MODELING RELIEF
Walton Sumner, II, MD (1), Jin Zhong Xu, Ph.D. (2),
Guy Roussel, MD (2), Michael D. Hagen, MD (2)
1. Washington University in St. Louis, Missouri
2. American Board of Family Medicine, Lexington, Kentucky

ABSTRACT
The American Board of Family Medicine deployed virtual patient simulations in 2004 to evaluate Diplomates’ diagnostic and management skills. A previously reported dynamic process generates general symptom histories from time series data representing baseline values and reactions to medications. The simulator also must answer queries about details such as palliation and provocation. These responses often describe some recurring pattern, such as, “this medicine relieves my symptoms in a few minutes.” The simulator can provide a detail stored as text, or it can evaluate a reference to a second query object. The second query object can generate details using a single Bayesian network to evaluate the effect of each drug in a virtual patient’s medication list. A new medication option may not require redesign of the second query object if its implementation is consistent with related drugs. We expect this mechanism to maintain realistic responses to detail questions in complex simulations.

INTRODUCTION
The American Board of Family Medicine began using a patient simulation program as part of its Maintenance of Certification process in 2004. The program stochastically simulates patients from a knowledge base in an effort to meet security and reusability goals. Bayesian networks define health states, control queries, and attempt to maintain consistency during stochastic operations. Many small networks are manually constructed to maintain a degree of consistency and clarity appropriate for testing purposes.

To test diagnostic and management skills, the simulator must report temporally and physiologically reasonable symptom histories. For instance, a simulated patient should report on demand the presence of pain, as well as palliative, provocative, quality, radiation, severity, and temporal details (sometimes called “PQRST” symptom details). The simulator design intended to support temporal reasoning by plotting the presence and absence of diseases, findings, and interventions over time. Attaching these data to a patient would allow the simulator to directly inspect durations of events, and then to produce temporally reasonable reports. For instance, the simulator could check the duration of hypothyroidism or depression, and cause the patient to report feeling fatigued for that amount of time.

General trends
A previously described process describes general temporal trends in symptoms, accounting for any number of distinct underlying problems and treatments. As shown in figure 1, this process identified points in time to sample, and at those time points executes a semi-physiologic model represented as a Bayesian network. A Bayesian network is an acyclic graph in which nodes represent general concepts and states of nodes represent specific facts. For instance, the node Sex should have states “Male” and “Female” while a disease stage node might have five states, from “Absent” to “Severe.” Arcs represent probabilistic inferences about child nodes based on the state of parent nodes. Bayes’ theorem allows the estimation of probabilities of states of unresolved nodes given any combination of input data. Node states may be associated with numeric values or ranges. Nodes may acquire numeric values directly from the simulator, from calculations based on other values, or from values assigned to inferred node states. For instance, two nodes might acquire the current baseline blood pressure or the current antihypertensive medication effect as inputs from the simulator. Another node in the Bayesian network might sum these values, and then classify the result as normal or abnormal. The Bayesian network generates a number indicating the severity of a symptom at that moment in time, taking into account relevant recent treatments and underlying illness.

Many queries accept as an input a series of time-value pairs as generated in the previous step. Nodes in a
Bayesian network may collect information about the series of time-value pairs or other patient data. Given these inputs, the network can be solved stochastically. To maintain consistency between solutions, any random numbers needed to obtain a stochastic solution can be saved for re-use. The states of nodes in the solved network typically are associated with text phrases. These phrases are concatenated to generate a report.

**Extending the temporal history model**

This process produces an accurate description of current symptoms, a good general summary of symptom trends, and can describe the net effect of multiple interacting events. However, the mechanism cannot offer any insights about PQRST details.

Temporal effects of provocative and palliative events represent a complex modeling challenge. For instance, how can the simulator answer the question, “During an angina attack, what relieves your pain?” As with the general trend problem, this may require an analysis of a series of time-value pairs describing, for instance, the efficacy of sublingual nitroglycerin as a vasodilator. However, the question has several important differences. First, this question has two contextual components. The interviewer has restricted the anatomic location to the chest (or wherever the patient feels angina pain) and the time to ‘during an angina attack’. The detail question lacks any specific time reference, and is nonsensical unless the patient has angina. Second, the interviewer may want the patient to summarize observations made during several angina episodes. The most complete response might not describe a single logically possible event. For instance, this response probably summarizes observations made over several angina episodes: “Rest completely relieves my pain in about 20 minutes. Nitroglycerin completely relieves my pain in a few minutes, but not if I keep exercising.” Third, the response may have multiple parts, each describing how the symptom responds to a different palliative action (rest, sublingual nitroglycerin). Fourth, the time-value pairs being inspected might already exist in the knowledge base, as a description of the dilating effects of nitroglycerin. The Bayesian network may need to adjust the description of the series to account for other patient characteristics, but the series only needs to be located.

**METHODS**

Figure 2 illustrates a symptom detail algorithm with five major steps. The first step identifies the detail of interest. The second step identifies interventions of potential interest. The third step identifies finding responses of interest, called features. The fourth step identifies interventions that trigger interesting features, providing time-value series for the next step. The fifth step uses a Bayesian network to inspect the series, finally merging with the previously developed algorithm for general symptom trends.

Figure 2

1. **Identifying details of interest**

Users interact with the patient simulator through a web-based presentation system. An internally developed natural language matching program maps users’ free text symptom questions to query objects. When a user types a free text question, such as “Do you have chest pain?” or simply “angina,” the matching systems finds an identification number (id) for the best matched query object. The presentation system sends the id to the simulator. The query object in the simulator includes the following attributes:

- **Query object**
  - Sampling List [Time point]
  - Value Calculation [Numeric calculation]
  - Shape Source [Finding/Feature/Site]
  - Logic [Bayesian Network]
  - Root Node [Bayesian Node]
  - State List [Node state]
  - Report template [Text Formula]
  - Detail List
    - Detail Type {PQRST list}
    - Root Node State [Node state]
    - Report template [Text Formula]
    - Query [Query]

The simulator would typically calculate a series of time-value pairs, and apply the query’s logic to the series, following the previously described algorithm. In the process of solving the Bayesian network, the state of that network’s root node is determined. Although
most Bayesian networks applications do not define a root node concept, our networks always have a node that broadly summarizes the content the simulator requires. In the example of chest pain, the root node might have separate states indicating presence or absence of exertional chest pain.

The report template is a string with embedded references to nodes named in the query’s logic. The simulator replaces these references with text associated with nodes’ states, producing a report to return to the presentation system. The simulator assembles an XML structure with this string tagged as the main result.

The detail list may contain any number of items. The simulator scans this list for details where the root node state matches the solution of the root node in the query’s logic. If the detail does not identify a root node state, it matches all root node states. Any non-empty subset of items matched to the root node state will include items with one or more distinct types. The subset could have multiple items with the same type. For each unique type in the subset, the simulator adds a <Detail> tag to the XML document, with the type identified as an attribute of the tag, e.g.: 

<Detail type = “Palliative”>…</Detail>

If the detail report template is not empty, then the simulator uses the parent query’s logic to supply phrases to replace any node references in the template. The simulator places the template solution between the <Detail> tags, and marks it as text. This mechanism will often suffice for details regarding quality, radiation, and severity of a symptom.

If the detail report template is empty, then the detail must identify a query object. In this case, the query object id is embedded in the detail tag.

The final XML result looks like this:

```
<QueryReport>
  <QueryName>Chest pain</QueryName>
  <QueryKey>1234</QueryKey>
  <HText>I have chest pain with exertion.</HText>
  …
  <Details>
    <Detail type = “Palliative”>
      <DText/>
      <QueryID>9999</QueryID>
    </Detail>
    <Detail type = “Quality”>
      <DText>It is a tight feeling.</DText>
      <QueryID/>
    </Detail>
  </Details>
</QueryReport>
```

The presentation system receives this XML result and displays the content marked by the <HText> tag as the virtual patient’s response to the original query. The presentation system may draw attention to the available details with hyperlinks or any suitable control. The user reviews the result of the query, and selects any details to pursue. When the user selects a detail with pre-specified text, the presentation system immediately returns that text. When the user selects a detail that references a query id, the presentation system sends that key back to the simulator.

In addition, the presentation system could allow the user to search for details using free text. For instance, the user might ask, “How does nitroglycerin affect your pain?” The natural language matching program would identify the query as a palliation detail involving a specific intervention, sublingual nitroglycerin. The presentation system would return the ids of the detail query and the intervention.

2. Identifying interventions of potential interest
The second step identifies interventions of potential interest when the virtual patient responds to the detail query. If the user requests information about specific interventions, then this step finds the intersection of the requested set of interventions and the set of interventions that the virtual patient has actually used. Otherwise, all of the interventions that the patient uses now, or has used in the past, are of potential interest.

This step prepares the virtual patient to either answer the question about specific interventions, or survey all of the medications it has taken in search of any that are relevant to the detail. If the patient does not use an intervention identified with a query, then a default negative answer is returned.

3. Identifying findings and features of interest
The simulator uses the detail query id to locate a query object. This object may have the same structure as the general query, or it may use the shape source attribute to identify a finding, a finding feature, and site. The simulator uses these shape source data to locate a finding attached to the patient.

Findings describe relatively low-level physiologic data, such as diastolic or systolic blood pressure, glucose level, or myocardial oxygen supply. Findings are divided into Specific Findings, which are typically related to diagnoses. For instance, different myocardial oxygen supply parameters might correspond to different simulated stages of coronary artery disease.
Specific Findings comprise any number of Specific Features, such as baseline value, circadian rhythm, and response to medication. Specific Features have at least one Pattern (triggering event and a list of time-value pairs). Within a Finding, every Specific Finding’s list of Specific Features should describe the same list of Features, but the Specific Feature data differ between Specific Findings. The Finding structure is:

Finding  
Specific Finding List  
Specific Feature List  
 Feature Name  
Pattern List  
Trigger  
Intervention  
Dose  
Time-Value Series

4. Identifying intervention-triggered effects
By comparing an intervention selected in step 2 with the list of trigger interventions located in step 3, the simulator may locate one or more related time-value series. If none are located, the intervention from step 2 has no effect. If one or more time-value series are located, then the simulator can compare the dose associated with each to the dose actually taken by the patient, interpolating as necessary to predict the effect of the patient’s dose. For instance, a Finding structure can describe in one Pattern the vasodilating effect of 0.3 mg of nitroglycerin and, in a second Pattern, the effect of 0.6 mg of nitroglycerin. If the patient uses some intermediate dose, such as 0.45 mg, the simulator can predict the vasodilating effect as intermediate in duration and efficacy between the effects of 0.3 and 0.6 mg doses.

If the virtual patient does not currently use an intervention, the simulator determines whether the patient used the intervention during the time interval described in the original general question. If the intervention has any overlap in time with the symptom, the patient is allowed to describe the intervention’s effect in detail.

At the conclusion of this step, the simulator has a list of interventions relevant to the patient and the Finding and Feature specified by the user’s query, and a time-value series associated with each intervention.

5. Merging with the general trend algorithm
The process that generates a general symptom trend uses a Bayesian network (the logic attribute of a Query object) to analyze a time-value series generated by repeatedly sampling a virtual patient’s symptom status. This Bayesian network can use a variety of predefined analysis nodes to describe different qualities of the time-value series, such as its duration, the time of maximum value, and the maximum value. The Bayesian network can also access any other descriptive information about the patient, such as the presence or absence of specific diseases or interventions.

After setting node states that directly reflect input data, the simulator obtains a stochastic solution to the Bayesian network. For each node having an indeterminate state, a random number between 0 and 1 is recovered (if the node has been resolved before and consistency is desired) or generated. The number is used to assign one of the possible states. The node states in the solution provide text fragments to insert in the report.

Although the times in details’ time-value series may be much shorter than those in the general temporal response data, most predefined analytic nodes require few if any changes are required to manage details. At this step, for each detail query object, for each intervention, the logic Bayesian network evaluates the intervention’s time-value series. The report generated by this evaluation is appended to a report listing the effects of each intervention. The complete report is a list of interventions and effects on the details of interest.

RESULTS
A coronary artery disease model demonstrated that this process permits the simulator to produce a report about palliation of angina symptoms. A Bayesian network surmises general angina symptoms on demand, based on the stage of disease modeled and presence or absence of atypical symptoms. A presentation system displays a symptom description and lists available details provided by the XML message. Request for a list of palliative interventions yields a phrase built from the following template, where [Intervention] gets the generic name of an intervention, and [Shape__Report] gets a string describing a time-value series representing the vasodilating effects of an intervention on a ten point scale:

[Intervention] gives me a [ShapeAbsoluteMaximum__Report] amount of relief in [ShapeAgeAtAbsoluteMaximum__Report].

In a patient with a sublingual nitroglycerin prescription, this template resolves to:

“Sublingual nitroglycerin gives me a very high amount of relief in a few minutes.”

This process has generated symptom details in online virtual patient simulations of several domains for two
years. When additional details are available, the current interface provides a single hypertext link labeled “more”. Users routinely click this link to obtain details, and seldom identify the link or the extra step required as user interface problems.

DISCUSSION

One imperative for the ABFM simulation system is accurate time management of simulated patients. Existing patient data structures support queries and sub-queries of any patient on any finding at any time. This extension to support detailed sub-queries required only minor modifications to the query class data structure and presentation system, and modest modifications to simulator algorithms.

Generating symptom details requires completion of previously enumerated finding descriptions rather than new or substantially more complex query structures and algorithms. The Specific Feature class previously did not require detailed distinctions between or descriptions of the kinds of Features being modeled. Now much more descriptive labeling of Feature types and more extensive modeling of Specific Features is required to model a broad range of palliative and provocative intervention effects. Equally important, knowledge acquisition efforts require a deliberate approach to managing modeled intervention effects and side effects.

Diplomates of the ABFM have been able to access this information during online simulations. Diplomates share many observations on the Maintenance of Certification process, including the virtual patient interface, but have generally accepted detail content and the step required to disclose it.

In our current testing domains, we have primarily used the relatively simple mechanism of supplying details as text linked to reports. While domains are restricted to chronic diseases with few flares or treatment failures, this is likely to meet requirements for disclosing details. As simulations grow in complexity, users may have opportunities to respond to unsuccessful treatments, side effects, partial adherence to regimens, and other realistic challenges. These will predictably overwhelm a design requiring text linked to reports, but the detail sub-query mechanism developed and tested with angina should gracefully accommodate these situations.

Limitations

In spite of the progress obtained thus far, a number of important limitations in our temporal modeling remain. First, our experience with detailed sub-queries is limited to a few domains involving relatively chronic diseases. More acute scenarios may require additional changes.

Second, we have not provided for combining multiple time-value series or setting exact intervention times in the current algorithm. Many drug interactions probably can be modeled just by knowing that both drugs are in use at the same time, using Bayesian nodes to query for the presence of a drug that enhances or hinders the effect of a second drug. However, accurate descriptions of some drug interactions might require a more precise processing of two time series of data. We do not believe that our virtual patient simulations will require this level of detail in the immediate future.

CONCLUSIONS

Using a physiologic model of instantaneous symptom status to sample the patient’s history allowed arbitrarily detailed review of general symptom histories. The structures and algorithms extend gracefully to support detailed sub-queries including palliative and provocative events. Challenges now shift to facilitating knowledge acquisition tasks and coordinated management of objects.

References