

**FOCUS**
**CLINICAL & BUSINESS INTELLIGENCE**

# The Immunization Calculation Engine

## Open Source Clinical Decision Support for Immunizations

By Michael J. Suralik, MCS; Noam H. Arzt, PhD, FHIMSS; Daryl Chertcoff; and Regina Austin

### ABSTRACT

Ensuring that patients receive recommended vaccines is an integral aspect of preventive care and an important public health function. However, as the immunization schedule and the number of vaccines continue to grow, it becomes more challenging for providers and public health agencies to monitor and consistently adhere to the frequently changing guidelines. Computer-based clinical decision support (CDS) can be an effective strategy for improving the quality of care. However, many clinical information systems either lack CDS or are slow to keep their CDS in line with the changing guidelines. A public-private partnership of healthcare and information technology experts are collaborating to provide a solution to this problem through the development of a free, modular, open-source immunization CDS system, which can be integrated with immunization information systems, electronic health record systems and other clinical information systems. This Immunization Calculation Engine (ICE) is standards-based, deployable in diverse technical environments, fully configurable and may be maintained by clinical subject matter experts without the assistance of programmers. The ICE collaboration is composed of two public health departments, HLN Consulting, LLC, and the University of Utah, Department of Biomedical Informatics.

### KEYWORDS

Clinical decision support, CDS, immunization, open source, immunization information systems, public health, electronic health record, EHR, electronic medical record, EMR.

**T**HE DEVELOPMENT and routine administration of vaccines is one of the most important public health achievements of the 20<sup>th</sup> century. Routine vaccination is an integral component of preventive care in the United States and has led to a dramatic reduction in the incidence, morbidity and mortality of a number of diseases. As the field of medicine advanced over the last century, the number of vaccine preventable conditions grew dramatically, as did the number of routinely recommended immunizations. In the 1980s, a 5-year-old child would be up-to-date (UTD) with 10 doses of three vaccines (DTP, Polio and MMR)<sup>1</sup> protecting against seven diseases. In 2013, a 5-year-old healthy, fully immunized child would need 9 different vaccines, not counting the annual influenza dose. Those nine routinely recommended vaccines protect against 13 diseases and need to be administered in 28 different doses,<sup>1</sup> making tracking a very challenging task.

Each of these vaccines is recommended at a specified age, often being invalid if given below a certain minimum age. In addition, these vaccines have to be administered several times to be maximally effective. Each dose has spacing requirements from the previous dose, as well as from doses of other vaccines. The Advisory Committee on Immunization Practices (ACIP)<sup>2</sup> defines

**FOCUS: THE IMMUNIZATION CALCULATION ENGINE**

**COMPUTER-BASED CLINICAL DECISION SUPPORT (CDS)** can be an effective strategy for improving the quality of care. However, many clinical information systems either lack CDS or are slow to keep their CDS in line with the changing guidelines.

and publishes a recommended immunization schedule, which constitutes the best practices for immunization, and which is updated and refined several times per year. The routine childhood schedule for 0-18-year-olds has 13 separate footnotes, with as many as 13 sub-bullets for some footnotes. The end result is that it is difficult for providers to monitor and consistently adhere to the ACIP guidelines<sup>3</sup> which are lengthy, complicated and growing.

### **CLINICAL DECISION SUPPORT FOR IMMUNIZATIONS (CDSi)**

One strategy for helping providers to improve the quality of their care is to employ the use of computer-based clinical decision support (CDS)<sup>4</sup> at the point of care. Through a number of techniques, CDS systems bring medical knowledge to bear in the context of a specific patient's medical history to assist in diagnosing a patient's condition. Many, though not all, CDS systems are rule-based: by evaluating patient data against a specific set of rules that leverage published medical knowledge, they help the clinician determine a diagnosis and often suggest a course of treatment.

Immunization Information Systems (IIS), known originally as Immunization Registries, are specifically designed to help providers increase immunization coverage rates and were among the first to provide CDS for immunization (CDSi), sometimes referred to as immunization forecasting. One of the 2013-2017 Centers for Disease Control and Prevention (CDC) IIS Functional Standards<sup>5</sup> specifies that, "The IIS has an automated function that determines vaccines due, past due, or coming due ("vaccine forecast") in a manner consis-

tent with current ACIP recommendations. Any deficiency is visible to the clinical user each time an individual's record is viewed." More specifically, CDC defines CDSi as "an automated process that determines the recommended immunizations needed for a patient and delivers these recommendations to the healthcare provider." Many IIS already meet this standard.

Development and maintenance of CDSi requires a multidisciplinary team with fairly sophisticated background and training: medical/nursing staff to understand and interpret the clinical guidelines; analytical staff to translate the medical rules into computer-accessible instructions; programming staff to code the rules and related interfaces in a programming language; testers to develop test cases and test the algorithm as it is developed; and project managers to make sure all the participants work productively together on schedule and within budget.

Most common CDSi software in IIS today generate clinically accurate CDSi for groups of vaccines that are routinely administered to children, adolescents and adults in accordance with the ACIP guidelines. CDSi software includes evaluation of the validity of each immunization in a patient's history, as well as a recommendation for each vaccine group (e.g., the date on which the next dose is due, series completed, etc.)

Although CDSi software are functionally adequate in most cases, they have some limitations. First, in order to implement even minor changes to the immunization schedule, software developers often must modify the source code. Testing procedures for changes usually require an iterative series of manual steps that are performed by up

to three different sets of individuals: the software developers who verify their modifications or additions, the business analysts working with the developers who verify the changes against the agency's specifications, and immunization program personnel who have deep experience with the immunization schedule and do the final acceptance testing to ensure that the changes have been made to their satisfaction. In addition, because of the inherent complexity of the subject matter, as well as the complexity of the software implementation, regression testing must also be undertaken to ensure that a change to one rule does not "break" another.

A second limitation of common CDSi software is that it only supports a single immunization schedule. This means that IIS do not have the option of evaluating a record against a second immunization schedule, for example, to determine if students are up-to-date with just the specific immunizations that are required for admission to school.

### **ICE INITIATIVE**

Two public health departments, HLN Consulting, LLC, and the University of Utah, Department of Biomedical Informatics are collaborating on the development of a new CDSi system that is named the Immunization Calculation Engine (ICE). The design goals of the ICE initiative include: (1) the ability to support multiple immunization schedules, (2) the ability to simultaneously process multiple requests for CDSi, (3) the implementation of a fully automated testing process, and (4) the creation of graphical user interface (GUI) tools that empower clinically-oriented subject matter experts (SMEs) to update and maintain the immu-

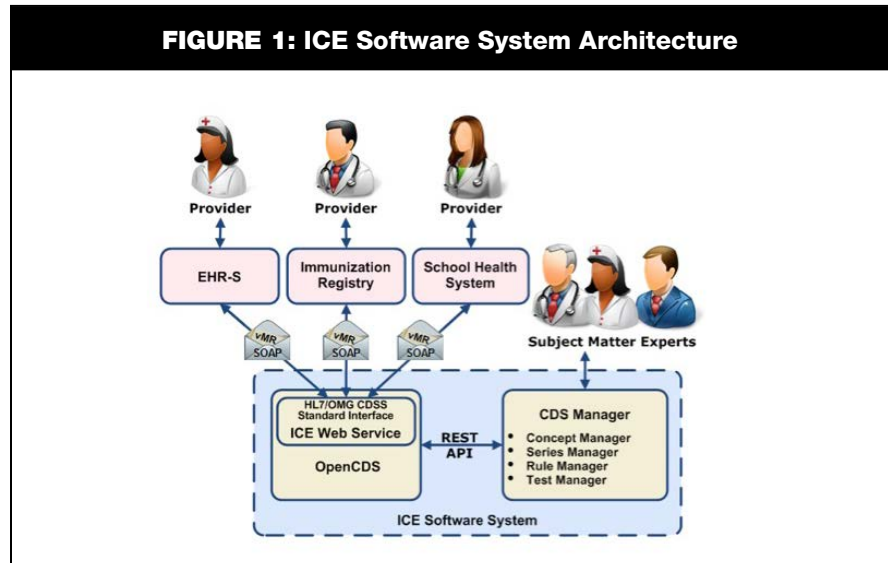
**FOCUS: THE IMMUNIZATION CALCULATION ENGINE**

nization schedule without any involvement from programmers. In addition, after the ICE software is completed, it will be released as an open source system, so that any IIS, EHR-S, health information exchange (HIE), or other clinical information system may utilize ICE for the benefit of all patients. Therefore, it was also required that ICE be a self-contained module that could be deployed in diverse technical environments and accessed by other systems through a standards-based Web Service interface.

There were no known CDSi systems that meet all of the stated design goals. In particular, there were no known systems that empower SMEs to update the immunization schedule without any involvement from programmers, are accessible via a Web Service interface that is based on national standards for CDS services and are available via a commonly used, standard open source license. The project team researched existing CDS frameworks that could be leveraged to create an implementation of ICE that would meet the stated design goals. The CDS framework used was OpenCDS, an open-source, standards-based, service-oriented framework for software developers that is designed to support the delivery of CDS in any healthcare domain (available online at: [www.opencds.org](http://www.opencds.org)). The OpenCDS software is itself the result of a multi-institutional collaborative effort, which is spearheaded by the OpenCDS team at the University of Utah Department of Biomedical Informatics.

### ICE SOFTWARE SYSTEM

The ICE software system has two major components. The core component is the ICE Web Service, which runs in OpenCDS. Third-party clinical information systems are able to integrate with this CDSi service, which evaluates a patient's immunization history and generates the appropriate immunization recommendations for the patient. The second component is the Clinical Decision Support Manager (CDS Manager). This web-based tool with a GUI sits alongside of OpenCDS and enables SMEs to configure and manage ICE without the intervention of software developers.

**FIGURE 1: ICE Software System Architecture**

Through this tool, SMEs may manage the concepts, series and rules that are utilized by ICE, as well as manage and run automated tests of ICE.

When it is completed, the entire ICE software system will be publicly released under the open source GNU Lesser General Public v3 (LGPL v3) license which permits any system, including commercial systems, to freely use, modify and/or integrate with the ICE software system. Under this license, any organization or individual that modifies the ICE software system must freely share the modifications with the open source community.

### TECHNICAL ARCHITECTURE

The ICE software system is implemented with Enterprise Java Beans (EJBs), runs on a Java application server, and utilizes the JBoss Drools rules engine. It can support other rule engines if desired. ICE utilizes a standards-based SOAP Web Service interface, so that it easily integrates with IIS, EHR-S, HIEs and other healthcare information systems. ICE is aligned with the CDS standards from the Centers for Disease Control and Prevention (CDC), Health Level Seven International (HL7), and the Object Management Group (OMG). These standards include the OMG & HL7 Clinical Decision Support Service (CDSS) standard,<sup>6</sup> which specifies the technical capabilities and interfaces of a CDSS, as well as the

Virtual Medical Record (vMR) standard,<sup>7</sup> a standards project within the HL7 Clinical Decision Support Workgroup, which specifies a data model for representing clinical information input and outputs within CDS engines. The CDS Manager deploys concepts, series and rules to the ICE Web Service via the Representational State Transfer (REST) application programming interface (API). Strong reliance on existing standards—and participation by the ICE project team in the continuing development of those standards – ensures the compatibility of ICE with national initiatives to promote health information exchange and interoperability. **Figure 1** illustrates the ICE Software System Architecture.

### ICE WEB SERVICE

Client applications may invoke ICE by passing a vMR to its standard CDSS Web Service. ICE utilizes its immunization rules and the data in the vMR, including the patient's date of birth, gender, immunization history and disease indicators, to evaluate and return the validity of each immunization in the patient's history, along with one or more reasons an immunization is invalid, if it is. ICE also returns a recommendation for each vaccine group (e.g., the date on which the next dose is due, series completed, etc.)

ICE's Web Service architecture scales to support simultaneous real-time pro-

**FOCUS: THE IMMUNIZATION CALCULATION ENGINE**

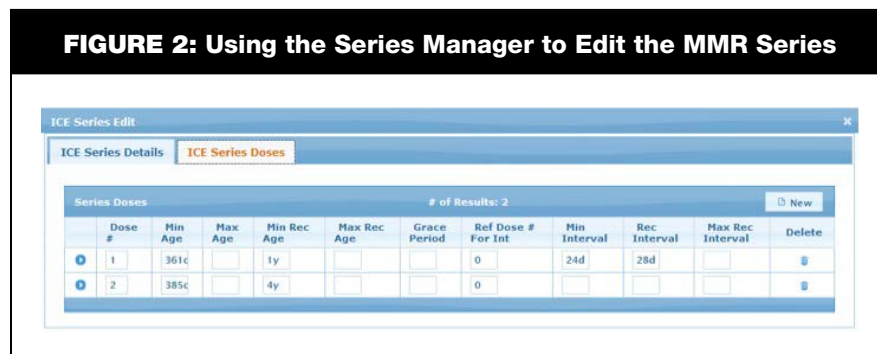
cessing of many patients submitted by one or more systems. It can also scale to service requests for multiple immunization schedules. For example, a single ICE deployment could enforce the standard ACIP rules that would be utilized by IIS and EHR-S, as well as a second set of rules used by a school health system to determine if students have received just the immunizations required for admission to school, and finally a third set of rules utilized by an EHR-S that is deployed in a special clinical setting with unique immunization requirements. Because ICE is Java-based, it can be deployed in diverse technical environments on a variety of hardware and operating system platforms.

In the summer of 2013, an SME workgroup consisting of a dozen immunization and information technology experts will finish configuring ICE for the full set of fourteen vaccine groups that are routinely administered to children, adolescents, and adults (i.e., hepatitis B, rotavirus, diphtheria-tetanus toxoids- acellular pertussis (DTaP), *Haemophilus influenzae* type B, pneumococcal conjugate, pneumococcal polysaccharide, polio, influenza, measles-mumps-rubella (MMR), varicella, hepatitis A, human papillomavirus (HPV), meningococcal, and H1N1). Because ICE is flexible and fully configurable, SMEs at any organization may configure it to support additional vaccine groups, if new ones are added in the immunization schedule in the future.

### CDS MANAGER

The ICE configuration consists of code sets, rules and other data that are stored in a flexible knowledge repository. SMEs may manage the configuration of ICE through the four sub-components of the CDS Manager: a) the Concept Manager, b) Series Manager, c) Rule Manager, and d) Test Manager.

**Concept Manager.** The Concept Manager provides the necessary functionality that enables SMEs to create and manage the code sets, referred to as concepts, which are utilized by ICE. The current set of ICE concepts includes, but is not limited to, diseases, vaccines and vaccine groups, as well as evaluation reasons and recom-



mendation reasons. When a user selects a concept, a new window pops up that enables the user to view and edit the code values for that concept. For example, the user can select the list of vaccines currently supported by ICE, and then add the name and CVX code of a new vaccine that just came into the market. With this capability SMEs can adapt ICE to changing ACIP requirements and the specific needs of their organizations.

**Series Manager.** The Series Manager enables SMEs to modify existing series or create new series. The purpose of a series is to specify a typical path to immunity<sup>8</sup> for one or more diseases through a series of immunizations that are administered at the appropriate ages and intervals. Any number of series can be defined for each vaccine group. Through screens such as that depicted in Figure 3, users manage the fundamental parameters for each dose in a series, such as minimum ages and intervals, and recommended ages and intervals. All of the Series Manager parameters are expressed simply as values in a table. Figure 2 presents all the parameters that a user can manage through the Series Manager, in this case for the MMR series. For example, for dose 1, the recommended age is one year of age, the minimum age for a vaccine to be valid is 361 days (365 days minus a 4-day grace period), and the minimum interval to the next dose is 28 days. The recommended age for dose 2 is 4 years of age.

**Rule Manager.** Although much of the data necessary to correctly evaluate and forecast can be expressed through the table-driven Series Manager, some of the necessary rules require a more sophisticated representation. In those cases,

SMEs may use the Rule Manager to create or edit rules. These rules fall into four categories: disease immunity, immunization history evaluation, series selection, and forecasting. The ability of non-technical, clinically-oriented SMEs to manage even the most complicated immunization rules on their own, without the intervention of a programmer, is a significant innovation that improves the maintainability of the immunization rules.

ICE stores the rules inside the knowledge repository of JBoss Drools, which is a robust rule engine that is well suited to support CDSi. Rule engines are powerful tools for implementing programmatic solutions to problems with algorithms that are complex, change frequently, or are not fully understood. They can support an explanation capability, which logs decisions along with the rules (i.e., the reasons) that the rule engine utilized to make its decision. The use of a rule engine also enables software developers to build solutions that represent business logic in ways that non-technical SMEs can understand and work with directly.

When writing rules, SMEs choose from a list of context-aware, dropdown sentence templates, or “sentences” that together form the complete rule. ICE ships with a suite of sentence templates that can be used as is, and additional sentence templates can be added by administrators to change or expand the authoring capabilities of the editor. **Figure 3** depicts a rule for immunity to Varicella. This rule specifies that if a patient who was born before 1980 has not completed the series for Varicella, then ICE should “conditionally recommend” a Varicella immunization for high-risk groups. ICE will utilize this rule in conjunction with the other relevant rules



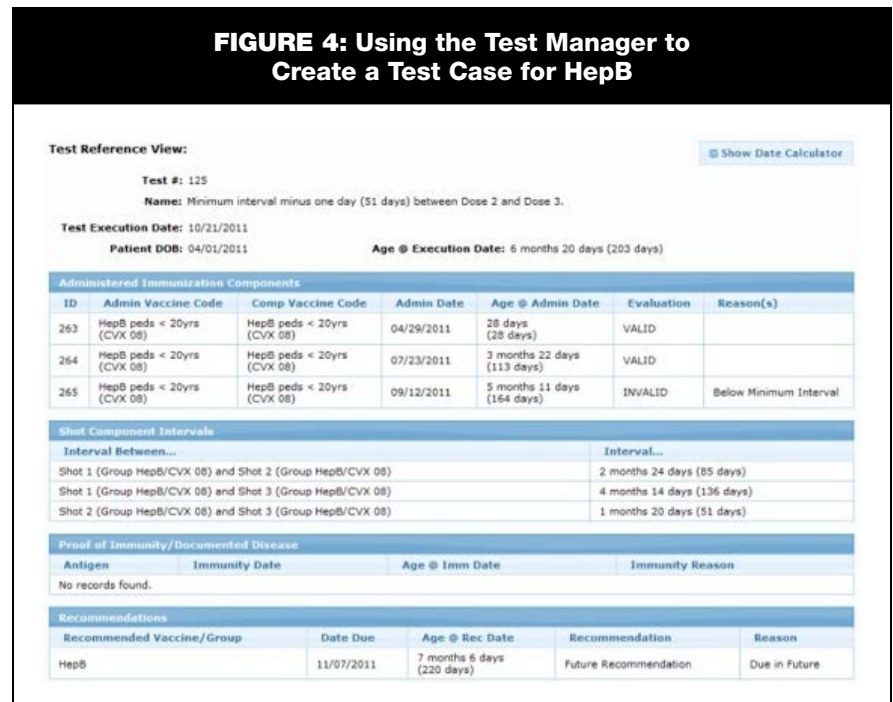
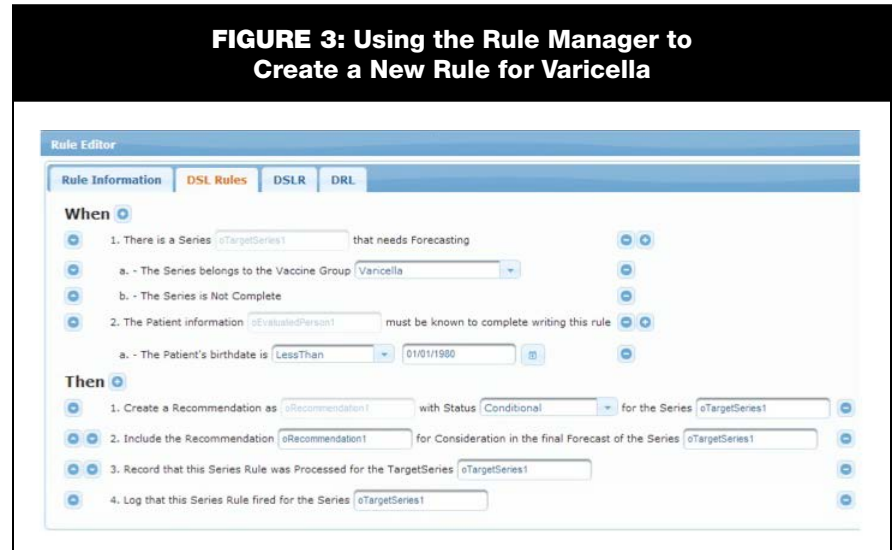
## FOCUS: THE IMMUNIZATION CALCULATION ENGINE

and Series Manager parameters to generate a recommendation for the patient.

**Test Manager.** The Test Manager is a web-based tool that makes it easy for SMEs to create, manage and run test cases against implemented rules without the assistance of a software developer. As rules are implemented, an SME can create corresponding test cases and instantly run those tests to see if they pass or fail. As rules change and regression testing is needed, an entire suite of previously created tests can be re-used, saving SMEs from having to re-create test cases each time. The Test Manager dramatically reduces the time and effort necessary to validate ICE whenever modifications are made to the rules.

When creating a test case, the SME names the test based on the scenario being tested, documents the rule being tested, and selects the vaccine group(s) that will be used in the test. The SME then enters the test details, such as the patient's date of birth, the execution date of the test and the types and dates of the administered vaccines. The SME also enters the expected results for the test case, including the expected evaluation result for each shot administered (e.g., Valid or Invalid / Below Minimum Age), and the expected recommendation result (e.g., Recommended for a future dose on 06/30/2015). **Figure 4** illustrates a specific test case designed to test the minimum interval between dose 2 and 3 of Hepatitis B (8 weeks). This is an example of a border case, where the third shot meets the absolute minimum age requirement (24 weeks minus 4 days or 164 days) but is one day short of the minimum interval between dose 2 and 3 (8 weeks minus 4 days).

Each time that the Test Manager runs a test, it automatically submits that test case's input parameters to the ICE Web Service, receives the response that is returned by ICE, compares the actual evaluations and recommendations in that response to the expected evaluations and recommendations that are stored as a part of that test case and determines if there are any differences between the actual and expected values. If there are no differences, then the Test Manager counts that test as having passed. If there are any differences, then



it counts the test as having failed. The Test Manager and the ICE Web Service work together to process hundreds of test cases in a matter of seconds, which is useful for regression testing. As of March 2013, the Test Manager library contained 1,171 test cases and was still growing. In a test environment, the entire library of tests was executed within 70 seconds (60 ms/test).

When the automated test run is com-

pleted, the Test Manager provides summary data about the test run. An example screen shot of the summary of test results is presented in **Figure 5**. In addition, the Test Manager displays the test results in a list view that includes the name of each test along with determinations of whether the evaluations, recommendations, and test (as a whole) passed. For any tests that failed, the Test Manager will display a "Differenc-

**FOCUS: THE IMMUNIZATION CALCULATION ENGINE**

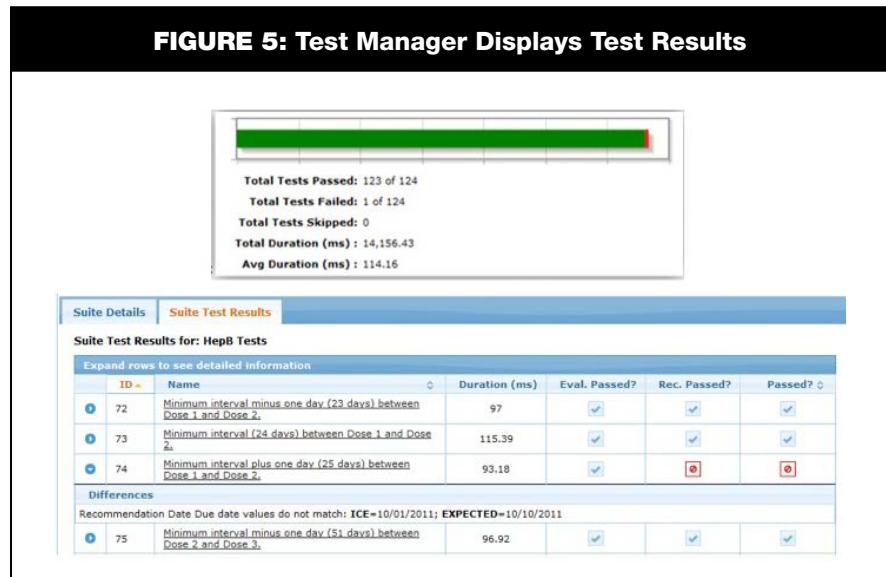
es” section directly below the test, which will clearly indicate why the test failed. **Figure 5** presents the results for 4 test cases, with one case “not passing” because there was a difference between the expected and actual due date of the next dose for that case. The SME may then click on the test name to review the details of that test, correct the test if necessary, or, if the test details are correct, utilize the other components of the CDS Manager to find and correct the rule that is causing the failure.

### THE FUTURE OF ICE AND CDS

Attention to CDS will increase as the Centers for Medicare and Medicaid Services (CMS) EHR Incentive Programs intensifies. One of the core set of measures in both Stage 1 and Stage 2 Meaningful Use involve implementation of CDS to support clinical quality. All indications are that Stage 3 will raise the bar even further and expect even more use of CDS.

As the ICE initiative moves forward, it will face some challenges. Not every organization that seeks to utilize ICE will have the ideal skill set to easily develop interfaces that conform to the standards for clinical decision support services, which are relatively new and robust. Also, once ICE is publicly released, effort will be required to monitor and coordinate the assimilation of any enhancements that organizations may choose to make to this open source software.

Future enhancements to ICE may include the development of adapters that enable systems to send and receive data in the CDA (Clinical Document Architecture) and other standard formats in addition to the vMR format that ICE already accepts. In addition, one can envision enhancing the Test Manager so that it enables users to not only verify that the test cases pass, but for each test case, view which rules fired and the order in which they fired. This would enable users to determine that the rules that are being utilized to arrive at the answer are the correct ones. Furthermore, the Test Manager could be enhanced to perform test coverage analysis to ensure that the test case suites adequately exercise all of the rules and encompass the immunization schedule as a whole. Finally,

**FIGURE 5: Test Manager Displays Test Results**


since the underlying OpenCDS platform is designed to support a variety of clinical domains, the CDS Manager could be enhanced and extended to manage concepts, rules, and automated tests beyond the immunization domain.

### CONCLUSION

ICE is a modular, open source CDSi system that delivers patient-specific immunization evaluations and recommendations to third-party clinical information systems. It runs on a wide variety of hardware and operating system platforms and is accessible through a standards-based web service. Clinical information systems that integrate with ICE will enable providers to ensure that their patients are up to date on their recommended immunizations despite the complicated and frequently changing immunization schedule, which will only continue to grow as new vaccines are developed. **JHIM**

**Michael J. Suralik, MCS**, is a Project Manager with HLN Consulting, LLC, where he heads HLN's ICE project and provides IT consulting services to public health organizations and their partners. For ten years, he has worked with immunization information systems and led teams to produce business analysis, develop custom software and integrate healthcare information systems. He can be reached at [suralik@hln.com](mailto:suralik@hln.com).

**Noam H. Arzt, PhD, FHIMSS**, is president of HLN Consulting, LLC. He has been working with immunization information systems projects for nearly twenty years. He can be reached at [arzt@hln.com](mailto:arzt@hln.com).

**Daryl Chertcoff** is a project manager and solution architect at HLN Consulting, LLC and has been designing and supporting public health systems for nearly fifteen years. He is the technical lead for HLN's ICE project. Daryl can be reached at [daryl@hln.com](mailto:daryl@hln.com).

**Regina Austin** is a senior business analyst at HLN Consulting, LLC and has been supporting the design and development of public health systems for nearly ten years. Austin has been the primary lead on the business team developing rule definitions and test cases for ICE. She can be reached at [regina@hln.com](mailto:regina@hln.com).

### REFERENCES

- 1,2. <http://www.chop.edu/service/vaccine-education-center/vaccine-schedule/history-of-vaccine-schedule.html>
3. <http://www.cdc.gov/vaccines/schedules/index.html>
4. <http://www.cdc.gov/vaccines/schedules/downloads/child/0-18yrs-schedule.pdf>
5. <http://jamia.bmj.com/content/14/2/141.full>
6. <http://www.cdc.gov/vaccines/programs/iis/func-stds.html>
7. [http://www.hl7.org/implement/standards/product\\_brief.cfm?product\\_id=12](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=12)
8. [http://www.hl7.org/implement/standards/product\\_brief.cfm?product\\_id=270](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=270)
9. <http://www.cdc.gov/vaccines/programs/iis/interop-proj/downloads/logic-spec-acip-rec.pdf>