

# Meeting the Electronic Health Record “Meaningful Use” Criterion for the HL7 Infobutton Standard Using OpenInfobutton and the Librarian Infobutton Tailoring Environment (LITE)

James J. Cimino, MD;<sup>1</sup> Xia Jing, MD, PhD;<sup>1</sup> and Guilherme Del Fiol, MD, PhD<sup>2</sup>

<sup>1</sup>Laboratory for Informatics Development, NIH Clinical Center, Bethesda, MD

<sup>2</sup>Department of Biomedical Informatics, University of Utah, Salt Lake City, UT

## Abstract

*Infobuttons are clinical decision support tools that use information about the clinical context (institution, user, patient) in which an information need arises to provide direct access to relevant information from knowledge resources. Two freely available resources make infobutton implementation possible for virtually any EHR system. OpenInfobutton is an HL7-compliant system that accepts context parameters from an EHR and, using its knowledge base of resources and information needs, generates a set of links that direct the user to relevant information. The Librarian Infobutton Tailoring Environment (LITE) is a second system that allows institutional librarians to specify which resources should be selected in a given context by OpenInfobutton. This paper describes the steps needed to use LITE to customize OpenInfobutton and to integrate OpenInfobutton into an EHR.*

## Introduction

In its recent Notice of Proposed Rule Making on the requirements for certifying “meaningful use” of electronic health records (EHRs), the Office of the National Coordinator for Health Information Technology (ONC) states “We propose to require the HL-7 Context-Aware Knowledge Retrieval (“Infobutton”) Standard, International Normative Edition 2010 for retrieving diagnostic or therapeutic reference information.”[1] The standard to which this proposal refers is an information model that defines the structure and semantics of a request from an EHR to a knowledge resource that enables the resource to respond with context-specific information.[2] The full standard may appear large and complex, giving the impression that implementation would be costly. However, the requirement can be met, at a minimum, by providing in the EHR user interface a Uniform Resource Locator (URL) link that passes some patient information (say, a medication or diagnosis) to an HL7-compliant knowledge resource. Many such resources are now available; for example, MedlinePlus Connect, at the National Library of Medicine (NLM) is a source of authoritative health information for patients and health care providers that supports the HL7 standard. Integrating it into an EHR to meet the proposed ONC requirement would simply require an EHR to dynamically generate the following URL:

<http://apps.nlm.nih.gov/medlineplus/services/mpconnect.cfm?mainSearchCriteria.v.dn=<clinical concept>>

where “<clinical concept>” would be replaced with the clinical term of interest (medication, diagnosis, etc.).\*

While the above link will retrieve information relevant to specific patient data, it fails to take advantage of the rich set of parameters that the HL7 standard uses to represent the *context* in which the user is evoking the resource. However, if the resource can make use of data such as patient age and gender to provide more targeted information, the addition of these parameters to the URL should require only a small additional effort by the EHR developer.

A hard-coded EHR-to-resource URL, such as the one above, will meet the letter of the proposed ONC requirement, but it meets the spirit of the proposal in only a minimal way by providing a one-size-fits-all solution. Studies of the information needs of EHR users have repeatedly shown that a variety of clinical questions, requiring multiple resources, arise – sometimes varying based on the type of user (nurse, physician, student, patient, etc.).[3-5] Some EHR developers have addressed this problem by interposing an intermediary system (sometimes referred to as an *infobutton manager*) that uses the context parameters to select from a list of resources those that appear to be appropriate to the situation, returning to the user a page of links. The user can then select the link that is most appropriate to his or her specific information need.[6-9]

---

\* This scenario assumes a Web-based EHR. For a non-Web-based EHR, the client application would need to be able to evoke a Web browser on the user’s computer, passing the URL to the browser.

OpenInfobutton is an HL7-compliant infobutton manager that is maintained at the University of Utah.[10] It is freely available; EHR developers can now provide their users with a link to OpenInfobutton, rather than a single knowledge resource. However, the list of links that OpenInfobutton returns are specific to the user's institution. This is appropriate, since different institutions will have different local resources (such as guidelines and policy manuals) and licenses to different commercial resources. Each institution therefore needs one or more people (such as a medical librarian or someone acting in that capacity) to be responsible for determining the list of resources to be made available to the users and the contexts in which those resources should be selected by OpenInfobutton.

OpenInfobutton maintains institution-specific information about resources and contexts in a set of XML files. Such files can be created with a simple text editor, but getting the syntax and semantics right is not easy. OpenInfobutton does not currently include a user-friendly knowledge base editor. However, a second application, called the Librarian Infobutton Tailoring Environment (LITE), provides this capability. An early version of LITE was originally created at Columbia University; the current version, created and maintained at the National Library of Medicine, is hosted at the University of Utah.[11] Figure 1 shows the relationship between EHR users, the EHR, OpenInfobutton, LITE, and the medical librarian. The Utah installations of OpenInfobutton and LITE are freely available for use in any EHR that wishes to use them.

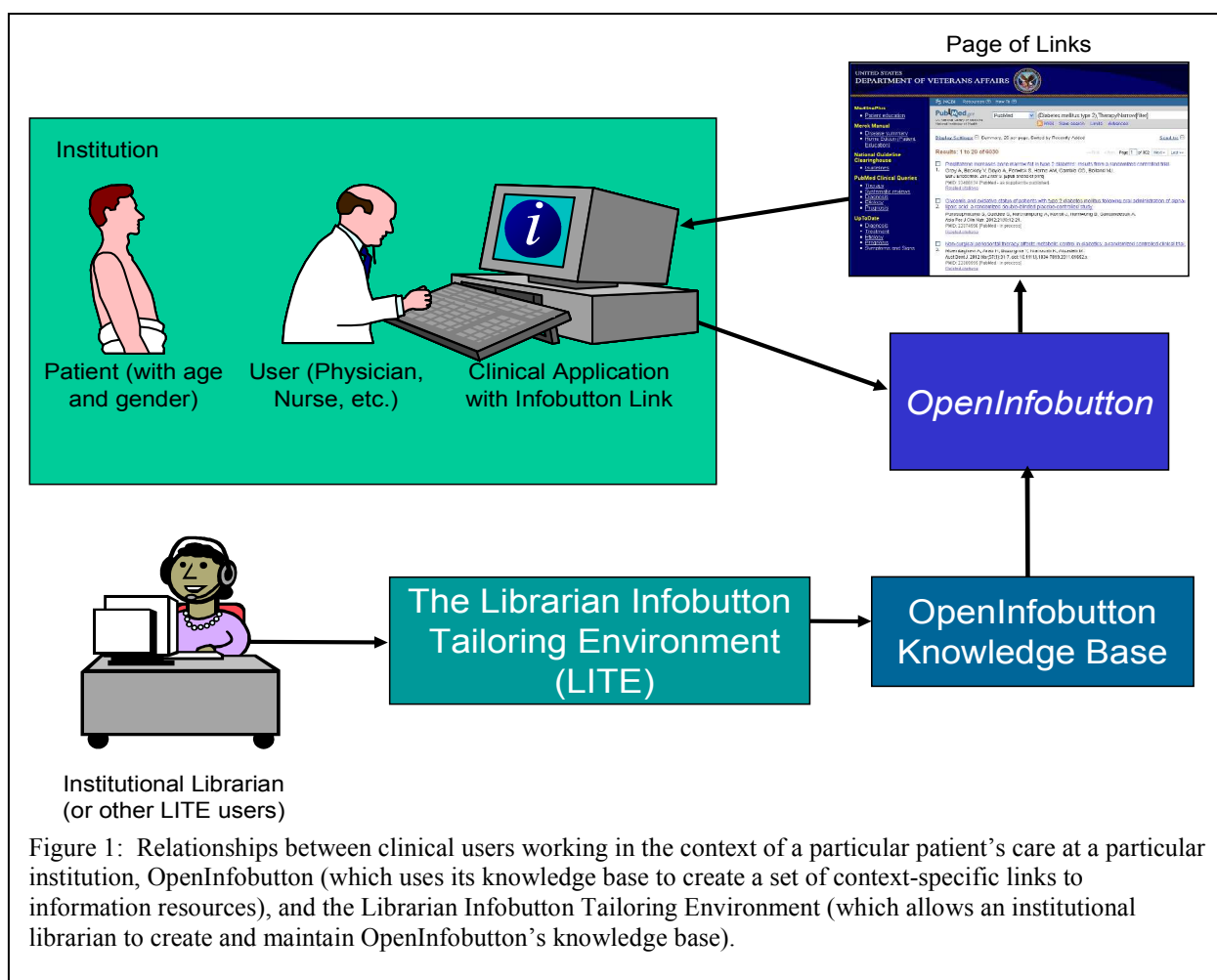


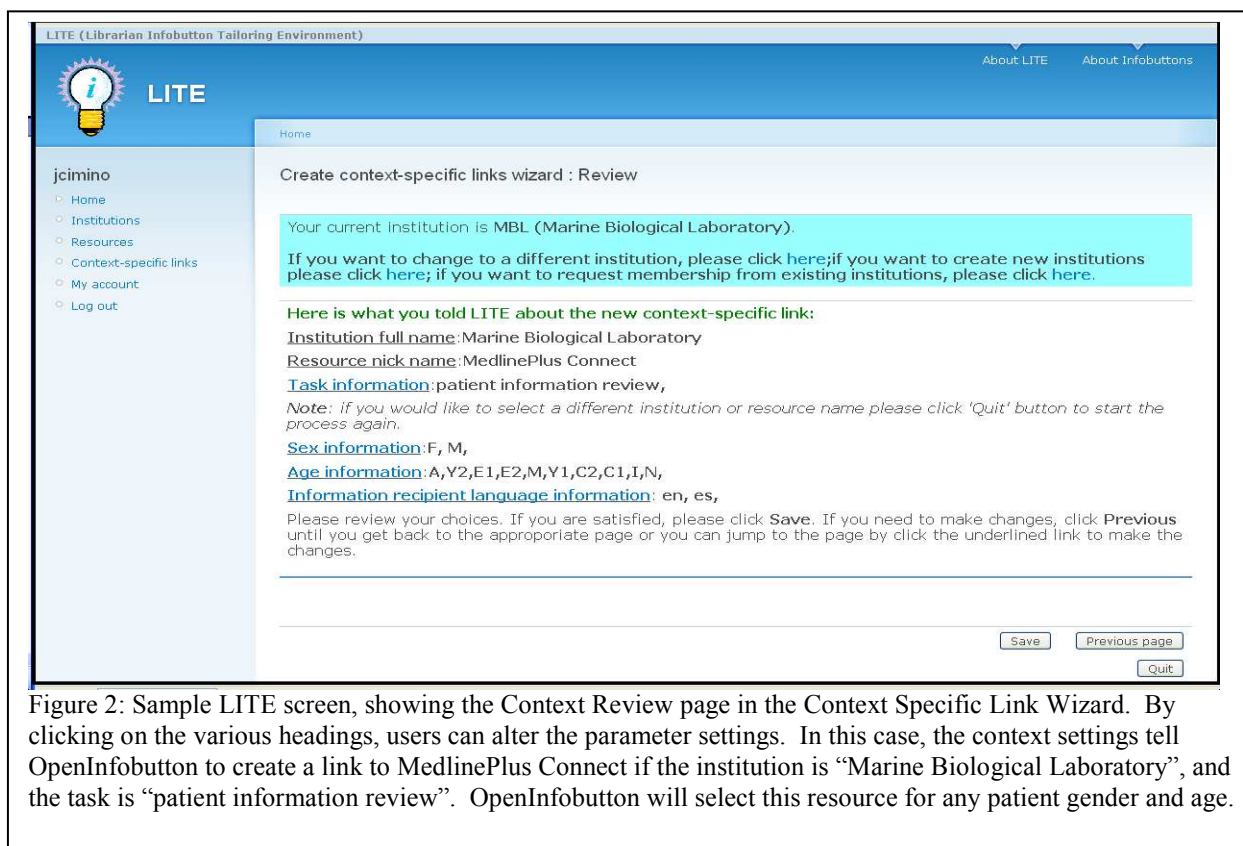
Figure 1: Relationships between clinical users working in the context of a particular patient's care at a particular institution, OpenInfobutton (which uses its knowledge base to create a set of context-specific links to information resources), and the Librarian Infobutton Tailoring Environment (which allows an institutional librarian to create and maintain OpenInfobutton's knowledge base).

The combination of OpenInfobutton and LITE offers a way forward for EHR developers to meet the ONC's requirement for using the HL7 infobutton standard, including the use of resources that do not, themselves, comply with the HL7 standard. A previous paper examined the experience of various implementers of HL7-compliant infobutton managers.[12] The purpose of this paper is clarify the roles of OpenInfobutton and LITE and to describe a method by which institutions can integrate an infobutton manager (OpenInfobutton) into their own EHR without requiring major system development. A key aspect of this method is the capability for each institution to be able to customize the infobutton manager (using LITE) to meet the information needs of its constituents.

## Using LITE

The HL7 standard includes many context parameters that relate to the role and task of the EHR user, as well as demographic and clinical aspects of the patient at hand. The intention of these parameters is to be used as *selection criteria*, for determining which resources should be chosen by OpenInfobutton in a particular situation, and as *search criteria*, for inclusion in the links to the knowledge resources to enable those resources to select relevant information. However, the HL7 parameters do not describe the resources themselves. LITE allows users to specify the subject matter of a resource by indicating the acceptable values for a particular parameter, which may be a subset of all the values allowed by the standard. In the case of non-HL7-compliant resources, LITE also allows users to describe information about resource parameters. LITE then generates the XML files used by OpenInfobutton.

LITE has been built using a Web-based, open-source content management system called Drupal.[13] LITE users can choose from a menu of options to define their institution, manage user memberships to institutions, describe resources, specify which resources should be selected for particular context values, and test their specifications. Figure 2 shows a typical LITE screen, with the menu of functions on the left and a particular task (in this case, context definition) in progress in the main window frame.



### Managing User Accounts and Institutions

In order to use LITE, a user must have an account and be associated with at least one institution that is defined in LITE. Users can obtain accounts by filling out a request screen at <http://lite.bmi.utah.edu> that collects the user’s name, e-mail address, role, and reason for requesting the account. Once the request has been approved (usually within a few hours), the user can log in and indicate the institutions for which he or she would like to define resources and contexts. If the institution has not been previously defined in LITE, the user can define it, using the Object Identifier (OID, an Abstract Syntax Notation One (ASN.1) standard for identifying information objects) for the institution if known; the user then becomes the first member of the institution. If the institution already exists, the user can request membership, in which case all current members are sent an e-mail with the request and any one of them can log into LITE and approve it. Once the user is added, he or she is a full and equal member of the institution and can approve or remove other users.

Other than the user profile, LITE users add information after selecting an institution. Once the user is a member of an institution, the institution is selected at login time. If the user is a member of more than one institution, the initial institution set by default to the last institution the user had chosen; a different institution can be chosen at any time.

### *Describing Resources*

The LITE interface provides users with a step-by-step “wizard” for creating resource descriptions. The wizard asks for the resource name, URL and resource type (HL7-compliant or not). In the case of non-HL7-compliant resources, the wizard asks for the names of parameters used by the resource and the controlled terminology (if any) used by each parameter. For example, a non-HL7-compliant resource might accept the patient gender using the parameter name “sex”, rather than the HL7 parameter “patientPerson.administrativeGenderCode.c” and use the values “Male” and “Female” rather than “M” and “F”.

For each parameter, the wizard also asks the user whether the resource (HL7 or otherwise) uses a particular parameter for searching, whether the parameter is required or optional, and whether the resource’s content is limited to a certain subset of the valid parameter values. The relationship between these aspects of the parameters can be complex. For example, an on-line textbook might not use a gender parameter for searching, but its content might be limited to a particular gender (for example, a gynecology textbook). As will be shown below, providing this information to LITE in the resource description can be useful for determining the clinical contexts in which the resource should be selected. Table 1 shows examples of how resources might be defined by parameter descriptions.

Table 1: Examples of resource descriptions. Descriptions include the topics covered by the resource, and which context parameters can be used as search criteria by the resource. For example, PubMed covers all topics, genders and age groups and allows searching by the main topic of interest (the element in the EHR associated with the Infobutton link) and gender. Drug knowledge bases might cover all ages or only a limited set (such as pediatric age groups) and might or might not allow searching by gender and age. A hypothetical obstetrics and gynecology textbook is described as covering only the female gender but does not use (nor does it need to use) the gender parameter for searching. The knowledge of this limitation on domain coverage can be used for identifying appropriate contexts for choosing the resource (for example, when the patient’s gender is female).

<u>Resource</u>	<u>Main Topic</u>	<u>Gender</u>	<u>Age</u>	<u>Search Parameters</u>
PubMed	Any	Male, Female	Any	Main Topic, Gender
General Drug KB	Drugs	Male, Female	Any	Main Topic, Gender
Pediatric Drug KB	Drugs	Male, Female	Infants, Children	Main Topic, Age
OB/GYN Textbook	Female Disorders	Female	Any	None

Once a resource has been defined, the LITE user can modify any of the settings for the resource. The user can also mark a resource description as “sharable” with other users. Conversely, a user can browse the resource descriptions shared by other users and copy a description into the resource list for his or her own institution. The user can then modify the description for use in that institution, without changing the original shared copy.

### *Defining Resource Selection Criteria (Contexts)*

LITE provides a second wizard for allowing users to define the criteria by which OpenInfobutton will select a resource. The wizard first asks the user to pick one of the resources described for the institution (see above) and then steps the user through the HL7 parameters to determine the values which, if sent from the EHR, would cause OpenInfobutton to select the resource. In order to simplify the process for the user, LITE chooses default values for the parameter and then allows the user to alter them. For example, if a resource does not use the gender parameter, LITE will set the selection criteria to “U,M,F” (unknown, male, and female) so that the resource can be selected no matter what gender value is sent from the EHR. Alternatively, LITE will set the default to “F” if the resource has been defined as having content limited to the female gender (such as the gynecology text book, mentioned above). This tells OpenInfobutton to select the resource only if the patient is known to be female. The LITE user can override this, for example by changing the values to “U,F” to tell OpenInfobutton that it is acceptable to select the resources if the patient is either known to be female or if the patient’s gender is unknown.

For simplicity, LITE only asks the users about the few parameters that are most commonly used in resource selection: clinical task, patient gender, patient age and main search criteria. Once the user has defined these, LITE displays a page showing all of the user’s choices, plus all the default values selected for the other parameters (see Figure 2). At this point, the user has the option of changing any of the parameter values.

### Publishing and Testing the Context Descriptions

Once a context description is complete, the LITE user can “publish” the changes to OpenInfobutton. This process generates an XML file that contains the resource description and context parameter values that specify when the resource should be selected; this file can then be used by OpenInfobutton.

Once published, the LITE user can test the context by accessing a mock-up EHR. The mock-up allows the user to set parameter values (for example the institution and the name of a drug or diagnosis) that simulate the context to be tested. By clicking on an infobutton icon (a solid blue circle with a white “i” inside) in the mock-up, OpenInfobutton is evoked as if by a user of the real EHR at the LITE user’s institution. Figure 3 and 4 show an example of this testing process.

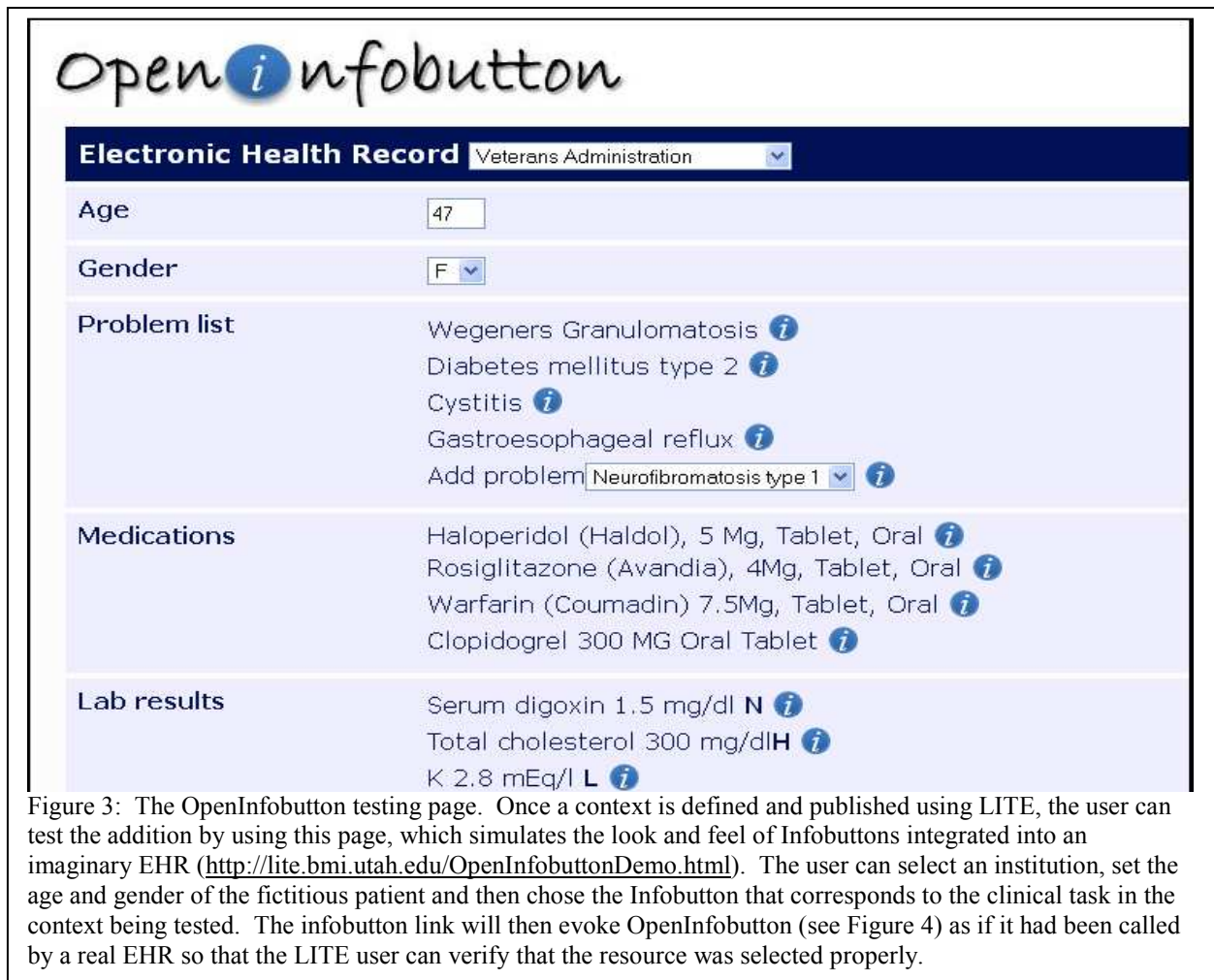


Figure 3: The OpenInfobutton testing page. Once a context is defined and published using LITE, the user can test the addition by using this page, which simulates the look and feel of Infobuttons integrated into an imaginary EHR (<http://lite.bmi.utah.edu/OpenInfobuttonDemo.html>). The user can select an institution, set the age and gender of the fictitious patient and then chose the Infobutton that corresponds to the clinical task in the context being tested. The infobutton link will then evoke OpenInfobutton (see Figure 4) as if it had been called by a real EHR so that the LITE user can verify that the resource was selected properly.

### Implementing the OpenInfobutton Link in an EHR

An EHR developer who wishes to implement a link to OpenInfobutton must be able to create a text string consisting of three parts, shown in the example below: a constant part of the URL (depicted here in plain text), constant values of some parameters (depicted in underlined italic text), and variable values for some parameters (depicted in **bold underlined text**):

```
http://dev-service.oib.utah.edu:8080/infobutton-service/infoRequest?representedOrganization.id.root=  
1.3.6.1.4.1.3768&patientPerson.administrativeGenderCode.c=F&age.v.v=47&age.v.u=a&taskContext.c.  
c=MEDOE&mainSearchCriteria.v.c=310672&mainSearchCriteria.v.cs=2.16.840.1.113883.6.88&mainSe  
archCriteria.v.dn=Haloperidol
```



Figure 4: The OpenInfobutton as it appears when called from the testing page shown in Figure 3. The tester had selected “Veterans Administration” as the institution, specified that patient is an adult female, and clicked on the infobutton icon adjacent to “Haloperidol (Haldol), 5mg, Tablet, Oral”. OpenInfobutton displays a set of links to various resources on the left side, evokes the first link (in this case, “FDA Drug Label”) and displays the result in the main window frame. Note that the Daily Med page displayed is for Haloperidol but that the age and gender information was ignored.

Once created, the developer must insert the URL into a hypertext link and then place the link in the appropriate part of the display being presented to the user. For example, if the link is being placed in an HTML page that contains a table of medication data, the link might look like this:

```
Digoxin<a href="(URL string goes here)" ><img="(infobutton.gif)"></a>
```

producing a link that looks like this to the user: Digoxin 

The above example is appropriate for Web-based EHRs. In the case where the user interface is based on some other technology, the application could evoke a Web browser on the user’s computer, passing it the URL string as a parameter, or it could pass an XML representation of the link directly to OpenInfobutton.

### Putting It All Together

The examples described above are somewhat simplified for illustration purposes, but they are real, functional examples nonetheless. In order for an EHR to actually meet the proposed “meaningful use” criteria, the following steps are needed:

- 1) Obtain a LITE account
- 2) Define the institution in LITE
- 3) Create resource descriptions in LITE, or simply copy some from the library of shared resource descriptions
- 4) Create context descriptions
- 5) Publish context descriptions
- 6) Test OpenInfobutton
- 7) Modify EHR software so that OpenInfobutton links are generated and displayed dynamically
- 8) Click on infobutton link

Steps 1-6 are performed by an institutional librarian (or someone acting in that role), Step 7 is performed by the EHR developer, and Step 8 is performed by the EHR user.

## Current Status

The LITE and OpenInfobutton applications are available for use today at the University of Utah; no local installation of software is required. However, not all functionality is currently available to regular users. Because the resource definition process is somewhat lengthy and technical, the LITE interface currently does not expose the Resource Wizard to regular users. Instead, we provide a set of definitions for popular resources that are automatically added to each institution at the time the institution is established. If a LITE user needs a resource to be added, he or she can request the addition, which will be done by the LITE team. Similarly, the Context Wizard hides many of the HL7 parameters from the user (such as subTopic, severityObservation, and encounter) to reduce the complexity of the task because of the low likelihood that these parameters will be available in the typical EHR. To date, our experience shows that typical users are overwhelmed by the number of parameters and options available in the HL7 standard. As we gain experience working with LITE users, we will modify the wizards to make them more user-friendly.

The publication step is currently functional for contexts that make use of HL7-compliant knowledge resources. For non-HL7-compliant parameters, it can handle the two parameters required by OpenInfobutton, namely mainSearchCriteria and taskContext. Other parameters will be added to the publication process as the need arises.

As with any freely available software, potential users often ask if LITE and OpenInfobutton can be made available as “open source” products. As stated above, LITE functions within the open-source Drupal software and the LITE code (in the form of Drupal data tables) is free for the asking. However, we do not currently have the resources to convert LITE to a true open source product, so it is available only on an as-is basis. Interested parties need to be aware, however, that LITE only produces tables for use by an OpenInfobutton installation that resides at the same institution. An institution cannot install LITE and expect that it will be able to maintain the institution’s knowledge base in the Utah installation of OpenInfobutton. OpenInfobutton itself continues to undergo development and is not yet available in an open source mode. However, the Veterans Health Administration (VHA) plans to release OpenInfobutton as part of the VHA’s emerging open source framework. For the time being, it is our hope that interested parties will be satisfied with using our installations for free, rather than attempting to install and maintain the same software at their own expense.

Questions also arise regarding the confidentiality of data that flow to LITE and OpenInfobutton. LITE maintains a record of user activities for debugging and system improvement; LITE users agree to this recording as a condition of using LITE. Information about changes to institution-specific data can be made available to institution members upon request. LITE is not evoked by EHRs, so no patient data are ever sent to LITE.

Similarly, OpenInfobutton maintains a log of usage, including context parameters, resources it selects, and resource links selected by users. However, the context parameters do not include any personally identifiable or otherwise private information, nor does OpenInfobutton attempt to track usage of a resource, once it is selected by a user.

## Discussion

Automated clinical decision support (CDS), in the form of EHR systems that issue patient-specific alerts and reminders, has been with us for forty years.[14] While these systems have been shown repeatedly to improve the quality of patient care, increase patient safety and decrease cost, they also have well-documented challenges. Despite the incorporation of rule engines in commercial EHR systems, the creation, testing and maintenance of the rules themselves remains a resource-intensive process. Even when rules are perfectly implemented, their messages are often ignored due to “alert fatigue” and other cognitive factors.

Infobuttons are a relatively newer technology, only recently included in some commercial systems. Unlike rule-based systems, the knowledge used by Infobuttons is created and maintained by outside entities, such as publishers and knowledge base vendors. They also have the advantage that they are a “pull” technology, as opposed to the “push” approach of traditional CDS, so that the user, not the computer, decides when the access to knowledge is appropriate and useful.

Infobutton managers, on the other hand, have been largely restricted to institutions that have strong informatics research and development programs, such as Vanderbilt, Harvard, Utah and Columbia Universities. The recent proposal by ONC to include infobuttons in their criteria for meaningful use of EHRs poses a potential obstacle to institutions without the resources to develop their own solutions. The creation of OpenInfobutton now provides them with an engine to meet this requirement in a way that goes beyond a simple, hard-wired EHR-resource link. LITE provides the knowledge engineering piece of the puzzle. Unlike the knowledge required to create decision

support rules, the knowledge collected by LITE, and used by OpenInfobutton, is much simpler and less volatile: what information need is likely to arise in a given circumstance and what is the right knowledge resource to address that need. Once the EHR-OpenInfobutton link is established, the maintenance becomes a simple matter of answering questions in LITE wizards, with no programming involved. Furthermore, as the number of LITE users increases, we expect a user community to develop provide that will not only share resource descriptions but will provide advice about when include particular resources and even perhaps “crowd-source” the improvement of the LITE user interface.

There are still some challenges to the full implementation of OpenInfobutton and LITE. While the functions in LITE are operational, we are wary of the complexity of the task and are unsure as to whether the user interface we have established will be comprehensible to the typical LITE user. We have therefore limited the tasks in LITE to those related to establishing institution groups and defining contexts for a predefined set of resources. As we gain experience with the use of this limited functionality, we expect to learn better ways for LITE to ask questions and present information. We will then modify and release additional functions in an incremental manner.

Another challenge that remains for our architecture, and infobuttons in general, is the issue of controlled terminology. In particular, the mainSearchCriteria parameters can include codes from controlled code sets. While EHRs are increasing their use of controlled terminologies, the information resources may not be able to recognize the codes or preferred names from these terminologies. Overcoming this challenge may require adding significant terminology services to OpenInfobutton so that it can create URL links that will be understandable by the resources to which they link.

Despite the challenges, the infobutton solution described in this paper exists today and is capable of providing context-specific information to EHRs users, including clinicians and patients. Several institutions are already using OpenInfobutton in their EHRs and are poised to begin using LITE to customize OpenInfobutton.

## Conclusions

The integration of OpenInfobutton into an EHR is no more complicated than integrating a single HL7-compliant resource, but offers broad flexibility for allowing the EHR user to find the right information at the right time. LITE is a key part in the architecture, because it places the control of OpenInfobutton customization in the hands of those who best know their institutional resources and their users. The free availability of both resources means that any EHR developer, whether proprietary or commercial, can now add this capability to their own systems for a fraction of the cost and effort that would be involved in building it themselves.

## Acknowledgments

Drs. Cimino and Jing have been supported in part by intramural research funds from the NIH Clinical Center and the National Library of Medicine. Dr. Del Fiol has been supported by grant number K01HS018352 from the Agency for Healthcare Research and Quality (AHRQ). The authors thank Jianhua Li, of Columbia University, for his initial work on LITE and subsequent assistance porting it to NIH, and Vojtech Huser and Andria Cimino for editorial advice.

## References

1. Federal Register Volume 77, Number 45 (Wednesday, March 7, 2012), Pages 13832-13885 (see page 13847).
2. HL7 Version 3 Standard: Context-aware Information Retrieval (“Infobutton”) Release 1; 2010.
3. Osheroff JA, Forsythe DE, Buchanan BG, Bankowitz RA, Blumenfeld BH, Miller RA. Physicians' information needs: analysis of questions posed during clinical teaching. *Ann Intern Med.* 1991 Apr 1;114(7):576-81.
4. Ely JW, Osheroff JA, Ebell MH, Bergus GR, Levy BT, Chambliss ML, Evans ER. Analysis of questions asked by family physicians regarding patient care. *West J Med.* 2000 May;172(5):315-9.
5. McKnight L, Stetson PD, Bakken S, Curran C, Cimino JJ. Perceived information needs and communication difficulties of inpatient physicians and nurses. *Proc AMIA Symp.* 2001:453-7.
6. Rosenbloom ST, Geissbuhler AJ, Dupont WD, Giuse DA, Talbert DA, Tierney WM, Plummer WD, Stead WW, Miller RA. Effect of CPOE user interface design on user-initiated access to educational and patient information during clinical care. *J Am Med Inform Assoc.* 2005 Jul-Aug;12(4):458-73. Epub 2005 Mar 31.
7. Del Fiol G, Rocha RA, Clayton PD. Infobuttons at Intermountain Healthcare: utilization and infrastructure. *AMIA Annu Symp Proc.* 2006;:180-4.
8. Maviglia SM, Yoon CS, Bates DW, Kuperman G. KnowledgeLink: impact of context-sensitive information retrieval on clinicians' information needs. *J Am Med Inform Assoc.* 2006 Jan-Feb;13(1):67-73.



9. Cimino JJ, Li J, Bakken S, Patel VL. Theoretical, empirical and practical approaches to resolving the unmet information needs of clinical information system users. Proc AMIA Symp. 2002:170-4.
10. <http://www.openinfobutton.org>
11. <http://lite.bmi.utah.edu>
12. Del Fiol G, Huser V, Strasberg HR, Maviglia SM, Curtis C, Cimino JJ. Implementations of the HL7 Context-Aware Knowledge Retrieval ("Infobutton") Standard: Challenges, strengths, limitations, and uptake. J Biomed Informatics; 2012 Jan 2. [Epub ahead of print]
13. <http://drupal.org>
14. Warner HR, Olmsted CM, Rutherford BD. HELP--a program for medical decision-making. Comput Biomed Res. 1972 Feb;5(1):65-74.