Hypertension encoded in GLIF

Guideline 2 (Based on the hypertension guideline.
Simplified (not all contraindications, relative contra-indications, and relative indications
are specified).
Drug interactions simplified.
Contains 2 recommendations).

4) If the patient is not at the goal blood pressure, the response to the initial drug choice for
hypertension is inadequate after reaching the full dose, and the patient is tolerating the first
choice well, then add a second drug from another class. If a diuretic is not chosen as the first
drug, it is usually indicated as a second-step agent because its addition will enhance the effects
of other agents.

The possible drug classes are:

ACE I
beta blockers (distinguish between drugs that have intrinsic sympathomimetic activity (ISA) and
those that do not)
calcium channel blockers (distinguish between dihydropyridines and non-dihydropyridines,
and between long-acting (long duration) dihydropyridines and non-long-
acting dihydropyridines)
diuretics (distinguish between thiazide-, loop- and potassium-sparing diuretics)

The following compelling indications exist (unless contra-indicated):

<table>
<thead>
<tr>
<th>Indication</th>
<th>Drug Therapy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diabetes mellitus (type 1) with proteinuria</td>
<td>ACE-I</td>
</tr>
<tr>
<td>Heart failure</td>
<td></td>
</tr>
<tr>
<td>Isolated systolic hypertension in older patients</td>
<td>diuretics (preferred), dihydropyridines</td>
</tr>
<tr>
<td>Myocardial infarction</td>
<td>non-ISA beta-blockers</td>
</tr>
<tr>
<td>Myocardial infarction with systolic dysfunction</td>
<td>ACE-I</td>
</tr>
<tr>
<td>Patient does not have co-morbidities that are</td>
<td>diuretics, beta-blockers</td>
</tr>
<tr>
<td>compelling indications of other antihypertensives.</td>
<td></td>
</tr>
</tbody>
</table>

The following relative indications exist (unless contra-indicated):

<table>
<thead>
<tr>
<th>Indication</th>
<th>Drug Therapy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diabetes mellitus (type 1) with proteinuria</td>
<td>ACE-I (preferred), calcium channel blockers</td>
</tr>
</tbody>
</table>

The following relative contraindications exist:

<table>
<thead>
<tr>
<th>Indication</th>
<th>Drug Therapy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diabetes mellitus (type 1) with proteinuria</td>
<td>beta blockers</td>
</tr>
</tbody>
</table>

The following contraindications exist:

<table>
<thead>
<tr>
<th>Indication</th>
<th>Drug Therapy</th>
</tr>
</thead>
<tbody>
<tr>
<td>asthma or chronic airway disease</td>
<td>Beta-blockers</td>
</tr>
<tr>
<td>second- or third-degree heart block</td>
<td>Beta-blockers, non-dihydropyridine</td>
</tr>
</tbody>
</table>
The following drug interactions exist:

<table>
<thead>
<tr>
<th>Drug class</th>
<th>Evidence based combinations</th>
<th>Evidence based combinations to avoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACEI</td>
<td>potassium-sparing diuretics</td>
<td></td>
</tr>
<tr>
<td>Beta-blockers</td>
<td>thiazide diuretics</td>
<td>non-dihydropyridines</td>
</tr>
<tr>
<td>Dihydropyridines</td>
<td>beta-blockers</td>
<td>non-dihydropyridines</td>
</tr>
<tr>
<td>Non-dihydropyridines</td>
<td></td>
<td>beta-blockers, dihydropyridines</td>
</tr>
</tbody>
</table>

* This representation of drug interactions is based on interpretation that the VA physicians gave to the drug interactions that were specified in Table 11 of the original guideline. Evidence based combinations are drugs that, when taken together, have synergetic effects. Evidence based combinations to avoid are groups of drugs that should not be taken together.

We will make the following assumptions for this example:
1) The history of what other drugs were previously given to the patient is irrelevant.
2) The patient is on a single drug and another drug class will be added.
3) The patient is already taking the maximum dose of the drug.
4) The response to the first drug choice for hypertension is inadequate after reaching the full dose, and the patient is tolerating the first choice well.
5) There is no need to check if a specific patient is allergic to a specific drug.
6) It will be enough to suggest a drug class; it is not necessary to suggest a specific drug and dose.
7) Never give a contra-indicated drug to a patient.
8) Never choose as a second drug a drug that is in the "Evidence-based combinations to avoid" list.
9) compelling indications outweigh Evidence-based combinations and relative indications.

Please model the decision to add another antihypertensive drug in a modular way, which will take into account that there may be other indications, contraindications, and drug interactions that may later be added into the model. The model should be constructed in such a way that will enable adding these without changing the way the decision model is represented.

The guideline model should show not only one possible drug set that can be given, but show sets of drugs, based on contraindications, relative contraindications, Evidence-based combinations, compelling indications, and relative indications. For example:

One set can be the set of drugs that can be given (not contraindicated or a Evidence based combinations to avoid), have compelling indications and are Evidence-based combinations.

Another set can be the set of drugs that can be given (not contraindicated or a drug partner to avoid) and have compelling indications but are not Evidence-based combinations.

A third set can be the set of drugs that can be given (not contraindicated or a drug partner to avoid), do not have compelling indications and are not Evidence-based combinations.

There is no need to model the criterion the checks whether the BP goal is maintained, the patient is taking the full dose, the patient is taking only one anti-hypertensive drug, the response to the initial drug choice for hypertension is inadequate after reaching the full dose, and the patient is tolerating the first choice well.

(dimensions 1, 2, 3, 9, 10)

5) The goal of the (new) anti-hypertensive treatment discussed in recommendation (1) is to control blood pressure to < 140/90 mm Hg (systolic BP below 140 mm Hg and diastolic BP below 90 mm Hg). For patients with Diabetes Mellitus, the blood pressure should be controlled to below
Blood pressure should be controlled to 125/75 in patients with proteinuria in excess of 1 gram per 24 hours, and to 130/85 mm Hg in patients with proteinuria with at most 1 gram per 24 hours with whatever anti-hypertensive therapy is necessary.

Figure 1. Top-Level Hypertension Guideline. Data_Item are pointers to data items used by this guideline. Parameters passed are pointers to the parameters passed in/out of this guideline to its sub-guideline. Both comrbidities and medications are passed into this guideline from its sub-guideline.
Note that the goal of the add a second drug guideline is represented as a text string in the “Intention” slot of the guideline.

**Figure 2. Maintenance information of the Hypertension guideline**
Table 1: Hypertension Guideline Information

<table>
<thead>
<tr>
<th>Name</th>
<th>Hypertension guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Evidence</td>
</tr>
<tr>
<td>Items (1 values)</td>
<td>URL</td>
</tr>
<tr>
<td>Name</td>
<td>Hypertension Guideline</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://www.nhlbi.nih.gov/guidelines/hypertension/jnc6.pdf">Link to Hypertension Guideline</a></td>
</tr>
</tbody>
</table>

**Figure 3.** Supplemental material for the Hypertension guideline, referenced by Figure 1
See Figure 5

Patient state step

Get patient comorbidities and medication

Get patient-specific medical knowledge

compelling medications?

no

Add compelling medications

yes

See Figure 6

Action step

Calcium channel blocker

ACEI

diuretic

thiazide diuretic

loop diuretic

ACEI

potassium sparing diuretics

beta-blocker

non ISA beta-blocker

end add second drug

Second drug choice

Choice step
A Patient_State_Step is a guideline step (a node in the flowchart) that is used for two purposes. One purpose is to serve as a label that describes a patient state that is achieved by previous steps. This way, a guideline may be viewed as a state transition graph, where states are scenarios, or patient states, and transitions between these states are the networks of guideline steps (excluding patient state steps) that occur between two patient state steps. The other purpose of a patient state step is an entry point to the guideline (e.g., patient came back to the clinic at clinical state A). The two patient state steps seen in this algorithm serve as entry points. Figure 5 shows the details of one of these patient state steps.

Figure 5. A patient state step from Figure 4
Figure 6. The “Get patient comorbidities and medications” action step of Figure 4. This action step has a task of type Subguideline_Action. The Action_Detail slot shows points to a Guideline object, which is a subguideline designed to extract the patient’s comorbidities and medications.

We now go to see the details of this sub-guideline, shown in Figure 7.
Figure 7. The “Get patient comorbidities and medication” Subguideline. Medications and comorbidities are passed from this sub-guideline out to the top-level guideline.

Figure 8 shows the Algorithm of this sub-guideline.
Figure 8. The “Get patient comorbidities and medication” algorithm

Again, the patient state steps serve as entry points in to the guideline.
Figure 9. The “Get patient comorbidities” action step of the algorithm shown in Figure 8

This action step has many tasks, executed in the order in which they appear. They are all of two types `Medically_Oriented_Action_Specification`, and `Assignment_Action`. One example of the `Medically_Oriented_Action_Specifications` is shown in Figure 10. The `Assignment_Actions` will be discussed
The variable named “asthma” will hold the Query_Result of this Get_Data_Action. The source of the data will be the “Asthma” Data_Item, taken from the EMR. The definition of this data item is shown in Figure 11.

A data item points to a vocabulary concept id, which in this example is taken from UMLS. It also points to a data value, which is an Observation object, defined by GLIF3’s default Reference Information Model (RIM), which is HL-7’s RIM version 1.0. This way a data item is defined formally. The reference to a vocabulary code defines the data item’s “subject”. The reference to a RIM defines the structural attributes of the data item. In the HL-7 RIM, patient data items can be either procedures performed on a patient, observations about the patient, medications given to him/her, and other kinds of services.
The different sub-classes contain additional attributes that help characterize the Act. For example, Observation has a value attribute, whereas Medication has attributes that provide information about the dose and route by which the medication is given. Act objects have a mood that distinguishes the various ways in which they can be conceived: an event that occurred, a definition, intent, order, etc.

![hypertension_00644](instance of Observation)

**Figure 12.** The Asthma Observation, in event mood, signifying that we are concerned with observations about Asthma that already happened.

Data_Items should be used in decision criteria. The data items that we use in GLIF are complex data types. The expression language that we use in GLIF to specify decision criteria is based on Arden Syntax that has a very simple data model. Arden Syntax does not contain complex data types. It supports the following simple types: null, Boolean,
number, time (timestamp), duration, string. In addition, it supports a list of simple types and a query result, which is a list, whose values are associated with a primary timestamp. An element of a list or a query result can therefore have only one data value.

In order to overcome this difference between GLIF's and Arden's data models, we use the Get_Data_Action. Get_Data_Action retrieves patient data from the EMR and presents it in a form of a query result. It allows specifying a mapping from the default data model of GLIF3 to that of Arden Syntax. A guideline author can use Get_Data_Action to specify an attribute of a complex RIM class to be the source of data values for the query result, and values of another attribute to serve as the primary time in the query result. Thus, the query result is a list of value and primary time pairs similar to Arden's query result data type. However, the value attribute in GEL’s query result holds a simple or a complex GEL type. Get_Data_Action specifies the data item from the EMR that will serve as the source of data, and the attribute that will be selected from that data item. In this way, specific attributes of the data item can serve as the source of the data, and not necessarily the entire data item. For example, as shown in Figure 13, Get_Data_Action can retrieve all instances of Observation data items of the patient that refer to Asthma, and assign the value of their data value attribute, which is a RIM Observation object (see Figure 12), to the query result’s “value” attribute, and the end time of each Observation (critical_time.high) to the “primary time” attribute of the query result elements.

<table>
<thead>
<tr>
<th>Instance of Get_Data_Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_item: Asthma (see Figure 11)</td>
</tr>
<tr>
<td>attribute_to_be_assigned: data_value</td>
</tr>
<tr>
<td>variable_name: Asthma</td>
</tr>
<tr>
<td>primary_time: data_value.critical_time.high</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance of Query_Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>value: (Observation instance see Figure 12)</td>
</tr>
<tr>
<td>value: (Observation instance see Figure 12)</td>
</tr>
<tr>
<td>primary_time: 2002-01-08</td>
</tr>
<tr>
<td>primary_time: 1999-03-02</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Figure 13. The Get_Data_Action and its query result that holds Asthma Observation values and their primary times.

The “Get patient comorbidities” action_step queried each relevant comorbidity separately. So, we now have different variables that each holds a Query_Result that is specific for each comorbidity. We would like to merge these observations into a single list that will hold the Concepts (pointed to by the Observation objects) of all the comorbidities. First, we extract the concepts out of the Query_Results returned by Get_Data. This is done by the Assignment_Actions shown in Figure 9. One example of a concept extraction is shown in Figure 14.
Figure 14. The Assignment_Action that assigns an Asthma concept to the variable AsthmaIndication

The expression is written in GLIF’s Expression Language, a formal language called GEL (Guideline Expression Language), which is based on Arden syntax but supports complex types as data values and functions that can manipulate them. The GEL expression used to extract the Asthma Concept from the Asthma Query_Result, returned by The Get_Data_Action of Figure 13, and assign it to the AsthmaIndication variable, is:

“selectAttribute("service_cd", latest asthma where time of it >= now)”

selectAttribute is a function that serves like the “dot” notation. Let’s look at this expression more closely, starting from the inner part.

“latest asthma where time of it >= now”

This part looks at the asthma query result and returns the latest element (by primary_time) whose time is current (meaning that the observation value persists into the future, and is therefore current).

From this latest Observation we select the attribute “service_cd”, which holds the asthma concept. Note that if the inner expression returns null, then the entire expression returns null.

Now we can unite the different variables that hold concepts of comorbidities, or null values (if the patient does not have an indication) into one list. This is done by the "Unite comorbidities into one list" action step (see Figure 8), which contains one Assignment_Action task that merges the single comorbidity variables together. The expression used by the Assignment_Action is:

{AsthmaIndication, DMIndication, heartFailureIndication, systolicHypertensionIndication, MIIndication, MIWithSystolicDysfunctionIndication, chronicAirwayDiseaseIndication, secondDegreeHeartBlockIndication, thirdDegreeHeartBlockIndication, pregnancyIndication}
It forms a list of concepts from the single concepts. This expression is written in GEL.

In a similar way to getting the current comorbidities and uniting them into one list of concepts, the algorithm shown in Figure 8 gets the current active medications and unites them into one list. The list contains the single anti-hypertensive medication that the patient is currently taking.

We now go back to the top-level algorithm of Figure 4, which adds a second antihypertensive drug. The next action step is: “Get patient-specific medical knowledge”. It is done in a similar way of obtaining comorbidities of the patient from the EMR and uniting them into a list of concepts that represent current comorbidities of the patient. “Get patient-specific medical knowledge” has several tasks that compute patient-specific medical knowledge. They are shown in Figure 15.

![Figure 15. The ”Get patient-specific medical knowledge” action step](image-url)
Again, we first obtain some input and then perform some extraction from it, using GEL expressions. The input, in this case, is not patient data from the EMR, but general medical knowledge that represents contraindications, compelling indications, relative indications and contraindications, and good and bad drug partners. They are all represented in the form of concept relationships. An example of a contraindication concept relationship is shown in Figure 16.

![Figure 16. A contraindication concept relationship. ACEI and Pregnancy are concepts, defined by UMLS codes](image)

The “Get patient-specific medical knowledge” action step of Figure 15 needs to first obtain the medical knowledge, expressed as concepts relationships (Note: medical knowledge is expressed as either Concepts, or Concept_Relationships). The “Get ContraindicationList” task, expanded in Figure 17, is a Get_Knowledge action specification that is used to obtain medical knowledge.
The "Get ContraindicationList" action specification stores concept relationships that represent contraindications in a variable called "contraindicationList".

The next step is to extract from these contraindication relationships the relationships that involve indications that the patient has (are in the comorbidities list of concepts). This is done by the Assignment_Actions of the action step "Get patient-specific medical knowledge", shown in Figure 15. These Assignment_Actions use GEL expressions to assign the patient-specific knowledge to variables. For example, "Assign contraindicated drugs for the patient" assigns to a variable called "contraindicatedDrugs" a GEL expression that computes the list of drugs that are contraindicated by the patient’s comorbidities. The GEL expression used is:

\[
\text{selectAttributeFromList("concept_from", contraindicationList where (containsValues(comorbidites, selectAttributeFromList("concept_to", contraindicationList)))})
\]

containsValues accepts two list arguments, \(l_1\) and \(l_2\). The function returns a list of Booleans of length equal to the length of \(l_2\). The value of the Boolean in position \(i\) of the returned result is TRUE if in position \(i\) of \(l_2\) there exists a value that is contained in \(l_1\).

By now we have all the patient-specific information that is needed to make the decision on a second drug choice: contraindications, compelling indications, relative indications and contraindications, and good and bad drug partners that all apply to the patient’s current comorbidities and current anti-hypertensive medications.

Before we make the choice we check if the patient has no comorbidities, and if he doesn’t assign two compelling medication classes: Diuretics, and Beta-Blockers, as shown in
Figure 8. The decision of whether the patient has no comorbidities can be done automatically, hence it is represented by a Case_Step. The case step provides a means to represent conditional selection of one and only one path from among several alternatives. Figure 18 shows the details of the “compelling medication?” case step.

![Image of the Case_Step for the compelling medication? case step](hypertension_00452_instance_of_Case_Step.png)

**Figure 18.** The “compelling medication?” case step.

The “yes” and “no” options contain expressions of “True” and “False”, respectively, and direct flow of control to the next guideline steps. The details of the “yes” option is shown in Figure 19.
The “Add compelling medications” action step of Figure 8, assigns (via an Assignment Action Specification task) a list of two concepts, Diuretics, and Beta-blockers, into the compellingMedication variable. The expression used to create the list is: “{Diuretics, BetaBlockers}”, where Diuretics and BetaBlockers are concepts that were extracted from using Get_Knowledge_Action, in a similar way to that shown in Figure 17.

Now we can finally make the decision on a second drug. This decision is represented by the Choice Step of Figure 8.

Choice steps represent a decision between guideline steps for which the guideline does not provide deterministic selection criteria. An external agent, such as a human or another program, must make the decision in choice steps, and select one of the decision options.

There are 3 subclasses of the Choice class: RuleIn Choice, Weighted Choice, and Utility Choice. Here, we use a RuleInChoice.

RuleInChoices specify rule-in, rule-out, strict-rule-in and strict-rule-out criteria for each decision option. These criteria help the user choose one of the decision options.

The strict-rule-in criteria rank a choice as the best among several options. For example, when there are competing diagnoses for a disease, a pathognomonic condition would be a strict-rule-in for the disease.

A strict rule out is analogous to an absolute contraindication. For example, “allergy to penicillin” is a strict rule out for giving penicillin.

A strict-rule-out takes precedence over strict-rule-in when ranking options. If an option contains both a strict-rule-in criterion and a strict-rule-out criterion, and both evaluate to true, then that option should be the last choice.

Strict-rule-ins take precedence over rule-ins and rule-outs. The ranking of rule-ins and rule-outs is left to the user who may use his or her clinical judgment or may develop their own ranking schemes.

All the strict-rule-outs of the same choice are related to each other using the OR relationship (i.e., if there are 2 rule-ins, A and B, then they are equivalent to a single rule-in stating A OR B). Similarly, all the strict-rule-ins of the same choice are related to each other using the OR relationship.
Figure 20. Second drug choice step

The choice step lists the different options of medications. For each medication option, the same rule-in, rule-out, strict_rule_in and strict_rule_out are used! The 4 rules that apply to one medication option all use the same proposedDrug value. Different proposedDrug values are used by different medication options.

Note that there is a didactic element associated with the choice step of Figure 20. It is shown in Figure 21.
Figure 21. A didactics for choosing appropriate second drug

Figure 22 shows the 4 rules for selecting ACEI.

Figure 22. the rule in choice for selecting ACEI

The details of the strict rule-in are shown in Figure 23
Figure 23. The details of the strict rule-in

The proposedDrug variable is assigned with an ACEI concept using Get_Knowledge, as shown in Figure 24.
Finally, after choosing a drug, an appropriate Action is carried out to prescribe the appropriate medication, as shown in Figure 8. Each of these actions has a medically oriented action specification task.

Figure 25 shows the details of ordering an ACEI.
Figure 25. A medically_Oriented action specification used to order ACEI

**File Format**

The guideline is saved in RDF format. Following is part of the specification of the hypertension guideline. Since the GLIF representation is very complex, it is very hard to follow the guideline encoding without a viewing/authoring tool like Protégé.

```xml
<!DOCTYPE rdf:RDF [
  <!ENTITY a 'http://www.rdfschema.org/mynamespace.rdf#'>
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
]> 
<rdf:RDF xmlns:a="&a;" xmlns:rdf="&rdf;">
  <a:Guideline rdf:about="&a;hypertension_00001">
    <a:name>Add a second drug</a:name>
    <a:intention xml:space='preserve'>Sustain goal BP of < 140/90 mm Hg.
    In patients with Diabetes Mellitus, the BP should be controlled to < 130/85 mm Hg.
    In patients with proteinuria in excess of 1 gram per 24 hours the BP should be < 125/75 mm Hg.
    In patients with proteinuria with at most 1 gram per 24 hours the BP should be < 130/85]]>
  </a:intention>
  <a:eligibility_criteria rdf:resource="&a;hypertension_00002"/>
</a:Guideline>
</rdf:RDF>
```
The patient is not at the goal blood pressure, the response to the initial drug choice for hypertension is inadequate after reaching the full dose, and the patient is tolerating the first choice well.