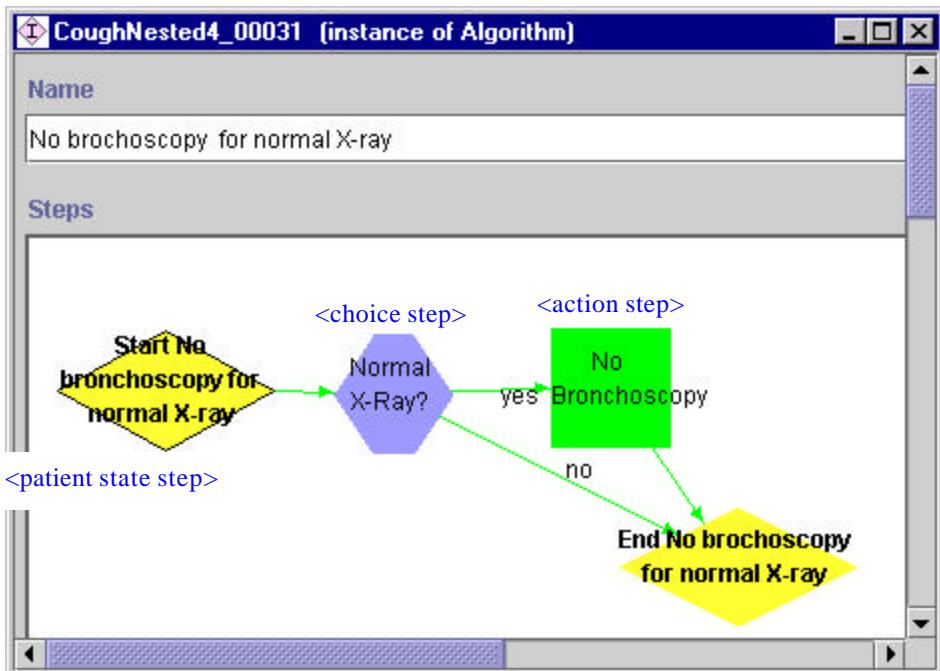


Figure 6. The “no bronchoscopy” algorithm.

The domain expert completes the guideline specification (Level A) in a similar manner. The rest of the specification is shown in Figure 7 through Figure 16.

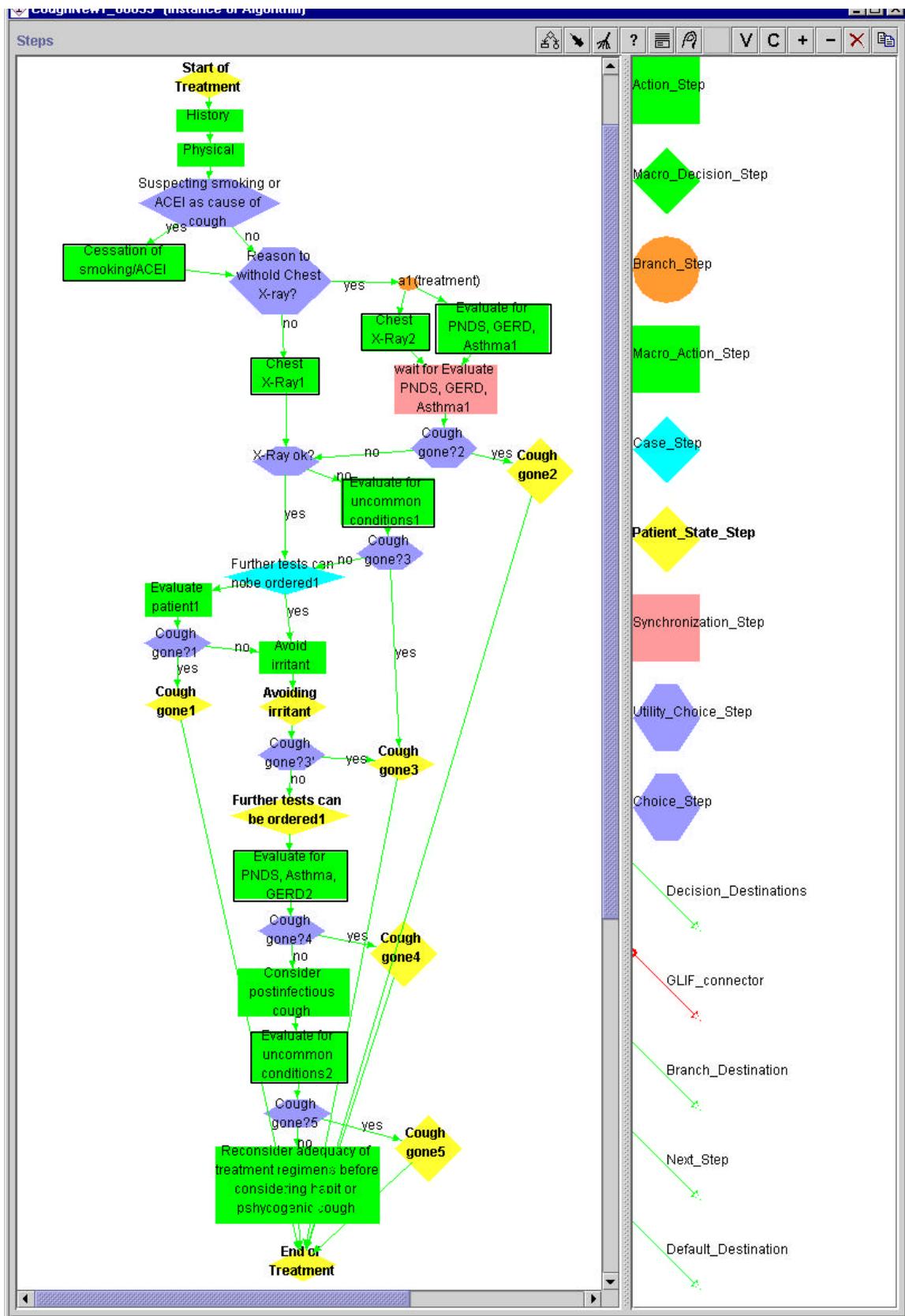


Figure 7. The treatment algorithm that expands the action step shown in Figure 3.

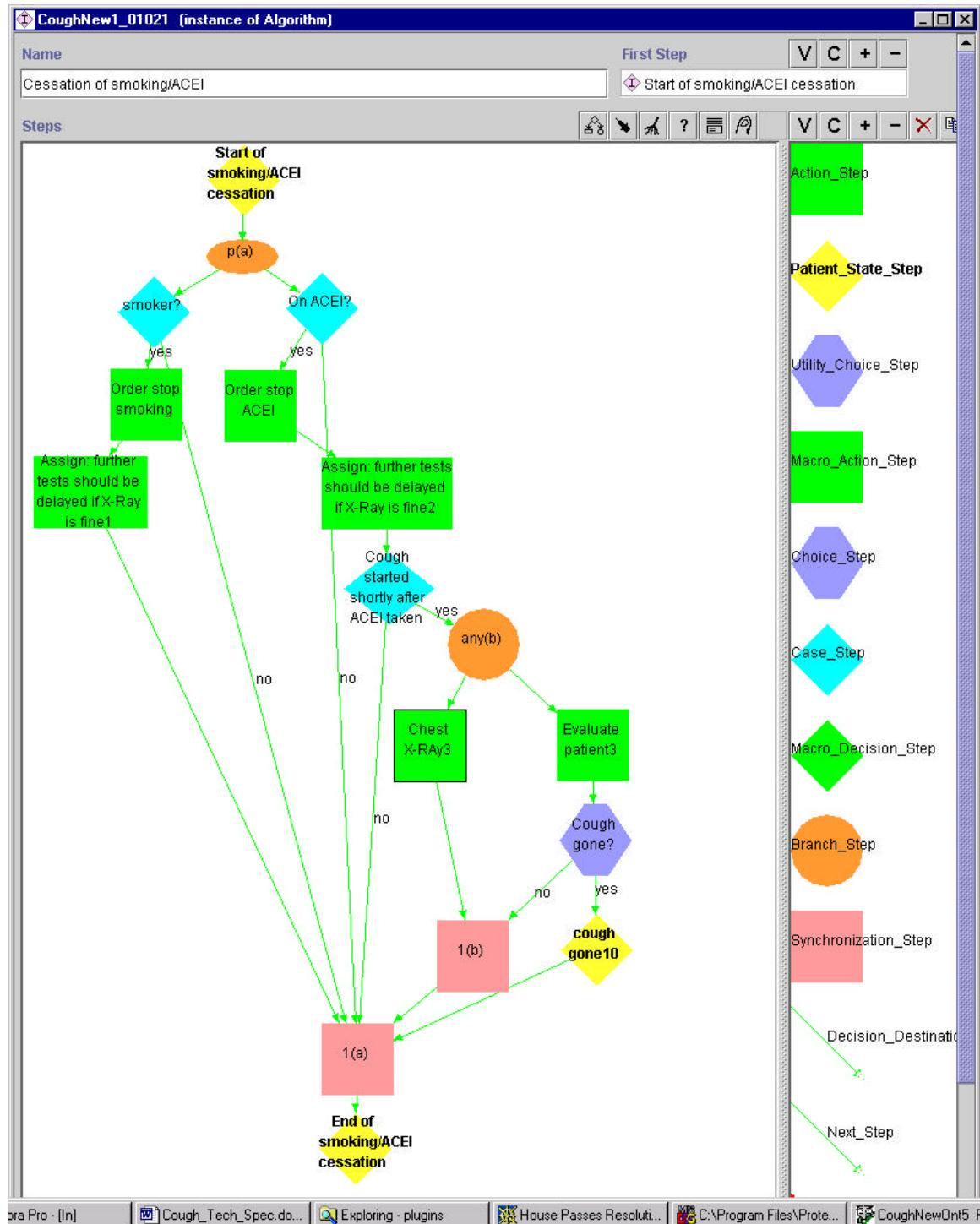


Figure 8. The “cessation of smoking/ACEI” algorithm, which shows the details of the corresponding action step of Figure 7.

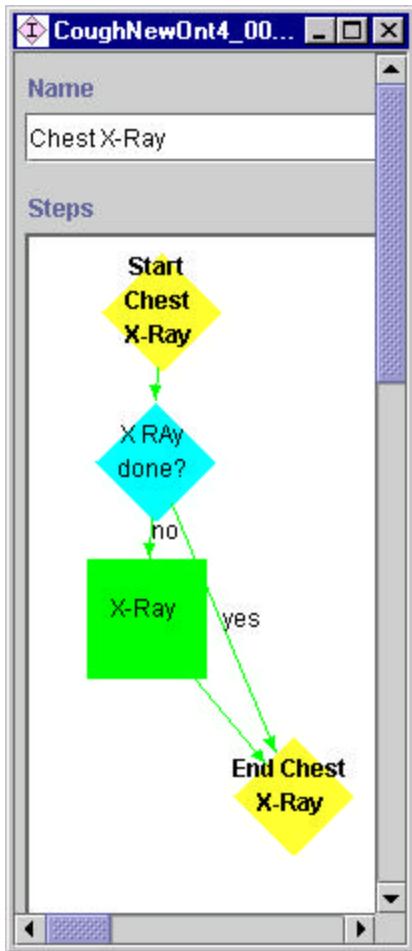


Figure 9. The “Chest X-Ray” algorithm, which shows the details of the corresponding action steps of Figure 7 and Figure 8.

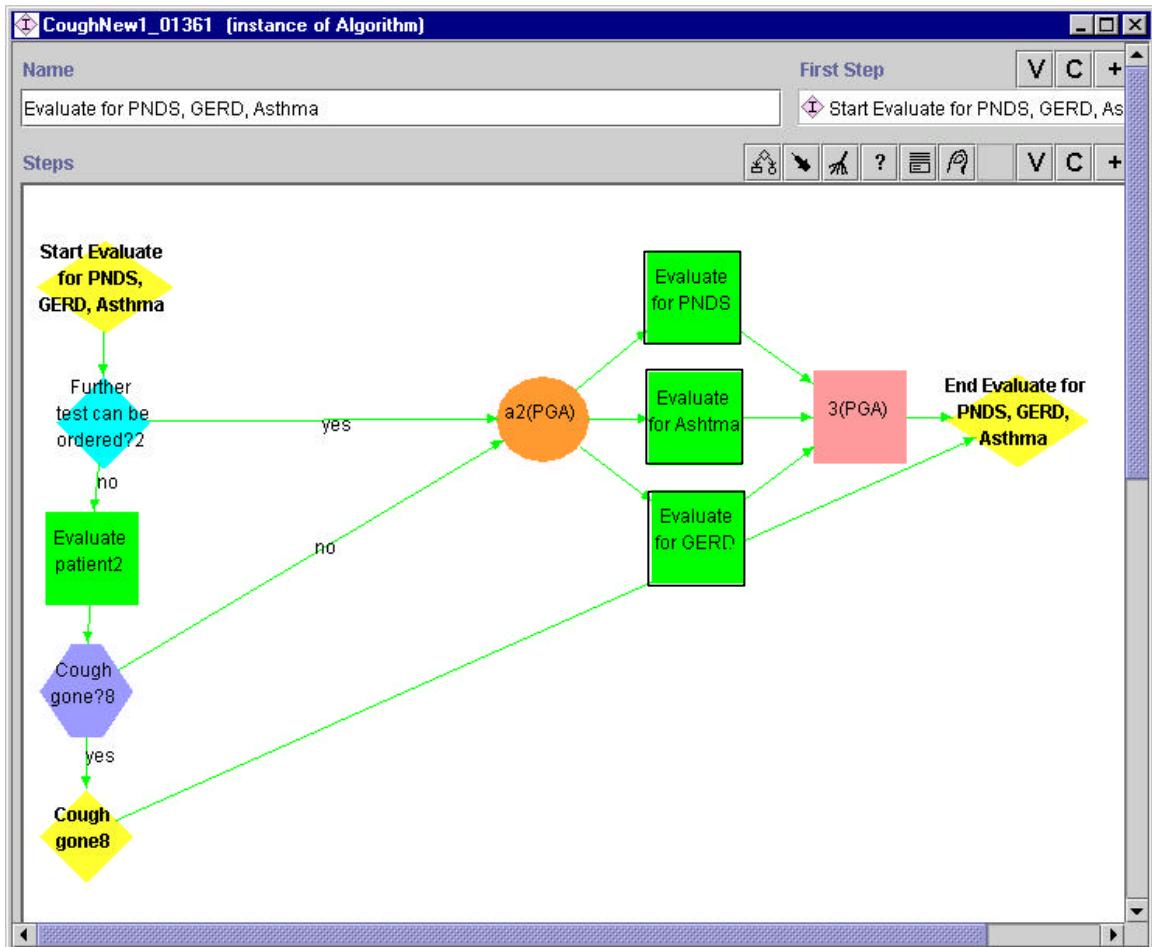


Figure 10. The “Evaluate for PNDS, GERD, Asthma” algorithm, which shows the details of the corresponding action steps of Figure 7.

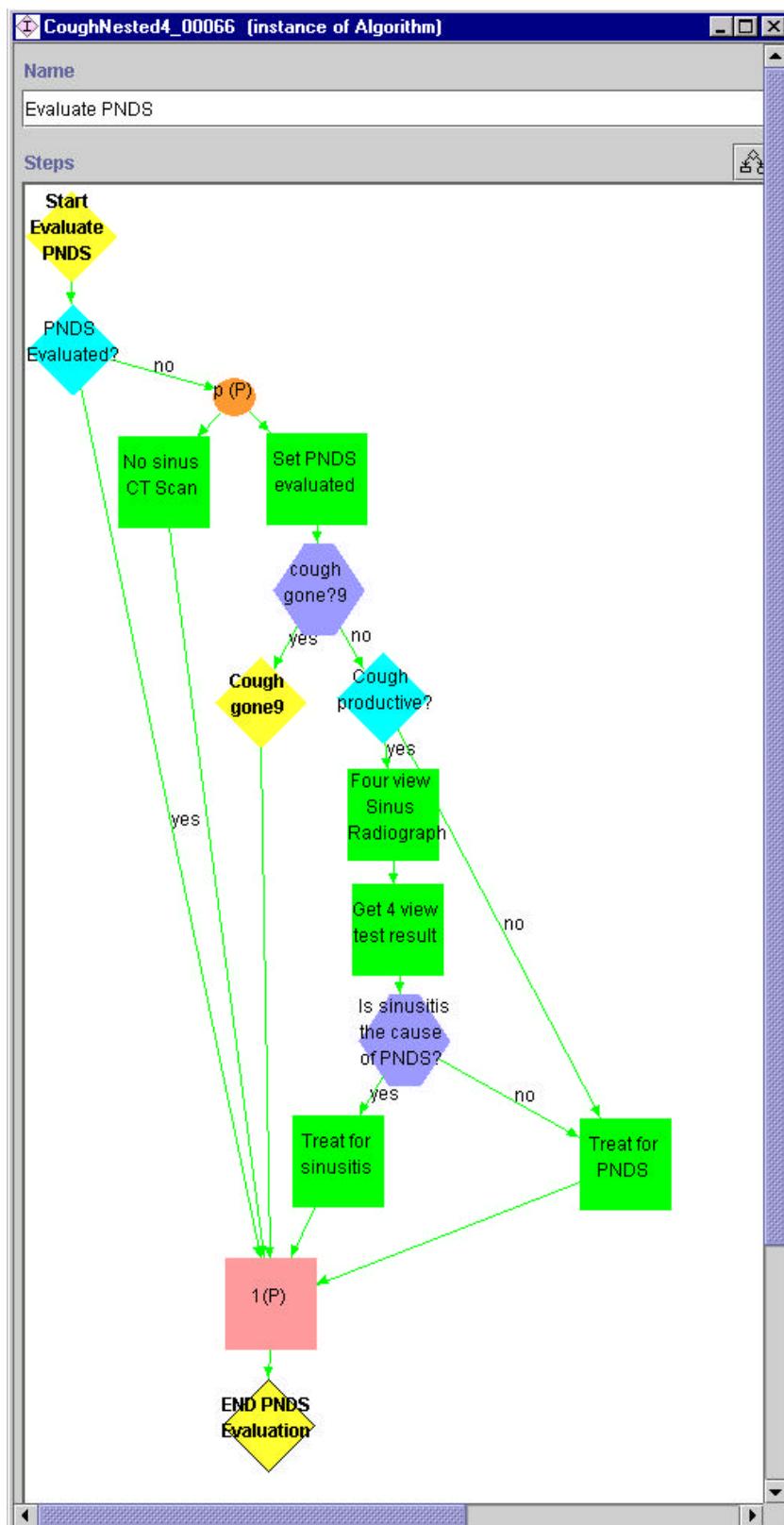


Figure 11. The “Evaluate for PNDS” algorithm, which shows the details of the corresponding action steps of Figure 10.

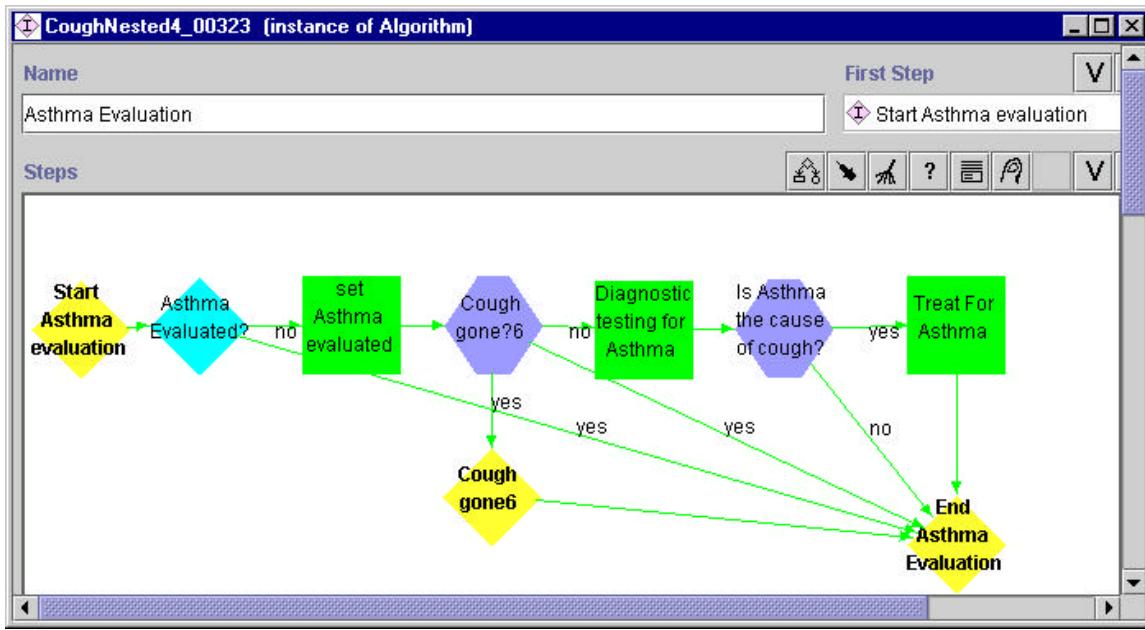


Figure 12. The “Evaluate for Asthma” algorithm, which shows the details of the corresponding action steps of Figure 10.

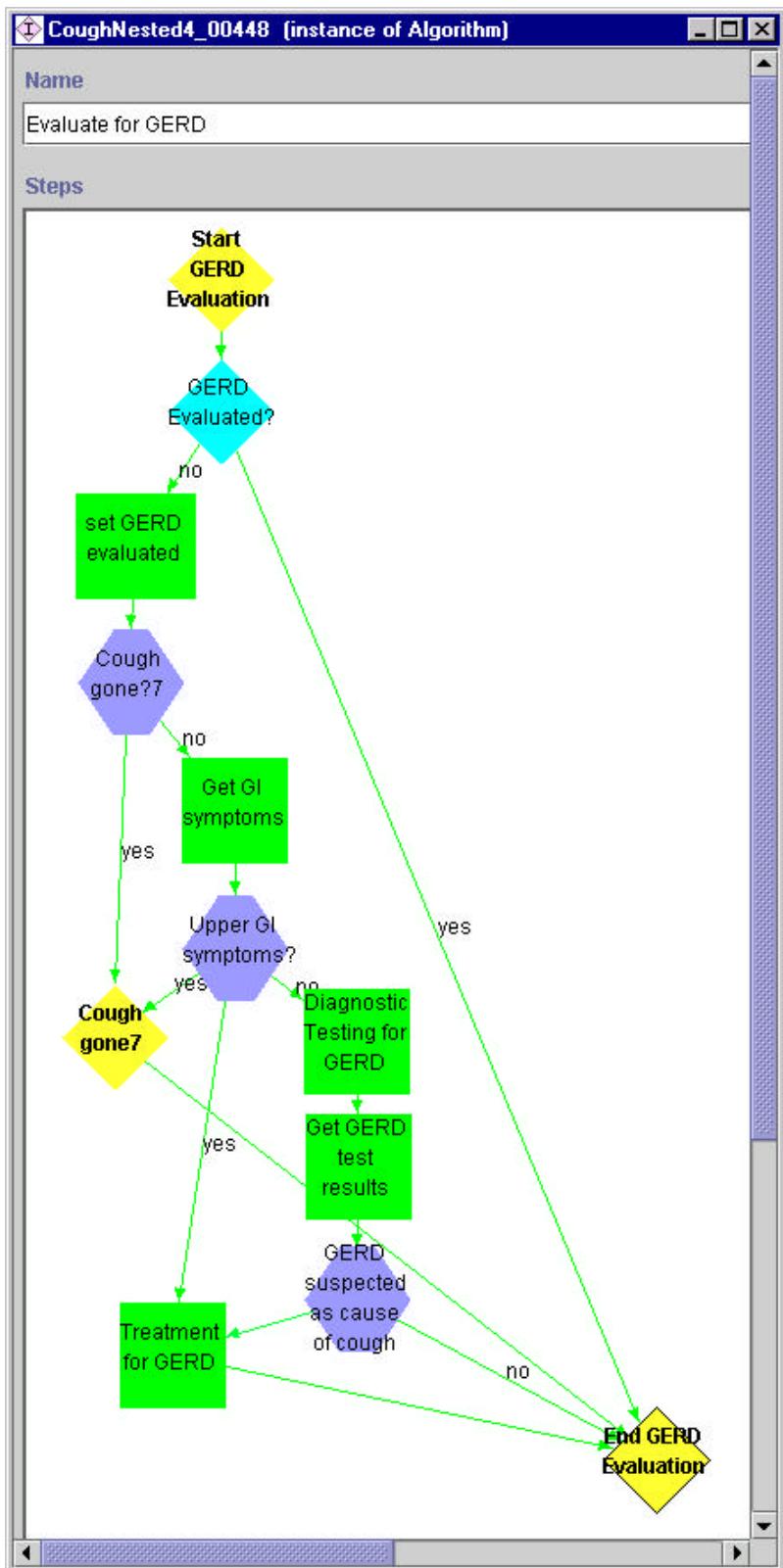


Figure 13. The “Evaluate for GERD” algorithm, which shows the details of the corresponding action steps of Figure 10.

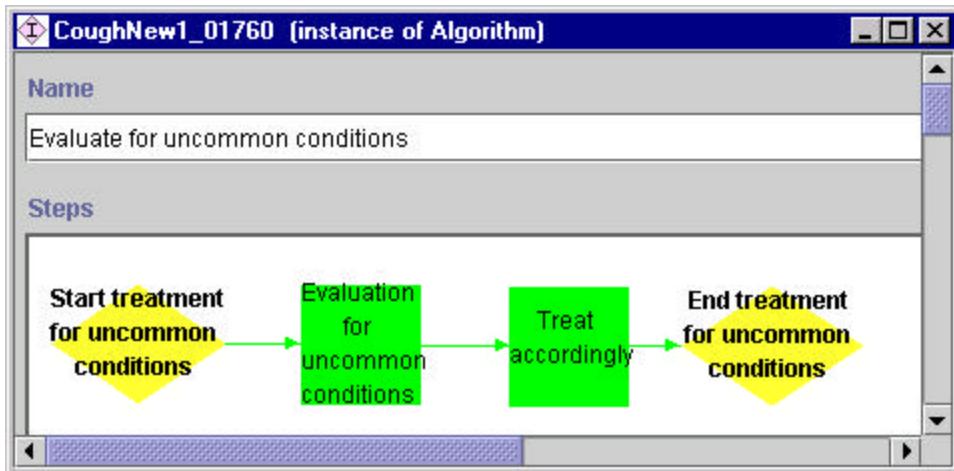


Figure 14. The “Evaluate for uncommon conditions” algorithm, which shows the details of the corresponding action steps of Figure 7.

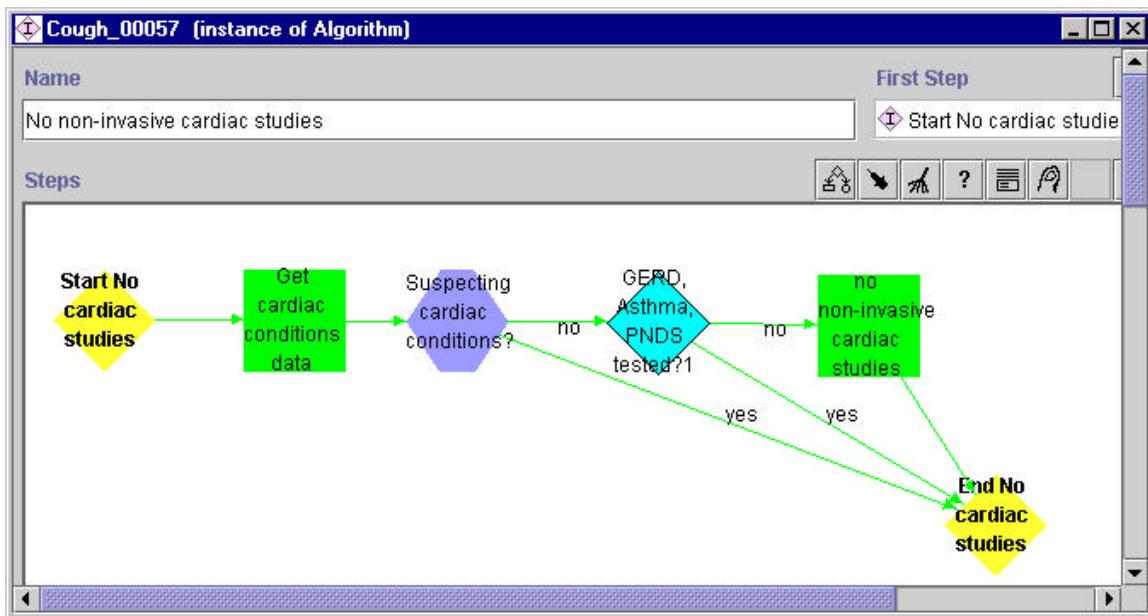


Figure 15. The “No modified Barium Swallow with videofluoroscopy” algorithm which shows the details of the action step of Figure 3.

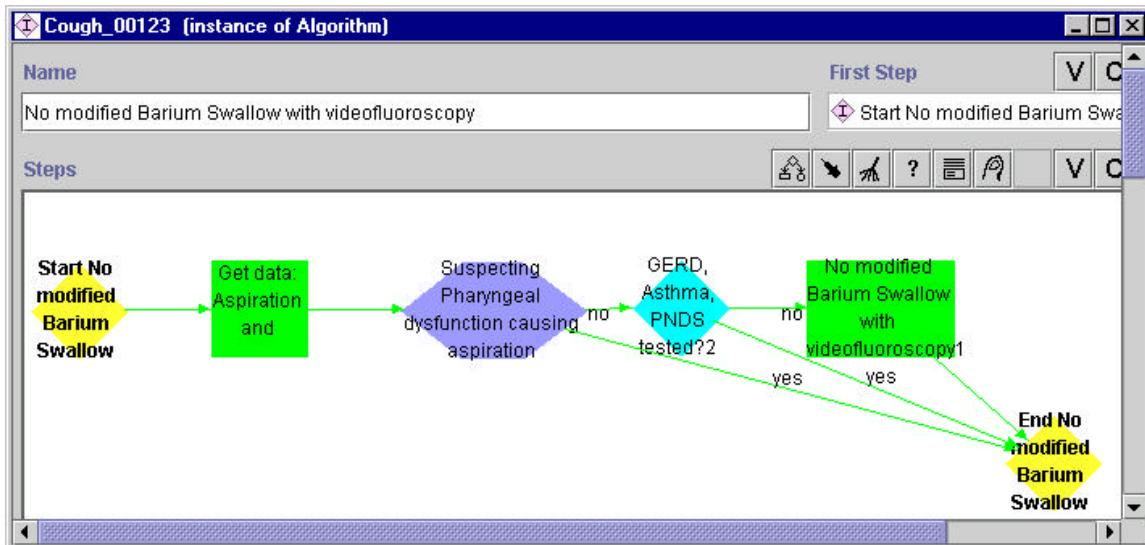


Figure 16. The “No non-invasive cardiac studies” algorithm, which shows the details of the action step of Figure 3.

Level A specification

After the domain expert generated the Level A specification, it is not the task of the informatician to encode Level B of the specification. The informatician consults with the domain expert in issues that require medical expertise.

The starting point for informatician is the top-level guideline, shown in

The screenshot shows a software interface for managing a medical guideline. The title bar reads "CoughNested4_00021 (instance of Guideline)". The main area is divided into several sections:

- Name:** Cough guideline (top-level)
- Intention:** Manage chronic cough
- Eligibility Criteria:** Chronic cough in immunocompetent adults
- Algorithm:** Top-level Cough algorithm
- Didactics:** A link to a website: <http://www.chestnet.org/health.science.policy/#c>
- Maintenance Info (1 values):** Contains fields for Title, Author, Authoring Date, Developing Institution, and Guideline Version.
- Title:** Managing cough as a defense mechanism and as a symptom
- Author:** Richard S. Irwin, MD, FCCP
Worcester, MA
- Encoder:** Mor Peleg, PhD
- Authoring Date:** August 1998
- Encoded Last Modified:** 10/25/2000
- Developing Institution:** American College of Chest Physicians
American Thoracic Society
Canadian Thoracic Society
- Adapting Institution:** (empty)
- Guideline Version:** 2.0

Figure 2. The first task of the informatician is to go over the algorithms (flowcharts) and check them to see that the logic looks complete and consistent. At the moment we do not have tools that will aid in automatic validation, but we are planning to develop such tools. The informatician will specifically look at:

- 1) Case steps and see that the decision options are mutually exclusive and that there are no missing options (e.g., only “yes” exists, but “no” doesn’t).
- 2) Branch and synchronization steps match.
- 3) Patient state steps exist at places where new encounters can begin
- 4) Optionally, if such design is sought, that every guideline (and sub-guideline) starts with one patient state step and ends in one patient state step, and all paths in a given guideline reach the terminal guideline step. This was done in the cough guideline.
- 5) A given action should not be done more than once unless it is desirable. For example, in the cough guideline, the Action “X-Ray” is indicated three times, twice in the Treatment algorithm, and once in the cessation of smoking/ACEI algorithm. However, all of these actions reference the Chest X-Ray sub-guideline that checks whether an X-ray has already been done before another X-Ray is ordered.
- 6) Variables are initialized and set in the correct places. For example, the variables that hold the state of an order (X-Ray taken or not yet taken) should be initialized. This was not done by the algorithm shown in Figure 3, and therefore an action step called “Initialize Parameters ” is introduced, as shown in Figure 17.

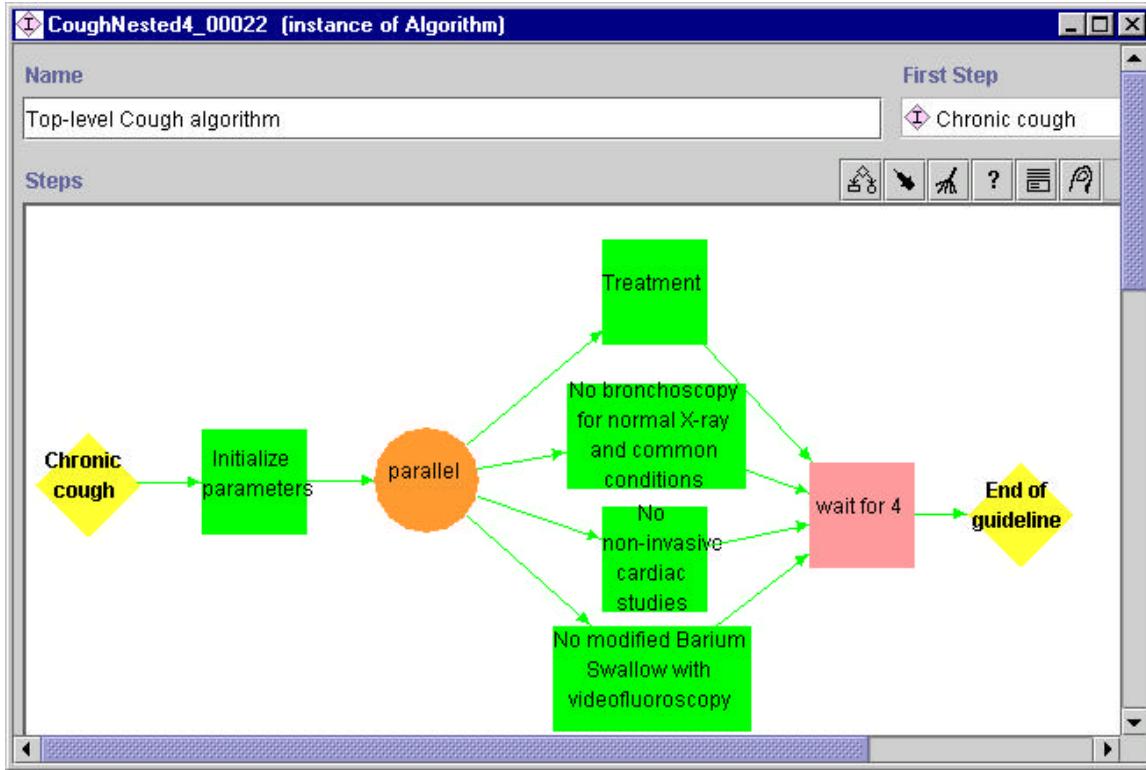


Figure 17. The top-level cough algorithm that includes a parameter initialization step.

The informatician specifies level B attributes according to the level A attribute values that the domain expert filled in. The following paragraphs discuss how the different types of guideline steps are specified.

Specifying action steps

Every action should have at least one task specified. The medical domain expert fills in subguideline tasks (see Figure 5), but not other action types of tasks. Tasks are of type `action_specification` class, discussed in Section 0. There are several kinds of action specification. We will demonstrate how they are filled in.

Assignment task

Figure 18 shows an example of an assignment task. A primitive (simple) data item is assigned with an expression.

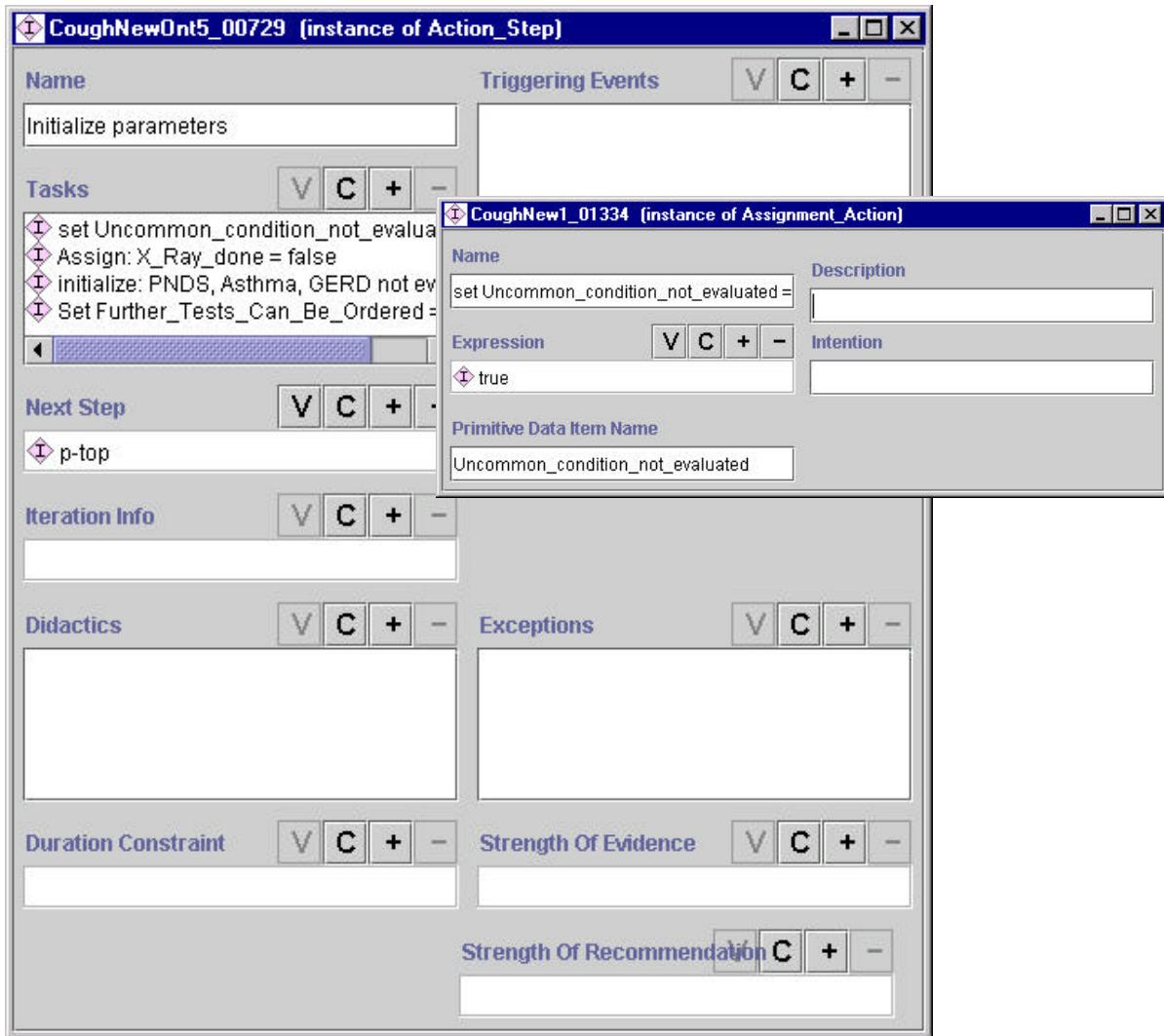


Figure 18. The Initialize parameters action step of Figure 17.

Medically-oriented action specification

A medically-oriented task (action specification) refers to a literal data item (See Section 2.3.1). Literal data items refer to a medical concept, a data model class, and to a single patient data item, which in the case that the USAM reference information model is used, can be a medication, observation, or medical procedure. Usually, the moods that are used in medically-oriented tasks are **order** or **order-not-to**.

The literal data items that are referenced by the medically-oriented tasks need to be added to the guideline that contains the medically-oriented tasks, as shown in Figure 29.

CoughNew1_01052 (instance of Action_Step)

Name	Triggering Events
Order stop smoking	V C + -
Tasks	V C + -
order stop smoking for 4 weeks starting now	
Assign Time_	

CoughNew_INSTANCE_00148 (instance of Literal_Data_Item)

Name	Data Model Class Id	
Order stop smoking	Procedure	
Concept Id	Data Model Source Id	
C0037369	USAM	
Concept Source Id	UMLS	
Data Value (1 values)		
Service Cd	V C + -	Id
Smoking		
Mood Cd	order-not-to	Method Cd
Critical Time	V C + -	
Activity Time	V C + -	Confidentiality Cd

Figure 19. The “order stop smoking” action step of Figure 8.

Message task

When the guideline does not specify enough information so that an action can be specified by medically-oriented tasks that refer to medical concepts, its task can only be specified by a message action. For example, the cough guideline says that Asthma should be treated, but does not specify how this should be done. The appropriate message task is shown in Figure 20.

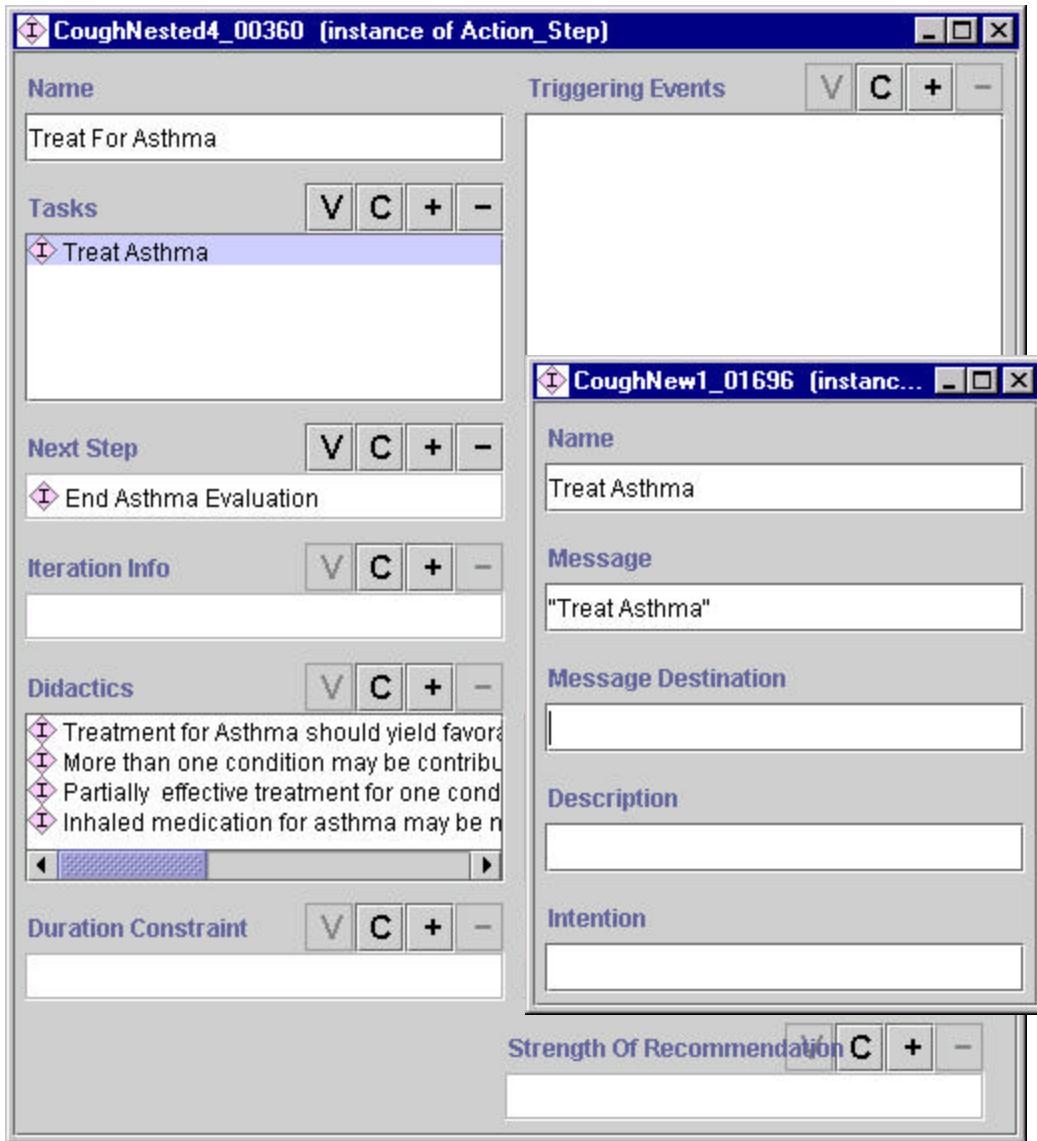


Figure 20. The “treat asthma” action step of Figure 12.

Get data task (action specification)

The get data task specifies what primitive attribute of a data item would be read and into which variable. Figure 21 shows how the ending time of the latest cough (in the list of cough instances of the patient for the past 8 weeks) is entered into the variable latest_cough_end_time. The temporal constraint is used to select one data item out of a list of data items of the patient that correspond to a single concept. It is usually the earliest or latest data item, where latest/earliest is chosen either by critical time or recording time. Other types of constraints may be applied to the list of data items before the temporal constraint is applied. Figure 22 shows an example of retrieving the latest body temperature measurement of all the measurements in the past 8 weeks that are

recorded in Celsius. All the attributes of the data item (recording time, body-site) may serve as constraints.

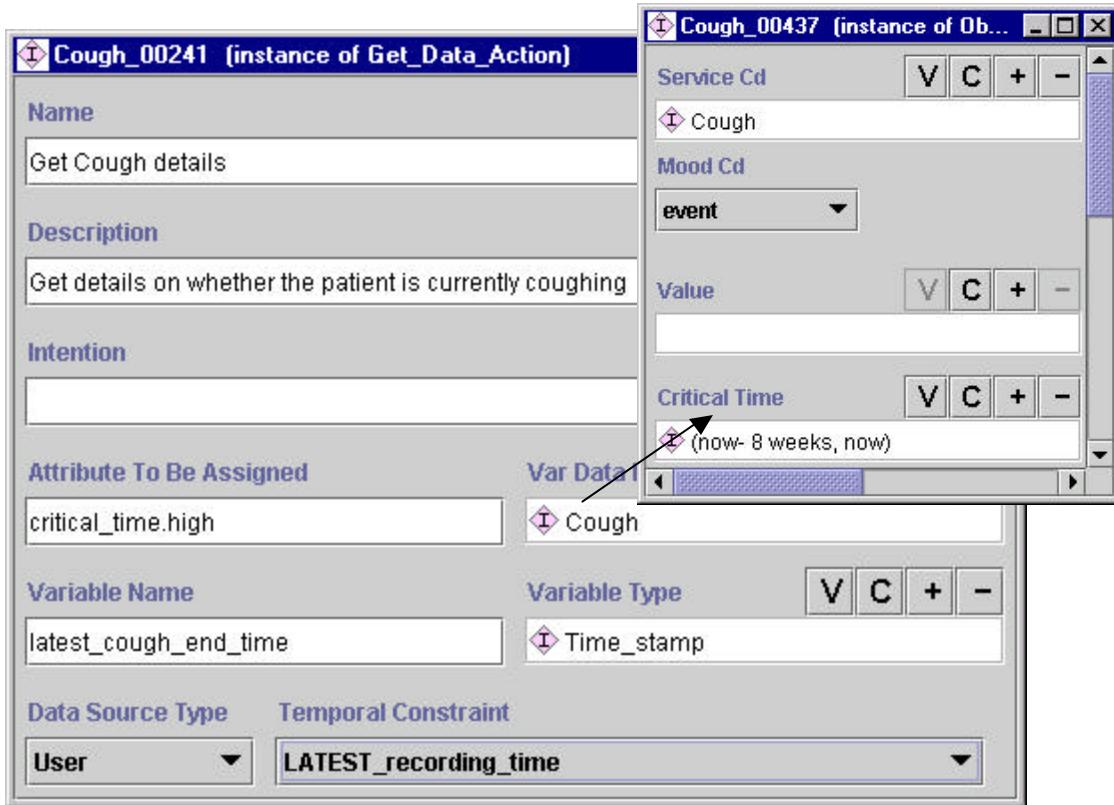


Figure 21. The “get cough details” task that is done within the history action step shown in Figure 7.

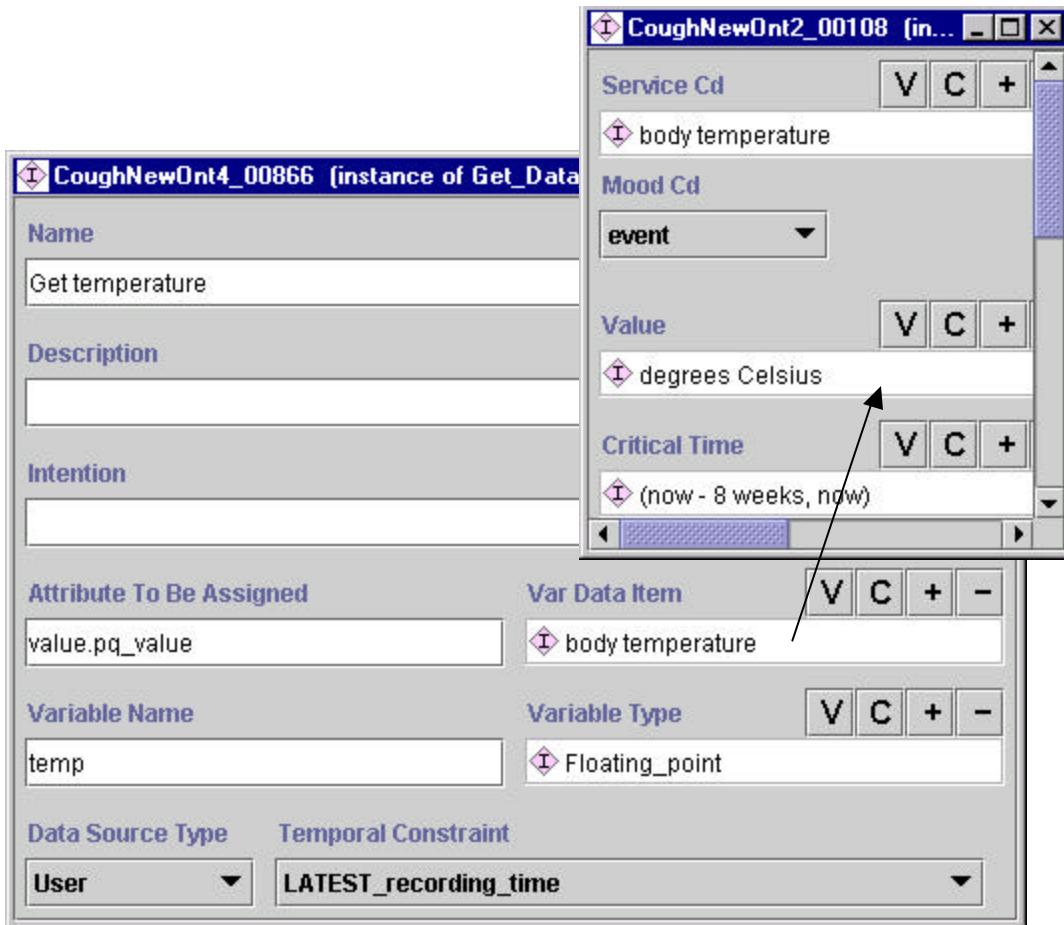


Figure 22. An example of getting a data item where some constraints apply (constraints on the unit of the measured value).

Other slots of Action_Steps that are specified at Level B

As Figure 18 shows, there are other slots of Action_Step, apart from tasks, that need to be specified at level B. These slots are:

- iteration_info (see Section 5.2)
- duration_constraint (specify minimum and maximum duration of the action step)
- triggering events (not finalized yet)
- exceptions (not finalized yet)

Specifying branch steps

An informatician representing a branch step at level B needs to specify whether the order constraint is “parallel” or “any order”, as shown in Figure 23.

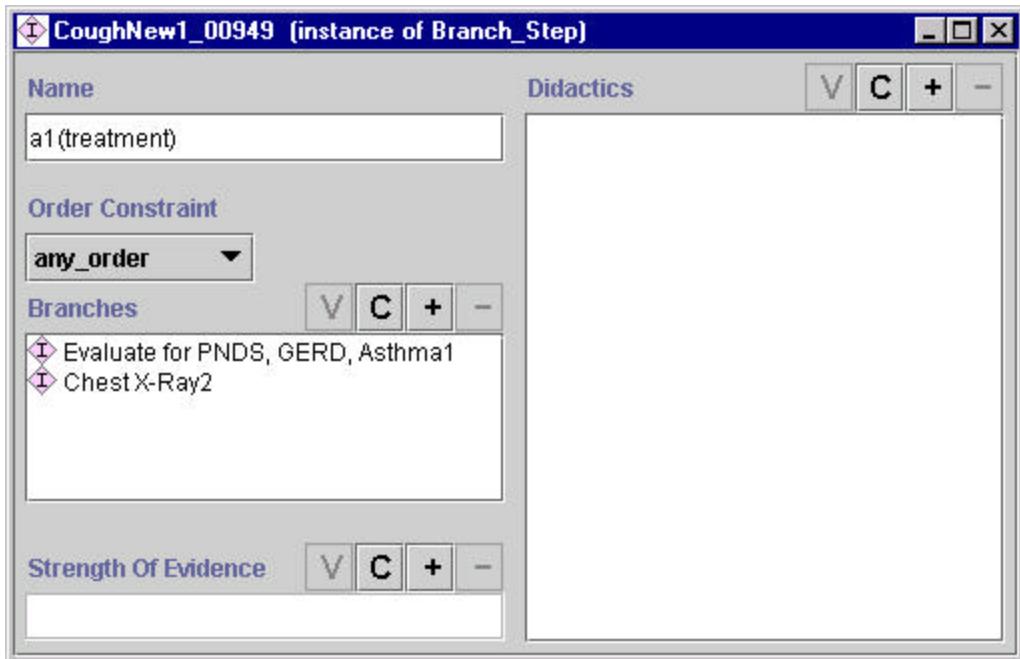


Figure 23. An example of the branch step, shown in Figure 7

Specifying synchronization steps

A level B representation of synchronization steps needs to specify the continuation slot of the step. There are two possibilities of specification. One, is to write a logical combination of guideline steps, as was done in the Flu guideline encoding. The other possibility is to say that the synchronization needs to wait for at least n incoming arcs from the guideline steps that connect to that synchronization steps. An example is shown below.

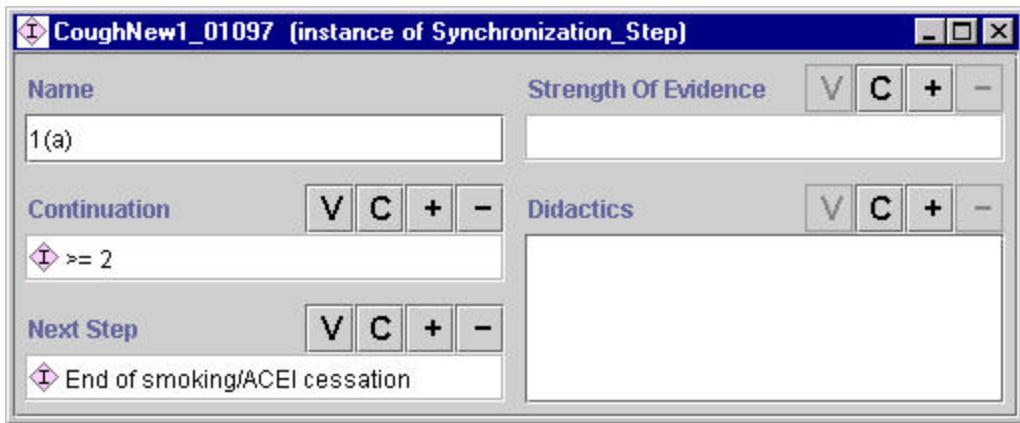


Figure 24. The details of the synchronization step shown in Figure 8

Patient State Steps

The informatician needs to formally specify the decision criterion of the patient state step according to the textual name of that step. For example, the patient state step “chronic cough”, shown in Figure 3 is specified as shown in Figure 26.

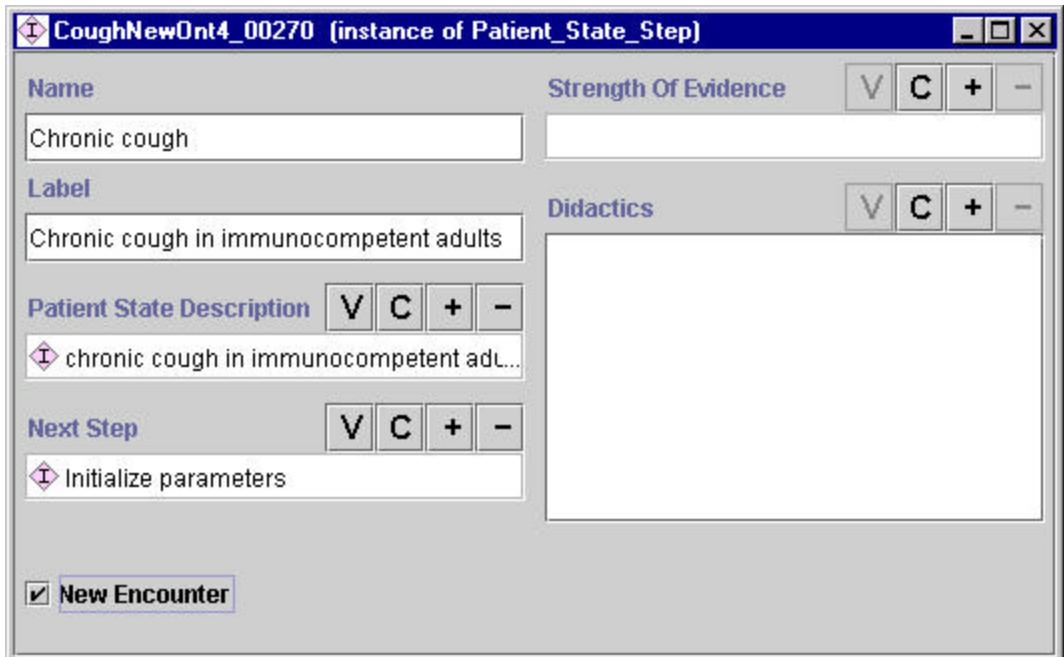


Figure 25. The details of the chronic cough patient state step shown in Figure 3

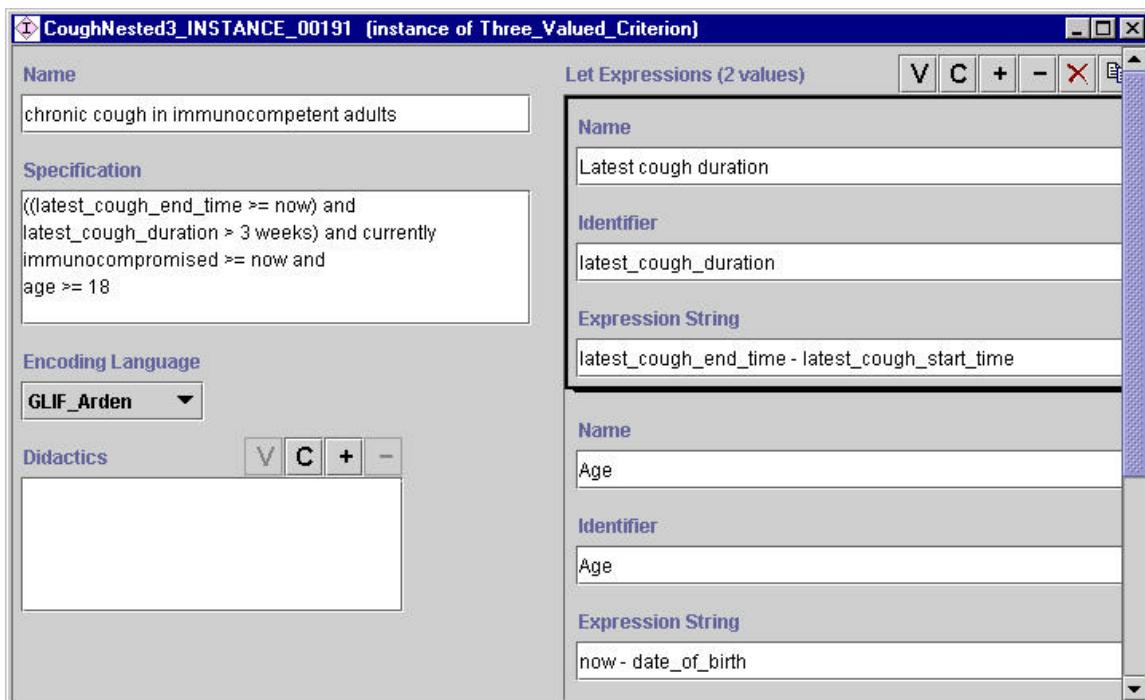


Figure 26. The details of the patient state description of the chronic cough step

Figure 26 shows a specification slot value that references four terms. latest_cough_end_time is a primitive_data_item that was retrieved before, as shown in Figure 21. latest_immunocompromised_end_time is also a primitive_data_item, that is defined in a similar manner to latest_cough_end_time. The two other terms are defined by Let_Expressions. latest_cough_duration is defined as the difference between two

primitive_data_items whose values were retrieved beforehand. Age is defined in reference to the primitive_data_item date_of_birth.

The literal and variable data items that are referenced by the patient state description need to be added to the guideline that contains the patient state description, as shown in Figure 29.

Specifying decision steps

Decision steps contain attributes that need to be filled in at level B. Some of them are similar to those of an action step (see Figure 27). They are:

iteration_info info

duration_constraint (specify minimum and maximum duration of the action step)

triggering events (not finalized yet)

exceptions (not finalized yet)

Case Steps

A level B representation needs to specify the expression and decision options of a case step, as shown in Figure 27. The expression is specified in a similar manner to the specification of decision criteria, as shown in

Figure 26. The decision option's condition values (e.g., true) specify the alternative values of the expression.

Choice Steps

The choice step has similar attributes as the case step, but it does not have an expression attribute. Instead, the different decision option condition values are expressed as criteria that evaluate to true, false, or unknown, as shown in

Figure 28.

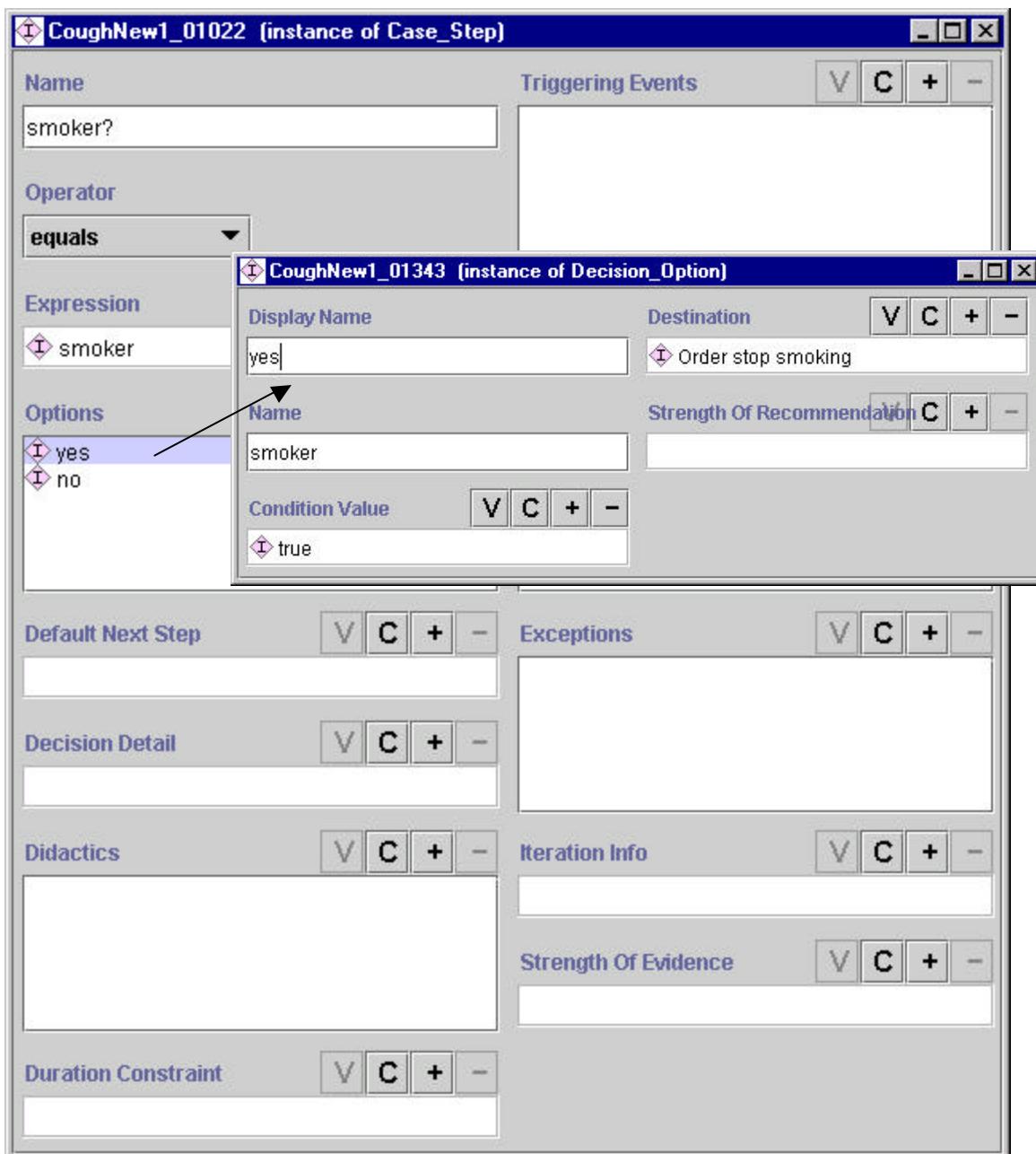


Figure 27. The level B specification of a case step

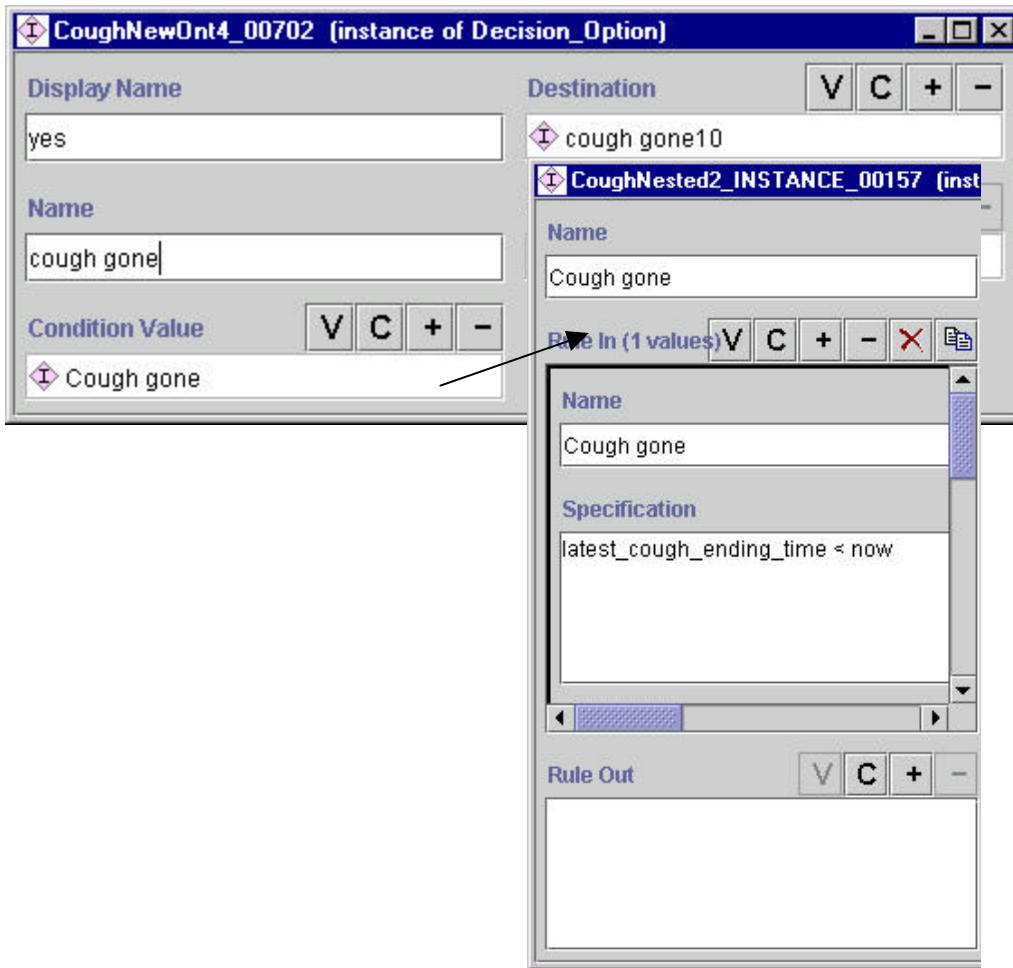


Figure 28. A decision option of the choice step “cough gone?”

Specifying guideline objects

Guidelines need to be aware of the data items that they use (in decision criteria and action specifications). They also should define parameters that are passed to them and/or that they pass out. For example, the treatment guideline, whose algorithm is shown in Figure 7, defines several data items in its data_item_list slot, as shown in Figure 29. Some of these data items (e.g., pregnancy) are not parameters that need to be passed to other guidelines. The parameters_passed slot specifies the parameters that need to be passed in or out of other (sub) guidelines. For example, the treatment guideline has a sub-guideline called “cessation of smoking/ACEI”, whose algorithm is shown in Figure 8. The sub-guideline needs to “read” the following attributes from the outer treatment guideline: ACEI, smoker, cough, and X_Ray_done. Therefore, the treatment guideline needs to export these parameters out (to the sub-guideline). The cessation of smoking/ACEI sub-guideline may perform an X-ray. It will then need to change (export out) the values of the following data items: X-Ray_done and X-Ray_result. The treatment guideline will therefore need to import (read in) these two data items. Similarly, Further_Test_Can_Be_Ordered needs to be passed in and out between the treatment and

cessation of smoking/ACEI guidelines. The Time_Of_Stopping_Smoking_ACEI_Order needs to be passed out of the cessation guideline and into the treatment guideline.

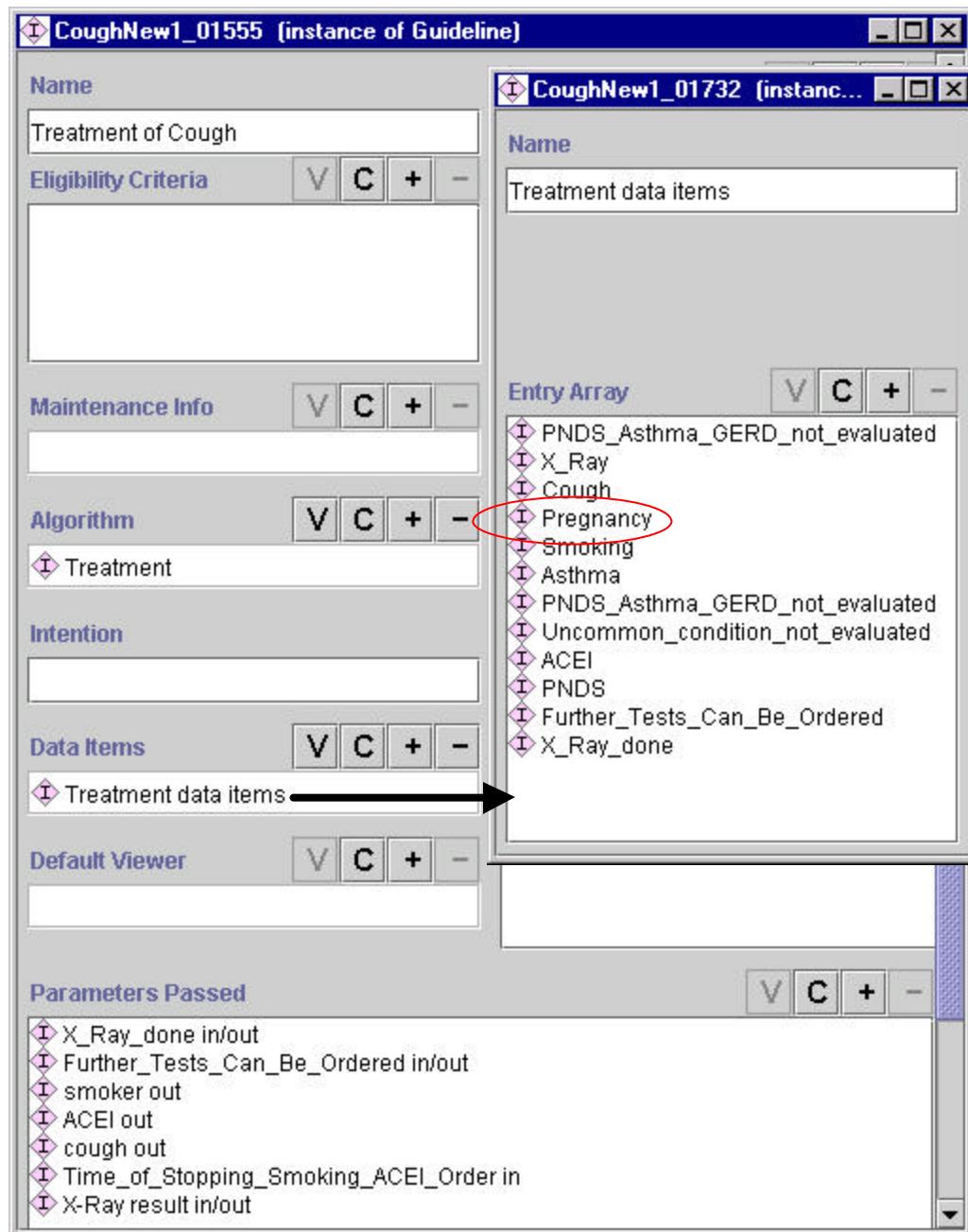


Figure 29. Specifying the level B attributes of a guideline object