

ATHENA-CDS System Guide
(Assessment and Treatment for Healthcare:
Evidence-based Automation - Clinical
Decision Support System Guide)

Documentation for the ATHENA-CDS SYSTEM
Developed and Implemented in the
ATHENA-CDS Project

- Draft -

Document authors:

Robert W. Coleman, MS Pharm¹
Susana Martins, MD¹
Samson W. Tu, MS²
Mary K. Goldstein, MD^{1, 2, 3}

1. Geriatric Research Education and Clinical Center (GRECC),
VA Palo Alto Health Care System, Palo Alto, California
2. Stanford BioMedical Informatics Research (BMIR), Department of Medicine,
Stanford University, Stanford, California
3. Center for Primary Care and Outcomes Research (PCOR),
Department of Medicine, Stanford University, Stanford, California

Views expressed are those of the authors and not necessarily those of the Department of
Veterans Affairs

Note: ATHENA-CDS was previously known as “ATHENA-DSS”. ATHENA-CDS refers to a system that includes multiple knowledge bases in different clinical domains. This Manual was developed for the initial system, which focused on hypertension (HTN). We currently use “ATHENA-CDS” to refer to the system as a whole, and use “ATHENA-HTN” to refer to the hypertension system specifically. We have pilot tested and published about ATHENA-Opiate Therapy (OT), developed to assist health professionals with safe management of opiates in chronic pain. We have also developed ATHENA-CDS knowledge bases for heart failure (HF), diabetes mellitus (DM) and other conditions.

ATHENA-CDS has evolved over the past decade. Methods used early on have in some cases been superseded by more recent development. Many descriptions and illustrations are from older versions of programs or evidence sources and are shown for purposes of illustration only. Most recently, a new user interface, new methods of connection to the VA Regional Data Warehouse, and other extensions have been developed under a VHA Innovations Award.

Nothing in this Manual should be taken as medical advice. Medical information is displayed for purposes of illustration only.

No actual patient information is contained in this Manual. Any patient information displayed, such as blood pressure or list of medications or diagnoses, is synthetic data not linked to an actual patient.

For further information, contact:

Mary K. Goldstein, MD
Director, Geriatrics Research Education and Clinical Center (GRECC)
VA Palo Alto Health Care System

Phone: 650-493-5000 extension 6-2105
Email: mary.goldstein@va.gov

Alternate contact: GRECC administrative staff 650-858-3933

TABLE OF CONTENTS

I. INTRODUCTION.....	6
I.1. ATHENA-CDS SYSTEM	6
I.1.1. What Is ATHENA-CDS SYSTEM?	6
I.1.2. ATHENA-CDS's Technical Features.....	7
I.1.3. ATHENA-CDS Deployment	7
I.2. About This Guide: Purpose and Intended Audience	8
ATHENA-CDS FUNCTIONS AND ARCHITECTURE	10
I.3. System Functions and Display.....	10
I.3.1. Recommendations Tab.....	10
I.3.2. Precautions and Assumptions Tabs	13
I.3.3. Lifestyle, Adherence, and Glossary Tabs.....	15
I.3.4. Supporting Documents.....	18
I.3.5. BP-Prescription Graphs	19
I.3.6. Patient Summary.....	20
I.4. System Architecture	21
I.4.1. EON System Architecture	22
I.4.2. ATHENA-CDS System Architecture.....	26
II. ATHENA-CDS INSTALLATION	34
II.1. Requirements.....	34
II.1.1. System and Server Requirements.....	34
II.1.2. Extraction Process Requirements.....	34
II.1.3. Monitoring, Archiving Requirements	35
II.1.4. Client-side Requirements.....	35
II.1.5. Security and Permissions.....	35
II.2. VistA Installation.....	35
II.2.1. ATHENA-CDS Data Extraction	35
II.3. SERVER INSTALLATION	38
II.3.1. ATHENA-CDS Distribution.....	38
II.3.2. Data Transformation Service (DTS).....	39
II.3.3. SQL Server Jobs	41
II.4. CLIENT INSTALLATION	42
II.5. SERVER-SIDE OPERATION	43
II.6. CLIENT-SIDE OPERATION	44
II.7. MONITORING AND ARCHIVING.....	44
II.7.1. Post-installation Tests	45
III. KNOWLEDGE RESOURCES: STRUCTURE AND UPDATE PROCEDURE	47
III.1. PROTÉGÉ.....	47
III.1.1. Protégé Knowledge Model	48
III.1.2. Metaclass in Protégé	50
III.1.3. Constraints in Protégé: Facets and PAL Constraints	53
III.1.4. Protégé's File Format.....	54
III.1.5. Protégé User Interface	55
III.1.6. Two Common Tasks	65
III.2. EON MODELS AND THE ATHENA-CDS KNOWLEDGE BASE.....	70
III.2.1. Patient Data Model.....	73
III.2.2. Medical Concept Model	74
III.2.3. EON Guideline Model.....	83
III.2.4. Expressions	114
III.3. GUIDELINE INTERPRETER'S USE OF THE KNOWLEDGE BASE	125

<i>III.3.1. Overview</i>	125
<i>III.3.2. Structure of Guideline Advisories</i>	129
<i>III.3.3. Generation of Guideline Advisories</i>	134
III.4. TEXTUAL KNOWLEDGE SOURCES	137
<i>III.4.1. Supporting Material</i>	138
<i>III.4.2. Lifestyle, Glossary, and Adherence Tabs</i>	148
III.5. UPDATING KNOWLEDGE SOURCES IN ATHENA-CDS SYSTEM	149
<i>III.5.1. Update the Rules Document</i>	149
<i>III.5.2. Update the Knowledge Base in Protégé</i>	149
<i>III.5.3. Update the Drugs Table (Mapping Table) in the SQL Server Database</i>	200
<i>III.5.4. Update the Tabs in the ATHENA-CDS Hypertension Advisory</i>	202
<i>III.5.5. Prepare and Run Test Cases</i>	202
<i>III.5.6. Update Deployed System</i>	202
III.6. ATHENAEUM MAPPING	203
IV. APPENDICES	210
IV.1. ATHENA-CDS AND ATHENEON DB SCHEMA.....	210
IV.2. M EXTRACTION PROGRAM	210
IV.3. SQL TRANSFORMATION CODE	210
IV.4. ATHENA-CDS TRAINING MANUAL.....	210
IV.5. ATHENA-CDS CONVERTOR	211
IV.6. GLOSSARY	237
V. ACKNOWLEDGEMENTS	241
VI. CONTACT INFORMATION	242
VII. REFERENCES	243

Preface

We believe that the clinical content of the Clinical Decision Support (CDS) should be based on an appropriate authoritative source, which should be identified by the health care system that is implementing the CDS. We take the “S” in CDS seriously, that is, we believe that the CDS should support the health professional, not push or attempt to replace the health professional. The CDS should present information that will be helpful to the decision-maker, not attempt to make the decision for them. It should assume a professional approach and recognize the importance of clinical judgment, presenting options for consideration with the final decision to be made based on clinical judgment in consideration of patient goals and preferences. Good clinical care calls for individualization of therapy. Computers should do things that computers do well – search, retrieval, calculation – and leave judgment to the human decision-maker. In order to be able to do the above, it is useful to have a computing infrastructure that allows for rapid entry and updating of clinical knowledge, revision of message content, etc, so that the clinical content encoded into computable format can be tailored to the content desired by the appropriate clinical authority.

I. Introduction

I.1. ATHENA-CDS SYSTEM

I.1.1. What Is ATHENA-CDS SYSTEM?

ATHENA-CDS SYSTEM (Assessment and Treatment of Hypertension: Evidence-based Automation, Clinical Decision Support System) [1] is a clinical decision-support system that generates guideline-based recommendation for hypertension using Stanford Center for Biomedical Informatics Research (BMIR) EON architecture. ATHENA-CDS SYSTEM is currently configured for installation in health-care systems to provide patient-specific decision support at points of care. The end-users of the system are clinicians who are making decisions on the management of care for patients who have hypertension.

ATHENA-CDS SYSTEM—also referred to simply as ATHENA-CDS—takes patient information from an electronic medical record (EMR), combines it with encoded knowledge of hypertension, and generates patient-specific recommendations, explanations and evidence-based education. ATHENA-CDS uses guideline knowledge based on the widely endorsed *Sixth Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure* (JNC 6) [2]¹ and on Department of Veteran Affairs guidelines [3]. ATHENA-CDS displays recommendations in a pop-up window over the EMR cover page for specified patients at selected times.² It makes recommendations to add, substitute, delete, or change drug dosages, according to how well blood pressures are controlled and to comorbid diseases that represent “compelling” (per JNC 6) indications (e.g., beta adrenergic receptor antagonists after myocardial infarction), relative indications (e.g., thiazides for patients with osteoporosis), relative contraindications (e.g., beta adrenergic receptor antagonists for patients with depression), and strong contraindications (e.g., beta adrenergic receptor antagonists for patients with asthma). The system also displays messages that detail precautions, assumptions the system makes in generating recommendations, and additional explanatory text [1, 4].

ATHENA-CDS is part of a larger research program that has developed methods for and evaluated the implementation of clinical practice guidelines, using hypertension as a model.

¹ ATHENA-CDS has been updated to incorporate the recommendations of JNC 7.

² The pop-up window is currently displayed for clinicians who are designated ATHENA-CDS users and who access the records of patients who have been diagnosed with hypertension, meet a set of inclusion criteria, and have appointments at primary care clinics within a five-day window.

I.1.2. ATHENA-CDS's Technical Features

ATHENA-CDS SYSTEM uses the EON architecture for developing guideline-based decision-support systems [5]. The EON architecture, created at BMIR, provides ATHENA-CDS with the platform-independent technology to:

1. encode hypertension guidelines in computer-interpretable form [6, 7];
2. query patient data from a temporal database that provides the capabilities to manage and query time-oriented data [8]; and
3. apply guideline knowledge to data for individual patients, through the EON Guideline Interpreter [9].

ATHENA-CDS uses Protégé [10] as the tool to build and maintain its knowledge base.³ Also developed at Stanford, Protégé provides an interface that clinician-managers can easily use to browse and update the guideline knowledge base.

A platform-independent system, ATHENA-CDS is designed for integration with legacy patient data systems, such as the system used by clinics at Department of Veteran Affairs (VA) medical centers. VA clinics' data system is composed of the Veterans Health Information Systems and Technology Architecture (VistA) [12] and its user interface, Computerized Patient Record System-Graphical User Interface (CPRS-GUI; [13-15]. ATHENA-CDS has been implemented on VistA and integrated into the CPRS, displaying recommendation pop-ups when appropriate (see Subsection II.1.1).

I.1.3. ATHENA-CDS Deployment

ATHENA-CDS has been deployed and evaluated in a randomized, controlled trial in the primary care clinics at three VA medical centers—VA Palo Alto Health Care System (VAPAHCS), San Francisco VA Medical Center, and Durham VA Medical Center. The trial lasted 15 months, ending in July 2003. The trial evaluated the impact of providing ATHENA-CDS's guideline-based decision support for patient care. The primary outcome measures for the trial were patients' blood pressure control and clinicians' concordance with guideline-based drug recommendations.

Testing of system installation is currently underway in additional VA medical centers.

³ A knowledge base is a collection of statements—expressed in a computer-interpretable knowledge-representation language—that represents assertions about the world (11. Russell, S. and P. Norvig, *Artificial Intelligence: A Modern Approach*. 2nd edition ed. 2003, Upper Saddle River, New Jersey: Pearson Education, Inc. In the case of ATHENA-CDS, as will be described later in more detail, the knowledge base statements include assertions about the structure of computer-interpretable guidelines, medical concepts, and detailed clinical knowledge on the management of hypertension.

To deploy ATHENA-CDS at a VA medical center, clinical managers who are expected to decide whether or not to implement and monitor ATHENA-CDS may want to browse the knowledge in the system and test the system in their own clinical practice. Section IV of this manual addresses the structure of the encoded knowledge, how to update it, and how to test updates.

Clinical managers at each medical center could, in theory, change the ATHENA-CDS Knowledge Base or other aspects of the system for their medical center. For instance, a clinician-administrator can modify the hypertension knowledge in the knowledge base (e.g., formulary preferred drugs or the strength of indications and contraindications). However, to date, the clinical managers outside VAPAHCS have opted to leave all maintenance of the ATHENA-CDS Knowledge Base to the developers at VAPAHCS. This relieves them of taking on the responsibility and training entailed by testing the system for accuracy and installing updates to the system.

In addition to choices about encoding the guideline knowledge, system implementation involves decisions requiring clinical input such as when to display recommendations. For instance, a clinical manager can modify the triggers for popping up ATHENA-CDS's advisory window. She may consider questions such as: Do clinicians want to see the recommendations for all patients with primary hypertension who meet the inclusion criteria? Or only those patients whose blood pressure above target? Do they want to see recommendations only on the day of a clinic appointment? Or do they want to see recommendations within a window of a certain number of days surrounding a clinic appointment? The current system is programmed to pop up an advisory window when a participating clinician in a primary care clinic logs onto CPRS-GUI and opens the record of an eligible patient within a five-day window of a previously scheduled primary care clinic appointment. However, the system can be reprogrammed to function differently in different settings, as desired and appropriate.

In contrast, clinician end-users will not generally make such changes to the ATHENA-CDS Knowledge Base, but can use ATHENA-CDS to receive guideline-based advisories on the management of care for patients who have hypertension. They may want to use ATHENA-CDS to try what-if scenarios, modifying a patient's clinical data through the ATHENA-CDS user interface to see what advisories the system would generate.⁴

I.2. About This Guide: Purpose and Intended Audience

The purpose of this guide is to provide an overview of the functions and architecture of ATHENA-CDS SYSTEM, as well as a manual for installing and maintaining the system and for updating the knowledge resources used by ATHENA-CDS. It is aimed at three audiences:

1. clinical investigators and managers who need a conceptual understanding of the capability, architecture, and operational requirements of ATHENA-CDS SYSTEM;
2. information technology personnel at VA sites responsible for installing, testing, operating, and monitoring ATHENA-CDS SYSTEM; and

⁴ The modified clinical data are not saved to CPRS at present.

3. clinicians who may have to modify the ATHENA-CDS Knowledge Base.

This guide is divided into three main parts. Anyone responsible for maintaining and supporting ATHENA-CDS should read Section II. It describes the functions and architecture of the ATHENA-CDS SYSTEM, including its components, its construction using EON architecture, and the flow of information in generating guideline-based recommendations. It gives a general idea of how various components work and affect each other. Section III is a manual that details steps in installing, testing, operating, and monitoring the operations of ATHENA-CDS SYSTEM. Some of this section assumes familiarity with Microsoft SQL Server, which is the database management tool used by the ATHENA-CDS system. Section IV describes the ATHENA-CDS knowledge resources and gives examples of how to make common changes to them. The ATHENA-CDS Knowledge Base is built on the EON Guideline Model and was constructed using the Protégé tool. Protégé and the EON Guideline Model are covered in Subsections IV.1 and IV.2, respectively.

Separate documents are available for clinicians who are end-users of the ATHENA-CDS SYSTEM (see Subsection V.4, Appendix: ATHENA-CDS Training Manual).

II. ATHENA-CDS Functions and Architecture

This section summarizes the display window and ATHENA-CDS functions from the perspective of the user (Subsection II.1), then describes the architecture of the overall system and how components of the system work together to generate these functions (Subsection II.2).

II.1. System Functions and Display

ATHENA-CDS delivers its decision-support functions through a pop-up window that appears on-screen within the Computerized Patient Record System (CPRS) window. When an eligible health-care provider accesses the electronic medical record of a patient diagnosed with hypertension and medical conditions that satisfy ATHENA-CDS's inclusion criteria, the window appears superimposed on the CPRS coversheet. From the first pop-up window (Figure 1), the user has the option to bring up a second pop-up window that displays a summary of patient information relevant to the management of hypertension.

At the top of the first pop-up window, shown in Figure 1, are the name (not shown in the figure, but displayed in the deployed system) and Social Security Number (also not shown in the figure, but displayed in the deployed system) of a dummy patient. Below the demographic information are the most recent blood pressures, used to generate the recommendation.

ATHENA-CDS computes the appropriate target blood pressures based on the comorbidities of the patient (e.g., diabetes diagnosis) and displays the guideline goal systolic blood pressure (SBP) and diastolic blood pressure (DBP). It compares the most recent blood-pressure measurements to the targets in order to determine whether the patient's blood pressures are under control. A message flags when blood pressure is not under control, as shown in red in Figure 1. An ATHENA-CDS user has the option of entering today's decision-making diastolic and systolic blood pressures (BP taken by a nurse, a BP recheck, home BPs, or an aggregate of these) and getting an updated advisory (i.e., set of recommendations).

The rest of ATHENA-CDS's decision-support functions are organized in a number of tabs: Recommendations, Precautions, Assumptions, Lifestyle, Adherence, Glossary, and BP-Prescription Graphs. Additional explanations are provided through ATHENA-CDS's evidence-based supporting documents (see Subsection IV.4.1). The tabs, supporting documents, and patient information summary are described in the following subsections.

II.1.1. Recommendations Tab

The Recommendations tab presents recommendations to bring the health management of the selected patient into compliance with the encoded hypertension guidelines. This tab appears as

default, it includes a primary recommendation message section, a table of drug-related recommendations with corresponding reasons for the suggested changes, an area for the clinician to enter comments, and buttons for the clinician to indicate whether the recommendations were taken into consideration.

Primary recommendation: The primary recommendation, consisting of the most important advisory messages, is seen at the top of the tab page. For a patient with blood pressure that does not seem to be controlled, a message may say “Consider INTENSIFYING drug treatment; BP ELEVATED based on most recent BP” (see Figure 1). This example tells the clinician that the patient's blood pressure is not at a level the guidelines consider to be acceptable for a patient in the given condition, and that the patient's prescription(s) should be modified. In this case, the details are provided in the drug recommendation table, below the primary recommendation table.

Drug recommendation: The recommendations for modifying drug treatment in accord with the primary recommendation(s) are shown in the table below the primary recommendation. In Figure 1, “Increase dosage of lisinopril,” “Add DHP Calcium Channel Blocker,” and “Add Cardioselective Beta Blocker” are offered as guideline-concordant therapeutic possibilities for a clinician to consider. Clicking on the Info buttons next to these suggestions brings up additional information that may be important in evaluating each possibility.

Entries in the Reasons column encapsulate patient-specific considerations involved in each therapeutic recommendation. Icons for Compelling Indication, Relative Indication, Strong Contraindication, Relative Contraindication, and Adverse Events indicate the status of each consideration. For example, DHP calcium channel blocker is recommended because “Isolated Systolic Hypertension” is a compelling indication for its use. Cardioselective beta blocker has a relative indication of the patient's coronary artery disease, but the patient's obstructive pulmonary disease is a relative contraindication for its use.

The Feedback buttons in the right column give a clinician the opportunity to supply coded reactions to each drug recommendation (e.g., “Patient had not tolerated this drug in the past” or “Patient is not adhering to medications already prescribed”; see Figure 2).

Comment section: A textbox, shown below the drug recommendation table in Figure 1, allows clinicians to enter free-text comments. The comments can help ATHENA-CDS developers understand what clinicians find useful or problematic in ATHENA-CDS, so they can improve the system.

Check box for DO NOT DISPLAY: Until this box is checked the advisory appears in a pop-up window each time the patient's record is accessed by an appropriate clinician within the 5 day window of the clinic appointment. Some clinicians review charts prior to clinic visits and want to see the advisory again when the patient is in clinic and an updated BP has been taken. However,

they do not want to continue to see the advisory when accessing the patient record for other reasons after they have addressed the blood pressure.

Disposition buttons: Black buttons at the bottom of the Recommendations tab (*Recommendations considered, Not Read, Not a clinical priority today* in Figure 1) provide a quick way for a clinician to close the ATHENA-CDS pop-up window, while also supplying feedback on their use of the ATHENA-CDS SYSTEM.

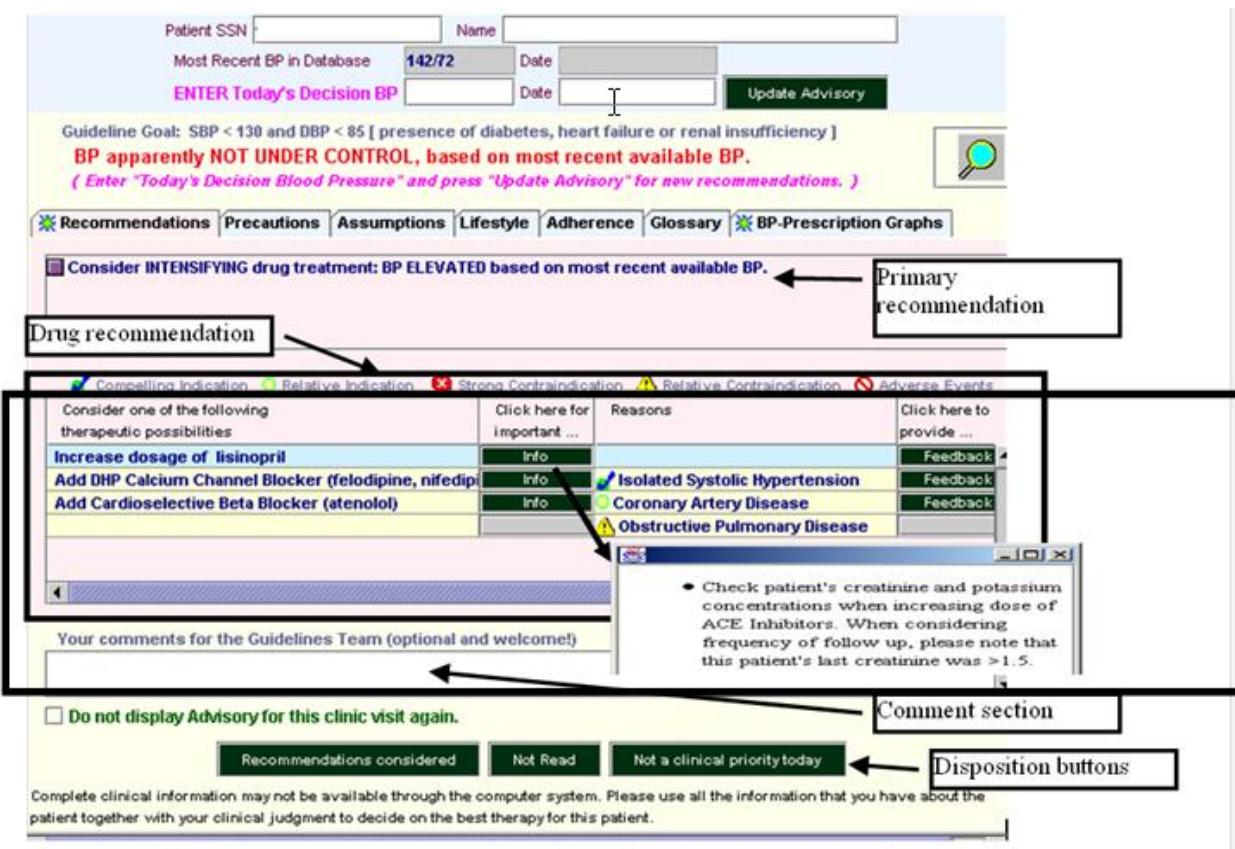


Figure 1 - ATHENA-CDS hypertension advisory window, showing details of the Recommendations tab

The screenshot shows the ATHENA-CDS software interface. At the top, there are input fields for Patient SSN, Name, Most Recent BP in Database (142/72), Date, and ENTER Today's Decision BP. Below these is a message: "Guideline Goal: SBP < 130 and DBP < 85 [presence of diabetes, heart failure]. BP apparently NOT UNDER CONTROL, based on most recent blood pressure reading. Enter 'Today's Decision Blood Pressure' and press 'Update Advisor'." A navigation bar below includes Recommendations, Precautions, Assumptions, Lifestyle, and Adherence tabs. The Recommendations tab is active, displaying a section titled "Consider INTENSIFYING drug treatment: BP ELEVATED based on most recent blood pressure reading." It lists three therapeutic possibilities: "Increase dosage of lisinopril" (selected), "Add DHP Calcium Channel Blocker (felodipine, nifedipine)", and "Add Cardioselective Beta Blocker (atenolol)". Each option has an "Info" button and a small status icon. A "Feedback" dialog box is overlaid on the screen, prompting the user to check boxes for various reasons related to the treatment recommendation. The "OK" and "Cancel" buttons are at the bottom of the dialog.

Figure 2 - ATHENA-CDS Recommendations tab, showing Feedback form

II.1.2. Precautions and Assumptions Tabs

The Precautions tab (Figure 3) contains patient-specific comments and warnings about the patient's medical conditions and about the antihypertensive agents that have already been prescribed.

Patient SSN		Name	
Most Recent BP in Database	142/72	Date	
ENTER Today's Decision BP		Date	
<input type="button" value="Update Advisory"/>			

Guideline Goal: SBP < 130 and DBP < 85 [presence of diabetes, heart failure or renal insufficiency]
BP apparently NOT UNDER CONTROL, based on most recent available BP.
(Enter "Today's Decision Blood Pressure" and press "Update Advisory" for new recommendations.)



Recommendations Precautions Assumptions Lifestyle Adherence Glossary BP-Prescription Graphs

Patient is taking furosemide or a loop diuretic and appears to have neither CHF nor renal insufficiency.
 Hydrochlorothiazide is generally a more efficacious diuretic in the management of hypertension.

Both thiazide diuretics and dihydropyridine calcium channel blockers are effective in lowering the risk of stroke in ISH, but there have been no head to head comparisons. Thiazides are far less expensive.

VA Hypertension guidelines recommend beta blockers for patients with coronary artery disease. Current evidence strongly supports beta blockers for secondary prevention of myocardial infarction however for primary prevention it remains unclear if they are superior to thiazide diuretics.

Strictly speaking, ISH is defined as pre-treatment SBP ≥ 140 and DBP < 90 .

This patient has clinically manifest cardiovascular disease or target organ damage (Risk Group C). The largest benefit of anti-hypertensive therapy is seen in patients with the highest baseline risk of cardiovascular disease, therefore this patient should be considered for prompt pharmacologic therapy, even if the BP is in the high-normal range. Appropriate lifestyle modification and attention to other reversible cardiovascular risk factors is also strongly recommended.

Warning: creatinine value is greater than 1.5 mg/dL, upper limit of normal for males.

Figure 3 - ATHENA-CDS Precautions tab

The Assumptions tab (Figure 4) contains messages that state ATHENA-CDS's boundaries, limitations and assumptions. For example, ATHENA-CDS does not deduce any diagnoses from procedure reports (e.g., EKG, X-Ray). Furthermore, if no adverse reaction is listed for a currently prescribed antihypertensive drug, then ATHENA-CDS assumes that there is none.

Patient SSN	Name
Most Recent BP in Database	142/72
ENTER Today's Decision BP	Date
	Date
Update Advisory	

Guideline Goal: SBP < 130 and DBP < 85 [presence of diabetes, heart failure or renal insufficiency]
BP apparently NOT UNDER CONTROL, based on most recent available BP.
(Enter "Today's Decision Blood Pressure" and press "Update Advisory" for new recommendations.)



Recommendations Precautions Assumptions Lifestyle Adherence Glossary BP-Prescription Graphs

- If this patient does not have hypertension, or if this patient already has a diagnosis of secondary hypertension on the problem list, please take a moment to tell us that, using the "comment" box at the bottom of the recommendations screen. You can also send us an email, if you like, on DHCP: VA Palo Alto: Parisa Gholami (Gholami.Parisa@Palo-Alto.va.gov); VA San Francisco: Michelle Odden (Michelle.Odden@med.va.gov) or VA Durham: Melinda Orr (orr00007@mc.duke.edu). You can open DHCP to send email without closing your CPRS-GUI window.
- This program does NOT provide recommendations for patients with SECONDARY HYPERTENSION, NARCOLEPSY, ASCITES, SPINAL-CORD INJURY , or IDIOPATHIC HYPERTROPHIC SUBAORTIC STENOSIS. If this patient has one or more of these diagnoses, please ADD it to the patient's PROBLEM LIST.
- A BP entered into the ATHENA system today will not be entered into the patient's electronic medical record. The program will use the BP to update the recommendations on the screen, but it will not write the BP to VistA, so the BP will not show up in CPRS-GUI. You can enter additional BP's into the patient's permanent medical record by clicking 'BP' in the Vitals box on the CPRS-GUI main screen. Click 'Enter Vitals' in the top left of the screen, select the appropriate appointment location, and then enter the new blood pressure at the prompt. Any BP you add to the patient's medical record today will be available to this program at the patient's next visit.
- Diagnoses from procedure reports (EKG, X-ray, etc.) are not recognizable to ATHENA DSS. ATHENA pulls diagnoses from the problem list and the list of diagnoses for each visit on the encounter forms. The Patient Summary in ATHENA shows the diagnoses for this patient that are known to the program. If patient has a diagnosis that is not included in the

Figure 4 - ATHENA-CDS Assumptions tab

II.1.3. Lifestyle, Adherence, and Glossary Tabs

The Lifestyle, Adherence, and Glossary tabs respectively give:

1. evidence supporting lifestyle changes that may have an impact on hypertension, regarding factors such as diet, smoking cessation, exercise, and alcohol consumption (Figure 5);
2. general suggestions of ways to improve patient adherence to a therapeutic regime (Figure 6); and
3. a glossary providing definitions for terms used in the pop-up windows (Figure 7).

These three tabs open static pages containing general content not custom tailored to the specific conditions of a patient.

Most Recent BP in Database **142/72** Date
ENTER Today's Decision BP Date

Guideline Goal: SBP < 130 and DBP < 85 [presence of diabetes, heart failure or renal insufficiency]
BP apparently NOT UNDER CONTROL, based on most recent available BP.
(Enter "Today's Decision Blood Pressure" and press "Update Advisory" for new recommendations.)



Recommendations **Precautions** **Assumptions** **Lifestyle** **Adherence** **Glossary** **BP-Prescription Graphs**

Topic	Recommendation
Weight Reduction	Modest weight reductions of 3-9% of body weight may lead to reduction in blood pressure (table 1) in obese people with hypertension. Trials that allowed adjustment of antihypertensive regimens found that lower doses and fewer antihypertensive drugs were needed in the weight reduction groups compared with control group [1].
Smoking Cessation	There is strong evidence that smoking cessation reduces risk of cardiovascular disease.[2] There is no direct evidence that stopping smoking reduces blood pressure in people with hypertension [11].
Alcohol Consumption	Increased alcohol consumption (>2 standard drinks/day) is associated with increased blood pressure and risks of hemorrhagic stroke, cancer, accidents, injuries, diseases of the liver, pancreas and
Exercise/Physical Activity	Regular aerobic exercise is associated with a lower all cause and cardiovascular mortality in hypertensive men[4]. Blood pressure may decrease as a result of the exercise program (table 1). [5, 6].
Salt Restriction	Dietary sodium restriction reduces blood pressure (table 1) in hypertensive patients, especially in older individuals. [7, 8].
Diet	A diet rich in fruits, vegetables, and low-fat dairy products and with reduced saturated and total fat decreases blood pressure (table 1)[9]. Changing dietary habits may have multiple benefits.
Calcium and	There is no evidence to support the use of calcium (table 1) [10] or magnesium supplements [11] to

Figure 5 - ATHENA-CDS Lifestyle tab

Patient SSN		Name	
Most Recent BP in Database	142/72	Date	
ENTER Today's Decision BP		Date	
Guideline Goal: SBP < 130 and DBP < 85 [presence of diabetes, heart failure or renal insufficiency] BP apparently NOT UNDER CONTROL, based on most recent available BP. <i>(Enter "Today's Decision Blood Pressure" and press "Update Advisory" for new recommendations.)</i>			
			
Recommendations Precautions Assumptions Lifestyle Adherence Glossary BP-Prescription Graphs			
General Guidelines to Improve Patient Adherence to Antihypertensive Therapy* <ul style="list-style-type: none"> 1. Be aware of signs of patient nonadherence to therapy. 2. Establish the goal of therapy early: to reduce BP to non-hypertensive levels with minimal or no adverse effects. 3. Educate patients about the disease, and involve them and their families in its treatment. Have them measure blood pressure at home. 4. Maintain contact with patients; consider telecommunication. 5. Encourage lifestyle modifications. 6. Integrate pill taking into routine activities of daily living. 7. Prescribe medications according to pharmacologic principles, favoring long-acting formulations. 8. Be willing to stop unsuccessful therapy and try a different approach. 9. Anticipate adverse effects and adjust therapy to prevent, minimize, or ameliorate side effects. 			
<small>* As displayed in the VHA/DOD Clinical Practice Guideline for Diagnosis and Management of Hypertension in the Primary Care Setting</small>			

Figure 6 - ATHENA-CDS Adherence tab

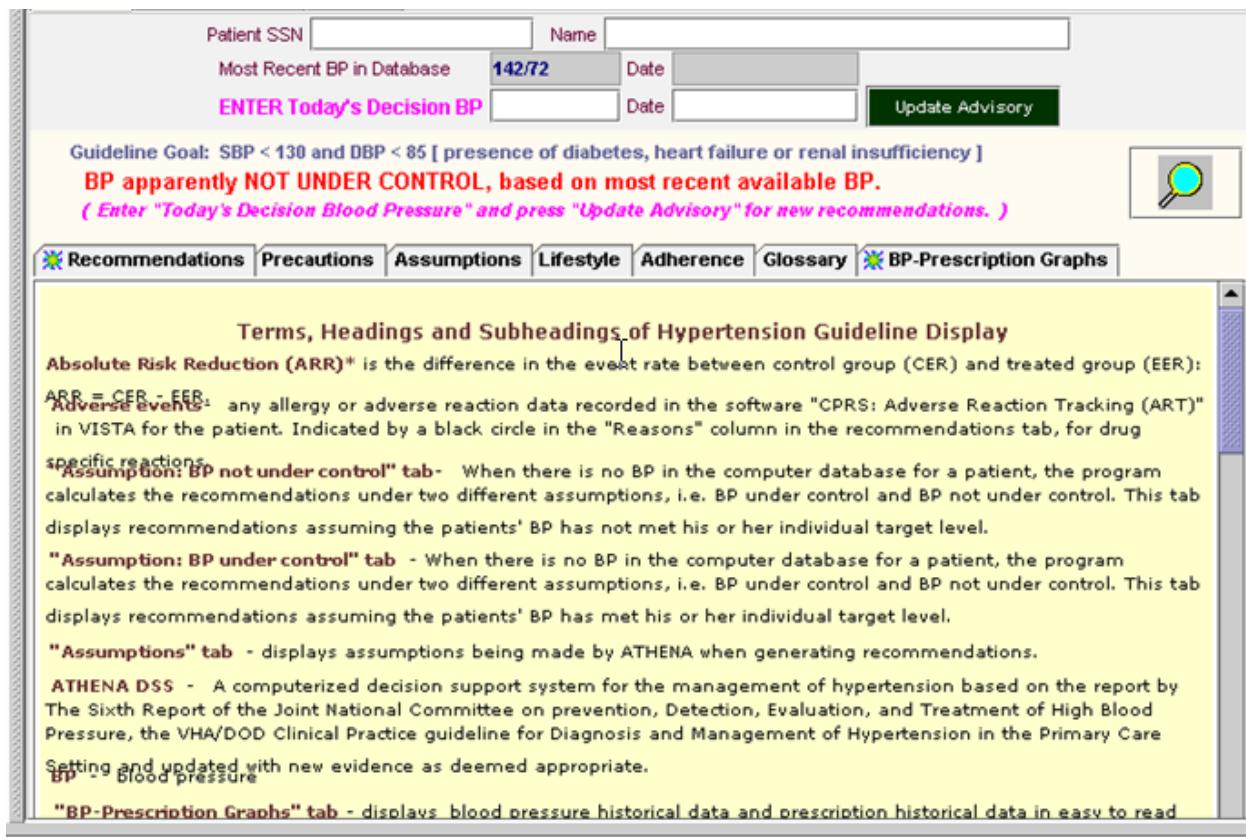


Figure 7 - ATHENA-CDS Glossary tab

II.1.4. Supporting Documents

The ATHENA-CDS system provides explanatory supporting documents on the use of specific drugs in the presence of certain comorbidities. It activates these documents when a user drags the magnifying glass icon (🔍) and drops it on top of a compelling indication (e.g., Isolated Systolic Hypertension) for the use of an antihypertensive agent (e.g., DHP Calcium Channel Blocker). These supporting documents are provided in HTML format and accessible through a tab-based interface (Figure 8). They include studies, guidelines, and evidence supporting the particular recommendations made for a patient. The Short Summary tab (see Figure 8) provides a summary of the evidence related to the use of a class of antihypertensive agents in the presence of a comorbidity. Subsequent tabs present relevant literature from *ACP Journal Club*, *Clinical Evidence*, *The Sixth Report of the Joint National Committee Guidelines on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure (JNC 6)*, *PubMed*, and the *VA Guidelines*.

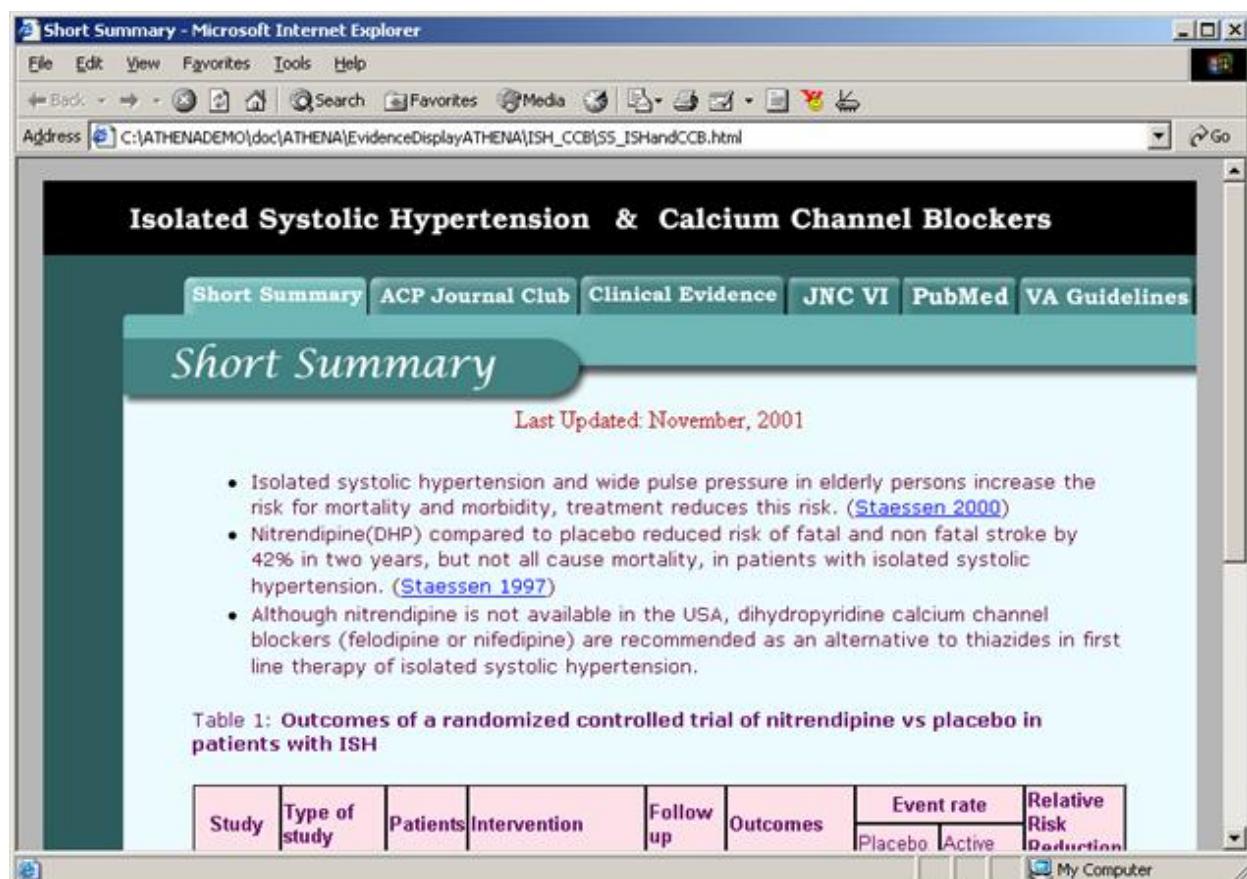


Figure 8 - Short Summary portion of ATHENA-CDS's supporting documentation on the use of a class of antihypertensive agents in the presence of a comorbidity. The other tabs present relevant literature in ACP Journal Club, Clinical Evidence, The JNC 6 Guidelines, PubMed, and the VA Guidelines.

II. 1.5. BP-Prescription Graphs

The BP-Prescription Graphs (Figure 9) show past blood pressure records and prescription data, using a common time axis. Blood pressure data from three years before the current time to present are plotted together with the prescription history of antihypertensive agents. Drug doses are displayed in boxes at initiation of a drug or a change in dose. The extents of the prescription intervals are computed from prescription information and refill history. Gaps in the lines of the prescription history indicate that a patient should have run out of medications if he or she has adhered to the instructions in the prescription. Accordingly, gaps may reflect lack of medication adherence. A user can scroll the two graphs in lock step to see blood pressures and prescriptions recorded earlier. Thus, as shown in Figure 9, a user can see the prescription intervals for diltiazem SA, which was prescribed before, but not after, October 1999, by scrolling the graph to the left.

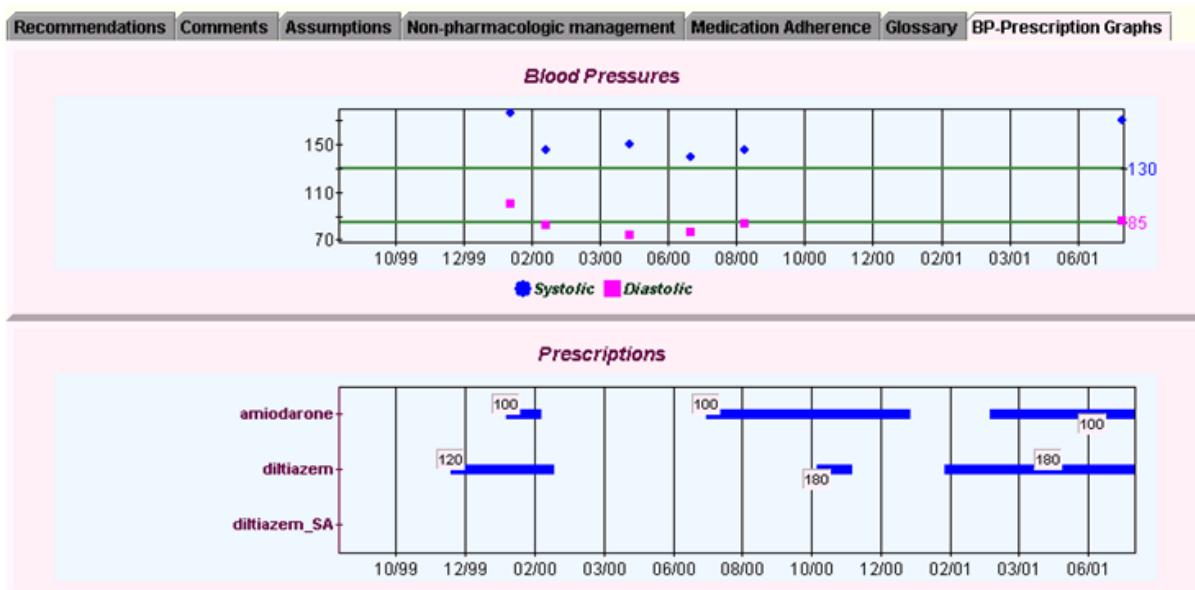


Figure 9 - ATHENA-CDS BP-Prescription Graphs

II.1.6. Patient Summary

By clicking the “Patient Summary” button, ATHENA-CDS users can bring up a Patient Summary pop-up window that displays an abstract of patient data relevant to hypertension management (see Figure 10).

The window organizes the data into categories such as Labs, Vitals, Related Drugs (currently active drugs for managing hypertension), Related Comorbidities (the patient's diseases and syndromes that may affect the management of hypertension), and ADRs/Allergies (adverse drug reactions or drug allergies the patient may have). Data shown in this window are taken into account when the system computes recommendations for the patient’s treatment.

By displaying the active prescriptions, ATHENA-CDS SYSTEM encourages medication reconciliation. The clinician can review the active prescriptions and update the status of medications, for example if the clinician asked the patient to stop taking the meds but did not discontinue it in the prescription file.

If the ATHENA-CDS user has additional patient information that is not in the VA database, or if the user wishes to see what ATHENA-CDS would recommend based on hypothetical data, she can change data by using the +, -, and -> buttons to bring up a dialog form. For example, she can add a new drug to the Related Drugs list, add a new comorbidity to Related Comorbidities, or change the dosage of an existing drug. By clicking the Update Advisory button at the bottom, she can get an updated advisory that uses the newly entered data.

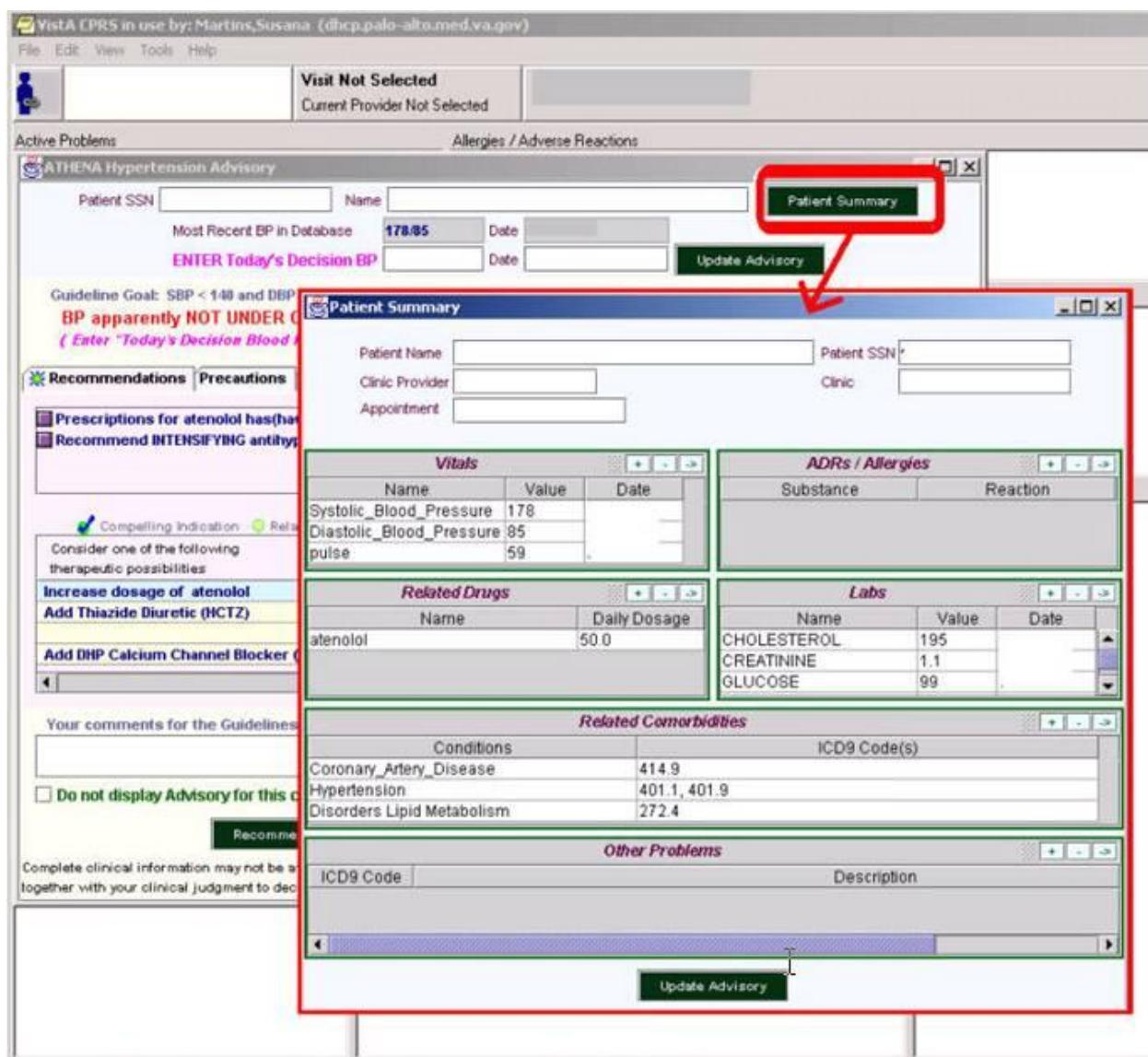


Figure 10 - The Patient Summary window. The window pops up on top of the ATHENA-CDS hypertension advisory window when a user clicks on the Patient Summary button highlighted in the figure.

II.2. System Architecture

The architecture of a software system consists of the system's constituent components and the relationships among them. ATHENA-CDS SYSTEM's system architecture reflects the fact that it uses SMI's EON technology for developing decision-support systems for guideline-based care [4-6, 8, 9]. Subsection II.2 will first present the EON system architecture, because it provides the technological foundation of ATHENA-CDS SYSTEM; it will then describe the architecture of ATHENA-CDS SYSTEM, showing how it has added components to embed EON software.

within the VA's clinical information system. Through the discussion of ATHENA-CDS components, the operation of ATHENA-CDS SYSTEM will be introduced, indicating how data are extracted from VA's VistA database, how advisories are computed, and how advisories are delivered to physicians at the point of care.

II.2.1. EON System Architecture

EON is an extensible architecture for developing decision-support tools for various aspects of protocol-based care. Initially developed to create systems for executing clinical trial protocols for the treatment of cancer and HIV infection [16], it has been extended for the management of chronic diseases and of other types of guidelines. Because of the generic nature of the EON technology, ATHENA-CDS SYSTEM has the potential to be extended to encode additional guidelines, to be extended for additional decision-support tasks, and to be implemented in clinical information systems outside the VA environment.

The functionality of the tools provided by EON has been shaped by the tasks they were designed to perform. As shown in Figure 11, an EON application may contain several classes of components (further described in Subsections II.2.1.1 to II.2.1.3.):

1. problem-solving modules such as the EON Guideline Interpreter;
2. the ChronusII temporal mediator;
3. a declarative guideline knowledge base that defines all clinical protocols and guidelines in terms of a general guideline model; and
4. client programs that access the services provided by problem-solving modules and by the ChronusII temporal mediator.

The current EON implementation uses Common Object Request Broker Architecture (CORBA) as its client/server infrastructure [17], where the interfaces to the server components are defined using Object Management Group's Interface Definition Language (IDL), a standard of the International Organization for Standardization (ISO).⁵ In addition to being accessed as server processes running in a distributed set up, like any CORBA component, both problem-solving modules and the ChronusII temporal mediator can also be linked into applications as library programs.

EON's problem-solving modules access the guideline knowledge base through Protégé's Application Programming Interface (API). Depending on the services a client program provides to end users, a client program may or may not access the knowledge base.

II.2.1.1. EON Problem-Solving Modules

⁵ A version of the EON Guideline Interpreter that uses web services 18. W3C. *Web Services Activity*. 2002 [cited 2003; Available from: <http://www.w3.org/2002/ws>. instead of CORBA is under development.

All problem-solving modules in the EON guideline applications (see Figure 11) access a guideline knowledge base consisting of models of clinical guidelines, patient data, and medical concepts that are created in and accessed through the Protégé tool. These problem-solving modules perform the computations necessary to automate specific tasks associated with guideline-directed therapy. The EON Guideline Interpreter, shown in the middle in Figure 11, is one such module. It takes as input a standard clinical guideline description and relevant patient data, and generates as output situation-specific recommendations for the current patient encounter [9].

Other modules developed in the past include a program for determining eligibility for clinical trial protocols. It generates as output the qualitative likelihood that a patient is eligible for the given protocol [19]. Another is an explanation-generation component that takes as input patient data and EON's guideline recommendations, and generates as output explanations based on an argumentation model [4].

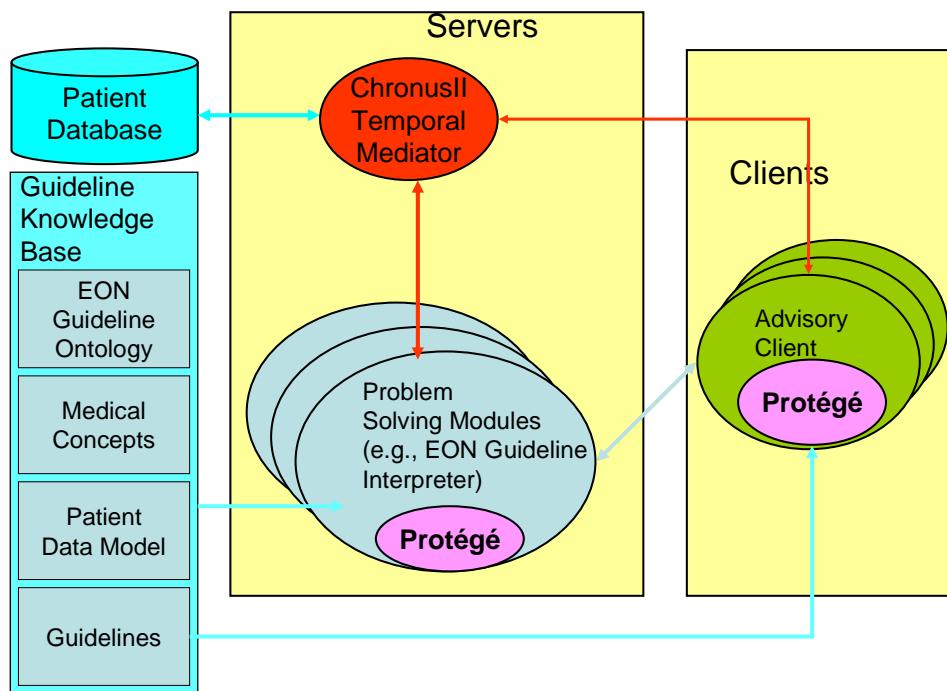


Figure 11 - Architecture of EON guideline applications

II.2.1.2. ChronusII Temporal Mediator

Clinical databases typically contain a significant amount of temporal information, such as when a specimen for a laboratory test is obtained and when a prescription is written and filled. This information is often crucial in medical decision-support systems. Although ad hoc queries that

involve time are common in clinical systems, the medical informatics field has no standard means for representing or querying temporal data. Over the past decade, the temporal database community has made significant progress in developing models of time and integrating them into database systems. Much of this research can be applied to clinical database systems.

As part of the EON project, a temporal database mediator has been developed that serves as the conduit between the problem-solving modules and the clinical data stored in an archival patient database [8]. Called ChronusII, it extends the standard relational model and the SQL query language to support temporal queries. It provides an expressive general-purpose temporal query language that is tuned to the querying requirements of clinical decision-support systems. When a calling program such as the EON Guideline Interpreter seeks to answer a question such as, “How long has the patient been taking lisinopril at a dose greater than 30 mg per day?”, it passes the query to ChronusII. ChronusII:

1. queries the prescription table in the database for rows of data where the dose of lisinopril is greater than 30 mg;
2. determines whether there are rows of data whose start and stop times meet or overlap and thus can be concatenated; and
3. determines the duration of the concatenated time interval.

ChronusII encapsulates temporal database functionalities, so that each problem-solving module does not have to duplicate these functionalities itself.

II.2.1.3. EON Knowledge Base

EON contains a declarative knowledge base that includes the EON Guideline Model, medical-concept model, and patient data model, which are described in detail in Subsection IV.2. This subsection gives a brief overview; and Figure 12 shows a portion of the EON knowledge base as represented in the Protégé editor.

1. The **EON Guideline Model** [6] consists of a set of classes and attributes that describe concepts and relations with which the content of clinical guidelines are formalized. The content of a particular guideline is encoded as instances and as attribute values of these classes. The EON Guideline Model includes classes and attributes for modeling concepts such as the target population of the guideline (eligibility_criteria attribute in Figure 12), goals (goal attribute in Figure 12), and a clinical algorithm that sequences the decisions and actions of a guideline (the screen shot in the upper-right corner of Figure 12). Depending on the nature of the guideline knowledge to be modeled, the EON Guideline Model may allow a clinical algorithm to be specified as a collection of scenarios, decisions to be made in these scenarios, and preferred alternatives at these decision points. Alternatively, when complex sequencing of decisions and actions is required, the model allows a clinical algorithm to make use of additional operators that perform iteration, multiple branching, and synchronization of concurrent threads of execution.

2. The **medical-concept model** defines the particular clinical interventions that are typical for a given area of medicine, and the types of patient findings and patient problems that are most commonly reported in a given medical discipline. The medical-concept model is an explicit component of the EON knowledge base, in the acknowledgement that different classes of health-care providers tend to make different classes of observations about their patients and perform different kinds of patient-care activities. In ATHENA-CDS, the medical-concept model consists primarily of comorbidities affecting the management of hypertension and of the classes of antihypertensive drugs that the system may need to cover.
3. A **patient data model** defines the classes and attributes of patient information required by the rest of the system. This simplified view of patient data, created for the purpose of aiding in clinical decisions, supports: (a) a structured data model for representing information related to individual patients, (b) domains for values of attributes in the data model, and (c) logical expressions through which guideline decision-support systems can test the states of the patient.

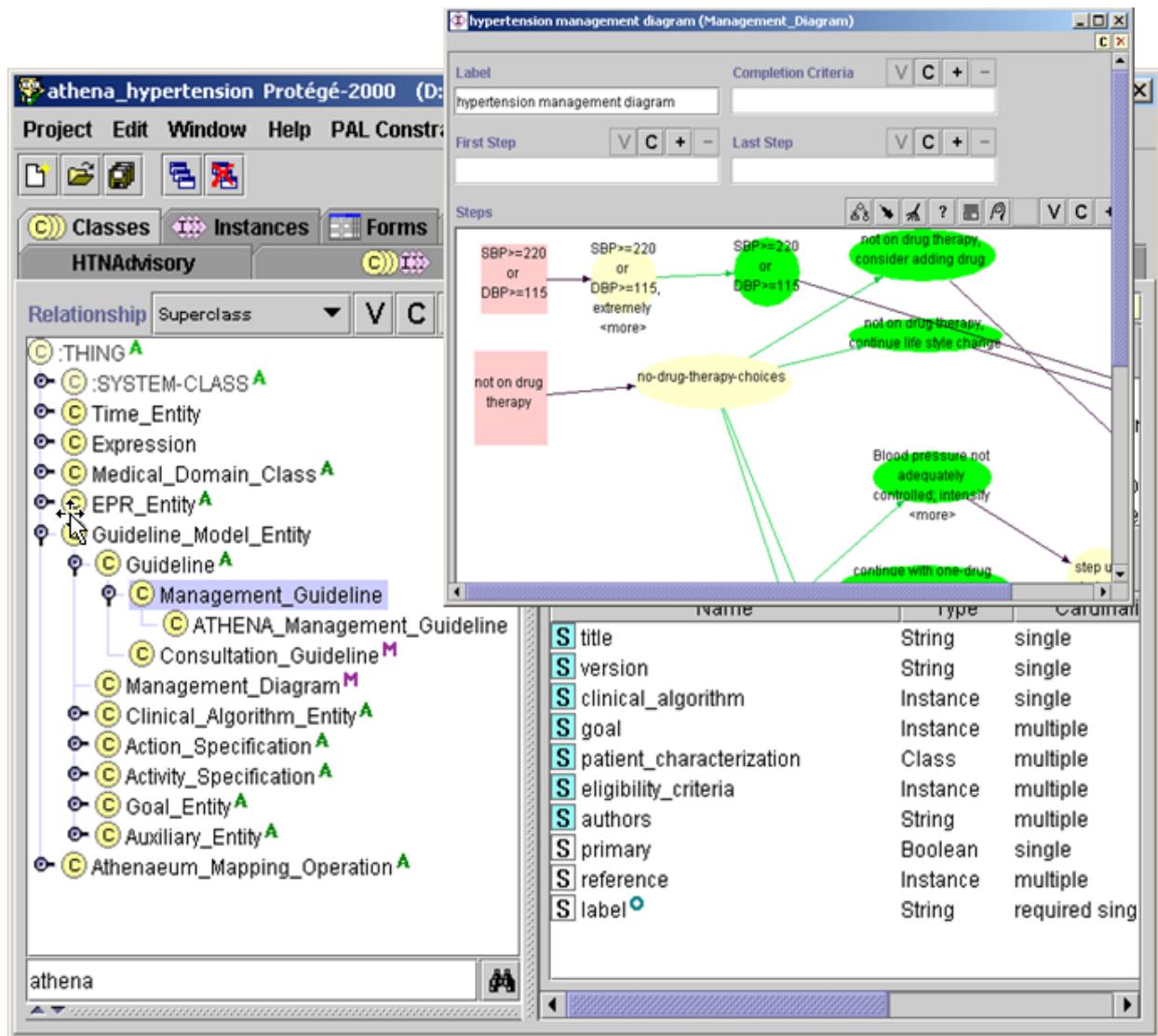


Figure 12 - The EON Guideline Model in the Protégé editor. The left panel shows part of the hierarchy of classes in the model. The Management_Guideline class is selected, shown by the violet highlighting. The template slots shown in the right panel include the clinical_algorithm slot. The clinical algorithm represents guideline recommendations in terms of patient scenarios, decision points, and action alternatives. The floating screen shot in the figure shows the hypertension management diagram (an instance of the Management_Diagram class) presented as a Protégé graph.

II.2.2. ATHENA-CDS System Architecture

Within the VA medical centers' information technology environment, ATHENA-CDS extends the EON architecture in five ways:

1. the ATHENA-CDS Knowledge Base, which encodes—according to nationally recognized hypertension guidelines—the knowledge required to support decisions for managing hypertension patients;

2. a system for extracting data from VA data sources and converting them into a form usable by the EON components;
3. a mechanism for precomputing and storing guideline-based advisories;
4. the ATHENA Client program, through which guideline-based recommendations are delivered to physicians in a pop-up window as described in Section II.1; and
5. an event monitor (also known as the Controller) that keeps track of provider logins and patient selection in VA's Computerized Patient Record System (CPRS) information system, and that activates ATHENA Client (see Section II.2.2.4).

The following sections describe the five ATHENA-CDS components.

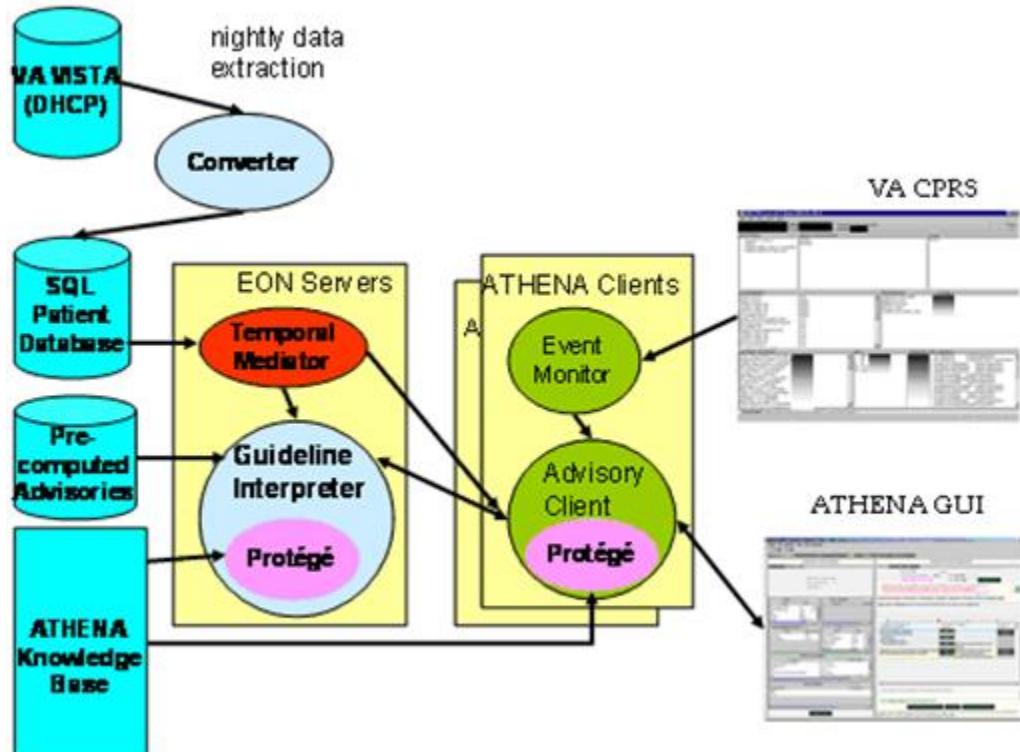


Figure 13 - ATHENA-CDS system architecture. The current implementation of ATHENA-CDS SYSTEM is integrated into the VA clinical information system.

II.2.2.1. ATHENA-CDS Knowledge Base

The ATHENA-CDS Knowledge Base extends the EON knowledge base, such that it: (1) represents medical concepts used in the treatment of hypertension; (2) encodes details of the

hypertension guidelines that are required to allow the Guideline Interpreter to reason about the treatment of hypertension and related comorbidities for a particular patient; and (3) maps guideline terms used in the ATHENA-CDS Knowledge Base to those used in VA's information system.

1. *Medical concepts* – Guidelines are usually written with general terms (eg “diabetes mellitus” that are represented in the patient data by many different files (eg ICD9 codes for diabetes include 250.01, 250.03, 250.11, 250.13 ...). When recommendations of a clinical guideline or protocol are encoded in a computer-interpreter model, they are constructed as formal statements. The statements reference medical concepts describing patient states of health and also interventions that have been or should be performed for a patient. These medical concepts are formalized as a hierarchy of concept classes and are part of the EON medical concept model. The concept classes are used to provide conceptual categories through which the patient findings, diagnosis, and ongoing treatment can be expressed. They include: vital signs (e.g., Weight), diagnostic terms (e.g., Diabetes-Mellitus-Type1), ICD9 codes (e.g., 274.9 for “Gout, unspecified”), laboratory-test results (e.g., Creatinine), demographic concepts (e.g., Race and Sex), drug classes (e.g., Thiazide_Diuretics), generic drugs (e.g., atenolol), units of measure (e.g., milligram), and defined concepts (e.g., risk group A: no risk factor and no target organ damage or clinical cardiovascular disease).⁶
2. *Guideline-based decision-support knowledge* – The decision-support knowledge for guideline-based care includes criteria for applying the hypertension guidelines to a patient case (e.g., patient must have a diagnosis of hypertension and must not be pregnant), patient scenarios for hypertension management, alternative choices and justifications for them, guideline-based indications and contraindications for each class of antihypertensive agents, and dose ranges of individual drug components. The decision-support knowledge is encoded as instances of the EON Guideline Model (see Figure 12).
3. *Mapping knowledge* Information related to vitals, demographic information, and laboratory-test terms used in the ATHENA-CDS Knowledge Base (e.g., HDL_Cholesterol) are mapped to those used in the VA clinical information system (e.g., HDL].

⁶ Classes in the medical concept hierarchy are derived from the VA database terms in four ways: (1) The VA uses ICD9 codes to record diagnostic information, the ICD9 codes in the concept hierarchy correspond directly to the VistA data dictionary; (2) The knowledge-base mapping described in #3 defines a correspondence between a concept (e.g., a laboratory test term such as Serum Creatinine) and a VA term. In the case of drugs, the mapping is done in the drug table in the ATHENEON database, matching drug names used in VA prescriptions to the generic drugs in the concept hierarchy. (ATHENA-CDS currently has a hybrid set up where some mapping occurs in the knowledge base and some in the database); (3) A defined concept (e.g., risk group A) has a computable definition (e.g., absence of any risk factor, and no target organ disease or clinical cardiovascular disease) that is evaluated at run-time in terms of other concepts and (4) Concepts that contain any of the previously mapped concepts (e.g., ICD9 codes) as direct or indirect subclasses match VA data through their subsumed relationship to these previously matched concepts.

In Figure 13, the ATHENA-CDS Knowledge Base is displayed as a rectangle. In the deployed system, the knowledge base is stored on a fileserver as a collection of Protégé files. The knowledge base is accessed by other EON and ATHENA-CDS components through Protégé's Application Programming Interface (API), shown as a pink oval in Figure 13.

II.2.2.2. Data Extractor, Converter, and Precomputation of Advisories

This subsection describes the data extraction and conversion that are necessary for the ATHENA-CDS SYSTEM to use data from the VA VistA database. It also covers the precomputation and storage of ATHENA-CDS advisories to be delivered to clinicians when they access patient records in VA's CPRS system. The process is diagrammed in Figure 14.

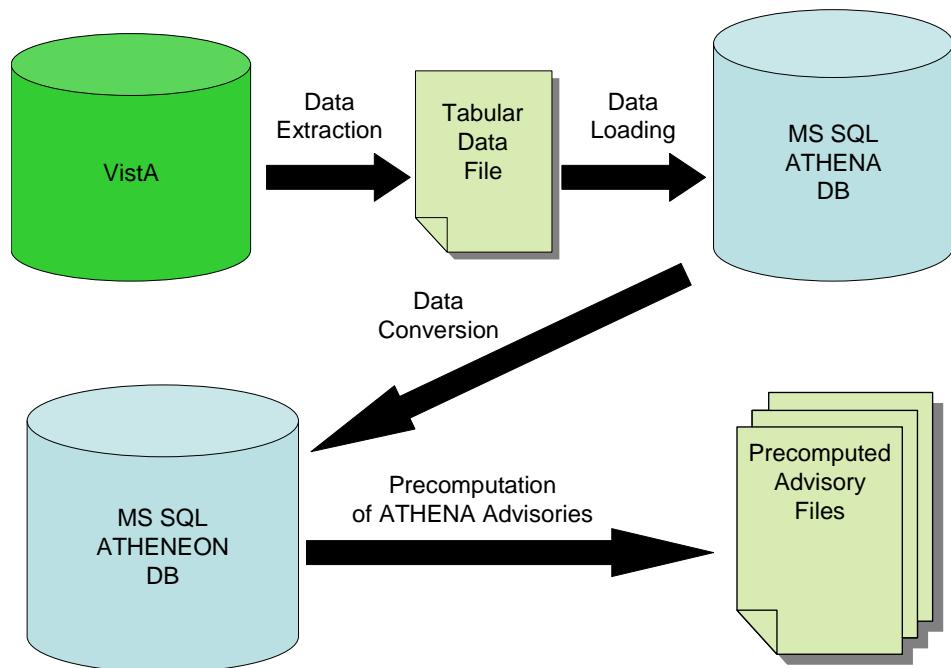


Figure 14 - Nightly extraction and conversion of data, and precomputation of ATHENA-CDS advisories

To generate patient-specific advisories for the management of patients with hypertension, ATHENA-CDS SYSTEM depends on data extracted from the VA VistA database. Because VistA uses the hierarchical MUMPS⁷ database system—not a relational database system, as required by EON software components—data must be extracted from VistA and transformed into a relational format. The extraction is done by a M program that writes out selected patient data into a tabular flat file. This file is then loaded, through MS SQL Server data transformation

⁷ Programming language M, formerly MUMPS originally stood for *Massachusetts General Hospital Utility Multi-Programming System*, which had been developed by Octo Barnett's lab at Massachusetts General Hospital (MGH) in the 1960s. The term “MUMPS” is no longer considered to be an acronym.

service (DTS) scripts, to tables in an SQL Server database, referred to as the ATHENA-CDS database.

The MS SQL Server DTS scripts take the caret-delimited (^) text file generated by the MUMPS data extraction routines and converts it into the following ATHENA-CDS SQL database tables: ALLERGY, ATHENA1, DEMOGRAPHICS, DIAGNOSIS, ENCOUNTER, LABS, PATIENTS, PRESCRIPTIONS, PROVIDERS, and VITALS (see Figure 15). Descriptions of table contents are in appendix V.5. The scripts perform basic data scrubbing operations. For instance, they:

1. convert date and numeric string data into date and numeric SQL Server fields;
2. transform the “systolic blood pressure/diastolic blood pressure” string used in VistA into two separate data entries;
3. separate Vitals data (e.g., heights, weights, blood pressure, and pulses) into another table;
4. combine diagnostic data derived from multiple sources into a single table; and
5. scrub out obvious erroneous data entries, such as dates in the future or impossible lab values.

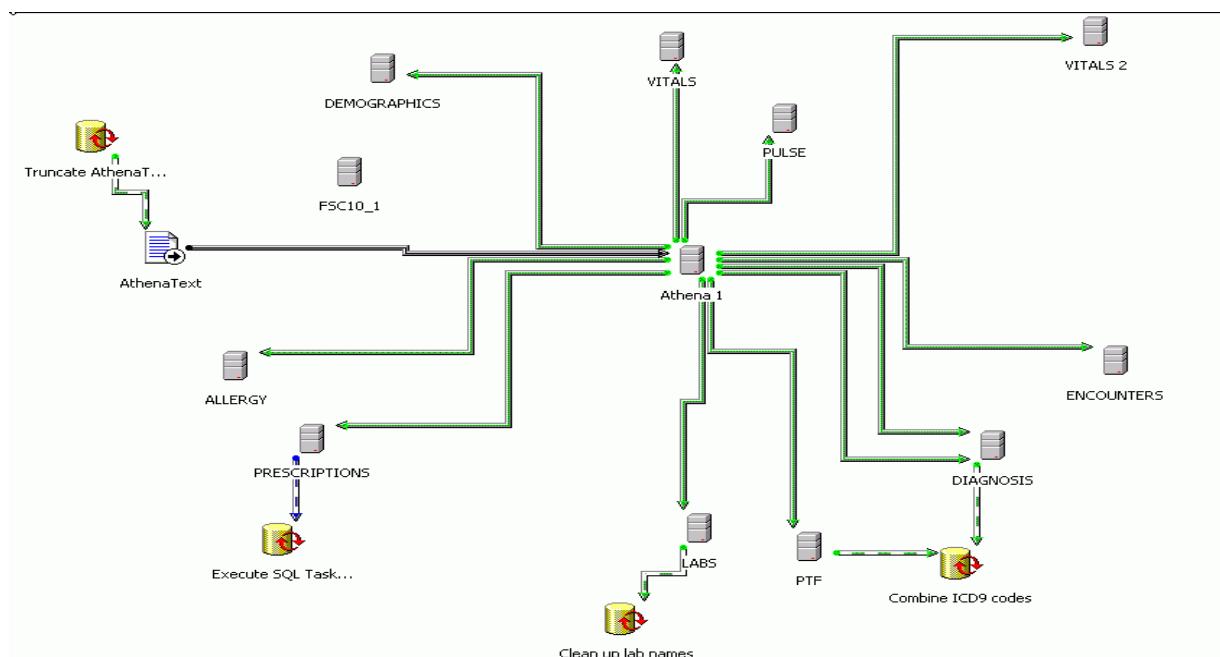


Figure 15 - The relational ATHENA-CDS SQL tables created by MS SQL Server data transformation service scripts from extraction text file.

After the creation of relational ATHENA-CDS tables, a Java program—Converter.java—performs further conversions to normalize and format data as required for ChronusII’s temporal database [8]. The Converter program draws from the ATHENA-CDS tables to populate tables in the ATHENEON database. ATHENEON is used by the EON components as well as ATHENA-CDS Client.

Among the data normalization functions it performs, the Converter:

1. Converts all tables from a standard relational form to a temporal relational form for use by the ChronusII temporal query language.
2. Converts adverse drug reaction data to a standardized form with three categories: drug, food, and other.
3. Computes a prescription’s daily dose in milligrams from the prescription information.
4. Generates prescription history where the valid time of a data record represents the period during which, judging by the prescription and refill history, the patient should have a supply of the prescribed medications.
5. Identifies prescriptions as “active and currently filled,” “active but not currently filled,” or “inactive”.
6. Converts lab table data in athena db to Studies table in atheneon db based on the mapping table ‘Lab_Limits’ located in athena db

Using solely data available from VistA, the ATHENA-CDS SYSTEM will pop up a window to display a patient-specific advisory based on the ATHENA Knowledge Base. In order to minimize the time it takes to display the pop-up window, a batch program precomputes advisories for all patients in the Encounters table in the ATHENEON database. To generate the precomputed advisories, the batch program uses the same Application Programming Interface (API) provided by the Guideline Interpreter and ChronusII mediator as the ATHENA-CDS Client uses to pop up the advisory window (see II.2.2.4).

The Guideline Interpreter has been programmed to make certain assumptions when relevant data are not available. For example, if a patient is not recorded as having a comorbidity, the Guideline Interpreter assumes that the patient does not have that comorbidity. If there are no blood pressure data at all, the Guideline Interpreter first computes an advisory assuming that blood pressure is under control, and then computes an alternative advisory assuming that blood pressure is not under control.

A number of MS SQL Server data transformation service scripts automate the process of extracting data from VistA to the file system. They load the data into MS SQL Server, convert the data into the form required by ChronusII, and then invoke the batch program to precompute and store advisories. The scripts are configured to run every night, extracting data and computing

advisories associated with any patient who has a primary-care appointment during a five-day window (three days before to one day after the date of extraction). Thus, for three days before and one day after a patient's appointment day, the ATHENA-CDS advisory for the patient will be available to clinicians caring for the patient. Because of the five-day window, if a clinician accesses the patient's medical record on Friday to plan a Monday encounter, he will see an ATHENA-CDS advisory for the expected patient. Similarly, a clinician who wishes to review results from the previous day's encounters will see ATHENA-CDS advisories.

II.2.2.3. Event Monitor

The Event Monitor (also known as **ATHENA Controller**) is a program that is started whenever a clinician logs into a clinical workstation where ATHENA-CDS SYSTEM is installed. The Event Monitor starts the ATHENA Client, a Java process that is responsible for managing the delivery of ATHENA advisories (sets of recommendations) to clinicians.

The Event Monitor is ATHENA's current method of integration with the Computerized Patient Record System (CPRS). The Event Monitor registers with CPRS and listens for CPRS messages. It maintains a communication channel with **ATHENA Client**, and passes the CPRS messages it receives to the **ATHENA Client**.⁸

II.2.2.4. ATHENA Client

When the ATHENA Client starts up on a clinical workstation, it establishes connections to the ChronusII and Guideline Interpreter servers. It also loads the ATHENA-CDS Knowledge Base, from which it gets the medical concept hierarchy used to assist entry of data into the Patient Summary pop-up. Once ATHENA Client is started up, it runs in the background, waiting for messages from the Event Monitor.

When ATHENA Client receives a message indicating the patient identifier of a record being accessed by a clinician, it queries the ChronusII temporal mediator to see if the clinician should receive ATHENA-CDS's decision support. If so, ATHENA Client then requests a precomputed advisory from the Guideline Interpreter. If there is no precomputed advisory, then it requests that an advisory be computed in real time using data available in the ATHENEON database. If the advisory—i.e., the results computed by the Guideline Interpreter—suggests that the patient does not satisfy the eligibility criteria of the **ATHENA hypertension guidelines**, then no pop-up is generated; otherwise, ATHENA Client pops up a window that displays the advisory on top of the CPRS cover sheet. It also generates a patient information summary (see Subsection II.1.6) from data queried from the ChronusII mediator. A user may enter data through the ATHENA Client's graphical user interface and get an updated advisory from the Guideline Interpreter.

⁸ The Event Monitor is being phased out and replaced by a new mechanism—based on CCOW, the Health Level Seven standard for context management (Health Level 7, 2004) for integration with CPRS.

The ATHENA Client can be configured to log its operations in two ways:

1. The data used by the system or entered by the user, the advisories that are precomputed, displayed or updated, and all user actions can be logged to SQL tables.
2. Errors in system operations (e.g., the ATHENA Client not being able to connect to the ChronusII mediator) can be logged to files that are scanned automatically to generate email messages that alert ATHENA-CDS team members of these errors

III. ATHENA-CDS Installation

This section gives the background and instructions for both Information Resource Management personnel and ATHENA-CDS staff necessary for installing the ATHENA-CDS SYSTEM. It describes the requirements for and the installation of the ATHENA-CDS software and extraction routines on the VA's computerized hospital information system (Veterans Health Information Systems and Technology Architecture orVistA), the ATHENA-CDS Server and the **ATHENA clients**.

III. 1. Requirements

This subsection describes the requirements that must be met in order to setup and install the ATHENA-CDS SYSTEM at a VA facility with its own implementation of VISTA. It also describes the setup of the ATHENA-CDS Server. The ATHENA-CDS server may support more than one VA facility.

III. 1.1. System and Server Requirements

Windows NT Server (2000 or 2003) and Microsoft SQL Server 2000 are required to implement ATHENA-CDS. A site administrator with a working knowledge of both server applications should perform the installation

The ATHENA-CDS server should be at least a dual 1 GHz+ processor server with 2 GB RAM and a 120 GB hard drive. The server must be running Windows 2000 or 2003 Enterprise Server software. The operating system needs to be configured with the Internet Information Service (IIS) installed. Follow Microsoft's installation instructions to install a secure FTP site named **ATHENA FTP** within the IIS for receiving data files. SQL Server 2000 must be installed using dual authentication, and SQL Mail should be configured to work with the Microsoft Exchange Server. ActivePerl-5.8.4.810-MSWin32-x86 or higher should also be installed, and a Borland Visibroker license is required.

III. 1.2. Extraction Process Requirements

The clinical information required by the ATHENA-CDS SYSTEM is extracted from the VA's computerized hospital information system (Veterans Health Information Systems and Technology Architecture orVistA), using a set of MUMPS routines (M or Cache). Patient and clinical data extracted from the VistA/DHCP system are converted and stored in an SQL Server database on the ATHENA-CDS server (as described in Section II). This process that normalizes the data used by the ATHENA-CDS system. These routines are distributed as a Kernel Installation and Distribution System (KIDS) build. The KIDS build must be installed on the local VistA system by a MUMPS programmer.

III.1.3. Monitoring, Archiving Requirements

Monitoring and archiving requires SQL Server Jobs, SQL Server Data Transformation Service (DTS), Perl scripts, Borland Visibroker license and the archiving application PKZIP. Their installation is part of the ATHENA-CDS distribution install described in the Server Installation section (III.3).

III.1.4. Client-side Requirements

Client computers (clinic computers) should be running the Computerized Patient Record System (CPRS) and Windows (NT 4.0, 2000, or XP). They should also have a 1 GHz+ processor, 256 MB of RAM, and at least 100 MB of free hard drive space. A version of ATHENA-CDS that will run on a ‘thin client’ is currently under development.

III.1.5. Security and Permissions

Security and permissions are maintained using Windows NT and SQL Server 2000 security settings.

III.2. VistA Installation

The Department of Veterans Affairs (VA) has had automated information systems in its medical facilities since 1985, beginning with the Decentralized Hospital Computer Program (DHCP) information system. The current system called Veterans Health Information Systems and Technology Architecture (VistA) supports ambulatory and inpatient care and includes the Computerized Patient Record System (CPRS). CPRS provides a single interface for health care providers to review and update a patient’s medical record and to place orders, including medications, special procedures, x-rays, patient care nursing orders, diets, and laboratory tests.

III.2.1. ATHENA-CDS Data Extraction

VistA data required by ATHENA-CDS are extracted from M files in VistA and loaded into the Microsoft SQL Server 2000 ATHENA-CDS database. This is accomplished with a set of M routines that are distributed in a KIDS build. Included in the KIDS build are the **ATHENA Clinic Enter/Edit option**, the **ATHENA Lab Enter/Edit option**, and the **ATHENA Extraction option**. The VA Palo Alto HCS M programmer (Larry Lou, 650-493-5000 x65999) will work with the M programmer to accomplish the installation of the KIDS build.

ATHENA Clinic Enter/Edit Option

After KIDS build installation, use the ATHENA Clinic Enter/Edit option created by the KIDS installation to enter the clinics of the providers who are authorized to receive the ATHENA-CDS SYSTEM. The clinics are stored in the HOSPITAL LOCATION file (#44). The *ATHENA Clinic Enter/Edit* option menu needs to be available, through VistA, to the person responsible for entering the clinics. Once on this person's VistA menu, enter ***^ATHENA*** at the VistA prompt to open a set of prompts similar to the following VistA screen capture illustration (user input is in large, bold italics):

Select Hospital Information Menu Option: ***^ATHENA*** [enter]

1. ATHENA Clinic Enter/Edit [AOVATHE CLINIC ENTER]
2. ATHENA Lab Enter/Edit [AOVATHE LAB ENTER]

Type '^' to stop, or choose a number from 1 to 2 : ***1*** [enter]

ATHENA Clinic Enter/Edit

Select ATHENA CLINICS CLINIC NAME: ***NEW ATHENA CLINIC*** [enter]

Are you adding '***NEW ATHENA CLINIC***' as a new ATHENA CLINICS (the 218TH)?

No// ***Y*** (Yes) [enter]

CLINIC NAME: ***NEW ATHENA CLINIC*** // [enter]

Select ATHENA CLINICS CLINIC NAME:

At this point the user can type ***^*** to stop, or can enter another clinic name.

ATHENA Lab Enter/Edit Option

After installation, employ the ATHENA Lab Enter/Edit option to enter the local lab names or the LOINC.of the clinical labs (VistA LAB DATA file #63). Use the option on ATHENA-CDS SYSTEM as follows (user input is in large, bold italics):

Select Hospital Information Menu Option: ***^ATHENA*** [enter]

1. ATHENA Clinic Enter/Edit [AOVATHE CLINIC ENTER]
2. ATHENA Lab Enter/Edit [AOVATHE LAB ENTER]

Type '^' to stop, or choose a number from 1 to 2 : ***2*** [enter]

ATHENA Lab Enter/Edit

Select ATHENA LAB TESTS: ***LABTEST*** [enter]

Are you adding 'LABTEST' as a new ATHENA LAB TESTS (the 16TH)? No// **Y** (Yes)
[enter]

LAB TEST: **LABTEST//** [enter]

LOCATION IN LAB DATA FILE: [enter]

LOINC CODE: **2951-2** [enter]

Select ATHENA LAB TESTS:

At this point the user can type ^ to stop, or can enter another lab test name.

Table 1 shows the labs for ATHENA, as of 12/01/05. Input should be sought from a local Laboratory Service supervisor if the mapping of the LOINC codes to the local lab test names are inaccurate.

LAB	LOINC
URIC ACID	3084-1
POTASSIUM	2823-3
URINE PROTEIN	20454-5
QUANT.URINE PROTEIN	2888-6
LDL CHOLESTEROL	13457-7
HEMOGLOBIN A1C	4548-4
TRIGLYCERIDE	2571-8
GLUCOSE	2345-7
ALBUMIN	1751-7
HDL	2085-9
CHOLESTEROL	2093-3
CREATININE	2160-0
SODIUM	2951-2
LDL, DIRECT	2574-2

Table 1 - Labs for ATHENA, LOINC codes as of 12/01/05

ATHENA-CDS Extraction Option

The extraction routine runs daily. It outputs data into a text format, and it uses a VistA Taskman routine to place data on an FTP site on the ATHENA-CDS server. It consists of the AOVATHE Build, AOVATHE Extract, and AOVATHE Push. These subroutines must be edited in order to change the site name and input the IP address of the FTP site. The user name and password for the FTP site must also be entered.

III.3. Server Installation

The server described in section 1.3.1 serves as the application, knowledge base and data base server. This section describes the installation of these components onto the ATHENA-CDS server.

III.3.1. ATHENA-CDS Distribution

The ATHENA-CDS distribution and included database files are installed from either a networked drive or a CD-ROM. The distribution includes:

- the ATHENA-CDS application and supporting Java and Protégé files,
- the ATHENA-CDS installation properties file (install.properties),
- a library of Microsoft SQL Server Data Transformation Service (DTS) routines,
- SQL scripts that create the ATHENA-CDS and ATHENEON databases, and
- SQL Server jobs required to run ATHENA-CDS SYSTEM.

The install properties file must be copied onto the root directory of the server's installation hard drive from the ATHENAINstallBin subdirectory of the networked drive or CD-ROM. Use any text editor to edit the install.properties file. Refer to the annotations given in the install.properties file; they define and give examples for all six parameters required for the server installation.

III.3.1.1. Running the ATHENA-CDS Installation Script

The installation script (runInstallServer.bat) is run from the \ATHENAINstallBin directory of the networked drive or CD-ROM. The script should be run separately for each instance of the application (e.g., live, development, or test). Run the script at a command prompt on the selected directory, or simply double click on the file.

The installation script creates and populates the directory structure in Figure 16. The shared drives (handheld folder icons in Figure 16) should be created by the ATHENA-CDS administrator, with permissions given to all ATHENA-CDS users. This is accomplished by creating an ATHENA-CDS users local group on the server and giving this group read permissions on the shared folders.

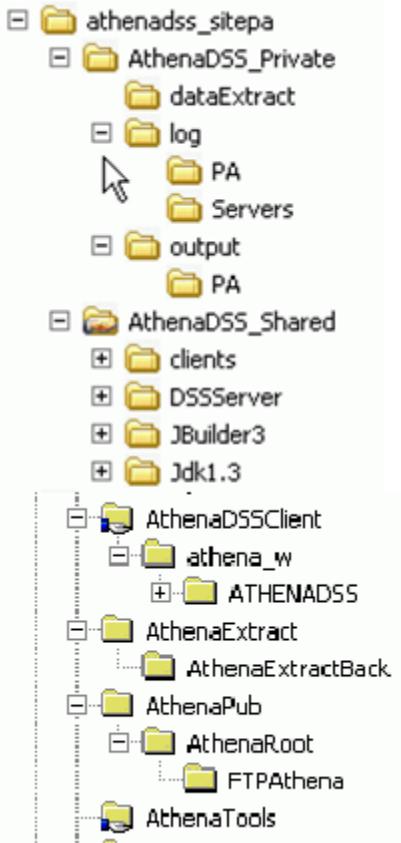


Figure 16 – ATHENA-CDS directory structure

III.3.1.2. Installing the ATHENA-CDS and ATHENEON Databases

The installation script runs the SQL code that creates and populates the ATHENA-CDS and ATHENEON databases.

III.3.2. Data Transformation Service (DTS)

The data extraction, transformation and loading (ETL) package that is part of the Microsoft SQL Server database is called the Data Transformation Service (DTS). This service allows building and running programs that prepare and load the data into the ATHENA-CDS database.

III.3.2.1. Installing ATHENA_Load Data Transformation Service Routine

The installation script runs the SQL code that creates and populates the ATHENA_Load.DTS routine. This routine sets up the loading routine that converts the caret-delimited (^) text file,

generated by the VISTA MUMPS (M) data extraction routines, into the following ATHENA SQL database tables: ALLERGY, ATHENA1, DEMOGRAPHICS, DIAGNOSIS, ENCOUNTER, LABS, PATIENTS, PRESCRIPTIONS, PROVIDERS and VITALS using the *ATHENA LOAD DTS* package (see Figure 17).

The installation DTS package is configured to work with an SQL Server database named ATHENA and with a text file connection located in the \ATHENAEExtract folder. The default text file name is athe_.txt. Generally, the extraction routine will add the number of the VA Station (e.g., 640 for Palo Alto HCS). The ATHENAText connection (see Figure 17) needs to be edited to reflect the number of the station (e.g., athe_640.txt). If the name of the ATHENA database is other than ATHENA, then each of the connections (e.g., Athena1, DEMOGRAPHICS, VITALS) and the corresponding data flow arrows must be edited to reflect the database's name (see Figure 17).

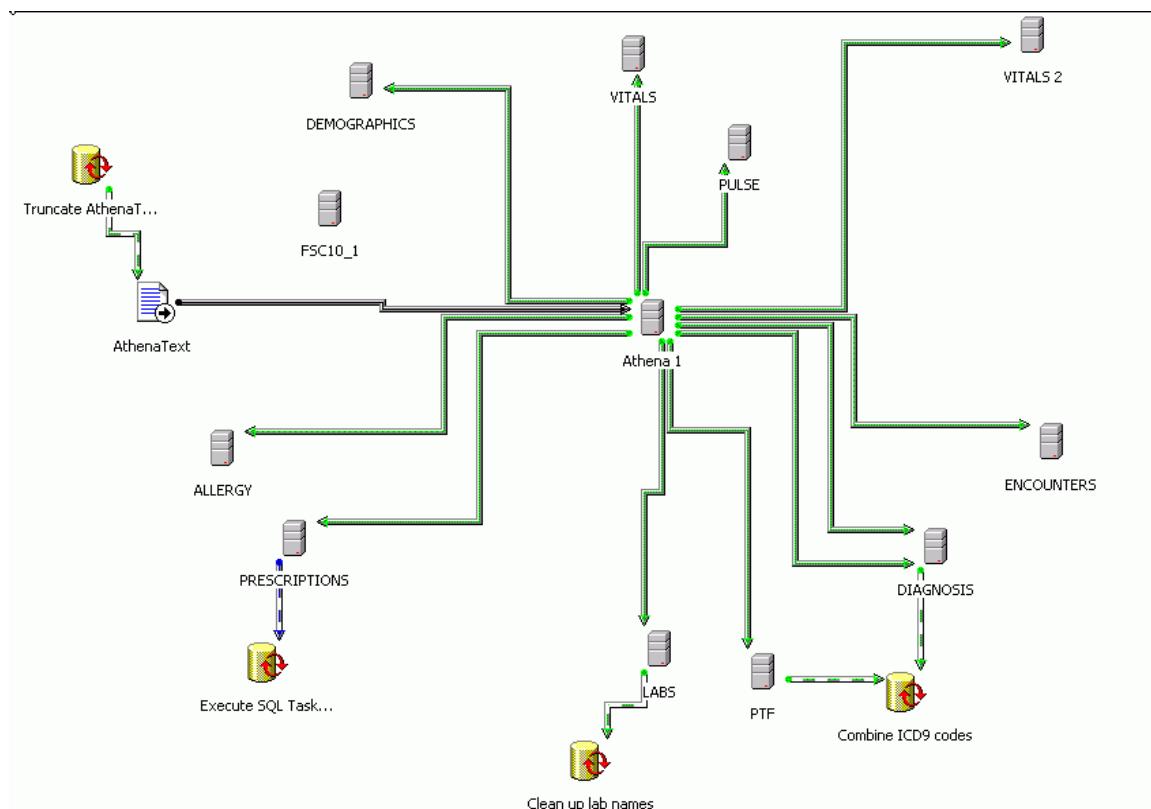


Figure 17 – ATHENA-CDS data flow. SQL tables are shown as named in the ATHENA database.

III.3.2.2. Installing the “Drug Check” Data Transformation Service (DTS) Routine

The installation script runs the SQL code that creates and populates the DrugCheck.dts routine. This routine sets up the DTS routine that transfers the table created by the DrugCheck SQL query within the Daily Job to a text file that can be emailed as an attachment to individuals who are monitoring the ATHENA-CDS program.

III.3.2.3. Installing the “Create Patients for CPRS Client” Data Transformation Service (DTS) Routine

The installation script runs the SQL code that creates and populates the CreatePatientsforCPRS Client.dts routine. This routine sets up the patient list that opens in the CPRS-emulating program used to test the system for the ATHENA-CDS advisory pop-up.

III.3.3. SQL Server Jobs

Automating the operation of Microsoft SQL Server is accomplished with SQL Server jobs. A job defines an administrative task or a set of tasks. Each job has one or more steps; each step specifies a Transact-SQL statement, Windows command, executable program, replication agent, or Microsoft ActiveX® script. Jobs can be run once, scheduled to run at periodic intervals, or specified to run when the server is idle.

III.3.3.1. Installing the “Daily Job” SQL Server Job

The installation script runs the SQL code that creates *Daily Job*, a SQL Server job. The Daily Job loads the CPRS data from the ATHENA M files extraction (see Subsection III.2.1) into SQL Server, and prepares the files for use by the ATHENA-CDS SYSTEM and the CPRS emulator program. After installation, this SQL Server job may be edited to match specific requirements of the site.

III.3.3.2. Installing the “Drug Check” SQL Server Job

The installation script runs the SQL code that creates *Drug Check*, an SQL Server job. Drug Check evaluates each drug in the prescription extraction to determine if there is an antihypertensive drug not mapped by ATHENA-CDS. Any drugs identified are emailed to the ATHENA-CDS administrator using SQL Server mail. After installation, this SQL Server job may be edited to match specific requirements of the site.

III.3.3.3. Installing the “Server Log Error Scan” SQL Server Job

The installation script runs the SQL code that creates *Server Log Error Scan*, an SQL Server job. The Server Log Error Scan scans the server error logs, extracts portions of any error messages, and emails the error information to the ATHENA-CDS administrator. After installation, this SQL Server job may be edited to match specific requirements of the site.

III.3.3.4. Installing the “Precompute Log Error Scan” SQL Server Job

The installation script runs the SQL code that creates *PrecomputeLogErrorScan*, an SQL Server job. The Precompute Log Error Scan scans the precomputed-advisories error logs, extracts portions of any error messages, and emails the error information to the ATHENA-CDS administrator. After installation, this SQL Server job may be edited to match specific requirements of the site.

III.3.3.5. Installing the “Client Log Error Scan” SQL Server Job

The installation script runs the SQL code that creates *ClientLogErrorScan*, an SQL Server job. The Client Log Error Scan scans client error logs, extracts portions of any error messages, and emails the error information to the ATHENA-CDS administrator. After installation, this SQL Server job may be edited to match specific requirements of the site.

III.3.3.6. Installing the “Monthly Client Log Backup” SQL Server Job

The installation script runs the SQL code that creates *Monthly Client Log Backup*, an SQL Server job. The Monthly Client Log Backup zips and moves all the log files located in the logs directory to a backup file named lastmonth.zip. After installation, this SQL Server job may be edited to match specific requirements of the site.

III.4. Client Installation

Running the ATHENA-CDS installation script creates and populates a subdirectory called ATHENADSSClient (See Figure 16). In this subdirectory, the batch file go.bat is used to perform the install on the client.

Client installation requires logging onto the client computer with the administrator’s logon. There, it is possible to access the Client folder and execute the go.bat file, following these steps after logon:

1. Click the START button on the Windows toolbar.
2. Select RUN from the menu.
3. Enter: \\{SERVER_name}\ATHENADSSClient

- Double click on go.bat.

SERVER_name is the computer name or IP address of the ATHENA-CDS server, e.g., VHAPALATHENA. ATHENADSSClient is the shared folder where the client files are stored on the ATHENA-CDS server (see Figure 16).

After running to completion, the installation can be tested by opening CPRS with a patient known to have been included in that day's data extraction and processed on the server. The ATHENA-CDS pop-up should appear.

III.5. Server-side Operation

In order to start the server processes that interact with the client computers (PCAServer, VisiBroker Smart Agent, ChronusIIServer, and the jbuilder and java executables) the INITEON.bat file must be started on the ATHENA-CDS Server using a logon that will allow all the started processes to continue running (i.e., a logon that does not time out). The rest of the operation of the ATHENA-CDS components is controlled using the SQL Server jobs. You can schedule jobs using the Schedules tab in each job (see Figure 18).

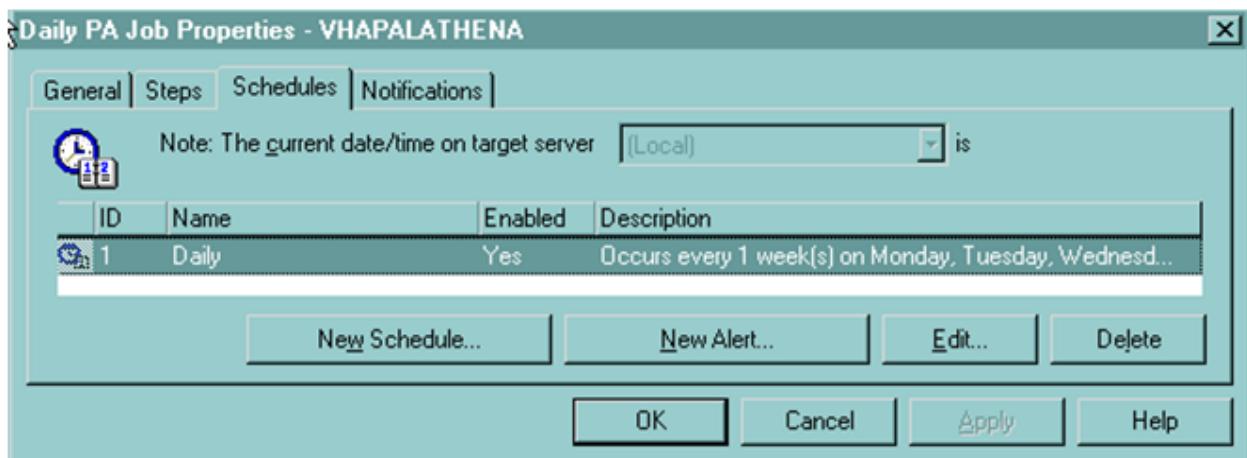


Figure 18 - SQL Server Job Agent. Scheduled jobs are shown under the Schedules tab. The Steps tab is used to view the subprocesses involved in a job.

In addition to the SQL Server event logs, successes and failures of SQL jobs can be monitored through SQL Server mail, which is controlled using the Notifications tab (see Figure 19).

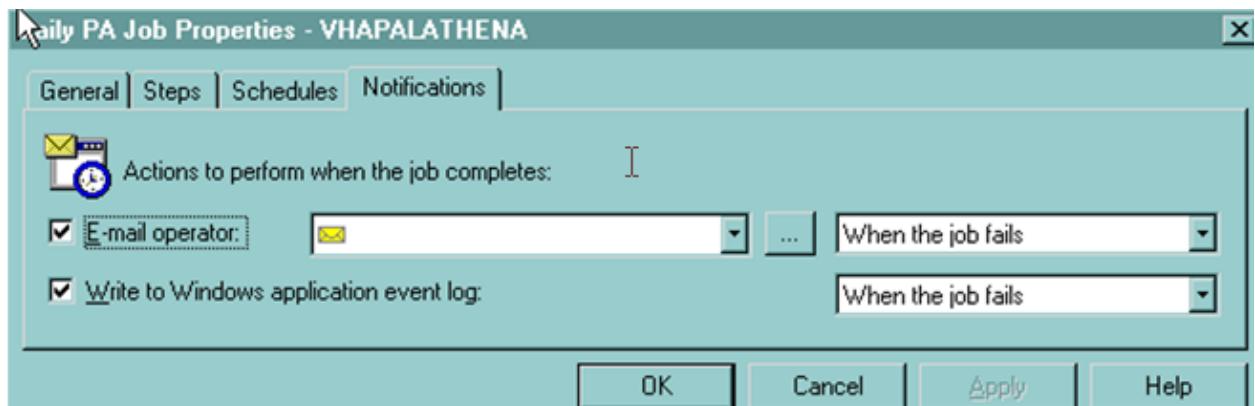


Figure 19 - Notifications tab in the SQL Server Job Agent. Notifications can be set up as email, event log entries, or both. The tab allows both email and event logging to be set up, triggered by the event of a job failure.

III.6. Client-side Operation

Clients are configured at installation to start the ATHENA Controller automatically each time a user logs onto a PC with the ATHENA-CDS client installed. The ATHENA Controller, in turn, starts the ATHENA Client as a background process. If this start-up process is successful, two processes are placed in the Task Manager window as shown in Figure 20 (the ATHENA Client process being `javaw.exe`). Both processes must be active in the Task Manager for the client to be operational.

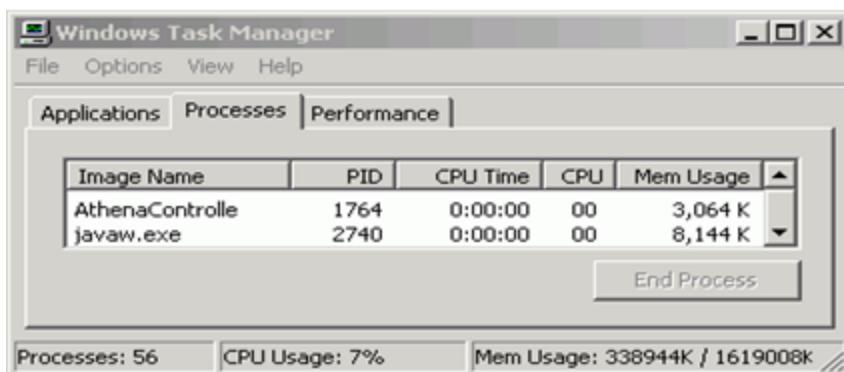


Figure 20 - Windows Task Manager showing usage of the client processes, ATHENAController, and the Client UI, running under the Java executable (javaw.exe).

III.7. Monitoring and Archiving

Monitoring and archiving ATHENA-CDS's operation is accomplished by using the SQL Server jobs and the log tables located in the ATHENEON database. Daily data extractions and precomputed advisories are processed into zip files, and can be stored for retrieval based on the policies of the institution.

III.7.1. Post-installation Tests

There are a number of processes on the server and client that must be operating before the ATHENA-CDS SYSTEM will function. The following sections outline these processes and how to identify if they are running or not.

III.7.1.1. Server Tests

For the server to be operational, the entire Daily Job needs to have been successfully completed; and the windows illustrated in Figure 21 and the applications indicated in Figure 22 must be open and running. These processes are started by the INITEON.bat batch file.

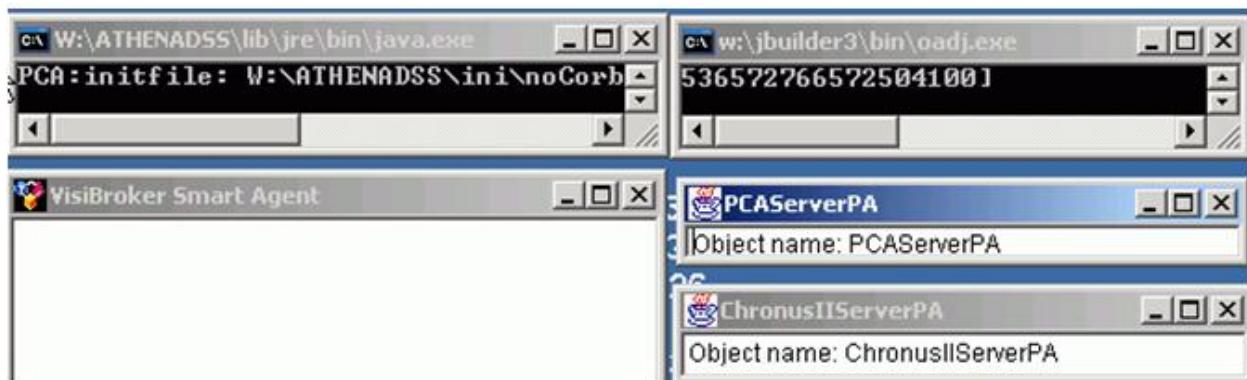


Figure 21 - Windows of operations started by the INITEON.bat file

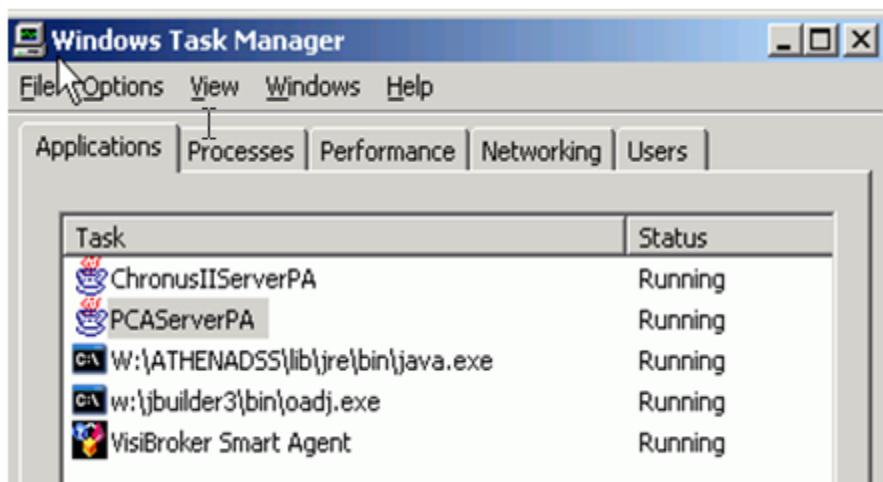


Figure 22 - Example of the Task Manager on the server

III.7.1.2. Client Tests

For the client to be operational, the ATHENA Controller needs to have successfully run. The batch file controlling this operation is located in the All Users startup folder, and should run automatically each time a user logs on to the computer. To verify the load, check the Task Manager. The processes illustrated in Figure 20 must be present.

III.7.1.3. Backup

Backup of ATHENA-CDS requires backing up the ATHENA_W directory and all its subdirectories. You should set up and automate a backup process, using either Windows Backup or another commercial backup product.

All ATHENA-CDS databases should be backed up on a weekly basis. Backing up the Tuesday load is recommended, because it will contain all the patients seen in a particular week, as the ATHENA-CDS extraction pulls patients from one day prior and three days post in addition to patients from the day it runs. You can automate the backups by creating the job, Database Maintenance, using the SQL Server Wizard for database maintenance.

IV. Knowledge Resources: Structure and Update Procedure

The ATHENA-CDS Knowledge Base is an encoding of the knowledge from clinical practice guidelines that is necessary for the management of hypertension in primary care. The knowledge base was developed by clinicians and knowledge engineers in the ATHENA-CDS Project, using a number of sources, including the *Sixth Report of Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure* (JNC 6), VA guidelines for the management of hypertension, and primary literature. The knowledge base focuses mostly on the use of appropriate anti-hypertensive agents in the context of a patient's most recent blood pressures, laboratory test results, current prescribed medications, drug allergies, and comorbidities. The ATHENA-CDS Knowledge Base encodes this guideline knowledge in the format required by the EON Guideline Model, using Protégé as the tool for encoding, browsing, and editing this knowledge base. Additionally, the ATHENA-CDS SYSTEM draws on supporting text material to explain its recommendations (see Subsections II.1.3 and II.1.4).

In this section, we describe Protégé (Subsection IV.1)), the structure of the models in EON and of the ATHENA-CDS Knowledge Base (Subsection IV.2), how the Guideline Interpreter makes use of the knowledge base to generate ATHENA-CDS advisories (Subsection IV.3), details about the textual sources accessed by ATHENA-CDS SYSTEM (Subsection IV.4), how to update the ATHENA-CDS Knowledge Base (Subsection IV.5), and how to specify some of the terminological mappings required to link terms used in the ATHENA-CDS Knowledge Base to those used in VistA (Subsection IV.6).

IV.1. Protégé

Protégé [10, 20] is an integrated software tool used by system developers and domain experts to develop knowledge-based systems. Created at Stanford Center for Biomedical Informatics Research (BMIR), Stanford University, Protégé is the tool used to develop the ATHENA-CDS Knowledge Base. Extensive documentation for Protégé is available on the Protégé website: <http://protege.stanford.edu>.

The following subsections give a basic introduction to Protégé. We will discuss Protégé's knowledge model [21], the integrity constraints that one can place in the knowledge base to make sure that it is well-formed, Protégé's file format, and, finally, its graphical user interface. More details can be found at: http://protege.stanford.edu/doc/users_guide/index.html.

The ATHENA-CDS Knowledge Base was developed using Protégé 1.7 and earlier versions. Between Protégé 1.7 and the current version of Protégé,⁹ its user interface (UI) has changed substantially. In this manual, we will use screen shots taken from Protégé 1.7.

IV.1.1. Protégé Knowledge Model

IV.1.1.1. Instances, Slots, and Classes

In Protégé, the basic unit of knowledge representation is a *frame*, a structure that holds specific types of information. Frames can be used to hold information about concrete individuals in the domain (e.g., *George Washington* or *JNC 6 guideline*), in which case, the frames are *instance frames* (or, more simply, *instances*). Frames can also be used to represent a named collection of individuals (e.g., *American Presidents*) or an abstract concept (e.g., the concept of a beta adrenergic antagonist drug), in which case, the frames are *classes*. Properties of instances—e.g., *eligibility_criteria* (shown as Eligibility Criteria in Figure 23),¹⁰ which define the target population of a guideline—are themselves frames called *slots*. Slots may have *values*. Slot values may be drawn from one of Protégé’s primitive types—float, integer, string, symbol, or Boolean—or they may be instances or classes.¹¹

⁹ As of June 29, 2009, Protégé 4.0 is the current version.

¹⁰ For each slot name displayed on the instance form, Protégé automatically replaces the underscore with a blank and capitalizes the first letter of each word. Thus, the slot *eligibility_criteria* is displayed as “Eligibility Criteria” on the instance form of the ATHENA_Management_Guideline class.

¹¹ In the rest of the document, we may use *attribute* synonymously with *slot*. They both represent properties or characteristics of an entity.

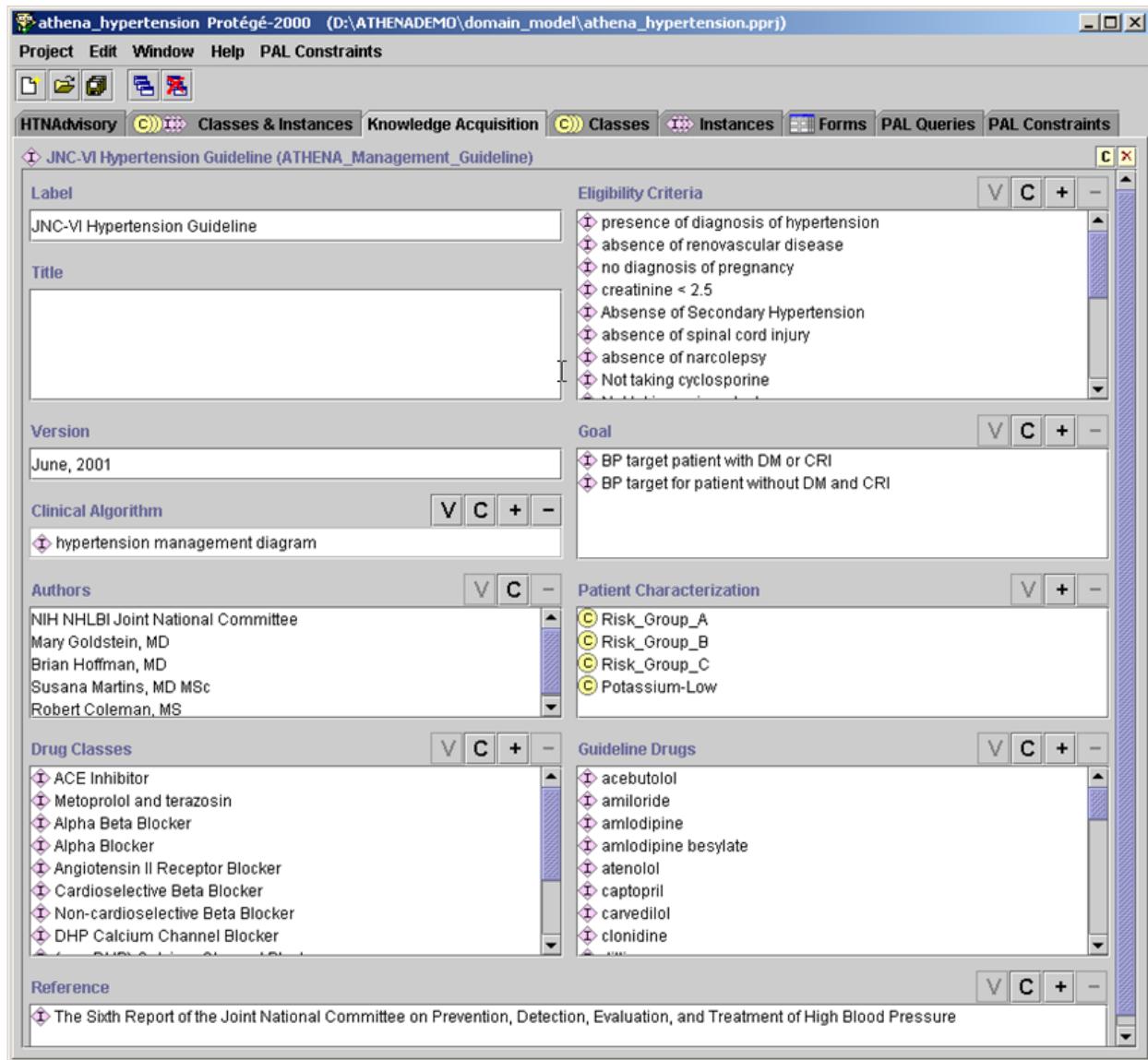


Figure 23 - Partial view of the JNC-VI Hypertension Guideline instance

In Protégé, classes are organized into classification hierarchies where children classes are specializations of parent classes. Furthermore, *template slots* associated with a class define the properties that instances of the class may have. Figure 24 shows the definition of the ATHENA_Management_Guideline class in Protégé. The left pane shows the class hierarchy of concepts in the ATHENA-CDS Knowledge Base. The middle pane shows instances (JNC-VI Hypertension Guideline) of the selected class (ATHENA_Management_Guideline), and the right pane shows the template slots associated with the selected class. Note that slots such as patient_characterization, eligibility_criteria, authors, version, and goal are precisely the slots that have values in Figure 23.

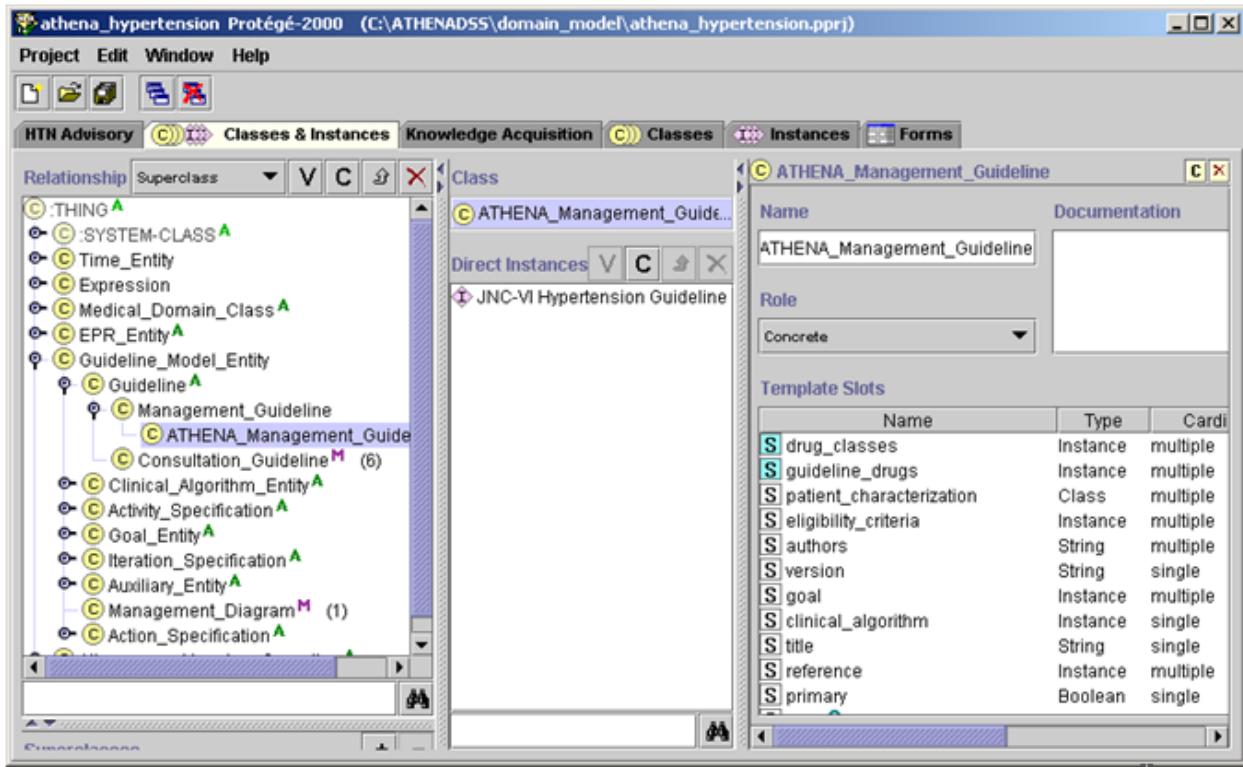


Figure 24 - Protégé graphical user interface showing the class hierarchy and the definition of a class

IV.1.2. Metaclass in Protégé

In Protégé, classes are themselves instances of other classes. Classes whose instances are classes are called *metaclasses*. Protégé defines a metaclass called :STANDARD-CLASS as the default metaclass for all classes. Users can define their own metaclasses by subclassing :STANDARD-CLASS. Figure 25 shows some of the metaclasses that have been defined for the EON/ATHENA-CDS system. This subsection will describe the reasons for users to define metaclasses. Details of the individual metaclasses will be discussed in Subsection IV.2.2.

There are two reasons for users of Protégé to define their own metaclasses:

1. Users can associate slots with a metaclass, as with regular classes. In Figure 26, the Diagnostic_Term_Metaclass has a DiagnosticCriteria slot. Thus, instances of Diagnostic_Term_Metaclass—e.g., the CRI (Renal Insufficiency) class (Figure 27)—may have values for the DiagnosticCriteria slot. The Guideline Interpreter would evaluate the DiagnosticCriteria slot value of CRI to determine whether a patient has renal insufficiency.

2. Another reason to have metaclasses is that, by making a collection of classes instances of a metaclass, special reasoning can be carried out for the collection. Thus, for example, by making drug concepts (such as lisinopril and atenolol) instances of the Medication_Metaclass, we can write decision criteria that range over these drug concepts (i.e., instances of Medication_Metaclass). This makes it possible to determine, say, whether a bad drug partner has already been prescribed for a possible drug recommendation.

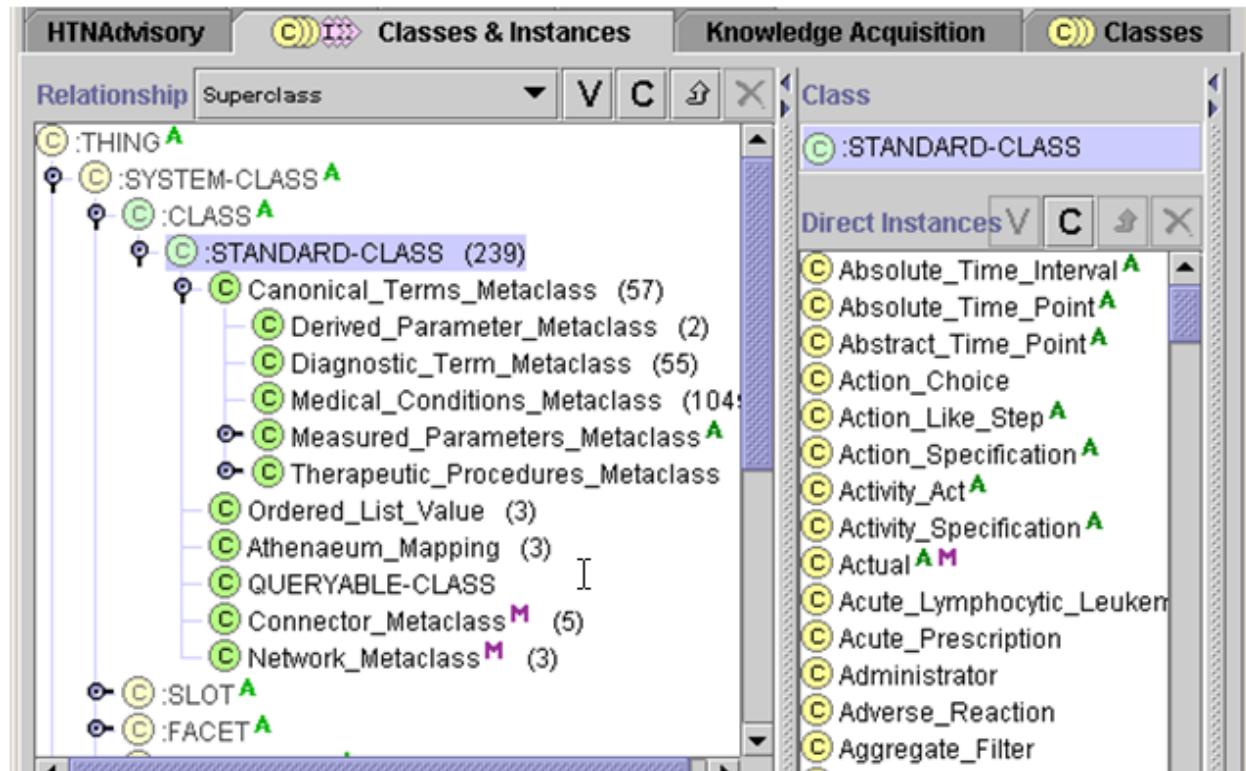


Figure 25 - The metaclass hierarchy in Protégé, showing classes (such as Absolute_Time_Interval) as instances of the :STANDARD-CLASS metaclass

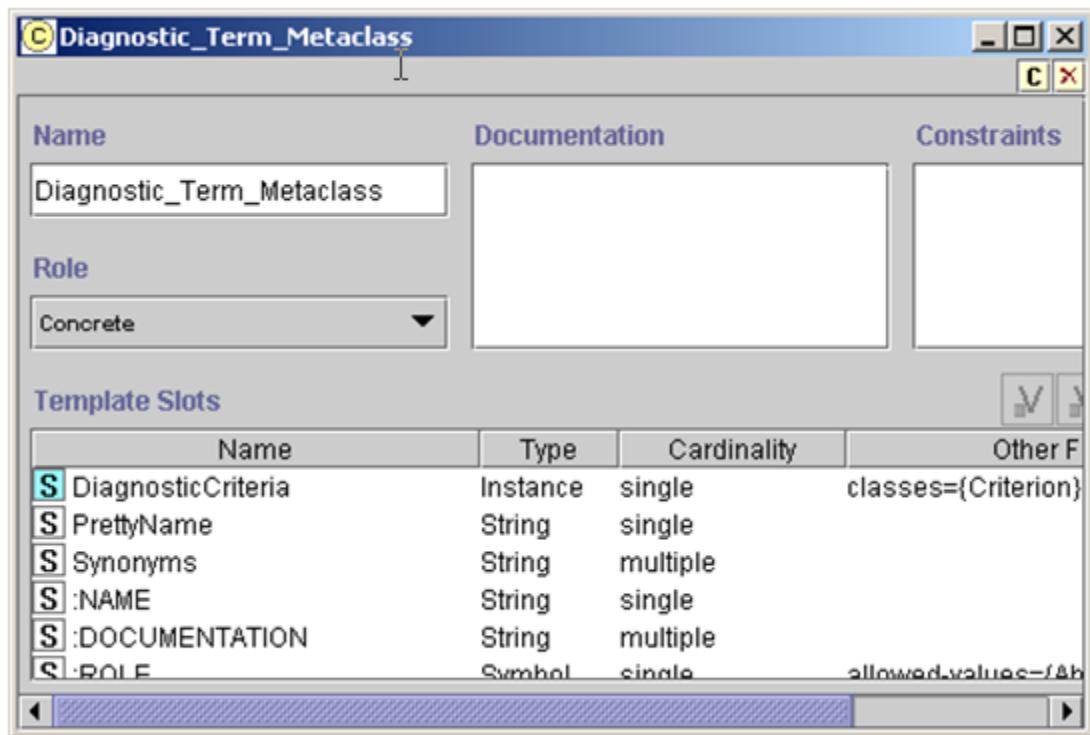


Figure 26 - The definition of Diagnostic_Term_Metaclass. DiagnosticCriteria is a slot whose value is specific to instances of this metaclass.

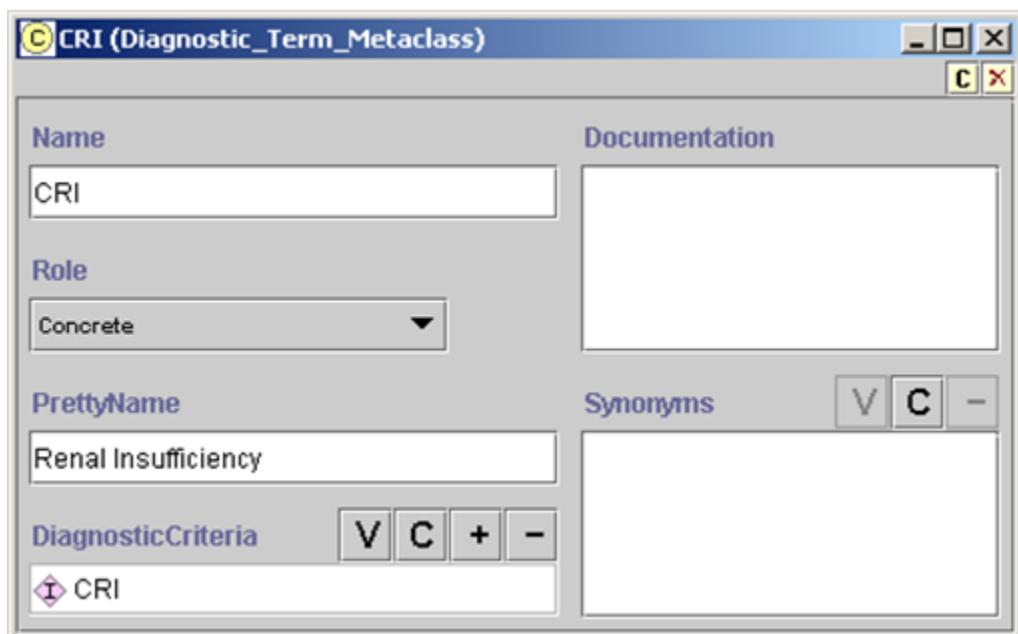


Figure 27 - The definition of the CRI class (an instance of the Diagnostic_Term_Metaclass). It has a PrettyName and a criterion defining Renal Insufficiency (in the DiagnosticCriteria slot) in terms of other data.

IV.1.3. Constraints in Protégé: Facets and PAL Constraints

In Protégé, slots are themselves frames that have properties. The properties of a slot, called *facets*, express constraints on possible values of slots. Figure 28 shows the facets of the *eligibility_criteria* slot as they appear in Protégé's UI. We can see that:

- values of the slot are constrained to be of the value type, Instance;
- the values must be instances of the allowed class, Criterion; and
- the slot may have zero or more values, since the *Cardinality multiple* checkbox is checked and there is no entry in the *Cardinality at least* box.

The Minimum and Maximum facets apply only to slots whose value types are integers or floating-point numbers. The Template Values of a slot are values inherited by all instances of the classes to which the slot is attached. (For example, if we associate a criterion as a template value of the *eligibility criteria* slot in the *Management_Guideline* class, then all guidelines—be they hypertension or screening guidelines—will have that criterion as one of their *eligibility criteria*.) The Default value of a slot is the value that a slot is initially given when an instance is created. The creator of the instance is free to change the initially assigned value. The Inverse Slot of a slot is one that has a reciprocal relationship with the slot (e.g., parent-of and child-of are inverses of each other: if John is the parent of Jim, then Jim is a child of John).

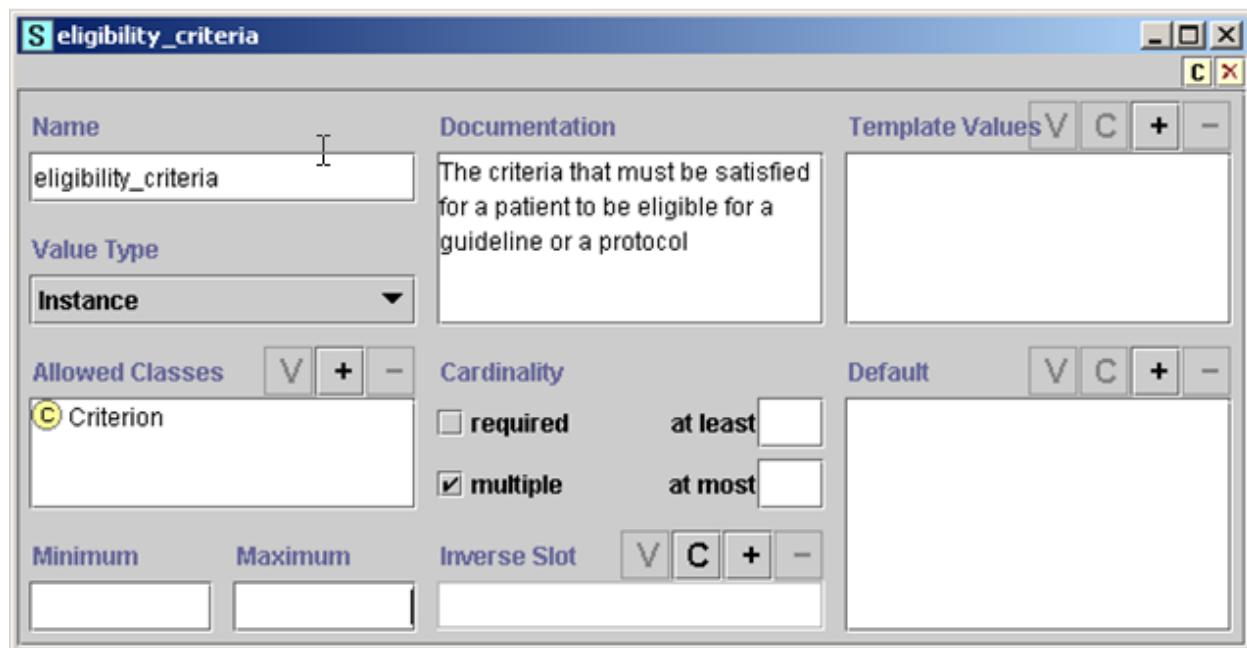


Figure 28 - The facets of the *eligibility_criteria* slot

Facets of a slot express constraints on a single slot. When a constraint is a complex relationship among multiple instances and slots, it can be written as a *Protégé Axiom Language (PAL) constraint* (Figure 29). A PAL constraint uses a logic-based language to express relationships among instances of a Protégé knowledge base that can span across multiple classes and slots.

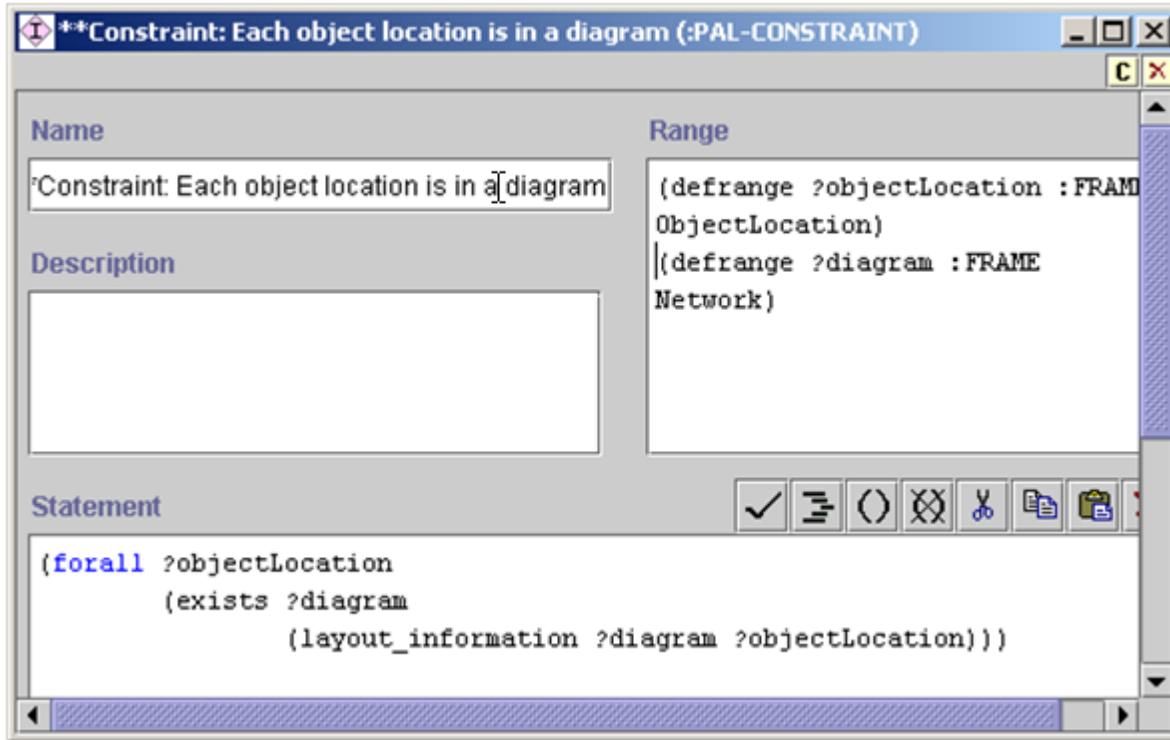


Figure 29 - A PAL constraint expressing the requirement that all instances of ObjectLocation must be part of a diagram, i.e., instances of the Network class

IV.1.4. Protégé's File Format

Protégé saves a knowledge base—called a *project* in Protégé—as a collection of three files. They have the extensions .pont, .pins, and .pprj. The pont file contains the definition of Protégé classes. The pins file contains the definition of and data about instances. The pprj file contains display information for classes and instances as well as information about relationships among different Protégé projects. The same pont and pins files may be used in different pprj files if different displays are desired for the same knowledge base. Thus, in ATHENA-CDS, a clinician may view and edit the ATHENA-CDS Knowledge Base using athena_hypertension.pprj, which references athena_all.pont and athena_all.pins and which has numerous UI customizations. The Guideline Interpreter, on the other hand—because it doesn't need the display information kept in the athena_hypertension.pprj file—loads the athena_server.pprj project, which also references the same athena_all.pont and athena_all.pins files.

IV.1.5. Protégé User Interface

This subsection introduces the Protégé user interface, describing its components and highlighting some of the common operations. It also gives two examples of working with the Protégé GUI:

1. searching for classes or instances whose display name matches a string, and
2. finding all places where a class or instance is referenced.

More information about using Protégé can be found at:

http://protégé.stanford.edu/doc/users_guide/index.html.

IV.1.5.1. Components of the Protégé UI

The Protégé UI is divided into tabs, each of which provides a view into the content of the knowledge base. This subsection describes how to work with the tabs—called Forms, Classes, Instances, Classes & Instances, and Knowledge Acquisition—occasionally referencing the Protégé User Guide.

Forms Tab

The basic UI paradigm in Protégé is that information associated with a frame (see Subsection IV.1.1) are displayed on a *form*. For each class, Protégé generates a prototypical form used to display and edit instances of that class. The Forms tab allows a developer to customize this automatically generated instance form of a class. Figure 30 shows the Forms tab for the ATHENA-CDS Knowledge Base. The left-hand side of the tab shows the class hierarchy of the knowledge base. Selecting a class in this hierarchy (e.g., the Scenario class) causes the prototypical form associated with the class to be displayed on the right side of the tab. Each slot of the class is shown with a default *slot widget*, a UI component that displays the slot value and through which the slot value can be edited. For example, in Figure 30, the *label* slot is shown as a text box with the title “Label” above it, the Boolean *new_encounter* slot is displayed as a check box entitled “New Encounter”. By default, slots that have instances as values are displayed with slot widgets that allow a user to view (**V** button), create (**C** button), add (**+** button), or remove (**-** button) instances as slot values.

The prototypical form associated with a class can be customized. By selecting a slot widget on the prototypical form and using the Selected Widget Type drop-down menu, a developer can change the slot widget associated with a slot. Figure 31 shows that the *new_encounter* Boolean slot can be displayed using either ComboBoxWidget or CheckBoxWidget. The Form Browser Key drop-down menu allows a user to select the slot whose value will be used as the display name of an instance. Figure 32 shows that the *label* slot has been selected as the slot whose value is the display name of Scenario instances.

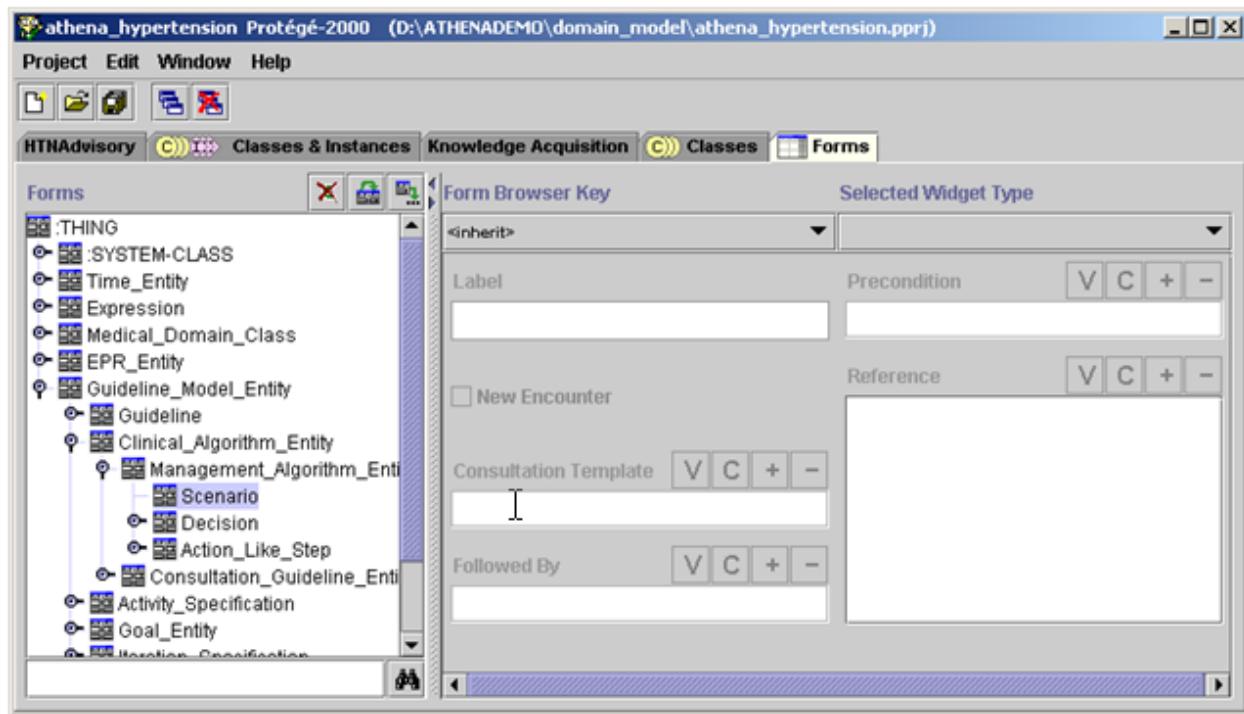


Figure 30 - The instance form associated with the Scenario class. It defines the UI for viewing and editing instances of the class.

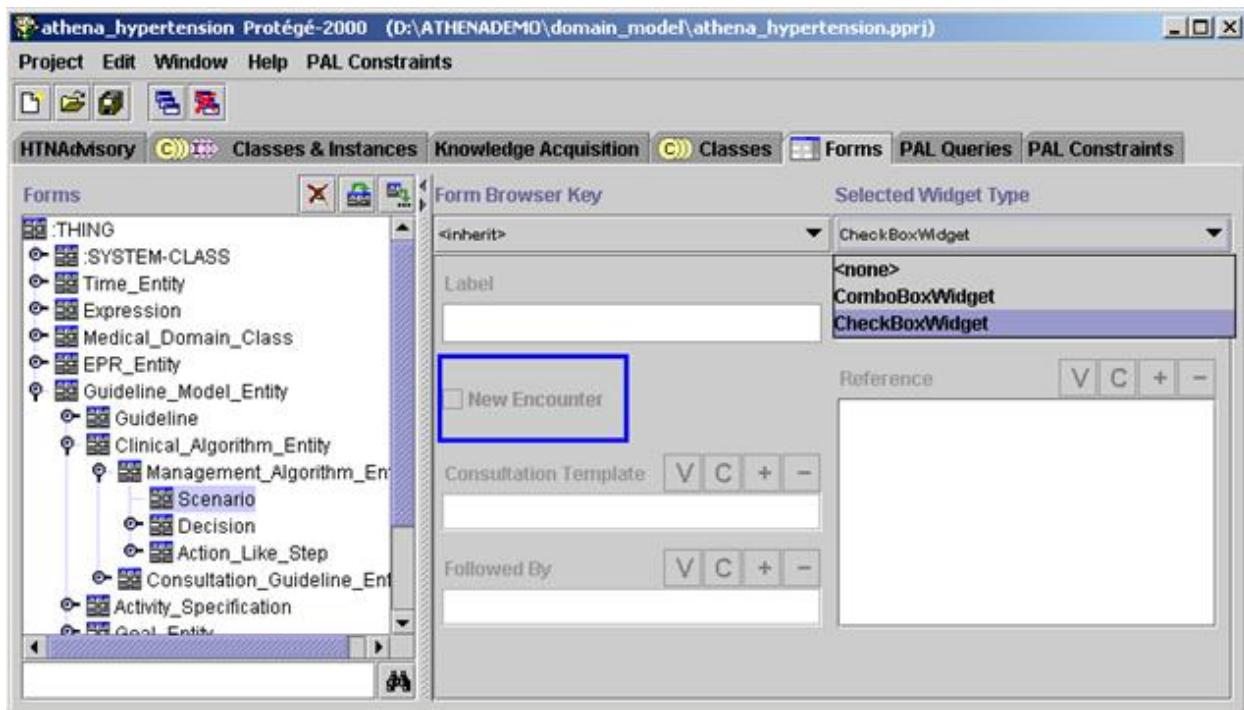


Figure 31 - Changing the slot widget assigned to the new_encounter slot

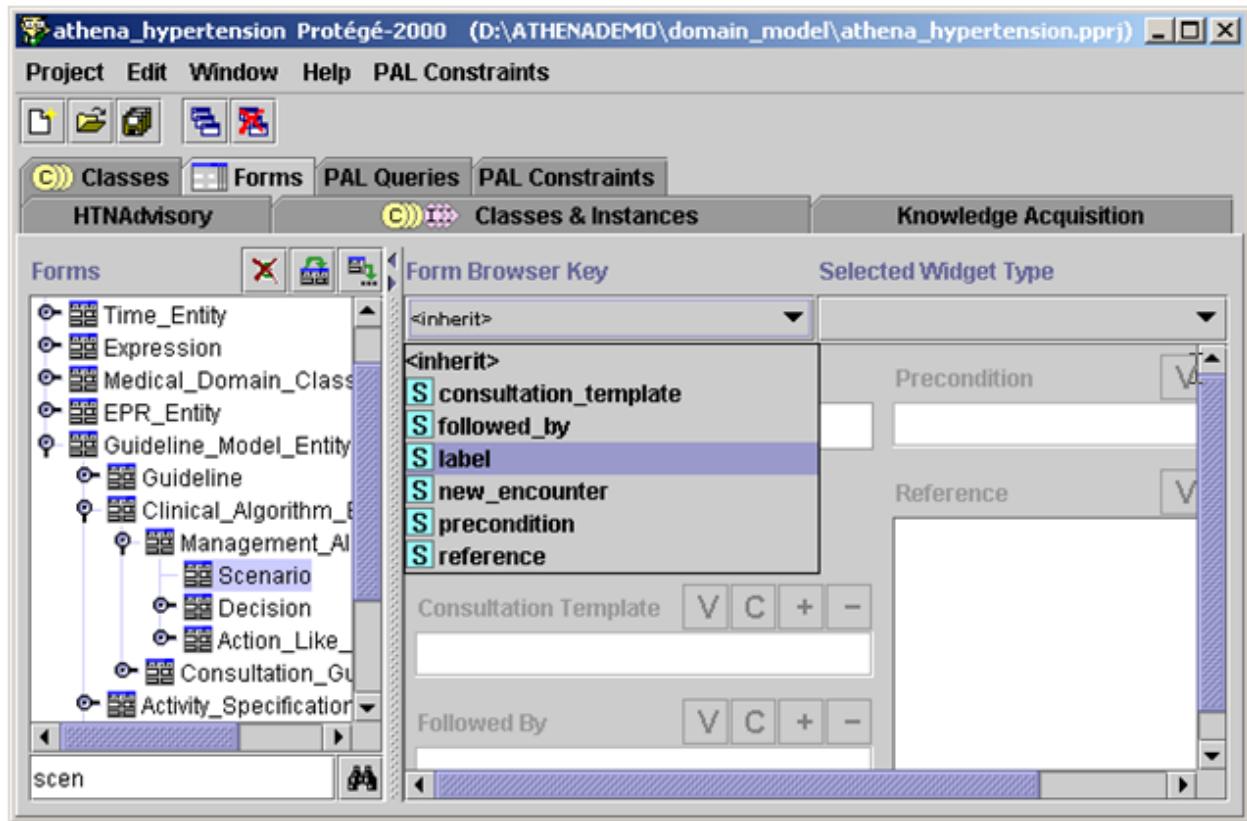


Figure 32 - Selecting the label slot to supply the display name of Scenario instances

Classes Tab

Quoting the Protégé User Guide:

"The Classes tab provides a single window in which you may view, create, and edit classes, which model concepts in your domain. An example is shown below (Figure 33). The window consists of three panes:

1. The **Class Relationship** pane in the upper left shows classes in a hierarchy and allows you to edit, create, and delete new classes. It also allows you to rearrange the class hierarchy by dragging a class to a replacement superclass.
2. The **Superclasses** pane in the lower left shows the superclasses of the selected class and allows you to add and remove superclasses for a class, as well as jump to a different superclass by clicking on it.
Note: If you cannot see the **Superclasses** Pane, your window may be too small. You can see the pane by enlarging your window or by dragging the slider bar at the bottom of the **Class Relationship** Pane.
3. When a single class is selected, the **Edit** pane on the right contains the Class Form for the selected class. The Class Form allows you to: name the class, choose its role, define constraints, provide a brief note, and,

most importantly, define and edit the template slots. The Class Form can also be displayed as a separate window by clicking the View  icon in the Class Relationship pane.”

The Class Form can also be viewed by double clicking on a selected class under the Classes tab.

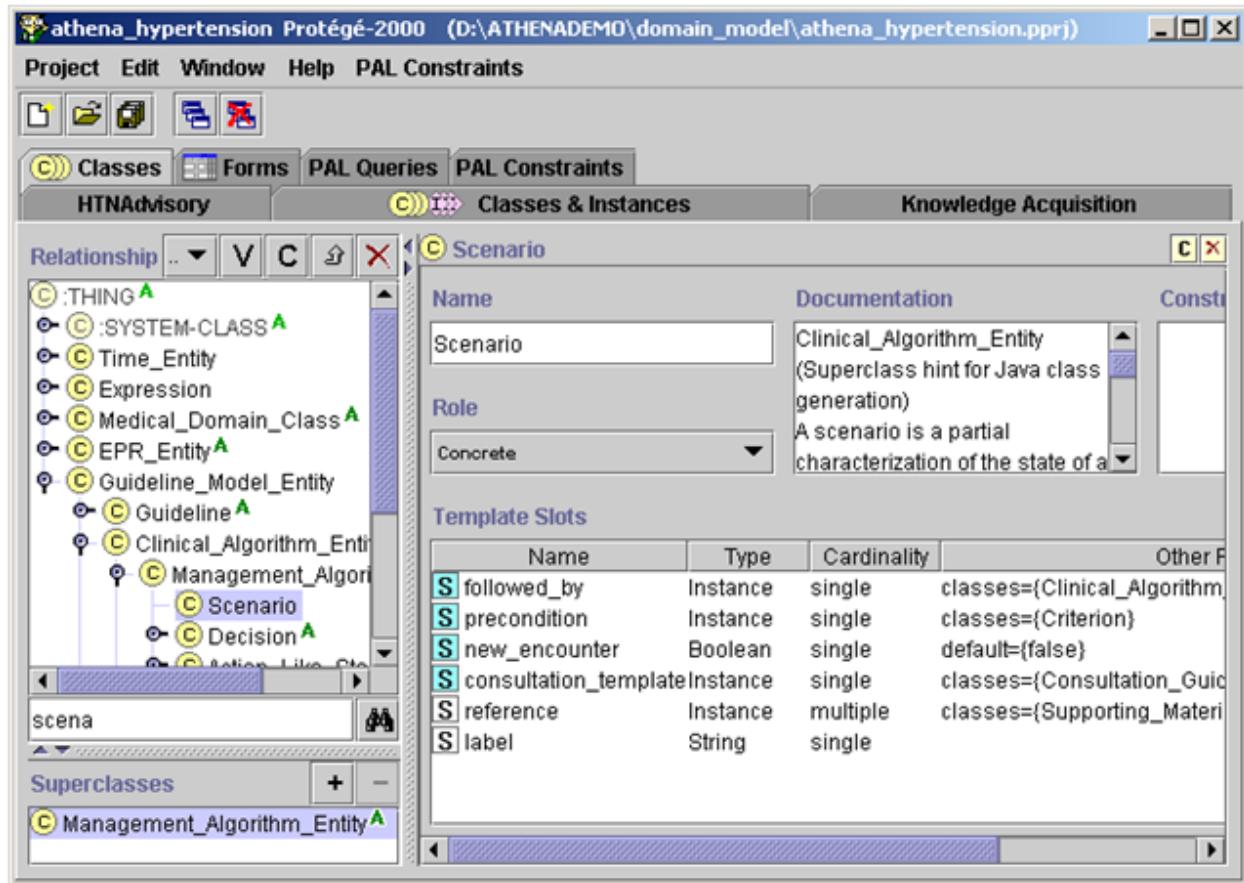


Figure 33 - The Classes tab in Protégé, showing the classes in the ATHENA-CDS Knowledge Base and the Class Form for editing the Scenario class

Instances Tab

Quoting the Protégé User Guide:

“The Instances tab provides a window in which you may view, create, and edit instances. (Classes model concepts in your domain, slots model properties of classes and any relationships between them, and instances model the actual data.) An example of the Instances Tab is shown below (Figure 34). The window consists of three panes:

1. A **Class pane** at the upper left shows the classes in a superclass/subclass relationship. The Instances Tab lets you view classes, but you cannot edit or rearrange them. ...

2. The **Direct Instances pane** in the center shows all the direct instances, if any, for the selected class, and allows you to view, edit, create, and delete direct instances.
3. When a single instance is selected, the Edit pane on the right contains the **Instance Form** for the selected instance. The Instance Form displays all the slots which apply to the instance, and allows you to edit them. The Instance Form can also be displayed as a separate window by clicking the View  icon in the Direct Instances pane.”

Double clicking on an instance in the Direct Instance pane under the Instances tab causes the Instance Form of that instance to pop up in a separate window. Elsewhere in this manual, an instance form is also called a “template”.

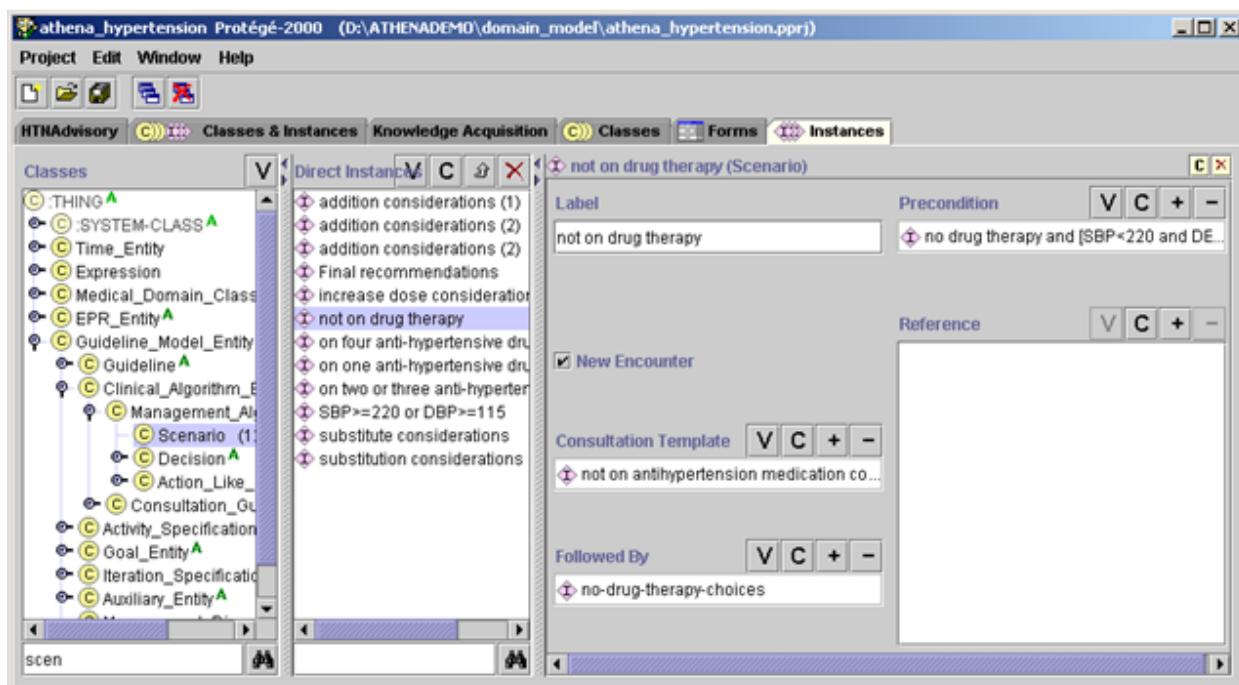


Figure 34 - The Instances tab, showing the Instance Form associated with the “not on drug therapy” instance of Scenario

Classes & Instances Tab

The Classes & Instances tab combines most of the functionalities of the Classes tab and the Instances tab in a single tab. Like the Instances tab, the Classes & Instances tab has three main panes: Class Relationship pane as it exists in the Class tab, Direct Instances pane, and the form for the selected class or instance. When a class is selected in the Class Relationship pane, but no instance is selected in the Direct Instances pane, then the form pane displays the class form associated with the selected class (Figure 35). When a class is selected in the Class Relationship

pane and an instance is selected in the Direct Instances pane, then the form pane displays the Instance Form associated with the selected instance (Figure 36).

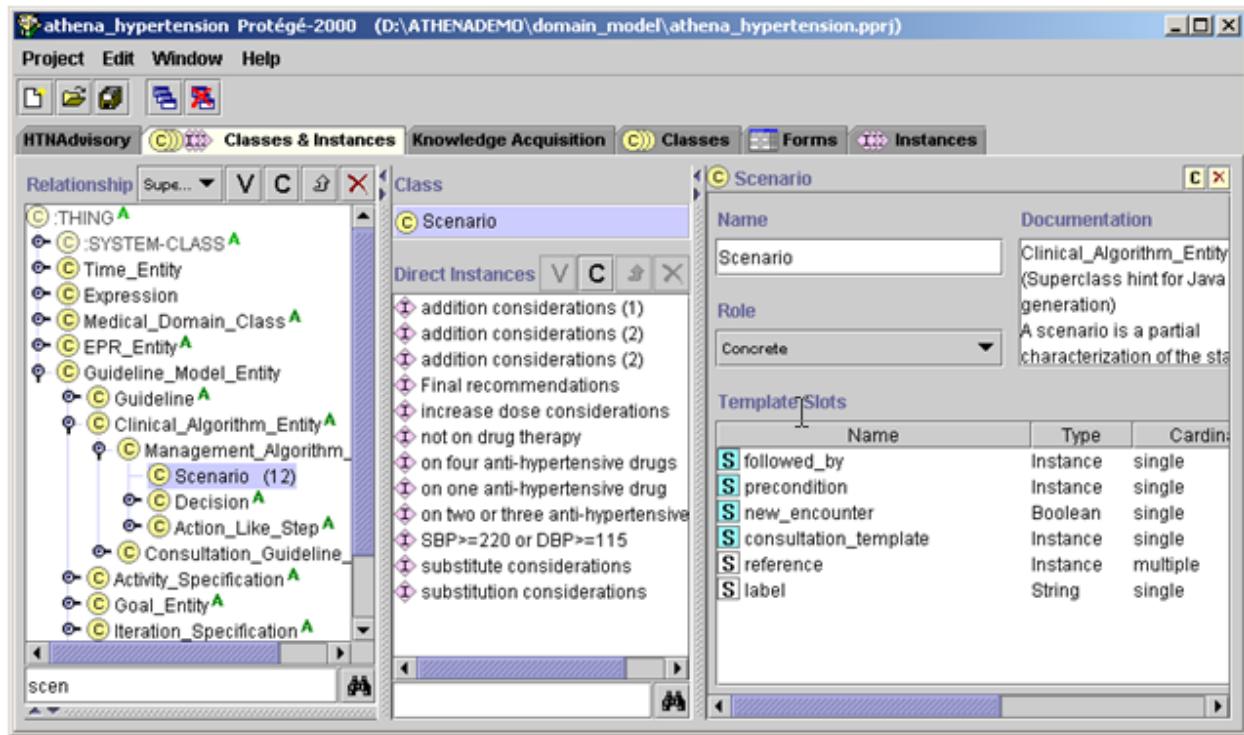


Figure 35 - The Classes & Instances tab. It shows a class hierarchy in the Relationship pane on the left, a Direct Instances pane in the center, and a Class Form for class Scenario on the right.

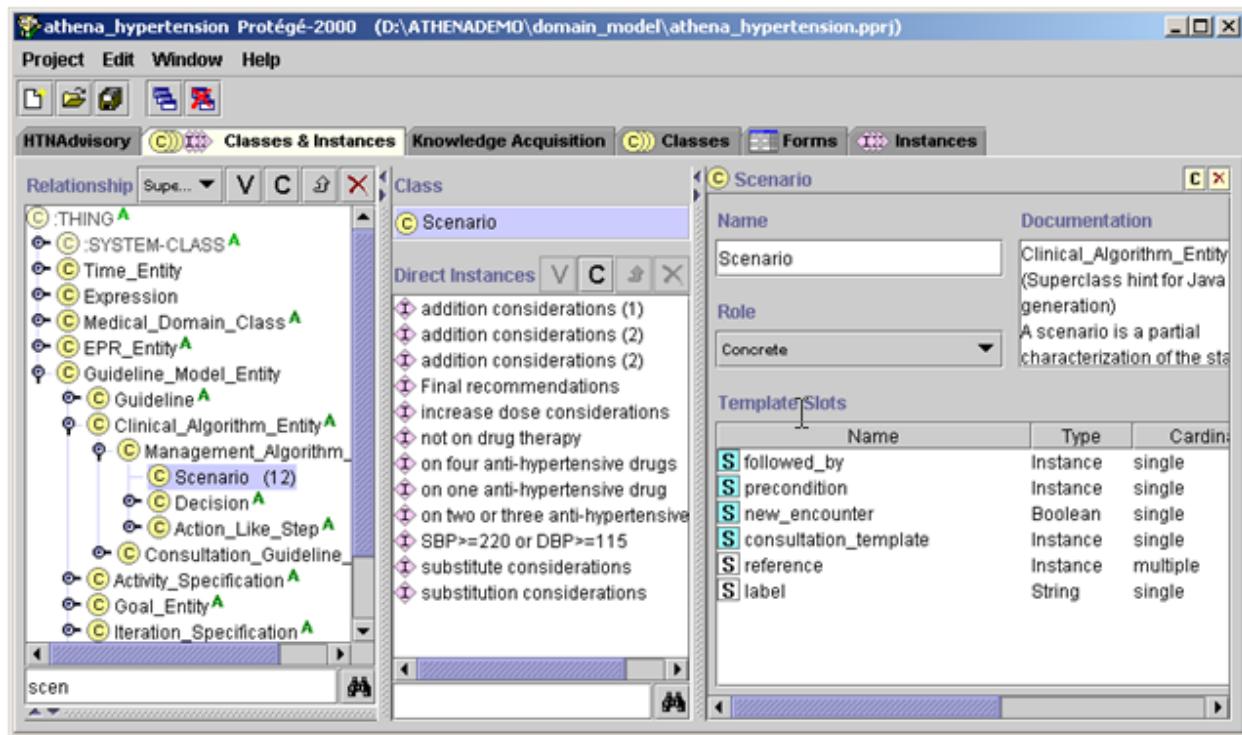


Figure 36 - The Classes & Instances tab, as in Figure 35, except that on the right it is displaying an Instance Form for the “not on drug therapy” instance

Knowledge Acquisition Tab

The Knowledge Acquisition tab is designed to facilitate navigation in a Protégé knowledge base by specifying an instance as the entry point for browsing the knowledge base. For the ATHENA-CDS Knowledge Base, the JNC-VI Hypertension Guideline instance of the ATHENA_Management_Guideline class is the starting point for browsing the ATHENA-CDS Knowledge Base (Figure 37).

athena_hypertension Protégé-2000 (C:\ATHENADEMO\domain_model\athena_hypertension....)

Project Edit Window Help

Classes Instances Forms PAL Queries

HTNAdvisory Classes & Instances Knowledge Acquisition

JNC-VI Hypertension Guideline (ATHENA_Management_Guideline)

Label JNC-VI Hypertension Guideline	Eligibility Criteria
Title	<ul style="list-style-type: none"> presence of diagnosis of hypertension absence of renovascular disease no diagnosis of pregnancy creatinine < 2.5 Absense of Secondary Hypertension absence of spinal cord injury absence of narcolepsy Not taking cyclosporine
Version June, 2001	Goal
Clinical Algorithm hypertension management diagram	<ul style="list-style-type: none"> BP target patient with DM, CHF or CRI BP target for patient without DM, CHF, and CRI
Authors NIH NHLBI Joint National Committee Mary Goldstein, MD Brian Hoffman, MD Susana Martins, MD MSc Robert Coleman, MS	Patient Characterization
Drug Classes	Guideline Drugs
<ul style="list-style-type: none"> ACE Inhibitor Metoprolol and terazosin Alpha Beta Blocker Alpha Blocker Angiotensin II Receptor Blocker Cardioselective Beta Blocker Non-cardioselective Beta Blocker 	<ul style="list-style-type: none"> acebutolol amiloride amlodipine amlodipine besylate atenolol captopril carvedilol

Figure 37 - The Knowledge Acquisition tab, showing the Instance Form of the JNC-VI Hypertension Guideline instance of the ATHENA_Management_Guideline

The ATHENA_Management_Guideline class was designed to make most important entries in the ATHENA-CDS Knowledge Base visible on a single form.

Diagram Widget

To graphically display and edit the clinical algorithm associated with a guideline, the ATHENA-CDS Knowledge Base uses the diagram widget to represent the steps and connections among the steps (Figure 38). For the Management_Diagram class, the Steps slot is configured such that the associated diagram widget shows the allowed classes of the slot (Choice_Step, Action_Choice, Scenario, and Case_Step). They are available in the drawing palette of the diagram. For example, a user can create an instance of Scenario in the drawing canvas by selecting and dragging a Scenario icon from the drawing palette to the drawing canvas. Similarly, a connection between two icons (representing instances) can be created by selecting and dragging the connection arrows and attaching the ends of each arrow to the appropriate icons.¹²

¹² The diagram widget described here works only in Protégé 1.9 or earlier. Since Protégé 2.0, a new graph widget has replaced the diagram widget.

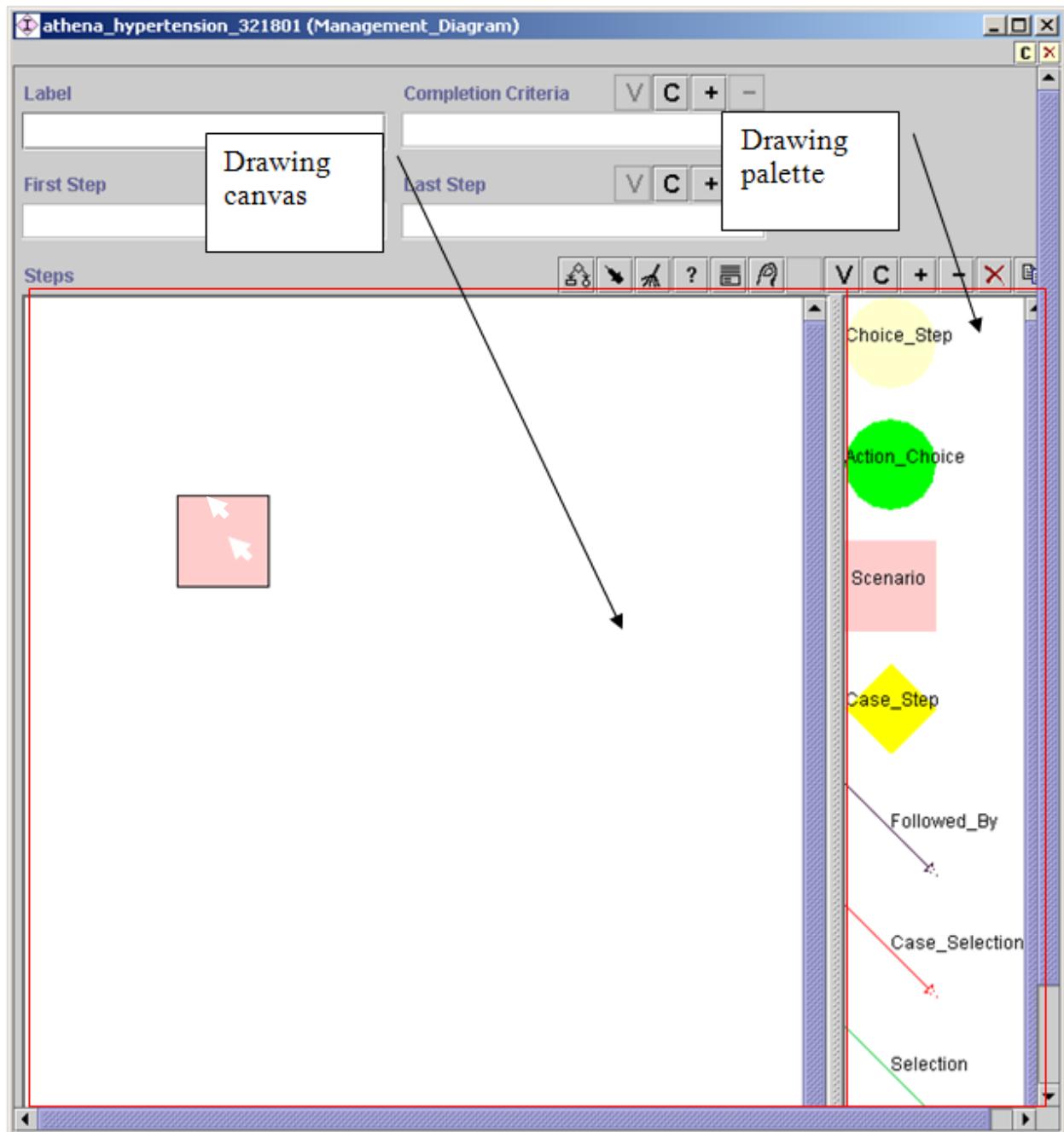


Figure 38 - The diagram interface for creating a clinical algorithm. A user creates an object in the diagram by selecting and dragging an icon from the drawing palette to the drawing canvas.

Figure 39 shows a portion of the main clinical algorithm of the ATHENA-CDS Knowledge Base.

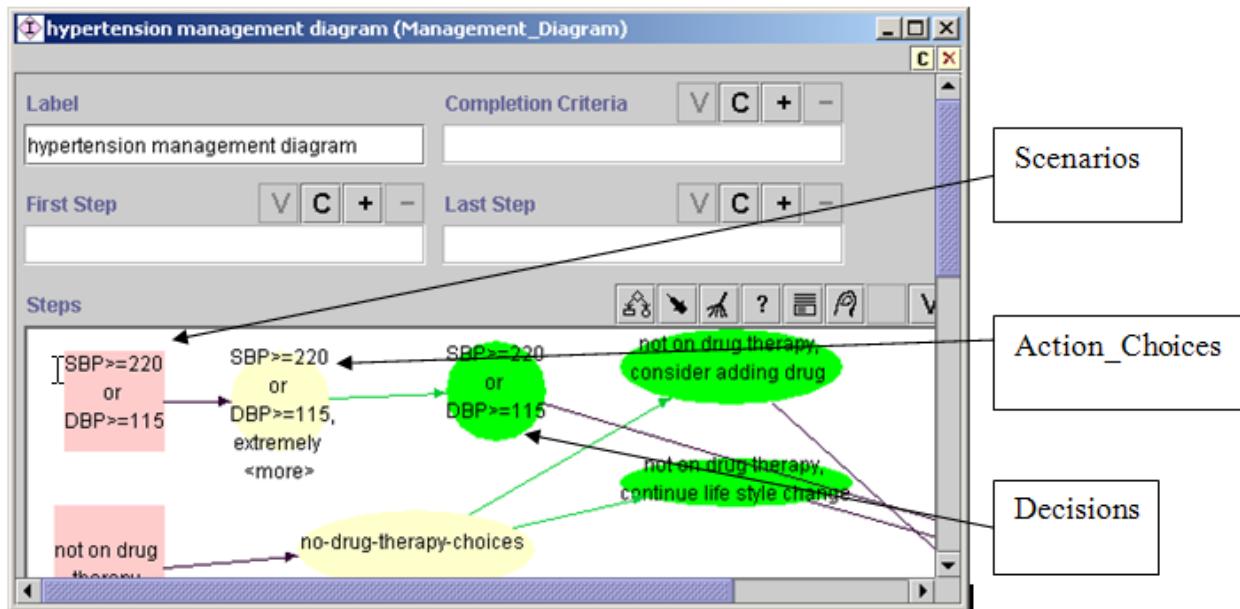


Figure 39 - Hypertension management diagram, showing Scenarios, Decisions (Choice_Steps), and Action_Choices

IV.1.6. Two Common Tasks

The following subsections describe two tasks that a user will commonly carry out in Protégé. One is using the search function to find existing classes or instances. The other is finding references to classes, slots, or instances in one or more places throughout the ATHENA-CDS Knowledge Base.

Using the Search Function to Find Classes or Instances

In Protégé, a user can search for existing classes or instances by their display names. This is done using the binocular search pane available in a number of Classes & Instances panes and dialog boxes (Figure 40). The binocular search pane consists of the search-string text field and the binocular search button.

Suppose a user is looking for a class with a name that begins with “anti”. He or she can enter the search string “anti” in the search-string text field and click the binocular button (Figure 40). If there is one match, that class will be highlighted. If there is more than one match, Protégé will show a dialog box containing all matches (Figure 41), and the user will be able to select one. String comparisons in searches are *not* case sensitive.

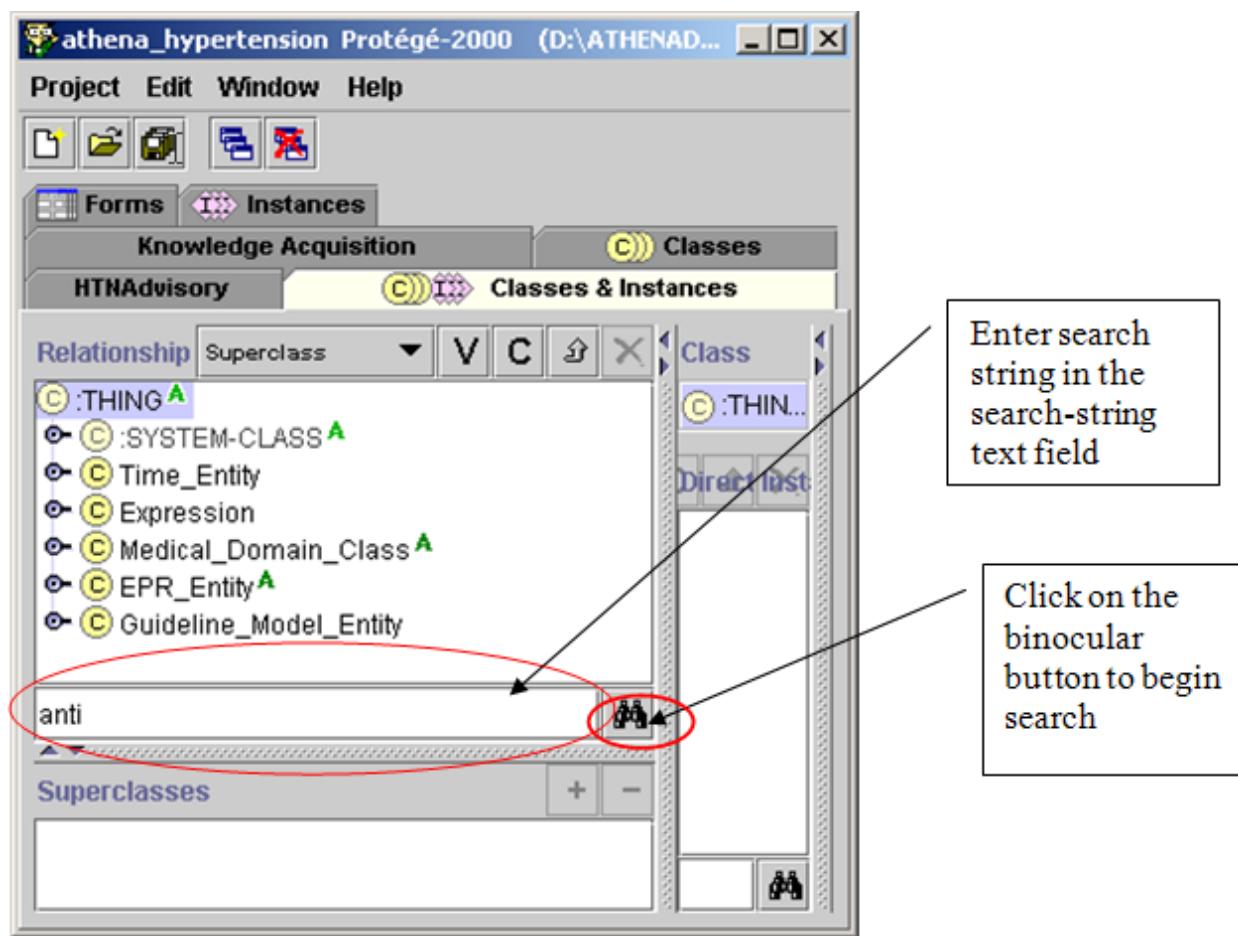


Figure 40 - Using the binocular search pane to look for classes with a name beginning in “anti”

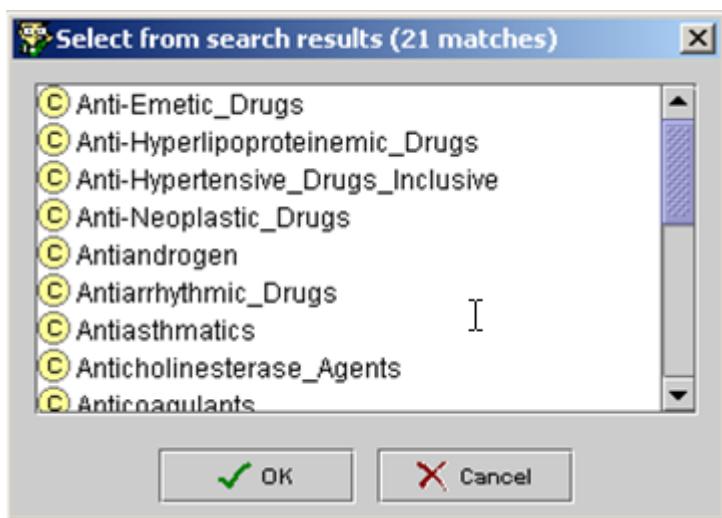


Figure 41 - Results for the search string “anti”

If you want to find classes with names containing “anti” in any position, you can add the wildcard character * at the beginning of the search string. Protégé will return all classes with names containing “anti” (Figure 42).

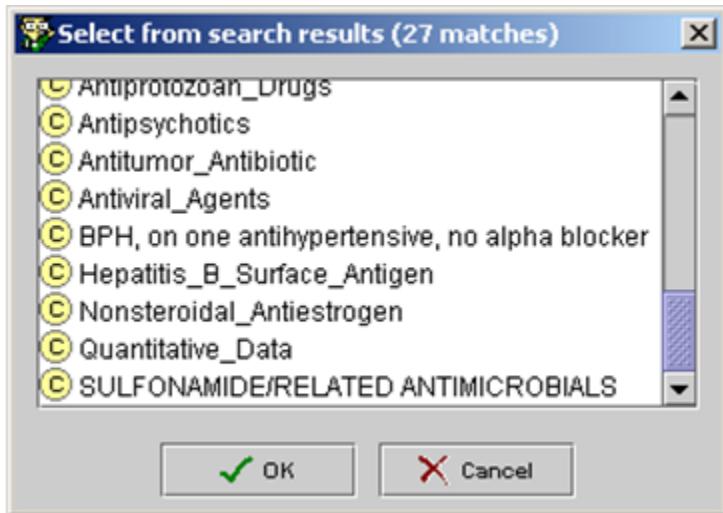


Figure 42 - Search results for the search string “*anti”

Finding References to Classes, Slots, or Instances throughout the Knowledge Base

A class or an instance may be used in one or more places in the knowledge base. If a user wants to modify or delete a class or an instance, it is essential to know all places this class or instance is used. He or she can use the references buttons, highlighted in Figure 43, to find all references to a particular class or instance.

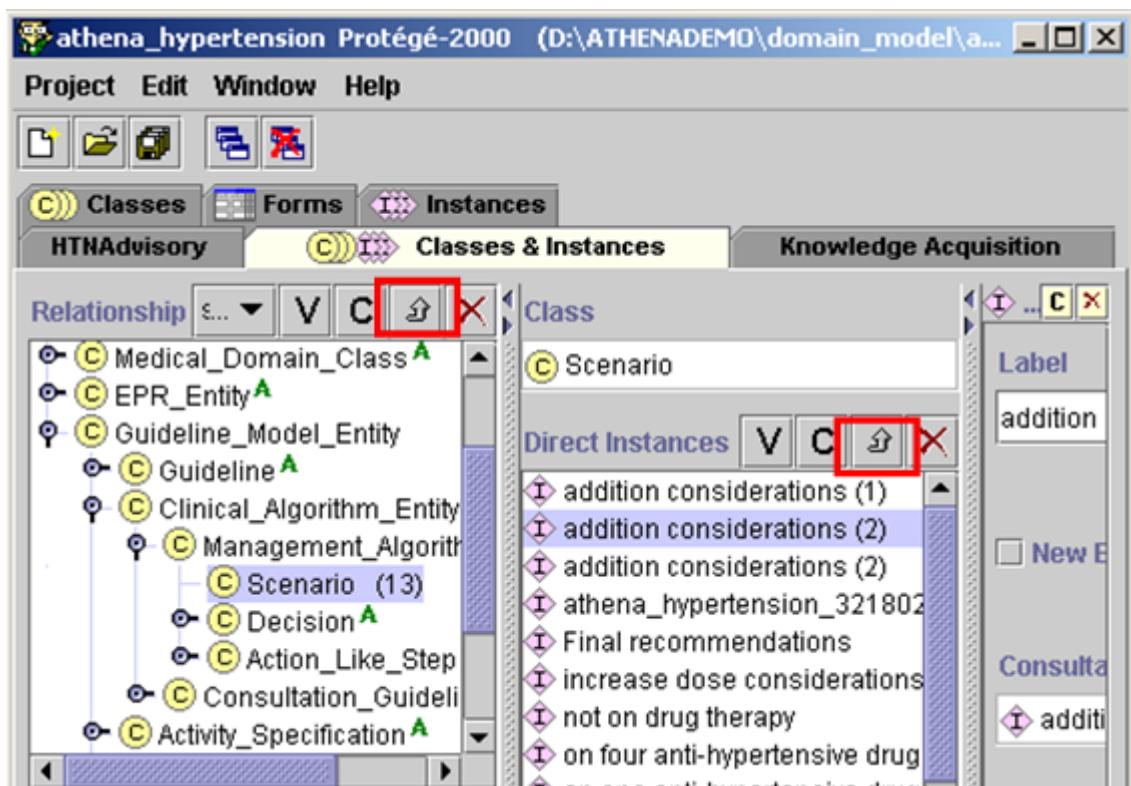


Figure 43 - References buttons (up arrow, between C and X buttons) in the Classes & Instances tab

If you select the Scenario class and click on the references button in the class hierarchy pane, for example, a window showing all references to the Scenario class will pop up (Figure 44). The window shows that: the Scenario class is an allowed class of the first_object slot in the Followed_By class; it is the direct instance of the :STANDARD-CLASS metaclass; it is the direct subclass of the Management_Algorithm_Entity class; it is the direct type of a number instances; and it is the slot value type of the previous_scenarios slot.

Frame	Slot	Facet
C Followed_By M	S first_object	allowed-classes
C :STANDARD-CLASS	S :DIRECT-INSTANCES	
C Management_Algorithm_Entity A	S :DIRECT-SUBCLASSES	
I addition considerations (1)	S :DIRECT-TYPE	
I addition considerations (2)	S :DIRECT-TYPE	
I addition considerations (2)	S :DIRECT-TYPE	
I athena_hypertension_321802	S :DIRECT-TYPE	
I Final recommendations	S :DIRECT-TYPE	
I increase dose considerations	S :DIRECT-TYPE	
I not on drug therapy	S :DIRECT-TYPE	
I on four anti-hypertensive drugs	S :DIRECT-TYPE	
I on one anti-hypertensive drug	S :DIRECT-TYPE	
I on two or three anti-hypertensive drugs	S :DIRECT-TYPE	
I SBP>=220 or DBP>=115	S :DIRECT-TYPE	
I substitute considerations	S :DIRECT-TYPE	
I substitution considerations	S :DIRECT-TYPE	
S previous_scenarios	S :SLOT-VALUE-TYPE	
C previous_scenarios	C :SLOT-VALUE-TYPE	

Figure 44 - Window showing all references to the Scenario class

From the window that shows all references to the Scenario class, you can select a class, slot, or instance (e.g., Management_Algorithm_Entity class in Figure 45), and click on the references button on the form to open another window that shows all references to the selected frame (e.g., Figure 46 shows all references to the Management_Algorithm_Entity class in the knowledge base).

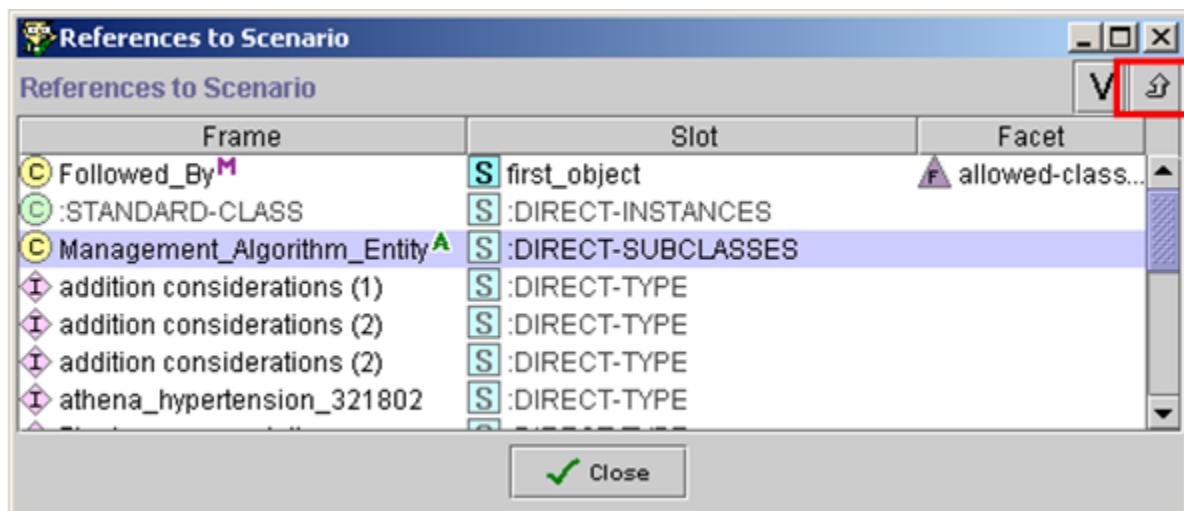


Figure 45 - Selecting a frame (e.g., Management_Algorithm_Entity class) in the References to Scenario window, and clicking on the references button to find all references to the selected frame

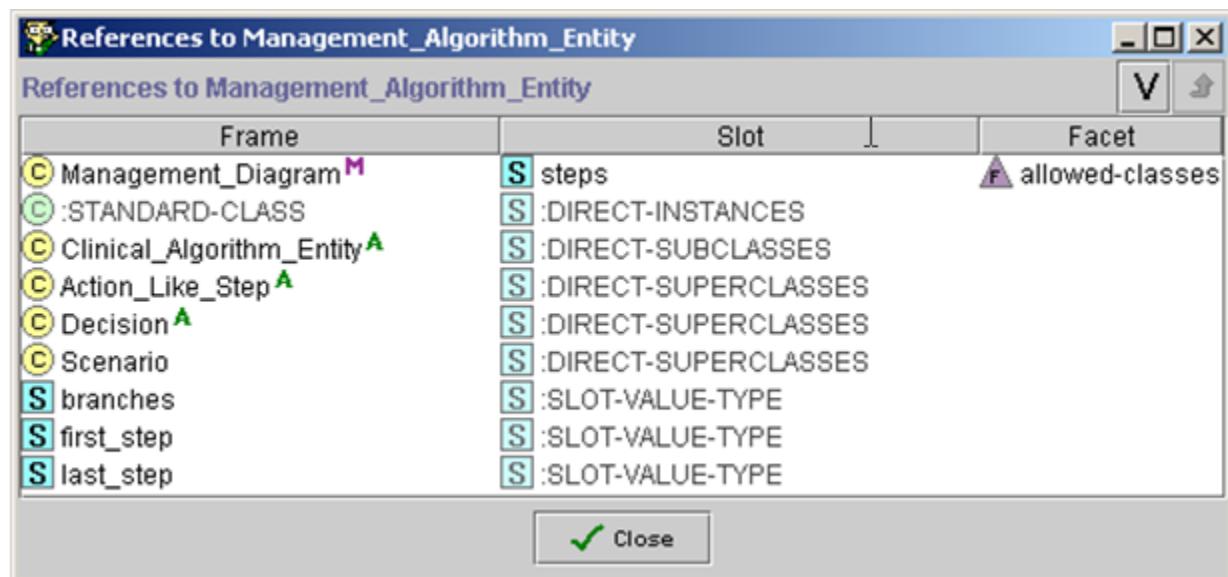


Figure 46 - All references to Management_Algorithm_Entity

IV.2. EON Models and the ATHENA-CDS Knowledge Base

The ATHENA-CDS Knowledge Base encodes select, authoritative hypertension knowledge in the format required by EON models. These models, consisting of an extensible set of Protégé projects, structure the knowledge and data so they can be used to encode guidelines and

protocols in a form suitable for generating patient-specific recommendations. The EON models include:

1. *Patient Data Model* (EPR_Entity hierarchy) – defines the structure of patient information used by the EON system. It converts the clinical data from the electronic medical record (or other source) into a format the Guideline Interpreter can process.
2. *Medical Concept Model* (Medical_Domain_Class hierarchy) – primarily defines the medical vocabulary used in encoding guidelines. The terms in the medical concept model are mapped to standard terminologies or to the terminology used in the host information system.
3. *EON Guideline Model* (Guideline_Model_Entity and Expression hierarchy) – defines the structure of a computable EON guideline.

As shown in Figure 47, the EON Guideline Model contains:

1. *Eligibility criteria* – define the target population of the guideline. For example, a guideline on Hypertension would have a target population of patients with high blood pressure as an eligibility criterion.
2. *Goals* (e.g., target blood pressures) – specify patient states the guideline aims to help achieve.
3. *Abstractions about patients* (e.g., the risk group to which a patient belongs) – represent interpretations about a patient's medical condition.
4. *A clinical algorithm* – organizes a collection of patient scenarios, decisions to be made, and possible action choices. Each action choice includes decision criteria for giving preference to the actions specified in the choice. The actions may include sending a message, referring a patient, evaluating activities to recommend, or starting, stopping and modifying activities.
5. *Activity specification* – represents an action that can take place over time (e.g., taking a medication to manage chronic problems). Activity specifications have properties, such as compelling indications, relative indications, relative contraindications, and absolute contraindications, that can be used to determine whether the activity is appropriate. Drug_Usage and Guideline_Drug are the two classes of activities heavily used in the ATHENA-CDS Knowledge Base. Instances of Drug_Usage (e.g., ACE Inhibitors) often contain information on classes of drugs, while instances of Guideline Drug contain specific information about a particular drug in a class (e.g., Captopril).

6. *Computable expressions* (e.g., eligibility criteria) – expressions, written using the expression languages available in the EON Guideline Model. They can be evaluated using coded patient data to infer valid statements about a patient.

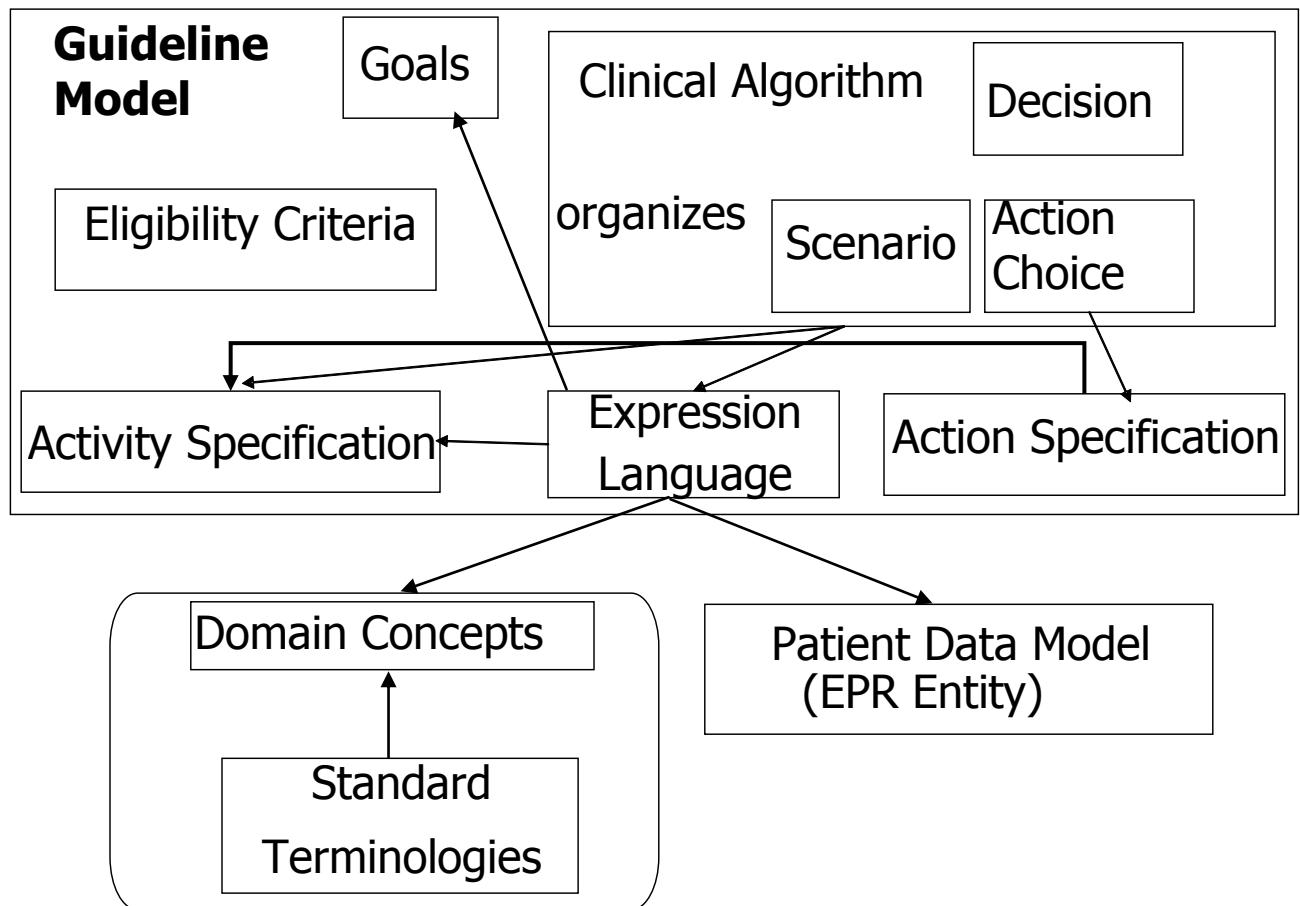


Figure 47 - Overview of the relationships among components of EON models

The following subsections describe the EON models and ATHENA-CDS Knowledge Base in more detail. They identify three levels: those features of the knowledge base that have impact on the display of the ATHENA-CDS GUI (as of November 2005), those features that are implemented by the EON Guideline Interpreter (as of November 2005) but are not displayed on the ATHENA-CDS GUI, and those features of the EON Guideline Model that are not fully implemented by the Guideline Interpreter. The subsections cover the patient data model, the guideline concept model, the EON Guideline Model, and the EON expression language.

IV.2.1. Patient Data Model

The EON patient data model defines the structure of the patient data used by the EON system. The Guideline Interpreter, for example, uses the data to determine whether a patient satisfies the eligibility criteria of the guidelines represented in the ATHENA-CDS Knowledge Base. Patient data are modeled as static (i.e., unchanging), as time stamped, or as having a time interval during which the information they represent is valid. For example, laboratory test results may be modeled as instances of *Numeric_Entry*, which has a code, a numeric value, and a time stamp. The Guideline Interpreter evaluates decision criteria using patient data represented in the format of the patient data model. The EON patient data model, represented by subclasses of the *EPR_Entity* class (see Figure 48), consists of the following classes:

- *Encounter class* – represents records of the encounters a patient has with health-care providers.
- *Patient class* – represents non-varying demographic information about specific patients (e.g., sex and date of birth).
- *Note_Entry class* – describes time-stamped non-numeric observations made by clinicians.
- *Numeric_Entry class* – represents results of quantitative measurements.
- *Adverse_Event class* – models adverse reactions to specific substances.
- *Condition class* – represents medical conditions that persist over intervals of time (i.e., it has a start time and may have a stop time).¹³
- *Medication class and Procedures class* – two intervention classes, they model drug prescriptions and other medical procedures that have been recommended, authorized, or used.

All entities in the patient data model are assertions about the demographic and clinical conditions of specific patients. The model is not designed to replicate all the contents of an electronic medical record. Rather, it includes only those distinctions relevant to modeling guidelines and protocols.

¹³ The Condition class is not used in ATHENA.

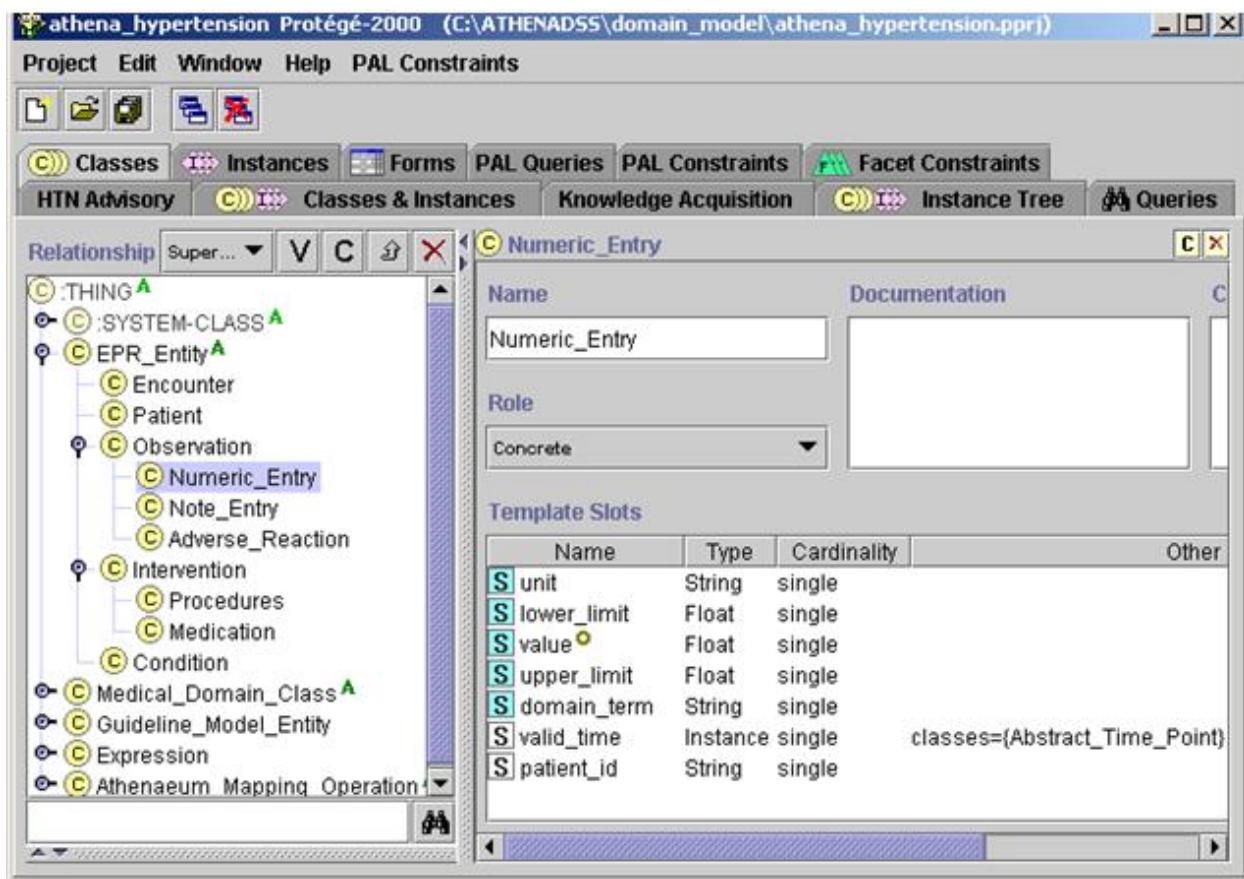


Figure 48 - The classes in the EON patient data model (EPR_Entity)

IV.2.2. Medical Concept Model

The medical concept model contains classes that represent:

1. terminological concepts necessary to encode patient information and guideline statements,
2. references to supporting material, and
3. relationships between different drugs, and between drugs and medical conditions.

These concepts and relationships are organized in a taxonomic hierarchy (see Figure 49). The particular collection of classes represented in the current EON medical concept model is a historical legacy that does not conform to any external standard such as SNOMED Clinical Term.

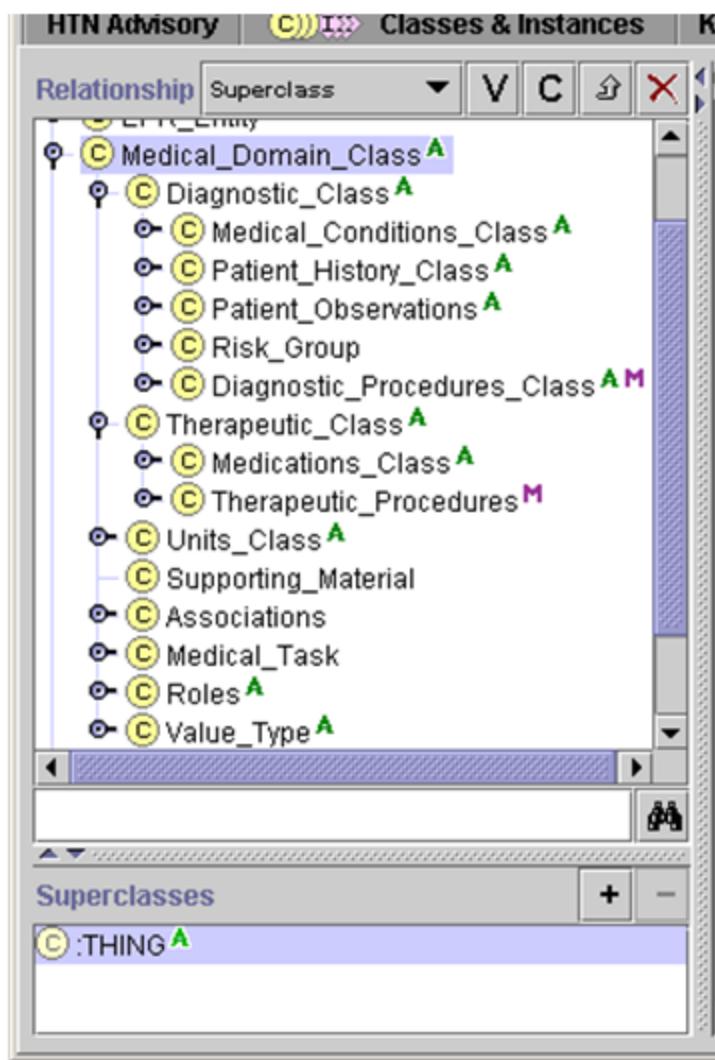


Figure 49 - The top-level classes in the medical concept model (subclasses of `Medical_Domain_Class`)

Classes in the `Medical_Domain_Class` hierarchy are instances of user-defined metaclasses. The important metaclasses to understand are:

1. *Canonical_Terms_Metaclass* – `Canonical_Terms_Metaclass` is the root of most metaclasses used in the `Medical_Domain_Class` hierarchy. (The only exception is `Ordered_List_Value`, see below). Medical concepts represented in the hierarchy should be instances of this metaclass or its subclasses. The `Canonical_Terms_Metaclass` provides a `PrettyName` slot that can be used to provide a more human-understandable name for the class. The `Synonyms` slot is not used.

2. *Medical_Conditions_Metaclass* – Instances of *Medical_Conditions_Metaclass* represent findings, diagnosis, and problems. They should be subclasses of the *Medical_Conditions_Class* in the *Medical_Domain_Class* hierarchy (Figure 50). A term representing a finding or diagnosis, such as *Atrial_Fibrillation*, may be mapped to patient data by making codes used in data subclasses of the guideline concept (e.g., the ICD9 codes 427.31 and 427.32). Like the *Canonical_Terms_Metaclass*, only the *PrettyName* slot is used in the current system. Note that the current ATHENA-CDS Knowledge Base has PAL criteria (see IV.2.4.2) whose variables range over instances of *Medical_Conditions_Metaclass*. (So if a finding is not an instance of *Medical_Conditions_Metaclass*—e.g., it is an instance of *Canonical_Terms_Metaclass*—it will not be considered when the system looks for, for instance, the contraindications of a drug.)

3. *Diagnostic_Term_Metaclass* – *Diagnostic_Term_Metaclass* provides a second way to relate guideline concepts to terms used in data. Instances of *Diagnostic_Term_Metaclass* have a *DiagnosticCriteria* slot that takes a Boolean criterion. If, for a patient, the criterion evaluates to *true*, then the EON Guideline Interpreter concludes that the finding is present for the patient. For example, in the ATHENA-CDS Knowledge Base, “hypertension without comorbidities that compellingly indicates thiazides or beta blocker” is represented as an instance of *Diagnostic_Term_Metaclass*, with *DiagnosticCriteria* “presence of hypertension and absence of myocardial infarction, diabetes, heart failure, and isolated systolic hypertension”.

4. *Derived_Parameter_Metaclass* – An instance of a *Derived_Parameter_Metaclass* is used to represent a concept whose value may come from multiple sources. In ATHENA-CDS, *Treatment_Systolic_BP* and *Treatment_Diastolic_BP* are instances of the *Derived_Parameter_Metaclass* where the *definition* slot specifies an ordered list of queries that can be used to obtain values for the two parameters. For example, the value for *Treatment_Systolic_BP* is derived by querying sequentially for: *MD_Typical_Systolic_BP* (the blood pressure a clinician uses for the purpose of managing hypertension), *MD_Clinical_Systolic_BP* (the most recent blood pressure measurement entered by a clinician through the ATHENA-CDS SYSTEM interface), and *DB_Systolic_BP* (the most recent blood pressure stored into the VistA database). The Guideline Interpreter uses the first data returned by this ordered list of queries. Thus, *MD_Typical_Systolic_BP*, if present, will be used instead of the *DB_Systolic_BP* blood pressure data in the database.

5. *Interval-Valued_AtomicTest_Metaclass* – Instances of *Interval-Valued_AtomicTest_Metaclass* are used to represent test results, such as the level of serum creatinine, that may have upper and lower limits of normal. The Guideline Interpreter also uses the *precision* slot to determine the number of digit to the right of the decimal point in displaying values of the test result.

6. *Medications_MetaData* – Instances of the Medications_MetaData are the generic drug names such as terazosin and lisinopril. Only the PrettyName slot is used. Again, because current PAL criteria use variables that range over instances of Medications_MetaData, a drug that is not an instance of this metaclass will be missed in the evaluation of PAL criteria.

7. *Ordered_List_Value* – Instances of the Ordered_List_Value metaclass have pointers to the next and previous instances in a list. This metaclass is used to model the sequence relationship among High_Dose, Medium_Dose, and Low_Dose.

Other metaclasses, such as Drug_Category_MetaData, play no role in the interpretation of the knowledge base.

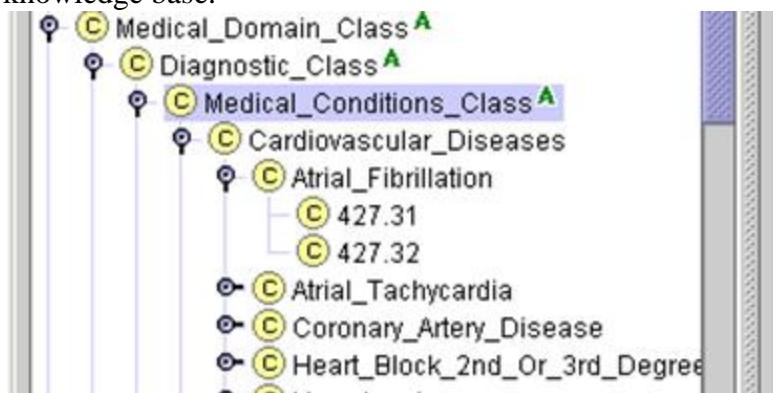


Figure 50 - Part of the ATHENA-CDS Medical Conditions Class hierarchy. Those classes in the hierarchy that have no subclasses should be either mapped to terms used in patient data or defined using Diagnostic_Term_MetaData.

IV.2.2.1. Terminology Hierarchies

The Diagnostic_Class, Therapeutic_Class, Medical_Task, Value_Type, and Units_Class hierarchies supply controlled terminologies for the formal encoding of guideline knowledge in the ATHENA-CDS Knowledge Base and for patient data (Figure 51). Units_Class is not used by the Guideline Interpreter, which assumes that the data and knowledge base are using consistent units of measure.

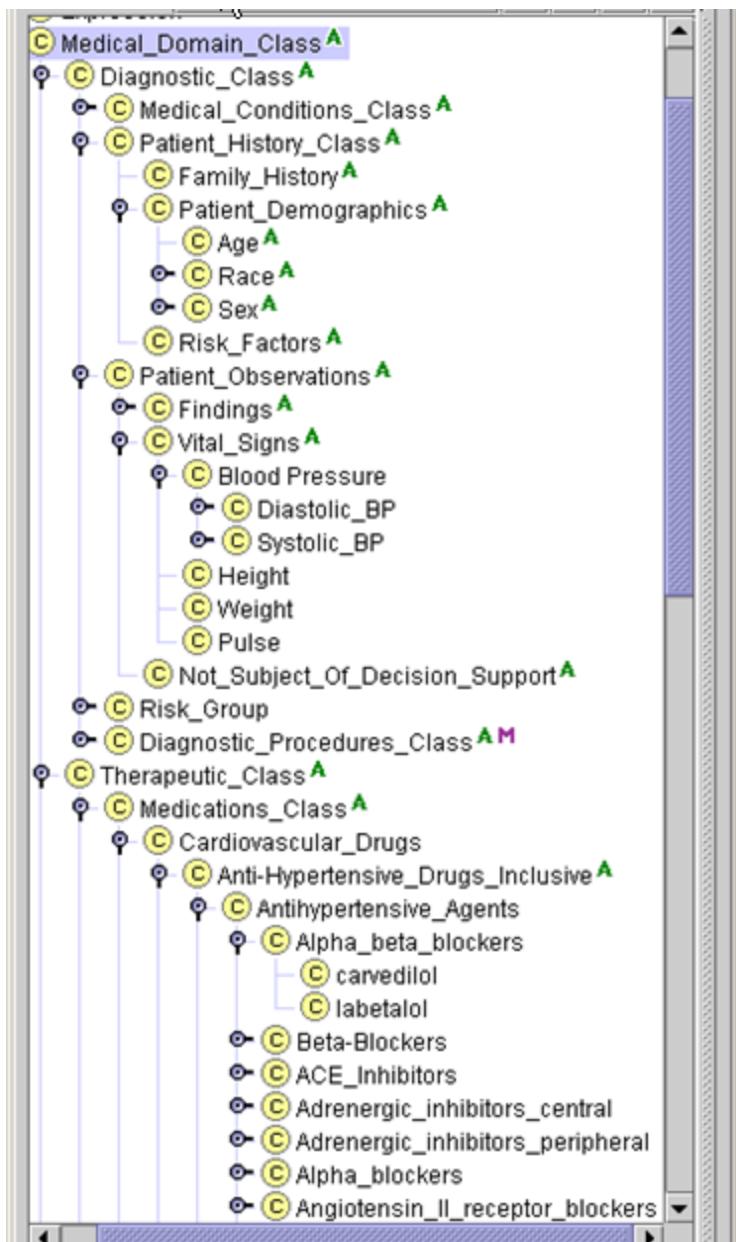


Figure 51 - Terminology classes in the Medical_Domain_Class hierarchy of the ATHENA-CDS Knowledge Base

With the exception of the Value_Type hierarchy, which is explained later in this subsection, the terminological hierarchies can be organized into any form by clinicians and knowledge engineers maintaining the ATHENA-CDS Knowledge Base, as long as a concept (represented by a class):

1. is mapped to a term in the patient data, with the mappings defined by the ATHENAeum_Mapping_Operation classes described in Subsection IV.6 (e.g., Creatinine class is mapped to the CREATININE string);¹⁴
2. has subclasses (e.g., ICD9 codes) that are mapped directly to patient data;
3. is defined as instances of Diagnostic_Term_MetaData or Derived_Parameter_MetaData; or
4. has subclasses that are mapped (as in case 1 or 2, above) or defined (as in 3).

In addition, the concept should be an instance of an appropriate metaclass (e.g., a generic drug class like *lisinopril* should be an instance of the Medications_MetaData and a finding such as *fever* should be an instance of Medical_Conditions_MetaData). The terminology hierarchy may contain additional classes (e.g., Patient_Observation) that are used to group concepts for navigational purposes; they play no role in the decision-support system.

The ATHENA-CDS Knowledge Base organizes medical conditions and drug classes into classification hierarchies as shown in Figure 50 and Figure 51. The medical condition classes either: have—at the most specific level—ICD9 codes that correspond directly to data in the VA database, or are defined using Diagnostic_Term_MetaData or Derived_Parameter_MetaData. The drug class hierarchy has, at the most specific level, generic drug names—such as labetalol in Figure 51—that are mapped to drug terms in VA prescriptions. (The drug mapping is maintained in an ATHENEON database table.) In addition, the ATHENA-CDS Knowledge Base enumerates the vitals, laboratory test results, and demographic terms used in encoding the hypertension guidelines. These terms are mapped to terms used in the VA patient database. The mapping is specified as instances of ATHENAeum_Mapping_Operation. The Converter (see Subsection II.2.2.2) transfers the specified mappings to the ATHENEON database.

IV.2.2.2. Value_Type Hierarchy

The Value_Type hierarchy defines concepts that are used in the EON system as enumerations of mutually exclusive categorical values. A complete hierarchy listing is shown in Figure 52.

¹⁴ The mappings of drug names are maintained in the SQL database without reference to mappings in the ATHENA Knowledge Base.

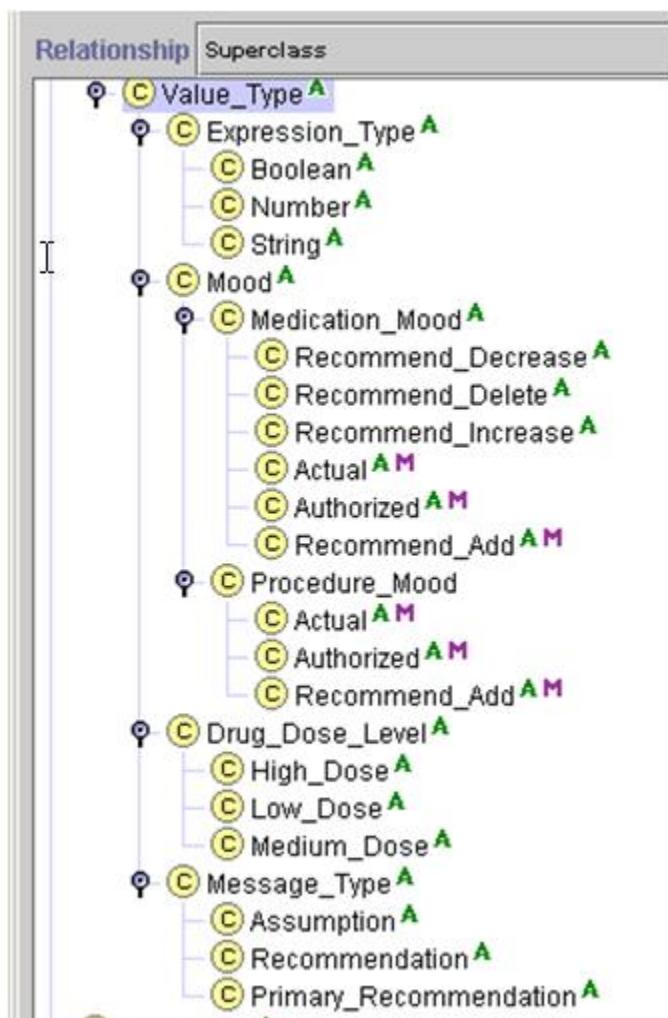


Figure 52 - The set of system-recognized terms, as organized in the Value_Type hierarchy

Medication Mood

As the Guideline Interpreter generates recommendations about drug usage, it needs to perform additional tasks based on these recommendations (or the lack thereof). For example, in order to generate patient-specific messages associated with the recommendation to add a drug,¹⁵ developers of the ATHENA-CDS Knowledge Base must encode messages that are based on the system's drug recommendations (i.e., the collateral actions associated with an instance of Drug_Usage). In effect, these recommendations must be represented as data upon which additional reasoning can be based. Accordingly, the Guideline Interpreter generates instances of the Medication class to represent its recommendations. To distinguish among system-generated

¹⁵ These patient-specific messages associated with a drug recommendation are displayed in the ATHENA GUI as the Info button messages next to a drug recommendation. See Section II.1.1.

Medication instances from patient data, the Guideline Interpreter uses Medication_Mood values (see Figure 52):

- The Authorized mood value represents the medication that is prescribed.
- The Actual mood value represents the medication that is currently being taken by the patient.
- The Recommend_Decrease, Recommend_Delete, Recommend_Increase, and Recommend_Add mood values represent possible recommendations for a drug.

Drug_Dose_Level

The possible values of Drug_Dose_Level in ATHENA-CDS are High_Dose, Low_Dose, and Medium_Dose. They are instances of the Ordered_List_Value metaclass that allows the specification of an ordered list (Figure 53). Thus, Low_Dose's *next* attribute value is Medium_Dose, whose *next* attribute is High_Dose. Similarly, the *previous* attribute of High_Dose is Medium_Dose; and the *previous* attribute of Medium_Dose is Low_Dose. These terms are used in the Guideline_Drug class to specify ordinal levels of drug doses. The actual ranges for the dose levels are defined in the dose_level_ranges slot of Guideline_Drug class. The Guideline Interpreter classifies a patient's medication dose according to the dose levels specified in instances of the Guideline_Drug class to determine whether the current daily dose is at the maximum level. The values of Drug_Dose_Level can be changed (e.g, Level_1, Level_2, etc. instead of Low_Dose, Medium_Dose and High_Dose), as long as the values are instances of Order_List_Value metaclass and the ranges are defined in instances of Guideline_Drug.

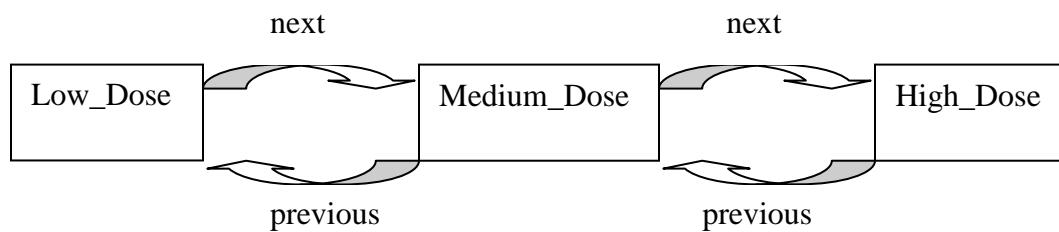


Figure 53 - Use of Ordered_List_Value metaclass to create a sequence of ordinal drug dose levels

Message_Type

Subclasses of this class define the possible values for the message_type slot of the On_Screen_Message class (see Figure 52 and Subsection IV.2.3.3). In ATHENA-CDS, the values are Assumption, Recommendation, and Primary_Recommendation. The Guideline Interpreter uses these values to annotate messages sent to the client and the current ATHENA-CDS GUI, using the message-type information to display messages in different locations. The

developer of the ATHENA-CDS Knowledge Base can change the values of these message types as long as the client program knows how to interpret the values.

IV.2.2.3. Associations

A collection of Drug_Relation classes have been defined to hold information necessary to associate supporting material with the drug-drug and drug-medical-condition relationships used to recommend changes in prescribed medications.

Figure 54 shows an example of an instance of Drug_Indication_Relation that holds supporting material for diabetes with proteinuria as a competing indication for the use of ACE inhibitors. In the ATHENA-CDS system, the GUI uses these instances to pop up HTML pages. The instances are not used by the Guideline Interpreter.

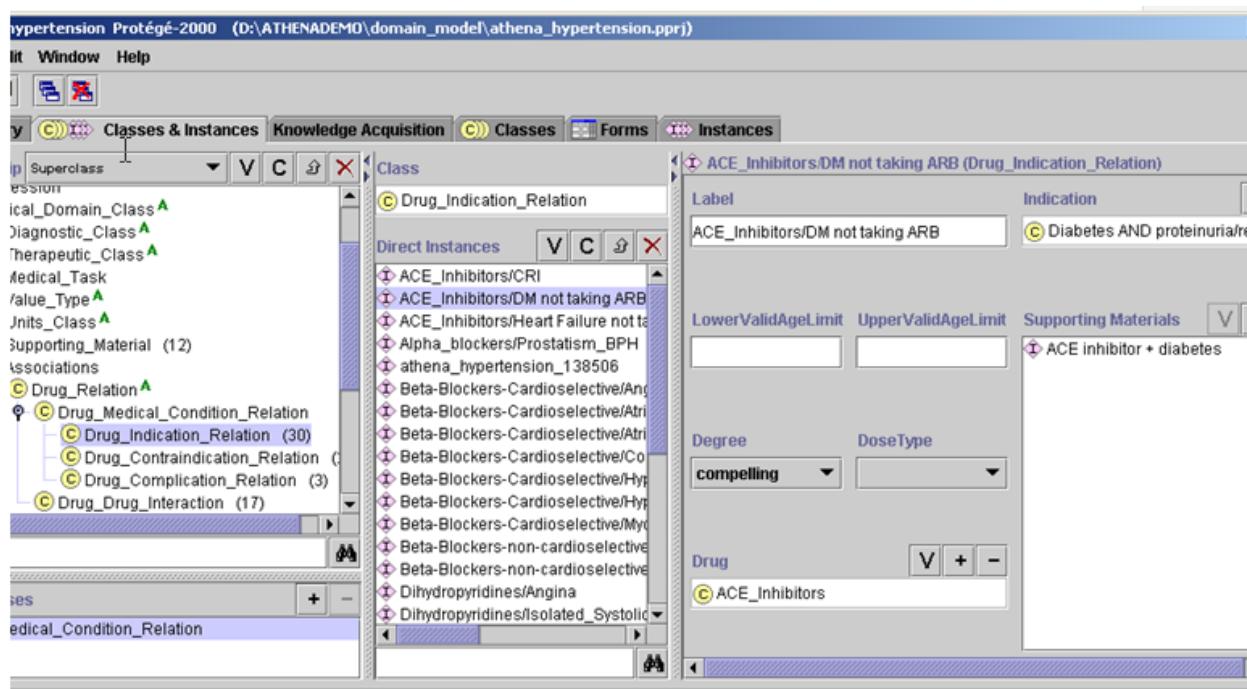


Figure 54 - An example of Drug_Indication_Relation that holds supporting material on the relationship between a medical condition and a drug class

IV.2.2.4. Supporting Material

Instances of the Supporting_Material class encode the location of reference material relevant to guidelines (Figure 55). In the ATHENA-CDS system, these instances are used by ATHENA-CDS Client to display HTML pages that support the compelling indications defined by instances

of Associations. The value of the URL slot is the relative path to the HTML page (e.g., EvidenceDisplayATHENA\Diabetes_ACE\SS_DiabetesandACE.html). The ATHENA-CDS Client concatenates the URL directory path specified in the initialization file (e.g., d:\ATHENADEMO\doc\ATHENA\). It uses values of the URL slot in instances of Supporting_Material to derive the absolute path to the HTML page (e.g., d:\ATHENADEMO\doc\ATHENA\EvidenceDisplayATHENA\Diabetes_ACE\SS_DiabetesandACE.html).

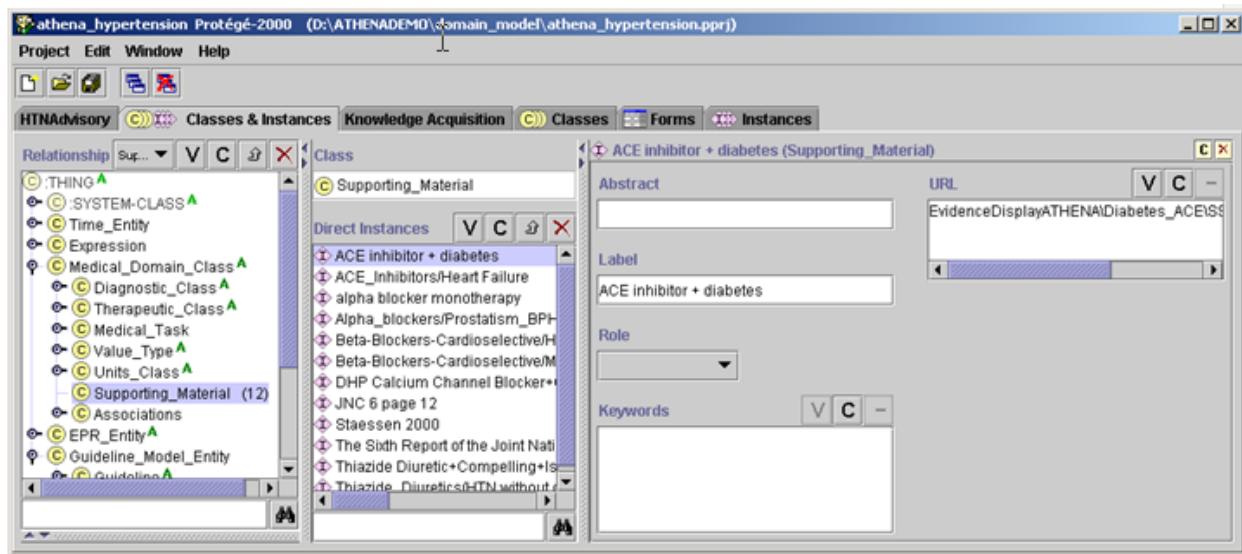


Figure 55 - Example of a Supporting_Material instance

IV.2.3. EON Guideline Model

The introduction to Subsection IV.2 describes the overall conceptualization of the EON Guideline Model. This subsection describes the details of the EON Guideline Model as it is used in the ATHENA-CDS Knowledge Base. It addresses individuals charged with maintaining the ATHENA-CDS Knowledge Base or with creating a guideline knowledge base like the ATHENA-CDS Knowledge Base.

Figure 23 shows the main components of the ATHENA-CDS Knowledge Base as represented in the instance of ATHENA_Management_Guideline class. The ATHENA_Management_Guideline class, created to facilitate navigation in the Protégé tool, is an ATHENA-specific extension of the EON Guideline Model. The ATHENA_Management_Guideline class is a subclass of the Management_Guideline class (Figure 56). The Guideline Interpreter reasons with instances of the Management_Guideline. (ATHENA_Management_Guideline being a subclass of Management_Guideline, instances of ATHENA_Management_Guideline are automatically instances of the Management_Guideline class.)

The description of the EON Guideline Model and the ATHENA-CDS Knowledge Base will be divided into four parts:

1. properties of the Management_Guideline class other than the clinical algorithm,
2. the clinical algorithm,
3. the actions (e.g., substituting drugs, raising doses, and sending messages) that the CDS SYSTEM can recommend to the end users
4. the clinical interventions (i.e., Drug_Usage and Guideline_Drug classes) whose properties (e.g., indications and dose range) are used by the Guideline Interpreter to generate recommendations.

The topics covered by these subsections represent the four areas that a developer of a knowledge base has to conceptualize the guideline knowledge in creating a computable guideline knowledge base in the EON format.

IV.2.3.1. Management Guideline

The *label* attribute is a short name for the guideline. It also serves as the identifier for the Guideline Interpreter to access the correct instance of Management_Guideline (i.e., JNC-VI Hypertension Guideline, in the case of ATHENA-CDS Knowledge Base.¹⁶

The *eligibility_criteria* attribute (Eligibility Criteria slot in Figure 28) defines the target population of a guideline encoded in EON. In applying the eligibility criteria, the Guideline Interpreter identifies a patient as *eligible* if none of the eligibility criteria explicitly rule out him or her (i.e., a criterion evaluates to *false*). Thus, the Guideline Interpreter will apply the guideline to the patient even if there is insufficient information to rule out a patient (i.e., if the criteria evaluate to *unknown*). Section IV.2.4 describes the formats of criteria.

¹⁶ Section IV.3. describes the setGuideline method in the PCASession interface, which a client program uses to specify the applicable guideline. The Guideline Interpreter uses the value of the *label* attribute to match for the string specified in the argument of the setGuideline method.

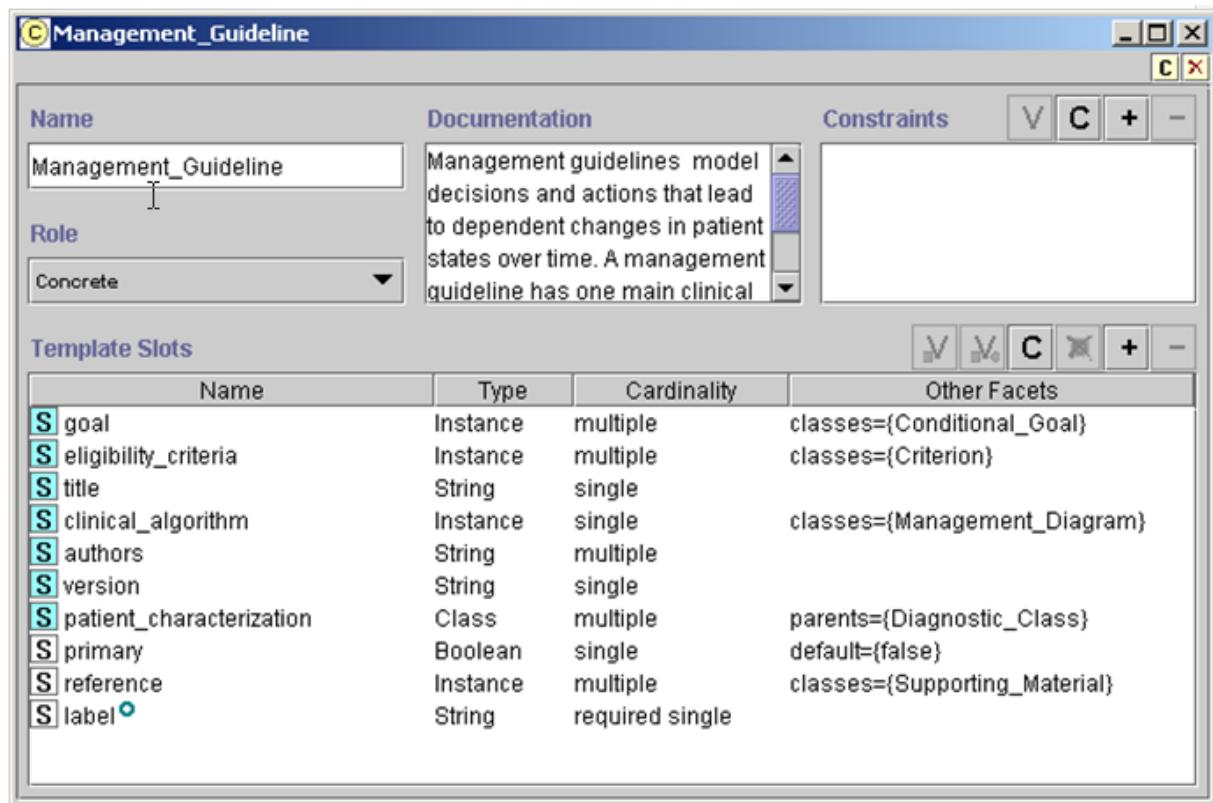


Figure 56 - The definition of the Management_Guideline class

The *goal* attribute is used to specify the goals a guideline establishes for each patient. Values of the attribute should be a list of Conditional_Goal instances. Figure 57 shows an example of a conditional goal. The selection_criterion attribute (Selection Criterion in Figure 57) is a Boolean criterion that, if evaluated to *true* for the patient, determines whether the criterion_to_achieve (Criterion To Achieve) applies to the patient.

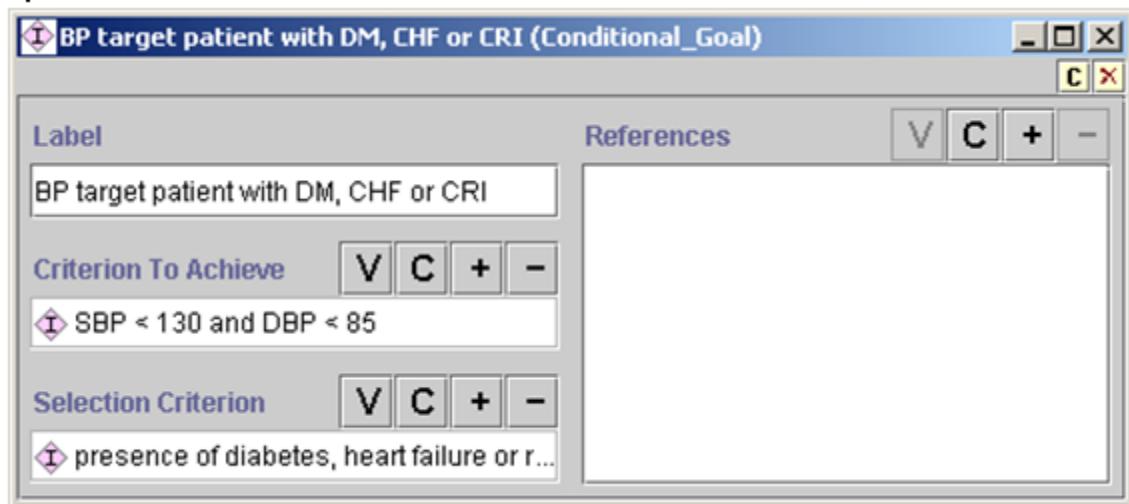


Figure 57 - Conditional_Goal example in the ATHENA-CDS Knowledge Base

Goals in the Guideline Interpreter should be mutually exclusive. That is, only one conditional goal should be applicable to a patient. The criterion to achieve can be a complex conjunction or disjunction (*and* and *or*) of other criteria. In the ATHENA-CDS Knowledge Base, the goals are target systolic and diastolic blood pressures.

The goal criteria can be used in decision-making through instances of the Goal_Criterion class. Figure 58 shows the goal criterion used in ATHENA-CDS. It is simply a reference to an instance of Management_Guideline. The Guideline Interpreter evaluates the conditional goals associated with the guideline, and returns *true*, *false*, or *unknown*. If the goal criterion evaluates to *unknown*, then the Guideline Interpreter generates alternative recommendations based on assumptions it makes about the patient's status (i.e., the assumption that the patient satisfies the goal or that the patient does not satisfy the goal).

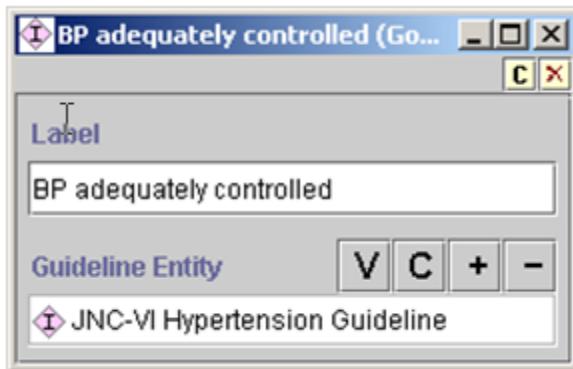


Figure 58 - Goal_Criterion used in the ATHENA-CDS Knowledge Base

The `patient_characterization` attribute is the place to specify abstractions about a patient case that the developer of the knowledge base wants the Guideline Interpreter to return as part of the recommendation. In the ATHENA-CDS Knowledge Base, the specified abstractions are risk groups used to classify patients (e.g., risk group A being no risk factor and no target organ damage or clinical cardiovascular disease). The values of this attribute are instances of the `Diagnostic_Term_Metaclass`. Currently, the ATHENA-CDS GUI does not display these patient abstractions that the Guideline Interpreter computes and returned as part of the recommendation.

The `title`, `version`, and `authors` attributes are not used by the CDS SYSTEM, but should be used to annotate the knowledge base.

IV.2.3.2. Clinical Algorithm

Individuals charged with creating a guideline knowledge base like the ATHENA-CDS Knowledge Base—the developers of the knowledge base—can specify the flow of the decisions and actions of a guideline in the `clinical_algorithm` attribute of the `Management_Guideline` class. The value of this attribute should be an instance of the `Management_Diagram` class. Protégé provides a graphical tool for viewing and editing the algorithms (Figure 59). A clinical algorithm in the EON system basically functions to organize guideline recommendations into a collection of starting scenarios (pink squares in Figure 59), such that a patient case falls into exactly one of them. Associated with each scenario is a *consultation guideline* where the developer can specify actions that should be performed for all patients classified into the scenario (actions such as showing warning messages about current medications).

Instances of the `Followed_By` class specify possible paths of an algorithm (black arrows in Figure 59). Control of flow in an algorithm is achieved by using instances of `Case_Step` (yellow diamond in Figure 59), by which exactly one path is selected (i.e., deterministic choice); or by using instances of `Choice_Step` (light yellow ovals), where more than one choice is possible (i.e., non-deterministic choices). Instances of `Action_Choice` (green ovals) follow a choice step. In action choices, the developer specifies the criteria for ruling out or for preferring the choice; and `Action_Specification` defines the recommended actions for that choice. Subclasses of `Action_Specification` fall into two categories: (1) those that perform actions directly (mostly sending messages to the user) and (2) those that do something to an “activity” (e.g., recommending adding or stopping the use of a drug, evaluating the set of drugs to add, and increasing the dose of current medications).

Figure 59 is a partial view of a simple clinical algorithm in the ATHENA-CDS Knowledge Base. It shows two possible starting scenarios. The first one (characterized by $SBP \geq 220$ or $DBP \geq 115$) leads to an `Action_Choice` step where an urgent attention message is sent. The second scenario includes cases in which the patient is not taking any antihypertensive medication, as the patient’s blood pressures are not in the range of those in the first scenario. This scenario leads to four possible management choices: adding an antihypertensive drug, recommending lifestyle changes, prescribing an ACE inhibitor, and prescribing a beta-blocker for secondary prevention. If a specific drug should be added but none has been indicated (see

branch of the *no indicated drug?* case step), the algorithm makes default choices; otherwise the algorithm continues to a scenario where messages are generated that suggest considering the addition of drugs.

The Guideline Interpreter uses a clinical algorithm by:

1. determining which starting scenario into which the patient case falls,
2. following the paths from that scenario, then
3. stopping when it reaches a terminal node (i.e., a node with no outgoing link) or when it encounters case steps or choice steps for which there is no preferred alternative.

As the Guideline Interpreter traverses the clinical algorithm, it constructs a recommendation that is eventually presented to the end user of the CDS SYSTEM.



Figure 59 - Part of a clinical algorithm in the ATHENA-CDS Knowledge Base

The components of the clinical algorithm are described in more detail, below.

Scenario

Figure 60 shows the details of the *not on drug therapy* scenario of the clinical algorithm. Because new_encounter is true (shown as New Encounter in the figure), the Guideline Interpreter will treat the scenario as a possible starting point in the clinical algorithm. The type of patient for whom a scenario is appropriate is defined by the criterion in the *precondition* slot of the scenario (shown as Precondition). At the start of its operation, the Guideline Interpreter collects all instances of Scenario class whose new_encounter slot is checked, and uses the value

of their *precondition* slot to determine which scenario to use for the patient. For example, in Figure 60, the clinical algorithm's new_encounter slot is checked, so the *not on drug therapy* scenario will be used as a starting point in generating the guideline advisory if its precondition (shown as *no drug therapy and [SBP<220 and DE..*, in Figure 60) evaluates to *true*. The followed_by slot (shown as Followed By in Figure 60) links the scenario to the next node in the clinical algorithm. The ATHENA-CDS algorithm contains five new_encounter scenarios, which are the potential starting points for a patient case: one based on blood pressure ($SBP \geq 220$ mmHg or $DBP \geq 115$ mmHg) and four based on the number of antihypertensive medications prescribed for the patient (0, 1, 2 or 3, and 4). In Figure 59, starting scenarios are the two orange rectangles: $SBP \geq 220$ or $DBP \geq 115$, and *not on drug therapy*.

In general, anyone creating a guideline knowledge base should ensure that the definitions of scenarios have preconditions that use commonly available data, and the classification of patients into scenarios should be unambiguous. Furthermore, one scenario should lead to management decisions that differ from decisions resulting from other scenarios. However, some overlap is acceptable (e.g., secondary prevention decisions are common to all ATHENA-CDS scenarios).

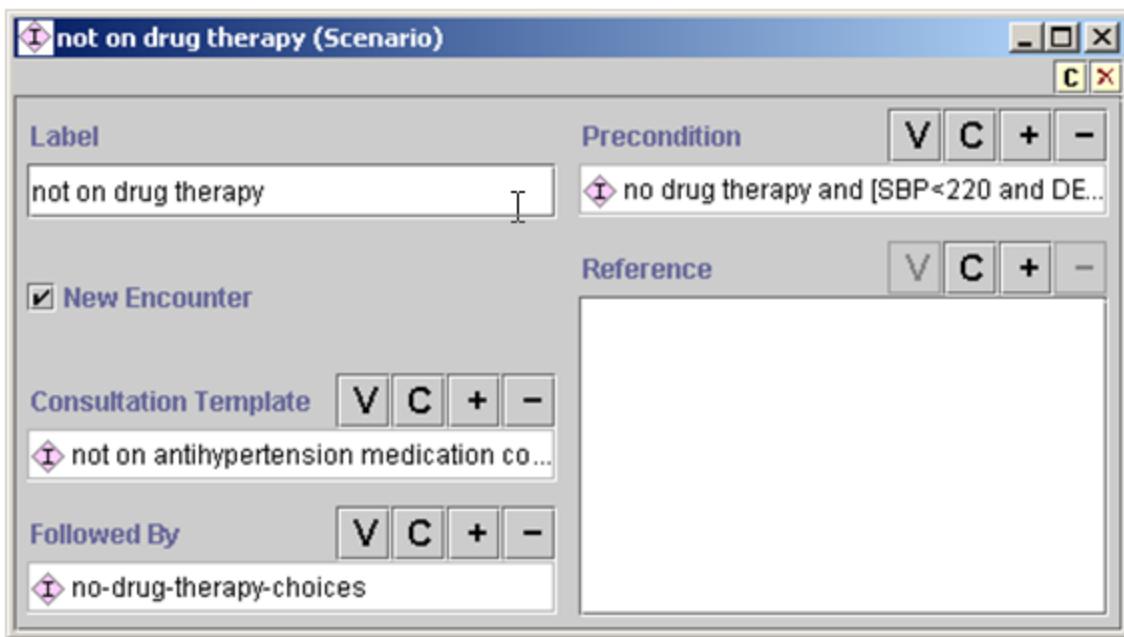


Figure 60 - Not on drug therapy scenario example in ATHENA-CDS

In addition to determining the starting point for generating a guideline advisory, the second function of a scenario is to specify actions that should be performed for patients who have been classified to meet the preconditions of the scenario (Figure 60 and Figure 61). The Consultation_Guideline has a *steps* slot where these scenario-based actions are specified. The steps of a Consultation_Guideline consist of instances of Consultation_Branch_Step (Figure 62) and Consultation_Action_Step (Figure 63).

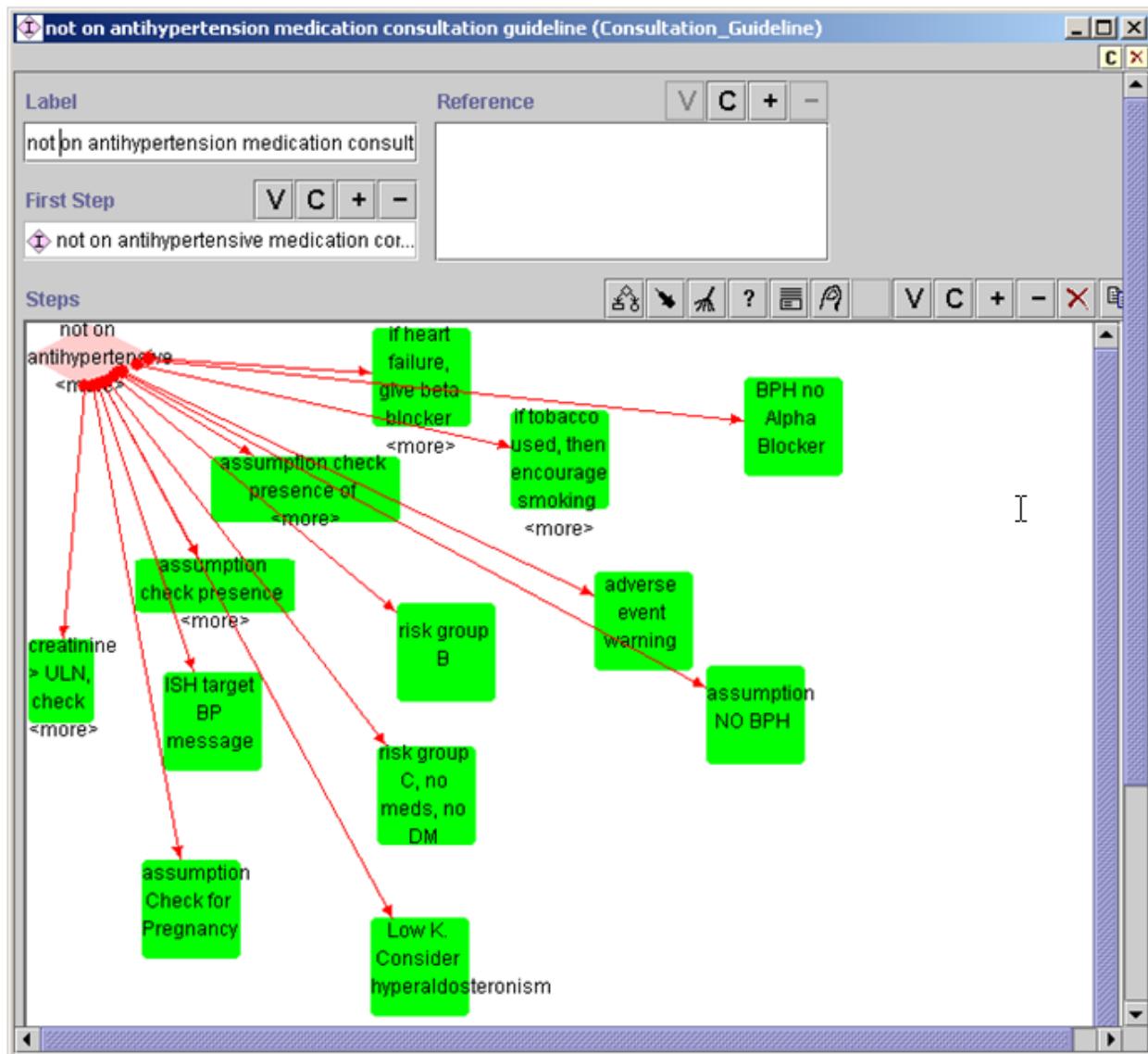


Figure 61 - Consultation_Action_Step generating messages within Consultation_Guideline. For instance, it encourages smoking cessation when the patient is a smoker.

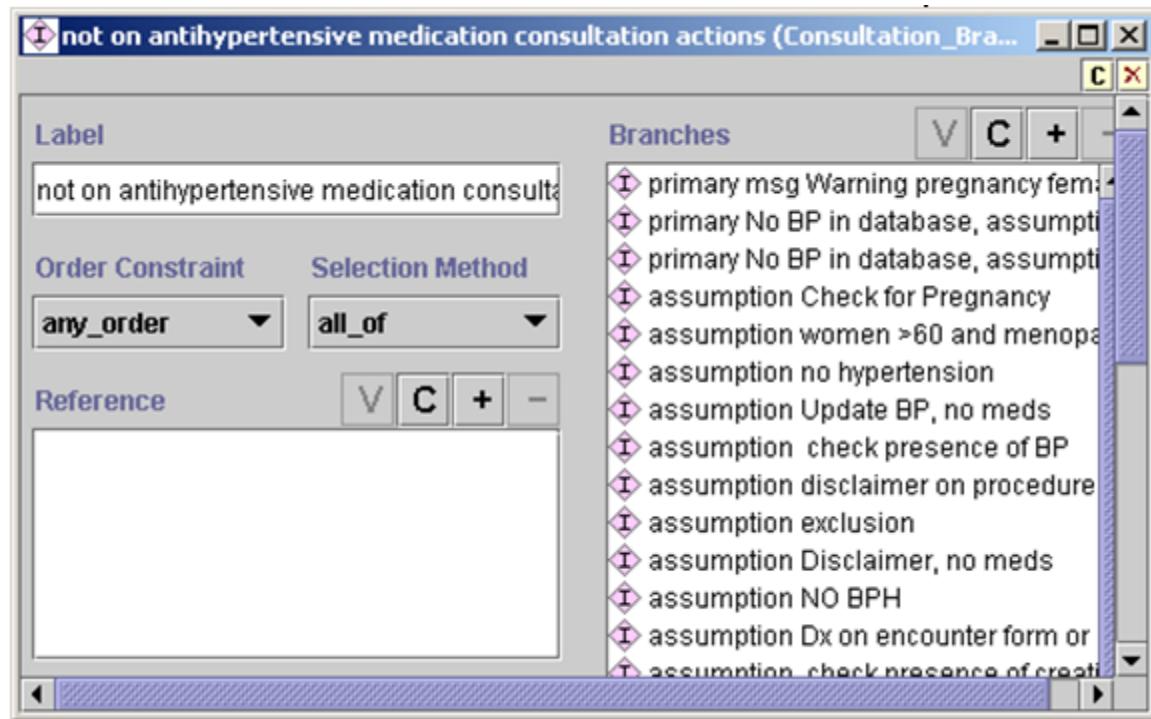


Figure 62 - Consultation_Branch_Step example

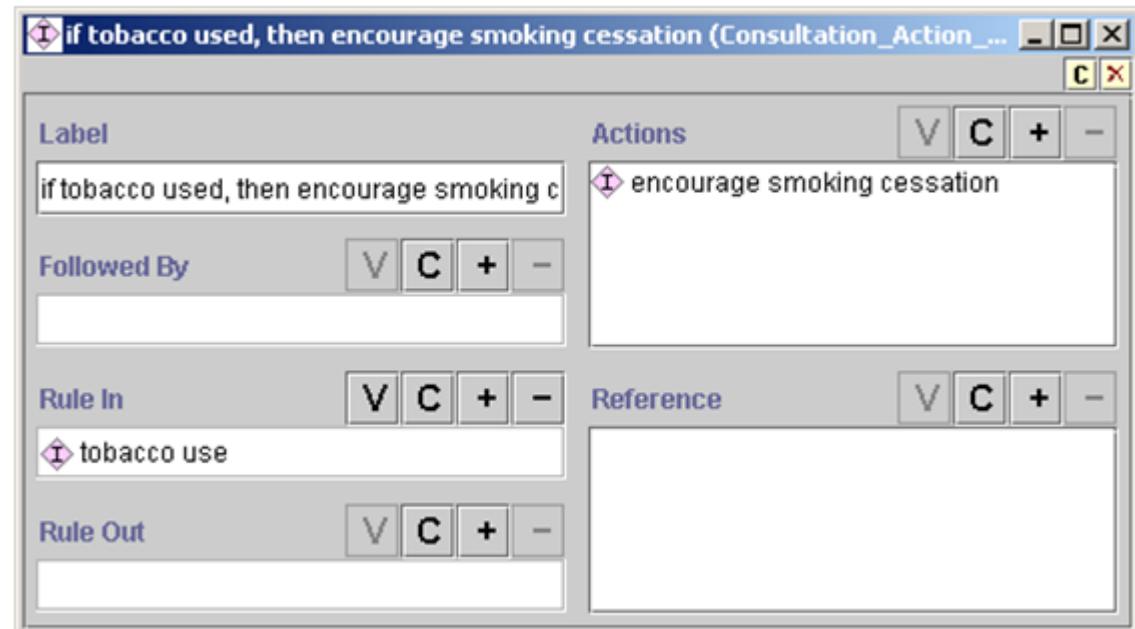


Figure 63 - Consultation_Action_Step example showing the “encourage smoking cessation” action if the “tobacco use” rule_in condition (shown as Rule In) is true

The Consultation_Guideline class was designed to allow interactive presentation of and requests for information. Each instance of Consultation_Branch_Step represents a potential decision point for determining what information to present or to request. In the current implementation of the Guideline Interpreter, however, the Consultation_Guideline allows presentation only of textual messages. An instance of Consultation_Guideline should start with an instance of Consultation_Branch_Step (specified in the first_step slot). Possible values of the *branches* slot (shown as Branches) are other instances of Consultation_Branch_Step and Consultation_Action_Step. An instance of Consultation_Action_Step has a rule-in and a rule-out condition, where an action step is marked as preferred if it is not ruled out and is ruled in. ATHENA-CDS GUI presents to the user only messages that are preferred. The *actions* slot (shown as Actions) in Consultation_Action_Step should contain instances of On_Screen_Messages or its subclasses. In the ATHENA-CDS Knowledge Base, these messages give warnings about specific conditions a patient may have and assumptions the system makes.

Case Step

A decision represents a choice from a set of competing alternatives. The EON Guideline Model supports various ways of describing alternatives, and also the corresponding selection mechanism. In the current model are two subclasses of decisions: Case_Step models decisions that are resolved by evaluating an expression and using the result to select one alternative; and Choice_Step models decisions for which one or more alternatives may be preferred, leaving the choice to the end user. Thus, ATHENA-CDS SYSTEM may recommend that, according to the guidelines, both Beta Blocker and ACE Inhibitor are good choices for a patient, leaving the actual decision for a clinician to make.

Figure 64 shows an instance of Case_Step that contains an expression (with label *no indicated drug and cannot increase dose?*). The expression, in this case, evaluates to true, false, or unknown. If the expression evaluates to true, then the system steps to the choice step to consider default drug recommendations. If the expression evaluates to false, then the system goes to a scenario where, through the use of a consultation guideline, messages are generated. If the expression evaluates to unknown, because no branch for the unknown value is specified, the Guideline Interpreter will not continue on this path.

The system determines which branch to take based on the value specified in the link between the case step (yellow diamond in Figure 64) and its branches (the red arrows). The link is an instance of the Case_Selection class (Figure 65), which has slots for the source and target of the link (first_object/First Object and second_object/Second Object, respectively) and the *value/Value* slot.¹⁷ If the expression in the first_object slot evaluates to the value in the Value slot, the object in the second_object slot should be activated.

¹⁷ The description of the attributes of the Case_Selection class applies only to the Protégé 1.7 version of the system. In Protégé 2.0 and later, the attributes of the Case_Selection class differ.

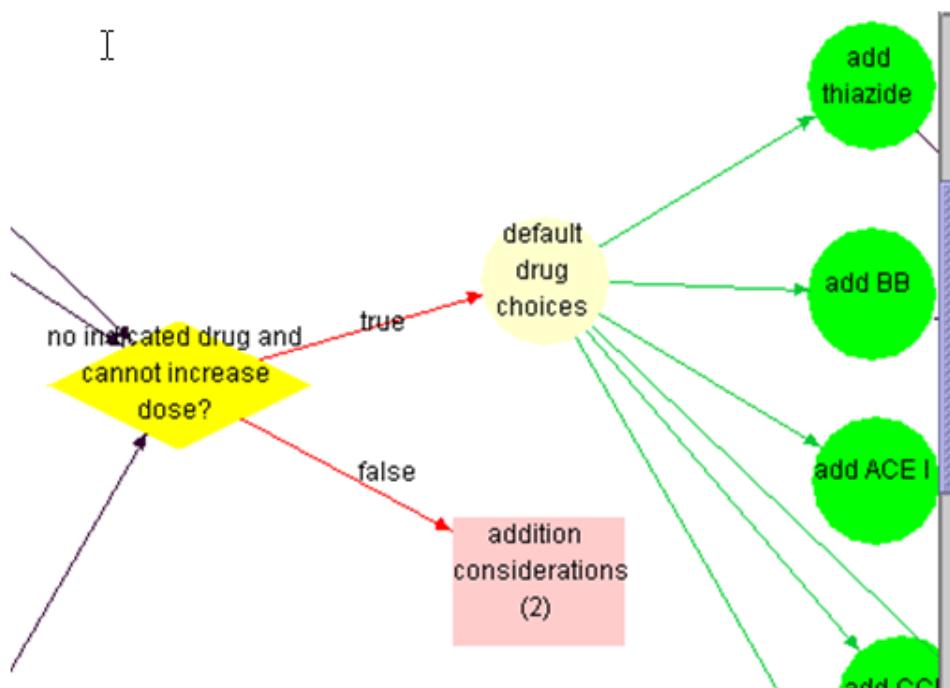


Figure 64 - Illustration of the use of Case_Step (yellow diamond). If an expression in Case Step evaluates to true (i.e., no drug has been recommended based on indications), the Guideline Interpreter evaluates the alternatives in the *default drug choices* step (yellow oval). If the expression evaluates to false, a scenario (*addition considerations (2)*) is used to generate appropriate messages. The red arrows branching from the case step (labeled *true* and *false* in the diagram) are instances of the *Case_Selection* class shown in Figure 65 - *Case_Selection* instance. It specifies that if the expression in the source (First Object) evaluates to the value in the *Value* slot, then take the step in the target (Second Object). Figure 65.

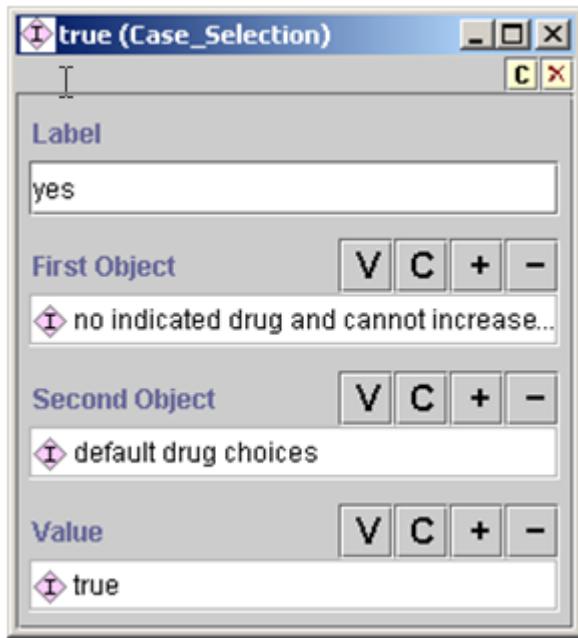


Figure 65 - Case_Selection instance. It specifies that if the expression in the source (First Object) evaluates to the value in the Value slot, then take the step in the target (Second Object).

Choice Step and Action Choice

Choice_Step and the associated Action_Choice are instances that model non-deterministic choices (i.e., more than one choice can be presented to the end user). An instance of the Choice_Step class (Figure 66) contains only links to the previous step and to the alternatives in the choice (*branches* slot, displayed as Branches). An Action_Choice instance (Figure 67) contains decision criteria (strict rule-in and strict rule-out conditions) for determining a preference for actions specified in the *actions* slot (shown as Actions). If a strict rule-out condition evaluates to true, then an alternative is rejected. If the strict rule-out condition is false or unknown and a strict rule-in condition evaluates to true, then the alternative is marked as preferred. If neither evaluates to true, then the preference for the choice can be determined by a default preference associated with the action choice. The current Guideline Interpreter is configured so that, if there is no default preference when neither strict_rule_in nor strict_rule_out conditions evaluate to true, the action choice and subsequent steps are not further pursued.

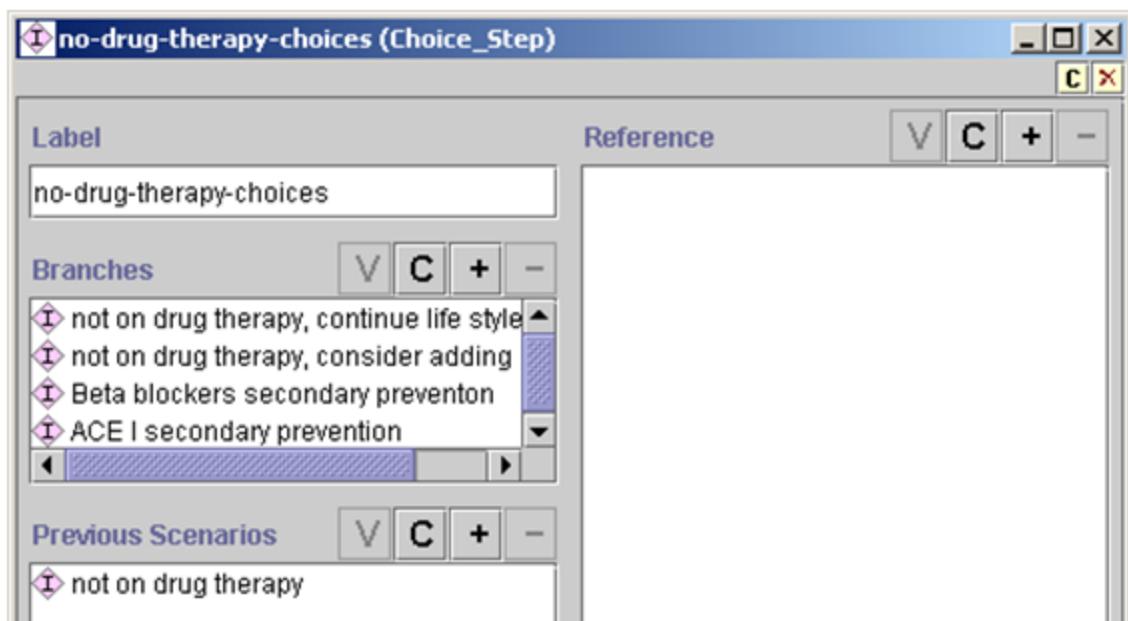


Figure 66 - An instance of Choice_Step. It contains only links to the previous step and alternatives in the choice (Branches slot).

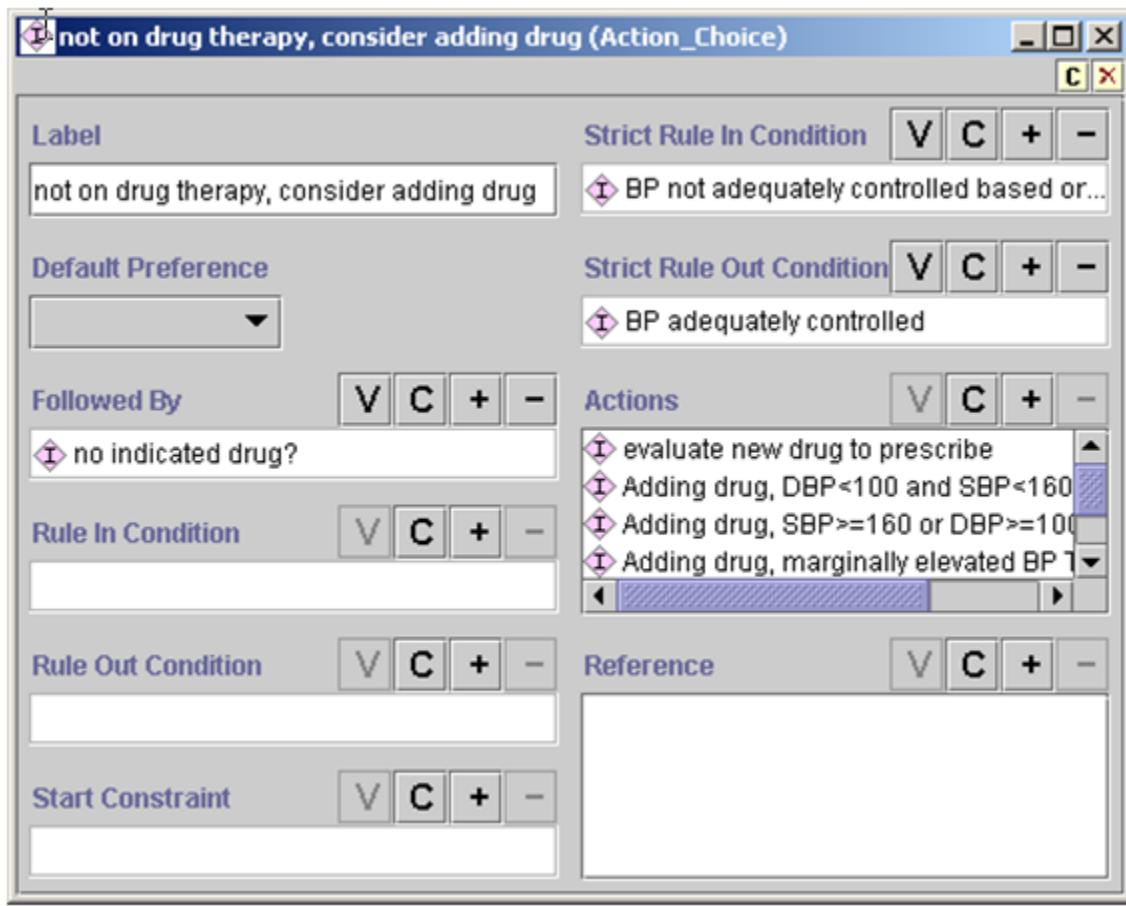


Figure 67 - Example of Action_Choice. Decision criteria (strict rule-in and strict rule-out conditions) determine preferences for actions specified in the Actions slot.

IV.2.3.3. Action Specification

Figure 68 shows the list of possible action specifications (Action_Specification) with instances that individuals building the ATHENA-CDS Knowledge Base can add to the *actions* slot of Action_Choice instances (shown as Actions in Figure 67). The action specifications can be placed into two broad categories: those generating textual output (shown as messages in the ATHENA-CDS Client) and those involving adding, deleting, or substituting activities (drug recommendations).

Messages

In the first category of Action_Specification are those action specifications in the EON Guideline Model that essentially send textual messages to the user. They include On_Screen_Message and its subclasses, and also—because of the lack of complete implementation in the current ATHENA-CDS version of the Guideline Interpreter—Procedure, Collect_Patient_Data (and its

subclasses), Present_Data, Schedule, Referral, Acute_Prescription, Modify_Activity, Step_Down_Activity, Step_Up_Activity, Stop_Activity, and Evaluate_Current_Activity classes. If instances of these classes are used (e.g., Schedule and Referral), the current Guideline Interpreter simply inserts the *label* and *description* slot values of these instances into the recommendation. In order to issue a scheduling recommendation, an instance of the Schedule class should be created. This should be done in case the Guideline Interpreter changes in the future, and also because the ATHENA-CDS GUI may be able to use the information that the scheduling recommendation is an instance of Schedule class.

Instances of On_Screen_Message class (Figure 69) contain the text of any message that will be part of a patient advisory displayed by an ATHENA-CDS Client. The class also has a message_type slot (shown as Message Type) whose possible values are specified as classes in the Value_Type hierarchy (Section IV.2.2.1). The message types are used by the ATHENA-CDS Client to determine the display location of the messages.

Instances of the Conditional_On_Screen_Message have a single rule-in condition. If the rule-in condition is true, the message is added to the advisory.



Figure 68 - Different types of Action_Specification in the EON Guideline Model

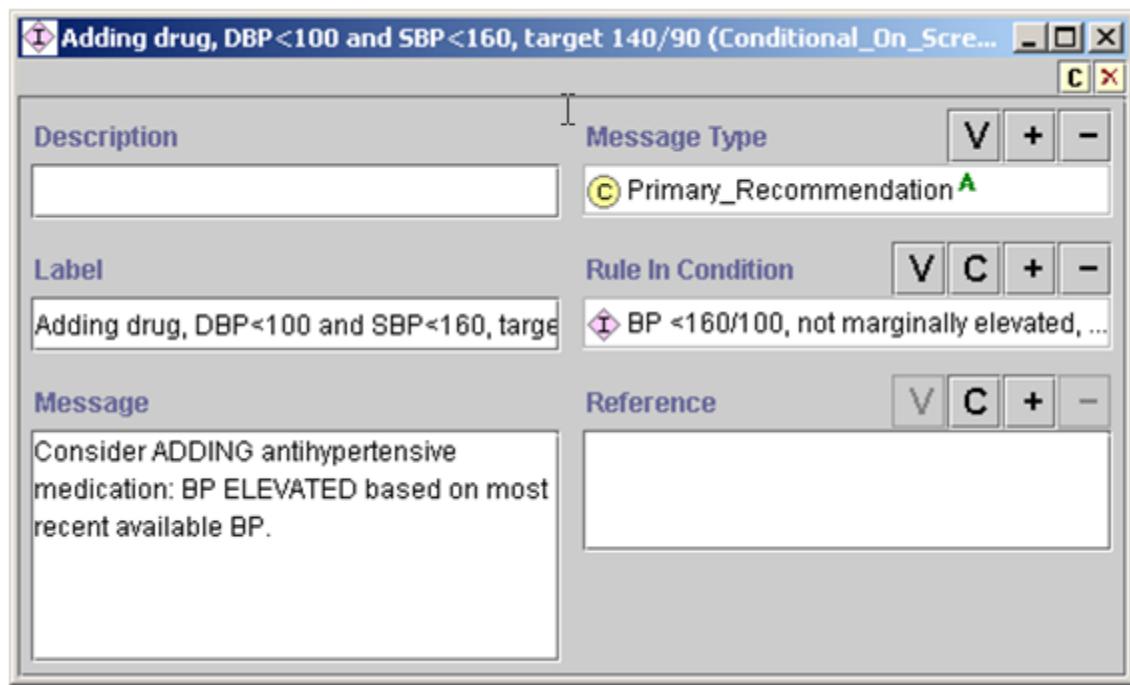


Figure 69 - Example of Conditional_On_Screen_Message

The Presence_Message class of messages responds to limitations in the previously described message classes. The class makes it possible both to check for the existence of certain conditions (e.g., allergy to a particular drug or the “active and not refilled” status of a medication) and to include the names of these conditions in dynamically generated messages (rather than the static text strings of both On_Screen_Message and Conditional_On_Screen_Message). There are two usages of Presence_Message, both of which meet the need for dynamically generated messages.

The first usage is shown in Figure 70, where the instance checks for the presence of medications for which drug_name is a subclass of Antihypertensive_Agents, and assessed_status (shown as Assessed Status) is “active and not refilled.” If there are such Medication instances, then the message “Prescription(s) for <drug_name> has(have) NOT BEEN FILLED RECENTLY” will be generated. The drug names of the queried medication instances will substitute the <drug_name> phrase.

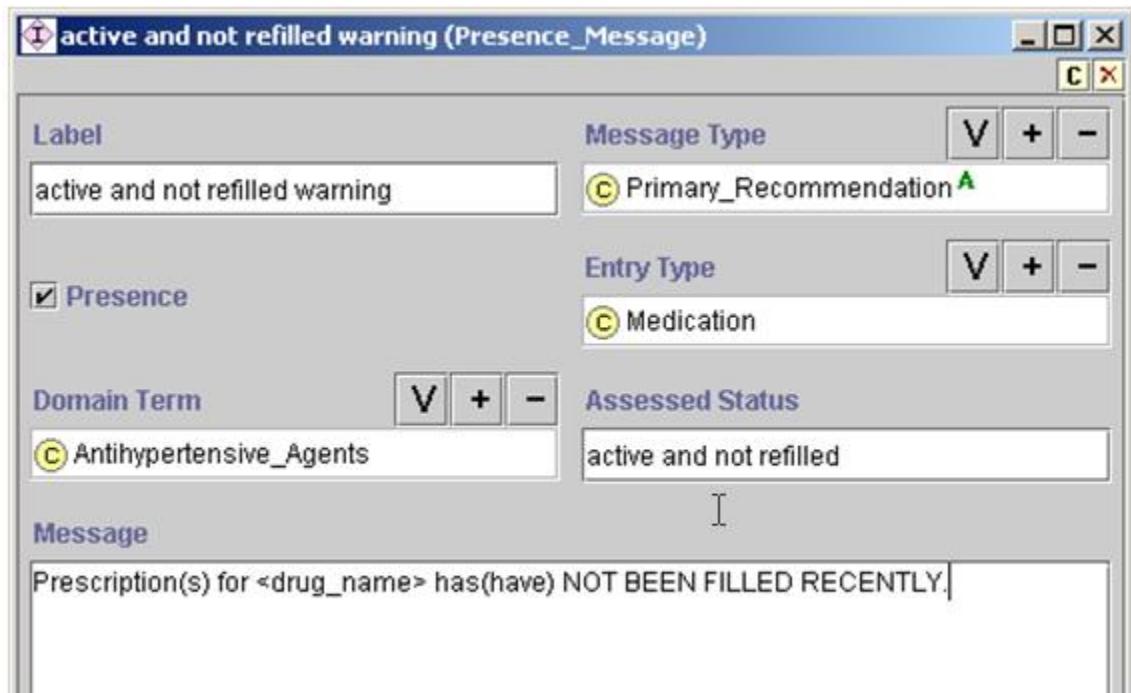


Figure 70 - An instance of Presence_Message, showing a query for medications that have the property of “active and not refilled”

The second usage of Presence_Message is shown in Figure 71. The instance specifies a query for instances of Adverse_Reaction, where the domain_term (shown as Domain Term) is a subclass of Alpha_blockers. The Guideline Interpreter will prefix the following string to the message specified in the message body: “The patient had *reaction1, reaction2...* reported as an adverse reaction/allergy to *allergic substance*.” The italicized string will be replaced with actual reactions and the allergic substance found in the patient data. For example, for the instance specified in Figure 71, the generated string might be: “The patient has dizziness and nasal congestion reported as an adverse reaction/allergy to terazosin. Use clinical judgment to determine its relevance in this patient.”

The prefixed string is generated in the Guideline Interpreter code and cannot be changed without changing and recompiling the Guideline Interpreter.

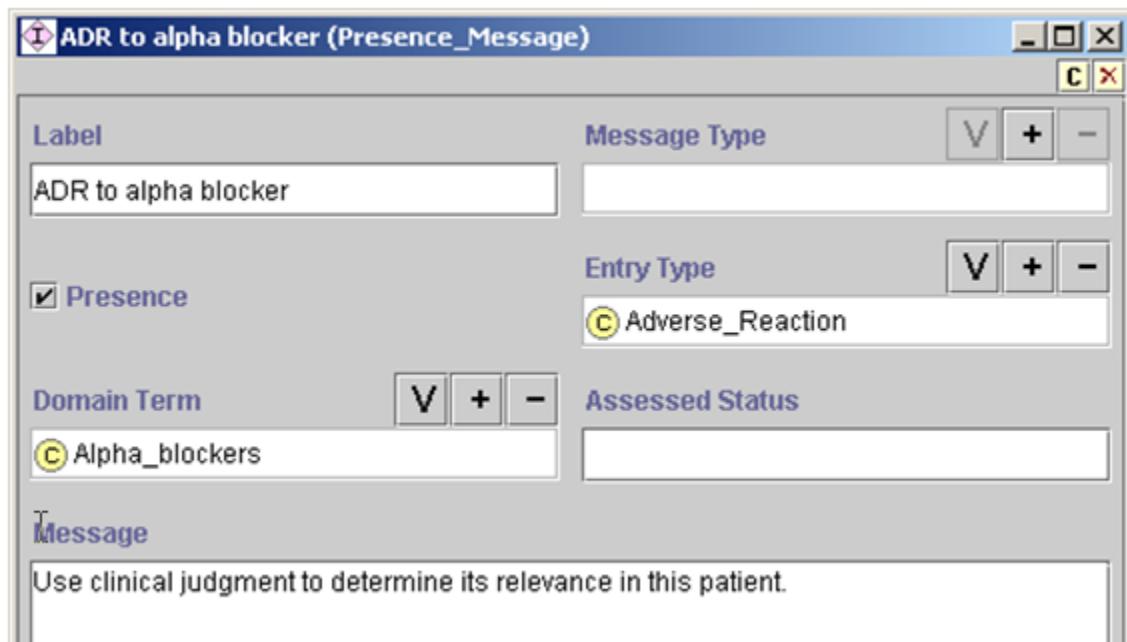


Figure 71 - An instance of Presence_Message, showing a query for an adverse reaction to alpha blockers

Drug Recommendations

In the second category of Action_Specification are recommendations to add, delete, substitute, or modify prescribed drugs (Figure 72). Subsection IV.2.3.4 describes the properties of drug classes and generic drugs that are used in generating drug-related recommendations. Below, the action specifications of this category are described.

```

Activity_Act
  Start_Activity
  Substitute_Activity
Evaluate_Activity_Act
  Evaluate_Modify_Activity
    Evaluate_Decrease_Activity_Intensity
    Evaluate_Increase_Activity_Intensity
  Evaluate_Start_Activity
  Evaluate_Stop_Activity
  Evaluate_Substitution_Activity

```

Figure 72 - List of actions that modify drug prescriptions

The difference between Activity_Act and Evaluate_Activity_Act is that, for both Start_Activity (Figure 73) and Substitute_Activity (Figure 74), the drug class or generic drug to add or substitute is always marked as preferred in the recommendation, regardless of whether the drug class or generic drug has a specific indication or not. Thus, ACE Inhibitor may have no specific indication (i.e., Diabetes and Proteinuria/Renal Manifestations or Heart Failure), but the Guideline Interpreter may nevertheless recommend it to be added in some circumstances (e.g., when using it as the default drug to add). In subclasses of Evaluate_Activity_Act, on the other hand, the Guideline Interpreter evaluates the indications, contraindications, drug partners, and complications of the selected drug classes, but it does not mark any of the evaluated drug classes as preferred. Instead, the results of evaluating these properties of selected drug classes are made available as part of an advisory.

For example, an instance of Start_Activity to start a Cardioselective Beta Blocker may, when applied to a particular patient, result in the following recommendation:

Add evaluation

```
activity_to_start: Cardioselective Beta Blocker
relative_contraindications: Obstructive_Pulmonary_Disease(492.8, 496.)
preference: preferred
messages:
    name: add beta blocker, obstructive pulmonary disease
    text: Cardiovascular benefits of beta blocker therapy may outweigh the increased
        risk of bronchospasm in this patient
    action_spec_class: On_Screen_Message
```

An Evaluate_Activity_Act to start Cardioselective Beta Blocker, when applied to the same patient case, will result in the same recommendation, except that the *preference* attribute will not have a value. (The current ATHENA-CDS GUI does not display this preference information.)

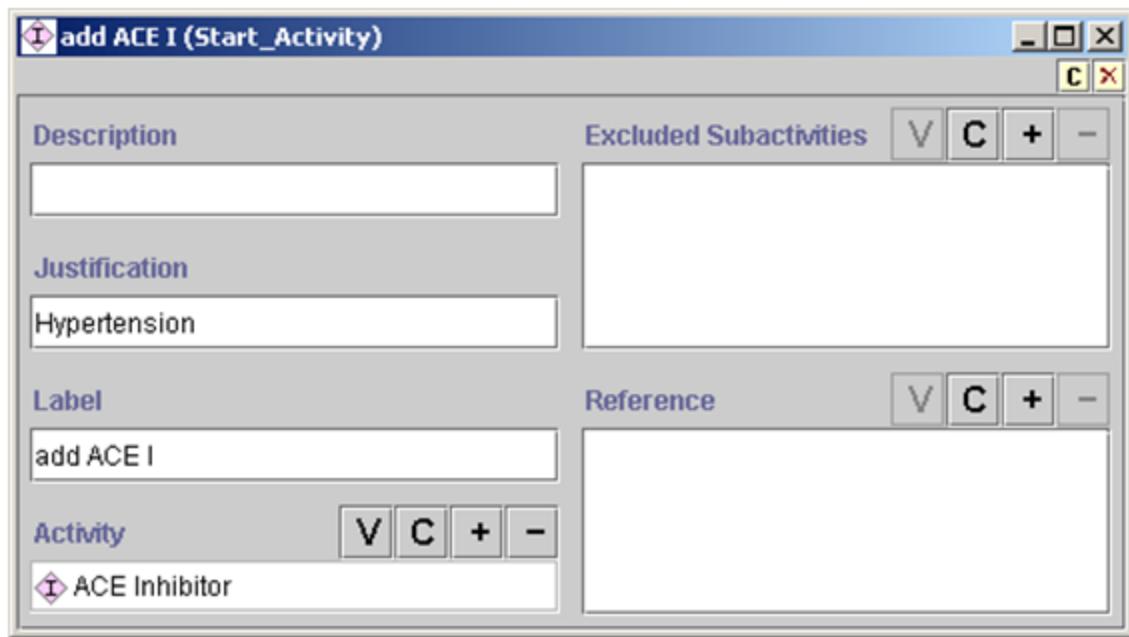


Figure 73 - An example of Start_Activity. The excluded_subactivities/Excluded Subactivities slot is not used.

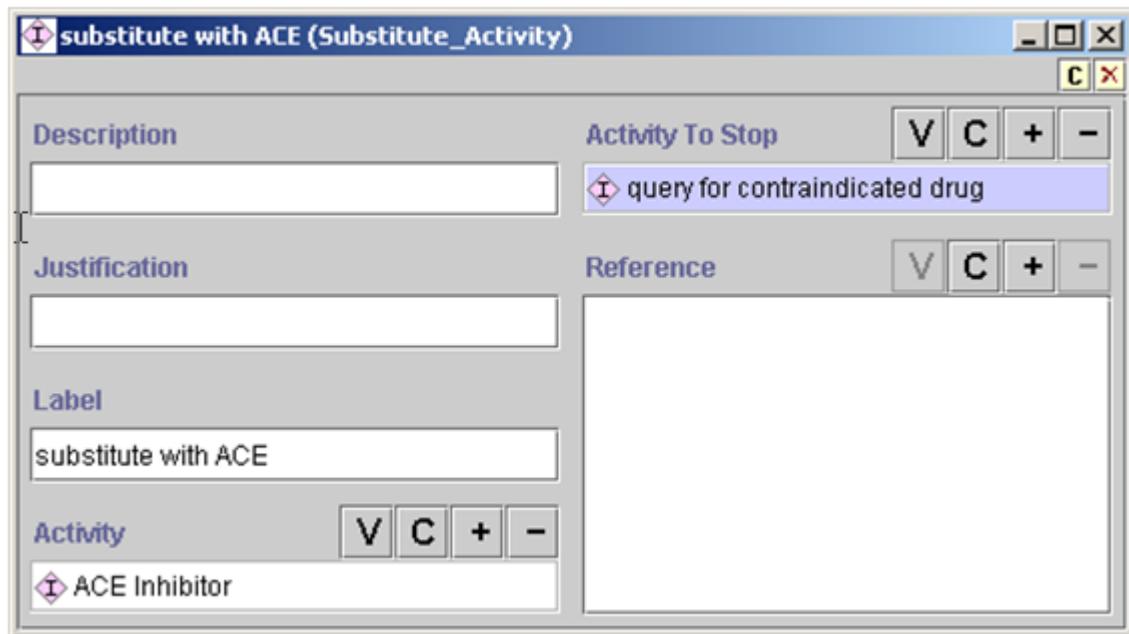


Figure 74 - An example of Substitute_Activity

In Substitute_Activity, the drug to be deleted is specified using a PAL_Query (see IV.2.4.2 for details about PAL query). Figure 75 shows an instance of PAL_Query that references a PAL query expression (not shown). The query expression, upon evaluation, returns an instance of

Medication for verapamil, if verapamil is one of the active prescriptions of the patient. Then, based on the value of key_slot (shown as Key Slot), returns the drug_name of that instance (i.e., the string *verapamil*).

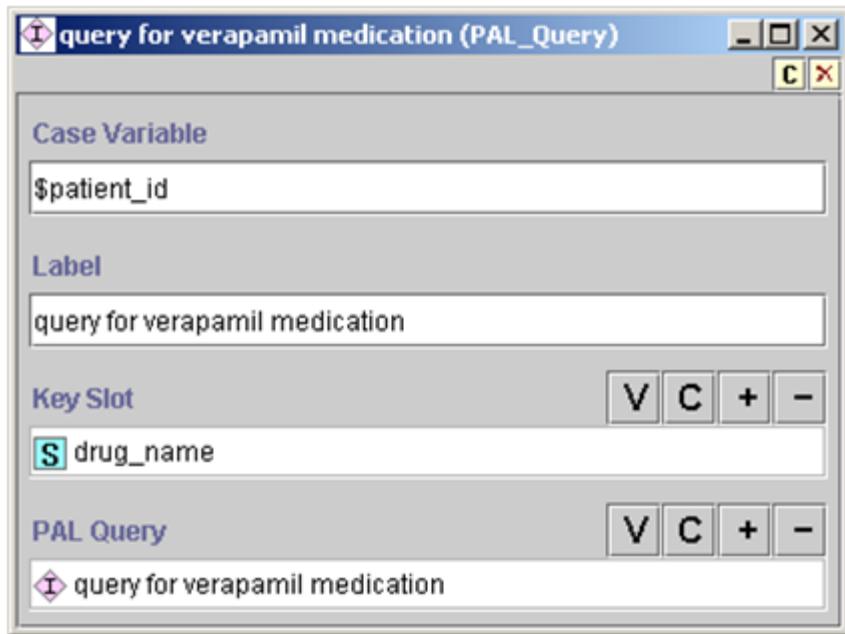


Figure 75 - PAL_Query that returns the drug_name of the query for verapamil medication

Among the subclasses of Evaluate_Activity_Act is Evaluate_Start_Activity (see Figure 76). For each instance of Drug_Usage enumerated in the alternatives slot, the Guideline Interpreter will evaluate its indications, contraindications, drug partners, drug partner to avoid, complication factor, formulary preferred drug,¹⁸ and collateral actions and make them available in the advisory.

¹⁸ A formulary is a list of prescription drugs that a health-care institution or health plan has approved for use by doctors.

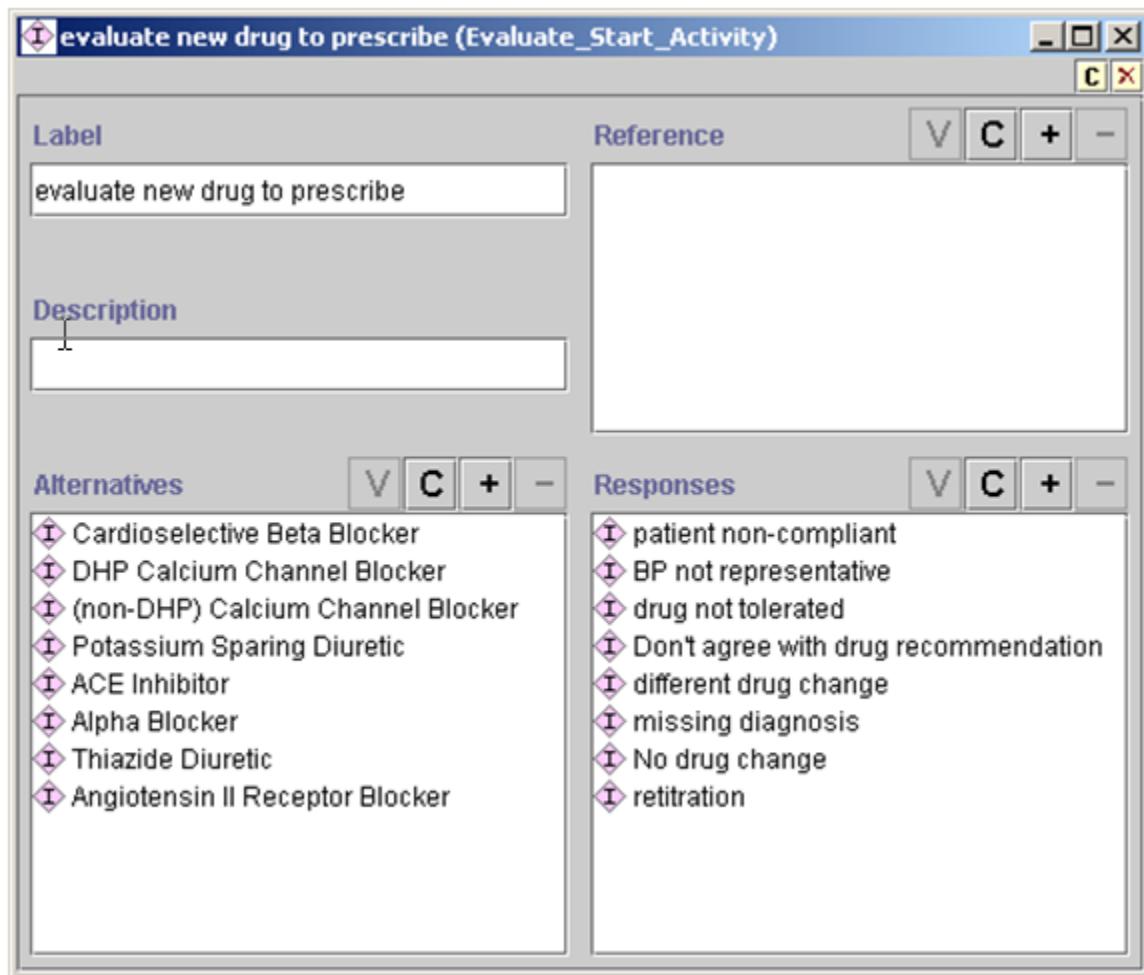


Figure 76 - Example of Evaluate_Start_Activity

Figure 77 shows an example of Evaluate_Stop_Activity, which is another subclass of Evaluate_Activity_Act. Given the example, the Guideline Interpreter will:

1. Find all instances of Medication (activity_class/Activity Class slot) whose domain term (domain_term/Domain Term) is subsumed by the Antihypertensive_Agents slot.
2. For these Medication instances, find instances of Drug_Usage (activity_spec/Activity Spec slot) whose Drug_Class_Name slot value subsumes the drug names of the Medication instance.
3. Evaluate the indications, contraindications, drug partners, drug partners to avoid, complication factor, formulary preferred drug, and collateral actions of the drug usage as part of the recommendation to discontinue this medication.

Using a PAL query is the alternative to this complicated sequence of operations.

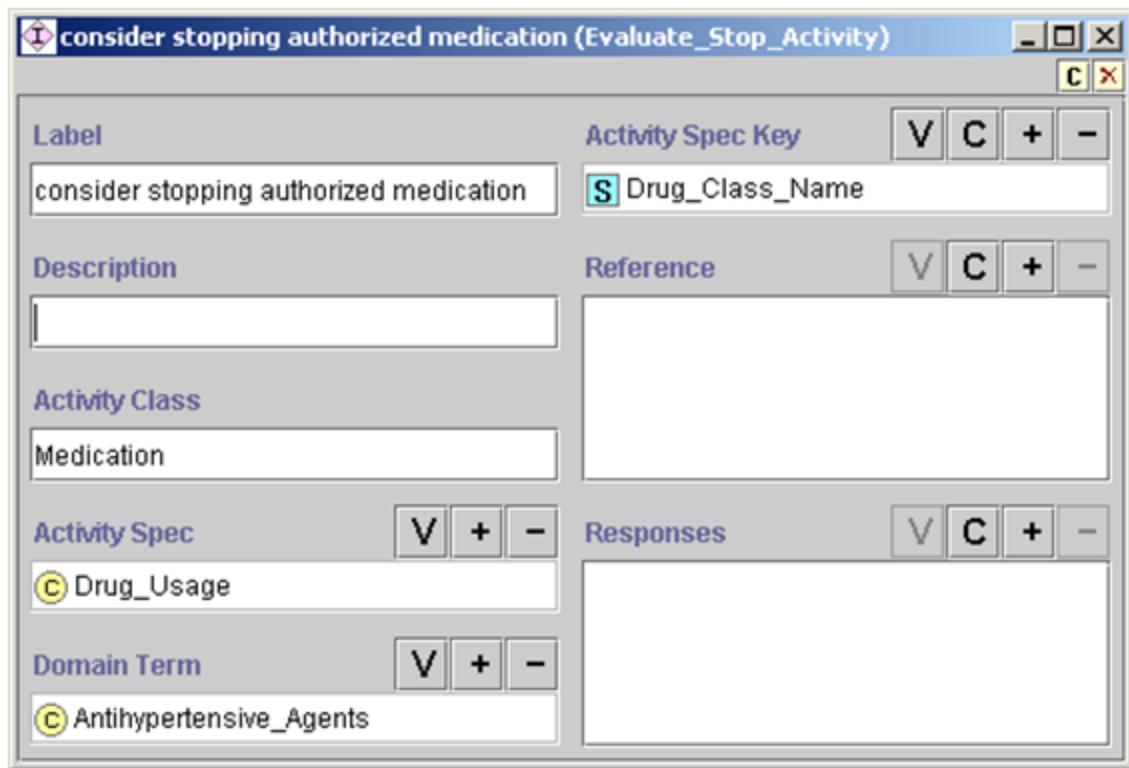


Figure 77 - An instance of Evaluate_Stop_Activity

Figure 78 shows a use of Evaluate_Substitute_Activity that does not use a PAL query to specify which drug is to be deleted in the substitution. The usage is exactly like Evaluate_Stop_Activity (Figure 77), except that—in addition to finding the Drug_Usage instance for the medication to be deleted (e.g., furosemide) and evaluating its properties—the Guideline Interpreter evaluates, for the drug class to be added (e.g., thiazide diuretics), the indications, contraindications, drug partners, drug partners to avoid, complication factor, formulary preferred drug, and collateral actions.

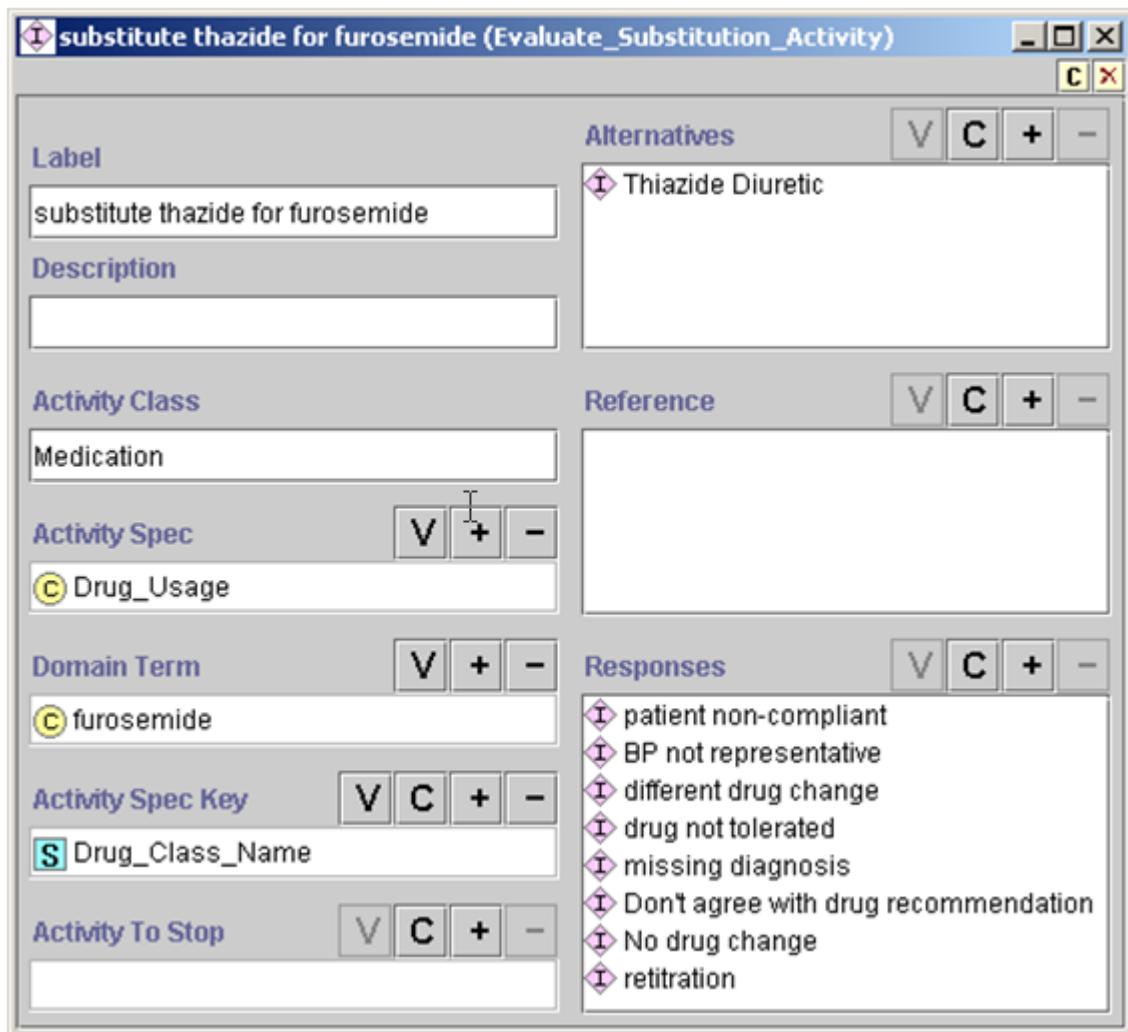


Figure 78 - Example of Evaluate_Substitution_Activity usage, without employing a PAL query

Figure 79 shows an example of using a PAL query to find the drug to delete in the substitution evaluation. The PAL query (not shown)¹⁹ finds antihypertensive medications that are to be discontinued. Then the Guideline Interpreter finds the Drug_Usage instances associated with the drugs to be discontinued²⁰ and evaluates their properties. Finally, the Guideline Interpreter evaluates the properties of the drug classes to be added in potential substitution. The Evaluate_Substitution_Activity results in recommendations to discontinue some medications and start others.

¹⁹ The PAL query should have “drug_name” in the *key_slot* so that the query will return a list of drug names to delete.

²⁰ These Drug_Usage instances are those whose Drug_Class_Name slot values (e.g., ACE Inhibitor) subsume the names of the drug to be discontinued (e.g., lisinopril).

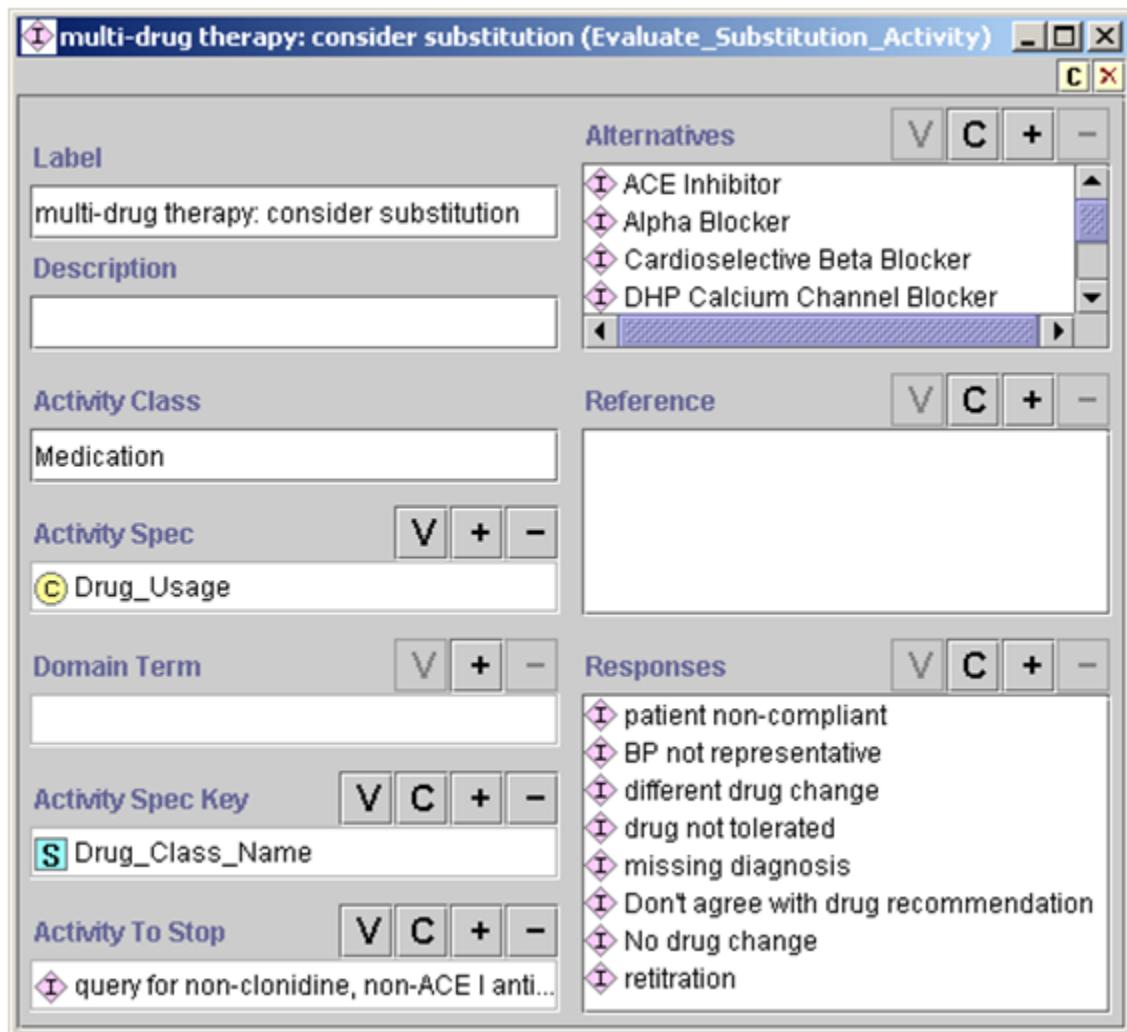


Figure 79 - Example of Evaluate_Substitution_Activity usage, employing a PAL query to determine which drugs should be substituted

Evaluate_Increase_Activity_Intensity (Figure 80) illustrates how to specify the evaluation of a drug dose for possible increase. The PAL query in the activities_to_modify/Activities To Modify slot returns the names of drugs that have a daily dose below the maximum level and do not have any contraindications or bad drug partners. The Guideline Interpreter then finds the Guideline_Drug instances (activity_spec/Activity Spec slot) whose generic_drug slot values match those of the medications. From the dose-level table in these Guideline Drug instances, the Guideline Interpreter determines the level to which the medications' daily doses should be increased. The current ATHENA-CDS GUI does not make use of this dose-level information.

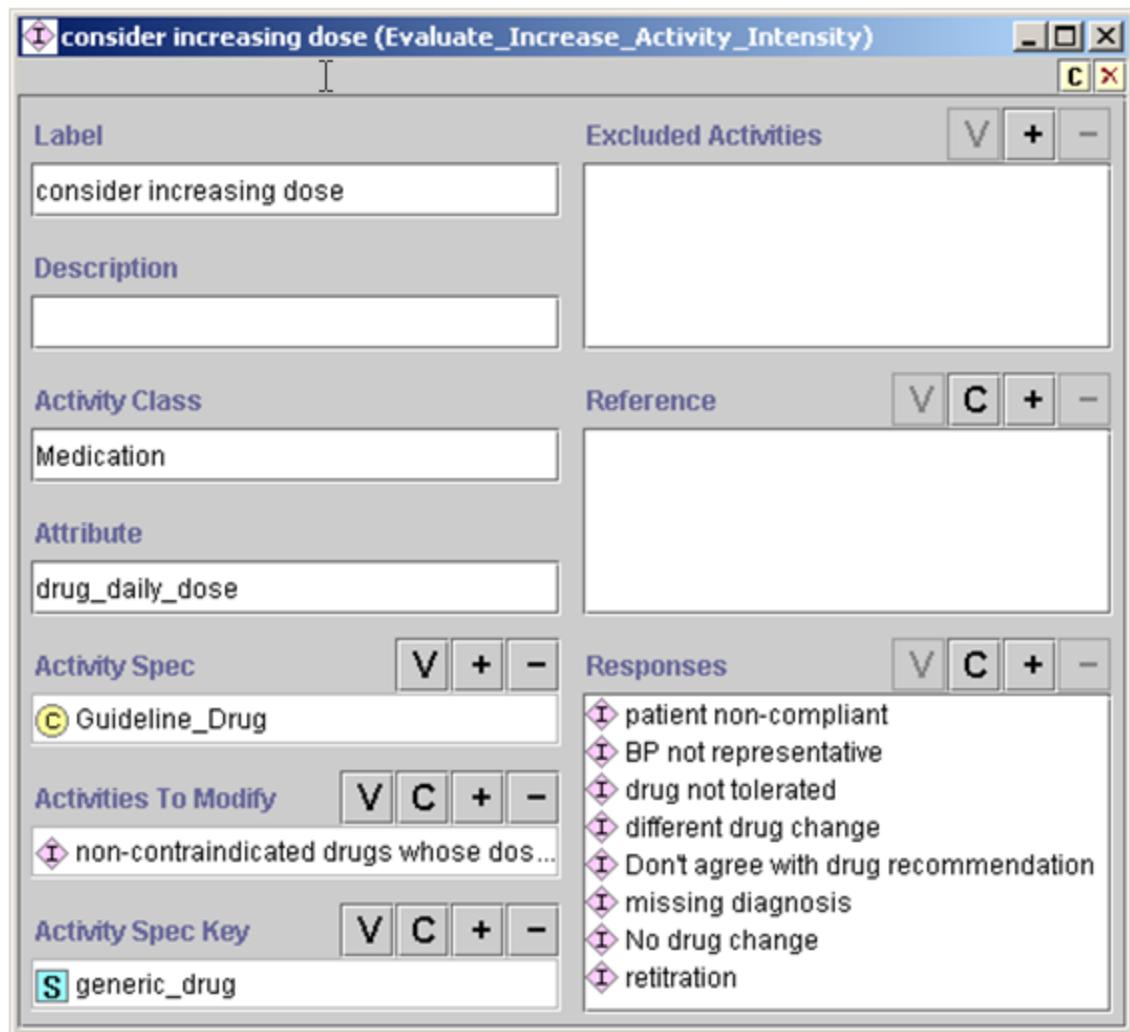


Figure 80 - Example of Evaluate_Increase_Activity_Intensity

IV.2.3.4. Drug Usage and Guideline Drug Activities

Drug Usage

The Drug_Usage class contains declarative information about properties of a drug class, such as Cardioselective Beta Blocker (Figure 81). The drug_class_name slot links this drug-usage instance to the corresponding drug class in the medical concept model's medication hierarchy. The formulary_preferred_drug_in_class slot (shown as Formulary Preferred Drug in Class in the figure) links this drug-usage instance to the generic drugs that can be prescribed for this drug class.

Cardioselective Beta Blocker (Drug_Usage)

Label	Formulary Preferred Drug Class
Cardioselective Beta Blocker	+ - C X
Drug Class Name	V + -
C Beta-Blockers-Cardioselective	
Compelling Indications	V + -
C Hypertensive_Without_Comorbidities C ISH on HCTZ and BP not controlled C MI (secondary prevention) BP controlled C MI (BP not controlled)	
Absolute Contraindications	V + -
C Bronchospastic_Disease C Heart block without pacemaker A C Unspecified Heart Block and no pacema	
Relative Indications	V + -
C Atrial_Tachycardia C Angina C Atrial_Fibrillation C Hyperthyroidism	
Relative Contraindications	V + -
C Depression C Obstructive_Pulmonary_Disease C sick sinus syndrome and no pacema C On Amiodarone	
Drug Partners	V + -
C Thiazide_Diuretics M	
Drug Partners To Avoid	V + -
C guanabenz C guanfacine C verapamil C beta-blockers-ISA C Beta-Blockers-non-cardioselective	
Side Effects	V + -
Collateral Actions	V C + -
I alpha blocker monotherapy rec ADD I Add Beta blocker and DM I Add Beta Blocker to Diltiazem (no MI) I Add Beta blocker and PVD	
Complication Factor	V + -
C Heart_Failure	
Components	V C + -

Figure 81 - Details of a Drug_Usage instance, showing the declarative properties of a drug class that can be used in the ATHENA-CDS SYSTEM

Algorithm for evaluating a drug class

In evaluating a drug class to add or delete:

1. The Guideline Interpreter determines whether the patient has any of the compelling and relative indications, absolute and relative contraindications, drug partners to avoid, or complication factors that are specified in the drug-usage instance of the drug class.
2. If the patient has at least one complication factor, the Guideline Interpreter will not generate any recommendation concerning the specified drug.
3. If the drug class is not absolutely contraindicated and there is at least one compelling or relative indication, then—unless the patient is already taking a drug of this class—the Guideline Interpreter will include the drug class among the ones it recommends adding.
4. If the recommendation includes adding a drug class, the Guideline Interpreter checks the formulary_preferred_drug_in_class slot (shown as Formulary Preferred Drug in Class) to see whether there is a more specific formulary preferred drug. These drugs are instances of Guideline_Drug class (Figure 84). If there is only one guideline drug specified in the Formulary Preferred Drug in Class slot and it is not ruled out as a result of evaluating the rule-out condition, then that drug is used as the formulary preferred drug. If there is more than one formulary preferred drug that is not ruled out, then the system evaluates the rule-in criteria. If at least one guideline drug is ruled in, then the ruled-in drugs are considered to be preferred. If no drug is ruled in, then the system returns all drugs that are not ruled out as possibilities that a clinician can select.
5. After determining whether a drug should be added or deleted or have its dose increased, the system—as part of the evaluation of a drug class—assesses the Collateral_Action instances in the collateral_actions slot of a Drug_Usage.

Figure 82 shows an instance of Collateral_Action class. It contains a mood/Mood slot that specifies the context (Recommend_Add, Recommend_Delete, or Recommend_Increase_Dose) where the actions specified in the actions/Actions slot are applicable. The *actions* slot should contain instances of Action_Specification that generate messages (e.g., Figure 83). In ATHENA-CDS, these collateral action messages are displayed as drug-specific info-button messages (Subsection II.1.1).

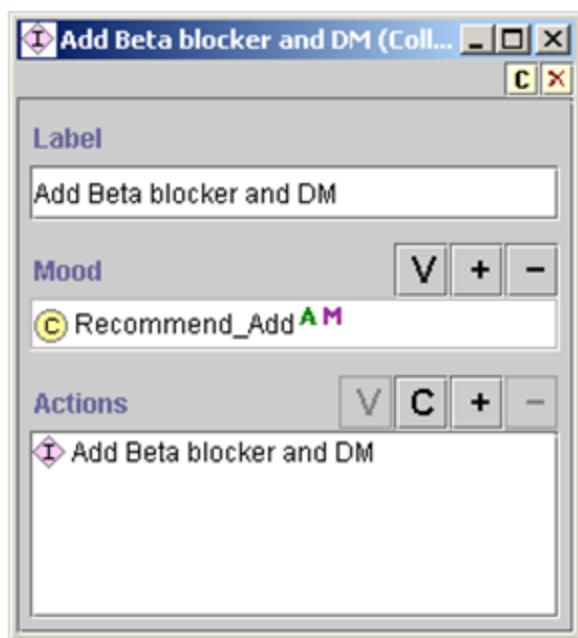


Figure 82 - An example of Collateral_Action associated with a drug class

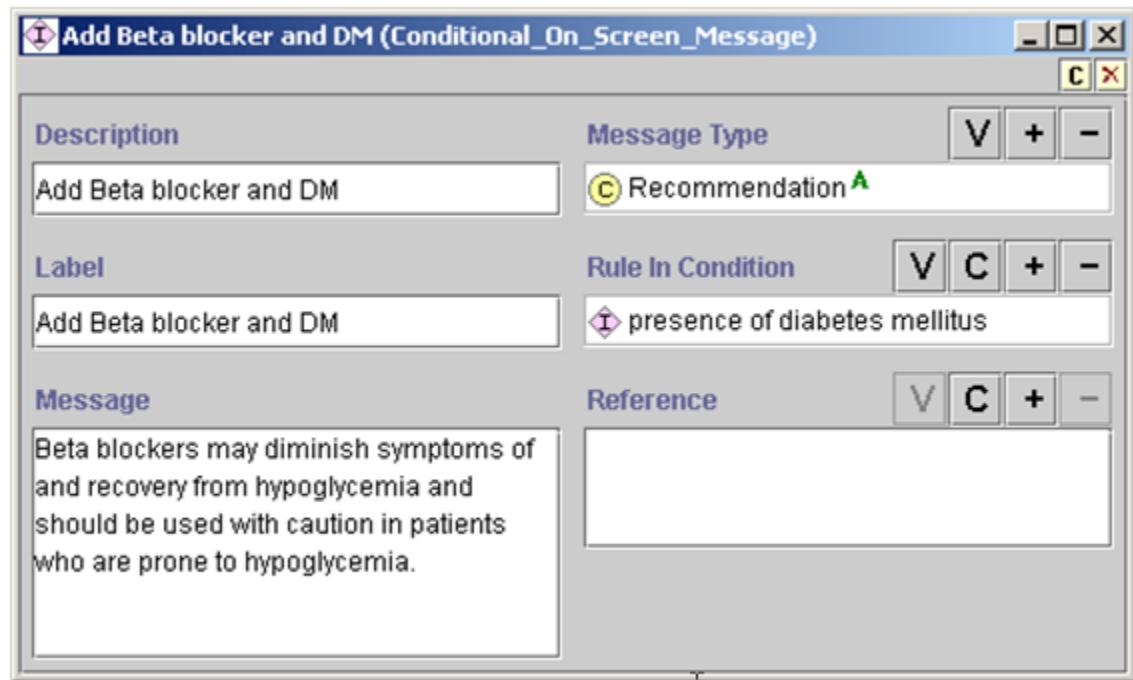


Figure 83 - An example of Message associated with a drug recommendation

Guideline Drug Activities

As described earlier in this subsection, instances of the Guideline_Drug class make it possible to specify preferred drugs in the formulary. The second role of Guideline_Drug is to encode dose information, such that the system can determine when to recommend increasing the dose of a drug. Figure 84 shows an instance of the Guideline_Drug class. It contains: a reference to the generic drug metoprolol in the medication hierarchy, a table of dose-level ranges (dose_level_ranges/Dose Level Ranges slot), and the specification of the maximum recommended dose level (max_recommended_dose_level/Max Recommended Dose Level slot). The dose-level range table is a set of Range_Mapping_Entry instances that specifies the lower and upper dose limits for a nominal dose level (Figure 85). The system would recommend increasing the dose of a drug only if the current daily dose is in a range below the maximum recommended dose level.

Currently, the Guideline Interpreter ignores the starting_dose, dose_strength_unit, drug_usage, duration_constraint, prescribable_items, and collateral_actions slots of the Guideline_Drug class.

metoprolol (Guideline_Drug)

Label	Collateral Actions
metoprolol	<input type="button" value="V"/> <input type="button" value="C"/> <input type="button" value="+"/> <input type="button" value="-"/>
Generic Drug	<input type="button" value="V"/> <input type="button" value="C"/> <input type="button" value="+"/> <input type="button" value="-"/>
(C) metoprolol	
Starting Dose	Dose Level Ranges
50.0	<input type="button" value="C"/> <input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="X"/> <input type="button" value="New"/>
Dose Strength Unit	abstract_val... lower_limit upper_limit
(C) mg A	Low_Dose 0.0 50.0
	Medium_Dose 51.0 200.0
	High_Dose 201.0 300.0
Max Recommended Dose Level	<input type="button" value="V"/> <input type="button" value="C"/> <input type="button" value="+"/> <input type="button" value="-"/>
(C) High_Dose A	
Rule In	Prescribable Items
(I) presence of MI	<input type="button" value="V"/> <input type="button" value="C"/> <input type="button" value="+"/> <input type="button" value="-"/>
Rule Out	
Drug Usage	Reference
	<input type="button" value="V"/> <input type="button" value="C"/> <input type="button" value="+"/> <input type="button" value="-"/>
Duration Constraint	

Figure 84 - An instance of Guideline_Drug class

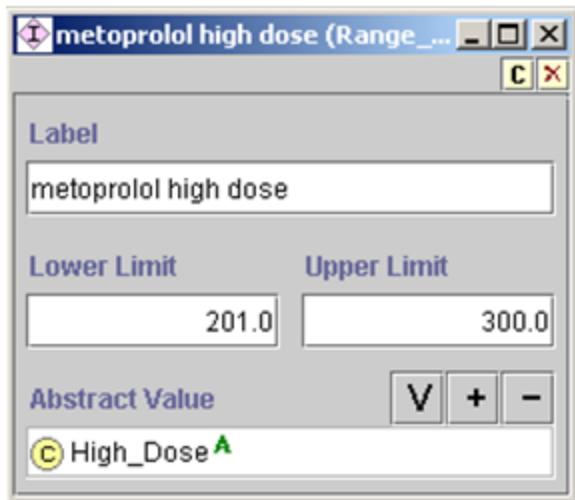


Figure 85 - An instance of Range_Mapping_Entry, showing the lower and upper limits of “high dose” for metoprolol.

IV.2.4. Expressions

The primary mechanism through which the EON Guideline Model interacts with the patient data model and medical concept model involves decision criteria and other expressions written in the EON’s expression languages. These decision criteria and expressions are what make encoded guideline knowledge bases, like that of ATHENA-CDS, “computable” (i.e., they can be used to automatically match patient data to conditions in a guideline).

Guideline authors and developers can write expressions and decision criteria in the EON Guideline Model in one of two languages.²¹ Expressions and decision criteria in these two languages are represented as Protégé classes and instances (Figure 86).

²¹There is a third language, but it has never been used in the ATHENA and is mostly untested. In this language, instances of the Temporal_Query class hold ChronusII queries and expressions. The Guideline Interpreter simply passes the queries and expressions to ChronusII for evaluation.

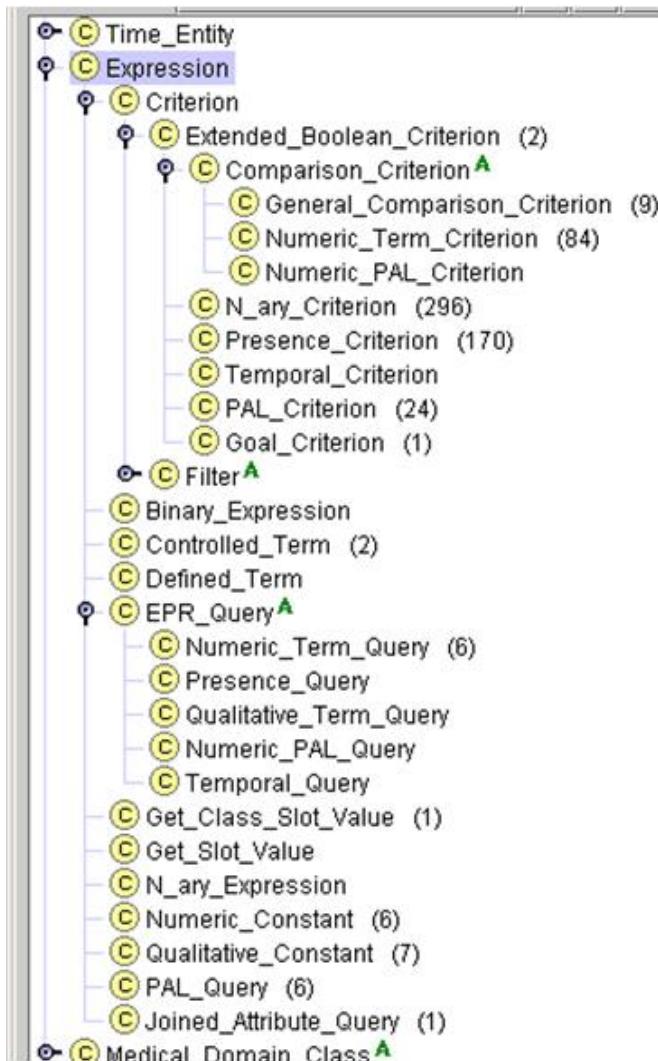


Figure 86 - Classes that make up expression languages in EON

IV.2.4.1. Template-based Expression Language

The first expression language consists of a set of object templates, allowing a guideline author to encode common but relatively simple criteria by filling in forms generated by Protégé. Expressions in this language are instances of those subclasses of the Expression class that are not Numeric_PAL_Criterion, PAL_Criterion, Temporal_Criterion, Temporal_Query, and PAL_Query.

Consider the example: “(Presence of diabetes mellitus) and (Minimum creatinine within last two months is greater than upper limit of normal)”. This example can be encoded using a nested series of object templates described below.

Criteria encoded in this template-based language evaluate to *true*, *false*, or *unknown*. For example, in the absence of any serum creatinine result, the criterion that compares the serum creatinine value to its upper limit may evaluate to *unknown*. However, if a patient does not have a medical problem, this fact is usually not explicitly recorded. When checking for the presence or absence of medical problems specified in the medical concept model, the Guideline Interpreter assumes that, if a problem is not explicitly recorded, the patient does not have that problem. Accordingly, these criteria evaluate to *false* when there is no record of a medical condition specified in these criteria.

In most of the EON Guideline Model, a criterion makes a difference when it evaluates to *true*. For example, strict rule-out and strict rule-in conditions apply only when the conditions evaluate to *true*. Thus, in most cases, the *unknown* truth value²² has the same effect as the *false* truth value. In three places, however, *unknown* makes a difference:

1. In checking eligibility for guidelines, *unknown* has the same effect as *true*. In other words, a patient is not eligible for a guideline only if at least one eligibility criterion evaluates to *false* (see Section IV.2.3.1).
2. If a goal criterion evaluates to *unknown*, then the Guideline Interpreter generates alternative recommendations based on assumptions that the goal is first satisfied and then unsatisfied (see Section IV.2.3.1).
3. In a Case_Step, flow of control is based on the value of the expression specified in the step. If the value of the expression is *unknown*, and if there is a branch that matches the *unknown* value, then that branch will be taken. Otherwise, the Guideline Interpreter will stop following the path.

Below, selected classes in the template-based language are described in detail. The construction and usage of the classes other than those described are similar.

Presence Criterion

Consider: (*Presence of diabetes mellitus*). This expression can be encoded using an instance of the Presence_Criterion class Figure 87. Presence_Criterion allows the specification of the presence or absence of patient data model instances (e.g., Note_Entry instances) that correspond to domain terms from the medical concept model (e.g., diabetes mellitus).

The *period* (shown as Period) attribute specifies how far back to look for the record. In the absence of a value for the *period* attribute, the Guideline Interpreter looks at all past data available to it. The *mood* (shown as Mood) attribute is used only with Medication records, and is explained in Subsection IV.2.2.2.

²² In logic, a truth value is a value indicating to what extent a statement is true. In classical logic, a proposition can only be *true* or *false*. This guide is departing from the requirements of classical logic.

The evaluation of a Presence_Criterion instance returns *true* or *false*. For a criterion with the *presence* (shown as Presence) flag checked, the criterion is true if and only if there exists an entry of the *entry_type* (Entry Type) that satisfies the *mood* and *period* constraints specified in the Presence_Criterion instance.

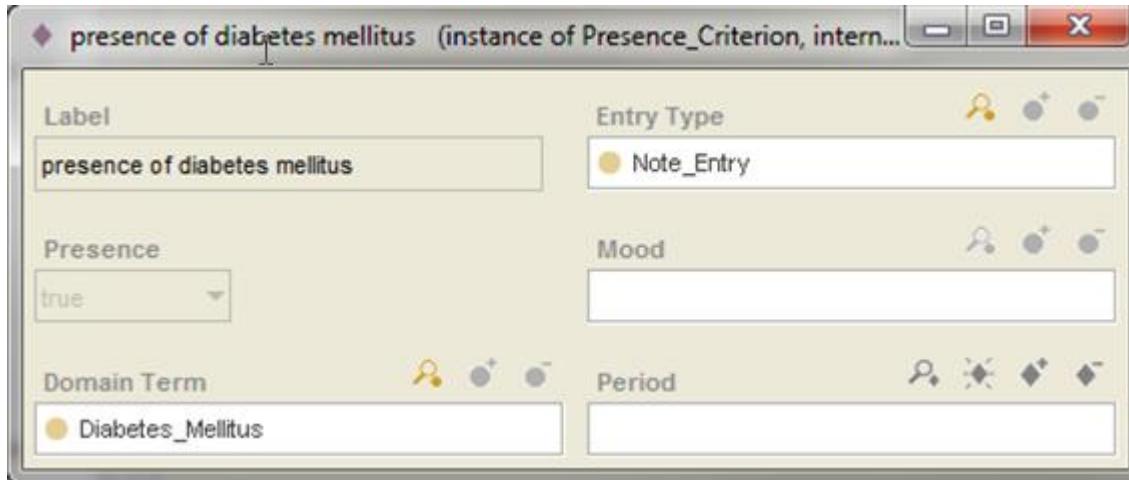


Figure 87 - An instance of Presence_Criterion that is checking whether there has ever been a note entry of diabetes mellitus diagnosis

General Comparison Criterion and Numeric Comparison Criterion

Next consider the condition: *Minimum serum creatinine within last two months is greater than upper limit of normal*. An individual maintaining the ATHENA-CDS Knowledge Base can encode this condition as an instance of the General_Comparison_Criterion. Figure 88 shows the *entry_type*, *domain_term*, and *valid_window* attributes, whose values are used by the Guideline Interpreter to get all numeric entry instances whose domain term is Creatinine and whose timestamp is *within last 2 months*. The aggregation operator, *minimum*, is used to identify the minimum among the queried values. The *value* (Value) attribute specifies that the minimum value has to be greater than the upper limit of normal of Creatinine. Following are explanations of each attribute:

- *Entry_type (Entry Type)* – The attribute can be Numeric_Entry or Note_Entry.
- *Domain_term (Domain Term)* – This is a class from the domain concept model.
- *Aggregation_operator (Aggregation Operator)* – Possible values are minimum, maximum, most recent, average, and count (the number of returned values).

- *Valid_window (Valid Window)* – The valid_window attribute specifies how far back to look for the value being queried. It is an instance of Relative_Time_Interval_Definite (Figure 89).²³
- *Operator* – Possible values are $>$, \geq , $<$, \leq , $=$, eq, and neq; eq and neq are used to compare non-numeric values.
- *Assume_if_no_value (Assume If No Value)* – Possible values are no_assumption, assume_satisfied, assume_unsatisfied, and use_default. If this slot is not filled in, or if no_assumption is specified, evaluation of the criterion makes no assumption and the result may be *true*, *false*, or *unknown*. Assume_satisfied or assume_unsatisfied mean the criterion is assumed to be true or false, respectively, if there is no value for the domain term. Use_default means the default value specified in the default_value (Default Value) slot should be used. If no assumption is specified and there are no data for evaluating the criterion, *unknown* is the result of evaluating the criterion.

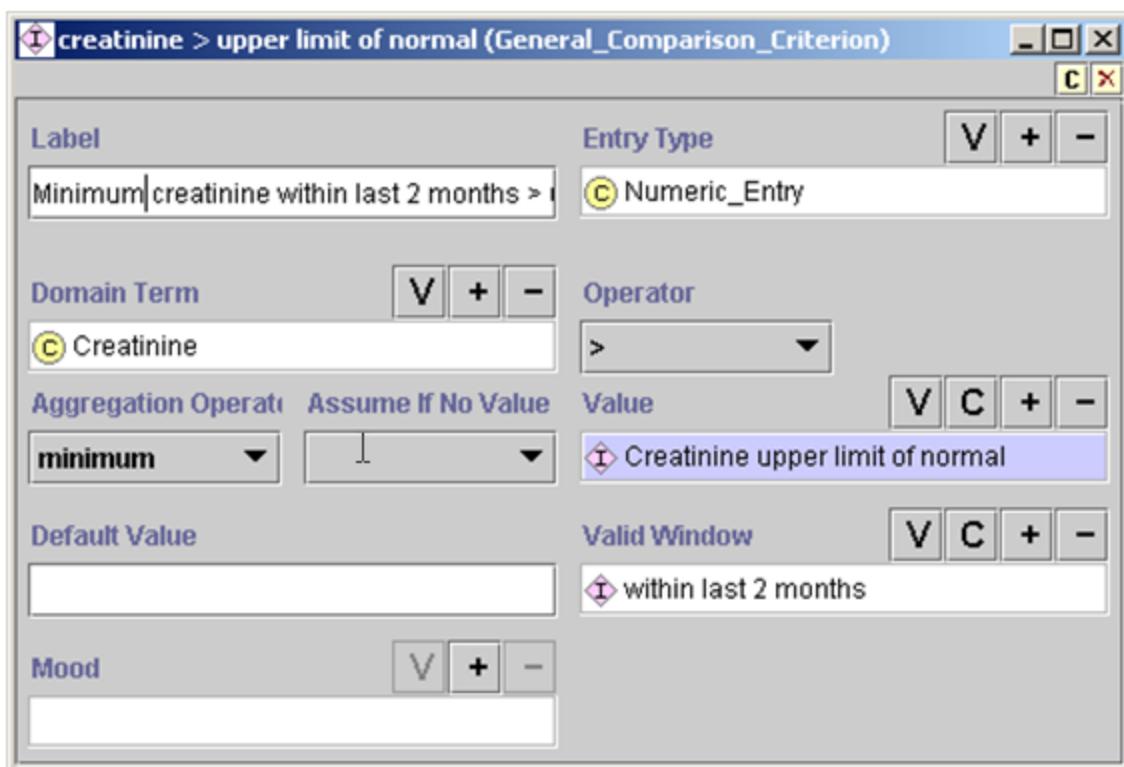


Figure 88 - An instance of General_Comparison_Criterion, stating that the minimum creatinine value within the last two months is greater than the upper limit of normal

²³ There are other types of time intervals and time points that can be specified using the classes in the Time_Entity hierarchy in the EON Guideline Model, but Relative_Time_Interval_Definite is the only one necessary for the ATHENA system. Other classes will not be documented in this manual.

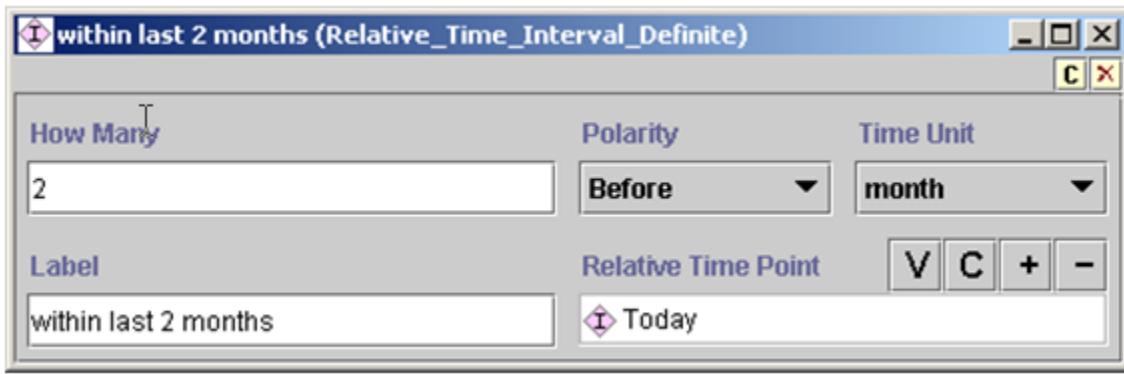


Figure 89 - An instance of Relative_Time_Internal_Definite, specifying the time interval “two months before today”

The *value* (Value) slot of a General_Comparison_Criterion should be an instance of Expression, whose subclasses are:

- *Binary_Expression* – subtraction or division
- *N_ary_Expression* – multiplication or addition of multiple arguments
- *Controlled_Term* – reference to controlled terminology; e.g., the Male class from the medical concept model
- *Numeric_Constant* – an instance with a number in its *value/Value* slot
- *Qualitative_Constant* – an instance with an arbitrary string in its *value/Value* slot
- *Get_Class_Slot_Value and Get_Slot_Value* – classes with instances specifying queries in Protégé to get a slot value either from a class or from an instance
- *EPR_Query* – Queries for patient data in the formats of the EPR_Entities (see Subsection IV.2.1)

The definitions of most Expression classes are straightforward. Here, only Get_Class_Slot_Value and the types of EPR_Query are illustrated. Both Get_Class_Slot_Value and Get_Slot_Value are designed to query values from the Protégé classes and instances so they can be compared with patient data. Figure 90 shows an instance of Get_Class_Slot_Value that queries for the value of the UpperLimitOfNormal slot from the Creatinine class. (Recall that Interval-Valued_AtomicTest_MetaData metaclass, described in Subsection IV.2.2.1, allows class instances of this metaclass to specify upper and lower limits of normal.)

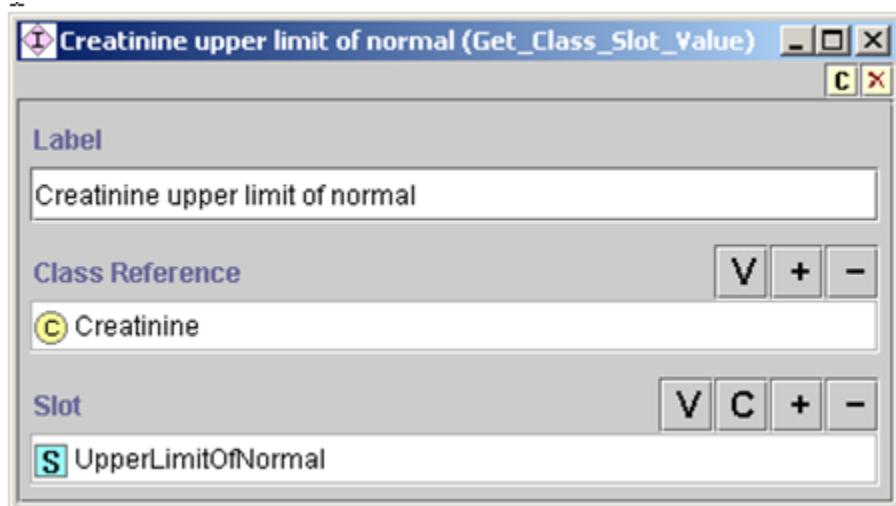


Figure 90 - An instance of Get_Class_Slot_Value. It obtains the value of the UpperLimitOfNormal slot from the Creatinine class.

Among the subclasses of the EPR_Query class, Numeric_Term_Query and Qualitative_Term_Query may be useful. An instance of Numeric_Term_Query is shown in Figure 91. It specifies a query for the most recent value of the MD_Typical_Systolic_BP. MD_Typical_Systolic_BP is the systolic blood pressure a clinician enters into ATHENA-CDS to update an advisory. The query returns a numeric result.

An instance of Qualitative_Term_Query analogously has the same format but returns a string as its result.

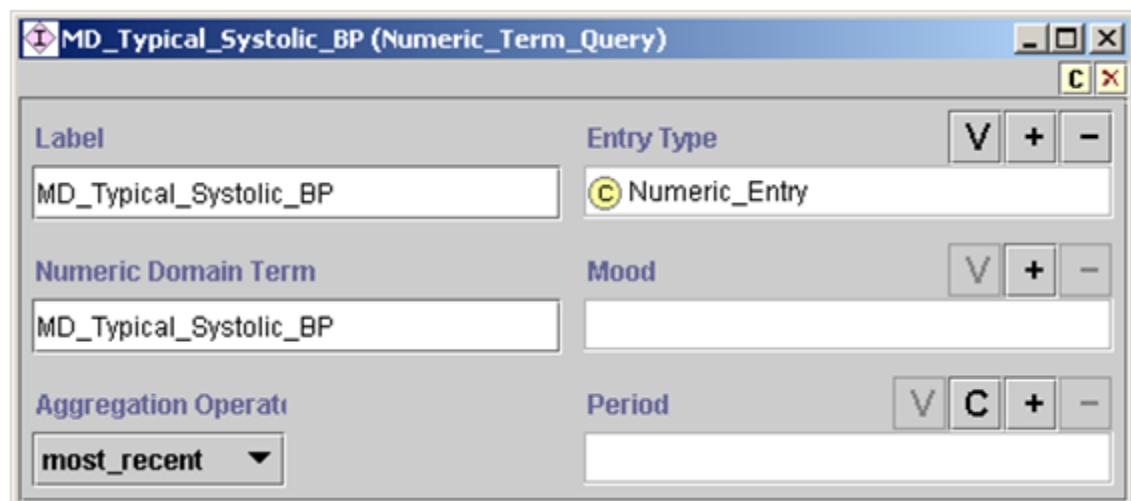


Figure 91 - A Numeric_Term_Query that asks for the most recent value of the MD_Typical_Systolic_BP

N_ary_Criterion

Instances of N_ary_Criterion²⁴ allow other criteria to be specified in Boolean combinations (*and*, *or*, and *not*). Figure 92 illustrates how to combine Presence_Criterion and General_Comparison_Criterion, discussed above, to form a conjunctive n-ary criterion using the *and* operator.

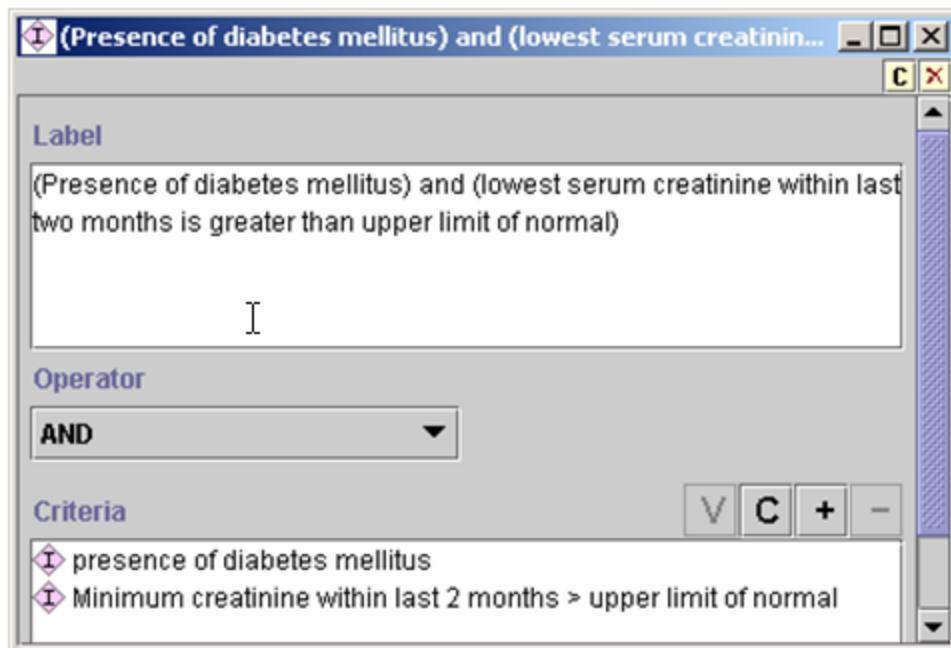


Figure 92 - An instance of N_ary_Criterion that requires both of its constituent criteria to be true, as indicated by the AND operator

Recall that criteria expressed in the template-based expression language can evaluate to *true*, *false*, or *unknown*. Accordingly, the truth tables used to derive the truth values of N_ary_Criterion take the *unknown* truth value into account. The following truth tables are used:

AND	true	false	unknown
true	true	false	unknown
false	false	false	false
unknown	unknown	false	unknown

OR	true	false	unknown
true	true	true	true

²⁴ By analogy with unary, binary, etc., “n-ary” means “any number of”. Here it refers to the fact that the *and* and *or* operators can take any number of arguments.

false	true	false	unknown
unknown	true	unknown	unknown

NOT	true	false	unknown
	false	true	unknown

Other criteria templates in the EON Guideline Model:

- support simplified numerical value comparisons (`Numeric_Term_Criterion`, as illustrated in Figure 93), where the value slot can only be a number, instead of an instance; and
- allow a statement expressing that the goals associated with a guideline are being reached (`Goal_Criterion` as explained in Subsection IV.2.3.1 and illustrated in Figure 58).

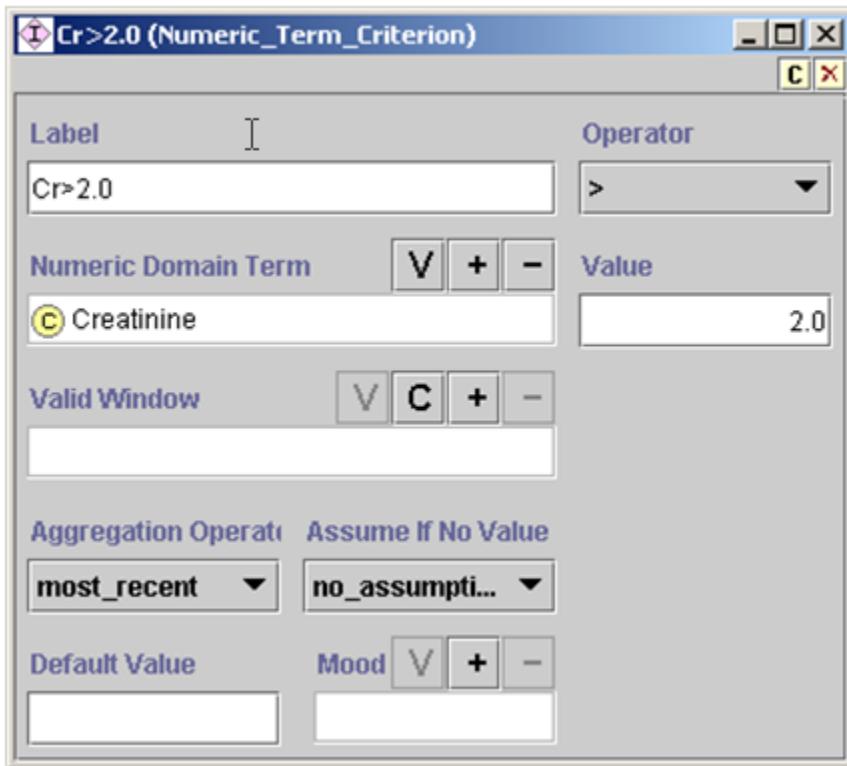


Figure 93 - An instance of `Numeric_Term_Criterion`. It is a simplified version of `General_Comparison_Criterion`.

IV.2.4.2. PAL-based Expression Language

A guideline author can use the templates described so far to write most common decision criteria with little training. The criteria that can be written in this way, however, are not sufficiently expressive. Take the expression, “Presence of an authorized medication that is contraindicated by some medical condition”. It requires that, for each authorized medication, the Guideline

Interpreter finds its contraindications from the ATHENA-CDS Knowledge Base and checks to see if there is any patient-data instance that suggests the presence of these contraindications.

Instead of trying to resolve such complex criteria procedurally, the Protégé Axiom Language (PAL)—a constraint language—is used to write them. The PAL constraint language implements a subset of first-order predicate logic written in Knowledge Interchange Format (KIF) syntax [22](Genesereth, 1991). It makes full use of Protégé’s frame-based knowledge model. For example, variables can range over instances of Protégé classes (e.g., in Figure 94, the variable ?current_med ranges over instances of Medication class). Attributes of classes (e.g., Absolute_Contraindications in Figure 94) can be used as a binary predicate to check that a constant or the value of a variable (e.g., ?contraindication) is a value of the slot. An attribute (e.g., domain_term in Figure 94) can also be used as a function of one argument that returns the

```
(defrange ?current_med :FRAME Medication)
(defrange ?problem :FRAME Note_Entry)
. . .
(exists ?current_med
  (and (patient_id ?current_med $patient_id)
    (exists ?med_class
      (and (subclass-of
        (drug_name ?current_med) ?med_class)
        (exists ?contraindication
          (and (Absolute_Contraindications
            ?med_class ?contraindication)
          (exists ?problem
            (and (patient_id ?problem $patient_id)
              (subclass-of
                (domain_term ?problem)
                ?contraindication)))))))
```

value of the slot for an instance. It is not expected that domain experts untrained in logic will formulate these complex logical criteria. Full documentation of the PAL constraint language is available at the URL: <http://protege.stanford.edu/plugins/paltabs/pal-documentation/index.htm>.

Figure 94 - Example of a PAL expression

The PAL language was originally designed for writing constraints on concepts and relationships in a Protégé knowledge base. In adapting it for writing decision criteria that involve patient data, it is necessary to introduce a variable (\$patient_id in Figure 94) that stands for the patient ID that, at run time, the Guideline Interpreter will use to replace the variable before evaluating a PAL criterion. Thus, an instance of PAL_Criterion (Figure 95), aside from a label, has two parts: (1) the case_variable slot (shown as Case Variable) that tells the Guideline Interpreter which string in the PAL expression stands for the patient ID, and (2) the slot that holds the PAL expression itself (such as the one in Figure 94).

The PAL constraint language has a query form that, instead of evaluating to *true* or *false*, returns instances that satisfy constraints written in PAL. Instances of PAL_Query, which has exactly the same format as PAL_Criterion, contain a PAL query (instead of a PAL constraint). PAL query is used in a number of places. For example, PAL query can be used to find all current medications, such that each of them is contraindicated by a problem a patient is experiencing. Furthermore, in Substitute_Activity, the drug to be deleted is specified using a PAL query. PAL query is also used to obtain the names of drugs whose doses need to be increased.

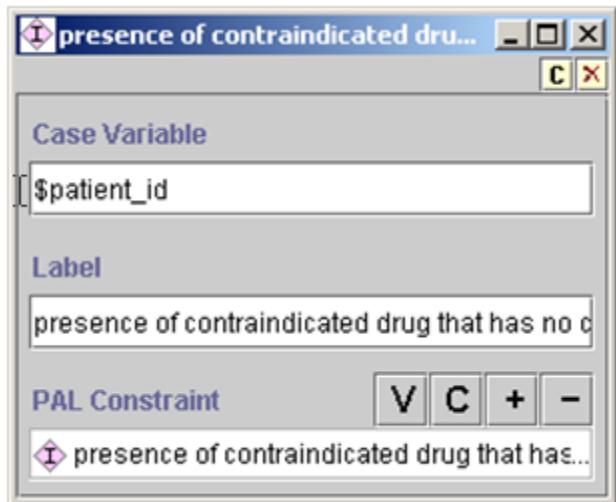


Figure 95 - An instance of PAL_Criterion

IV.2.4.3. Summary

Criteria languages in the EON Guideline Model are all informed by the same patient data model and serve complementary requirements. The template-based language provides form-based templates that domain experts can easily use to write the common decision criteria. The logic-based PAL language adds additional expressiveness for writing specific types of complex criteria. Both the template-based and the PAL languages make use of the taxonomic hierarchies in the medical concept model to infer generalization/specialization relationships and to make abstractions based on definitions embedded in the hierarchies (e.g., the definition of *hyponatremia*).

A common evaluation-result interface unifies the usage of the template-based and PAL criteria. For criteria written in each language, the guideline execution engine invokes the appropriate criteria-evaluation engine. Each criteria-evaluation engine returns the result consisting of the criterion being evaluated, the truth value of the criterion as applied to available patient data, and annotations that can be used for explanation purposes.

IV.3. Guideline Interpreter's Use of the Knowledge Base

This subsection considers how the Guideline Interpreter generates a guideline-based advisory in response to requests from client programs. It will refer to the PCAServer (Protocol Compliance Adviser Server), which is a computer program that provides a collection of methods for client programs (such as ATHENA-CDS Client) to request and obtain advisories computed by the Guideline Interpreter.

IV.3.1. Overview

As described in Subsection II.2.1, the PCAServer is implemented as a CORBA server. It has two external interfaces: the PCAServer interface and the PCASession interface. The PCAServer process is started on a central server machine with an initialization file that sets up a number of configuration parameters (see Table 2). A client program uses the PCAServer interface to open a client-specific PCASession. A client program can be the batch program that invokes the PCAServer to compute and store precomputed advisories, or it can be the ATHENA-CDS Client that runs on a clinical workstation. A client program interacts with the PCASession to obtain appropriate guideline recommendations (see Figure 96). The ATHENA-CDS Client, for example, may request updated advisories, precomputed advisories, or the computation of advisories when there are no precomputed advisories. All copies of the PCASession share the same guideline knowledge base loaded in the PCAServer, but otherwise are independent of each other. Thus, each copy of PCASession handles requests from a running ATHENA-CDS Client and keeps data for different patients separate from each other.

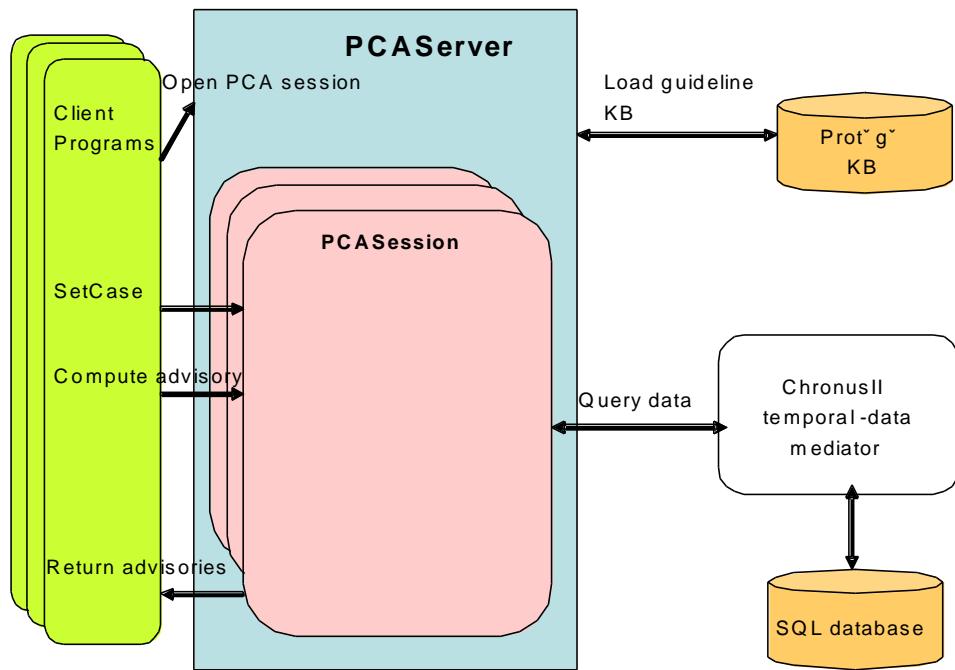


Figure 96 - The PCAServer and PCA Session. Each ATHENA-CDS client calls the PCAServer in order to open its own PCA Session, and interacts with the PCA Session to get advisories. PCA Sessions share the same guideline knowledge base loaded in the PCAServer, but manage patient data specific to the requested cases.

Parameter Name	Possible Parameter Values	Comment
DATABASE	String (e.g., ATHENEONPA)	ODBC data source of patient data
SERVER_LOGFILE	Path to a directory (e.g., \\myPC\c\ATHENADSS\log\PA\)	Directory to which server log files should be written
SERVERKB	Path to Protégé project (e.g., \\myPC\c\ATHENADSS\domain_model\ATHENA.pprj)	The knowledge base to load
PCAINIFILE	Path to ini file (e.g., \\myPC\c\ATHENADSS\ini\ATHENA.ini)	An argument in opening PCASession used to initialize ChronusII
PCAOUTPUTDIR	Path to a directory (e.g., \\myPC\c\ATHENADSS\output\PA\)	Directory from/to which precomputed advisories can be read/written

Table 2 - PCAServer initialization parameters

The PCASession interface has a number of operations that can be used by a client program such as ATHENA-CDS Client or the batch program (Table 3). For example, the setGuideline operation selects the guideline that should be used for the session, and setCase selects the patient for whom the guideline-based advisory should be generated. Other operations compute advisories, load precomputed advisories, send data updates to the server, and request updated advisories. A printData operation returns an HTML version of the data that are used to compute the advisories.

Operation Name	Parameters	Returned Value	Operation
setGuideline	guideline_name	None	Select the guideline to apply.
setCase	Patient id, session time	None	Load data from database, and set the time for which advisories are generated (default to current time).
computeAdvisories	None	Set of guideline service records	Given selected guideline and selected patient, compute advisories.

computeAndStoreAdvisories	Storage directory	Set of guideline service records	Precompute advisories.
loadPrecomputedAdvisories	Patient id, session time, storage directory	None	Load precomputed advisories from storage directory (session time not used).
containAdvisories	None	True/False	Check whether precomputed advisories are successfully loaded.
getAdvisories	None	Set of guideline service records	Get (usually precomputed) advisories.
resetAdvisories	None	None	Delete current advisories.
updateData	Patient data	None	Send a collection of patient data.
updateAdvisories	None	Set of guideline service records	Update advisories after data updates.
printData	None	String	Return a listing of patient data in HTML form.
setDummyCase	None	None	Set up a dummy patient case in order to generate advisories based purely on data entered through a user-interface.

Table 3 - Operations available in the PCA Session interface

PCASession provides operations, such as computing and loading precomputed advisories, which are custom-tailored to the requirements of the ATHENA-CDS SYSTEM. The EON guideline execution architecture allows such adaptations by providing a set of generic services that the PCASession can use to implement the application-specific functionalities. These generic services include, for example, evaluating eligibility criteria, testing guideline goals, deriving abstractions about patient states, and traversing a clinical algorithm to evaluate evidence-based preferences for alternatives in management decisions. The generic services, based on the capabilities of the EON Guideline Model, are defined by a separate interface that is implemented by a program referred to as the core EON guideline execution engine (CEGEE). It would go beyond the scope of the ATHENA-CDS manual to describe the CEGEE.

The Guideline Interpreter converts patient data into:

1. categorical conditions or interventions (e.g., atrial fibrillation and ACE inhibitor) whose presence or absence can be tested using Presence_Criterion (see Subsection IV.2.4.1),
2. qualitative data (e.g., proteinuria with values of 1+, 2+, etc.) that have a term and a nominal value, and that are used in General_Comparison_Criterion,
3. numeric data that have a term and a numeric value (e.g., that can be tested using Numeric_Term_Criterion or General_Comparison_Criterion),
4. demographic information (e.g., sex) that can have a nominal value (e.g., male) or a numeric value (e.g., 56 years old), and

5. allergy data indicating a substance (often drugs) to which the patient is allergic.

The following subsections give a sense of the guideline advisories and their relationship to the ATHENA-CDS Knowledge Base. They describe the structure of the advisories and how the PCASession's computeAdvisories operation generates guideline recommendations for the ATHENA-CDS Client.

IV.3.2. Structure of Guideline Advisories

The PCASession delivers to the client program guideline advisories consisting of a set of Guideline Service Records. Each Guideline Service Record is composed of:

- *an assumption* – in the form of a goal criterion and its assumed status (*true*, *false*, or *none* when no assumption is made);
- *a collection of conclusions* – in the form of parameter/value/justification, about the patient (e.g., eligibility status and patient characteristics, such as JNC risk group classification);
- *a collection of chosen scenarios* – e.g., patient in the scenario of receiving two anti-hypertensive medications;
- *a collection of guideline goals* – e.g., target BP=135/85 because of Diabetes Mellitus;
- *a collection of activities* – evaluated in terms of their indications, contraindications, interactions, and side effects (e.g., evaluation of adding the ACE inhibitor drug class);
- *details of the actions associated with a decision point* – e.g., onscreen messages, evaluating drugs to add, and dose increases; and
- *a collection of decision points* – each of which containing a ranked list of alternative actions. The decision points and their associated actions are based on the choice and branch nodes in the clinical algorithm and consultation guidelines. The result of evaluating (strict) rule-in and (strict) rule-out criteria determine the ranking of the alternatives. (For example, for a patient taking one antihypertensive agent, the alternatives include intensifying treatment, staying with the same drug regimen, and substituting another drug.)

When there is no blood pressure in the records, the advisories will be two sets of the Guideline Service Records, one for the assumption that the target blood pressure is satisfied and one for the assumption that it is not.

Guideline_Service_Records are the recommendations that the Guideline Interpreter delivers to client programs, which present the recommendations in format appropriate for the user interacting with the client programs. ATHENA-CDS Client, for example, presents the recommendations in a graphical user interface described in Subsection II.1. The batch program that precomputes the advisories (Subsection II.2.2.2), on the other hand, generates an HTML document that shows the recommendations in textual format. The HTML view the recommendations is designed to facilitate the debugging of the output of the Guideline Interpreter.

The following examples illustrate the structure of a Guideline Service Record. Following each example, in boxed figures, is the HTML output based on the recommendation structure described in the example.

EXAMPLE of a Guideline Service Record – Part 1

Shown first are how the assumption, conclusions, chosen scenarios, and goals are represented, for a sample patient, in the Guideline Service Record.

Assumption: none²⁵
Conclusions: Eligible for JNC-VI Hypertension Guideline: true
Risk Group: C
Scenario Choice: on one anti-hypertensive drug
Guideline Goals: goals: SBP < 130 and DBP < 85
justification: presence of diabetes, heart failure or renal insufficiency
Goal state: failed evaluation record of goal criteria

²⁵ Assumption is *none* because, in this example, blood pressure data exist in ATHENA and, therefore, no assumption is made about whether the guideline goal is achieved.

Patient classification:

- JNC-VI Hypertension Guideline:true[because *Eligibility criteria evaluate to true(presence of diagnosis of hypertension && absence of renovascular disease && no diagnosis of pregnancy && Creatinine(1.3/2002-8-16) && Absense of Secondary Hypertension && absence of spinal cord injury && absence of narcolepsy && Not taking cyclosporine && Not taking spironolactone && Not taking minoxidil && absence of renovascular disease && absence of IHSS && Absence of Ascites && Not off guideline && Age(62.0/1999-12-23) && Antihypertensive_Agents && not taking tacrolimus && absence of transplant recipient)*]
- Risk Group C (presence of TOD/CCD or DM):[because *presence of TOD/CCD or DM evaluate to true(presence of TOD/CCD)*]

Scenario choice: on one anti-hypertensive drug

Goal: SBP < 130 and DBP < 85(presence of diabetes, heart failure or renal insufficiency)

Reached goal? failed(Treatment_Systolic_BP(142/DB_Systolic_BP))

Figure 97 HTML view of the parts of a Guideline_Service_Record that contains the PCAServer's conclusions about a patient (Patient classification), the scenario and goal chosen for the patient, and whether the patient reached the goal.

EXAMPLE of a Guideline Service Record – Part 2

The following shows a sample of three guideline decision points. The first one (*on medication: things to check*) is derived from the consultation guideline *on medication*. Each decision point has a set of associated actions (e.g., *Warning pregnancy female<50yrs and women >60 and menopause*). Each action has a preference (e.g., *ruled_out* or *preferred*) that is computed from the condition associated with the action. The result of computing the condition is detailed in the justification part (e.g., *not female less than 50 years*). The action_specification portion of the action contains details of the action (e.g., specific onscreen messages or increasing dose).

1. on medication: things to check
 - * Warning pregnancy female<50yrs

```
preference: ruled_out
justification: not female less than 50 years
action_specification - (detailed on-screen message)
```
 - * women >60 and menopause

```
preference: preferred
justification: rule in criterion female >=60 evaluate to true
action_specification - (detailed on-screen message)...
```
2. one-drug-therapy-choices
 - * Blood pressure not adequately controlled; intensify drug treatment

```
preference: preferred
```

```
justification: BP not controlled
action specification - (detailed conditional on-screen message)
* continue with one-drug regimen
preference: ruled_out
justification: BP not controlled...
* on one drug, consider substitution
preference: ruled_out
justification: BP not controlled...

3. step up choices
* Increasing dose
preference: preferred
justification: drug dose not at maximum
action specification - (Evaluate Increased Activity Intensity to evaluate
possibility of increasing dose)
* evaluate new drug to prescribe
preference: preferred
justification: always true
action specification - Evaluate Start Activity to evaluate various drug
classes...
```

Only actions that are preferred are shown on the HTML output. Accordingly, for the examples shown above, the text in Figure 98 will appear on the HTML output.

Action Choices

- **assumption women >60 and menopause preferred**(rule in criterion *female* $>= 60$ evaluate to **true** because *Sex(Female) && Age(62.0/1999-12-23)]*) We assume women older than 60 years are postmenopausal.
- ... (**other messages**)
- **Blood pressure not adequately controlled; intensify drug treatment preferred**(strict rule-in condition *BP not adequately controlled based on most recent BP* evaluate to **true** because *BP not adequately controlled based on most recent BPTreatment_Systolic_BP(142/DB_Systolic_BP)]*) Consider INTENSIFYING drug treatment: BP ELEVATED based on most recent available BP.
- **Increasing dose preferred**(strict rule-in condition *dose level not at maximum and if taking diuretics, not hypokalemia* evaluate to **true** because *thiazide diuretics not being used && there exists non-contraindicated drug not at maximum dose]*)
- **evaluate new drug to prescribe preferred**(strict rule-in condition *true* evaluate to **true**)]

Figure 98 HTML view of the parts of a Guideline_Service_Record that contains the PCAServer's conclusions about actions to be chosen and messages associated with the actions

EXAMPLE of Guideline Service Record – Part 3

There are four varieties of activity evaluation: add, delete, substitute, and change attribute. Following are brief examples of change attribute and add evaluations. The structure of delete evaluation is similar to that of add, and substitute evaluation is a combination of add and delete evaluations.

```
Change attribute evaluation
  name: lisinopril
  attribute_name: daily_dose
  change_direction: up
  messages: none
  side_effects: none
```

```
Add evaluation
  activity_to_start: Cardioselective Beta Blocker
  relative_indications: Coronary_Artery_Disease(412., 414.9)
  relative_contraindications: Obstructive_Pulmonary_Disease(492.8, 496.)
  messages:
    - name: add beta blocker, obstructive pulmonary disease
```

- text: Cardiovascular benefits of beta blocker therapy may outweigh the increased risk of bronchospasm in this patient
- action_spec_class: On_Screen_Message...

The corresponding HTML text is shown in Figure 99, below.

If changing attribute, consider:

- lisinopril(change drug_daily_dose: up)

Cardioselective Beta Blocker (atenolol)(based on ATHENAINSTANCE_00046)

- Relative indications: Coronary_Artery_Disease(412., 414.9)
- Relative contraindications: Obstructive_Pulmonary_Disease(492.8, 496.)
- add beta blocker, obstructive pulmonary disease(Cardiovascular benefits of beta blocker therapy may outweigh the increased risk of bronchospasm in this patient.)

Figure 99 The HTML output showing the drug recommendation portion of the Guideline_Service_Record

IV.3.3. Generation of Guideline Advisories

This subsection describes algorithms implemented by the computeAdvisories operation of PCASession.

The client program uses the setGuideline and setCase operations to indicate which guideline to apply to the patient and which patient case to load. The algorithm implemented in computeAdvisories involves the following steps (also see Figure 100):

1. Invoke the core EON guideline execution engine (CEGEE) to determine whether the patient satisfies the eligibility criteria of the guideline. If the result is *unknown* because of lack of sufficient data or is *true*, guideline advisories will be generated for the patient case.
2. Check to see if goals of the guideline have been met.
3. If the result of goal evaluation is *unknown*, computeAdvisories is programmed to make alternative assumptions. It will generate advisories assuming the goal has been met, and

advisories assuming the goal has not been met.²⁶ If the result of the goal evaluation is either *true* or *false*, then no assumption is made.

4. For each set of advisories, the computeAdvisories operation first asks the core EON guideline execution engine to evaluate a set of patient characteristics (e.g., hypertension risk groups). These patient characteristics are abstractions of the patient state that are computed even if they are not required for computing the rest of the guideline advisories.
5. To compute the rest of the advisory, the core EON guideline execution engine traverses the clinical algorithm of the guideline. It determines the starting point of traversal by evaluating the scenarios of the clinical algorithm (e.g., patient taking one anti-hypertensive agent). The CEGEE evaluates the preconditions of all scenarios, and presents the results for the calling program (computeAdvisories in this case) to make a selection.²⁷
6. The computeAdvisories method was programmed to make automatic selections based on the results of precondition evaluations. It selects the first scenario whose precondition evaluates to true.²⁸ Starting from the initial scenario, the core EON guideline execution engine traverses the clinical algorithm. For each decision point that the CEGEE reaches in its traversal, it creates a record of guideline decision structure—as shown in Subsection IV.3.2.—and fills in the various slots of that record. For each action choice alternative at a decision point, it evaluates the associated rule-in and rule-out criteria. The result of these evaluations is stored in a justification record, and a preference such as *preferred*, *neutral*, or *rule-out* is computed for each choice. The justification record refers to the evaluated guideline criteria, the patient data used, and assumptions, if any, made in evaluating the criteria. The *actions* slot associated with each action choice (see Subsection IV.2.3.2) defines the set of actions to be performed if the choice is selected (see item #7, below).
7. The computeAdvisories operation is programmed to make automatic selections based on the preferences determined through the rule-in and rule-out criteria. (An action choice is selected only if its preference rating is *preferred*, which occurs only if a strict rule-out criterion evaluates to *false* or *unknown*, and a strict rule-in criterion evaluates to *true*.)
8. For each choice that is selected by computeAdvisories, the CEGEE continues the traversal of the clinical algorithm, presenting additional choices to the computeAdvisories program if there are more decision points in the algorithm.

²⁶ If a guideline can set more than one goal for a patient, the implementers of ATHENA DSS will have to decide what assumptions to make and reprogram the PCASession code. The core EON guideline execution engine can accept any number of and any combination of assumptions.

²⁷ The core EON guideline execution engine can be adapted to different usages. Instead of having preprogrammed responses to the choices presented by the execution engine, for example, another system may involve human users making selections.

²⁸ Thus, it is important that all starting scenarios be mutually exclusive.

9. After the core EON guideline execution engine finishes traversing the clinical algorithm and evaluating the action choices selected by the computeAdvisories operation, computeAdvisories returns advisories to its calling program in the form of completed Guideline Service Record structures (see Subsection IV.3.2).

The computeAdvisories operation of PCASession is the basic mechanism for generating an ATHENA-CDS advisory. The computeAndStoreAdvisories operation is implemented by calling the computeAdvisories operation. It then stores the advisories and the data used to compute the advisories as a Java serialization file. The updateAdvisories operation deletes previously computed results and calls computeAdvisories to generate new advisories. The loadPrecomputedAdvisories operation simply loads the precomputed advisory file, restores the Guideline Service Record structures, and returns them to the client.

In Figure 100, each highlighted rectangle represents the invocation of a service provided by the core EON guideline execution engine. The continueComputeAdvisory circle represents a sub-algorithm that may be invoked with assumptions of the guideline goal either being satisfied or not (rectangles labeled “3.” in Figure 100).

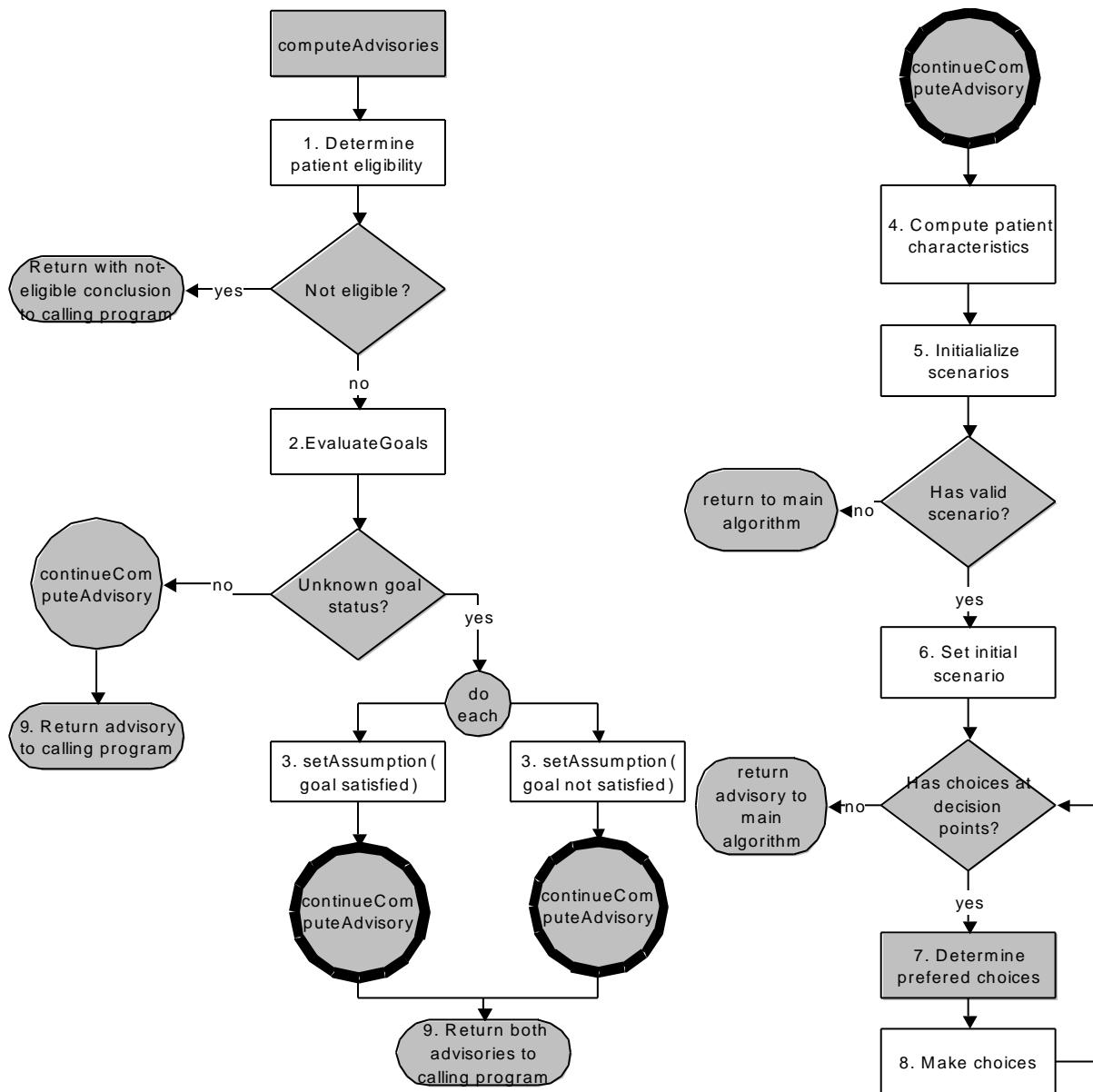


Figure 100 - The computational algorithm used by the computeAdvisories operation

IV.4. Textual Knowledge Sources

The ATHENA-CDS SYSTEM contains two basic textual knowledge sources:

1. Supporting Material, which contains HTML pages with information and references for the relationship between a specific drug class and its compelling indications, and

2. the textual information displayed in the Lifestyle, Adherence, and Glossary tabs of the ATHENA-CDS hypertension advisory.

IV.4.1. Supporting Material

The ATHENA-CDS System initially generated advisories with hypertension management recommendations based on the *JNC 6* guidelines, the national *VA-DoD Guidelines*, and new evidence published after the guidelines publication that has been agreed upon by the collaborating investigators for ATHENA-HTN project. ATHENA-HTN was later updated to *JNC 7* guidelines [23]. Supporting Material substantiates ATHENA-CDS recommendations, by furnishing users with information at the basis of decisions and clinical suggestions in advisories. As described in detail in Subsection IV.4.1.3, users can drag and drop a magnifying glass icon onto a compelling indication to bring up supporting material for it in a web browser.

IV.4.1.1. Source Documents for Supporting Material

Much of the material in this section refers to the initial implementation of ATHENA-HTN in the clinical trial in early 2000's. Not all of this functionality was continued in later implementations.

The ATHENA-CDS Project conducted a literature review for the compelling indications present in the ATHENA-HTN Knowledge Base. The review identified the evidence base for ATHENA-HTN's compelling drug recommendations. Based on the literature for each compelling indication, a short summary was created with three or four summary statements about the evidence for each recommended drug together with a table describing the main elements of the studies supporting the drug recommendation. The system included links to display *ACP Journal Club* abstract if there was one, and *PubMed* abstracts for relevant papers.

The Supporting Material webpage contains six tabs: Short Summary, *ACP Journal Club*, Clinical Evidence, *JNC 6*, *PubMed*, and VA Guidelines. The documentation is in HTML pages contained within the ATHENA DSS directory (ATHENADSS\doc\ATHENA\EvidenceDisplayATHENA\...). The source documents for the supporting material are:

1. *ACP Journal Club* – Permission was obtained to display specific abstracts in HTML format for research purposes in the clinical trial of implementation of ATHENA-HTN in three medical centers in 2001-2002.
2. *Clinical Evidence* – Permission was obtained to display sections of this publication in HTML format for research purposes in the clinical trial of implementation of ATHENA-HTN in three medical centers in 2001-2002.

3. *The Sixth Report of the Joint National Committee* (JNC 6) – This publication is free of copyright. Sections were transformed into HTML pages for display.
4. *PubMed* – Abstracts are free of copyright. An HTML page with abstracts was created.
5. *VA Guidelines* – This publication is free of copyright, and it is available in HTML format.

IV.4.1.2. Creating an Association between a Medical Condition and a Drug Recommendation

A relationship can be established between a medical condition and a drug within the *Drug_Medical_Condition_Relation* class. *Drug_Medical_Condition_Relation* is a subclass of *Drug_Relation*, which is contained by *Medical_Domain_Class*. The relationship between the drug and the medical condition may be represented as the medical condition being an *indication* for the drug, the medical condition being a *contraindication* for the drug, or the medical condition being a condition that *complicates* the use of the drug without necessarily being either an indication or a contraindication. In an instance of *Drug_Medical_Condition_Relation*, the URL for Supporting Material can be specified by creating an instance of the *Supporting_Material* class, and added as the value of the Supporting Material slot.

How to create supporting material for a drug indication:

1. Go to *Drug_Medical_Condition_Relation* in the class hierarchy.
2. Select type of relation: *indication(Drug_Indication_Relation)*, *contraindication (Drug_Contraindication_Relation)* or *complication (Drug_Complication_Relation)*
3. In the class section (middle section) click on **C** to create a new instance.
4. Add the indication (condition of interest), the drug (drug class or specific drug), and the degree (compelling, relative, life_saving; see Figure 101). These *Drug_Medical_Condition_Relation* instances should correspond to the indications in the *Drug_Usage* form (e.g., ACE Inhibitor and Diabetes)

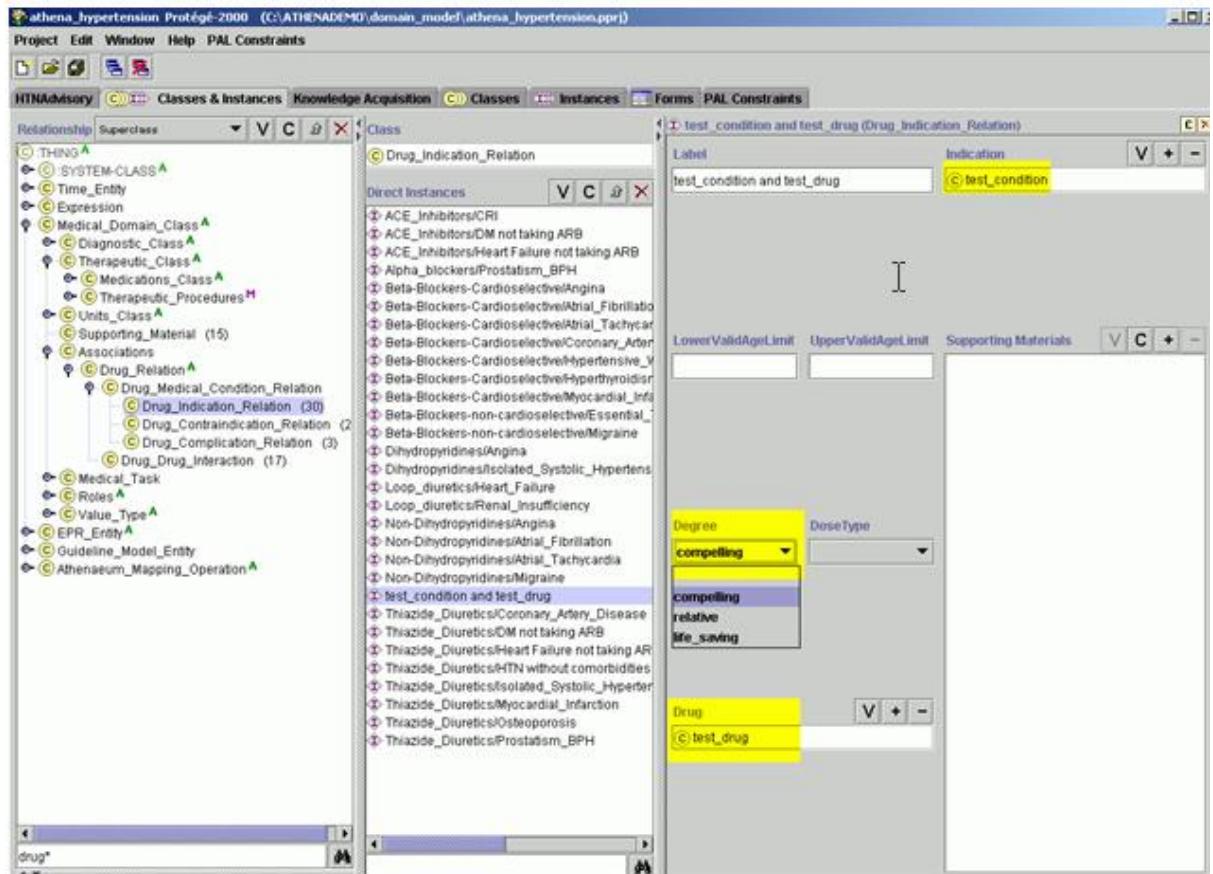


Figure 101 - Creating an association between a condition and a specific drug recommendation

5. Then in the Supporting Material slot, click on **C** to create an instance that holds the URL of the html page for the drug indication (see Figure 102).

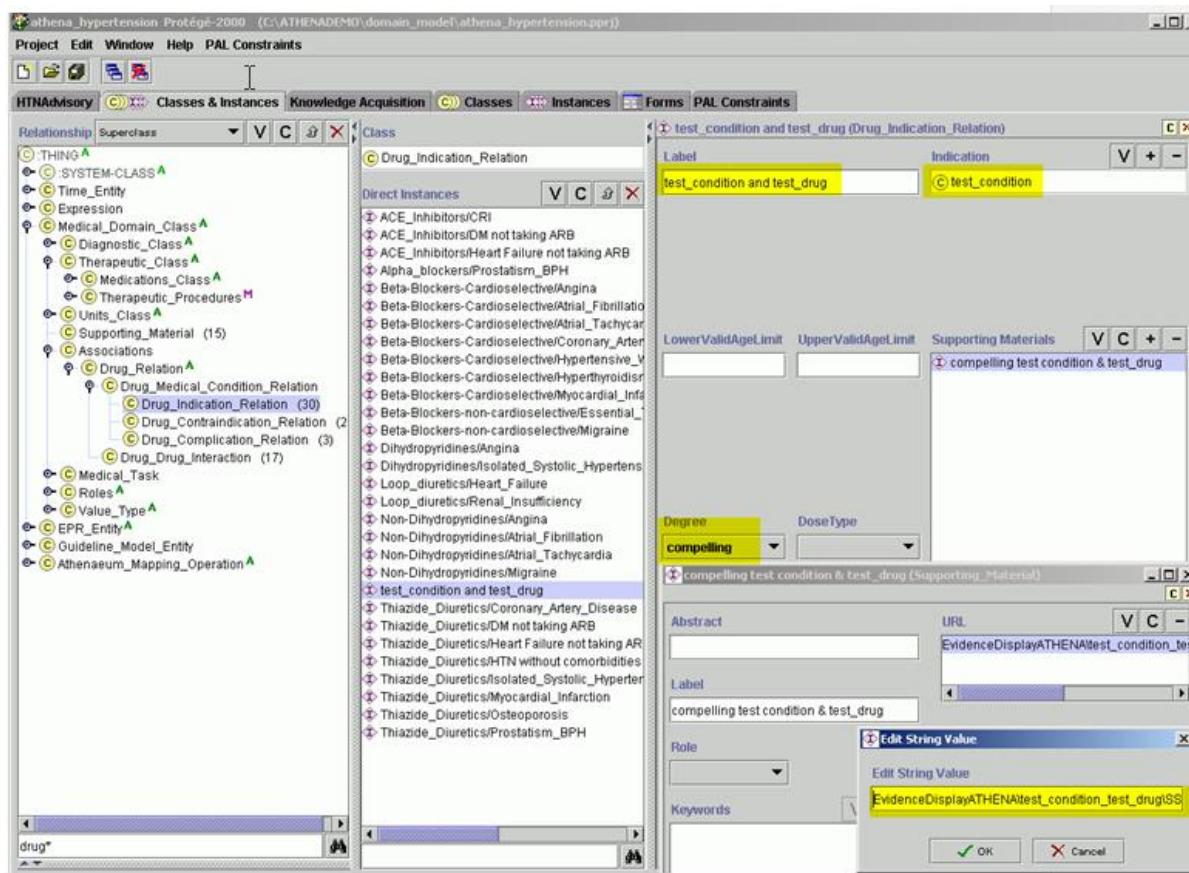


Figure 102 - Defining the URL link to the webpages containing supporting material

6. Currently, the URL always starts with EvidenceDisplayATHENA, followed by the folder name condition (for example:...\\EvidenceDisplayATHENA\\Diabetes_ACE). The correct way to insert the URL is, for example:
EvidenceDisplayATHENA\\test_condition\\test_drug\\SStest_conditionandtest_drug.²⁹
This is a relative URL, meaning that it is only part of the full path name to the file. The absolute path is composed inside the ATHENA-CDS Client when an instance of the Supporting_Material class is referenced. The documents or HTML files are kept on server filepath \\ATHENADSS\\doc\\ATHENA\\EvidenceDisplayATHENA to facilitate update management.

IV.4.1.3. Viewing Supporting Material for Compelling Indications

Currently, only compelling indications have supporting material. To display these materials in the ATHENA-CDS hypertension advisory, locate the compelling indication icon in the Reasons

²⁹ SS stands for short summary.

column of the Recommendations tab. Click on the magnifying class icon at the upper right of the panel and drag it over the compelling indication (see Figure 103).

The screenshot shows a software interface for managing blood pressure recommendations. At the top, there are fields for Patient SSN, Name, Most Recent BP in Database (142/72), Date, and a button to Update Advisory. Below this, a message states: "Guideline Goal: SBP < 130 and DBP < 85 [presence of diabetes, heart failure or renal insufficiency]" and "BP apparently NOT UNDER CONTROL, based on most recent available BP." A note says "(Enter "Today's Decision Blood Pressure" and press "Update Advisory" for new recommendations)". A yellow search icon is visible.

Below the message, a navigation bar includes tabs for Recommendations, Precautions, Assumptions, Lifestyle, Adherence, Glossary, and BP-Pr. A specific recommendation is highlighted: "Consider INTENSIFYING drug treatment: BP ELEVATED based on most recent available BP." To the right, a tooltip explains: "To view supporting material click the magnifying glass and hold and then drag it over a compelling indication in the "reasons" column in the "recommendations" tab." A red arrow points from this tooltip to the magnifying glass icon in the search bar.

The main content area displays a table of therapeutic possibilities:

Consider one of the following therapeutic possibilities	Click here for important ...	Reasons	Click here to provide ...
Increase dosage of lisinopril	Info	Isolated Systolic Hypertension	Feedback
Add DHP Calcium Channel Blocker (felodipine, nifedipine SA)	Info	Coronary Artery Disease	Feedback
Add Cardioselective Beta Blocker (atenolol)	Info	Obstructive Pulmonary Disease	Feedback

Figure 103 - Viewing supporting material

A new browser window will open, with the relevant HTML pages accessible through tabs (see Figure 104 through Figure 109). In the implemented system, these webpages are only available on the VA intranet; but they can also be viewed in the ATHENA-CDS demo version. Short Summary contains three to four bullet points about the benefits of adding the recommended drug, and a table displaying a summary of the studies supporting the drug recommendation. The other tabs contain abstracts of the main studies on the topic (*ACP Journal Club* and *PubMed* abstracts), a summary of the evidence on it (*Clinical Evidence*), the *VA Guidelines*, and *JNC 6*. Content relevant to the drug recommendation will be highlighted in red.

File Edit View Favorites Tools Help

Back Search Favorites Media Go

Address C:\ATHENA\doc\ATHENA\EvidenceDisplay\ATHENA\ISH_CCB\SS_ISHandCCB.html

Isolated Systolic Hypertension & Calcium Channel Blockers

[Short Summary](#) [ACP Journal Club](#) [Clinical Evidence](#) [JNC VI](#) [PubMed](#) [VA Guidelines](#)

Short Summary

Last Updated: November, 2001

- Isolated systolic hypertension and wide pulse pressure in elderly persons increase the risk for mortality and morbidity, treatment reduces this risk. ([Staessen 2000](#))
- Nitrendipine(DHP) compared to placebo reduced risk of fatal and non fatal stroke by 42% in two years, but not all cause mortality, in patients with isolated systolic hypertension. ([Staessen 1997](#))
- Although nitrendipine is not available in the USA, dihydropyridine calcium channel blockers (felodipine or nifedipine) are recommended as an alternative to thiazides in first line therapy of isolated systolic hypertension.

Table 1: Outcomes of a randomized controlled trial of nitrendipine vs placebo in patients with ISH

Study	Type of study	Patients	Intervention	Follow up	Outcomes	Event rate		Relative Risk Reduction
						Placebo	Active	
Staessen 1997	double blind randomized controlled trial	4695 patients, mean age 70 yrs, 66% women, mean SBP 173, mean DBP 87	Nitrendipine (step up with enalapril and hydrochlorothiazide) vs matching placebos	2 yrs (median)	Stroke/fatal and non fatal endpoints combined) a	77/2297 (3.4%)	47/2398 (2%)	42%
					Cardiac endpoints (fatal and non fatal endpoints combined) b,c	114/2297 (5%)	89/2398 (3.7%)	ns
					All-cause endpoints d	137/2297 (6%)	123/2398 (5.1%)	ns

Figure 104 - Supporting material (Short Summary webpage), Isolated Systolic Hypertension & Calcium Channel Blockers

Isolated Systolic Hypertension & Calcium Channel Blockers

[Short Summary](#) [ACP Journal Club](#) [Clinical Evidence](#) [JNC VI](#) [PubMed](#) [VA Guidelines](#)

ACP Journal Club ©

Therapeutics

Review: Isolated systolic hypertension increases mortality and morbidity in elderly persons and should be treated

ACP Journal Club. 2000 Sept-Oct;133:41.

Staessen JA, Gasowski J, Wang JG, et al. **Risks of untreated and treated isolated systolic hypertension in the elderly: meta-analysis of outcome trials.** Lancet. 2000 Mar 11;355:865-72.

Questions

In elderly persons, what are the risks associated with isolated systolic hypertension (≥ 160 mm Hg with diastolic blood pressure [BP] < 95 mm Hg), and what is the magnitude of the benefit associated with treatment?

Data sources

Studies were identified from 10 published overviews and 2 reports from trialist collaborations.

Study selection

Controlled trials were selected if elderly persons with isolated systolic hypertension were enrolled. Trials were excluded if all the study participants had comorbid conditions, such as stroke; if specialized care was

Figure 105 - Supporting material (ACP Journal Club webpage), Isolated Systolic Hypertension & Calcium Channel Blockers. Example shown for illustration. Portion of display shown with permission.

Isolated Systolic Hypertension & Calcium Channel Blockers

Short Summary ACP Journal Club Clinical Evidence JNC VI PubMed VA Guidelines

Clinical Evidence

Last Updated: January, 2002

Cardiovascular disorders
Primary prevention

[Clinical Evidence writers on primary prevention](#)

QUESTION What are the effects of drug treatment in primary hypertension?

OPTION Antihypertensive drugs versus placebo

Many systematic reviews of RCTs have found that drug treatment decreases the risk of fatal and non-fatal stroke, cardiac events, and total mortality in specific populations of people. The biggest benefit is seen in people with highest baseline risk of cardiovascular disease.

Benefits: We found many systematic reviews. One review (search date 1997, 17 RCTs with morbidity and mortality outcomes, duration > 1 year, 37 000 people) found that antihypertensive drugs versus placebo produced variable reductions of systolic/diastolic blood pressure that averaged around 12–16/5–10 mm Hg. [122] It found evidence of benefit in total death rate, cardiovascular death rate, stroke, major coronary events, and congestive cardiac failure, but the absolute results depended on age and the severity of the hypertension (see below). The biggest benefit was seen in those with the highest baseline risk. The trials mainly compared placebo versus diuretics (usually thiazides with the addition of amiloride or triamterene) and versus β blockers (usually atenolol or metoprolol) in a stepped care approach. One systematic review (search date 1999, 8 RCTs, 15 693 people) found that, in people over 60 years old with systolic hypertension, treatment of systolic pressures greater than 160 mm Hg decreased total mortality and fatal and non-fatal cardiovascular events. Absolute benefits were greater in men than women, in people aged over 70, and in those with prior cardiovascular events or wider pulse pressure. The relative hazard rates associated with a 10 mm Hg higher initial systolic blood pressure were 1.26 ($P = 0.0001$) for

Figure 106 - Supporting material (BMJ Clinical Evidence webpage), Isolated Systolic Hypertension & Calcium Channel Blockers. Example shown for illustration. Portion of a display shown with permission.

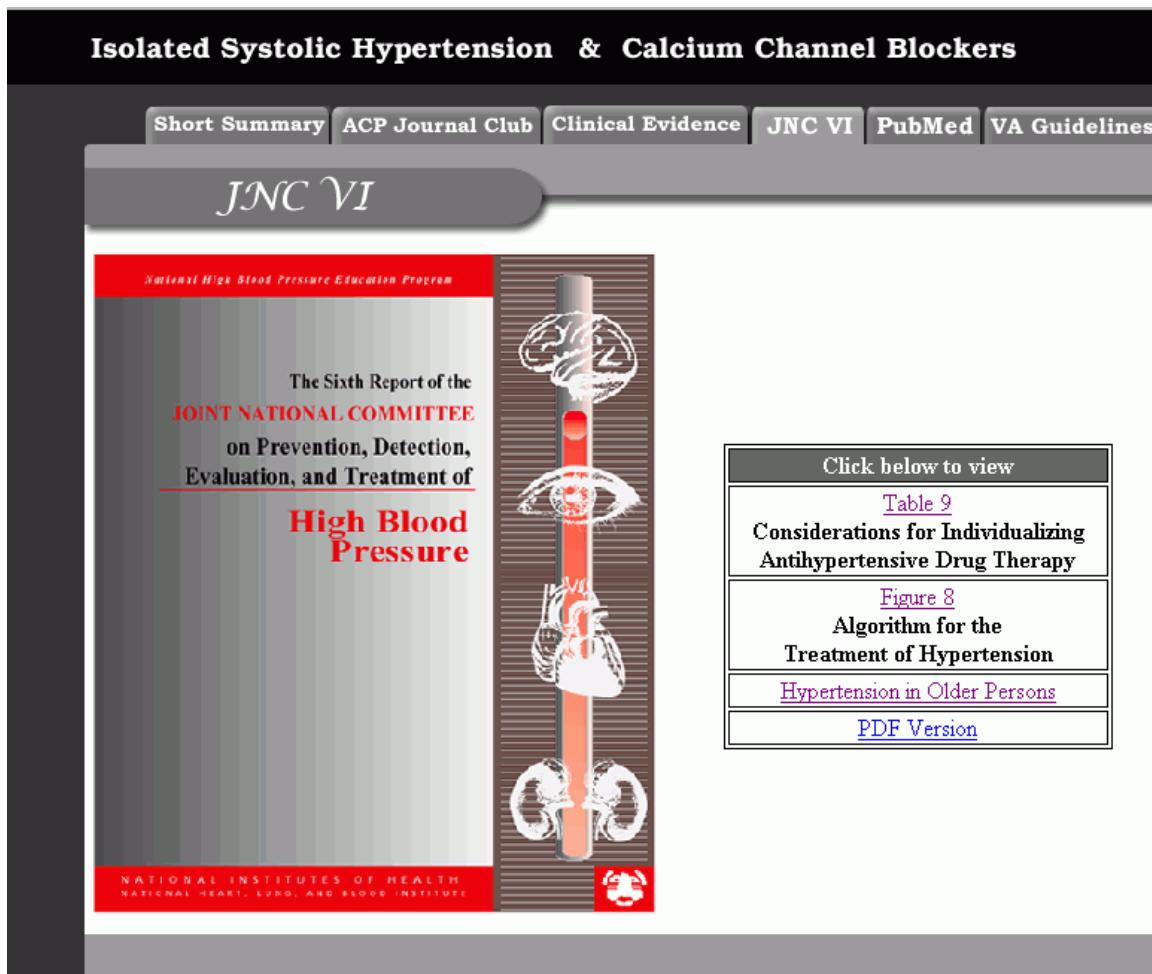


Figure 107 - Supporting material (JNC 6 webpage), Isolated Systolic Hypertension & Calcium Channel Blockers

Isolated Systolic Hypertension & Calcium Channel Blockers

[Short Summary](#) [ACP Journal Club](#) [Clinical Evidence](#) [JNC VI](#) [PubMed](#) [VA Guidelines](#)

PubMed

Select a study below to view abstract

[Staessen 1997](#)

[Staessen 2000](#)

Staessen, J. A., R. Fagard, et al. (1997). "Randomized double-blind comparison of placebo and active treatment for older patients with isolated systolic hypertension. The Systolic Hypertension in Europe (Syst-Eur) Trial Investigators [see comments]." *Lancet* 350(9080): 757-64.

Abstract:

BACKGROUND: Isolated systolic hypertension occurs in about 15% of people aged 60 years or older. In 1989, the European Working Party on High Blood Pressure in the Elderly investigated whether active treatment could reduce cardiovascular complications of isolated systolic hypertension. Fatal and non-fatal stroke combined was the primary endpoint. METHODS: All patients (> 60 years) were initially started on masked placebo. At three run-in visits 1 month apart, their average sitting systolic blood pressure was 160-219 mm Hg with a diastolic blood pressure lower than 95 mm Hg. After stratification for centre, sex, and previous cardiovascular complications, 4695 patients were randomly assigned to nitrindipine 10-40 mg daily, with the possible addition of enalapril 5-20 mg daily and hydrochlorothiazide 12.5-25.0 mg daily, or matching placebos. Patients withdrawing from double-blind treatment were still followed up. We compared occurrence of major endpoints by intention to treat. FINDINGS: At a median of 2 years' follow-up, sitting systolic and diastolic blood pressures had fallen by 13 mm Hg and 2 mm Hg in the placebo group ($n = 2297$) and by 23 mm Hg and 7 mm Hg in the active treatment group ($n = 2398$). The between-group differences were systolic 10.1 mm Hg (95% CI 8.8-11.4) and diastolic, 4.5 mm Hg (3.9-5.1). Active treatment reduced the total rate of stroke from 13.7 to 7.9 endpoints per 1000 patient-years (42% reduction; $p = 0.003$). Non-fatal stroke decreased by 44% ($p = 0.007$). In the active treatment group, all fatal and non-fatal cardiac endpoints, including sudden death, declined by 26% ($p = 0.03$). Non-fatal cardiac endpoints decreased by 33% ($p = 0.03$) and all fatal and non-fatal cardiovascular endpoints by 31% ($p < 0.001$). Cardiovascular mortality was slightly lower on active treatment (-27%, $p = 0.07$), but all-cause mortality was not influenced (-14%; $p = 0.22$).

Figure 108 - Supporting material (PubMed webpage), Isolated Systolic Hypertension & Calcium Channel Blockers

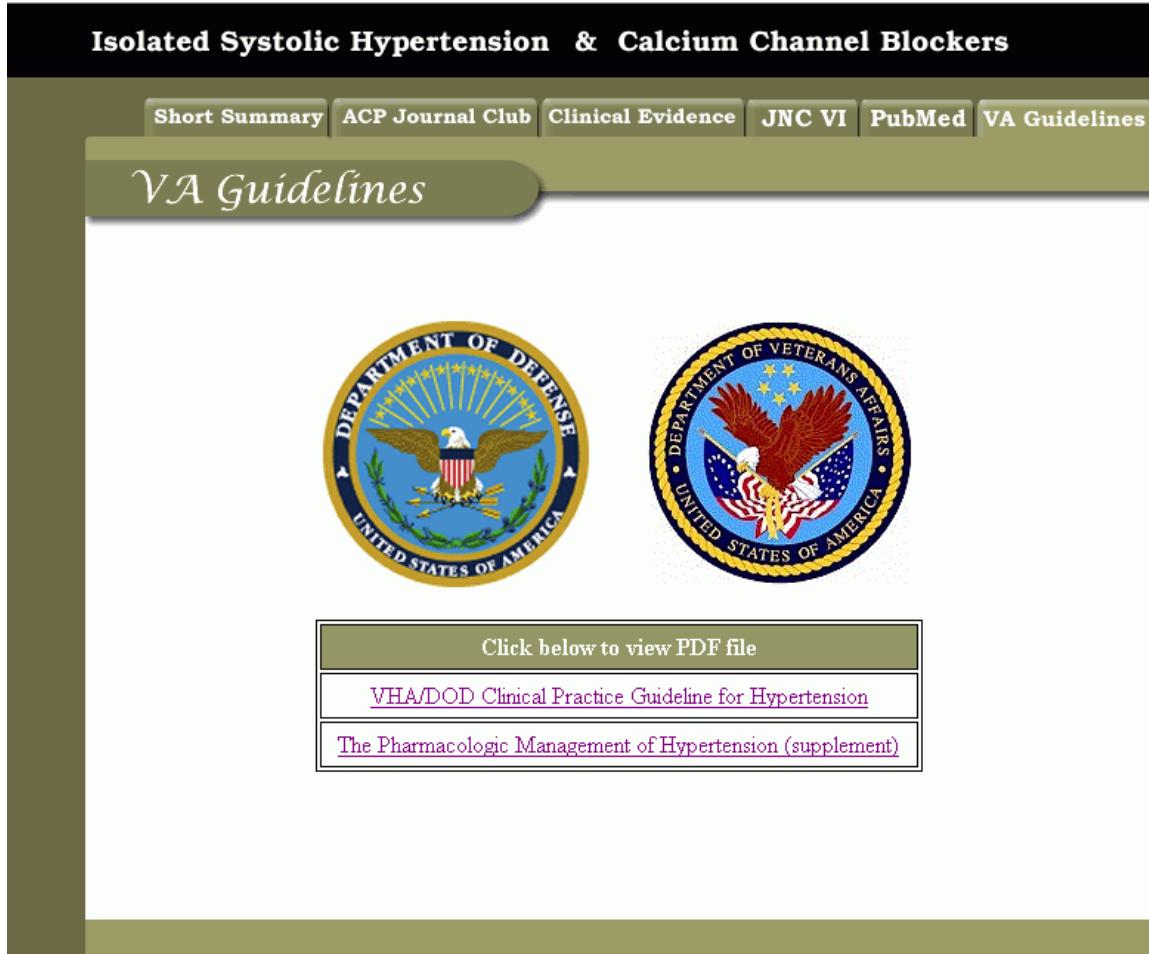


Figure 109 - Supporting material (VA *Guidelines* webpage), Isolated Systolic Hypertension & Calcium Channel Blockers

These webpages are stored in ATHENADSS\doc\ATHENA\EvidenceDisplayATHENA\ in a folder specific to the medical condition/drug combination. Subsection IV.5 explains how to update the webpages.

IV.4.2. Lifestyle, Glossary, and Adherence Tabs

The ATHENA-CDS hypertension advisory contains several tabs. In some tabs, such as Recommendations, Precautions, Assumptions, and BP-Prescription Graphs, the content is specific to each patient. Other tabs—Lifestyle, Glossary and Adherence—are HTML pages with fixed information:

- *Lifestyle tab* – contains information about the impact of lifestyle changes on blood pressure.

- *Glossary tab* – contains definitions for terms used in the advisory.
- *Adherence tab* – contains suggestions for improving patient adherence to antihypertensive medication extracted from the VA *Hypertension Guidelines*.

These HTML pages are located in W:\ATHENA_w\ATHENADSS\doc\ATHENA. To learn how to update the tab pages, go to Subsection IV.5 on updates.

IV.5. Updating Knowledge Sources in ATHENA-CDS SYSTEM

IV.5.1. Update the Rules Document

The Rules Book is maintained and updated by the VA ATHENA-CDS Project Team. The Rules Book is a description of the content of the Knowledge Base, with explicit mention of portions of the Knowledge Base that deviate from JNC 6. The VA ATHENA-CDS Project Team maintains two versions of the ATHENA-CDS SYSTEM: the production version (installed in the ATHENADSS directory) and the development version (installed in the DEVATHENADSS directory). The Rules Book and appendices that correlate to the Knowledge Base being used in clinics are kept in the folder ATHENADSS/doc, and the files that contain the Knowledge Base are in the folder ATHENADSS/domain_model.

The DEVATHENADSS/doc folder (developmental version of ATHENA-CDS SYSTEM) contains the most current version of the Knowledge Base and the Rules Book, with updates as determined by the VA ATHENA-CDS Project Team and consultants. These updates are the responsibility of the VA ATHENA-CDS Project Team. Each update is tested when entered into the Knowledge Base, then the update is made to the rules document. Transferring the developmental version of the Knowledge Base and rules to the live system requires extensive testing (see testing procedure for transfer of updated Knowledge Base. At this point, further, extensive testing is required when transferring updates from the developmental version to the live system (see Subsection IV.5.8). The Rules Book and appendices are archived at the VA, and their filenames have a suffix reflecting the date of change.

IV.5.2. Update the Knowledge Base in Protégé

IV.5.2.1. Add a New Drug

Each drug used in the CDS SYSTEM reasoning needs to be added to the appropriate drug class of the Medications_Class in the Knowledge Base class hierarchy (see Figure 110). A drug is added by following the steps described below:

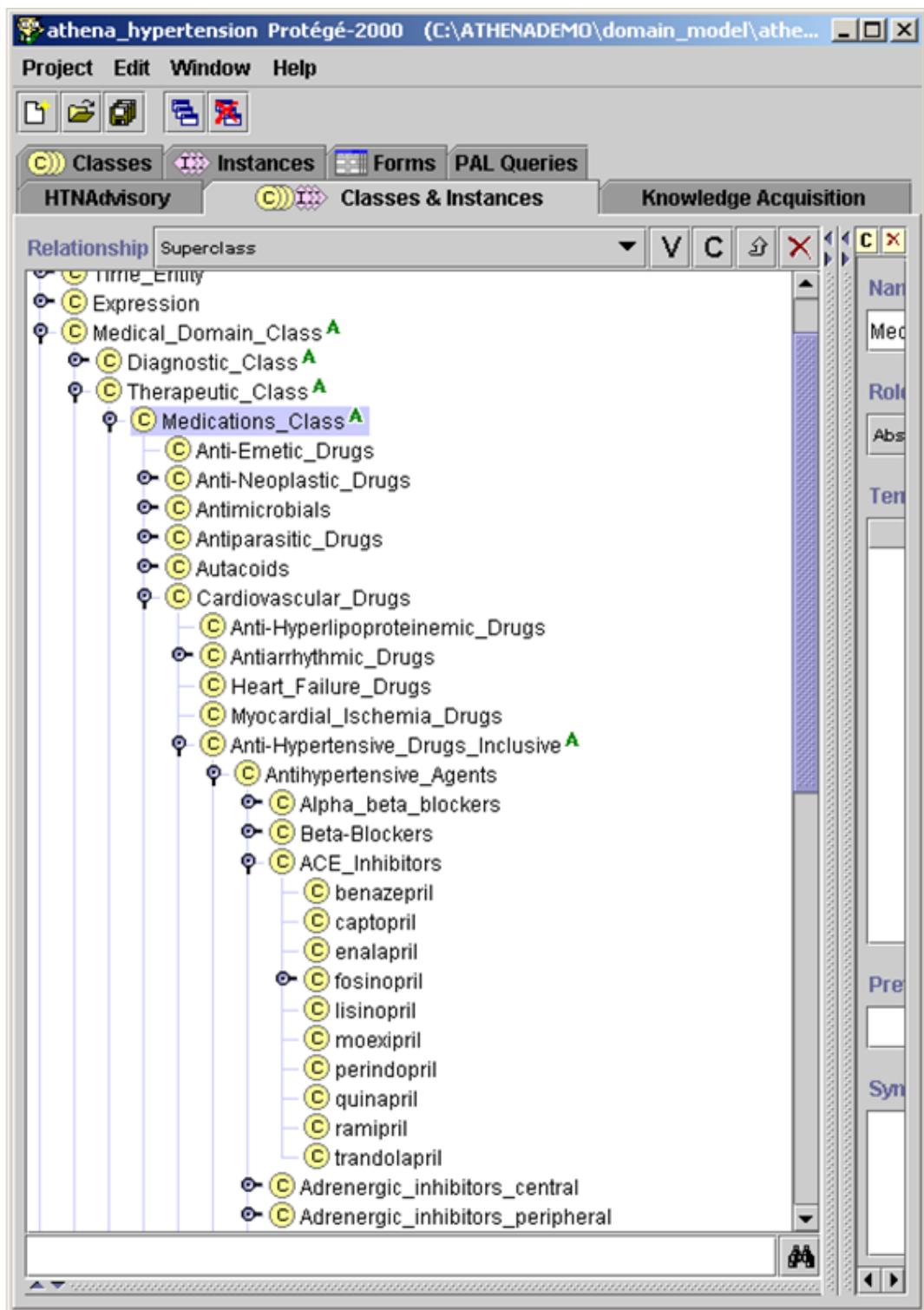


Figure 110 - Medications_Class in Medical_Domain_Class, within the class hierarchy

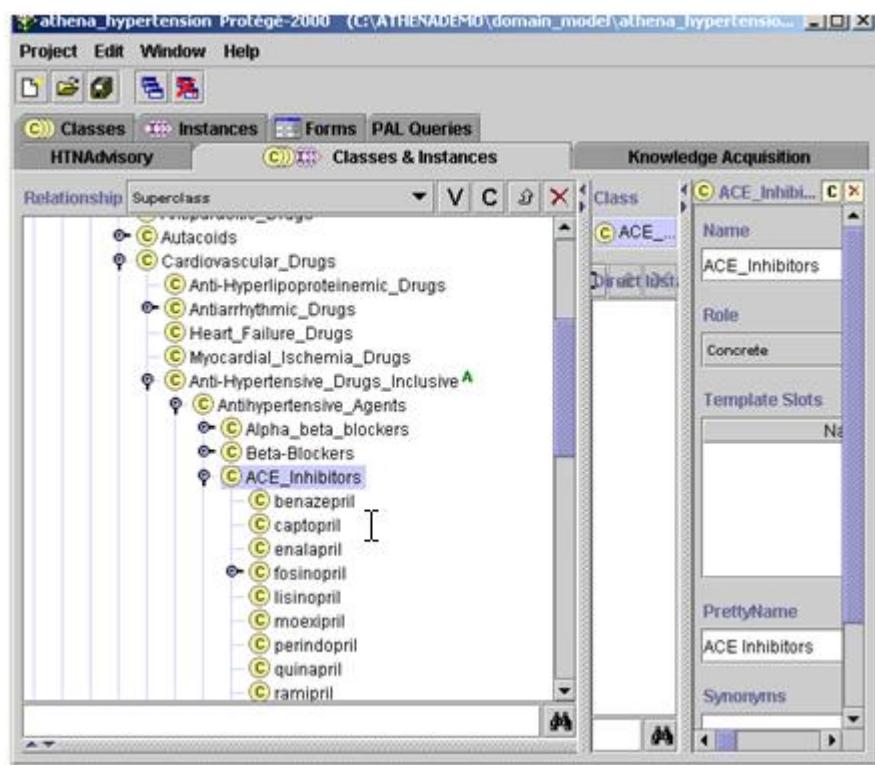


Figure 111 - Selecting the appropriate drug class

1. Select *Create subclass using metaclass...*. Figure 111 shows selection of a drug class: ACE Inhibitors. Figure 112 shows addition of a new drug to the drug class. (In Figure 114, test_drug is added to the ACE_Inhibitors class.)

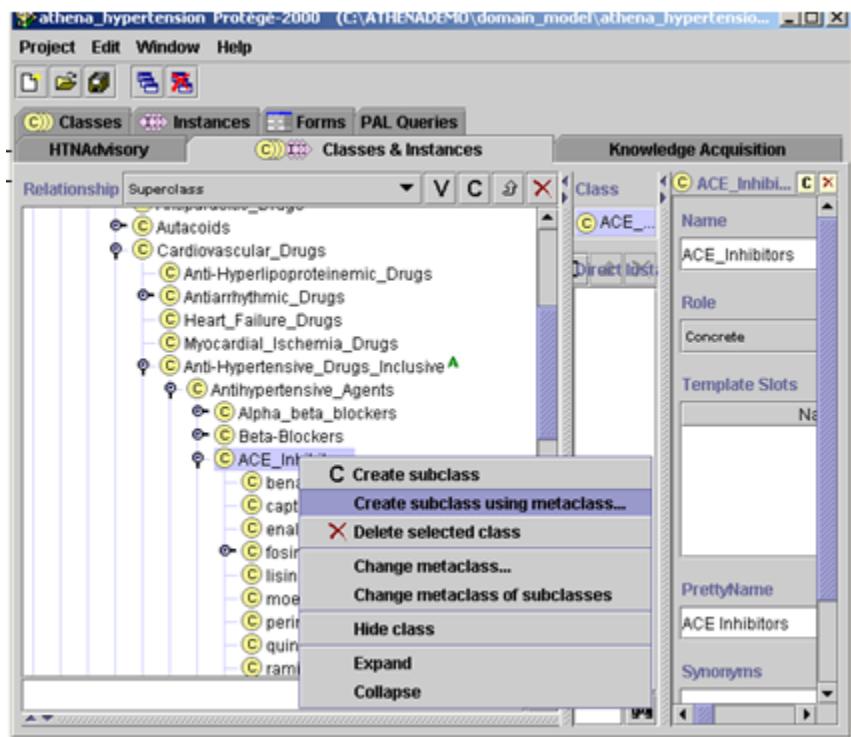


Figure 112 - Adding a new drug to the Medications_Class category, ACE_Inhibitors

2. Select Medications_Metaclass and click OK (Figure 113).

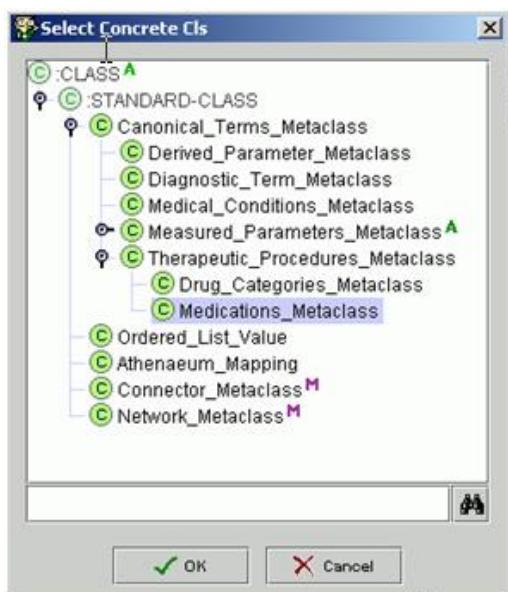


Figure 113 - Selecting Medications_Metaclass in order to add a new instance of a drug within Medications_Class

3. Add the drug name to the Name slot (Figure 114). Note that, for this drug to be identified and considered in the algorithm of the Knowledge Base, letter case and spelling must be identical to those in the drug mapping table ('DRUGS') in the SQL database (ATHENA and ATHENEON). Ignore the other slots.

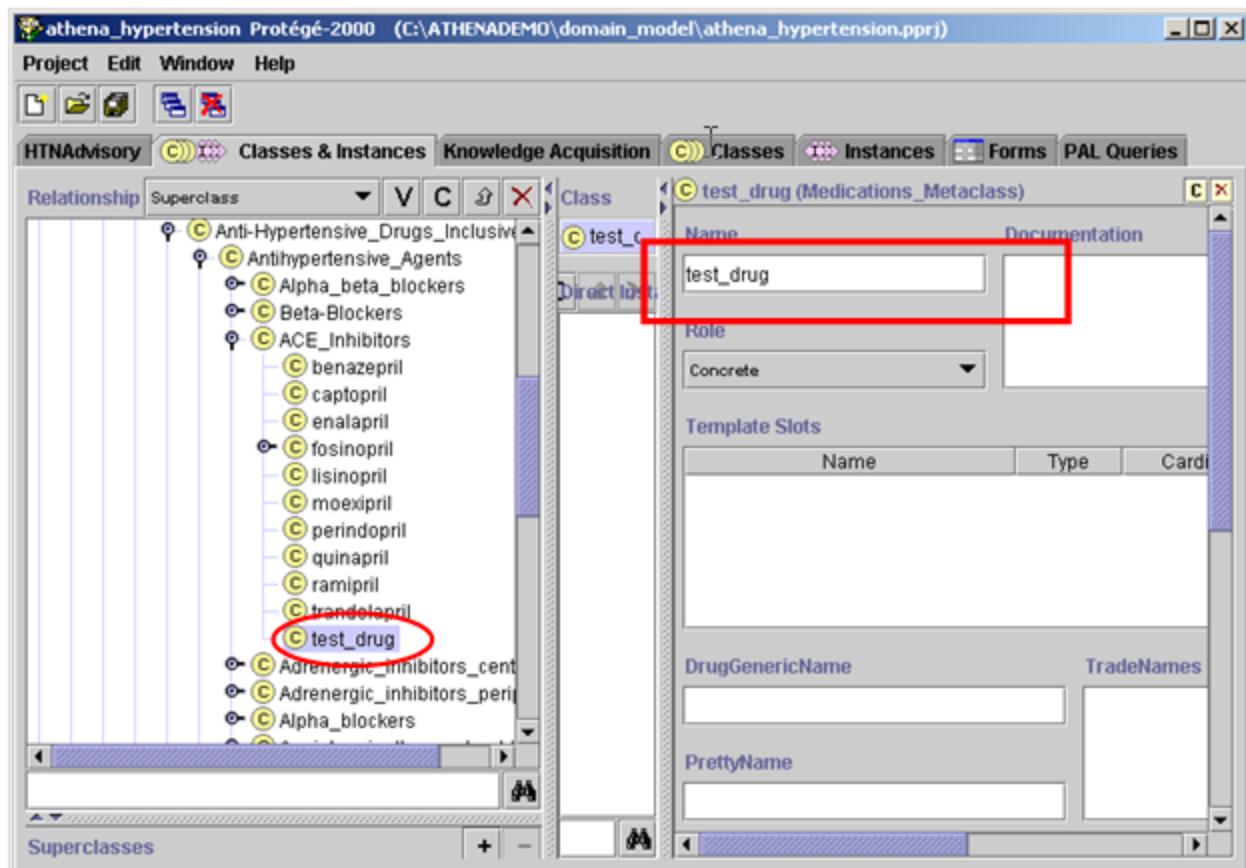


Figure 114 - Adding the new drug, test_drug, to the ACE_Inhibitors drug class

In specific situations, the drug update may require updating other areas of the Knowledge Base. For instance, it may be necessary to make changes so that a drug will be displayed as preferred or so that it is possible to recommend an increased dosage of a drug. These two procedures are outlined in the following subsections.

IV.5.2.2. Displaying a Drug as Preferred in the Advisory

An ATHENA-CDS hypertension advisory recommends a drug class if there are compelling or relative indications for it. For example, it will recommend ACE Inhibitors for patients who have

suffered heart failure. Within the drug class, a formulary preferred drug is displayed in parentheses beside the drug class being recommended. In Figure 115, lisinopril is indicated as a preferred drug: *Add ACE Inhibitor (lisinopril)*.

Several drugs can be designated as preferred within a drug class for specific scenarios, based on rule-in criteria. Multiple preferred drugs matching rule-in/-out criteria can be displayed.

Hypertension Guideline

Advisory **Advisory - HTML** **Eligibility**

Patient SSN Name
 Most Recent BP in Database Date
 ENTER Today's Decision BP Date Update Advisory

Guideline Goal: SBP < 130 and DBP < 85 [presence of diabetes, heart failure or renal insufficiency]
BP apparently NOT UNDER CONTROL, based on most recent available BP.
 (Enter "Today's Decision Blood Pressure" and press "Update Advisory" for new recommendations.) 

Recommendations **Precautions** **Assumptions** **Lifestyle** **Adherence** **Glossary** **BP-Prescription Graphs**

Recommend ADDING antihypertensive medication: BP ELEVATED based on most recent available BP.

Consider one of the following therapeutic possibilities:

Compelling Indication	Relative Indication	Strong Contraindication	Relative Contraindication	Adverse Events
Add DHP Calcium Channel Blocker (felodipine, nifedipine)	<input type="button" value="Info"/>	Isolated Systolic Hypertension	<input type="button" value="Feedback"/>	
Add ACE Inhibitors(lisinopril)	<input type="button" value="Info"/>	Renal Insufficiency	<input type="button" value="Feedback"/>	
Add Cardioselective Beta Blocker (atenolol)	<input type="button" value="Info"/>	Coronary Artery Disease	<input type="button" value="Feedback"/>	
		Obstructive Pulmonary Disease	<input type="button" value="Feedback"/>	

Your comments for the Guidelines Team (optional and welcome!)

Do not display Advisory for this clinic visit again.

Complete clinical information may not be available through the computer system. Please use all the information that you have about the patient together with your clinical judgment to decide on the best therapy for this patient.

Figure 115 - Preferred drugs for the drug classes DHP Calcium Channel Blocker, ACE Inhibitors, and Cardioselective Beta Blocker

To make test_drug a preferred drug in the ACE Inhibitors drug class:

1. Go to the Knowledge Acquisition tab and select the ACE Inhibitor class under Drug Classes (Figure 116).

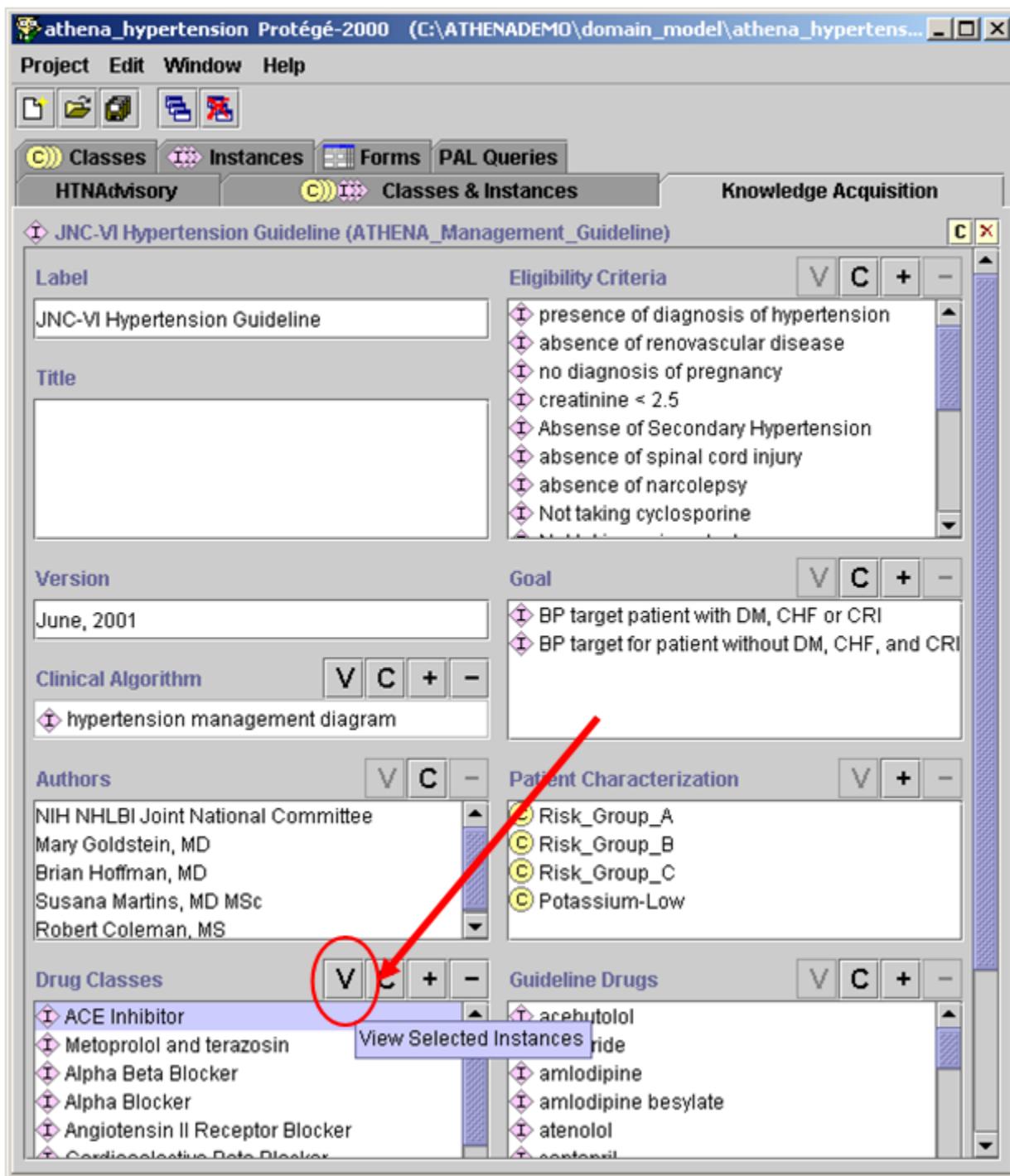


Figure 116 - View ACE Inhibitor drug class

- Click on **V** to view the ACE Inhibitor form shown in Figure 116.
- Go to the slot, Formulary Preferred Drug in Class (Figure 118), and add the preferred drug by clicking on the **C** button to create a guideline drug instance or on **+** to add an existing one. When creating a new guideline drug instance, the Guideline_Drug template will appear (Figure 118). Fill in the following fields:

The screenshot shows a software interface for managing drug classes. At the top, there's a title bar with the text "ACE Inhibitor (Drug_Usage)". Below the title bar, there are several tabs and sections. One tab is active, showing "Proteinuria equals 1+ or higher". Other tabs include "Renovascular_disease", "K>5.5 and absence ACE", and "Presence of K sparing diuretics". There are also sections for "Drug Partners", "Drug Partners To Avoid", "Side Effects", and "Complication Factor". A prominent section at the bottom left is "Formulary Preferred Drug In Class", which contains the entry "lisinopril". On the right side of the interface, there are buttons for "V" (View), "C" (Create), and "+" (Add Existing). The overall layout is a grid-based form editor.

Figure 117 Partial view of Drug Usage ACE Inhibitor class

- Label** – Identify the drug.
- Starting Dose** – Type in a starting dose
- Generic Drug** – Add from the class hierarchy.
- Max Recommended Dose Level** – The Guideline Interpreter will use the lower limit of this dose range when recommending a dose increase.
- Dose Level Ranges** – Supply these as appropriate. This field is currently used only for dose increase. If the dose is below the lower limit of the recommended dose range for High_Dose, then ATHENA-CDS will recommend a dose increase. For example, one can create dose ranges for low dose, medium dose, and high dose. If the Max Recommended

Dose Level is defined as the high range, and if the current dose is less than the lower limit of high dose range, ATHENA-CDS will recommend a dose increase.

test_drug (Guideline_Drug)

Duration Constraint	Collateral Actions												
<input type="text"/>	<input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/>												
Label	Dose Level Ranges												
<input type="text"/> test_drug	<input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/>												
Starting Dose	<table border="1"><tr><th>abstract_value</th><th>lower_limit</th><th>upper_limit</th></tr><tr><td>Low_Dose</td><td>0.0</td><td>10.0</td></tr><tr><td>Medium_Dose</td><td>11.0</td><td>30.0</td></tr><tr><td>High_Dose</td><td>31.0</td><td>60.0</td></tr></table>	abstract_value	lower_limit	upper_limit	Low_Dose	0.0	10.0	Medium_Dose	11.0	30.0	High_Dose	31.0	60.0
abstract_value	lower_limit	upper_limit											
Low_Dose	0.0	10.0											
Medium_Dose	11.0	30.0											
High_Dose	31.0	60.0											
Dose Strength Unit	<input type="button"/> <input type="button"/> <input type="button"/>												
<input checked="" type="radio"/> mg <input checked="" type="radio"/>													
Generic Drug	<input type="button"/> <input type="button"/> <input type="button"/>												
<input checked="" type="radio"/> test_drug <input checked="" type="radio"/>													
Max Recommended Dose Level	<input type="button"/> <input type="button"/> <input type="button"/>												
<input checked="" type="radio"/> High_Dose <input checked="" type="radio"/>													
Drug Usage	<input type="button"/> <input type="button"/> <input type="button"/>												
<input type="text"/>													
Rule In	<input type="button"/> <input type="button"/> <input type="button"/>												
<input type="text"/>													
Rule Out	<input type="button"/> <input type="button"/> <input type="button"/>												
<input type="text"/>													
Prescribable Items	<input type="button"/> <input type="button"/> <input type="button"/>												
<input type="text"/>													
Reference	<input type="button"/> <input type="button"/> <input type="button"/>												
<input type="text"/>													

Figure 118 - Guideline_Drug template

The label of test_drug can now be seen in the Formulary Preferred Drug in Class slot (Figure 119).

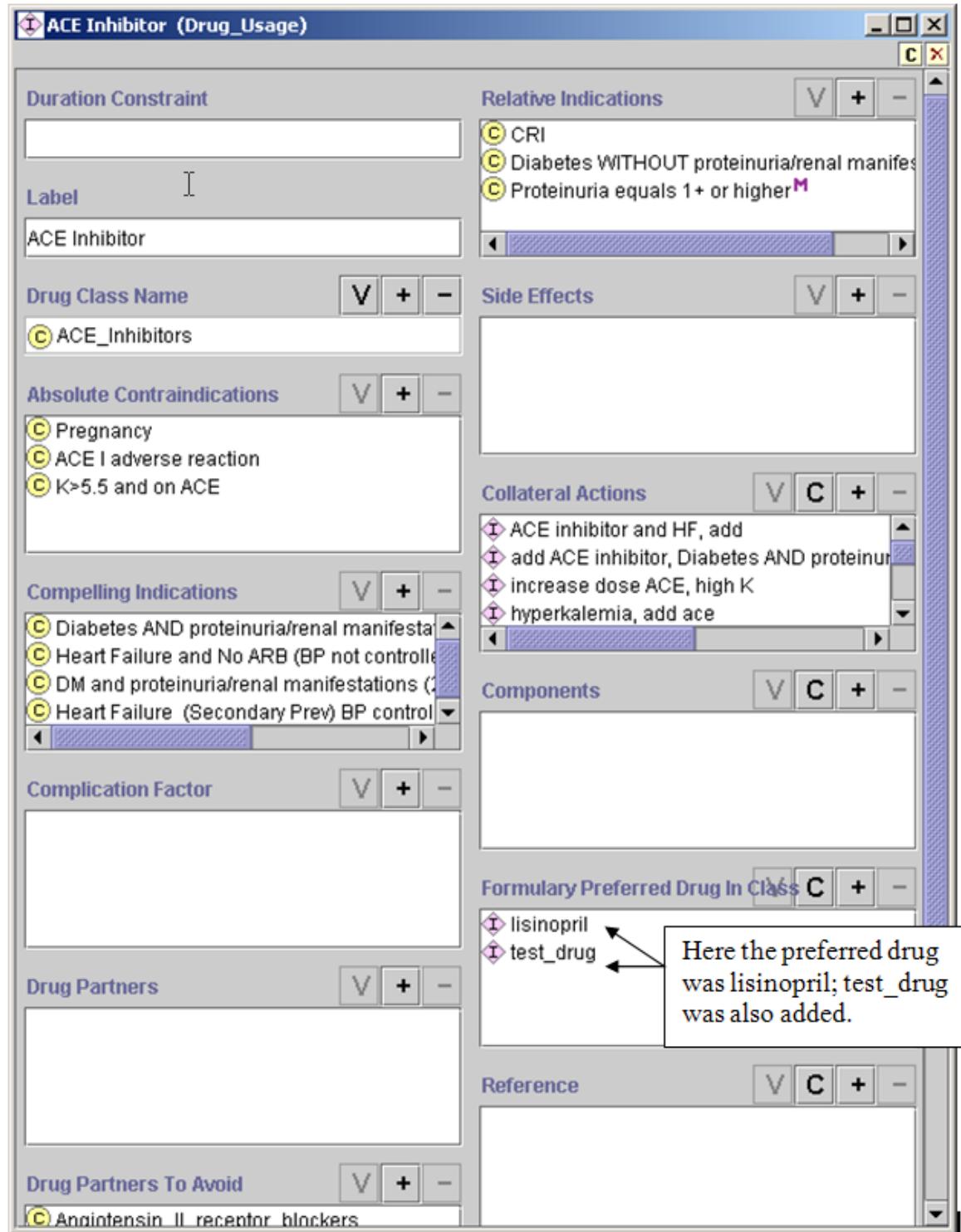


Figure 119 - Adding a new preferred drug to the drug class ACE Inhibitor

IV.5.2.3. Enabling Dose Increase Recommendations for a New Drug

For ATHENA-CDS to recommend drug dose increases, it is necessary first to create an instance of Guideline_Drug for the specific drug and define the dose ranges for low, medium, and high doses:

1. Go to the Knowledge Acquisition tab and find the Guideline Drugs slot (see Figure 120).

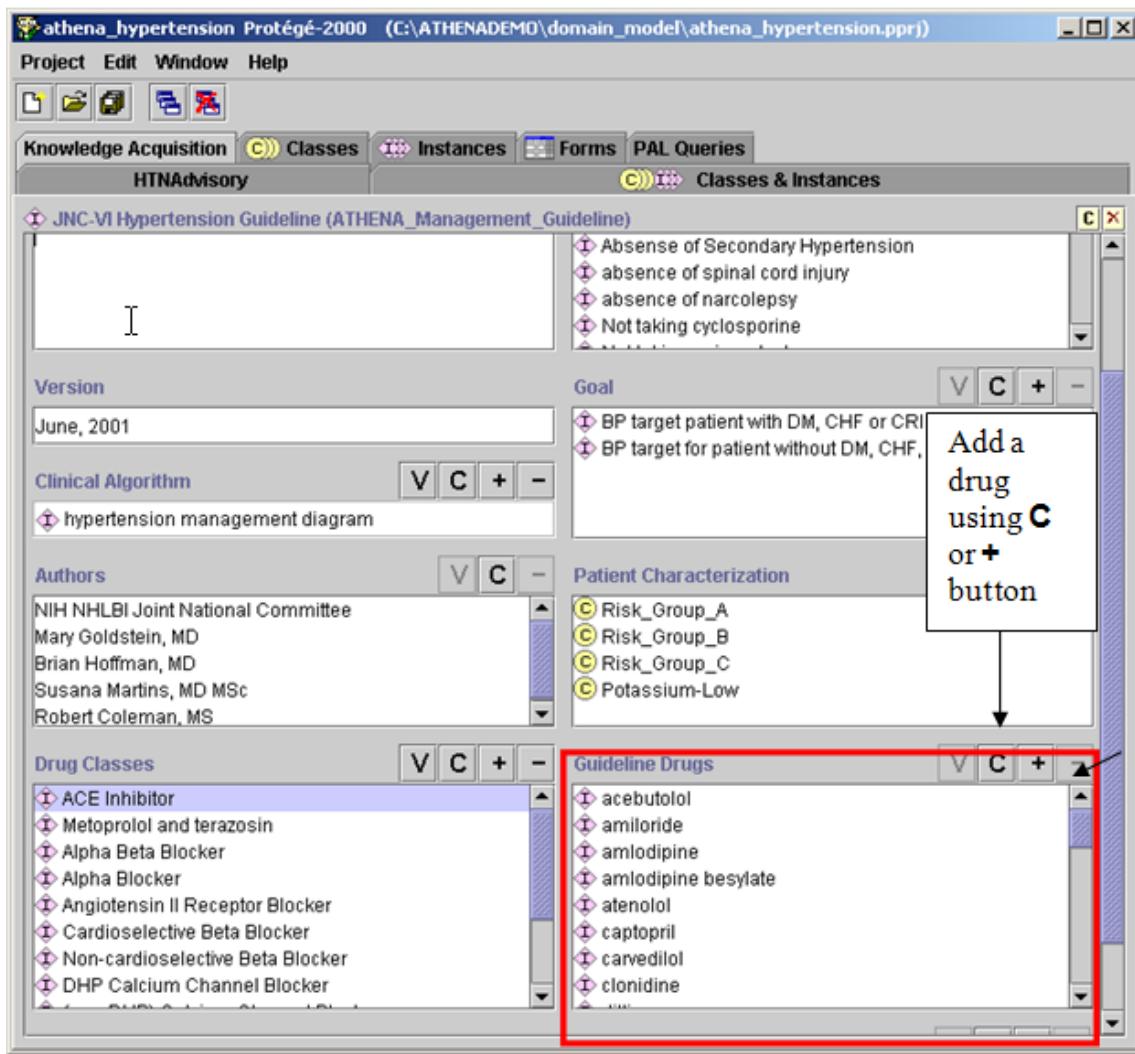


Figure 120 - Knowledge Acquisition tab: Guideline Drugs

2. Click on the **C** button in the Guideline Drugs slot to create a new Guideline Drugs instance (described in Subsection IV.5.2.2, #3), or click on **+** to add an existing one.

In Figure 121, test_drug has been introduced to the Guideline Drugs slot of the Knowledge Acquisition tab.

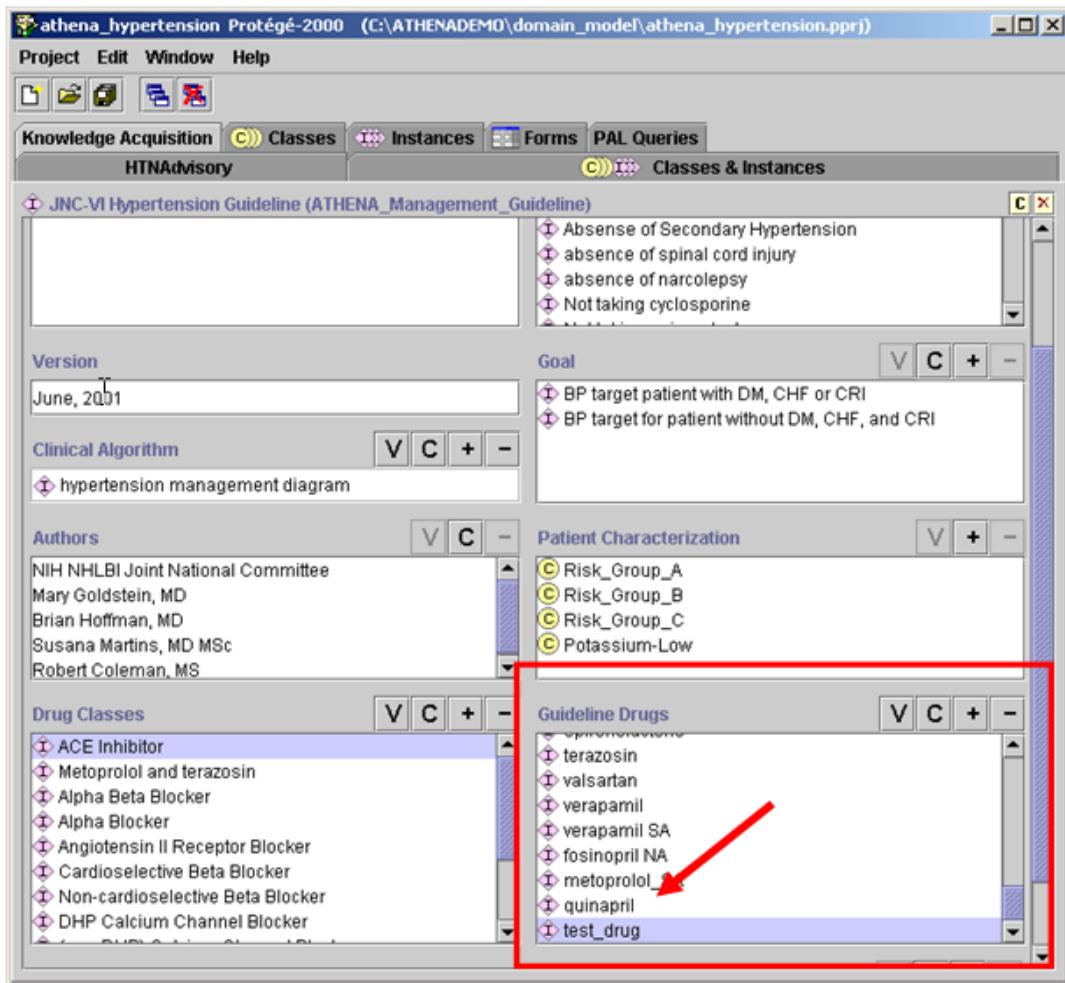


Figure 121 - Test_drug added to Guideline Drugs slot in the Knowledge Acquisition tab

IV.5.2.4. Adding a Compelling Indication to a Drug Class

The JNC6 guideline refers to “compelling indications”, that is, medical conditions that are a strong indication for use of particular drugs. To model JNC6, it is necessary to enter the diseases or medical conditions, if any, that are the compelling indications for using the drug class. The steps for adding a compelling indication are similar for adding a strong (or absolute) contraindication, relative contraindication, or relative indication. They are:

1. Go to the Knowledge Acquisition tab in Protégé.

2. Select from Drug Classes and click on **V** to view.

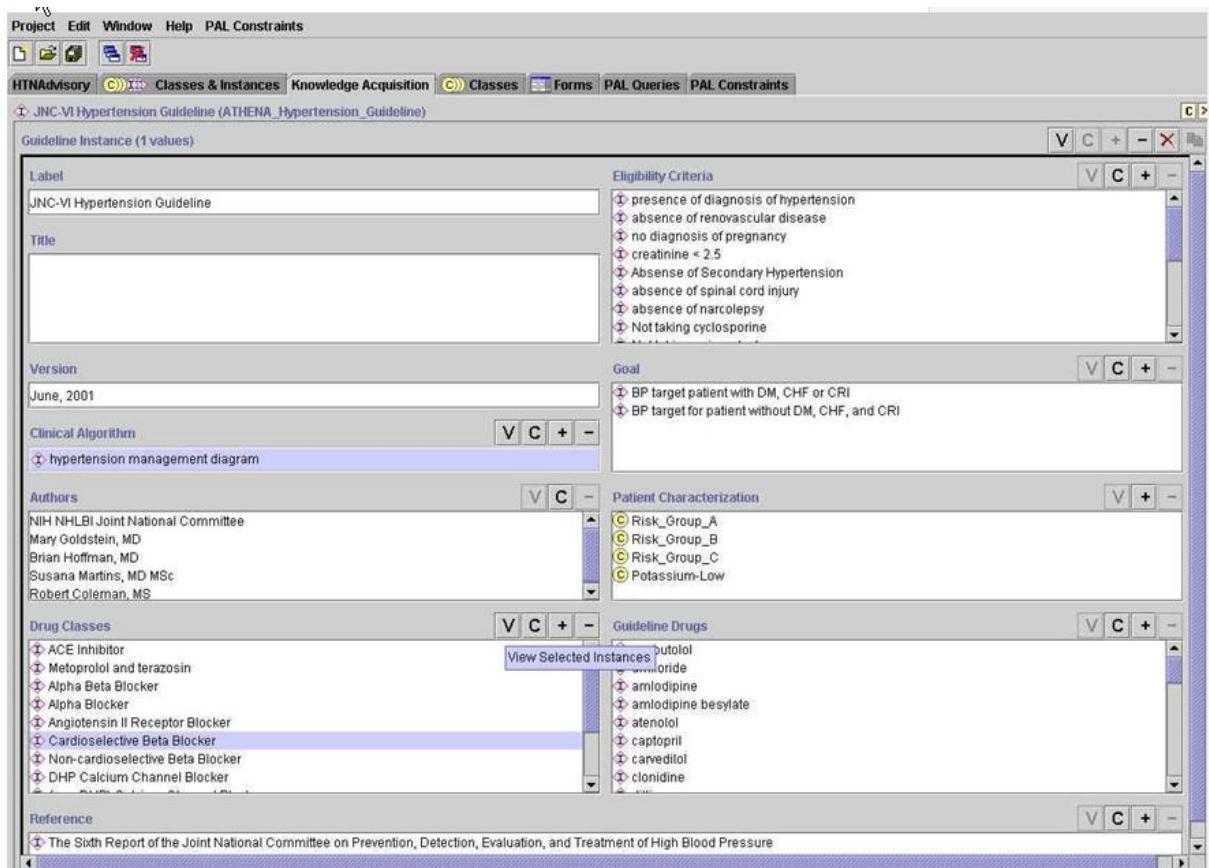


Figure 122 - Knowledge Acquisition tab

The Drug_Usage template will pop up (

Cardioselective Beta Blocker (Drug_Usage)

Duration Constraint [V + -]	Relative Indications C Atrial_Tachycardia C Angina C Atrial_Fibrillation C Hyperthyroidism V + -
Label Cardioselective Beta Blocker	Side Effects V + -
Drug Class Name C Beta-Blockers-Cardioselective V + -	Collateral Actions V C + - alpha blocker monotherapy rec ADD Add Beta blocker and DM Add Beta Blocker to Diltiazem (no MI) Add Beta blocker and PVD V + -
Absolute Contraindications C Bronchospastic_Disease C Heart block without pacemaker A C Unspecified Heart Block and no pacemaker V + -	Components V C + -
Compelling Indications C Hypertensive_Without_Comorbidities_That_Indi C ISH on HCTZ and BP not controlled C MI (secondary prevention) BP controlled C MI (BP not controlled) V + -	Formulary Preferred Drug In Class V C + - atenolol metoprolol V + -
Complication Factor C Heart_Failure V + -	Reference V C + -
Drug Partners C Thiazide_Diuretics M V + -	
Drug Partners To Avoid C guanabenz V + -	

Figure 123). It contains slots for characteristics of the selected drug. Locate the Compelling Indications slot (or the Absolute, Relative Contraindications, or Relative Indications slot).

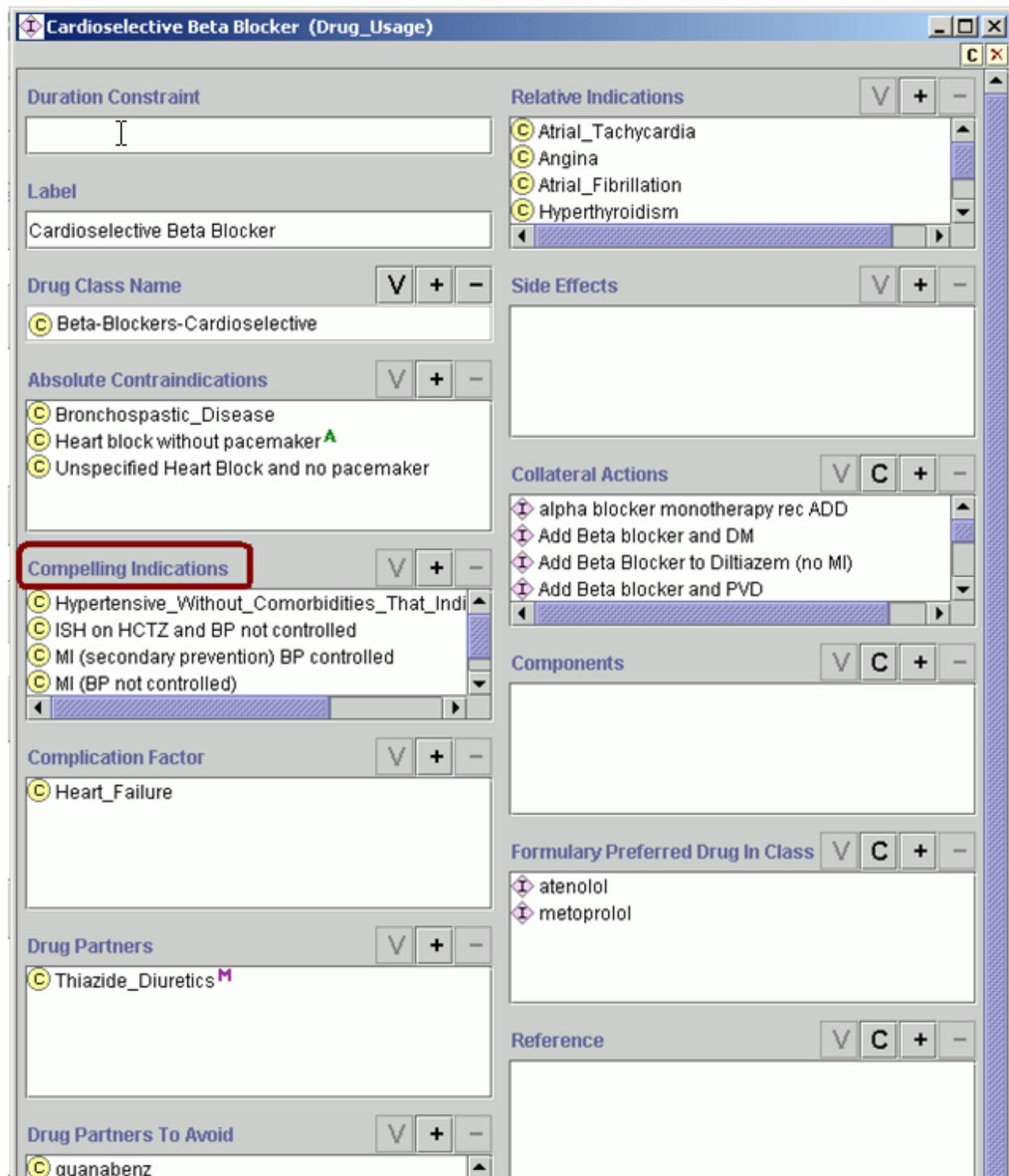


Figure 123 - Drug_Usage template, with Compelling Indications highlighted

3. Click on the **+** to add a compelling indication.

4. Select a medical condition from the class hierarchy, and click OK (Figure 124).

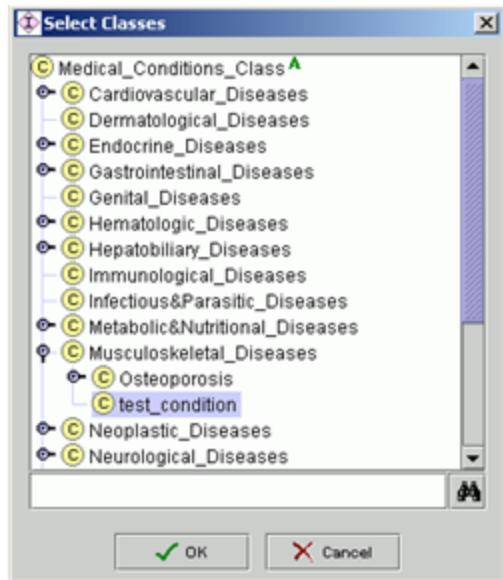


Figure 124 - Selecting the medical condition for a compelling indication

The new compelling indication will be displayed at the bottom of the list for Compelling Indications (see Figure 125).

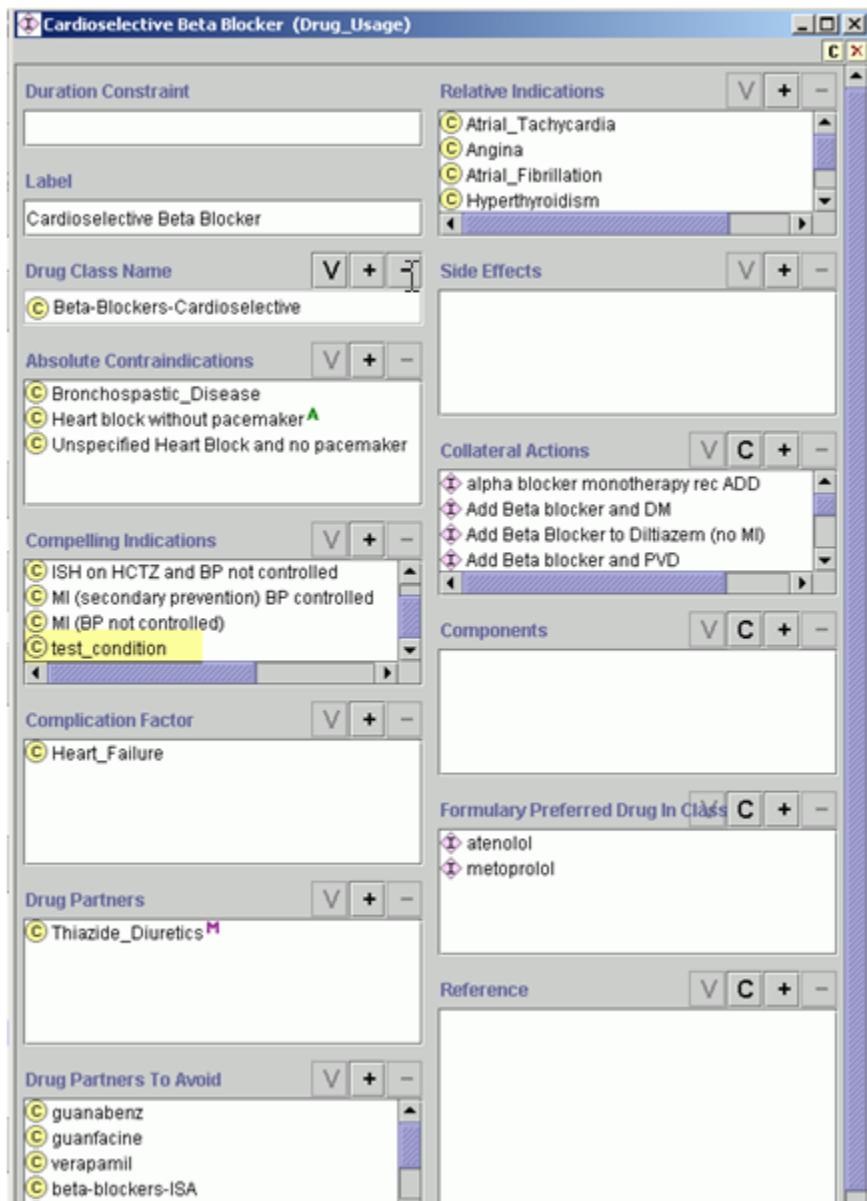


Figure 125 - Test_condition added as a compelling indication for Beta-Blockers-Cardioselective

5. Test the display. Go to the HTNAdvisory tab to create a scenario that will display the just-added condition.
6. Click on the **+** box in the Related Comorbidities section to add test_condition to this patient's comorbidities (see Figure 126 and Figure 127). A window will pop up (Figure 127). In this window click on 'Select' to choose from the class hierarchy (Figure 127).

Patients' Information

Patients List **Summary Sheet**

Patient Name	Patient SSN
Clinic Provider	Clinic
Appointment	

Vitals		
Name	Value	Date
DB_Diastolic_BP	72	
DB_Systolic_BP	142	
Age		
Gender		

ADRs / Allergies	
Substance	Reaction

Related Drugs	
Name	Daily Dosage
lisinopril	5.0
furosemide	20.0

Labs		
Name	Value	Date
Cholesterol	143.2	
HDL_Cholesterol	29.0	
Triglycerides	200.0	
Potassium	4.3	

Related Comorbidities	
Conditions	ICD9 Code(s)
Coronary_Artery_Disease	414.9, 412.
Hypertension	401.1, 401.9
Gout	274.9
Obstructive_Pulmonary_Disease	492.8, 496.

+ - >>

Other Problems	
ICD9 Code	Description
466.0	
173.9	

Update Advisory

Figure 126 – Patients' Information Summary Sheet, adding a new medical condition to the patient state

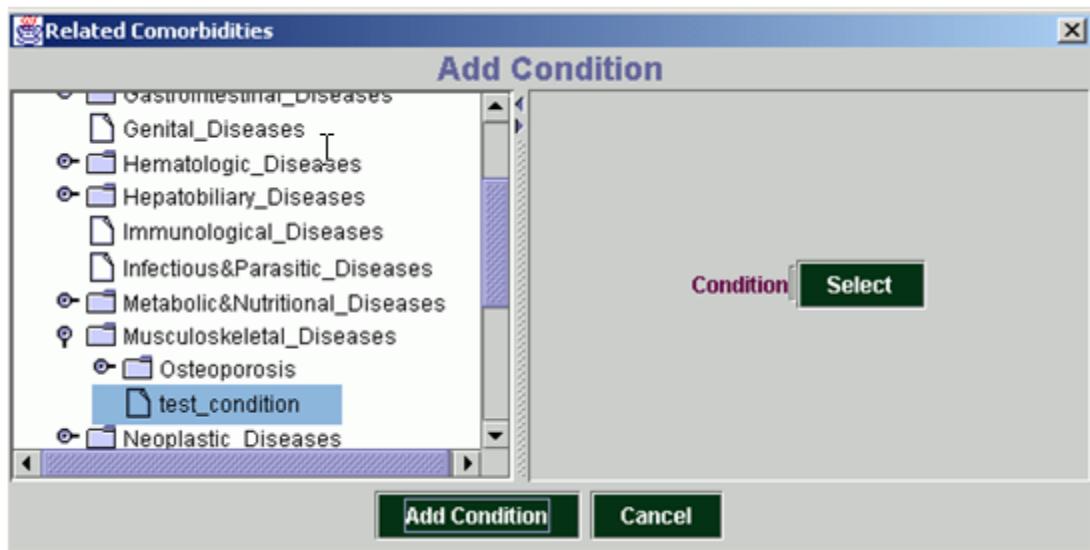


Figure 127 - Selecting test_condition

- Click Update Advisory on the Patients' Information Summary Sheet to register changes (Figure 128).

Figure 128 - Updated advisory showing test_condition added to Related Comorbidities and as a compelling indication for beta blockers

8. Confirm that the desired changes have been made to the Advisory.

In Figure 128, test_condition has been added to the list of Related Comorbidities. The Advisory has been updated, and this patient has test_condition as a compelling indication for adding a Cardioselective Beta Blocker.

IV.5.2.5. Removing Instances by Clicking the - Button

One may enter an instance by mistake in or may want to update it. By clicking the - button (see Figure 129), it is possible to remove criteria from rule-in/-out slots and from drug classes.

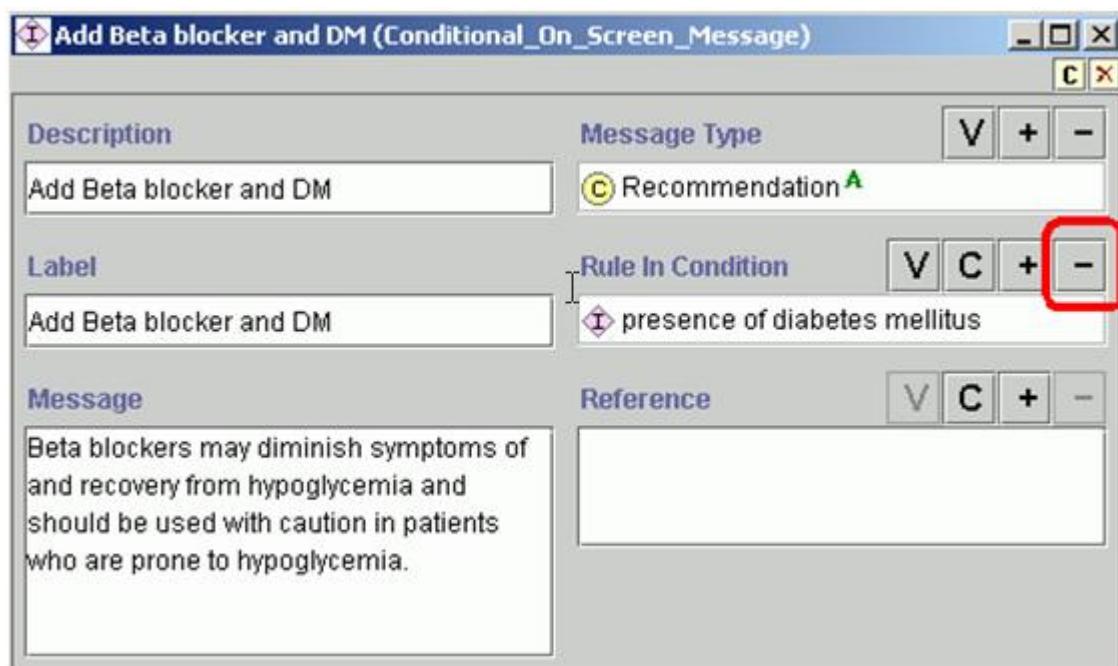


Figure 129 - Removing a rule-in condition from a slot

Do NOT delete criteria from the Protégé knowledge base, as these are the original criteria and may be used in different sections of the Knowledge Base. For example, do NOT select an instance of a criterion class in the Classes & Instances tab and click on the delete icon (X) without first checking whether the criterion is used elsewhere.

To check where specific criteria are used:

1. In the Classes & Instances tab, use the search function to find the criteria. A search can be conducted for instances or classes, depending on which pane is used. Add *—the wildcard symbol—before and after a term to find all criteria that contain that term. For example, typing *ace* in the class panel will find classes labeled ACE inhibitor, race, pacemaker, etc.
2. Click on the upward arrow button (highlighted in Figure 130) to bring up a new pop-up window for showing where criteria are used.
3. Click on the upward button again to display where the criteria are used. Repeat as many times as necessary to see the full extent of the use of the criteria.

The screenshot shows the HTNAdvisory application interface. On the left, there is a tree view of medical domain classes under 'Relationship Superclass'. A search bar at the top contains the text 'diastolic'. Four numbered callouts point to specific UI elements:

- 1**: Points to the search bar containing 'diastolic'.
- 2**: Points to the first 'References to Diastolic_BP' dialog, showing results for 'DB_Diastolic_BP'.
- 3**: Points to the second 'References to DB_Diastolic_BP' dialog, showing results for 'absence of diastolic BP'.
- 4**: Points to the third 'References to diastolic_BP present' dialog, showing results for 'BP measurement present'.

Figure 130 - Searching for where specific criteria are being used in the Knowledge Base

IV.5.3. Adding a New Medical Condition

In this section we will explain how to enter medical conditions defined by ICD9 codes or by other criteria.

Creating a New Disease or Condition Defined by ICD9 Code

1. Select the class in which to insert the new disease or condition. Right click on the selected class, and choose *Create subclass using metaclass...* (see Figure 131).

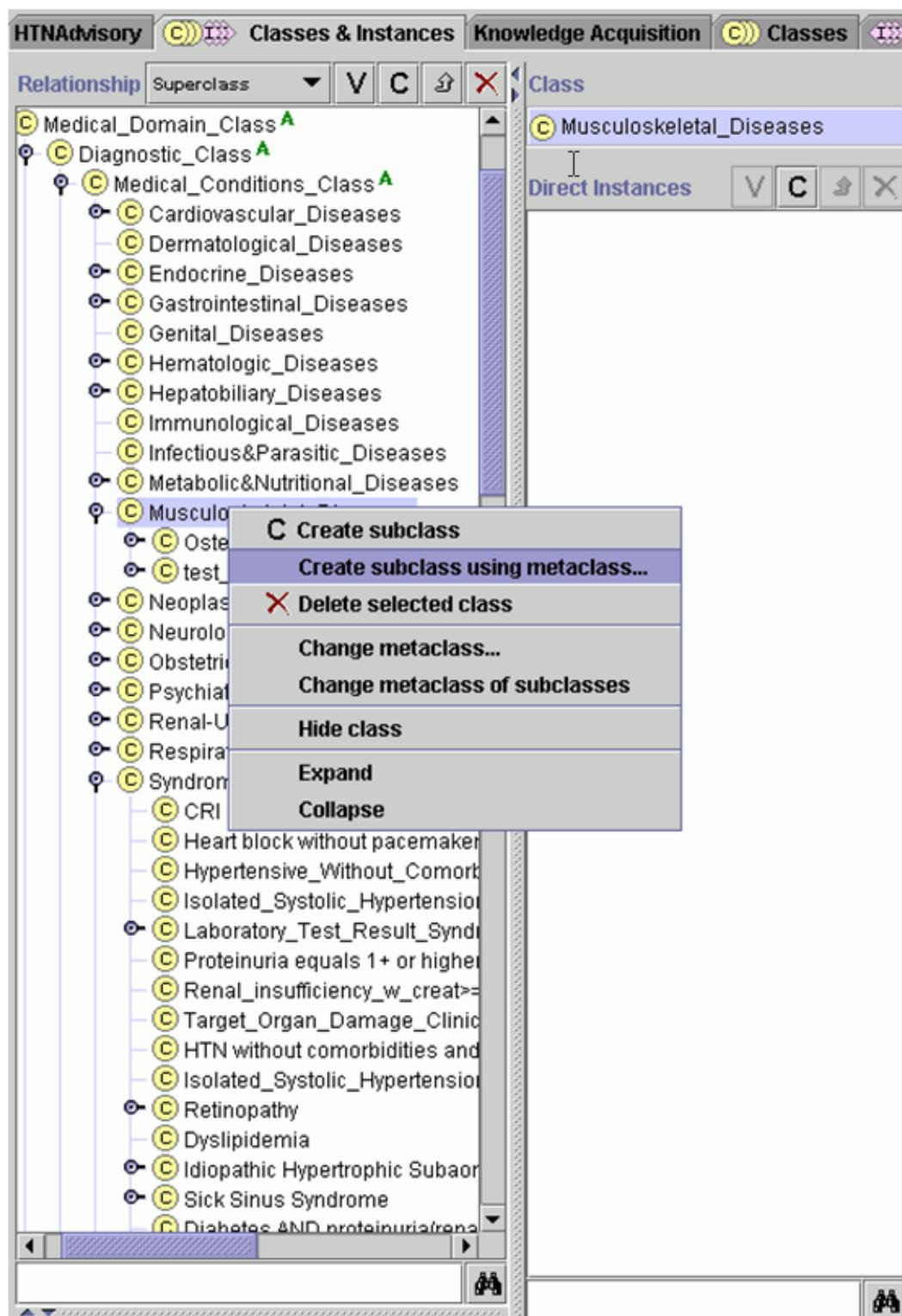


Figure 131 – Creating a subclass using a metaclass

2. Select Medical_Conditions_Metaclass (see Figure 132).

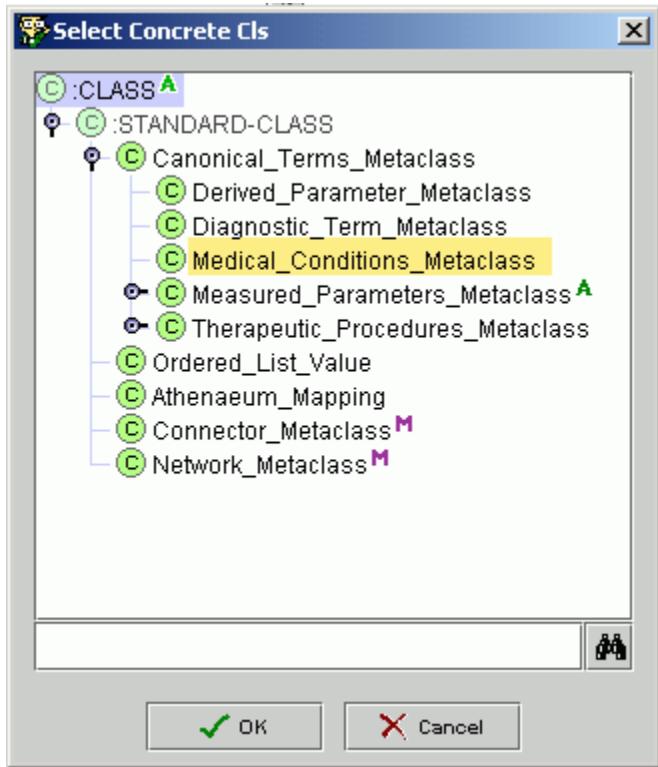


Figure 132 - Selecting the type of metaclass

3. Add the disease or condition to the Name slot. In the example in Figure 133, test_condition is being created in the subclass Musculoskeletal_Diseases.

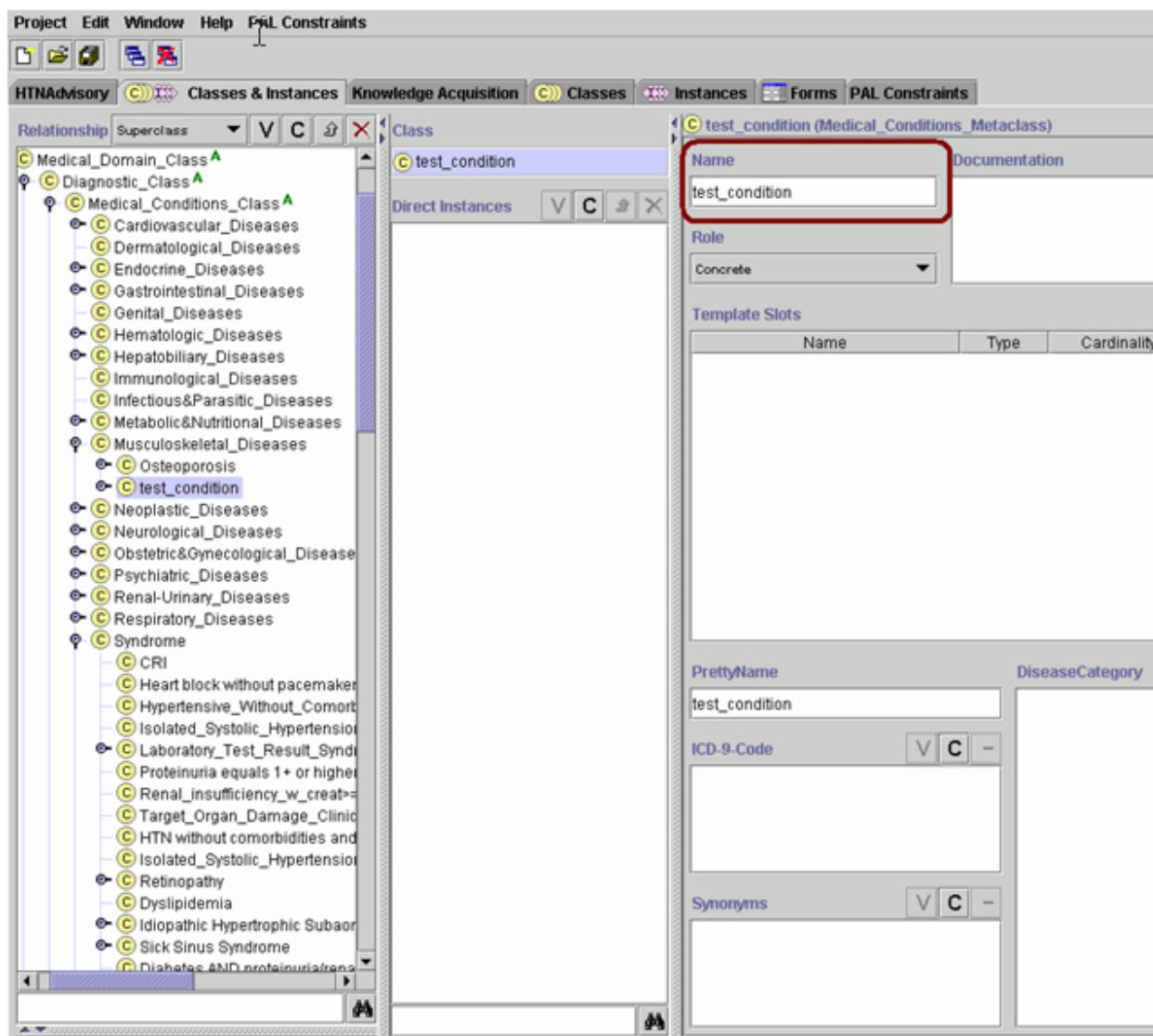


Figure 133 - Adding a new disease or condition to the class hierarchy

To add the ICD9³⁰ code, right click on the selected class and choose *Create subclass using metaclass....* Select *Medical_Conditions_Metaclass*. Add the ICD9 code to the Name slot. In Figure 134, the ICD9 code supplied for *test_condition* is XXXXX.

³⁰ International Classification of Diseases, 9th Edition

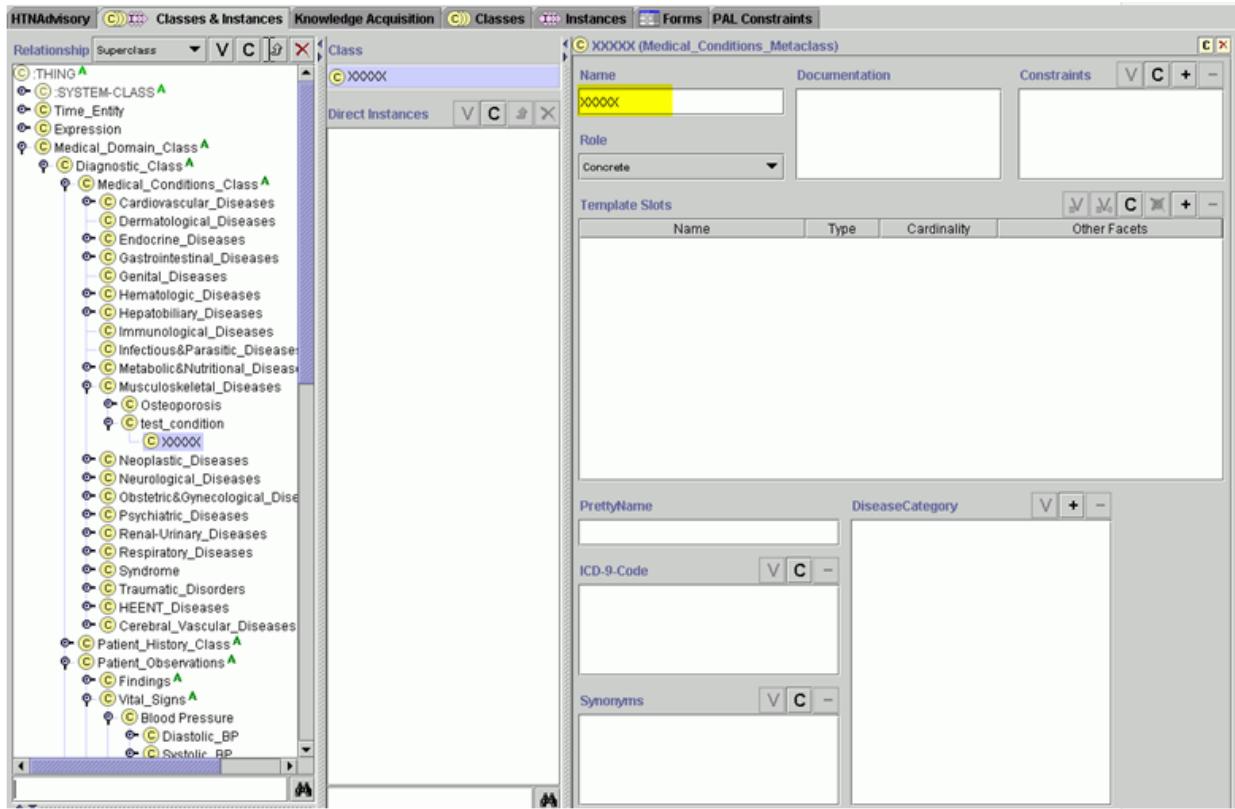


Figure 134 - Creating a condition using Medical_Conditions_Metaclass

Adding a Condition Defined by Criteria

Some conditions can only be defined in Protégé using diagnostic terms metaclasses, for example conditions defined by laboratory values such as renal insufficiency or conditions that have multiple criteria such as heart block and absence of pacemaker. Conditions defined by criteria are located under Syndrome in Medical_Domain_Class. To create a new condition, right click on the “syndrome” class and choose *Create subclass using metaclass....* Select Diagnostic_Term_Metaclass (Figure 135).

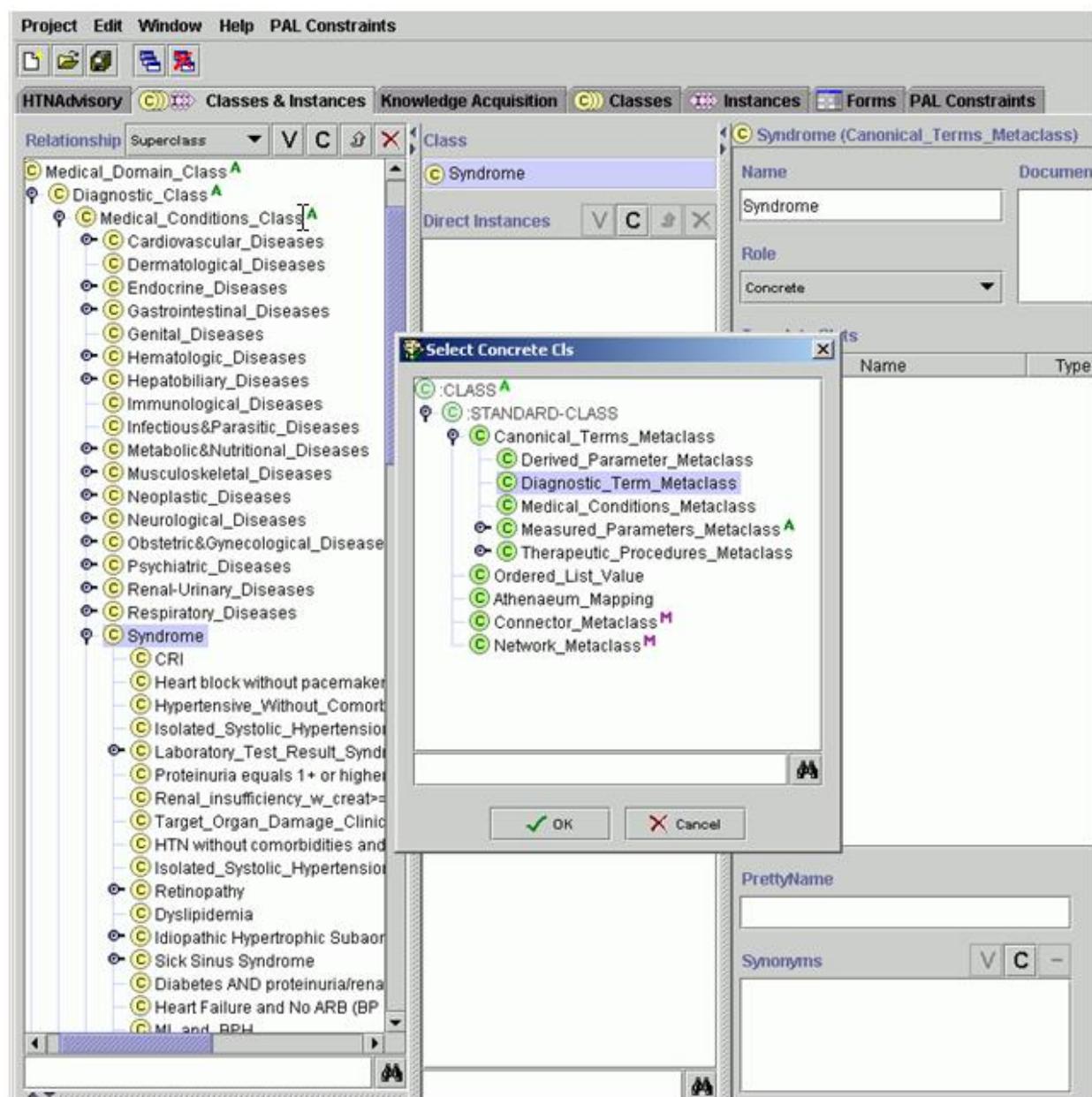


Figure 135 - Select Diagnostic_Term_Metaclass

There is a slot to select or create the diagnostic criteria for the new condition. In Figure 136, the new condition is called test_condition1, for which existing criteria were selected by clicking on the **+** button and scrolling through the options (Figure 137). New criteria can be created by clicking on **C** in the DiagnosticCriteria slot.

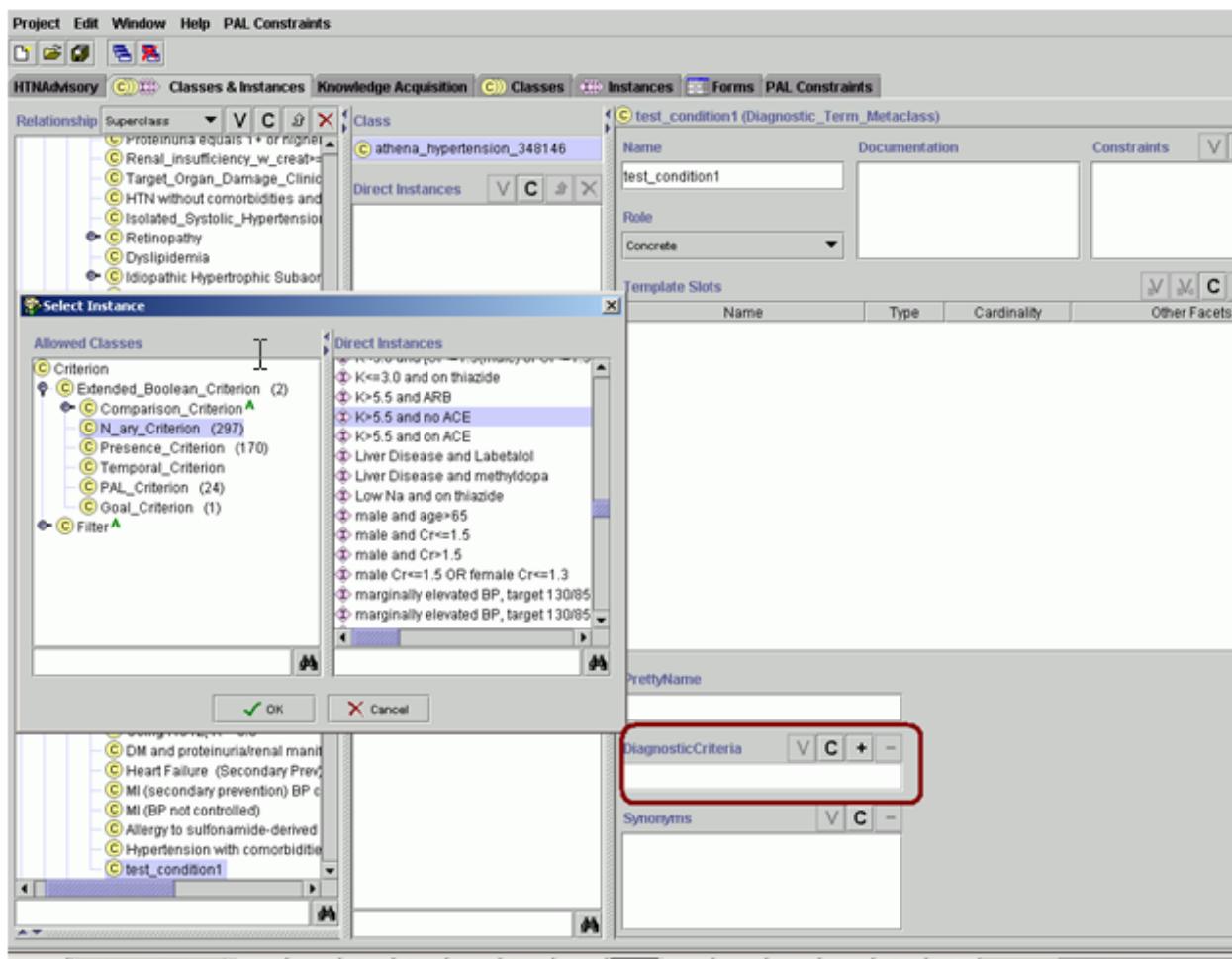


Figure 136 - Selecting rule-in criteria for test_condition1

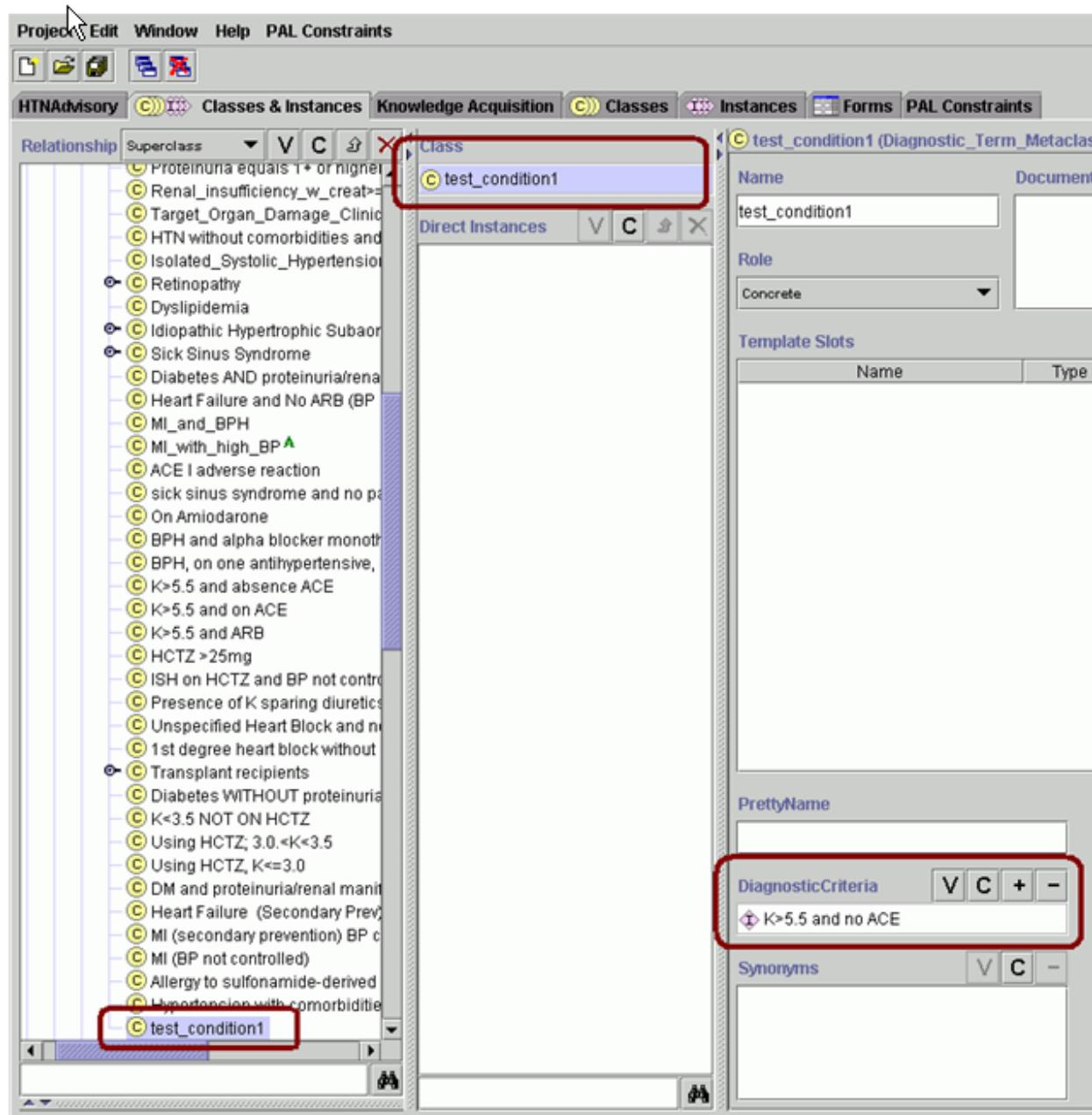


Figure 137 - Test_condition1 created

IV.5.4. Adding Messages to Display in the ATHENA-CDS Hypertension Advisory

Messages provide a way to alert and inform the ATHENA-CDS user about issues specific to the patient. Messages can be displayed in four places in an ATHENA-CDS hypertension advisory, namely the:

1. primary recommendation box (which contains “Consider INTENSIFYING drug treatment...” in Figure 138),
2. Precautions tab,
3. Assumptions tab, and
4. Info button (when related to a drug recommendation).

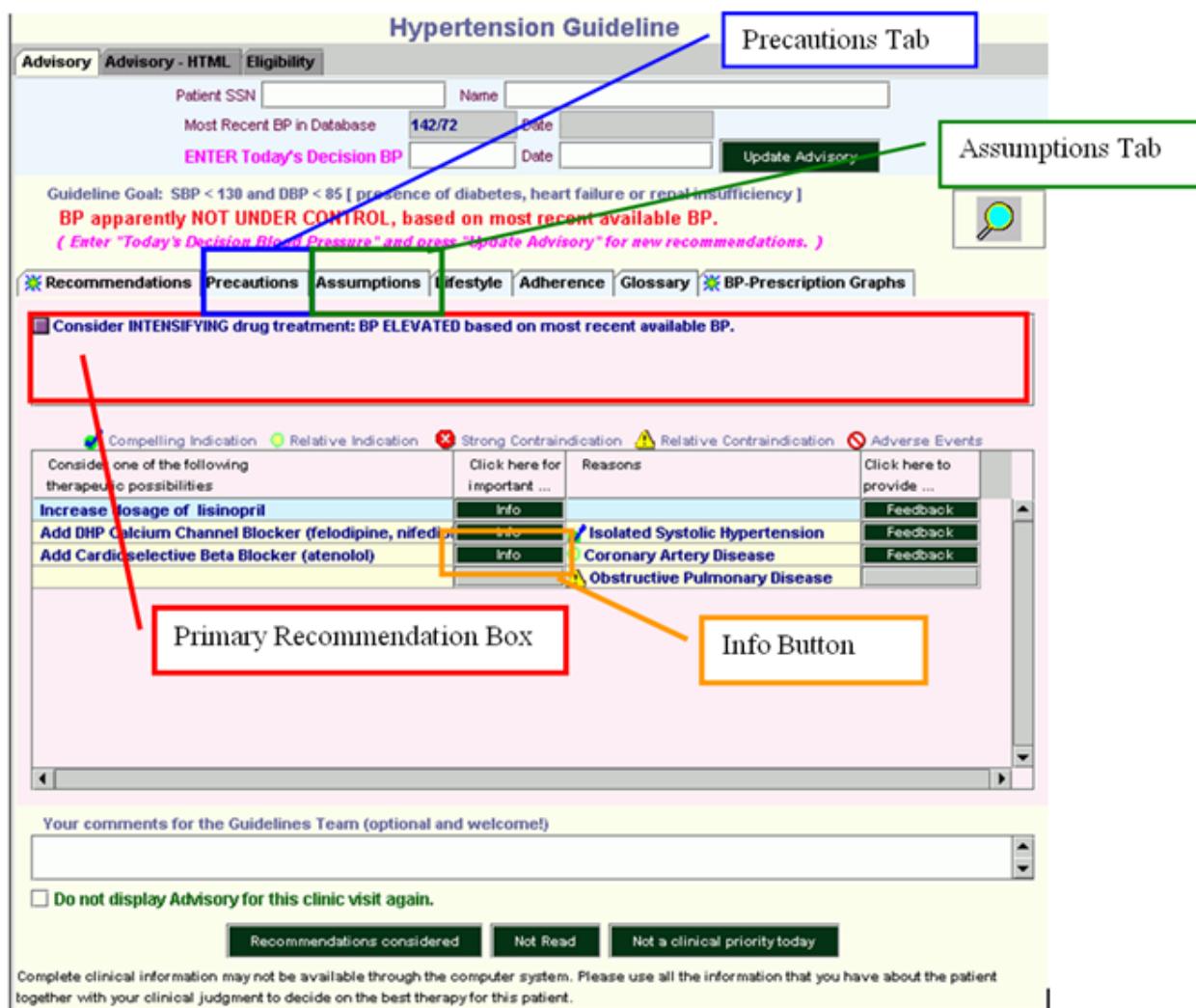


Figure 138 - ATHENA-CDS hypertension advisory

It is possible to add messages to the Knowledge Base in three distinct places:

1. *Scenario* – (pink boxes in the hypertension management diagram from the Knowledge Acquisition tab; see Figure 139 and Figure 140) primary recommendation box, Precautions tab, or Assumptions tab
2. *Action_Choice* – (green bubbles in the hypertension management diagram from the Knowledge Acquisition tab; see Figure 139 and Figure 140) primary recommendation box, Precautions tab, or Assumptions tab
3. *Drug Classes* – (Knowledge Acquisition tab) Info button

The message types and the rule-in/-out criteria of a message control when and where a message will be displayed in the advisory. The message types are covered in greater depth in the following three subsections.

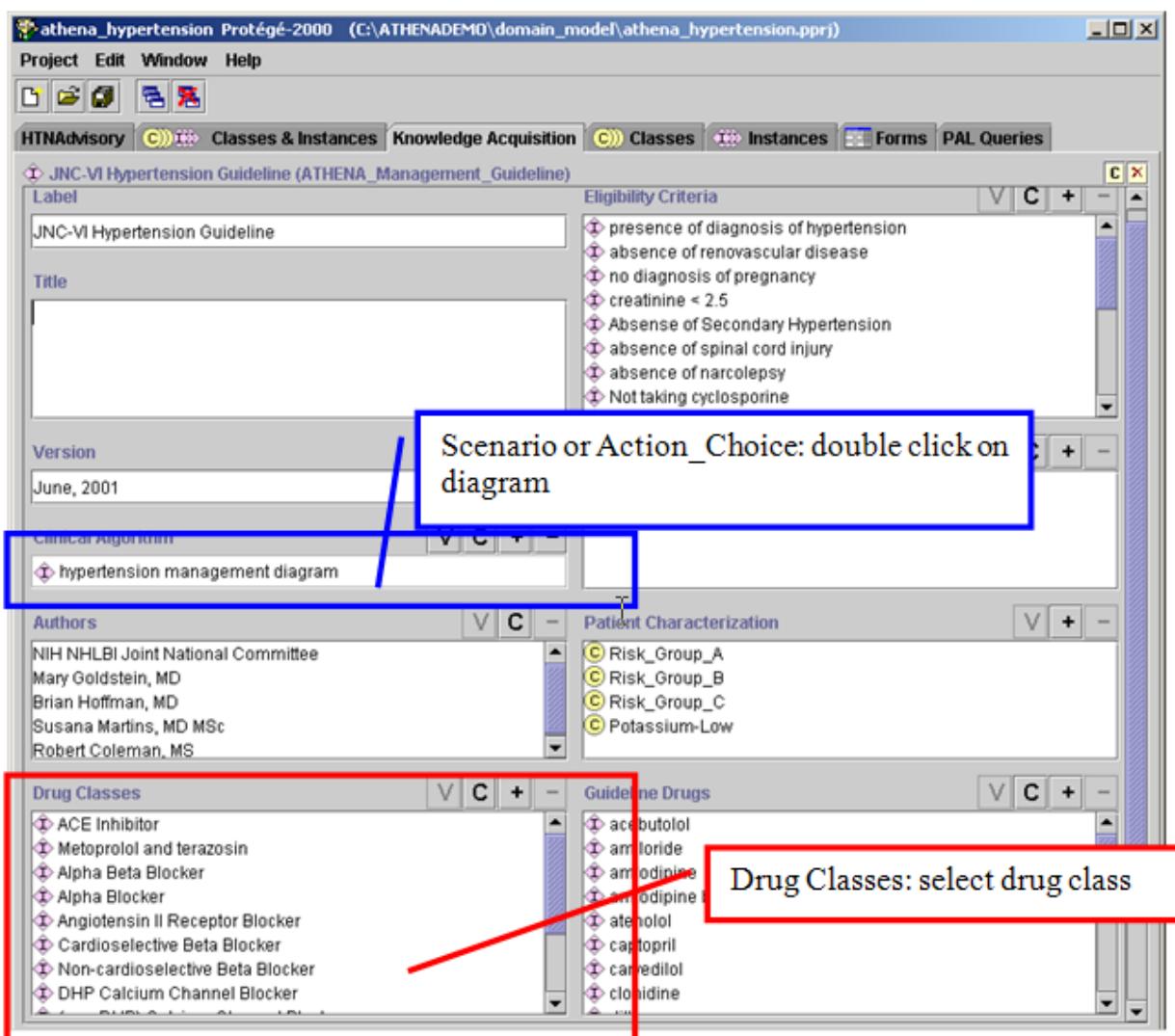


Figure 139 - Knowledge Acquisition tab

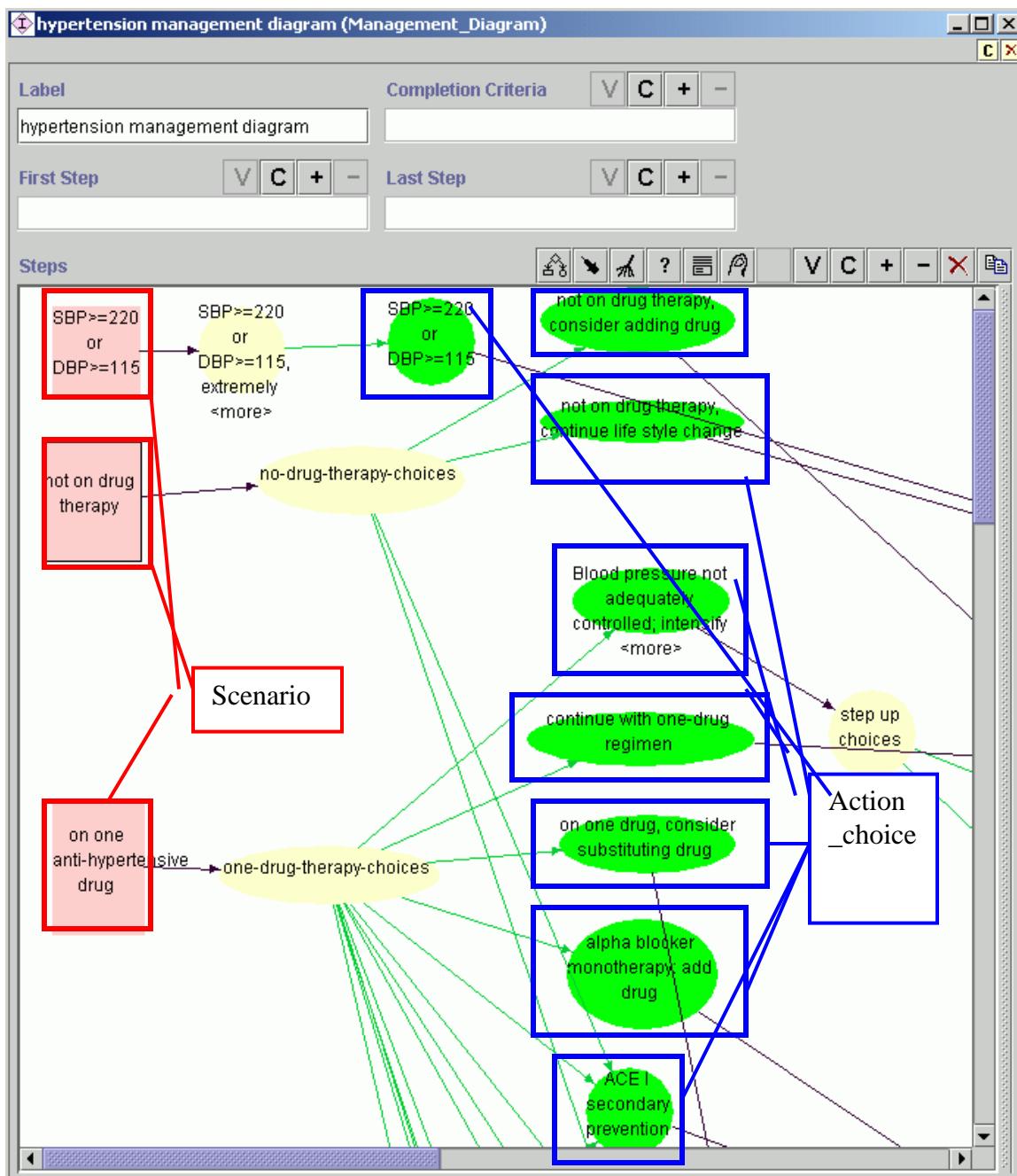


Figure 140 - Hypertension management diagram (Management_Diagram)

Scenario-related Message

A scenario-related message is associated with the scenarios (pink boxes) in the hypertension management diagram (see Figure 140) and are stored in the Consultation Template slot (Figure

141). Examples of scenarios are: “on one antihypertensive drug” or “on 2 or 3 antihypertensive drugs”.

To view the hypertension management diagram, go to the Knowledge Acquisition tab in Protégé; select the *hypertension management diagram* in the *clinical algorithm* slot; and click on **V** (Figure 139). In the diagram, pink boxes are instances of the Scenario class. These scenarios are mutually exclusive. It is necessary to add an identical message to each scenario in order to ensure that the message will display in all the desired scenarios.

For example, double click on the *not on drug therapy* scenario in the hypertension management diagram (Figure 140). In the form ‘not on drug therapy (scenario) that pops up there is a slot for the Consultation Template (Figure 141). Click on **V** to view the consultation template (see Figure 142).

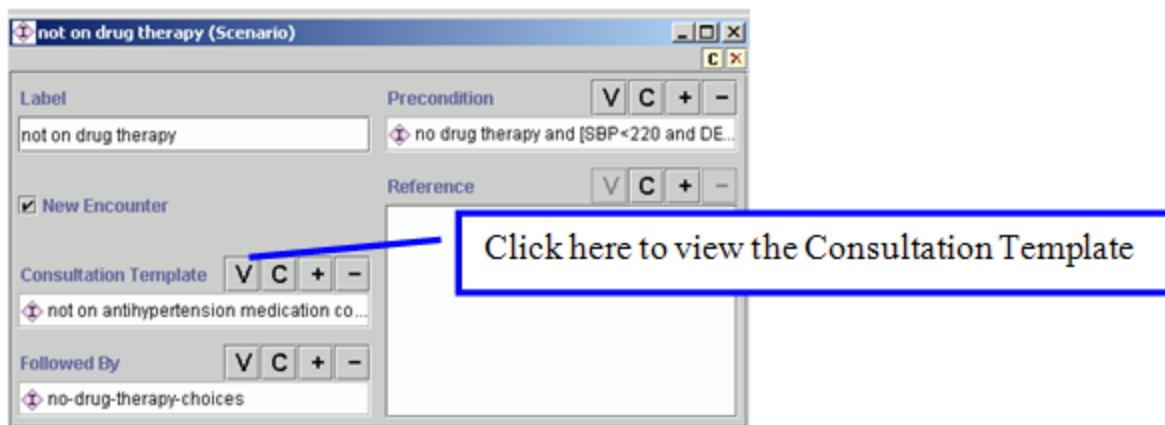


Figure 141 - "Not on drug therapy" scenario

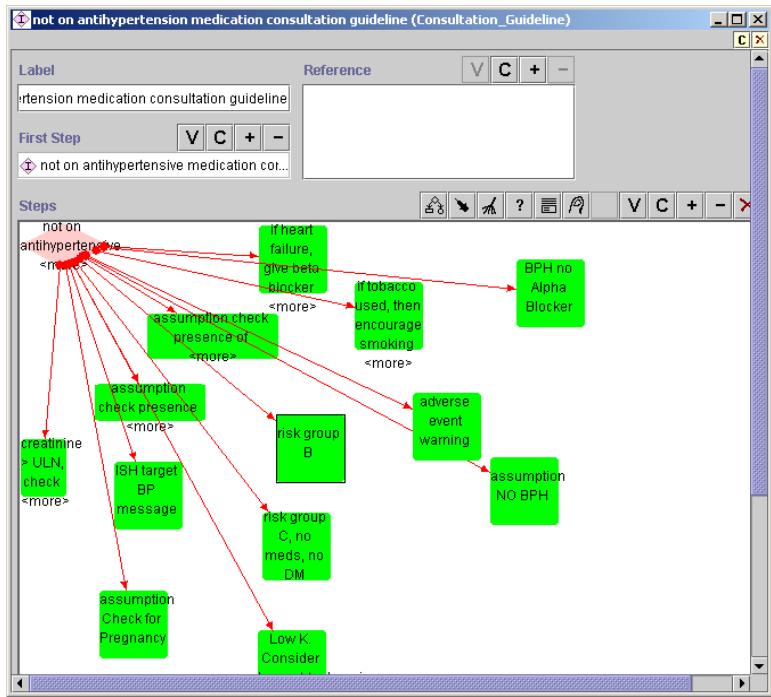


Figure 142 - Scenario *not on drug therapy* consultation template

To add a message to this scenario:

1. Create the Consultation_Action by clicking on **C** (Figure 143, then select Consultation_Action_Step (Figure 144).

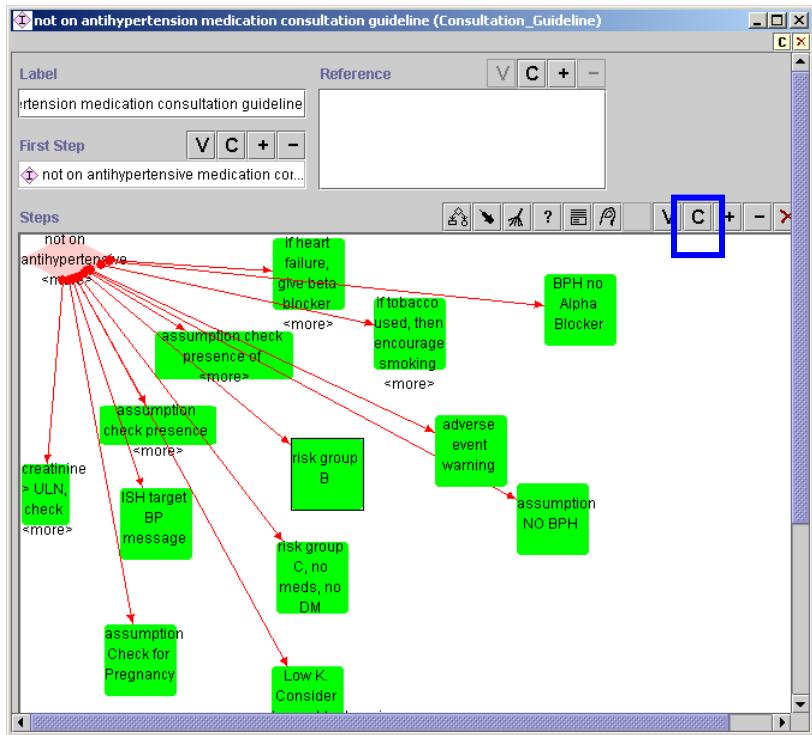


Figure 143 - Creating a new Consultation_Action_Step

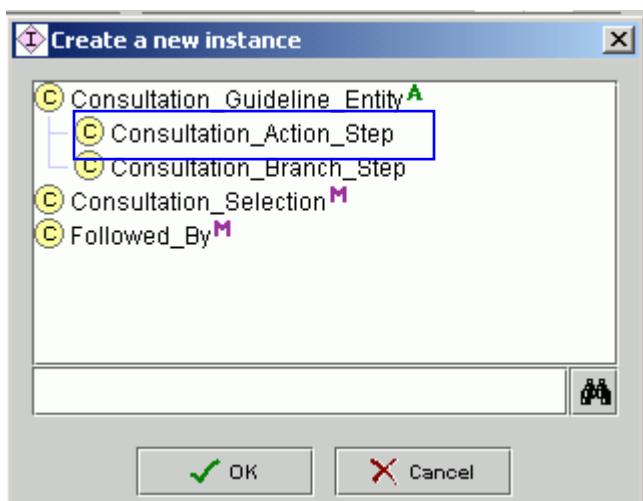


Figure 144 - Selecting Consultation_Action_Step

A new green square will appear in the diagram view of the Consultation_Guideline, with a default name. Double click on the new green square to view the template. Define the Label and

the criteria for Rule In and Rule Out. In Figure 145, a rule-in criterion is being created specifying that serum creatinine >2.

2. Add a descriptive label to the consultation action step, and add or create the rule-in and rule-out criteria for the consultation action step (see Figure 145). At the ‘Actions’ slot, create a new message by clicking on **C** and selecting the type of message; or add a message that already exists by clicking on **+**. Filling in the ‘Followed By’ slot is not required in order to display messages.

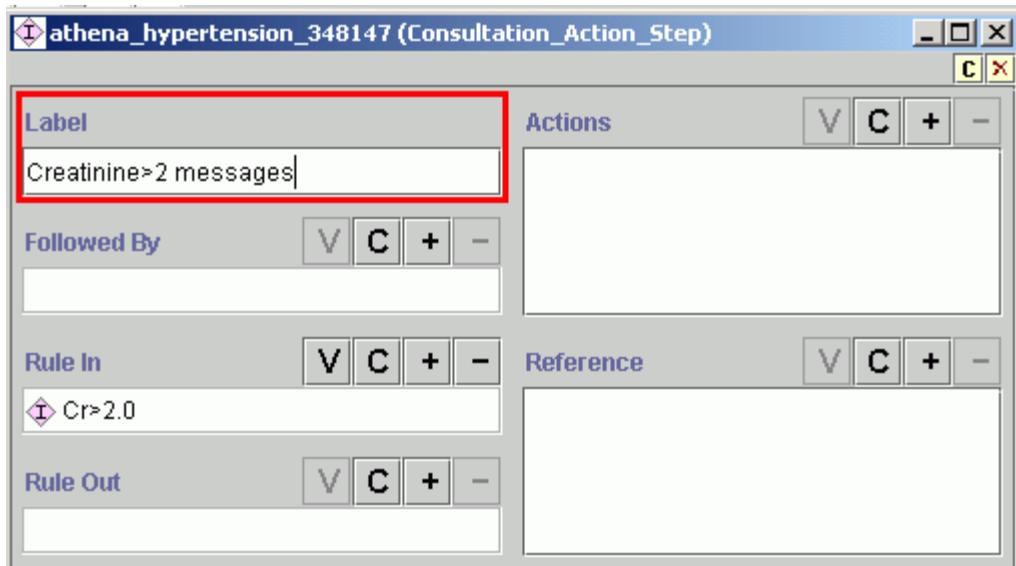


Figure 145 - Adding a label to Consultation_Action_Step; rule-in criterion Cr>2.0 established

When creating a new message, you will select from three types (see Figure 146):

- **On_Screen_Message** – Does not have rule-in criteria in the template.
- **Conditional_On_Screen_Message** – Has rule-in criteria in the template.
- **Presence_Message** – Has a Presence check box in the template for a medical domain term (for more details, see Subsection IV.2).

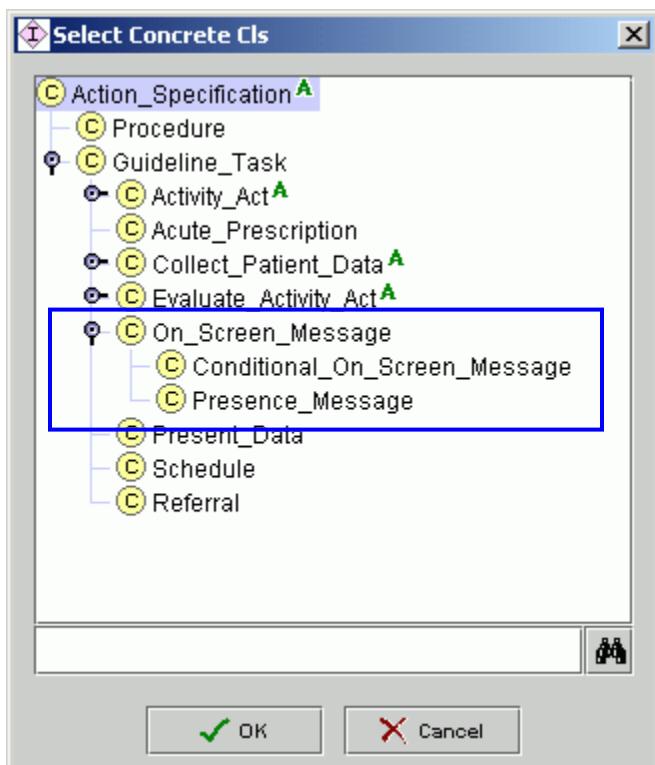


Figure 146 - Choice of message type, dialog box

3. Once you have selected the message type—for example, On_Screen_Message—the specific template will come up (Figure 147). Fill the Description and Label slots with descriptive content; enter the message to be displayed into the Message slot; and select the message type, according to where it is to be displayed in the Advisory (Primary_Recommendation in a primary recommendation box, Recommendation in the Precautions tab, and Assumption in the Assumptions tab). If you have a reference—that is, documentation supporting the advice delivered by the message—add that reference in the message's Reference slot.

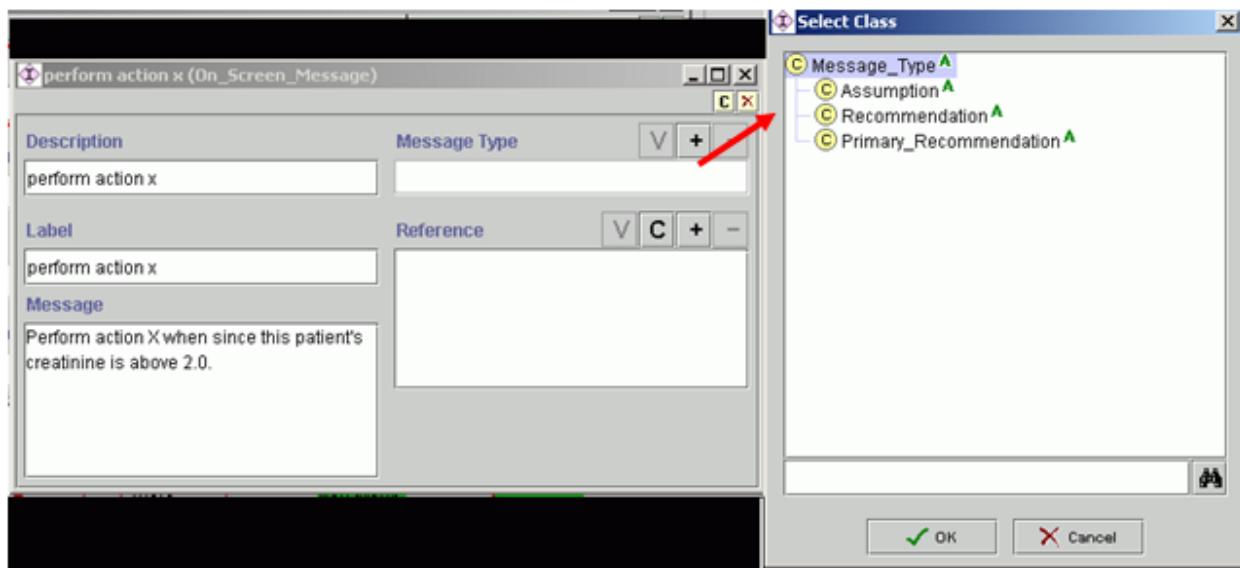


Figure 147 - On_Screen_Message

In Figure 147, the On_Screen_Message labeled *perform action X* has been created. It has the rule-in criteria: *Creatinine>2.0* and *not on drug therapy*. Accordingly, the message will be presented to the user only if creatinine is greater than 2.0 (because of the rule-in criterion in the consultation action step in Figure 147), and the patient is not on drug therapy (because of the precondition of the scenario in Figure 142). You can add numerous messages to the Actions slot, which will be constrained by the precondition criteria of the chosen scenario and the rule-in criteria of the consultation action step. If you need to constrain the criteria further, you can choose Conditional_On_Screen_Message (see Figure 146).

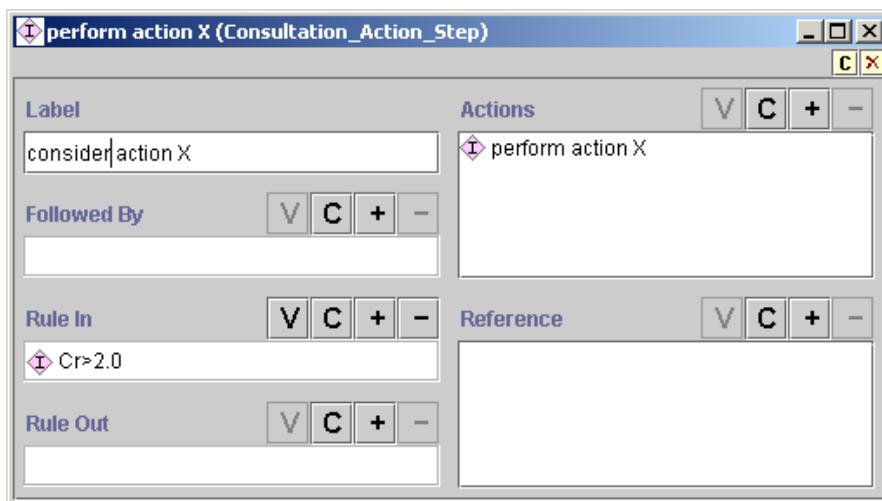


Figure 148 – Perform action X, with rule-in criterion Cr>2.0 that was inherited from scenario *not on drug therapy*

4. Create a link between the Consultation_Branch_Step (diamond) and the Consultation_Action you have just created using the *followed by* arrow in the menu on the right hand side. (Figure 149)

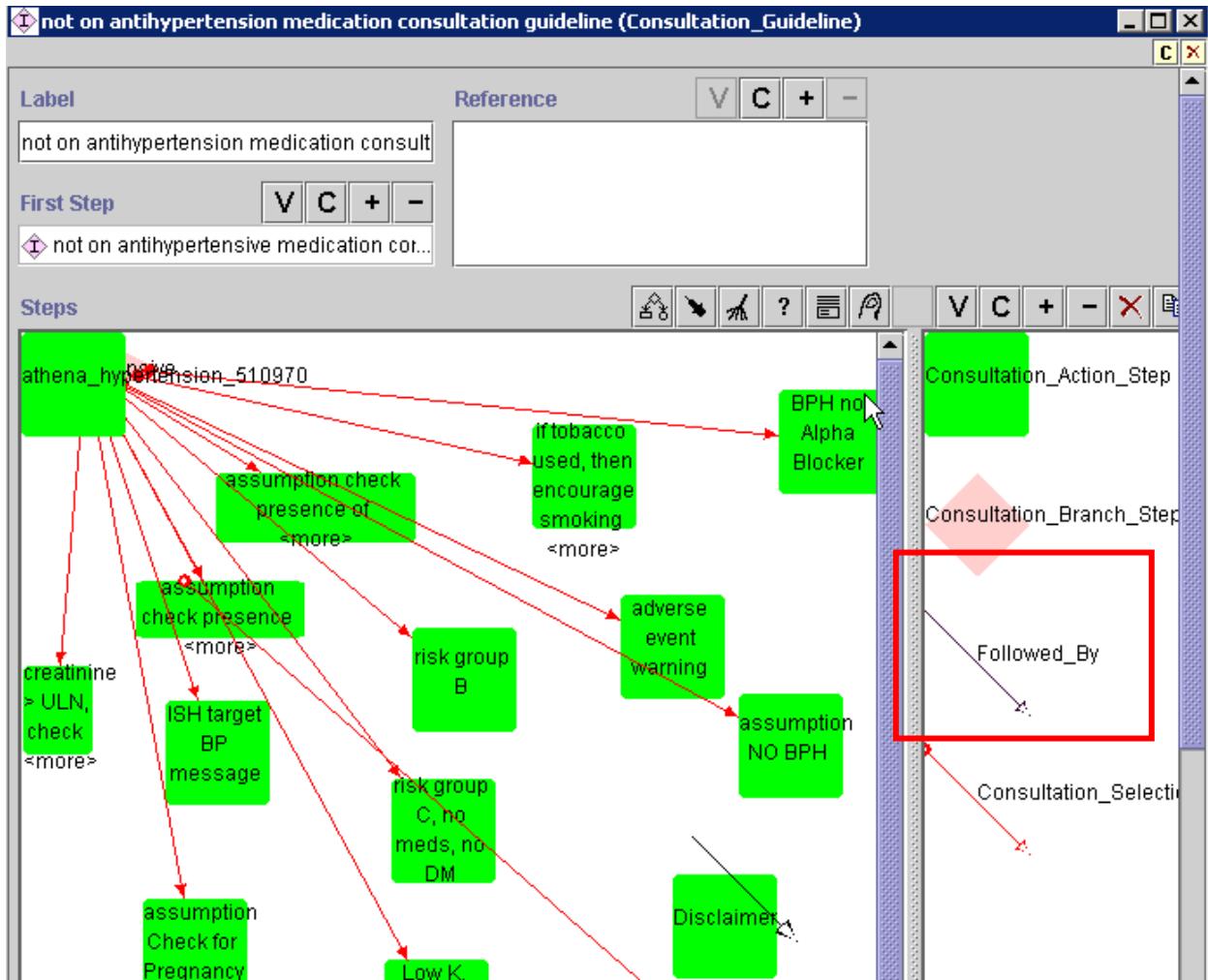


Figure 149: Consultation Guideline template: ‘Followed By’ arrow highlighted

5. Test the new message, as described on page 202.

Action_Choice-related Message

Action_Choice-related messages are those associated with alternative choices in a decision. They are represented as green bubbles in the hypertension management diagram (Figure 150).

Go to the Knowledge Acquisition tab of Protégé and view the hypertension management diagram. In the diagram, the action choices—green bubbles—may not be mutually exclusive

alternatives. Note that you may need to add an identical message in various action choices in order to ensure that the message will display in all the desired scenarios.

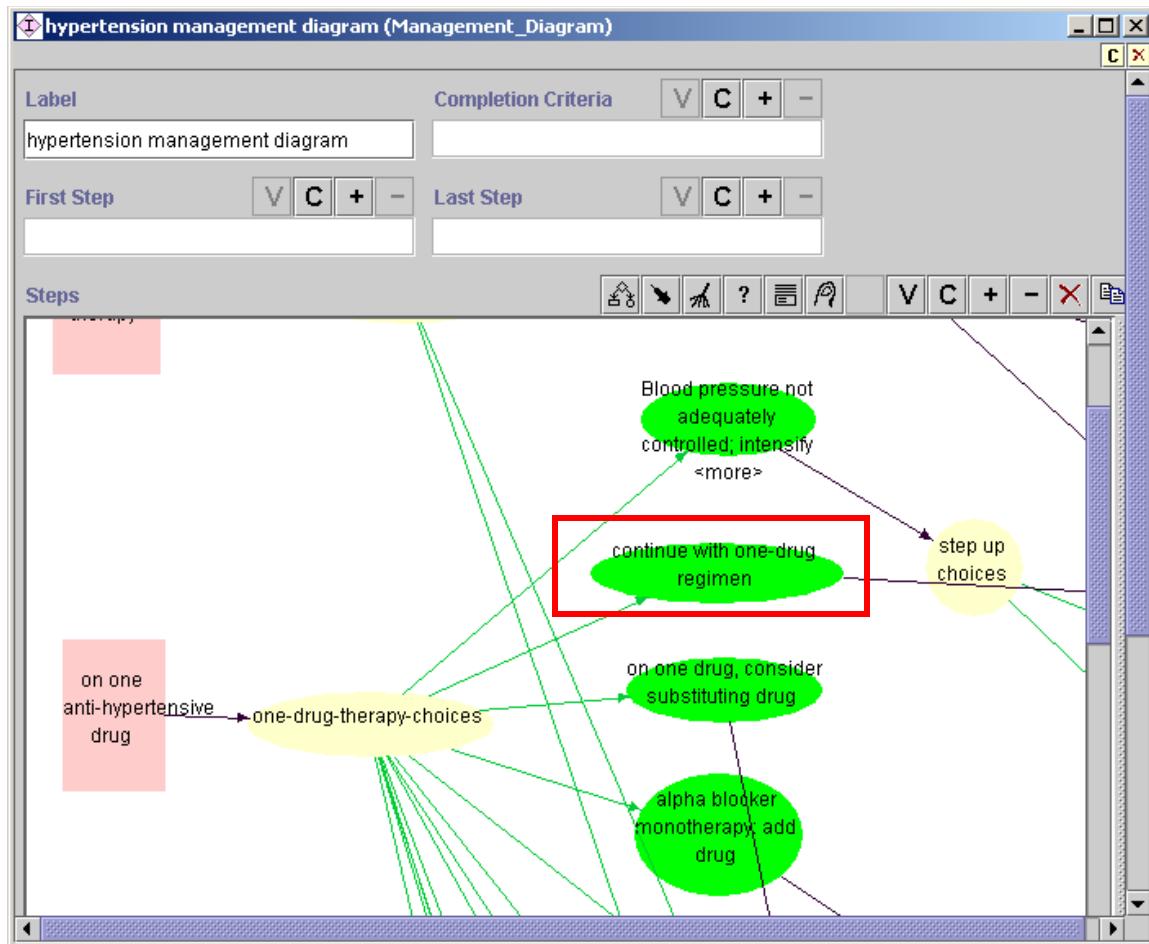


Figure 150 - Clinical algorithm diagram

To add a new Action_Choice-related message to the action choice, *continue with one-drug regimen*, carry out the following steps:

1. Add a message to the Advisory for patients on thiazide monotherapy when BP is controlled about a K (Potassium) test. From the action choices following the scenario—*on one anti-hypertensive drug* (pink square), *one-drug-therapy-choices* (yellow oval), etc.—select the *continue with one-drug regimen* action choice (green bubble) by clicking on it. This will call up the *continue with one-drug regimen* Action_Choice (Figure 151). The actions (messages) will be activated only if the ‘Strict Rule In Condition’(s) is/are true and the ‘Strict Rule Out Condition’(s) is/are not true for the Action_Choice as well as any criteria specific to the message (if a conditional message, additional ‘rule in’

criteria may be added). ATHENA-CDS is not currently using the ‘Rule In Condition’ and the ‘Rule Out Condition’ slots.

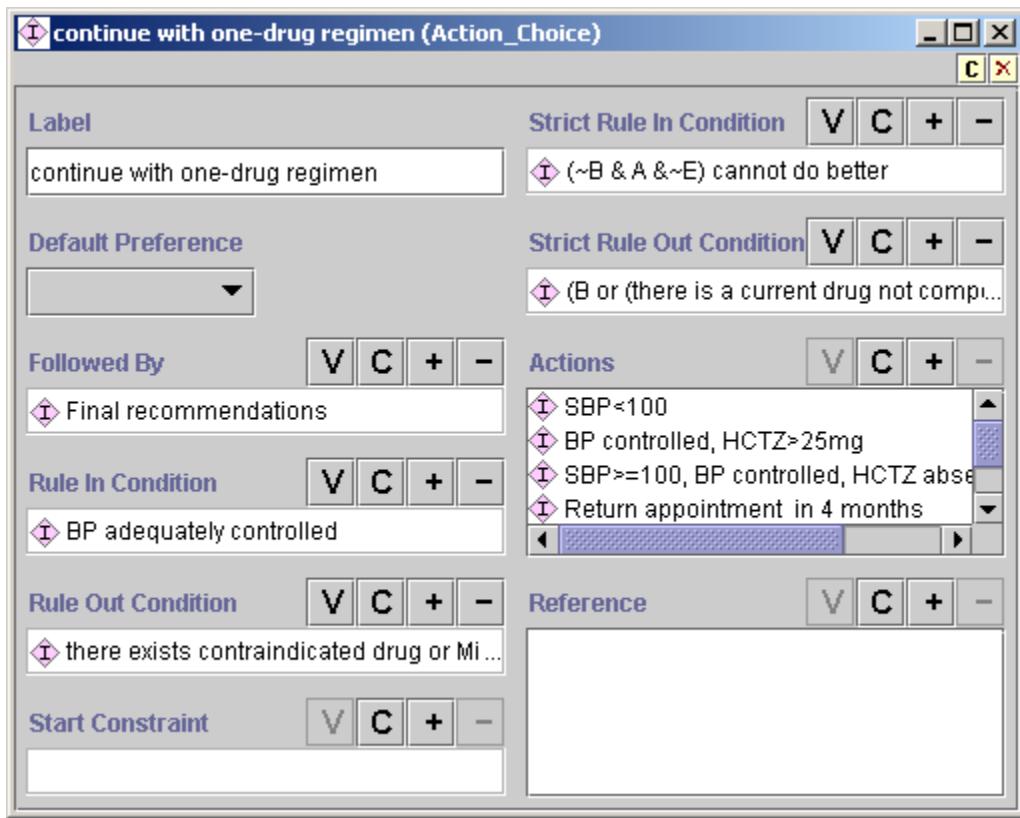


Figure 151 - Action_Choice, *continue with one-drug regimen*

2. Create a new message by clicking on the **C** button in the Actions slot (Figure 151). The following dialog box will appear (Figure 152). For a message specifying thiazide monotherapy, there are rule-in criteria other than the one determined by ‘*continue with one-drug regime*’, so select Conditional_On_Screen_Message. The template in Figure 153 will appear.

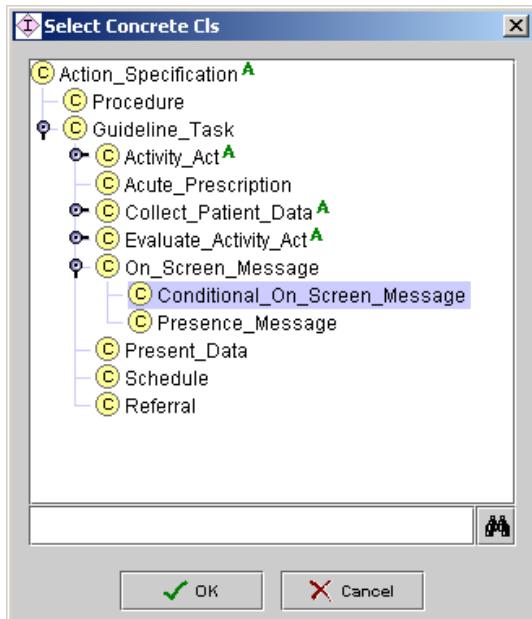


Figure 152 - Dialog box for selecting message type

3. In the template (Figure 153), add a descriptive label to the Label slot; type the message in the Message slot, as it is to appear in the Advisory; select from Message Type according to where the message is to be located in the Advisory; and add/create the rule-in condition in the Rule In Condition slot.

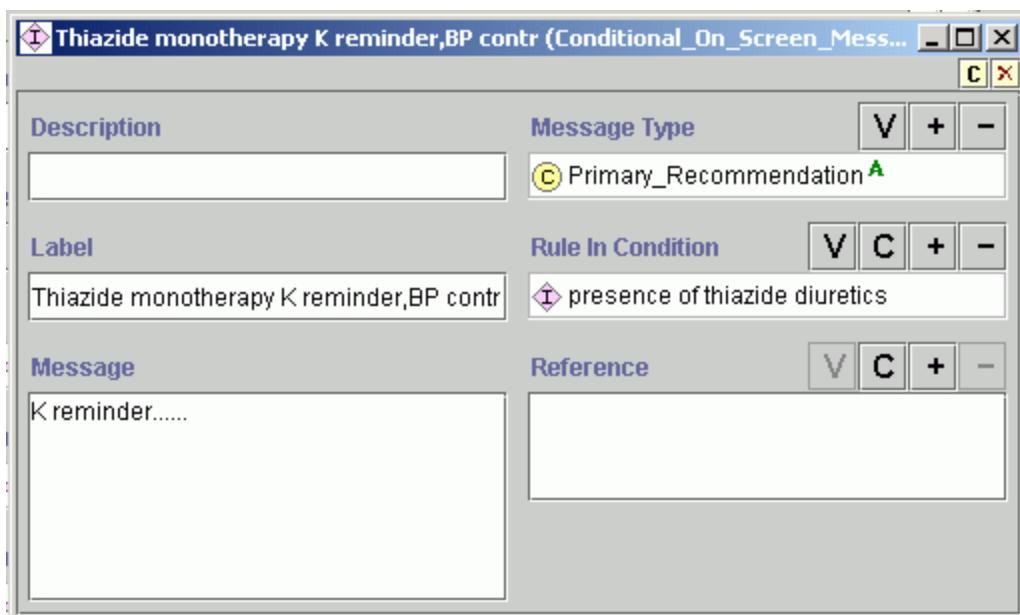


Figure 153 - Conditional_On_Screen_Message, K reminder

The new message should appear in the list in the Actions slot for Action_Choice (Figure 154).

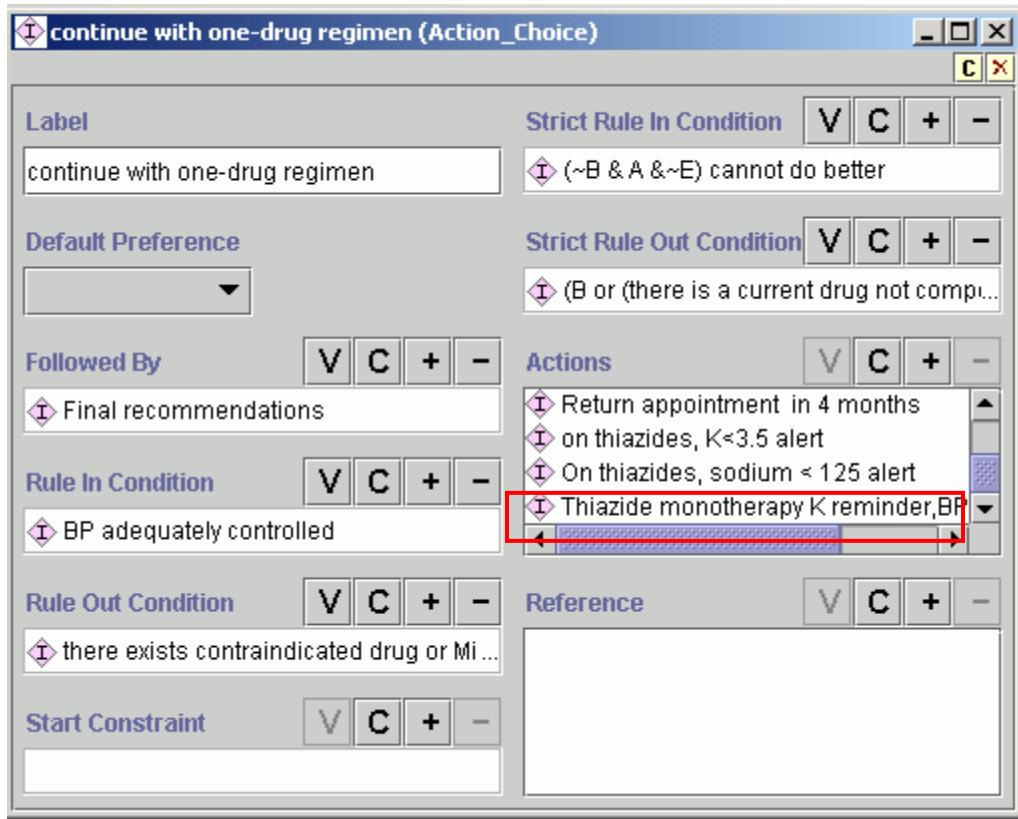


Figure 154 - Message added to the Action_Choice for *continue with one-drug regimen*

4. To add messages that have been previously created, click on the + button in the Actions slot (Figure 154), select the appropriate message type, highlight the message, and click OK (Figure 155).

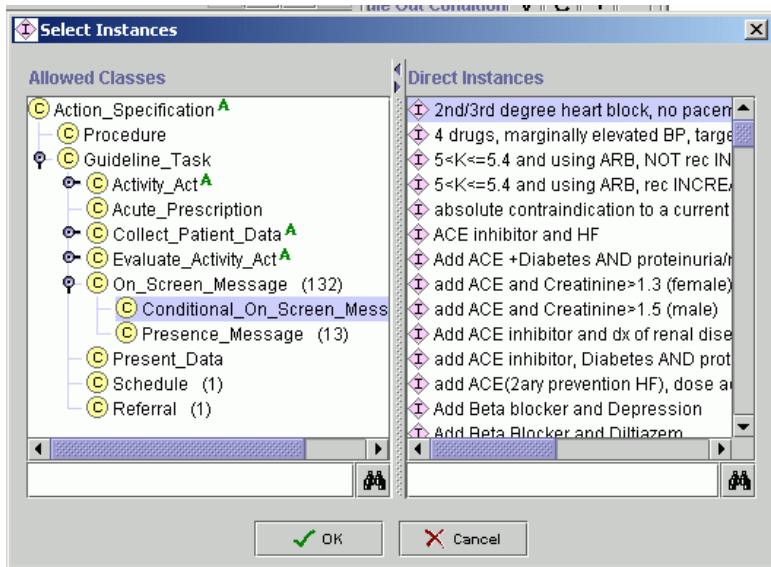


Figure 155 - Adding an already created message

5. Test the new message, as described in 202

Drug Classes-related Message

Drug class-related messages are messages that display when a clinician clicks on the Info button next to a specific drug recommendation.

Hypertension Guideline

Advisory **Advisory - HTML** **Eligibility**

Patient SSN	Name			
Most Recent BP in Database	142/72	Date		
ENTER Today's Decision BP		Date	<input type="button" value="Update Advisory"/>	

Guideline Goal: SBP < 130 and DBP < 85 [presence of diabetes, heart failure or renal insufficiency]
BP apparently NOT UNDER CONTROL, based on most recent available BP.
(Enter "Today's Decision Blood Pressure" and press "Update Advisory" for new recommendations.)

Recommendations **Precautions** **Assumptions** **Lifestyle** **Adherence** **Glossary** **BP-Prescription Graphs**

Consider INTENSIFYING drug treatment: BP ELEVATED based on most recent available BP.

Compelling Indication Relative Indication Strong Contraindication Relative Contraindication Adverse Events

Consider one of the following therapeutic possibilities	Click here for important ...	Reasons	Click here to provide ...
Increase dosage of lisinopril	<input type="button" value="Info"/>	Isolated Systolic Hypertension	<input type="button" value="Feedback"/>
Add DHP Calcium Channel Blocker (felodipine, nifedipine)	<input type="button" value="Info"/>	Coronary Artery Disease	<input type="button" value="Feedback"/>
Add Cardioselective Beta Blocker (atenolol)	<input type="button" value="Info"/>	Obstructive Pulmonary Disease	<input type="button" value="Feedback"/>

Your comments for the Guidelines Team (optional and welcome!)

Do not display Advisory for this clinic visit again.

Complete clinical information may not be available through the computer system. Please use all the information that you have about the patient together with your clinical judgment to decide on the best therapy for this patient.

Figure 156 - ATHENA-CDS hypertension advisory

Steps to create a drug classes-related message:

1. In the Knowledge Acquisition tab, double click on the drug class of interest, for example, ACE Inhibitor. The template in Figure 157 will appear.

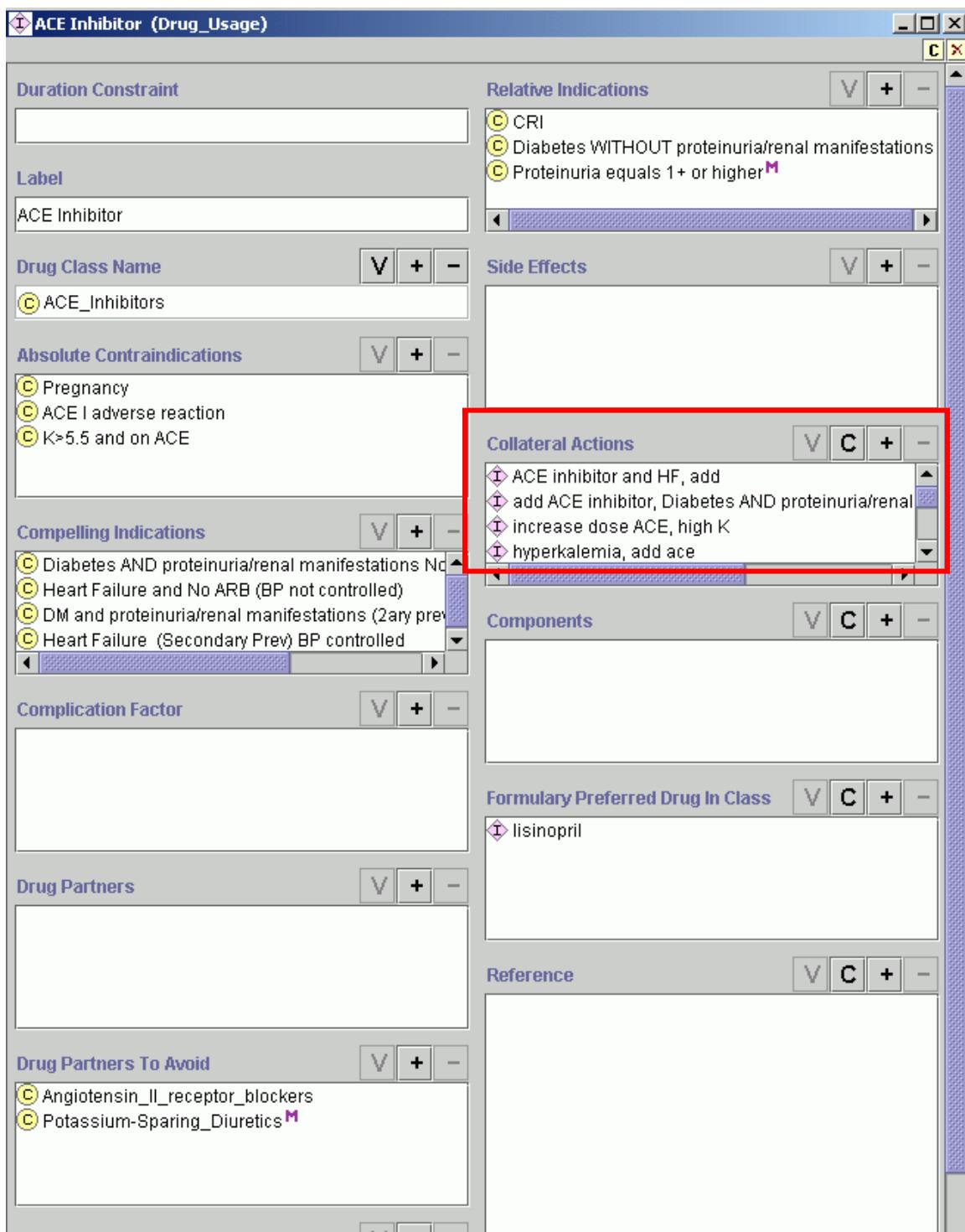


Figure 157 - Drug_Usage template

2. Go to the Collateral Actions slot, and add a previously created message by clicking on the **+** button, or create a new message by clicking on the **C** button (Figure 157). When creating a new message, the template in Figure 158 will appear.

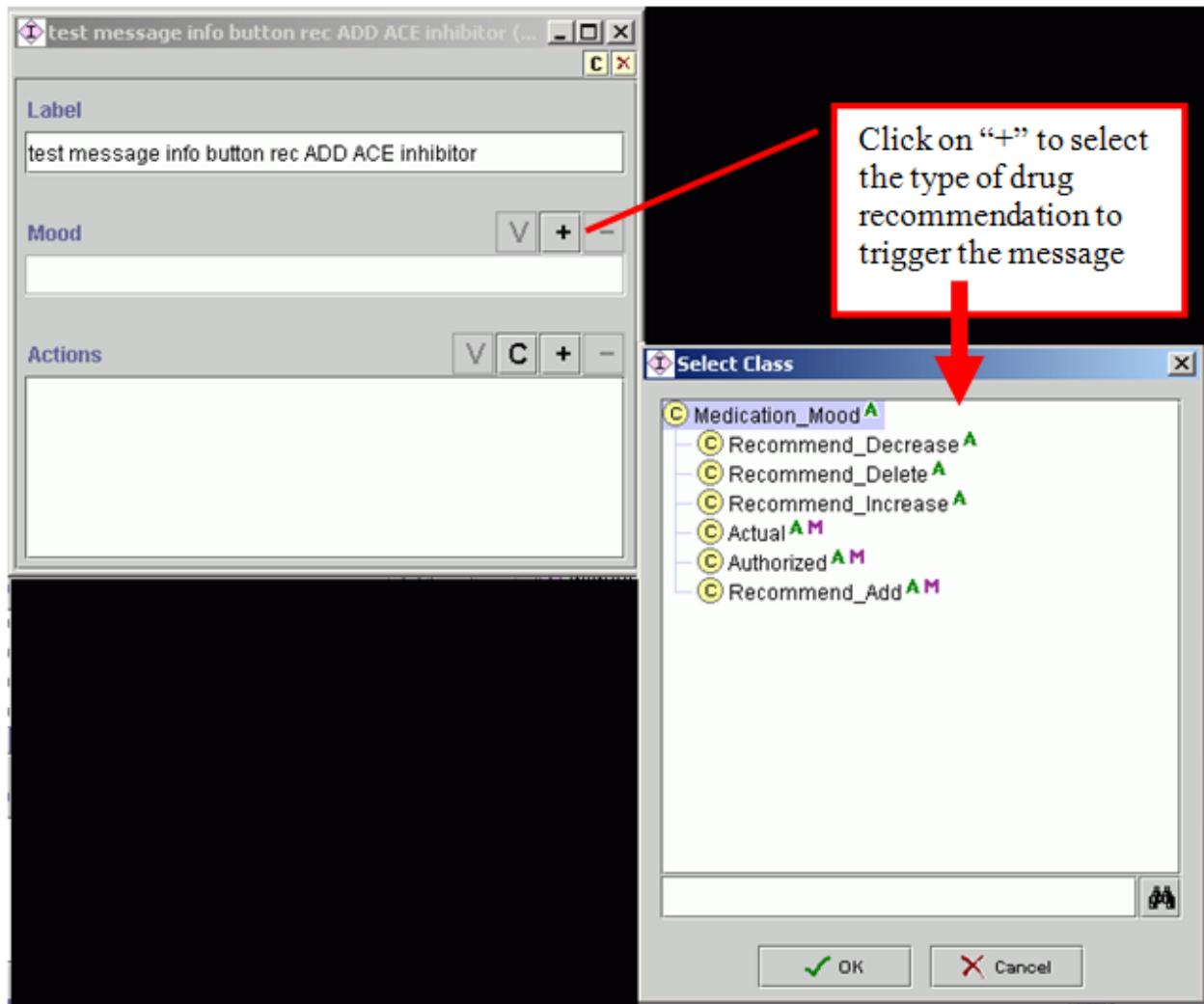


Figure 158 - Creating a drug classes-related message

3. Add a descriptive label to the message, then click on the **+** button in the Mood slot to select the type of drug recommendation to trigger the message (Figure 158):
 - *Recommend_Decrease* – when ATHENA-CDS is recommending a decrease in the dose of a specific drug class (in this example, ACE inhibitor)
 - *Recommend_Delete* – when ATHENA-CDS is recommending deleting a drug of a specific drug class (e.g., ACE inhibitor), for instance, in a substitution

- *Recommend_Increase* – when ATHENA-CDS is recommending the increased dose of a specific drug class (e.g., ACE inhibitor)
- *Actual* – for the inpatient setting; not used in ATHENA-CDS
- *Authorized* – when a patient has an active prescription for a drug
- *Recommend_Add* – when ATHENA-CDS is recommending adding a drug of a specific drug class (e.g., ACE inhibitor)

In the Actions slot, click on **+** to add an already-created message or click on the **C** button to create a new one. When creating a new message, if there are any additional rule-in criteria, select conditional_on_screen_message type from the dialog box. If no additional rule-in criteria are required, select the On_Screen_Message type. Add a descriptive label and the message content in the Message slot, as they are to be displayed in the GUI (Figure 159). It is not necessary to select a message type, as this message is linked to a drug recommendation. If you have a reference, you can add it or create a new one, using the **+** button and **C** button, respectively (Figure 159).

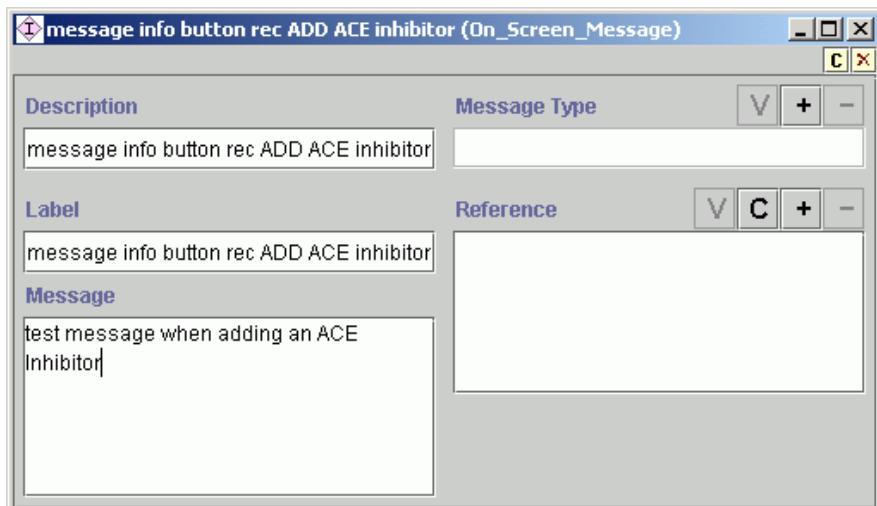


Figure 159 - On_Screen_Message for a drug class-related message

4. In the collateral action” slot of the Drug_Usage template for ACE Inhibitor, scroll down and see the new message labeled “Test message info button rec ADD ACE Inhibitor” (Figure 160)

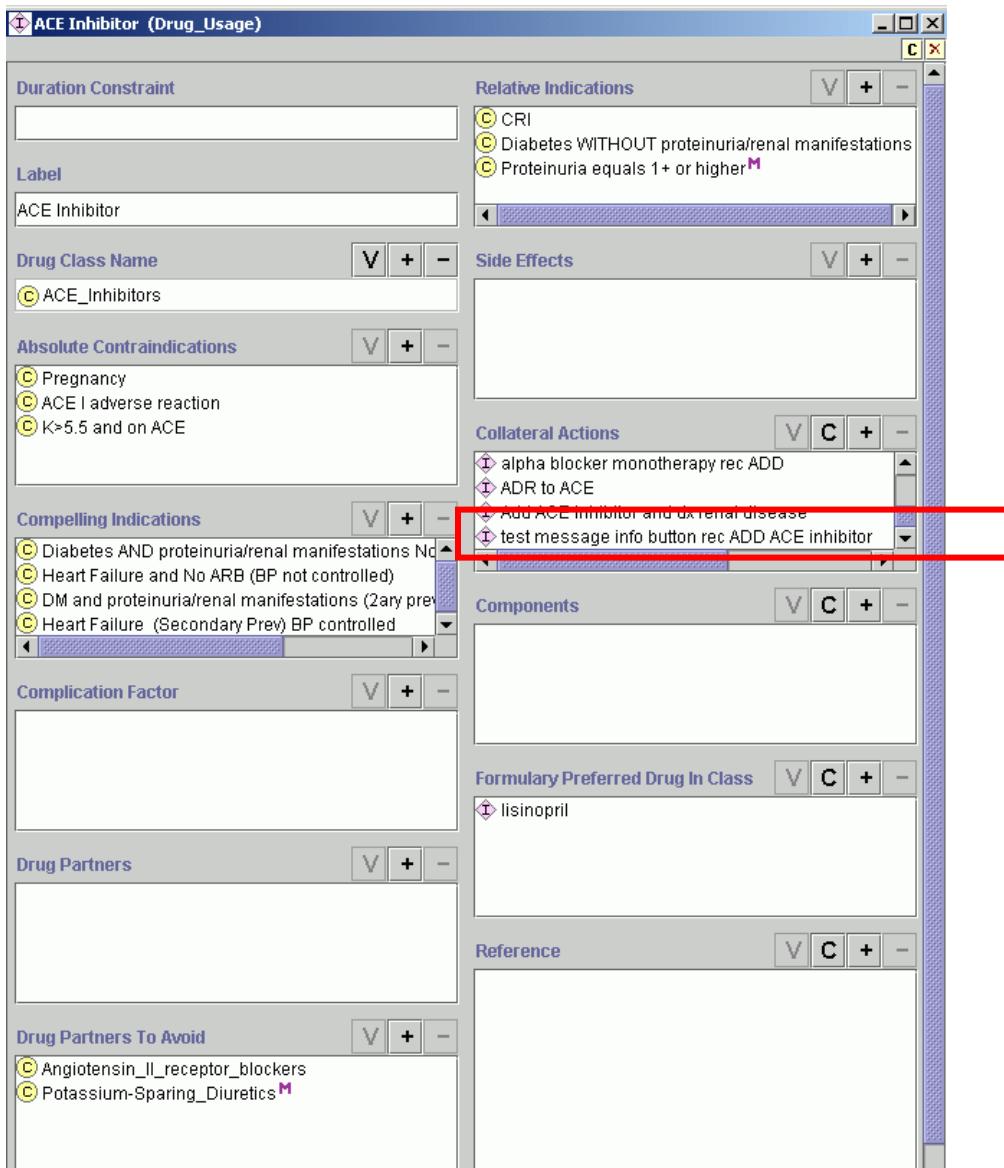


Figure 160 - Drug_Usage template with a new message in the Collateral Actions slot

5. Test the new message, as described in page 202.

IV.5.4.1. Creating a New Primary Message Using the Classes & Instances Tab

This subsection will walk through an example of how to add a message that has some rule-in criteria, i.e., a conditional on-screen message.

1. Go to the Classes & Instances tab.

2. Enter *message* into the search field (see Figure 161).
3. Select conditional_on_screen_message.
4. Click on **C** in Direct Instances, highlighted in Figure 161, to create a new message.

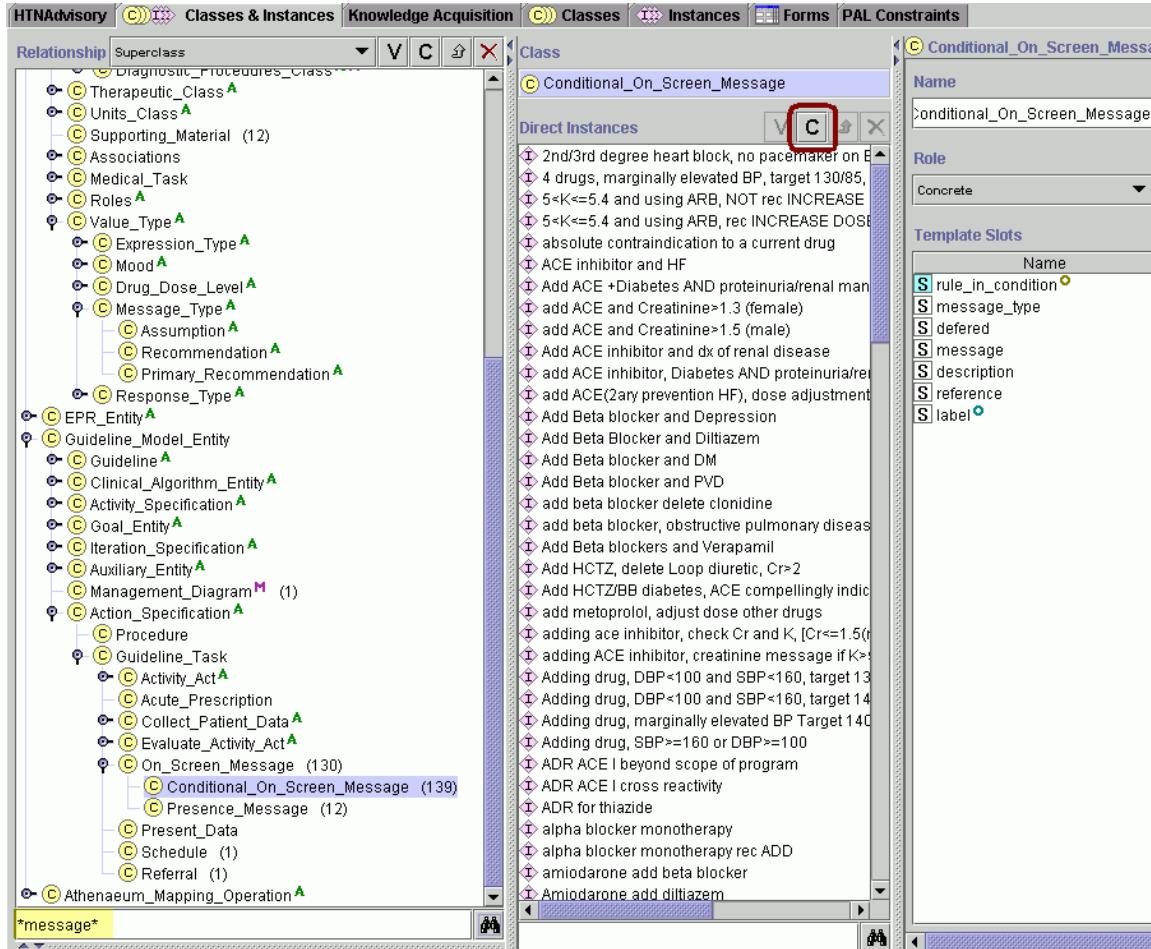


Figure 161 - Creating a conditional on-screen message

5. Fill in the slots highlighted in red in Figure 162:
 - *Description* – Add detail to the message description.
 - *Message Type* – Click on + to add the type of message. This will determine where it will display:
 - a) *Primary* – displays in a box above drug recommendations.
 - b) *Assumption* – displays in the Assumptions tab.
 - c) *Recommendation* – displays in the Precautions tab.

d) If left blank – displays in the Precautions tab.

- *Label* – Enter what is to be displayed in the Protégé user interface.
- *Rule In Condition* – Select the criteria to the display message. If criteria already exist, use **+** to find them and add them. If there need to be new criteria, then click on **C** to create them.
- *Message* – Enter the message as it is to be displayed.
- *Reference* – Insert a reference for the message (optional). Use **C** to create a new reference or **+** to find a reference that already exists in Protégé.

Create a test patient to evaluate if the message is displayed (see page 202).



Figure 162 - Template for a conditional on-screen message

IV.5.5. Update the Drugs Table (Mapping Table) in the SQL Server Database

The SQL Mail Support Service (within SQL Server)(for details-“Drug Check” SQL Server Job, described in III.3.3.2) will notify the software administrator by email if an antihypertensive drug is on the patient’s medication list, but not in the Drugs table in the ATHENA-CDS database. If a

clinic administrator receives an email reminder to add a drug, the email message will contain the information needed to update the mapping table in SQL Server.

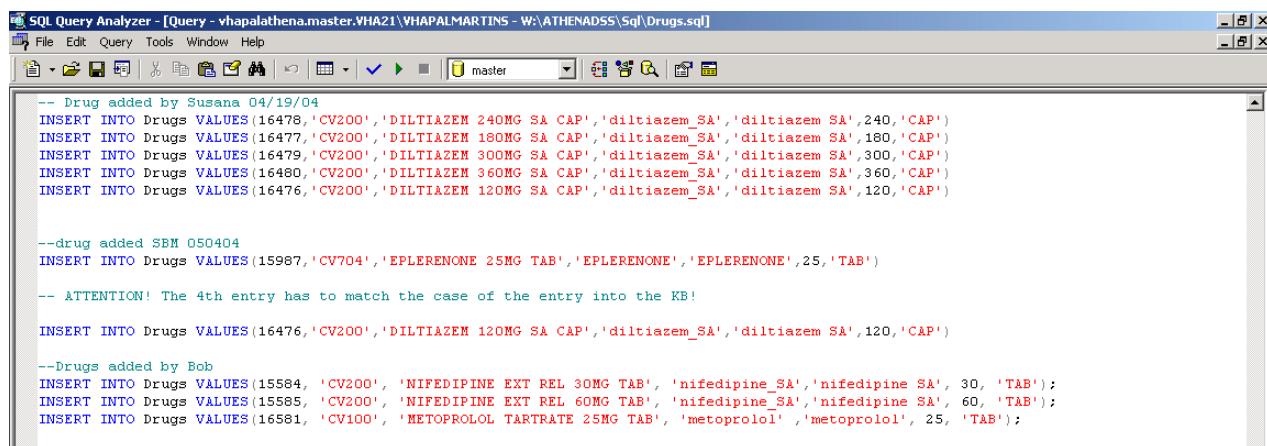
When adding a drug to the Knowledge Base class hierarchy, be sure to introduce it to the correct drug class. If it is to be a preferred drug or dose increases are to be recommended, then follow the instructions in 149 (Add a new drug) save, and exit Protégé.

To add a drug to the Drugs table in SQL Server so the drug can be identified by the guideline interpreter and the knowledge base:

1. Open SQL Enterprise Manager.
2. Open SQL Server Query Analyzer, under Tools in menu.
3. Open the file ATHENADSS/SQL/drugs in the Query Analyzer.
4. Add a new line of code to insert the drug. Follow existing SQL code as an example, and enter the data as seen in the email that was received: (ID, VAClass, Description, Name, Displayname, Dosage, Delivery). Be sure to enter the value for Name so that it exactly matches—case and spelling—the name you gave the drug in the Knowledge Base.

EXAMPLE of SQL Insert query to add new drug (Figure 163):

INSERT INTO Drugs VALUES (15987, CV704, EPLERENONE 25MG TAB, EPLERENONE, EPLERENONE, 25, TAB)



The screenshot shows the SQL Server Query Analyzer interface. The title bar reads "SQL Query Analyzer - [Query - vhapalathena.master.WHA21\WHAPALMARTINS - W:\ATHENADSS\Sql\Drugs.sql]". The main window displays an SQL script for adding a new drug. The script includes comments indicating drug additions by different users (Susana, SBM 050404, Bob) and specific drug entries. The code uses the 'VALUES' clause to insert data into the 'Drugs' table, specifying fields like ID, VAClass, Description, Name, Displayname, Dosage, and Delivery.

```
-- Drug added by Susana 04/19/04
INSERT INTO Drugs VALUES(16478,'CV200','DLITIAZEM 240MG SA CAP','diltiazem_SA','diltiazem_SA',240,'CAP')
INSERT INTO Drugs VALUES(16477,'CV200','DLITIAZEM 180MG SA CAP','diltiazem_SA','diltiazem_SA',180,'CAP')
INSERT INTO Drugs VALUES(16479,'CV200','DLITIAZEM 300MG SA CAP','diltiazem_SA','diltiazem_SA',300,'CAP')
INSERT INTO Drugs VALUES(16480,'CV200','DLITIAZEM 360MG SA CAP','diltiazem_SA','diltiazem_SA',360,'CAP')
INSERT INTO Drugs VALUES(16476,'CV200','DLITIAZEM 120MG SA CAP','diltiazem_SA','diltiazem_SA',120,'CAP')

--drug added SBM 050404
INSERT INTO Drugs VALUES(15987,'CV704','EPLERENONE 25MG TAB','EPLERENONE','EPLERENONE',25,'TAB')

-- ATTENTION! The 4th entry has to match the case of the entry into the KB!

INSERT INTO Drugs VALUES(16476,'CV200','DLITIAZEM 120MG SA CAP','diltiazem_SA','diltiazem_SA',120,'CAP')

--Drugs added by Bob
INSERT INTO Drugs VALUES(15584,'CV200','NIFEDIPINE EXT REL 30MG TAB','nifedipine_SA','nifedipine_SA',30,'TAB');
INSERT INTO Drugs VALUES(15585,'CV200','NIFEDIPINE EXT REL 60MG TAB','nifedipine_SA','nifedipine_SA',60,'TAB');
INSERT INTO Drugs VALUES(16581,'CV100','METOPROLOL TARTRATE 25MG TAB','metoprolol','metoprolol',25,'TAB');
```

Figure 163 - SQL query adding a new drug to the Drugs table in SQL Server

Select the new code and run the query in both the ATHENA DB and Atheneon DB.

IV.5.6. Update the Tabs in the ATHENA-CDS Hypertension Advisory

The content information for the Lifestyle, Adherence and Glossary tabs in the ATHENA-CDS hypertension advisory are stored in HTML pages that are located in \\ATHENADSS\doc\ATHENA. To change the content of these pages:

1. Create an HTML page with updated information.
2. Name the newly created page using the name of the page to be updated (replaced), making sure to store the new page in a different location *than the original file*. The current static html static pages are:
 - a) *Lifestyle – NonPharmacologic_Management.html*
 - b) *Adherence – Adherence to Antihypertensive Therapy.html*
 - c) *Glossary – Glossary for Hypertension Guideline Display.html*

In **ATHENA DEMO** version:

3. In order to save the original HTML file in use, rename the HTML file in C:\ATHENADEMO\doc\ATHENA \ by adding old to filename, for example NonPharmacologic_Management.html to NonPharmacologic_Management_OLD.html
4. Transfer the newly created pages to folder C:\ATHENADEMO\doc\ATHENA.
5. Test display in ATHENA DEMO by going to the HTN Advisory tab, selecting a patient and computing an advisory. The Advisory will contain the tabs with the updated information.
6. If the pages display as expected, then to update the server version of ATHENA-CDS SYSTEM, rename the old html pages in \\ATHENADSS\doc\ATHENA and add new pages to the folder.

IV.5.7. Prepare and Run Test Cases

This subsection is under construction. It will contain information on the offline testing of updates to the ATHENA-CDS Knowledge Base.

Access database for physician's review of cases of patient cases is located in vhapalATHENA/Comparison data STEVE & Nicki 022803/Comparison Table Expert Validation Study041400_NICKI_121302

IV.5.8. Update Deployed System

To update the knowledge base in the deployed system, substitute the following files in the \ATHENADSS\domain_model:

ATHENA_all.pins
ATHENA_all.pont
ATHENA_hypertension.pprj

IV.6. ATHENAeum Mapping

ATHENAeum Mapping Operations defined in Protégé are used to map vitals, demographic information, and laboratory-test terms used in the ATHENA-CDS Knowledge Base, to the terms used in the VA clinical information system [24]. The ATHENA-CDS SYSTEM makes only very limited use of available ATHENAeum Mapping Operations.

As described in Subsection IV.2, terms used in the ATHENA-CDS Knowledge Base (e.g., HDL_Cholesterol) must be mapped to those used in the VA clinical information system (e.g., HDL) so that the Guideline Interpreter can properly use VA data to evaluate eligibility and decision criteria that were written using knowledge-base terms. For patient problems that are coded using ICD9 in VistA, mapping makes the ICD9 codes into subclasses of guideline concepts used in ATHENA-CDS decision criteria (described in Subsection IV.2.2).

For prescriptions and refills, the Converter (Subsection II.2.2.2) normalizes the data into prescriptions for generic drugs (e.g., lisinopril) at doses expressed in mg/day (e.g., 10 mg/day). The necessary mapping information is spelled out in the Drugs table of the ATHENEON database (see Appendix V.1).

The Lab_Dictionary_Translation ATHENAeum mapping operation (Figure 164) makes it possible to specify—for each type of laboratory test result that is extracted from VistA and entered into the ATHENEON database—how a term used in the ATHENEON database (e.g., HDL) is mapped to a class in the Medical_Domain_Class hierarchy (e.g., the HDL_Cholesterol class; see Figure 165). The unit of measurement of the data to be used in ATHENA-CDS SYSTEM can be specified. However, that information is not used to normalize data to the chosen measurement unit. The system assumes that the data in the database are already expressed in that unit of measurement.

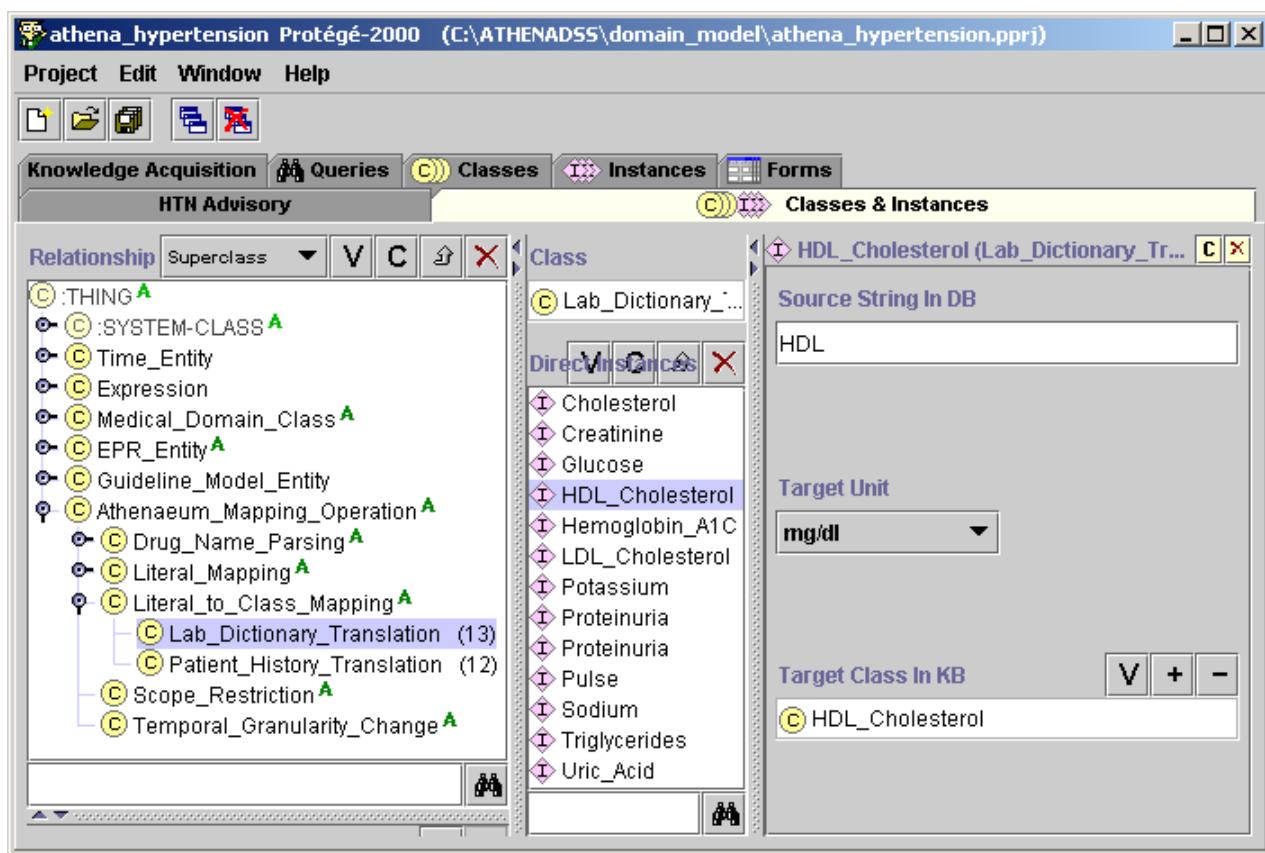


Figure 164 - Mapping of laboratory test terms to terms used in the ATHENA-CDS Hypertension (HTN) Knowledge Base

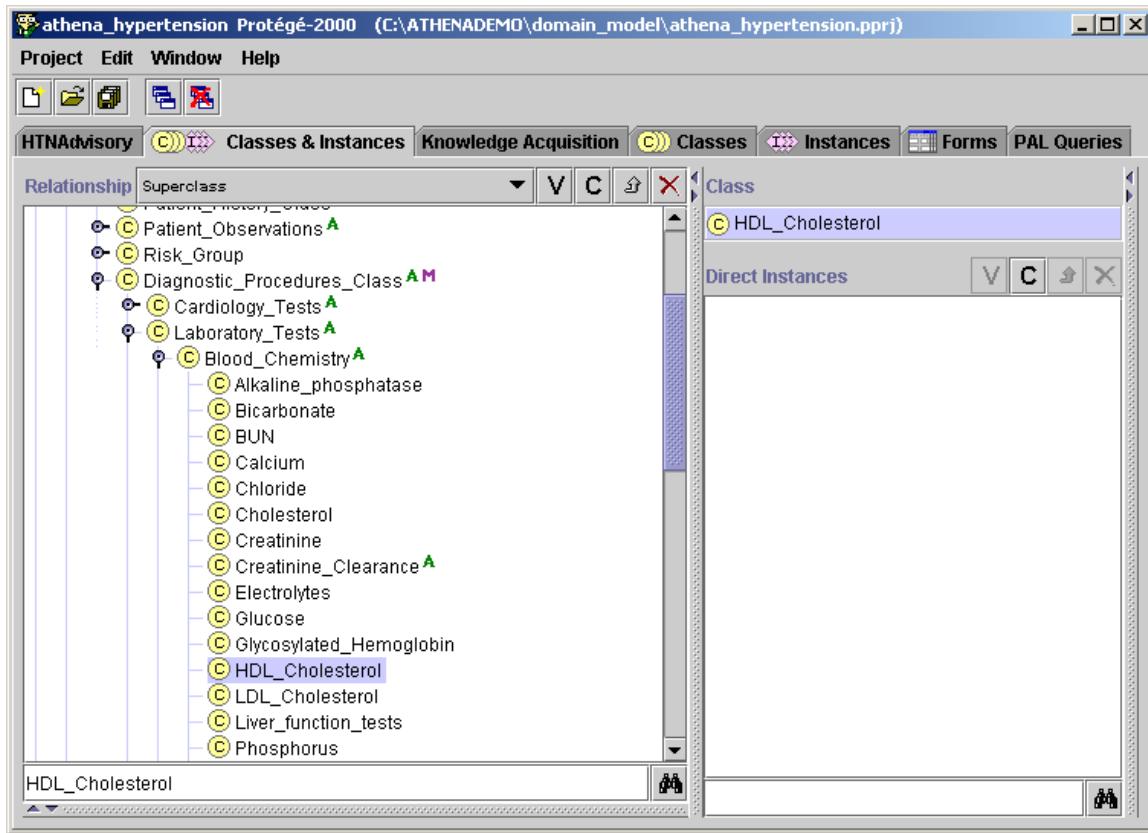


Figure 165 - Location of HDL_Cholesterol in the Medical_Domain_Class hierarchy

The Converter loads instances of the Lab_Dictionary_Translation and converts the information in these instance mappings to entries of the *LabMapping* table in the ATHENEON database.³¹

Similarly, Patient_History_Translation mapping (Figure 166) is used to specify the relationship between vitals and demographics terms in the ATHENEON database and terms in the ATHENA-CDS Knowledge Base. Thus, in Figure 166, the string “HISPANIC, BLACK” is mapped to the class Hispanic-Black in the Medical_Domain_Class hierarchy.

³¹ The Guideline Interpreter uses the LabMapping table to query for laboratory test results through the query: “temporal select target_class, value, unit from Studies, LabMapping where source_name = lab and SSN = XXX”. Source_name and target_class are the slots, Source String In DB and Target Class In KB, specified in instances of the Lab_Dictionary_Translation class. Lab, Value, and Unit are columns of the Studies table in the ATHENEON database.

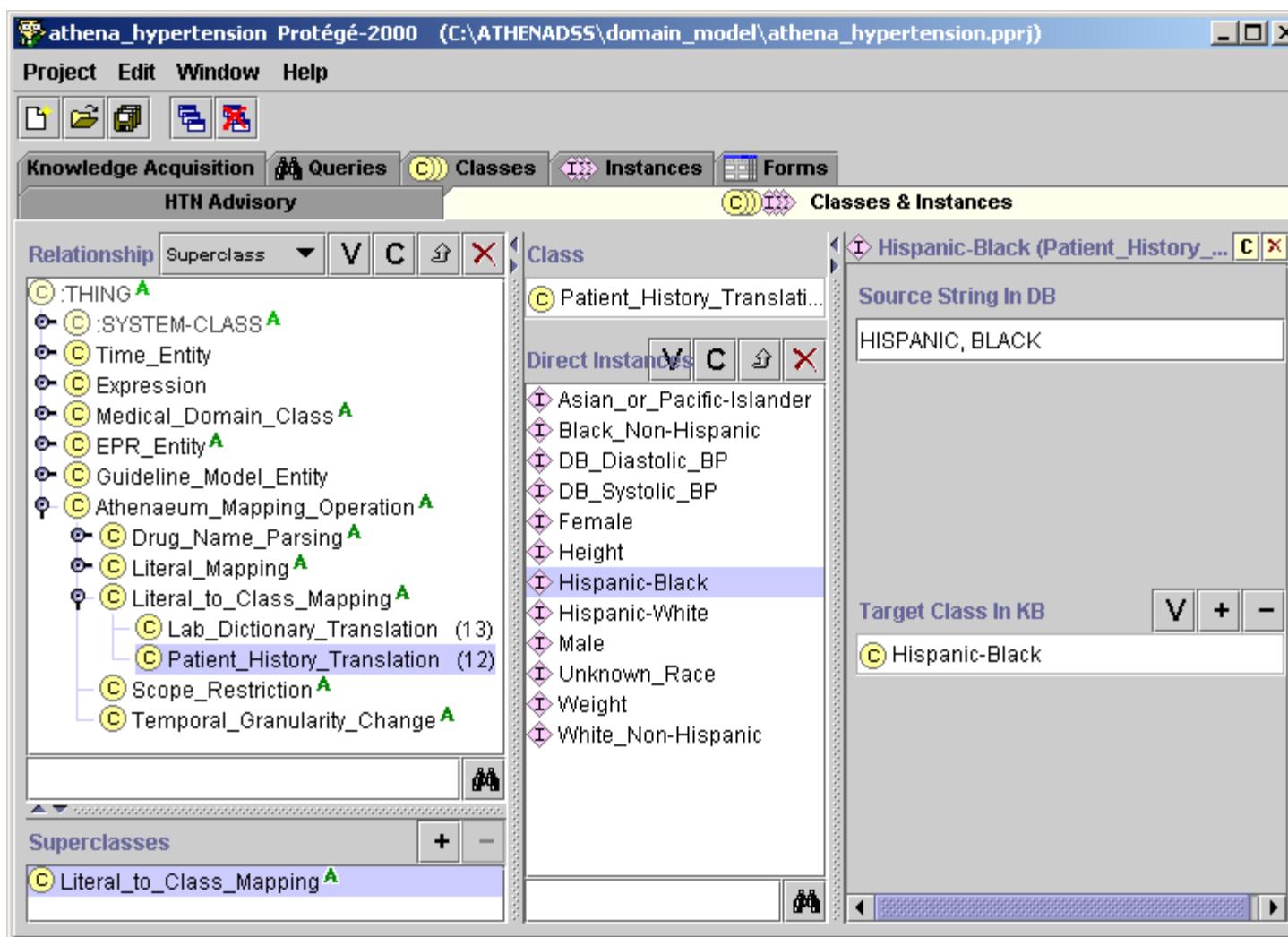


Figure 166 - Mapping of demographic information terms to terms used in the ATHENA-CDS Knowledge Base

Mappings can be many to one. Thus, the strings URINE PROTEIN and UR PROTEIN are mapped to the same Proteinuria class (Figure 167 and Figure 168). Data coded using either term would be indistinguishable to the Guideline Interpreter. Many-to-one mapping is done by creating multiple instances of the Lab_Dictionary_Translation class, where each instance has the same value (e.g., Proteinuria) for the Target Class In KB slot, but different values (e.g., UR PROTEIN and URINE PROTEIN) for the Source String In DB slot.

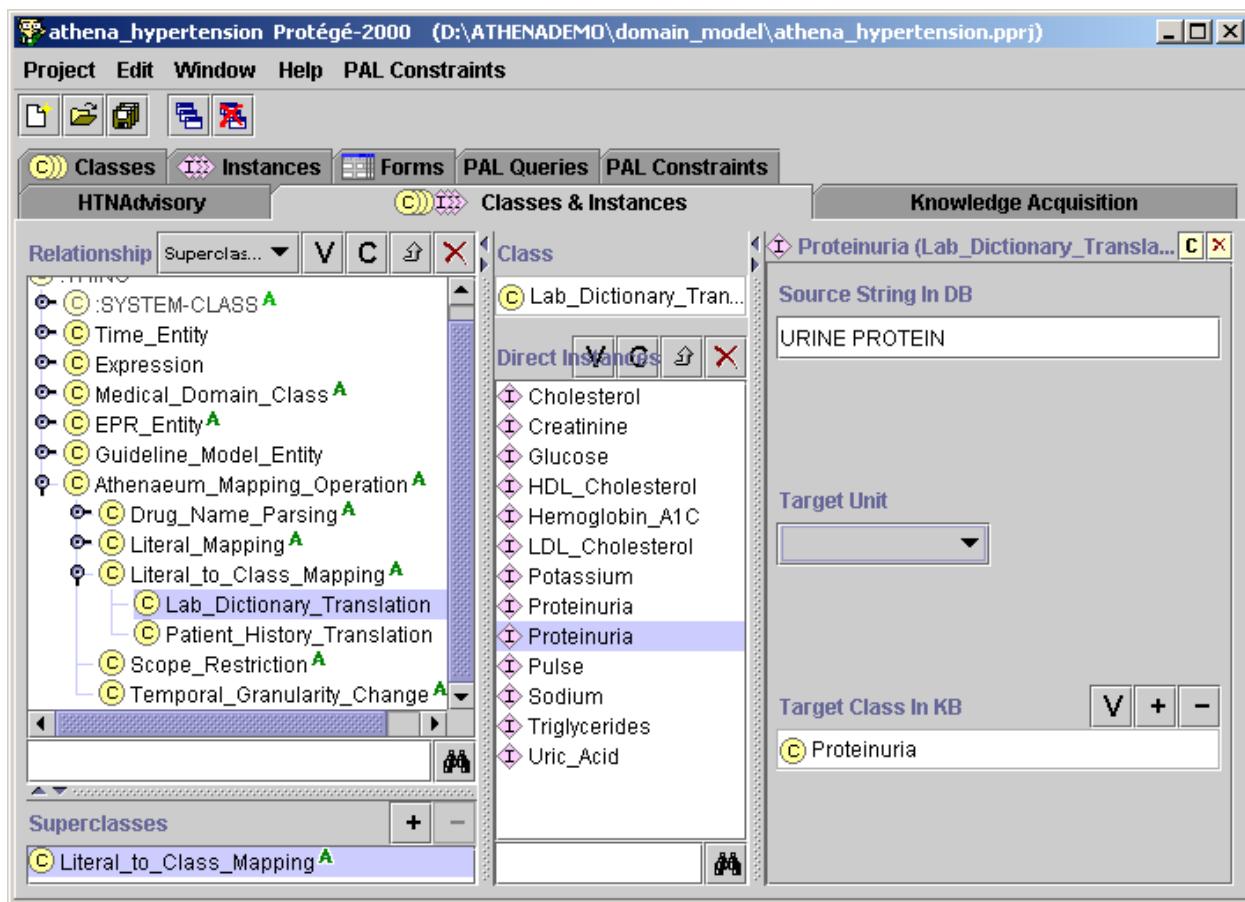


Figure 167 - Instance of `Lab_Dictionary_Translation` that maps the `Proteinuria` concept in the `Medical_Domain_Model` to the string “`URINE PROTEIN`” in the SQL database

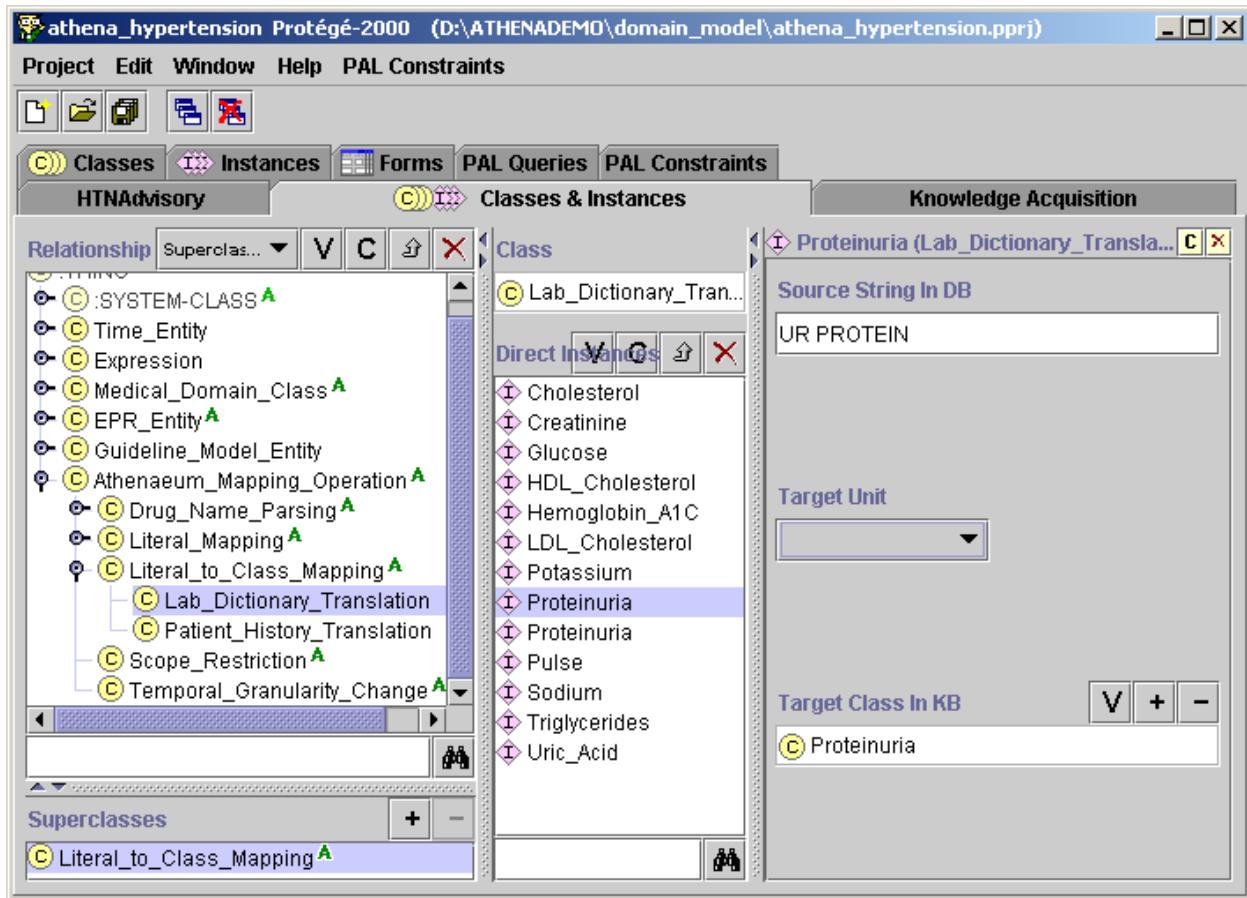


Figure 168 - Instance of Lab_Dictionary_Translation that maps the Proteinuria concept in the Medical_Domain_Model to the string “UR PROTEIN” in the SQL database

The Converter loads instances of the Patient_History_Translation and converts the information in these instance mappings into records of the PatientHistoryMapping table in the ATHENEON database.³²

³² The Guideline Interpreter uses the PatientHistoryMapping table to query for vitals and demographics information through the queries:

```
temporal select target_class, value from Vitals, tientHistoryMapping where
measurement = source_string and SSN = XXX
select DOB, target_class from Demographics, PatientHistoryMapping where sex =
source_string and SSN = XXX
select target_class from Demographics, PatientHistoryMapping where race =
source_string and SSN = XXX
```

Source_string and target_class are the slots, Source String in DB and Target Class in KB, specified in instances of the Patient_History_Translation class. Measurement and Value are columns in the Vitals table. DOB and Sex are columns in the Demographics table.

When lab, vitals, and demographics data elements different from those implemented in the ATHENACDS SYSTEM are extracted from VistA, or when ATHENA-CDS is installed at a location where the data dictionary is different from that of the original ATHENA-CDS site, instances of the Lab_Dictionary_Translation and Patient_History_Translation have to be updated.

V. Appendices

V.1. ATHENA-CDS and Atheneon DB Schema

Data Tables

Mapping Tables

Logging Tables

V.2. M Extraction Program

Available from ATHENA-CDS research team

V.3. SQL Transformation Code

Available from ATHENA-CDS research team

V.4. ATHENA-CDS Training Manual

Attached

V.5. ATHENA-CDS CONVERTOR

ATHENA-CDS Convertor

This document describes the functionality of the Convertor program used in the ATHENACDS SYSTEM Hypertension Advisory System [1]. This program takes a daily dump of the *Athena* relational database produced from the VA's M operational database and converts it to the *Atheneon* database used by the hypertension advisory system. This conversion takes the relatively low-level information contained in the ATHENA-CDS database and produces a set of normalized tables in the *Atheneon* database that have consistent representations for all relevant concepts contained in the tables. For example, all drug or laboratory test references in *Atheneon* use a standard nomenclature. An additional important functionality of the Convertor is that it produces tables that have a consistent representation of temporal information. This temporal information is represented using temporal tables supported by the Chronus II [2] temporal query system. Information on invoking the Convertor can be found in Appendix 1.

The ATHENA-CDS database contains both a set of daily dump tables (see Table 1) and a set of static tables (see Table 2). The daily dump tables are regenerated daily and populated with data for patients with hypertension who have an appointment within three days of the generation date. The static tables contain information to support the mapping process. They are generated once and do not typically change unless system updates are required.

Table 4. *Athena* daily dump tables. These tables are regenerated daily and populated with data for patients with hypertension who have an appointment within three days of the generation date.

Table	Description
ALLERGY	Contains a description of patients' allergic reactions, typically to either drugs or food.
DEMOGRAPHICS	Contains basic patient demographic information such as birth date, age, and sex.
PROVIDERS	Contains physician study group information.
LABS	Contains patient laboratory information.
VITALS	Contains patient vitals, such as blood pressure, height, and weight.
ENCOUNTER	Describes a scheduled visit for a patient; contains information such as visit date, location and provider.
DIAGNOSIS	Contains ICD9-encoded patient diagnoses.
PREScriptions	Contains detailed patient prescriptions.

Table 2. *Athena* static tables. These tables contain information to support the mapping process. They are generated once and do not typically change unless system updates are required.

Table	Description
LAB_LIMITS	Contains a list of laboratory tests to map and unit information for that test. If a test is not contained in the LAB_LIMITS table, it is not mapped to the <i>Atheneon</i> database.
COMORBID_BUILDER	Table that contains a mapping from the VA National Drug Identifier of a drug and its corresponding VA drug classification.
HTN_DRUG_TRANSLATION	Contains an exhaustive list of all drugs with their VA National Drug Identifier in addition to other VA-specific identifiers.
DRUGS	Table containing an exhaustive list of all drugs known by the Hypertension Advisory system. If a test is not contained in this table, it is not mapped to the <i>Atheneon</i> database.

The Convertor takes the daily dump and static tables in the ATHENA-CDS database and produces a set of normalized tables in the Atheneon database. In general, there is a one-to-one correspondence between the daily dump tables in the ATHENA-CDS database and the tables generated in the the Atheneon database (see Table 3).

Table 3. Convertor Table Mappings showing the correspondence between the *Athena* tables and their generated *Atheneon* tables.

Source <i>Athena</i> Table	Destination <i>Atheneon</i> Table
ALLERGY	ADVERSE_EVENTS
DEMOGRAPHICS	DEMOGRAPHICS
PROVIDERS	PROVIDERS
LABS	STUDIES
VITALS	VITALS
ENCOUNTER	ENCOUNTERS
DIAGNOSIS	CONDITIONS
PREScriptions	PREScriptions, PREScriptions_HISTORY

The Convertor also generates three advisory tables (see Table 4).

Table 4. Convertor *Atheneon* Advisory Tables.

Table	Description
DRUGCLASS	This is an intermediate table generated by the Convertor that lists the VA drug classes for each drug prescribed to a patient.
CONDITIONCLASS	This is an intermediate table generated by the Convertor that lists comorbidity classes for each condition listed for a patient (if any).
ADVISORIES	This table contains an advisory index for each patient indicating which advisory pertains to that patient.

Three mapping tables are also generated by the Convertor (see Tables 5).

Table 5. Convertor *Atheneon* Advisory Tables.

Table	Description
DRUGMAPPING	This table contains drug name to class name mappings. It is generated by the convertor from information contained in the Protégé Frames ontology supplied to it. These tables are used by the Hypertension Advisory system
LABMAPPING	This table contains laboratory name to class name mappings. It is generated by the convertor from information contained in the Protégé Frames ontology supplied to it. These tables are used by the Hypertension Advisory system
PATIENTHISTORYMAPPING	This table contains information to map patient history information to class names. It is generated by the convertor from information contained in the Protégé Frames ontology supplied to it. These tables are used by the Hypertension Advisory system

The next section describes the details of the mapping process for each table. It is followed by sections detailing the support tables used in this process.

Mapping from ATHENA-CDS to Atheneon Tables

This section outlines the mapping of daily dump *Athena* tables to their corresponding *Atheneon* tables. Table 3 lists the tables that are mapped.

Demographics

The *Athena* DEMOGRAPHICS table is mapped to the *Atheneon* DEMOGRAPHICS table. See Tables 6 and 7 for details of the mapping process.

Table 6. *Athena* DEMOGRAPHICS table.

Column	Mapping Notes
SSN	Patient Social Security Number. This column is mapped directly to the ID column in the <i>Atheneon</i> DEMOGRAPHICS table. If NULL, the row will be skipped and no corresponding row will be generated in the target table.
Patient	Name of the patient. This column is mapped directly to the Name column in the <i>Atheneon</i> DEMOGRAPHICS table. If NULL an empty string is generated in the Name column.
DOB	Name of the patient. This column is mapped directly to the Name column in the <i>Atheneon</i> DEMOGRAPHICS table. NULL values are allowed.
Sex	Sex of the patient. This column is mapped directly to the Sex column in the <i>Atheneon</i> DEMOGRAPHICS table. If NULL, the row will be skipped and no corresponding row will be generated in the target table.
Race	Race of the patient. This column is mapped directly to the Race column in the <i>Atheneon</i> DEMOGRAPHICS table. If NULL, the row will be skipped and no corresponding row will be generated in the target table.

Table 7. *Atheneon DEMOGRAPHICS* Table.

Column	Mapping Notes
SSN	Patient identifier. Mapped directly from the SSN column in the <i>Athena DEMOGRAPHICS</i> table.
Name	Patient name. Mapped directly from the Patient column in the <i>Athena DEMOGRAPHICS</i> table.
DOB	Date of birth of patient. Mapped directly from the DOB column in the <i>Athena DEMOGRAPHICS</i> table.
Race	Race of patient. Mapped directly from the Race column in the <i>Athena DEMOGRAPHICS</i> table.
Sex	Sex of patient. Mapped directly from the Sex column in the <i>Athena DEMOGRAPHICS</i> table.

Providers

The *Athena* PROVIDERS table is mapped to the *Atheneon* PROVIDERS table. See Tables 8 and 9 for details of the mapping process.

Table 8. *Athena* PROVIDERS table.

Column	Mapping Notes
SSN	Provider identifier. This column is mapped directly to the ID column in the <i>Atheneon</i> PROVIDERS table. If NULL, the row will be skipped and no corresponding row will be generated in the target table.
Name	Name of the provider. This column is mapped directly to the Name column in the <i>Atheneon</i> PROVIDERS table. If NULL an empty string is generated in the Name column.
StudyGroup	Study group name. This column is mapped directly to the StudyGroup column in the <i>Atheneon</i> PROVIDERS table. If NULL, the row will be skipped and no corresponding row will be generated in the target table.

Table 9. *Atheneon* PROVIDERS table.

Column	Mapping Notes
ID	Provider identifier. Mapped directly from the SSN column in the LABS table.
Name	Name of the provider. This column is mapped directly from the Name column in the <i>Athena</i> PROVIDERS table.
StudyGroup	Study group name. This column is mapped directly from the StudyGroup column in the <i>Athena</i> PROVIDERS table.

Studies

The *Athena* LABS table is mapped to the *Atheneon* STUDIES Chronus II temporal event table. The LAB_LIMITS table is used to limit the mapped laboratory test set. If a laboratory test is not listed in the **Lab** column of the LAB_LIMITS table it is not transferred from the *Athena* LABS table to the *Atheneon* STUDIES table. The LAB_LIMITS table is also used to select the appropriate units for the mapped laboratory test. See Tables 10 and 11 for details of the mapping process.

Table 10. *Athena* LABS Table.

Column	Mapping Notes
SSN	Patient Social Security Number. This column is mapped directly to the ID column in the STUDIES table. If NULL, the row will be skipped and no corresponding row will be generated in the STUDIES table.
Lab	Name of laboratory test. This column is mapped directly to the Lab column of the STUDIES table if an entry is present in the Lab column in the LAB_LIMITS table. If no entry is present, no mapping takes place. If NULL, the row will be skipped and no corresponding row will be generated in the STUDIES table.
Value	Laboratory value. If NULL, the row will be skipped and no corresponding row will be generated in the STUDIES table.
LabDate	The date the laboratory value was recorded. This column is mapped to the valid instant time of the STUDIES table. If NULL, the row will be skipped and no corresponding row will be generated in the STUDIES table.

Table 11. *Atheneon* STUDIES Chronus II temporal event table.

Column	Mapping Notes
SSN	Patient Social Security Number. Mapped directly from the SSN column in the LABS table.
Lab	Name of laboratory test. This column is mapped directly from the Lab column in the LABS table if an entry is present in the Lab column in the LAB_LIMITS table.
Value	Laboratory value. Mapped directly from Value column in LABS table.
Unit	Unit of laboratory value. Mapped from the Units column in the LABS_LIMITS table.
Valid Time (Instant)	Date of recording of laboratory value. Mapped from LabDate column in LABS table.

Vitals

The *Athena* VITALS table is mapped to the *Atheneon* VITALS Chronus II temporal event table. The *Athena* VITALS table may contain more than two readings per row, in which case two rows are generated in the *Atheneon* VITALS table, one for each reading. The only vitals that are currently understood by the convertor are blood pressure, weight and height and pulse. See Tables 12 and 13 for details of the mapping process.

Table 12. *Athena* VITALS Table.

Column	Mapping Notes
SSN	Patient Social Security Number. This column is mapped directly to the ID column in the <i>Atheneon</i> VITALS table. If NULL, the row will be skipped and no corresponding row will be generated in the VITALS table.
VDate	The date the vital is recorded. This column is mapped to the valid instant time of the <i>Atheneon</i> VITALS table. If NULL, the row will be skipped and no corresponding row will be generated in the target table.
Vital	The name of the vital. If this contains the value ‘blood pressure’ then the Value1 column contains the systolic pressure and the Value2 column contains the diastolic blood pressure; for the value ‘weight/height’ the Value1 column contains the weight and the Value2 column contains the height; for the value ‘pulse’ the Value1 column contains the pulse and the Value2 column is ignored. If any other values (including NULL) are present, the row will be skipped and no corresponding row will be generated in the target table. Two <i>Atheneon</i> VITALS rows are generated for blood pressure and weight and height combinations, and one row for each pulse value. The value of the Measurement column in these rows is set to one of ‘Systolic_Blood_Pressure’, ‘Diastolic_Blood_Pressure’, ‘weight’, ‘height’ or ‘pulse’, as appropriate
Value1	First vital value. Mapped to Value column in <i>Atheneon</i> VITALS table. See comments for Vital column for notes on this mapping.
Unit1	First vital unit. Mapped to Units column in <i>Atheneon</i> VITALS table. See comments for Vital column for notes on this mapping.
Value2	Second vital value. Mapped to Value column in <i>Atheneon</i> VITALS table. See comments for Vital column for notes on this mapping.
Unit2	Second vital unit. Mapped to Units column in <i>Atheneon</i> VITALS table.

	VITALS table. See comments for Vital column for notes on this mapping.
--	---

Table 13. *Atheneon* VITALS Chronus II temporal event table.

Column	Mapping Notes
ID	Patient Social Security Number. Mapped directly from the SSN column in the <i>Athena</i> VITALS table.
Measurement	Contains the name of the measurement. Will be one of ‘Systolic_Blood_Pressure’, ‘Diastolic_Blood_Pressure’, ‘weight’, ‘height’ or ‘pulse’. See comments for Vital column in Table 12 for notes on this mapping.
Units	Units of vital. Mapped from Unit1 or Unit2 columns in <i>Athena</i> VITALS table. See comments for Vital column in Table 12 for notes on this mapping. See comments for Vital column in Table 12 for notes on this mapping.
Value	Value of vital. Mapped from Value1 or Value2 columns in <i>Athena</i> VITALS table. See comments for Vital column in Table 12 for notes on this mapping.
Valid Time (Instant)	Date of recording of vital. Mapped from VDate column in LABS table.

Encounters

The *Athena* ENCOUNTER table is mapped to the *Atheneon* ENCOUNTERS Chronus II temporal event table. See Tables 14 and 15 for details of the mapping process.

Table 14. *Athena* ENCOUNTER Table.

Column	Mapping Notes
SSN	Patient Social Security Number. This column is mapped directly to the ID column in the <i>Atheneon</i> ENCOUNTERS table. If NULL, the row will be skipped and no corresponding row will be generated in the target table.
ApptDatetime	The date the appointment. This column is mapped to the valid instant time of the <i>Atheneon</i> ENCOUNTERS table. If NULL, the row will be skipped and no corresponding row will be generated in the target table.
Clinic	Name of the clinic. This column is mapped directly to the Clinic column in the <i>Atheneon</i> ENCOUNTERS table. If NULL an empty string is generated in the Clinic column.
Provider	Provider identifier. This column is mapped directly to the Provider column in the <i>Atheneon</i> ENCOUNTERS table. If NULL an empty string is generated in the Provider column.

Table 15. *Atheneon* ENCOUNTERS Chronus II temporal event table.

Column	Mapping Notes
ID	Patient identifier (SSN). Mapped from SSN column in <i>Athena</i> ENCOUNTER table.
Clinic	Clinic name. Mapped directly from Clinic column in <i>Athena</i> ENCOUNTER table.
Provider	Provider identifier. Mapped directly from Provider column in <i>Athena</i> ENCOUNTER table.
Valid Time (Instant)	Date of recording of vital. Mapped from ApptDatetime column in <i>Athena</i> ENCOUNTER table.

Conditions

The *Athena* DIAGNOSIS table is mapped to the *Atheneon* CONDITIONS Chronus II temporal state table. See Tables 16 and 17 for details of the mapping process.

Table 16. *Athena* DIAGNOSIS Table.

Column	Mapping Notes
SSN	Patient Social Security Number. This column is mapped directly to the ID column in the CONDITIONS table. If NULL, the row will be skipped and no corresponding row will be generated in the CONDITIONS table.
ICD9	ICD9 code of condition. Mapped to ICD9 column in CONDITIONS table. If NULL, the row will be skipped and no corresponding row will be generated in the target table.
FirstDate	First recorded date of condition. Mapped to the opening instant of the valid period of the CONDITIONS table. If NULL, a date of ‘1900-01-01’ is generated.
LastDate	Last recorded date of condition. Mapped to the Last_Date column of the CONDITIONS table. May be NULL.

Table 17. *Atheneon* CONDITIONS Chronus II temporal event table.

Column	Mapping Notes
ID	Patient identifier (SSN). Mapped from SSN column in DIAGNOSIS table.
ICD9	ICD9 code of condition. Mapped from ICD9 column in DIAGNOSIS table.
Last_Date	Last recorded date of condition. Mapped from LastDate column in DIAGNOSIS table.
Valid Time (Period)	Valid interval of condition. The start instant is mapped from FirstDate column in DIAGNOSIS table. The finish instant is ongoing. In other words, once a condition is recorded it is assumed to continue indefinitely.

Adverse Drug Reactions

The *Athena*.ALLERGY table is mapped to the *Atheneon*.ADVERSE_EVENTS Chronus II temporal event table. The ALLERGY table may contain more than two adverse reactions per row, in which case two rows are generated in the ADVERSE_EVENTS table, one for each reaction. See Tables 18 and 19 for details of the mapping process.

Table 18. *Athena*.ALLERGY Table.

Column	Mapping Notes
SSN	Patient Social Security Number. This column is mapped directly to the ID column in the ADVERSE_EVENTS table. If NULL, the row will be skipped and no corresponding row will be generated in the ADVERSE_EVENTS table.
Drug	Drug name. This column and it is mapped directly to the Substance column in the ADVERSE_EVENTS table. May be NULL.
FileNo	File number of drug in National Drug Identifier lookup table. In association with the RecNo and VAClass columns, this entry is used to look up the <i>Nation Drug Identifier</i> of the drug using the <i>Athena</i> .HTN_DRUG_TRANSLATION table. If reaction is not to a drug then this cell will be empty. Only file numbers 50.416, 50.6 or 50.605 are currently mapped. <i>National Drug Identifiers</i> are mapped to the DrugID column in the ADVERSE_EVENTS table. If this field contains another value then no mapping to a <i>National Drug Identifier</i> will be generated.
RecNo	Record number of drug in <i>National Drug Identifier</i> lookup table. Used in association with FileNo and VAClass columns to look up <i>National Drug Identifier</i> of drug using the <i>Athena</i> .HTN_DRUG_TRANSLATION table. If reaction is not to a drug then this cell will be NULL.
VAClass	VA classification of drug in <i>National Drug Identifier</i> lookup table. Used in association with FileNo and RecNo to look up <i>National Drug Identifier</i> of drug using the <i>Athena</i> .HTN_DRUG_TRANSLATION table. If reaction is not to a drug then the value will be NULL.
ADate	Date of reaction. This column is mapped to the temporal event time in the ADVERSE_EVENTS table. If NULL, this row will be skipped and no corresponding row will be generated in the ADVERSE_EVENTS table.
Reaction1	Primary reaction. This column is mapped directly to the Reaction column in the ADVERSE_EVENTS table. May be NULL.
Reaction2	Secondary reaction. If a secondary reaction is present, a new row is generated in the ADVERSE_EVENTS table and the value is mapped directly to the Reaction column in that table. May be NULL.
Hist_Obs	This column is mapped directly to the Hist_Obs column in the ADVERSE_EVENTS table. May be NULL.
Type	Reaction type. This column is mapped to the Type column in the ADVERSE_EVENTS table. Values 'O', 'D', and 'F' are mapped to values 'OTHER', 'DRUG', and 'FOOD', respectively; all other values are

	mapped directly. May be NULL.
Mechanism	This column is mapped directly to the Mechanism column in the ADVERSE_EVENTS table. May be NULL.

Table 19. *Atheneon.ADVERSE_EVENTS* Chronus II Temporal Event Table.

Column	Mapping Notes
ID	Patient identifier (SSN). Mapped from SSN column in ALLERGY table.
Substance	Substance. Mapped from Drug column in ALLERGY table.
DrugID	National Drug Identifier. Mapped from FileNo , RecNo , and VAClass columns in ALLERGY table.
VAClass	VA class of drug. Mapped from VAClass column in ALLERGY table.
Reaction	Reaction. Mapped from Reaction1 or Reaction2 columns in ALLERGY table.
Hist_Obs	Mapped from Hist_Obs column in ALLERGY table.
Type	Type of reaction. Mapped from Type column in ALLERGY table.
Mechanism	Mechanism of reaction. Mapped from Mechanism column in ALLERGY table.
ValidTime (Instant)	Date of reaction. Mapped from ADate column in ALLERGY table.

Prescriptions

The *Athena* PRESCRIPTIONS table is mapped to the *Atheneon* PRESCRIPTIONS and PRESCRIPTIONSHISTORY Chronus II temporal state table.

The *Athena* PRESCRIPTIONS table in the daily dump contains an entire prescription history for each patient. The *Atheneon* PRESCRIPTIONS table is filtered to include only currently active prescriptions. The *Athena* PRESCRIPTIONS table's **Status**, **Days_Supply**, **ExpDate**, and **Issue_Date** fields are used to perform this filtering. If the status is listed as 'ACTIVE' or 'SUSPENDED' and the patient still has a current supply of drugs (as determined by comparing the current data with the days supply and issue date) then the prescription is mapped. Prescriptions with a status of 'EXPIRED' are also mapped if the patient has a current supply of drugs. If the current date is beyond the drug's expiry date then it is not mapped.

The *Atheneon* PRESCRIPTIONSHISTORY table is not filtered and includes all prescriptions. Values in the PRESCRIPTIONSHISTORY table are also post-processed so that adjacent or overlapping prescriptions of the same drug are coalesced into single intervals.

See Tables 20, 21 and 22 for column-by-column details of the mapping process.

Table 20. *Athena* PRESCRIPTIONS table.

Column	Mapping Notes
SSN	Patient Social Security Number. This column is mapped directly to the ID column in the <i>Atheneon</i> PRESCRIPTIONS and PRESCRIPTIONSHISTORY tables. If NULL, the row will be skipped and no corresponding row will be generated in the target tables.
RXNo	Unique prescription number. This column is mapped directly to the RXNo column in the <i>Atheneon</i> PRESCRIPTIONS and PRESCRIPTIONSHISTORY tables. If NULL, the row will be skipped and no corresponding row will be generated in the target tables.
Issue_Date	Issue date of the drug. This column is mapped to the temporal event time in the <i>Atheneon</i> PRESCRIPTIONS and PRESCRIPTIONSHISTORY tables. As described above, it is also used to determine if a prescription is still active and should be mapped to the <i>Atheneon</i> PRESCRIPTIONS table. If NULL, this row will be skipped and no corresponding row will be generated in the target tables.
Provider	Provider identifier. This column is mapped directly to the Provider column in the <i>Atheneon</i> PRESCRIPTIONS and PRESCRIPTIONSHISTORY tables. If NULL, the row will be skipped and no corresponding row will be generated in the target tables.
NatDrugID	VA National Drug Identifier of the prescribed drug. This column is mapped directly to the DrugID column in the <i>Atheneon</i> PRESCRIPTIONS and PRESCRIPTIONSHISTORY tables. If NULL, the row will be skipped and no corresponding row will be generated in the target tables.
Name	Free text name of the prescribed drug. This column is mapped directly to the Ingredient column in the <i>Atheneon</i> PRESCRIPTIONS and PRESCRIPTIONSHISTORY tables. If NULL, the row will be skipped and no corresponding row will be generated in the target tables.
DisplayName	Display name of the prescribed drug. This column is mapped directly to the DisplayName column in the <i>Atheneon</i> PRESCRIPTIONS and PRESCRIPTIONSHISTORY tables. If NULL, the row will be skipped and no corresponding row will be generated in the target tables.
Qty	Quantity of the prescribed drug. This column is mapped directly to the Quantity column in the <i>Atheneon</i> PRESCRIPTIONS and PRESCRIPTIONSHISTORY tables. It is also used in the calculation of the daily dosage of the drug with the result mapped to the Daily_Dose field. If NULL, the row will be skipped and no

	corresponding row will be generated in the target tables.
Sig	This column is ignored and is not mapped.
Division	Division where the drug was prescribed. This column is mapped directly to the Last_Fill_Date column in the <i>Atheneon PRESCRIPTIONS</i> and <i>PREScriptionshistory</i> tables. May be NULL.
Dispense_Date	Date the drug was dispensed. This column is mapped directly to the Division column in the <i>Atheneon PRESCRIPTIONS</i> and <i>PREScriptionshistory</i> tables. May be NULL.
ExpDate	Date the drug will expire. This column is mapped directly to the Expire_Date column in the <i>Atheneon PRESCRIPTIONS</i> and <i>PREScriptionshistory</i> tables. As described above, it is also used to exclude prescriptions with expired drugs from the <i>Atheneon PRESCRIPTIONS</i> table. May be NULL.
LastFill_Date	Date the prescription was last filled. This column is mapped directly to the Last_Fill_Date column in the <i>Atheneon PRESCRIPTIONS</i> and <i>PREScriptionshistory</i> tables. May be NULL.
Days_Supply	Number of days supply of the prescribed drug. This column is mapped directly to the Days_Supply column in the <i>Atheneon PRESCRIPTIONS</i> and <i>PREScriptionshistory</i> tables. It is also used in the calculation of the daily dosage of the drug with the result mapped to the Daily_Dose field. If NULL, the row will be skipped and no corresponding row will be generated in the target tables.
Dosage	Dosage of the prescribed drug. This column is not mapped directly to the <i>Atheneon PRESCRIPTIONS</i> and <i>PREScriptionshistory</i> tables. Instead it is used in the calculation of the daily dosage of the drug with the result mapped to the Daily_Dose field. If NULL, the row will be skipped and no corresponding row will be generated in the target tables.
Status	This column is not mapped to the target table. However, as described above, it is used in determining if a prescription should be mapped to the <i>Atheneon PRESCRIPTIONS</i> table.
Station	Station where the drug was prescribed. This column is mapped directly to the Station column in the <i>Atheneon PRESCRIPTIONS</i> and <i>PREScriptionshistory</i> tables. If NULL, an empty string is supplied as a value for the Station column.
Fill_Type	Fill type of the prescription. This column is mapped directly to the Fill_Type column in the <i>Atheneon PRESCRIPTIONS</i> and <i>PREScriptionshistory</i> tables. If NULL, the row will be skipped and no corresponding row will be generated in the target tables.

Table 21. *Atheneon* PRESCRIPTIONS Chronus II temporal state table.

Column	Mapping Notes
ID	Patient identifier (SSN). Mapped from SSN column in <i>Athena</i> PRESCRIPTIONS table.
RXNo	Unique prescription number. Mapped from RXNo column in <i>Athena</i> PRESCRIPTIONS table.
Dispense_Date	Dispense date. Mapped from Dispense_Date column in <i>Athena</i> PRESCRIPTIONS table.
Provider	Provider identifier. Mapped from Provider column in <i>Athena</i> PRESCRIPTIONS table.
DrugID	VA National Drug Identifier of the prescribed drug. This column is mapped from the NatDrugID column in the <i>Athena</i> PRESCRIPTIONS table.
Ingredient	Free text name of prescribed drug. Mapped from Name column in <i>Athena</i> PRESCRIPTIONS table.
DisplayName	Display name of prescribed drug. Mapped from DisplayName column in <i>Athena</i> PRESCRIPTIONS table.
Quantity	Quantity of prescribed drug. Mapped from Qty column in <i>Athena</i> PRESCRIPTIONS table.
Sig	Comment field with values indicating how prescription was generated from the <i>Athena</i> PRESCRIPTIONS table. Possible values are ‘recently refilled; expired’, ‘active and not refilled’, and ‘active and currently filled’.
Division	Division prescribing the drug. Mapped from Division column in <i>Athena</i> PRESCRIPTIONS table.
Expire_Date	Date the prescribed drug will expire. Mapped from ExpDate column in <i>Athena</i> PRESCRIPTIONS table.
Last_Fill_Date	Date the prescription was last filled. Mapped from LastFill_Date column in <i>Athena</i> PRESCRIPTIONS table.
Days_Supply	Days supply of the prescribed drug. Mapped from Days_Supply column in <i>Athena</i> PRESCRIPTIONS table.
Daily_Dose	Daily dosage of prescribed drug. Calculated using the Qty , Days_Supply and Dosage columns in the <i>Athena</i> PRESCRIPTIONS table.
Station	Station prescribing the drug. Mapped from Station column in <i>Athena</i> PRESCRIPTIONS table.
Fill_Type	Fill type of the prescription. Mapped from Fill_Type column in <i>Athena</i> PRESCRIPTIONS table.
ValidTime (Instant)	Date of prescription. Mapped from Issue_Date column in <i>Athena</i> PRESCRIPTIONS table.

Table 22. *Atheneon PRESCRIPTIONSHISTORY Chronus II temporal state table.*

Column	Mapping Notes
ID	Patient identifier (SSN). Mapped from SSN column in <i>Athena PRESCRIPTIONS</i> table.
RXNo	Unique prescription number. Mapped from RXNo column in <i>Athena PRESCRIPTIONS</i> table.
Dispense_Date	Drug dispense date. Mapped from Dispense_Date column in <i>Athena PRESCRIPTIONS</i> table.
Provider	Provider identifier. Mapped from Provider column in <i>Athena PRESCRIPTIONS</i> table.
DrugID	VA National Drug Identifier of the prescribed drug. This column is mapped from the NatDrugID column in the <i>Athena PRESCRIPTIONS</i> table.
Ingredient	Free text name of prescribed drug. Mapped from Name column in <i>Athena PRESCRIPTIONS</i> table.
DisplayName	Display name of prescribed drug. Mapped from DisplayName column in <i>Athena PRESCRIPTIONS</i> table.
Quantity	Quantity of prescribed drug. Mapped from Qty column in <i>Athena PRESCRIPTIONS</i> table.
Sig	Is always NULL.
Division	Division prescribing the drug. Mapped from Division column in <i>Athena PRESCRIPTIONS</i> table.
Expire_Date	Date the prescribed drug will expire. Mapped from ExpDate column in <i>Athena PRESCRIPTIONS</i> table.
Last_Fill_Date	Date the prescription was last filled. Mapped from LastFill_Date column in <i>Athena PRESCRIPTIONS</i> table.
Days_Supply	Days supply of the prescribed drug. Mapped from Days_Supply column in <i>Athena PRESCRIPTIONS</i> table.
Daily_Dose	Daily dosage of prescribed drug. Calculated using the Qty , Days_Supply and Dosage columns in the <i>Athena PRESCRIPTIONS</i> table.
Station	Station prescribing the drug. Mapped from Station column in <i>Athena PRESCRIPTIONS</i> table.
Fill_Type	Fill type of the prescription. Mapped from Fill_Type column in <i>Athena PRESCRIPTIONS</i> table.
ValidTime (Instant)	Date of prescription. Mapped from Issue_Date column in <i>Athena PRESCRIPTIONS</i> table.

Advisory Tables

The advisory tables were used in the early stages of the ATHENACDS SYSTEM Hypertension Advisory project to generate an advisory for a patient based on a set of pre-canned criteria. They are no longer used, though the Convertor does generate an appropriate index of an advisory for each patient and populated the ADVISORY table in the *Atheneon* database. See Tables 23-25 for a description of the content of these tables.

Table 23. Convertor *Atheneon* DRUGCLASS table. This is an intermediate table generated by the convertor that lists the VA drug classes for each drug prescribed to a patient.

Table	Description
SSN	SSN of the patient.
Type	VA drug classification of drug prescribed to patient.

Table 24. Convertor *Atheneon* CONDITIONCLASS table. This is an intermediate table generated by the convertor that lists comorbidity classes for each condition listed for a patient (if any).

Table	Description
SSN	SSN of the patient.
Type	Coborbidity classification of each condition listed for patient.

Table 25. Convertor *Atheneon* ADVISORIES table. This tables contains an advisory index for each patient indicating which advisory pertains to that patient.

Table	Description
SSN	SSN of the patient.
CommentIndex	Index of advisory comment for that patient.

Mapping Tables

These tables are generated by the Convertor using information in a Protégé Frames ontology that is passed to it (see Appendix 1). They are used internally by the Hypertension Advisory system. See Tables 26-28 for a description of the content of these tables.

Table 26. Convertor *Atheneon* DRUGMAPPING Table. This table is generated from instances of the Drug_Dictionary_Translation class in the Frames ontology supplied to the converter. For each instance's Source_String_in_DB slot value a row is generated with the **SourceDescription** column populated with that value.

Table	Description
SourceDescription	Value extracted from the Source_String_in_DB slot value of each instance of the Drug_Dictionary_Translation class in the Frames ontology supplied to the Convertor.

Table 27. Convertor *Atheneon* LABMAPPING Table. This table is generated from instances of the Lab_Dictionary_Translation class in the Frames ontology supplied to the converter. For each instance's Source_String_in_DB, Target_Unit, and Target_Class_in_KB slot value a row is generated with the **Source_Name**, **Target_Class** and **Target_Unit** columns populated with these value

Table	Description
Source_Name	Value extracted from the Source_String_in_DB slot value of each instance of the Lab_Dictionary_Translation class in the Frames ontology supplied to the Convertor.
Target_Class	Value extracted from the Target_Class_in_KB slot value of each instance of the Lab_Dictionary_Translation class in the Frames ontology supplied to the Convertor.
Target_Unit	Value extracted from the Target_Unit slot value of each instance of the Lab_Dictionary_Translation class in the Frames ontology supplied to the Convertor.

Table 28. Convertor *Atheneon* PATIENTHISTORYMAPPING Table. This table is generated from instances of the Patient_History_Translation class in the Frames ontology supplied to the converter. For each instance's Source_String_in_DB and Target_Class_in_KB slot value a row is generated with the **Source_String** and **Target_Unit** columns populated with these value

Table	Description
Source_String	Value extracted from the Source_String_in_DB slot value of each instance of the Patient_History_Translation class in the Frames ontology supplied to the Convertor.
Target_Class	Value extracted from the Target_Class_in_KB slot value of each instance of the Patient_History_Translation class in the Frames ontology supplied to the Convertor.

ATHENA-CDS Static Tables

These tables are statically created and do not change during each daily dump. They are used internally by the Hypertension Advisory system. See Tables 29-32 for a description of the content of these tables.

Table 29. *Athena LAB_LIMITS* table. Used to limit the laboratory test mapped from the *Athena* database to the *Atheneon* database. Also supplies units information for the mapped tests.

Column	Description
Lab	Contains the name of a laboratory test to be mapped. If a test is not named in the column then it is not mapped from the <i>Athena</i> database to the <i>Atheneon</i> database.
Units	The <i>Athena</i> LABS tables does not contain the units of the laboratory tests it contains. This column is used to generate the appropriate units for that test in the <i>Atheneon</i> STUDIES table.

Table 30. *Athena COMORBID_BUILDER* table. Table that contains a mapping from the VA National Drug Identifier of a drug and its corresponding VA drug classification.

Column	Description
NatID	VA National Drug identifier.
VAClass	VA classification of drug.

Table 31. *Athena* and *Atheneon* DRUGS table. Contains an exhaustive list of all drugs that are to be mapped from the *Athena* database to the *Atheneon* database. If a drug is not listed in this table then it is not mapped to the *Atheneon* database. Appendix 2 contains the SQL script that can be used to generate the contents of this table. A copy of this table is required in both the *Athena* and the *Atheneon* databases.

Column	Description
ID	The identifier of the drug.
VAClass	The VA classification of the drug.
Description	A free text description of the drug.
DisplayName	Text that is displayed to physicians on by graphical applications.
Dosage	Dosage of drug.
Delivery	Delivery mechanism of drug.

Table 32. *Athena* and *Atheneon* HTN_DRUG_TRANSLATION table. Contains an exhaustive list of all drugs with their VA *National Drug Identifier* in addition to other VA-specific identifiers. Appendix 4 contains a SQL script that can be used to generate this table. A copy of this table is required in both the *Athena* and the *Atheneon* databases. See Table 19 for a description of how these values are used.

Column	Description
NatID	VA National Drug Identifier.
N50_ID	The VA N50 identifier of the drug.
VAClass	A free text description of the drug.
N50_NAME	The VA N50 name of the drug.
Drug	Name of drug.
Dosage	Dosage of drug.
Delivery	Delivery mechanism of drug.
N50_6_ID	The VA file number 50.6 identifier of the drug.
N50_6_NAME	The VA file number 50.6 name of the drug.
N50_416_ID	The VA file number 50.416 identifier of the drug.
N50_416_NAME	The VA file number 50.416 name of the drug.
N50_605_ID	The VA file number 50.605 identifier of the drug.
N50_605_NAME	The VA file number 50.605 name of the drug.

References

[1] M. K. Goldstein, B. B. Hoffman, R. W. Coleman, S. W. Tu, M. J. O'Connor, S. B. Martins, A. Advani, R. D. Shankar, M. A. Musen. (2001). *Patient Safety in Guideline-Based Decision Support for Hypertension Management: ATHENA DSS*. AMIA Annual Symposium, Washington, DC.

[2] M. J. O'Connor, S. W. Tu, M. A. Musen (2002). *The Chronus II Temporal Database Mediator*. AMIA Annual Symposium, San Antonio, TX.

Appendix 1. Invoking the Convertor.

The convertor code is contained in the Java class `ConvertorII`, which is in the package `edu.stanford.smi.Athena`. It can be invoked as follows:

```
java edu.stanford.smi.Athena.ConvertorII <initialization_file>
```

The initialization file is a standard properties file and contains the following entries:

`CONVERTOR_SOURCE_DATABASE` The ODBC URI of the *Athena* database.

`CONVERTOR_TARGET_DATABASE` The ODBC URI of the *Atheneon* database.

`CONVERTOR_SOURCE_USER` The user ID for the *Athena* database JDBC connection.

`CONVERTOR_TARGET_USER` The user ID for the *Atheneon* database JDBC connection.

`CONVERTOR_DEBUG_LEVEL` The debug level for the Convertor. 0 is minimal information; 1 is an intermediate amount; and 2 is verbose debugging information.

`CONVERTOR_PROTEGE_FILENAME` The file name of the Protégé Frames ontology used by the Convertor.

V.6. Glossary

Advisory

A set of recommendations. More specifically, in this manual it refers to the decision output (advisories) generated by the Guideline Interpreter.

ATHENA-CDS Controller

Also the Event Monitor. It is a program that listens for messages broadcasted by CPRS and forwards the clinician and patient identifiers to ATHENA-CDS Client when a clinician accesses the CPRS record of a patient.

ATHENA-CDS Client

A Java process that is responsible for managing: (1) the delivery of ATHENA-CDS advisories (sets of recommendations), including pop-ups, to clinicians, and (2) interactions between a clinician and the pop-ups (e.g., storing comments entered by clinicians and updating advisories upon request)

ATHENA-CDS Hypertension Guideline

A set of recommendations for treating patients with a diagnosis of primary hypertension in the primary care setting. It is based on the *The Sixth Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure*, VA Hypertension Guideline and newer evidence approved by the JNC.

ATHENA-CDS SYSTEM

Assessment and Treatment of Hypertension: Evidence-based Automation, Clinical Decision Support System. It is a clinical decision-support system that applies guidelines for hypertension management using Stanford Medical Informatics' EON architecture.

ATHENAEum

A set of relations, formalized as classes in Protégé, that allow mappings of terminologies used in Protégé to patient data

ATHENA-CDS Knowledge Base

Where the knowledge extracted from national guidelines for hypertension is formalized in a machine-interpretable format. It is stored in Protégé.

ATHENEON

The database, created from the ATHENA-CDS database, that has the schema used by the ChronusII temporal mediator

API

Application Programming Interface

CEGEE

Core EON guideline execution engine

Client

Any computer program that accesses services provided by server programs in the server-client architecture

CORBA

Common Object Request Broker Architecture

CPRS

Computerized Patient Record System

Domain

A particular field of knowledge, such as hypertension

DSS

Decision Support System

DTS

Data transformation service

EMR

Electronic medical record

EON

A technology, created by Stanford Medical Informatics, for developing decision-support systems for guideline-based care. The name *EON* is not an acronym.

EON Guideline Interpreter

See Guideline Interpreter.

EON Guideline Model

The set of classes and attributes, used by ATHENA-CDS, that describe concepts and relations with which the content of clinical guidelines are formalized for the purpose of providing automated decision support

HTN

Hypertension

ICD9 Code

The International Classification of Diseases, 9th Revision

IIS

Internet Information Service

ISO

International Standard Organization

Guideline Interpreter

Also EON Guideline Interpreter. It is the execution engine that combines patient data with the knowledge base that generates advisories.

JNC 6

The Sixth Report of Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure

KIDS

Kernel Installation and Distribution System

MUMPS

MUMPS, also known as M, is a procedural, interpreted general-purpose programming language oriented towards database applications. It is the programming language used to implement VA's VistA system. *MUMPS* is no longer considered to be an acronym.

n-ary

"Any number of," by analogy with unary, binary, etc.

Object Management Group

An open membership, not-for-profit consortium that produces and maintains computer industry specifications (e.g., CORBA and Unified Modeling Language) for interoperable enterprise applications

PAL

Protégé Axiom Language

PAL constraint

Constraints on relations in Protégé, written in the Protégé Axiom Language

PCAServer

The computer program that provides a collection of methods for client programs (such as ATHENA-CDS Client) to request and obtain guideline-based advisories computed by the Guideline Interpreter

Protégé

A tool for building knowledge bases and knowledge-based systems. It allows the modeling and encoding of guideline knowledge. Developed at Stanford, Protégé provides an interface through which clinician-managers can "easily" browse and update the guideline knowledge base.

RCT

Randomized controlled trial

Sig

The abbreviation of a Latin word meaning “let it be labeled.” It refers to the abbreviation of instructions for taking prescription medications, entered in the pharmacy database (e.g., “take 1 tab bid,” which is translated in the prescription label to “take 1 tablet twice daily”).

Slot

An attribute (property) of a class. For example, a class called Physician might have name, title, and phone number as slots.

SMI

Stanford Medical Informatics

System architecture

A software system’s constituent components and the relationships among them

ChronusII temporal mediator

A program, created at Stanford Medical Informatics, that extends the standard relational model and the SQL query language to support representation of time-oriented data and to make complex temporal queries

URL

Universal resource locator

VA ATHENA-CDS Team

The VA team that implemented ATHENA-CDS SYSTEM between 1999 and 2005: Mary Goldstein MD, Boc Coleman Pharm D, Parisa Gholami MPH, Susana Martins MD, Albert Chan MD

VistA

Veterans Health Information Systems and Technology Architecture, the health information system developed and used by the Veteran Health Administration

VI. Acknowledgements

The ATHENA-CDS Project has been funded in part by Department of Veterans Affairs (VA) Health Services Research and Development (HSR&D) grants CPG 97-006 “Guidelines for Drug Therapy of Hypertension: ‘Closing the Loop’” (1998 – 2000, Principal Investigators, Mary K. Goldstein, MD, and Brian B. Hoffman) and CPI 99-275 “Guidelines for Drug Therapy of Hypertension: Multi-Site Implementation” (2000 – 2005, Principal Investigators, Goldstein and Hoffman). Participating sites for the implementation and evaluation in the second grant phases were VA Palo Alto, VA Durham (site Principal Investigator, Euzene Oddone, MD), and VA San Francisco (site Principal Investigator: Michael Shlipak, MD). Project members have included Robert Coleman, MS Pharm; Susana Martins, MD; and additional participants from VA medical centers in Palo Alto, San Francisco, and Durham. Additional details about the overall projects are available in the Final Reports for grants CPG 97-006 and CPI 99-275, available on request from Dr. Goldstein.

The EON project used in the ATHENA-CDS SYSTEM was funded by NLM Grant LM05708. Dr. Mark Musen was the principal investigator. Project members included Aneel Advani, Amar Das, John Nguyen, Martin O’Connor, Yuval Shahar, Ravi Shankar, and Samson Tu.

Protégé was developed by Stanford Medical Informatics at the Stanford University School of Medicine with support from the Defense Advanced Research Projects Agency (DARPA), the National Cancer Institute (NCI), the National Institute of Standards and Technology (NIST), the National Library of Medicine (NLM), and the National Science Foundation (NSF). Protégé is a national biotechnology resource supported by grant P41 LM007885 from the National Library of Medicine.

The authors would like to thank Ms. Lesley Evensen for editorial assistance in the early stages of development of the manual, and Ms. Heidi Craig for editorial assistance in the later stages. Views expressed in this document are those of the authors and not necessarily those of the Department of Veterans Affairs or other funding agencies or affiliated organizations.

January, 2006.

VII. Contact Information

AUTHORS

Robert W. Coleman, MS Pharm – VAPAHCS, GRECC 182B, 3801 Miranda Ave, Palo Alto, CA 94304 ; bob.coleman@va.gov; phone 650 493 5000 x67136, fax [650-493-2505](#)

Mary K. Goldstein, MD – VAPAHCS, GRECC 182B, 3801 Miranda Ave, Palo Alto, CA 94304 ; mary.goldstein@va.gov ; phone work phone 650 493-5000 x62105, fax [650-493-2505](#)

Susana Martins, MD – VAPAHCS, GRECC 182B, 3801 Miranda Ave, Palo Alto, CA 94304; email : susana.martins@va.gov ; work phone 650 493-5000 x67167, fax [650-493-2505](#)

Samson W. Tu, MS – Stanford Medical Informatics, Department of Medicine, Stanford University, MOSB X259, 251 Campus Drive, Stanford, CA 94305-5479; swt@stanford.edu ; phone 650-725-3391, fax 650-725-7944

EDITOR

Heidi Craig – heidilcraig@gmail.com; phone 415 676 7740

TECHNICAL SUPPORT

Larry Lou – VA Palo Alto Health Care System, email address in VA national MS Exchange address book ; phone 650 493 5000 x65999. A VA Palo Alto HCS M (MUMPS) programmer, Mr. Lou will work with M programmer in IRMS to accomplish the installation of the KIDS build.

VIII. References

1. Goldstein, M.K., et al. *Implementing Clinical Practice Guidelines While Taking Account of Changing Evidence: ATHENA, an Easily Modifiable Decision-Support System for Management of Hypertension in Primary Care.* in *Proc AMIA Symp.* 2000. Los Angeles, USA.
2. NHLBI, *The Sixth Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure.* 1997, National Institute of Health.
3. VA/DoD Evidence-Based Clinical Practice Guideline Working Group, *Diagnosis and Management of Hypertension in the Primary Care Setting.* 1999, Office of Quality and Performance, Veterans Health Administration, Department of Veterans Affairs , and Health Affairs, Department of Defense: Washington, DC.
4. Shankar, R.D., et al. *Building an Explanation Function for a Hypertension Decision-Support System.* in *Proceedings of MedInfo2001.* 2001. London, UK.
5. Musen, M.A., et al., *EON: A Component-Based Approach to Automation of Protocol-Directed Therapy.* J Am Med Inform Assoc, 1996. **3:** p. 367-388.
6. Tu, S.W. and M.A. Musen. *A Flexible Approach to Guideline Modeling.* in *Proc AMIA Symp.* 1999. Washington DC: Hanley & Belfus, Inc.
7. Tu, S.W. and M.A. Musen. *Modeling Data and Knowledge in the EON Guideline Architecture.* in *MedInfo.* 2001. London, UK.
8. O'Connor, M.J., S.W. Tu, and M.A. Musen. *The Chronus II Temporal Database Mediator.* in *Proc AMIA Symp.* 2002.
9. Tu, S.W. and M.A. Musen. *From Guideline Modeling to Guideline Execution: Defining Guideline-Based Decision-Support Services.* in *Proc AMIA Symp.* 2000. Los Angeles, CA: Hanley & Belfus, Inc.
10. Gennari, J.H., et al., *The Evolution of Protégé: An Environment for Knowledge-Based Systems Development.* Int J Hum Comput Stud, 2003. **58**(1): p. 89-123.
11. Russell, S. and P. Norvig, *Artificial Intelligence: A Modern Approach.* 2nd edition ed. 2003, Upper Saddle River, New Jersey: Pearson Education, Inc.
12. Kolodner, R.M. and J.V. Douglas, *Computerizing Large Integrated Health Networks: the Va Success.* 1997: Springer-Verlag.
13. Andrus, R.W., *Integrating a Clinical System,* in *Computerizing Large Integrated Health Networks: The VA Success,* R.M. Kolodner, Editor. 1997, Springer-Verlag: New York.
14. VHA Office of Information. *VistA / CPRS Demo Site.* 2005 [cited 2005; Available from: <http://www1.va.gov/cprsdemo/>].
15. Hynes, D., et al., *Informatics resources to support health care quality improvement in the Veterans Health Administration.* JAMIA, 2004. **11**(5): p. 344-350.
16. Tu, S.W. and M.A. Musen. *The EON Model of Intervention Protocols and Guidelines.* in *Proc AMIA Symp.* 1996. Washington, DC: Hanley & Belfus.
17. Object Management Group. *Common Object Request Broker Architecture (CORBA/IIOP).* 2005 [cited 2005; Available from: http://www.omg.org/technology/documents/formal/corba_iiop.htm].

18. W3C. *Web Services Activity*. 2002 [cited 2003; Available from: <http://www.w3.org/2002/ws>.
19. Tu, S.W., et al., *A Methodology for Determining Patients' Eligibility for Clinical Trials*. Methods of Information in Medicine, 1993. **32**(4): p. 317-325.
20. Musen, M.A., *Scalable Software Architectures for Decision Support*. Methods of Information in Medicine, 1999. **38**: p. 229-238.
21. Noy, N.F., R.W. Fergerson, and M.A. Musen. *The Knowledge Model of Protege-2000: Combining Interoperability and Flexibility*. in *2th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000)*. 2000. Juan-les-Pins, France.
22. Genesereth, M.R. *Knowledge Interchange Format*. in *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*. 1991. San Francisco, CA, USA: Morgan Kaufmann.
23. NHLBI, *The Seventh Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure*. 2004, National Institute of Health.
24. Advani, A., et al. *Integrating a Modern Knowledge-Based System Architecture with a Legacy VA Database: The ATHENA and EON Projects at Stanford*. in *Proc AMIA Symp*. 1999. Washington, D.C.