Combining Tabular, Rule-Based, and Procedural Knowledge in Computer-Based Guidelines for Childhood Immunization

Perry L. Miller,* Sandra J. Frawley,*† Frederick G. Sayward,† William A. Yasnoff,‡ Lorraine Duncan,‡ and David W. Fleming‡

*Center for Medical Informatics, Yale University School of Medicine, New Haven, Connecticut 06520-8009; †Medical Decision Associates, New Haven, Connecticut; and ‡Oregon Health Division, Portland, Oregon

Received December 30, 1996

IMM/Serve is a computer program which implements the clinical guidelines for childhood immunization. IMM/Serve accepts as input a child's immunization history. It then indicates which vaccinations are due and which vaccinations should be scheduled next. The clinical guidelines for immunization are quite complex and are modified quite frequently. As a result, it is important that IMM/Serve's knowledge be represented in a format that facilitates the maintenance of that knowledge as the field evolves over time. To achieve this goal, IMM/Serve uses four representations for different parts of its knowledge base: (1) Immunization forecasting parameters that specify the minimum ages and wait-intervals for each dose are stored in tabular form. (2) The clinical logic that determines which set of forecasting parameters applies for a particular patient in each vaccine series is represented using if–then rules. (3) The temporal logic that combines dates, ages, and intervals to calculate recommended dates, is expressed procedurally. (4) The screening logic that checks each previous dose for validity is performed using a decision table that combines minimum ages and wait intervals with a small amount of clinical logic. A knowledge maintenance tool, IMM/Def, has been developed to help maintain the rule-based logic. The paper describes the design of IMM/Serve and the rationale and role of the different forms of knowledge used. © 1997 Academic Press

1. INTRODUCTION

IMM/Serve is a computer program developed to help manage childhood immunization for initial use by the Oregon Health Division. IMM/Serve handles the six childhood immunizations which are routinely administered: Haemophilus influenzae type b (Hib), hepatitis B (HepB), varicella (Var), measles mumps rubella (MMR), diphtheria tetanus pertussis (DTP), and polio (OPV/IPV). IMM/Serve is being developed for use in three modes: a forecasting mode, which is implemented, and reminder and assessment modes, which are under development. In its forecasting mode, IMM/Serve indicates which immunizations are due on a specific date and which should be scheduled next. In its reminder mode,
IMM/Serve will indicate when a patient should be contacted (e.g., by postcard or phone) because immunizations are due. In its assessment mode, IMM/Serve will help analyze how well a population of patients complies with the immunization guidelines. IMM/Serve can be accessed for demonstration purposes on the World-Wide Web (http://ycmi.med.yale.edu/immserve/).

Consensus guidelines for childhood immunization are currently developed by two national organizations: the American Academy of Pediatrics (AAP) and the CDC’s Advisory Committee on Immunization Practices (ACIP) (1–3). The guidelines attempt to specify in detail how the immunizations should be given. The logic is tailored to a variety of conditions, including contraindications, delayed doses, and other clinical factors. As described later in the paper, these guidelines are quite complex. In addition, the guidelines are frequently modified. Any computer program that embodies the guidelines must not only be able to handle the complexity, it must also facilitate the maintenance of the logic as the clinical field itself evolves over time.

There is a growing need for computer-based immunization guidelines. In a major national initiative, many states, school systems, and health organizations have built (or are currently building) immunization registries (4) either as databases focused specifically on patients’ immunization histories or as part of a broader computer-based patient record. All of these immunization registries could potentially benefit from the incorporation of a computer program for immunization forecasting, reminding, and assessment (IFRA) which embodies the national immunization guidelines, but which allows customization to reflect certain aspects of local practice. Particularly in view of the complexity of the logic and the need to update and maintain that logic over time, it would be very inefficient for each immunization registry to build and maintain its own IFRA program.

One possible solution to this problem would be for a computer program to serve as an intermediary between the national immunizations panels that formulate the guidelines and the many local immunization registries that use them. In the process of building such a program, one could help the panels assure that the guidelines were comprehensive and unambiguous and at the same time provide a mechanism for disseminating the logic in a computer-based form to the immunization registries. Such a computer program might even evolve over time to become an explicit definition of the standard of care. The current IMM/Serve program represents a step toward this goal.

In building IMM/Serve, we have worked to create a design that facilitates the process of modifying and maintaining the logic. In so doing, we have attempted to define the most appropriate representation for the various components of the immunization knowledge. As a result, IMM/Serve combines four different types of knowledge. (1) Tables are used to represent the various forecasting parameters associated with each vaccine dose (such as minimum age and minimum wait intervals from previous doses). (2) If–then rules represent the clinical knowledge that determines which set of forecasting parameters should be used in any given set of clinical circumstances. (3) Procedural logic is used to combine the various
temporal parameters (dates of previous doses, minimum ages, and temporal intervals) to produce dates when future doses should be scheduled. (4) A decision table stores the screening parameters for each dose together with a small amount of clinical logic, which determine whether previous doses are valid.

The paper first describes the operation of IMM/Serve. It then describes IMM/Serve's design, focusing on the use of the four different types of knowledge and the rationale for each. Finally the paper discusses certain design issues that arose during the implementation of IMM/Serve.

2. BACKGROUND

There are a number of projects currently underway to place clinical guidelines into computer-based form so that the knowledge can be presented in a fashion that is structured around a patient's clinical presentation. These projects range from (1) the development of alerts and reminders which function as modest-size logical modules, each complete in and of itself, to (2) the construction of comprehensive consultation systems which attempt to respond to all combinations of conditions relevant to a clinical problem. The IMM/Serve project falls into the latter category. Its goal is to produce appropriate recommendations for every patient described and to cover all possible presentations.

A variety of different approaches have been used to represent computer-based guideline knowledge.

1. The Arden Syntax (5–8) has emerged as a standard for representing clinical alerts and reminders, with the goal of facilitating the interinstitutional sharing of the logic. The heart of Arden Syntax is procedural code. Procedural representation of clinical logic does not facilitate the development of software tools to help verify that logic. Such tools, however, are not as necessary in the case of alerts and reminders where each module is typically separate and independent and where there is not an attempt to respond to every combination of a set of clinical conditions.

2. If–then rules have been extensively used as a representation for clinical logic, for both alerts and reminders (9–11), for more comprehensive computer-based guidelines (12), and for clinical expert consultation systems (13).

3. Decision tables have also been used as a representation for clinical guidelines (14, 15). A decision table can be thought of as a mechanism for organizing a set of if–then rules in a way that helps assure that all combinations of input conditions are dealt with and that allows for verification of knowledge consistency.

4. Recent research in the design of expert systems has developed a variety of approaches to representing domain knowledge in ways that avoid some of the problems inherent, for example, in systems built with large numbers of if–then rules. These newer systems, which have been called second generation expert systems (16), explore ways of capturing various features of a domain, including its higher level structure, thereby facilitating the acquisition, maintenance, and manipulation of the knowledge. Clinical examples include PROSTEGÉ-II and its predecessors (17). IMM/Serve differs from such projects in that its goal is not
to explore new approaches to knowledge representation. Its goal is to use methods at hand to produce an efficient, operational solution to an important clinical problem.

As described later in this paper, IMM/Serve combines tabular, rule-based, and procedural representation to encode its domain knowledge. Its design illustrates how different components of a knowledge base can be represented using different paradigms to enhance better the maintenance of the logic over time.

3. OVERVIEW OF IMM/SERVE IN OPERATION

This section introduces IMM/Serve by showing the system in operation in its forecasting mode. This mode is designed for use, for example, when a child arrives at a clinic. IMM/Serve takes as its input clinical data about the child, including the immunization history, as shown in Fig. 1. These data include:

1. The child's date of birth.
2. The date for which the forecast is to be performed. In clinical practice, this will be the date when the patient arrives at the clinic. For testing purposes, for reminding, and for compliance assessment, this might be some other data.
3. The dates of all previous vaccinations, including optionally the vaccine brand.
4. A list of any vaccines which are contraindicated for the child and which therefore should not be given.
5. Other relevant clinical information about the patient, such as the mother's hepatitis B surface antigen (HBsAg) status.

IMM/Serve processes these input data and produces the output seen in Fig. 2. This output first indicates whether any of the doses were given too soon and therefore cannot be counted as part of the list of valid vaccinations. (No such doses are identified in Fig. 2.) The forecast then indicates any vaccinations which are due (as of the forecast date), any vaccinations which should be scheduled next (indicating the appropriate date or range of dates), any vaccination series which are complete or no longer relevant, and other notes and comments.
The following immunization(s) are due on 11/1/1996:
- DTaP 4
- Hib 3
- OPV 3
- VAR 1

The following immunization(s) will be due:
- D/T Series dose 5, on or after 8/1/1999 but before 8/1/2001
  (if DTaP 4 is given on 11/1/1996)
- OPV 4, on or after 8/1/1999 but before 8/1/2001
  (if OPV 3 is given on 11/1/1996)
- MMR 2 or Me 2, on or after 8/1/2000

The following vaccine series are either complete or no longer relevant for this case:
- HepB

Advisory Notices:
- In Oregon, the D/T Series dose 5 and the OPV/IPV Series dose 4 are due prior to kindergarten entry.

Fig. 2. Sample output produced by IMM/Serve, as described in the text, for the case described in Fig. 1.

In its internal design, the core IMM/Serve program is designed to function as a set of callable routines. It takes its input and produces its output in coded form using standardized C structures. We call these structures (1) a universal immunization input structure (UIIS) and (2) a universal immunization output structure (UIOS). The use of the UIIS allows IMM/Serve to be activated in a variety of different ways. Input data can be accepted directly from an immunization database, interactively from a clinician using a desktop or laptop machine, via the World-Wide Web, or from a file of case data. No matter how the data are derived, they are converted to the UIIS format for input to IMM/Serve.

The use of the UIOS similarly allows IMM/Serve’s output to be used for various purposes. The coded information can be placed directly into a database for further analysis, such as for assessment of patient compliance or for generation of reminders. Alternatively, the UIOS can be processed by a report generator to produce an English prose forecast for a clinician to read. Figure 2 shows the output produced by IMM/Serve’s current forecasting report generator.

4. SOURCES OF COMPLEXITY IN THE IMMUNIZATION GUIDELINES

For a child who has no vaccine contraindications and who comes on time for all vaccinations, the vaccination schedules are fairly straightforward. When one considers all the different combinations of circumstances which might alter the standard schedule (as discussed below), however, the logic becomes quite complex. One format for representing this complexity can be seen in the material produced by the CDC in the Morbidity and Mortality Weekly Report (2). In that version of the guidelines, a grmphical chart indicates the standard immunization
schedules. Below the chart, however, there is a quite extensive list of detailed footnotes which expand on the recommendations for special situations. In addition, considerably more detailed recommendations are available for individual vaccine series dealing with issues such as late starts and delayed schedules (18).

The complexity of the immunization guidelines stems in part from the need to balance a number of factors. First, there is the scientific and clinical knowledge of the immunization process for each vaccine, indicating how the vaccine is best given. This knowledge must be balanced against the need to produce a set of vaccination schedules that can be implemented in the real world, schedules that are not too complex to understand and which do not require an excessive number of visits to the clinic. In addition, one does not want to subject the child to an excessive number of needle sticks, both over time and especially at a single visit.

The remainder of this section discusses specific examples of complexity in the current immunization guidelines, divided into issues of clinical and temporal complexity.

4.1. Clinical Complexity

A variety of clinical issues contribute to the complexity of the immunization logic. For example:

1. Hib immunization is designed to protect the child during the first 5 years of life. As a result, if the child has missed early Hib vaccinations, then the recommended number and schedule of remaining vaccinations changes based on the length of the delay. In the extreme case, if the child reaches 5 years with no Hib vaccinations, none is required.

2. A further complexity is that the recommended number and schedule of Hib doses vary depending on the brand used. The guidelines must also allow for the possibility that the brand used in previous vaccinations is not known.

3. In the polio vaccine series, the guidelines must anticipate how to handle a mixed history of vaccinations with OPV (an oral live vaccine) and IPV (an inactivated vaccine).

4. In several vaccine series, the guidelines must allow the possibility that not all antigens have been given at a past vaccination. In the MMR series, for example, a dose may have involved only measles vaccine.

5. Certain live vaccines (OPV, MMR, Var) must be given either at the same time or at least 30 days apart. The exact set of vaccines to which this rule applies may vary depending on local practice.

6. Other clinical factors also influence the guidelines. For example, the HepB recommendations vary depending on whether the mother is HBsAg positive and must allow for the possibility that the mother’s HBsAg status is not known.

7. An additional source of complexity stems from the need to screen each dose in a child’s immunization history using minimum ages and wait intervals. If a dose is invalid, it is generally not counted as past of the history, but in certain circumstances (e.g., with live vaccines) may still affect the analysis.
4.2. Temporal Complexity

Temporal issues affect the immunization guidelines in several ways. One source of complexity is that time is expressed in several different units. In different parts of the guidelines, the patient’s age may be expressed in weeks, months, or years. Similarly, time intervals may be expressed in days, weeks, months, or years. As a result, IMM/Serve must be prepared to combine a specific date from the input data (such as the current date of the most recent vaccination in a series) together with several parameters from the guidelines involving ages (expressed in several units) and intervals (expressed in several units) to calculate a specific date or date range. In performing these calculations, IMM/Serve must also handle the peculiarities of our calendar, including leap years and the fact that months have different lengths.

A further temporal complexity involves the question of how to define a month. For scheduling future immunizations, it is clinically natural to use calendar months. When analyzing a set of past vaccinations (e.g., when screening the immunization history for “valid” vaccinations, or when performing compliance assessment), however, it makes sense to use a fixed length month, for example 28 days. (Otherwise, if a vaccine series required a 1-month interval between vaccinations, a 28-day interval would be valid in February, but would be invalid in any other month.) IMM/Serve is currently designed to use (1) calendar months when dealing with all ages expressed in months and with present and future intervals expressed in months and (2) fixed length months (with 28 days as a default) when analyzing past intervals expressed in months.

5. OVERVIEW OF IMM/SERVE’S DESIGN

Figure 3 provides a schematic overview of IMM/Serve’s internal processing. The system has three main components: the preprocessor, the rule interpreter, and the postprocessor. The operation of these components is described later in more detail. This section provides an overview of the system as a whole.

1. The IMM/Serve Preprocessor: The preprocessor takes as its input a case description (as shown in Fig. 1) along with various parameters such as the value to use as the fixed length of a month. It processes this input, transforming it into a set of facts that can be tested by IMM/Serve’s rules, as shown below. Example facts “asserted” by the preprocessor could include “age_in_months” (the child’s current age in months), “Var.prior” (the number of valid prior varicella doses which the child has received), “HepB.contraindicated” (hepatitis B vaccine is contraindicated), and “Hib1.brand” (the brand of Hib vaccine used for dose 1). At the same time, the preprocessor also screens the input data to identify any past vaccinations that were given too early based on age and interval criteria. Any such vaccinations must in general be ignored in formulating the immunization recommendations (although they are still considered in certain circumstances). If an invalid dose is found, an error message is produced. The set of facts produced for input to the rule interpreter reflects only those vaccinations that are valid.

2. The Rule Interpreter: The rule interpreter contains the clinical forecasting
logic which determines which set of age and interval parameters apply for each vaccine series, as described later in detail.

3. The Postprocessor: The postprocessor converts the recommendations produced by the rule interpreter (which involve ages and time intervals involving days, weeks, months, and/or years) into specific dates.

5.1. Implementation Details

The preprocessor and postprocessor are both written using the C programming language. The rule interpreter is implemented using a rule-based shell program (KnowledgeCraft) which converts the rules into a structure which is interpreted by a C “driver” program to execute the rules. As a result, the whole of IMM/Serve is executable as a C program and is therefore easily and quite efficiently run on a wide range of different machines. The program currently executes roughly 29 cases per second running on a laptop PC with a 100 MHz 486 processor, 127 cases per second on a PC with a 200 MHz Pentium Pro processor, and 23 cases per second running on a SparcStation 20 with a 50 MHz RISC processor.

As mentioned previously IMM/Serve is designed to operate as an independent module, accepting its input and producing its output as coded information contained in C structures. This design allows the system to be used in a variety of
ways. It can be integrated directly into an immunization registry system, which is how it is being used in Oregon. It can also be accessed through the Web and implemented on a stand-alone machine.

6. IMM/SERVE’S FOUR FORMATS FOR KNOWLEDGE REPRESENTATION

This section describes the four different types of knowledge represented in IMM/Serve. It also discusses how and why each type is used to provide different functionalities and how the different forms of knowledge are integrated.

6.1. Tabular Knowledge for Immunization Forecasting

Each vaccine series involves a sequence of doses. When a typical child is on schedule, there are a set of standard parameters for each dose which indicate whether the child is ready for that dose. These parameters may include (1) a minimum acceptable age (at which the dose may be given if the child happens to be at the clinic), (2) a recommended age (at which the dose should be scheduled), (3) a past-due age, (4) a minimum wait-interval for the previous dose, (5) a minimum wait interval from earlier doses of the vaccine, and (6) in the case of live vaccines, a minimum wait-interval from previous doses of other live vaccines.

In the presence of certain clinical factors (such as the presence of any vaccine contraindications, or the fact that the child is late for one or more doses in the series), these standard dose parameters may not apply. Instead, a different set of parameters applies.

Figure 4 shows how IMM/Serve stores, in tabular form, the various sets of parameters that apply for the four doses of Hib vaccine. Notice that each set of parameters has a name (Hib1, Hib2, Hib2-final, etc.). Hib1, Hib2, Hib3, and Hib4 are the four standard sets of parameters that are used when a child is on schedule. Hib2-final and Hib3-final are alternative sets of parameters which are used for the second and third doses of Hib, respectively, when a child was late for earlier doses. The Hib3-final parameters apply in several circumstances including when the first Hib dose occurred after 7 months of age but before 12 months of age. The Hib2-final parameters apply when the first Hib dose occurred after 12 months and before 15 months of age.

The contents of each parameter set are dictated by the immunization guidelines. The ages and intervals may be expressed in any of four units: days, weeks, months, and years. In all vaccine series, there is a minimum wait interval between successive doses of that vaccine. In certain vaccine series, other types of intervals apply as well. For example, in the HepB series, the third HepB dose must be at least 4 months later than the first HepB dose, in addition to being at least 2 months later than the second HepB dose. Also, for the live vaccine series (OPV, MMR, and Var), doses must be at least 30 days later than certain previous live vaccine doses (although multiple live vaccines may be given on the same day).

Expressing these parameters in tabular form has a number of major advantages. It is easy to inspect the knowledge. It is also easy to update the parameters to
reflect changes in the guideline, or to customize the parameters to accommodate local practice preferences. In a previous version of IMM/Serve, these parameters were included in the rules (which are described below). This required that the rules be recoded each time these parameters were changed. The current tabular format significantly enhances the ability to maintain this knowledge as it evolves over time.

This tabular forecasting knowledge is used in two phases of IMM/Serve’s operation. (1) The parameter sets are tested by the rule-based logic, as described below. (2) The parameter sets are used in the postprocessing logic to determine actual dates.

### 6.2. Rule-Based Knowledge for Immunization Forecasting

In addition to specifying the forecasting parameters associated with each dose of each vaccine series as described above, the immunization guidelines also specify the clinical conditions that determine when each set of parameters applies. In IMM/Serve, this clinical logic is encoded using if–then rules, which operate in a forward-chaining fashion. For each vaccine series, there are three main groups of rules:

**Hib1**
- acceptable
- recommended
- past due
- wait
- 6 weeks
- 2 months
- 3 months
- -

**Hib2**
- acceptable
- recommended
- past due
- wait
- 10 weeks
- 4 months
- 5 months
- Hib1 1 month

**Hib2_final**
- acceptable
- recommended
- past due
- wait
- 14 months
- 15 months
- 16 months
- Hib1 2 months

**Hib3**
- acceptable
- recommended
- past due
- wait
- 18 weeks
- 6 months
- 7 months
- Hib2 1 month

**Hib3_final**
- acceptable
- recommended
- past due
- wait
- 12 months
- 15 months
- 16 months
- Hib2 2 months

**Hib4**
- acceptable
- recommended
- past due
- wait
- 12 months
- 15 months
- 16 months
- Hib3 2 months

---

Fig. 4. The tabular parameters used by IMM/Serve for the Hib vaccine series.
1. **Initial rules that set flags to guide later processing:** Figure 5 shows three if–then rules that test input facts and assert inferred facts to guide the later rule-based logic. “Hib.prior” (the number of prior doses of Hib, which may be zero), “Hib1_age_in_months” (the child’s age when Hib1 was given, which may be undefined if no Hib doses have been given), and “Hib12_elapsed_months” (the elapsed time between Hib1 and Hib2) are input facts asserted by the preprocessor based on the patient description. These initial rules test these facts and assert additional inferred facts, such as “late_start_12_months” (the child has had a late start in the Hib series, starting at or over 12 months of age). These initial rules set the stage for the heart of the rule-based logic which follows. In the current IMM/Serve, there are 29 of rules of this sort.

2. **The rule “kernel”:** Figure 6 shows six of IMM/Serve’s kernel rules which form the central core of the rule-based logic. These rules may test both input facts and inferred facts and define the clinical logic which determines which set of tabular parameters applies. They also determine which of three “contexts” apply: namely (a) whether the dose is due now, (b) whether the dose is not yet due, and (c) whether a next dose should be scheduled to follow a dose which is due now. The rules shown in Fig. 6 perform this logic for dose Hib2.

   a. The first two rules determine whether Hib2 is due now, and if so, which parameter set applies. To allow this determination, the preprocessor has processed the input data using the parameter table, and has asserted the facts “Hib2_final_parameters_met” and “Hib2_parameters_met” if the age and interval parameters associated with Hib2.final and Hib2 (see Fig. 4) are met.

   b. The last two rules determine whether Hib2 is not yet due and, if so, what parameter set applies.
The middle two rules determine whether a third Hib dose should be scheduled to follow a Hib2 dose which is due now and, if so, which set of parameters apply. In the current IMM/Serve, there are 114 of these kernel rules.

3. Later rules that may specify the vaccine brand or preparation to use: The final set of rules (see Fig. 7) test the facts asserted by the kernel rules and, if appropriate, indicate a specific vaccine brand or preparation to use. For example, the first rule tests whether the specific vaccine brand PRPOMP should be recom-
Abort if: Hib_contraindicated or Hib_complete

Hib_{prior} = 0: Hib1

Hib_{prior} = 1: if Hib1_{age in months} >= 15 then done
                   if Hib1_{age in months} >= 12 and Hib1_{age in months} < 15 then Hib2_final
                   else Hib2

Hib_{prior} = 2: if Hib2_{age in months} >= 15 then done
                   if prpomp_series or Hib2_{age in months} >= 12 or Hib1_{age in months} >= 7 then Hib3_final
                   else Hib3

Hib_{prior} = 3: if Hib3_{age in months} >= 15 then done
                   else Hib4

Fig. 8. The “definition logic” used by IMM/Def to generate automatically the kernel rules for Hib. IMM/Def is currently used to help double-check the rule-based logic.

mended. The output of these rules is in the form of coded textual “comments” which serve as input to the postprocessor. In the current IMM/Serve, there are 120 of these rules that may specify the brand or preparation to be administered. These rules are generally quite simple.

**IMM/Def—A Software Tool to Help Maintain the Kernel Rules:** In our experience building and refining IMM/Serve through several major revisions of its knowledge base, it has become clear that the biggest problem is maintaining the kernel rules. These rules need to be able to react to all possible combinations of input conditions and to determine which parameter sets apply in each of the three contexts described above. Keeping track of all the different combinations and being sure that every one is handled correctly is a very difficult task for the knowledge engineer. Fortunately, it became clear that we could build a software tool (which we call IMM/Def) to assist in this process.

It turns out that the rule kernel can be perceived (1) as being based on a more fundamental set of logic (which we call “definition logic”) that indicates which parameter sets to use in different circumstances and (2) as applying that logic in the three contexts described above (a dose is due now, a dose is not yet due, and a next dose should be scheduled to follow a dose which is due now). The rule kernel implements the same basic definition logic for each of these three contexts. IMM/Def takes as input the definition logic for each vaccine series and generates the rule kernel automatically. Figure 8 shows the definition logic for Hib, which is clearly much more concise than the corresponding kernel rules. IMM/Def takes this definition logic and processes it to generate 17 kernel rules for Hib, including the six rules shown in Fig. 6. In the process of generating these rules, IMM/Def must modify the definition logic in various ways so that it will operate correctly in each of the three different contexts in which it applies.

IMM/Def can help make sure that the rule kernel deals correctly with all the
combinations of conditions, in all three contexts, and thereby greatly simplifies the verification of the kernel rules. So far we have used IMM/Def as a tool to double-check the rule kernel. By going back and forth between the two forms of the same logic, we are able to test and refine both representations. In the future, we anticipate that IMM/Def may be able to generate a rule kernel which can be directly incorporated into IMM/Serve, thereby eliminating the need for hand coding the kernel rules at all. We anticipate that IMM/Def will significantly facilitate the process of refining and maintaining IMM/Serve’s rule-based logic as it evolves over time. As described later, IMM/Def is one of a set of planned knowledge maintenance tools which will help test and maintain IMM/Serve’s logic.

6.3. Procedural Knowledge to Handle Temporal Complexities

The rule-based logic does not attempt to determine specific dates on which future vaccinations should be scheduled. Instead, the knowledge to perform this calculation is expressed in procedural code which takes the parameters specifying minimum ages, wait-intervals, etc., together with the dates of previous vaccinations and combines this information to derive future vaccination schedules. Although the logic to do this may seem to be quite straightforward conceptually, a host of special cases arise from (1) the need to integrate ages and intervals expressed in multiple units (days, weeks, months, and years) together with (2) the need to deal with factors like months of different lengths, leap years, and issues involving the year 2000. As a result, this logic is tricky to write, and is much easier to write procedurally rather than, for example, using if–then rules. Also, as long as our society maintains its current calendar, this logic will not need to be modified. An additional advantage of this design is that there is a clean separation between the rule-based clinical logic and this temporal logic. The rules can be written, updated, and maintained without worrying about the temporal complexities at all.

6.4. A Decision Table for Screening Immunization History

The logic used to screen doses to assure that they are valid is shown in Fig. 9. Here a decision table format is used, which allows a small amount of clinical logic to be included in the table. The screening logic is essentially a very reduced subset of the forecasting logic. The logic is sufficiently different, however, that we have chosen to represent it separately. In addition, whereas the clinical forecasting logic is represented separately from the table of dose parameters (using if–then rules), the clinical screening logic is included directly in the table itself. The rationale for this design is discussed more fully later.

When screening a child’s immunization history, IMM/Serve looks at each dose sequentially (from the first dose to the last), consulting the table to see if the dose violates age or interval criteria. To do so, IMM/Serve examines each row of the table starting at the top until it finds a row where the conditions in the left half of the row are met. The right half of that row then indicates the age and interval criteria to use.
Fig. 9. This decision table contains the logic for screening the immunization history for all six vaccine series, as described in the text. Each row specifies (1) a dose being screened and any clinical conditions to be met (to the left of the vertical bar) and (2) the minimum age and wait interval criteria including those associated with live vaccine (LV) series (to the right of the vertical bar).

For example, in screening a child’s first hepatitis B vaccination (HepB1), the first row of the table indicates that there is no minimum age and no wait interval. Similarly, the second row indicates that for HepB2 to be valid, the child’s age must be at least 1 month and at least 1 month must have passed since HepB1. Although the HepB screening contains no clinical logic, other series include modest amounts of such logic.
1. All of the D/T series vaccinations (DT, DTP, DTaP, and Td) are straightforward except one. If the Td was used as the third D/T series shot (Td3) and if the patient is 7 years old or more, then there is no minimum age and 6 months must have passed since the previous D/T series dose. In all other situations involving the third D/T series shot (DT3, DTP3, or DTaP3 at any age, or Td3 at less than 7 years old), a minimum age of 18 weeks and a 1 month wait-interval apply.

2. In certain circumstances, the screening parameters for Hib depend on the brand of the dose being screened and on the brand of previous doses.

3. In the polio series, the screening parameters for OPV3 and for IPV2 depend on whether the previous vaccination in the series was OPV or IPV.

7. CURRENT STATUS AND FUTURE DIRECTIONS

In its forecasting mode, IMM/Serve has been extensively tested in several ways.

1. During its development and refinement, IMM/Serve was tested using several hundred hand-crafted test cases and several hundred actual patient records from the Oregon Health Division’s (OHD) immunization registry.

2. We recently developed a pilot version of an automated immunization test case generator (IMM/Test) which systematically creates test cases for all combinations of clinical conditions affecting the kernel rules. IMM/Serve runs correctly on all of these cases. The automated generation of test cases by IMM/Test, which is currently being extended and refined, and the analysis of those cases by IMM/Serve may be viewed via the World-Wide Web (http://ycmi.med.yale.edu/immtest/).

3. IMM/Serve is now linked to the OHD registry where it is being further tested on a pilot basis, in preparation for full operational use. IMM/Serve’s reminder mode and assessment mode are still undergoing development and refinement.

An important related goal is to develop a suite of immunization knowledge maintenance (KM) tools that will help in the process of understanding, updating, and maintaining IMM/Serve’s knowledge base. IMM/Def is one example of such a tool. We have also developed a graphical format for presenting immunization logic that has greatly facilitated understanding and discussing that logic (19). Other KM tools which are currently being tested and refined include programs: (1) to help verify the completeness of the rule-based kernel and (2) to generate test cases for IMM/Serve automatically. These tools operate on the tabular and rule-based knowledge and are designed to help in the modification, verification, and testing of successive versions of that knowledge.

A further goal is to work with other immunization registries to explore the issues involved in disseminating the system more broadly. A long-term ambition is to use IMM/Serve as a model to illustrate how a computer-based version of a clinical guideline might serve as the repository of the standard of care, maintained in association with clinical panels and disseminated for use by many different organizations that need to have the guidelines in computer-based form.
8. DISCUSSION

This section discusses four issues that arose in the course of building IMM/Serve.

8.1. The Rationale for Using the Four Types of Knowledge

A central design principle in building IMM/Serve was to facilitate the modification and maintenance of the knowledge. In our experience, this is by far the most difficult task in building a complex computer-based clinical guideline of this sort. This capability will be important both to accommodate changes in the recommendations as the clinical field evolves and to allow the system to be customized, hopefully in relatively modest ways, to reflect the local practice of specific states and health organizations. The four types of knowledge chosen for different components of IMM/Serve reflect this desire to facilitate modification and maintenance of the knowledge.

Tabular Forecasting Knowledge. Expressing the forecasting parameters associated with each vaccine dose in tabular form clearly facilitates maintenance. As long as the underlying clinical logic which dictates which set of parameters applies is unchanged, the specific minimum ages, wait times, etc., can be changed simply by changing the table.

Rule-Based Forecasting Knowledge. The clinical logic that determines which set of forecasting parameters applies cannot be expressed in a simple tabular form, but can be expressed readily using if–then rules. Were that logic to be expressed in procedural code, it would in all likelihood not be as cleanly separated from the rest of the program. In addition, once the clinical knowledge is expressed in if–then rules, the knowledge can be manipulated and tested by software tools developed to help assure its completeness and consistency. The IMM/Def program, which currently helps double-check the logic of the rule kernel, is an example of such a tool. As described previously, we are developing several tools to operate on the rule-based knowledge.

Decision Table Logic for Screening. The logic used for screening previous doses for validity involves a subset of the forecasting parameters, together with a small amount of clinical logic. All this information fits readily into a decision table, which allows easy modification. The rationale for using a different representation of clinical logic for screening vs forecasting is discussed below.

Procedural Knowledge. The various temporal complexities described previously are much more easily handled using procedural logic than, for example, using if–then rules. Also, this logic should not need to be modified as the clinical field changes. As a result, an additional advantage of the current design is that there is a clean separation between the clinical logic and the temporal complexities. The clinical logic can be written and maintained without worrying about temporal issues at all. The rules produce recommendations indicating sets of age and interval parameters that apply, without specifying about how those values actually get translated into specific dates.
8.2. How Best to Combine Clinical Logic with Age and Interval Parameters

As described above, IMM/Serve takes two different approaches to integrating clinical decision logic with minimum age and wait interval parameters. For forecasting, if-then rules are used to express the clinical logic. For screening, the clinical logic is included in a decision table. This section discusses two factors that influence the most appropriate use of these alternative approaches.

1. **Complexity of the Clinical Logic:** A major factor in determining how to represent the clinical logic is the complexity of that logic. The modest amount of clinical logic required for screening fits readily into a decision table. If a decision table was used for forecasting, however, the number of clinical parameters to be tested would be much greater, and the number of meaningful combinations of those parameters would number in the thousands. As a result, if-then rules provide a more organized and more understandable representation for the forecasting logic than would a decision table format.

2. **How the Logic is Used:** Another factor involves how the logic is used. For screening, the logic needs to be consulted multiple times (once for each previous dose) when processing a case. A particular dose cannot be screened until each previous dose is screened to determine whether it is valid or not. It would not be very efficient to call IMM/Serve’s rule interpreter so many times for independent analyses. Once the screening of previous doses is completed, however, the entire set of facts describing the case can then be passed to the rule interpreter for forecasting. As a result, the rule interpreter needs to be invoked only once per case, to perform the forecasting analysis.

8.3. Does the Use of Multiple Knowledge Representations Increase or Decrease the Complexity of Immunization Knowledge Engineering?

A related question concerns the relative merits using a single knowledge representation vs combining several different representations for expressing immunization knowledge. Might the task of the knowledge engineer be simplified by having a single representation? We believe that this is not the case.

1. The immunization knowledge that fits naturally into tabular form is much more easily read, understood, and modified when expressed in that form.

2. For immunization logic that does not fit naturally into tabular form but which needs to be modified frequently, however, rules are preferable to purely procedural representation. Rules allow the clinical guideline logic to be captured in a way that separates it cleanly from the other procedural parts of the IMM/Serve program as a whole and which allows the development of knowledge maintenance tools to help verify and test the logic as it evolves over time. The rule-based knowledge is, however, much more time-consuming to debug and maintain than the tables.

3. For immunization knowledge which does not need to be modified, procedural representation may be most efficient.

In a sense, our goal has been to focus on the immunization logic that requires modification over time, and to express as much as possible in tabular form. In
this way, we reduce the amount of knowledge expressed in rules to a relative minimum. We believe that this combined approach simplifies the knowledge engineering task for the group developing the logic. In addition, we believe that users (e.g., immunization registry staff) will find it quite straightforward to modify the tabular parameters. It would require a very sophisticated user, however, to modify the rule-based logic and thoroughly verify those modifications. For the immediate future, we anticipate that changes to the rules will only be made by IMM/Serve’s developers.

8.4. Defining the Scope of IMM/Serve’s Functionality

We anticipate that a major use for IMM/Serve will be as a component of an immunization registry. Such a registry will typically have a variety of different computer-based capabilities. The question arises as to what capabilities should be embedded in IMM/Serve itself, and what capabilities more logically should reside in the other modules of an immunization registry system. We have taken the approach that the IMM/Serve’s function should primarily be to embody the recommendations of the national immunization panels (customized as needed to reflect local practice).

There are a number of implications of this philosophy. When forecasting dates on which future immunizations might be scheduled, IMM/Serve produces the exact date or range of dates recommended by the guideline. A particular clinic, however, might wish the computer to indicate the first weekday or the first available clinic date. Alternatively, a clinic might want the computer to recognize when several vaccinations can be given together, even though some might need to be delayed slightly past their due dates, and indicate the date when all could be given at once. Another desired capability might be to recognize when Hib and DTP could be given together as a combined vaccine and to recommend that vaccination. IMM/Serve is not designed to provide these functionalities. If a clinic wants these capabilities, then either (1) a human will have to read IMM/Serve’s output and make the necessary adjustments or (2) additional computer programming will be required to take IMM/Serve’s coded output and transform it as desired.

IMM/Serve is designed to serve a well-defined need, to implement the logic of the guidelines so that they can be incorporated into an immunization registry as a callable routine. The registry therefore need not worry about coding or maintaining that logic. The registry is then free to adapt IMM/Serve’s output in any way that facilitates its clinical operations.

9. SUMMARY

Childhood immunization provides an exciting domain to which to showcase the capabilities of medical informatics techniques. The domain applies to many patients and is particularly important in the current environment of managed care with its heavy emphasis on preventive medicine and quality of care. There is a real opportunity for a computer program to serve as an intermediary between national panels developing immunization guidelines and a host of health organi-
zations which need an operational computer-based version of those guidelines. This process will be greatly enhanced if the overall system design is structured to facilitate knowledge maintenance and if a suite of knowledge maintenance tools are developed to assist in the process. The current IMM/Serve program is one step toward this goal.

ACKNOWLEDGMENT

This research was supported in part by NIH Grant R44 GM48289.

REFERENCES
