COMPUTERS AND BIOMEDICAL RESEARCH **31,** 172–189 (1998)
ARTICLE NO. CO981472

Tools for Immunization Guideline Knowledge Maintenance

I. Automated Generation of the Logic "Kernel" for Immunization Forecasting

Perry L. Miller

Center for Medical Informatics, Yale University School of Medicine, New Haven, Connecticut

Received July 31, 1997

IMM/Def is a prototype computer program designed to facilitate the building and maintenance of a rule-based program which performs childhood immunization forecasting. An immunization forecasting program takes as input a child's immunization history and produces recommendations as to which vaccinations are due and which should be scheduled next. A significant amount of the knowledge required for immunization forecasting can be expressed in tabular form, including the parameters that indicate the minimum age when each dose may be given and the minimum intervals between doses. The choice of which of these sets of parameters apply to a particular case depends upon additional clinical logic. To perform forecasting, this logic must be applied in three temporal contexts: (1) a dose is due now, (2) a dose is not yet due, and (3) a dose must be scheduled to follow a dose which is due now. Building and maintaining this logic by hand is a formidable challenge. IMM/Def demonstrates how this task can be simplified by first defining immunization "definition logic" which can be automatically translated into if-then rules for each of the three contexts. The approach has been applied successfully to the six childhood vaccination series which are routinely administered. A key advantage is that IMM/Def allows one to have two specifications of the logic that can be examined independently and that can be cross-checked to help assure completeness, consistency, and accuracy of the logic. The paper describes how IMM/Def performs its translation and discusses several design issues and lessons learned. © 1998 Academic Press

1. INTRODUCTION

There is currently a major national initiative to develop clinical practice guidelines and to develop mechanisms to link those guidelines to the electronic medical record to provide patient-specific recommendations for patient care. The process of taking a complex guideline and converting it into computer-based form is a challenging task, particularly if the resulting program must respond appropriately to all combinations of a range of clinical conditions. An extensive process of testing and validation must be carried out to assure that the program responds appropriately throughout its intended domain.

0010-4809/98 $25.00

In addition, once such a computer-based guideline is built, the task of refinement and verification is far from over. The guideline must be maintained as the clinical field itself evolves. The maintenance of such a program over time is at least as challenging as its initial creation (*1*). Once the complex task of building and testing a computer-based guideline is completed, it may prove difficult to generate the same level of enthusiasm, attention, and resources to support the ongoing task of maintaining the program's knowledge for as long as the program is used. In addition, the staff who initially developed the program may move on to other jobs or other problems, compounding the problem of maintaining the knowledge.

One approach which may help both in the initial development of computer-based guidelines and in the continuing maintenance of the knowledge is to develop software tools that assist in the process. This paper discusses an example of such a tool, IMM/Def, developed to assist in building, refining, and maintaining a computer-based guideline for childhood immunization.

Childhood immunization is a very topical domain in which to explore these issues. A large number of childhood immunization registries have been built, or are being built, to serve the needs of states and various health organizations (*2*). Once information about a child's immunization history is online, a natural next step is to link that data to an immunization ''forecasting'' program that produces patient-specific recommendations as to (1) what vaccinations are due and (2) when future vaccinations should be scheduled.

The American Academy of Pediatrics (AAP) and the CDC's Advisory Committee on Immunization Practices (ACIP) both produce consensus recommendations for childhood immunization (*3–5*). These recommendations are natural candidates to incorporate into a computer-based immunization forecasting program (although the recommendations may need to be customized to reflect the local practice of specific states and health organizations).

There are several overall approaches to building an immunization forecasting program. A significant amount of information can be stored in tabular form. This information includes the minimum acceptable age for each dose in an immunization series, the recommended age, the age at which the dose is overdue, and the minimum interval after a dose before the next dose can be given. In addition to these data, there is a variety of clinical logic which indicates when each of these sets of parameters should be used. One approach, which we have implemented in a program called IMM/Serve (*6*), is to code this additional clinical logic using if-then rules.

However it is implemented, an immunization forecasting program must be able to apply the underlying logic in the following three temporal contexts: (1) a dose is due now, (2) a dose is not yet due, and (3) a future dose should be scheduled to follow the current dose which is due now. In each of these three contexts, the same underlying logic must be expressed differently. Expressing the immunization logic to handle these three contexts correctly is difficult. The updating, maintaining, and revalidating of this logic as the recommendations themselves evolve over time represent a significant additional effort.

IMM/Def is a prototype computer program built to explore how a central core portion, or "kernel," of the logic required to perform immunization forecasting can be expressed in a quite simple format (which we call "definition logic") which can then be automatically translated into rule-based "implementation logic" to handle the three temporal contexts in which the logic must be applied. IMM/Def therefore provides a valuable tool to double-check the rule kernel. By going back and forth between these two forms of the same logic, one is able to test and refine both representations. This in turn helps assure completeness, consistency, and accuracy of the knowledge. We anticipate that IMM/Def will significantly facilitate the process of refining and maintaining immunization logic as it evolves over time.

2. BACKGROUND

A number of different approaches are being taken to encode clinical guidelines in computer-based form to provide patient-specific advice.

1. The Arden Syntax is a procedural representation being used to encode clinical alerts and reminders (*7–9*). Since the Arden Syntax is procedural, it does not lend itself as naturally to automated validation as do other more manipulable representations. This potential limitation is not a major issue, however, since the alerts and reminders are usually constrained, independent "chunks" of clinical logic. There is typically no need for extensive cross-checking of different Arden modules against one another for logical consistency.

2. If-then rules have been used extensively to solve many problems in medicine and in areas beyond medicine (*10–12*). A well-known advantage of rules is that in addition to being executed in an operational program, they can also be processed for other purposes (*13*), including (a) the explanation of the program's internal logic to the user and (b) the validation of the program for completeness and consistency.

3. Decision tables have also been used to encode clinical guidelines and to assist in verification (*14*). Decision tables may become unwieldy, however, if the number of conditions that need to be considered becomes relatively large.

4. More recent expert system research has focused on some of the limitations of rules. By creating a program containing if-then rules, one reduces the solution of a problem to a set of independent, atomistic problem-solving elements. In the process, a great deal of the underlying structure of the domain may be lost. This limitation may significantly hamper the building, testing, and maintaining of a complex system. As a result, researchers have explored ways to include a representation of the underlying domain structure, so that this structure can help guide the processes of knowledge acquisition, validation, and maintenance. Such systems have been termed "second generation" expert systems (*15*). A well-known example in the clinical arena is PROTÉGÉ-II, which was originally designed for use with cancer chemotherapy protocols and which has since been applied to other clinical domains (*16*). Its goal is to provide a domain-independent approach that allows the underlying

structure of a clinical domain to be captured in a way that facilitates acquisition and manipulation of the knowledge.

In building IMM/Serve, our goal was to produce a flexible, operational system. We used a rule-based approach for several reasons. First, the combination of rule-based and tabular knowledge turned out to be very well suited for this domain. In addition, the rule-based knowledge in IMM/Serve is expressed in a shell which allows the system as a whole to be run very efficiently as a C program on a wide range of different machines. The ultimate goal of the IMM/Def project is also to develop an operational tool, based on the rules used to encode IMM/Serve's knowledge.

As described later in the paper, IMM/Def's "definition logic" expresses the heart of immunization forecasting logic in a quite simple format that preserves its underlying structure, yet allows the full rule-based logic to be generated automatically. IMM/Def makes no attempt to develop a domain-independent approach. Its approach is specially tailored to immunization. As a result. IMM/Def's design may not be easily portable to other clinical problems. A potential advantage is that the IMM/Def may be able to handle the problems of immunization forecasting more naturally than would a general purpose tool.

3. OVERVIEW OF IMM/SERVE'S IMMUNIZATION FORECASTING

This section uses IMM/Serve to illustrate the operation of an immunization forecasting program. IMM/Serve (http://ycmi.med.yale.edu/immserve/) produces recommendations for the following six childhood vaccination series: *Haemophilus influenzae* type b (Hib), Hepatitis B (HepB), Varicella (Var), Measles Mumps Rubella (MMR), Diphtheria Tetanus Pertussis (DTP), and Polio (OPV/IPV). IMM/Serve has been developed in collaboration with the Oregon Health Division, and its logic reflects their desired practice.

IMM/Serve accepts as input a description of child's immunization history, as illustrated in Fig. 1a. These data include: (1) the date of the forecast, (2) the child's birth date, (3) the date and optionally the brand of each previous vaccination, and (4) additional clinical information, such as whether any vaccine is contraindicated and the mother's HBsAg status. Figure 1b shows IMM/Serve's recommendations for the case described.

To produce this analysis, IMM/Serve performs three processing phases.

1. *Preprocessing*. This phase first checks for obvious errors such as dates that do not make sense, and then "screens" the doses of each series, using a decision table, to see if any doses were given too early to be counted as valid vaccinations.

2. *Rule-Based Inferencing*. The second phase involves applying rule-based logic to determine which vaccinations are due now, which should be scheduled next, and which specific set of parameters (ages and time intervals) apply.

3. *Postprocessing*. The final phase involves producing recommendations containing specific dates.

a) Date used for forecast: 4/1/1996
Date of birth: 1/1/95
HepB: 1/1/1995, 3/1/1995, 7/1/1995
DTP: 3/1/1995, 5/1/1995, 7/1/1995
Hib: 3/1/1995, 5/1/1995, 7/1/1995
OPV: 3/1/1995, 5/1/1995, 7/1/1995

b) The following immunization(s) are due on 4/1/1996:
DTaP 4
Hib 4
MMR 1
VAR 1

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

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

FIG. 1. (a) An example immunization history which might be input to the IMM/Serve immunization forecasting program, and (b) the recommendations produced by IMM/Serve for that case.

The second phase, rule-based inferencing, is the most complex part of the program's immunization knowledge and is the focus of this paper.

4. IMM/DEF DEFINITION LOGIC

IMM/Def is a prototype program, written in Lisp, built to demonstrate how a core set of immunization forecasting rules can be generated automatically from a concise description of the logic that is much simpler than the rules themselves. We call this simpler representation of the recommendations "definition logic." This definition logic is not designed to be executed directly. Rather, it is designed to be processed to create the rule-based logic required to analyze a case. Figure 2 shows the definition logic for Hib. This logic has three main components: (1) an "abort" condition, (2) several dose parameter sets, and (3) a set of dose definition rules.

The Abort Condition. At the top of Fig. 2 is a logical condition which, if met, will result in no Hib rules firing. This logic will be incorporated into each rule.

Parameter Sets. At the bottom of Fig. 2 are various sets of parameters that are associated with each dose in different clinical situations. Each parameter set has a name. The exact set of parameters that applies to a particular case depends on the dose definition rules shown immediately above the parameter sets. If "Hib2_final" applies for Hib dose 2, then the following parameters apply.

```
Abort if:   completed_Hib or Hib_contraindicated

Dose 1:     Hib1

Dose 2:     if  (Hib1_age_in_months >= 12
                   and Hib1_age_in_months < 15)  then  Hib2_final (final)
            else  Hib2

Dose 3:     if  (PRPOMP_series
                  or Hib2_age_in_months >= 12
                   or Hib1_age_in_months >= 7)  then  Hib3_final (final)
            else  Hib3

Dose 4:      Hib4
```

Parameter sets:

Hib1	acceptable: 6 weeks recommended: 2 months	Hib3	acceptable: 18 weeks recommended: 6 months dose_wait: 1 month
Hib2	acceptable: 10 weeks recommended: 4 months dose_wait: 1 month	Hib3_final	acceptable: 12 months recommended: 15 months dose_wait: 2 months
Hib2_final	acceptable: 12 months recommended: 15 months dose_wait: 2 months	Hib4	acceptable: 12 months recommended: 15 months dose_wait: 2 months

FIG. 2. IMM/Def's definition logic for the Hib vaccine series, as described in the text.

1. The minimum acceptable age for the dose is 12 months. (If the child is at the clinic for whatever reason on or after this age, the dose may be given.)
2. The recommended age for the dose is 15 months. (This is the age when the child should normally be scheduled for the dose.)
3. The wait interval from the previous dose is 2 months. (Even if the child is over the recommended age, the dose is not due until at least 2 months have passed since Hib dose 1.)

Dose Definition Rules. For each dose, definition logic indicates the conditions under which each set of dose parameters should be used. For example:

1. For dose 1, no logic is specified and only one parameter set applies (Hib1).
2. For dose 2, the logic indicates the following:
   a. If Hib dose 1 was given between ages 12 and 15 months, then Hib dose 2 should be given in accordance with the parameter set labeled "Hib2_final." (The parenthetical term "final" indicates that if this parameter set is used, the series is complete, and no further doses are needed.)
   b. Otherwise Hib dose 2 should be given in accordance with the parameter set labelled "Hib2."
3. For dose 3:
   a. If (1) the brand "PRPOMP" was used for both Hib doses 1 and 2 ("PRPOMP_series" is a terse expression of this fact), *or* (2) Hib dose 2 was

given at or after the age of 12 months, *or* (3) Hib dose 1 was given at or after the age of 7 months, *then* Hib dose 3 should be given in accordance with the parameter set labelled "Hib3_final."

b. Otherwise Hib dose 3 should be given in accordance with the parameter set labeled "Hib3."

4. For dose 4, only a single parameter set applies (Hib4).

The Hib vaccination schedule was chosen for this example because it is of moderate complexity. It is complex enough to illustrate the underlying issues, but not so complex that it might be confusing. The next section shows how IMM/Def processes this Hib definition logic so that it is applied correctly in the three temporal contexts.

5. AUTOMATIC PROCESSING IMM/DEF'S DEFINITION LOGIC TO ACCOMMODATE THREE TEMPORAL CONTEXTS

This section shows how the central kernel of rule-based logic used by IMM/Serve to produce its recommendations is generated automatically by IMM/Def. For each vaccine series, this logic is organized by dose: starting with rules that apply for dose 1, followed by rules that apply for dose 2, etc. In the logic for each dose, separate rules deal with each of the three *contexts* in which the immunization logic must be applied.

1. The dose is due now.
2. The dose is not yet due.
3. The dose must be scheduled to follow the previous dose which is due now.

The core rule-based logic for all six immunization series follows this same basic paradigm for each dose. In addition to these core rules, IMM/Serve's knowledge base includes a number of other, mostly quite simple, rules. These perform "housekeeping" duties, including (1) setting flags (such as "Hib_complete") which are tested by the core rules and (2) determining which vaccine brand or preparation should be used if a preference exists.

The best way to illustrate how IMM/Def processes its definition logic is to work through a specific example. In this section, we describe how this is done for the Hib definition logic described above. A later section discusses certain aspects of this translation process in more detail.

Hib Dose 1. The two rules shown below handle Hib dose 1 for the cases where the dose is *due* and where the dose is *not due yet*, and are created in a straightforward fashion from the definition logic. "Hib_prior" is a variable that is set by the IMM/Serve preprocessor to the number of valid previous Hib doses. "Hib_parameters_met" is a variable set to *true* by the preprocessor if all the parameters of the parameter set "Hib1" are satisfied in the case. Thus the first rule says "if there are no previous Hib doses *and* the abort condition "(completed_Hib or Hib_contraindicated)" does not hold *and* the Hib1 parameter set is satisfied, *then* the first Hib dose is due and the Hib1 parameter set should be used." The logic of the second rule is virtually identical, except that it tests whether the Hib1 parameter set is *not* satisfied. If not, the first Hib dose is *not yet due*, and the Hib1 parameter set should be used to determine when it should be scheduled.

```
if      Hib_prior = 0 and not (completed_Hib or Hib_contraindicated)
              and Hib1_parameters_met
then    due_Hib1

if      Hib_prior = 0 and not (completed_Hib or Hib_contraindicated)
              and not Hib1_parameters_met
then    next_Hib1
```

If the first Hib dose *is* due, then Hib dose 2 should be scheduled to follow. As seen in the definition logic, two parameter sets are possible. The choice depends on the child's age at dose 1. Since we are contemplating giving dose 1 right now, there is no age for dose 1 recorded in the immunization history. As a result, the definition logic must be modified for this temporal context to test the *current* age in months, not the age in months of Hib dose 1. IMM/Def therefore makes these substitutions to the definition logic to produce the two rules shown below.

```
if      (due_Hib1)
           and (age_in_months >= 12 and age_in_months < 15)
then    next_Hib2_final

if      (due_Hib1)
           and (age_in_months < 12)
then    next_Hib2
```

In the second rule, line 2 could have simply been the negation of line 2 from the previous rule, e.g., "and not (age_in_months >= 12 and age_in_months < 15)." (The condition "age_in_months >= 15" does not need to be considered because at or above 15 months of age, Hib dose 1 completes the series.) Negating the conditions used in previous rules is necessary when translating a series of conditions in the definition logic which select among several parameter sets for a dose.

Notice that in this case, however, IMM/Def has recognized that if the condition "(age_in_months >= 12 and age_in_months < 15)" is *not* met, then the condition "(age_in_months < 12)" applies. IMM/Def is able to make this substitution because it has a list of sets of mutually exclusive conditions which it uses to express the logic as concisely as possible, as described later in the paper.

Hib Dose 2. The logic for Hib dose 2 is considerably more complicated than for dose 1 and is composed from the definition logic in a similar fashion. The complexity arises because there are two possible parameter sets that may apply for dose 2, and also for dose 3 (which needs to be scheduled if dose 2 is due). The first four rules test whether Hib dose 2 is *due* or *not yet due* and if so, which parameter set (Hib2_final or Hib2) applies.

```
if      Hib_prior = 1 and not (completed_Hib or Hib_contraindicated)
                and (Hib1_age_in_months >= 12 and Hib1_age_in_months < 15)
                and Hib2_final_parameters_met
then    due_Hib2_final

if      Hib_prior = 1 and not (completed_Hib or Hib_contraindicated)
                and (Hib1_age_in_months >= 12 and Hib1_age_in_months < 15)
                and not Hib2_final_parameters_met
then    next_Hib2_final

if      Hib_prior = 1 and not (completed_Hib or Hib_contraindicated)
                and (Hib1_age_in_months < 12)
                and Hib2_parameters_met
then    due_Hib2

if      Hib_prior = 1 and not (completed_Hib or Hib_contraindicated)
                and (Hib1_age_in_months < 12)
                and not Hib2_parameters_met
then    next_Hib2
```

If Hib dose 2 is due, the following two rules determine whether Hib dose 3 should be scheduled to follow and if so, which parameter set (Hib3_final or Hib3) applies. (If Hib dose 2 is due, but the parameter set Hib2_final applied, no further doses need to be scheduled since this “final” parameter set completes the series.) Notice that here again the definition logic needs to be modified to fit the correct temporal context. “Hib2_age_in_months” has been translated to “age_in_months.” Also, “PRPOMP_series” (a variable set to “true” by the preprocessor if the brand “PRPOMP” has been used for both Hib doses 1 and 2) has been translated to “Hib1_brand = PRPOMP.” (This is appropriate because Hib dose 2 is due but has not been given, and is recommended to be PRPOMP when dose 1 is PRPOMP.)

```
if      (due_Hib2)
                and ((Hib1_brand = PRPOMP) or (age_in_months >= 12)
                        or (Hib1_age_in_months >= 7))
then    next_Hib3_final

if      (due_Hib2)
                and not ((Hib1_brand = PRPOMP) or (age_in_months >= 12)
                        or (Hib1_age_in_months >= 7))
then    next_Hib3
```

Hib Dose 3. The flow of logic for Hib dose 3 is similar to that for the previous two doses. Here only one parameter set is available for use if Hib dose 4 needs to be scheduled next.

```
if      Hib_prior = 2 and not (completed_Hib or Hib_contraindicated)
                and (PRPOMP_series or (Hib2_age_in_months >= 12)
                     or (Hib1_age_in_months >= 7))
                and Hib3_final_parameters_met
```

```
then    due_Hib3_final

if      Hib_prior = 2 and not (completed_Hib or Hib_contraindicated)
                and (PRPOMP_series or (Hib2_age_in_months >= 12)
                        or (Hib1_age_in_months >= 7))
                and not Hib3_final_parameters_met
then    next_Hib3_final

if      Hib_prior = 2 and not (completed_Hib or Hib_contraindicated)
                and not (PRPOMP_series or (Hib2_age_in_months >= 12)
                        or (Hib1_age_in_months >= 7))
                and Hib3_parameters_met
then    due_Hib3

if      Hib_prior = 2 and not (completed_Hib or Hib_contraindicated)
              and not (PRPOMP_series or (Hib2_age_in_months >= 12)
                        or (Hib1_age_in_months >= 7))
              and not Hib3_parameters_met
then    next_Hib3

if      (due_Hib3)
then    next_Hib4
```

Hib Dose 4. The rules for Hib dose 4 are very simple, merely testing whether the Hib4 parameter set is satisfied or not to determine whether the dose is *due* or *not yet due*. Since there is no fifth Hib dose, no logic needs to be included to schedule a subsequent dose.

```
if      Hib_prior = 3 and not (completed_Hib or Hib_contraindicated)
                and Hib4_parameters_met
then    due_Hib4

if      Hib_prior = 3 and not (completed_Hib or Hib_contraindicated)
                and not Hib4_parameters_met
then    next_Hib4
```

This section has shown all 14 rules produced by IMM/Def by processing the Hib definition logic. It is quite striking that although the underlying definition logic (as seen in Fig. 2) is very simple and easy to understand, the logic required to implement that logic for all three temporal contexts is much more complex.

It is important to point out that additional rules, beyond the rule "kernel" shown above, are required to implement completely the Hib logic. These additional rules involve (1) 7 "pre-kernel rules" that set facts (such as "completed_Hib," and "PRPOMP_series") which are tested by the rule kernel and (2) 16 post-kernel rules (most of which are very simple) which may indicate that a specific brand of Hib vaccine should be used. The kernel rules shown above contain by far the most complexity and are by far the most difficult to debug and maintain since they must react comprehensively to a range of clinical conditions in all three contexts.

TABLE 1

Vaccine series	Dose number	Parameter sets	Conditional clauses	Rules generated
DTP	1	1	0	3
	2	1	0	4
	3	2	1	8
	4	4	3	11
	5	3	2	8
	6	2	1	4
HepB	1	2	1	7
	2	3	2	9
	3	3	2	6
Hib	1	1	0	4
	2	2	1	6
	3	2	1	5
	4	1	0	2
MMR	1	1	0	3
	2	1	0	2
Polio	1	2	2	8
	2	4	3	12
	3	4	3	10
	4	2	2	4
Var	1	1	0	3
	2	1	1	2
Total	(21 doses)	43	25	121

Note. For each dose of each vaccine series (as of early 1997), shown are the number of parameter sets and the number of conditional clauses in the IMM/Def definition logic and the number of rules generated from that definition logic. (Where only a single parameter set applies for a dose, the number of clauses is counted as 0. Conditional clauses consisting solely of ‘‘else’’ are also not counted.)

It is also important to point out that the kernel rules shown above are already simpler than they might otherwise be, since the tabular dose parameter sets allow several parameters to be tested as a unit by the rules. If each individual parameter was represented explicitly in the rules (which was the case in an early version of IMM/Serve), then instead of the simple test ‘‘Hib2_parameters_met,’’ for example, the following expression would need to be included: ‘‘(age_in_weeks >= 10 and Hib_elapsed_months >= 1).’’ Including all parameters explicitly would significantly increase the overall complexity of the rules.

Table 1 shows, for each dose of each vaccine series, the total number of parameter sets, the total number of IMM/Def conditional clauses, and the total

TABLE 2

The Hib Portion of the Table Used by IMM/Def to Make Substitutions in Its Definition Logic, as Described in the Text

Hib dose	Search	Replacement
1	Hib1_age_in_months	age_in_months
2	Hib2_age_in_months	age_in_months
	PRPOMP_series	Hib1_brand = PRPOMP
3	Hib3_age_in_months	age_in_months

number of rules produced by IMM/Def. This table illustrates how much simpler the definition logic is compared to the rules required to implement that logic.

6. IMM/DEF DESIGN ISSUES

In the process of working through the example in the previous section, we described several design issues that arise. This section discusses two of these issues in more detail.

6.1. Translating the Definition Logic to Fit the Temporal Context

We have already described several examples where the definition logic must be translated in the "scheduling a dose to follow" context (e.g., when Hib dose 2 must be scheduled next to follow Hib dose 1 which is due). The definition logic for each dose is written assuming that all previous doses have been given. In "scheduling a dose to follow," Hib dose 2 is being scheduled when Hib dose 1 is due but not yet given. In this context, as described above, the definition logic for Hib dose 2 must be modified.

Making Substitutions in the Definition Logic. One way in which the definition logic must be modified in the "scheduling a dose to follow" context is that certain substitutions must be made. For example, "Hib1_age_in_months" must be translated to "age_in_months" when Hib dose 1 is due but not given. IMM/Def performs these substitutions using a table (see Table 2). For each dose, a set of pairs of terms are listed. Whenever the first term of a pair is found in processing of the definition logic in the "scheduling a dose to follow" context, it is replaced by the second term of the pair. For example:

1. For Hib dose 1, "Hib1_age_in_months" must be replaced by "age_in_months."
2. For Hib dose 2, "Hib2_age_in_months" must be replaced by "age_in_months," and "PRPOMP_series" must be replaced by the condition "Hib1_brand = PRPOMP."
3. For Hib dose 3, "Hib3_age_in_months" must be replaced by "age_in_months."

In the previous examples, we have seen the effects of these substitutions on the logic generated.

Adding Conditions to the Definition Logic. In addition to making substitutions in the definition logic when it is implemented in the "scheduling a dose to follow" context, it is sometimes necessary to add logic. This situation occurs in the polio vaccine (PV) immunization logic. For polio, the "abort" condition, includes the condition "age_in_years >= 18" indicating that once the patient is 18 years old, none of the rules apply. A portion of the polio vaccine definition logic is shown below.

```
Dose 4:  if OPV_indicated and (PV3_age_in_years < 4 or prior_IPV) then
         OPV4
         if IPV_indicated and (PV3_age_in_years < 4 or prior_OPV) then
         IPV4
```

Because of the abort condition, none of this logic will be used in any of the three contexts once the patient is 18 years old. In the "scheduling a dose to follow" context, however, when PV dose 3 is due, PV dose 4 must be scheduled next. Since there is a 6-month interval required between the two doses, PV dose 4 should *not* be scheduled next if the patient is over $17\frac{1}{2}$ years (210 months). As a result, the condition "and age_in_months < 210" is added to the rules generated in the "scheduling a dose to follow" context to prevent PV dose 4 from being scheduled inappropriately.

6.2. Enhancing Readability of the Generated Logic

Another issue that arises concerns the readability of the automatically generated logic. IMM/Def takes a very methodical approach to processing the definition logic, but does not have any real-world (semantic) knowledge of how the various conditions interact with one another. A particular problem occurs because there are several sets of mutually exclusive conditions tested by the logic. Unless IMM/Def knows about these relationships between conditions, it will sometimes generate logic that is redundant and awkward. For example, in the Hib logic discussed previously the following rule was generated by a previous version of the system.

```
if      (Hib1_due)
              and not (age_in_months >= 12 and age_in_months < 15)
then    Hib2_next
```

With the current version of IMM/Def, the entire second line is replaced by "age_in_months < 12," which results in more natural logic. (The condition "age_in_months >= 15," which is the third component of this set of mutually exclusive conditions, is ruled out by pre-kernel rules.) This potential awkwardness arises because, as discussed previously, IMM/Def must include logic to negate all conditions used by previous rules to activate parameter sets for a dose when generating the logic to activate a later parameter set. A similar situation arises

TABLE 3

Three of the Sets of Mutually Exclusive Conditions Which IMM/Def Uses to Enhance the Readability of the Rules It Generates

1.	HepB_class = positive
	HepB_class = negative
	HepB_class = unknown
	HepB_class = adolescent
2.	Hib1_age_in_months >= 15
	Hib1_age_in_months >= 12 and Hib1_age_in_months < 15
	Hib1_age_in_months < 12
3.	age_in_months >= 15
	age_in_months >= 12 and age_in_months < 15
	age_in_months < 12

in the Hepatitis B (HepB) rules, where, for example, the following rule was generated by a previous version of the system.

```
if      not HepB_contraindicated and (HepB_prior = 1)
              and not (HepB_class = positive or HepB_class = unknown)
              and (HepB_class = adolescent)
              and HepB2_adolescent_parameters_met
then    HepB2_adolescent_due
```

The variable “HepB_class” may have four values, all of which are mutually exclusive. As a result, if “HepB_class = adolescent,” then “HepB_class” cannot be “positive” or “unknown.” Therefore the second line of the rule is redundant and is now removed.

IMM/Def has a table of mutually exclusive conditions (shown in part in Table 3) which it uses to recognize when its logic could be made more readable and concise by taking advantage of these relationships between certain of the conditions it is processing.

6.3. Generality of the Algorithm

It is important to emphasize that a single version of the IMM/Def program is able to operate successfully for all six vaccine series. For each vaccine series, IMM/Def takes as input data structures which include: (1) the definition logic, (2) tables describing any substitutions and any additions to be made in the “scheduling a dose to follow” context, and (3) a table of any mutually exclusive conditions. From these, the implementation logic is generated. As a result, new vaccine series can be added without changing the program itself. In fact, this was done recently for Hepatitis A.

7. CURRENT STATUS AND FUTURE DIRECTIONS

The current IMM/Def program is a prototype which has been applied successfully to the immunization logic for all six vaccine series used in routine childhood immunization, correctly generating that logic contained in the current version of IMM/Serve. We emphasize, however, that although IMM/Def is able to generate the core immunization rule-based logic, this is only one portion of an immunization forecasting program. Other components include:

1. The preprocessing module which screens the immunization history to identify invalid doses. This task can be done in a fairly straightforward table-driven fashion.
2. The postprocessing module which includes logic to compute specific dates for the future doses to be scheduled. This temporal logic should never need to change in a fundamental way.
3. Additional rule-based logic, some of which sets flags which are tested in the core rules, and some of which recommend specific vaccine brands or preparations to use.

The core rules which IMM/Def generates, however, represent by far the most difficult clinical logic in IMM/Serve.

As we look to the future, one next step will be to gain experience using IMM/Def with different versions of immunization logic: (1) as national panels periodically modify their recommendations and (2) as different states and health organizations require customized versions. This experience will allow us to refine IMM/Def's current capabilities and to see if additional capabilities are required.

8. LESSONS LEARNED

This section discusses certain issues that arose in the implementation of IMM/Def and in its application to the six childhood vaccine series.

8.1. Double-Checking the Rule Kernel vs Generating It Automatically

Although IMM/Def is able to generate automatically the rule kernel for each vaccine series, we have not used it to replace the manual version of the rules. Instead, we continue to maintain the set of rules which were originally created by hand and which have been massaged over time to be essentially identical (semantically equivalent, with minor syntactic differences) to the IMM/Def-generated rules.

The reason we use IMM/Def in this way is that we have found it extremely valuable to have two different representations of the logic which can be cross-checked against one another. In manually maintaining the rules, it is very difficult to think through all the different combinations of the logic and make sure each detail is correct and mutually consistent. On the other hand, because IMM/Def is able to elaborate all these combinations in an automatic, methodical fashion, it can play a very valuable role in helping assure that these issues are handled correctly.

At the same time, especially when making significant changes to the IMM/Def logic, it is very useful to examine all the rules generated in detail and compare them to the hand-generated version of the rules. In the process of so doing, we sometimes find problems in IMM/Def's definition logic. Once those problems are corrected, the generated rules can be used to update the rules in IMM/Serve.

Having two representations of the same knowledge which can be compared through the use of IMM/Def is extremely valuable, particularly as significant changes need to be made to that knowledge. It can be very useful to go back and forth between the definition logic, which represents the logic concisely, and the implementation logic, which spells out all the logical combinations in gory detail, to make sure all those combinations are dealt with correctly.

8.2. The Relative Merits of a Domain-Specific vs a Domain-Independent Approach

An important question concerns the feasibility of developing domain-independent tools to assist in the maintenance of computer-based clinical guidelines. If such tools were available, they could facilitate the knowledge maintenance of guidelines in many clinical domains. If specialized tools need to be developed for each domain, however, then the process of introducing computer-based guidelines into clinical practice will be more difficult, since one will need to develop not only the application itself but also a customized set of tools to help refine and verify the knowledge.

IMM/Def is an example of a domain-specific approach. It takes advantage of a very specific feature of the immunization forecasting domain: the need to apply the same logic in three different temporal contexts. Since this feature is also the source of a great deal of the underlying complexity in building a computer-based guideline for immunization, IMM/Def makes a valuable contribution. We suspect that a domain-independent tool would not have been able to contribute as much, and certainly not as naturally, as this customized solution. To what degree other clinical domain have unique complexities which require customized tools, and to what degree they can benefit from domain-independent tools, is a question for the future.

8.3. Different Approaches to the Simplification of Immunization Forecasting Logic

Because of the complexity of immunization forecasting logic, we have tried in a number of ways to simplify the knowledge. First of all, a major part of the knowledge (including the minimum ages and intervals for each dose) can be readily stored in tabular form. The clinical logic that determines which set of tabular parameters applies to a particular case, however, was more easily represented using rules which were still quite complex.

IMM/Def has allowed us to distill out the core rule-based logic so that it can more readily be examined and understood. Even though the definition logic is in the form of if-then rules, these rules are not designed to be executed. These definition rules are really “logic templates” which are processed to generate the more complex operational logic.

The potential value of the approach can be readily appreciated by comparing the Hib definition logic shown in Fig. 2 with the set of generated Hib rules shown in Section 5. These both express precisely the same logic. The definition logic is vastly simpler and much easier to understand.

9. SUMMARY

The IMM/Def program demonstrates that the core clinical logic of an immunization forecasting program can be expressed using a simple form of definition logic. This definition logic can then be automatically translated into rules that apply that underlying logic in three temporal contexts. A key advantage of using IMM/Def is that it allows us to have two specifications of the logic that we can examine independently and which can therefore be cross-checked to help assure completeness, consistency, and accuracy. The approach has been developed specifically for the immunization forecasting domain. We anticipate that the approach will prove valuable in helping make immunization forecasting programs easier both to build and to maintain as the clinical logic evolves over time.

ACKNOWLEDGMENT

This research was supported in part by NIH Grant G08 LM05583 from the National Library of Medicine.

REFERENCES

1. Coenen, F., and Bench-Capon, T. "Maintenance of Knowledge-Based Systems." Academic Press, New York, 1993.

2. Gostin, L. O., and Lazzarine, Z. Childhood immunization registries. *J. Am. Med. Assoc.* **274,** 1793 (1995).

3. "Report of the Committee on Infectious Diseases." American Academy of Pediatrics, Evanston, IL, 1994.

4. Centers for Disease Control and Prevention. "Recommended Childhood Immunization Schedule—United States, January–June 1996." MMWR 44 (Nos. 51 and 52), pp. 942–943, 1996.

5. Atkinson, W., Gandtt, J., Mayfield, M., and Furphy, L., Eds. "Epidemiology and Prevention of Vaccine-Preventable Diseases." Centers for Disease Control, Atlanta, GA, 1995.

6. Miller, P. L., Frawley, S. J., Sayward, F. G., Yasnoff, W. A., Duncan, L., and Fleming, D. W. Combining tabular, rule-based and procedural knowledge in computer-based guidelines for childhood immunization. *Comput. Biomed. Res.* **30,** 211 (1997).

7. Hripcsak, G. Writing Arden Syntax medical logic modules. *Comput. Biol. Med.* **24,** 331 (1994).

8. ASTM. E1460 Standard specification for defining and sharing modular health knowledge bases (Arden Syntax for medical logic modules). *In* "1992 Annual Book of ASTM Standards," pp. 539–587. American Society for Testing and Materials, Philadelphia, 1992.

9. Hripcsak, G., Ludemann, P., Pryor, T. A., Wigertz, O. B., and Clayton, P. D. Rationale for the Arden Syntax. *Comput. Biomed. Res.* **27,** 291 (1994).

10. Tierney, W., Miller, M., and McDonald, C. The effect on test ordering of informing physicians of the charges for outpatient diagnostic tests. *New England J. Med.* **322,** 1499 (1990).

11. Rind, D. M., Safran, C., Phillips, R. S., Wang, Q., Calkins, D. R., Delbanco, T. L., Bleich, H. L., and Slack, W. V. Effect of computer-based alerts on the treatment and outcomes of hospitalized patients. *Arch. Intern. Med.* **154,** 1511–1117 (1994).

12. Tierney, W. M., Overhage, J. M., Takesue, B. Y., Harris, L. E., Murray, M. D., Vargo, D. L., and McDonald, C. J. Computerizing guidelines to improve care and patient outcomes: The example of heart failure. *JAMIA* **2,** 316 (1995).

13. Buchannan, B. G., and Shortliffe, E. H., Eds. ‘‘Rule-Based Expert Systems: The MYCIN Experiments of the Heuristic Programming Project.’’ Addison–Wesley, Reading, MA, 1984.

14. Shiffman, R. N., and Greenes, R. A. Improving clinical guidelines with logic and decision table techniques: Application to hepatitis immunization recommendations. *Med. Decision Making* **14,** 245 (1994).

15. David, J-M., Krivine, J-P., and Simmons, R., Eds., ‘‘Second Generation Expert Systems.’’ Springer-Verlag, New York, 1993.

16. Musen, M. A., Gennari, J. H., Eriksson, H., Tu, S. W., and Puerta, A. R. PROTÉGÉ-II: Computer support for development of intelligent systems from libraries of components. *In* ‘‘Proceedings of Medinfo ’95’’ (R. A. Greenes, H. E. Peterson, and D. J. Protti, Eds.), pp. 766–770. North Holland, New York, 1995.