IAIMS and UMLS at Columbia–Presbyterian Medical Center

JAMES J. CIMINO, MD, SOUMITRA SENGUPTA, PhD

The authors use an example to illustrate combining Integrated Academic Information Management System (IAIMS) components (applications) into an integral whole, to facilitate using the components simultaneously or in sequence. They examine a model for classifying IAIMS systems, proposing ways in which the Unified Medical Language System (UMLS) can be exploited in them. *Key words*: Integrated Academic Information Management System; Unified Medical Language System. **(Med Decis Making 1991;11(suppl):S89–S93)**

The Columbia–Presbyterian Medical Center (CPMC) is one of several recipients of an Integrated Academic Information Management System (IAIMS) grant from the National Library of Medicine (NLM). Each of the IAIMS sites has its own particular approach to the kinds of integration that are being developed. Examples of IAIMS integration efforts include physical network connections, software communications protocols, and coordination of diverse personnel.[1–9] This paper deals with an aspect of IAIMS integration that is common to all of the IAIMS sites: combining system components (applications) into an integral whole, to facilitate using the components simultaneously or in sequence. Many scenarios can be imagined in which application integration can be applied. We use a single example throughout this paper to illustrate some of the issues of such integration. In this scenario, a physician using one application desires access to a second application to obtain additional information in support of a clinical decision. For example, a physician obtaining a laboratory result from a clinical information system might wish to perform a bibliographic search to help answer questions about the significance of the result or about the further diagnostic interventions that are required. The need for such an arrangement is commonly described[10] and examples of solutions are often given.[11,12] In this scenario, there is some information available in the first application (the source) that is required by the second application (the target) in order for it to properly respond to the physician's information need. By definition, any operation that facilitates the transfer of that information constitutes application integration. The issue we examine here is: what are the different arrangements by which such integration can be carried out?

One tool that has recently become available for realizing application integration is the Unified Medical Language System (UMLS). The UMLS is a project under development by the NLM to develop methods and resources that will facilitate access to diverse information sources, despite their use of disparate controlled vocabularies and protocols for information exchange.[13] This access to and exchange of information is precisely the kind that is envisioned in IAIMS.[11,14] The UMLS is presently considered experimental; therefore, the roles for which the UMLS is suited in IAIMS remain to be determined. The purpose of this paper is to describe some of the ways that are being explored at CPMC for applying the UMLS in application integration tasks.

In order to clarify these roles, we examine different integration arrangements ("classes") that are typically

Received from the Center for Medical Informatics, Columbia University, New York, New York.

Address correspondence and reprint requests to Dr. Cimino: Center for Medical Informatics, Atchley Pavilion, Room 1310, Columbia–Presbyterian Medical Center, New York, NY 10032.

found in IAIMS and non-IAIMS settings. We then propose ways in which the UMLS can be exploited in each arrangement and proved relevant examples from the CPMC IAIMS effort.

Class 0—No Application Integration

One of the motivations for the support of IAIMS activities is that, historically, there is often no scheme for integrating diverse applications in the complex environment of the academic medical center. We refer to this state as Class 0 integration. In this setting, terminals providing access to the various applications are distant from each other. For information to be transferred, the user must obtain information from one application, record it in some way (memorization, transcription, or printing), travel to the terminal for the next application in the chain, log onto it, enter the previously obtained information, request the information needed for the next step in the information-gathering process, record that information, and proceed to the next checkpoint in the journey. The disadvantages of attempting to integrate in this way are obvious: imagine, for example, the difficulty that results when a transcription error occurs at the first terminal.

Class 0 is common. As Class 0 is the antithesis of IAIMS, we will not examine it further and do not attempt to identify a corresponding role for the UMLS.

Class 1—Using Wires to Integrate Applications

Access to disparate systems can be achieved by placing terminals for those systems side by side. The addition to a hospital information system of something as simple as wires to allow the installation of additional terminals can have a profound impact on the way information flows between applications. The user can be signed on to the source and target applications simultaneously and can simply copy the information from the former to the latter. This obviates the need for generating a persistent copy of the information (through memorization or interim transcription) and is especially advantageous because transcription errors become much less annoying than they might be in Class 0. Disadvantages of this approach include the cost of redundant equipment and difficulties related to the limitations of space in the user's work place. As in Class 0, the information transfer is carried out solely by the user.

The evolution to Class 1 constitutes an acknowledgement on the part of medical center information professionals that medical decision making often requires information from multiple sources. Like Class 0, it is an arrangement that commonly develops; however, IAIMS sites have invariably chosen more sophisticated solutions. Discussion of a role for the UMLS in IAIMS Class 1 is, therefore, irrelevant.

Class 2—Using Hardware to Integrate Applications

A more technically difficult approach to integration involves providing access to multiple applications, one at a time, through the same physical terminal. Various approaches include: 1) porting the applications to a single host machine; 2) connecting a single host machine to other machines running various applications and providing access to those machines through terminal emulation; 3) allowing (with the use of a selector switch) the terminal to connect to multiple machines; and 4) emulating a terminal using a personal computer and connecting it (via separate wires or a network) to multiple host machines. Each of these approaches help address the cost and space issues. However, from an information-flow perspective, Class 2 removes some of the advantages of Class 1. Since only one application can be used at a time, information must be recorded (memorized or transcribed) before it can be transferred to a target application.

Class 2 is the integration scheme that is common to the developmental paths of all IAIMS sites. For example, most care providers at CPMC use a networked personal computer with a terminal emulation program. Together, the hardware and software facilitate access to multiple applications. How can the UMLS facilitate access to diverse information sources in Class 2? UMLS knowledge sources are not provided in a hard-copy form, but if they were, they would probably be unwieldy (especially the Metathesaurus, with its 208,559 names[15]). Computer tools for browsing the UMLS Metathesaurus[16] and the Semantic Network[17] are available and could be incorporated into a Class 2 setting. However, to use these tools for information transfer, the user would have to record the relevant information, leave the source application, start the UMLS application, enter the information, obtain new information, exit the UMLS application, start the target application, and enter the new information. Obviously, this arrangement introduces additional complexity and effort to the task of application integration.

A potential benefit from this effort can be realized if the UMLS application helps the user select an appropriate target application or converts the information into a form that is useful to the target application. The UMLS application might, however, request additional information that requires leaving that application and returning to the source application. Clearly, there are both potential advantages and disadvantages to using the UMLS in a Class 2 setting. IAIMS projects that are based on the Class 2 arrangement might choose to explore whether the UMLS can contribute to this kind of application integration. The CPMC IAIMS proj-

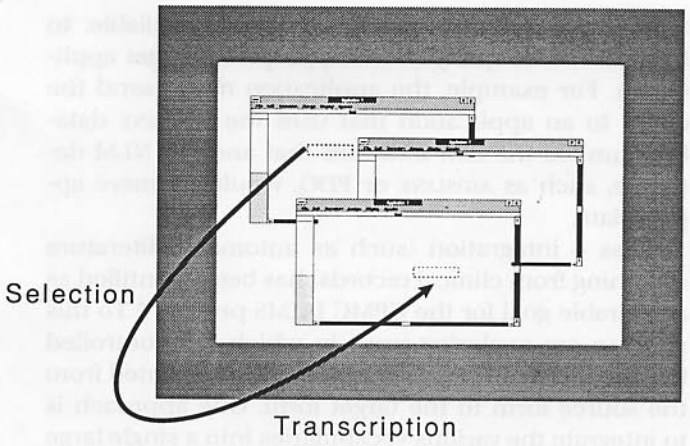

FIGURE 1. Class 3 integration. Information transfer requires the user to select the desired information from the source application and transfer it to the target application. Manual copying may be required, or the operating system environment might provide a "cut-and-paste" feature for automating the transfer.

ect is currently moving away from the Class 2 integration and is not pursuing answers to this question.

Class 3—Using System Software to Integrate Applications

The next integration class, Class 3, reintroduces the notion of concurrent use of applications that is possible in Class 1. However, in this case, multiple simultaneous access is possible due to an operating system environment that permits the access via a single terminal. The particular methods by which an operating system accomplishes this depend on where the applications are actually running. From the user's standpoint, the single terminal allows rapid switching between applications without requiring the sign-off-and-sign-on steps needed in Class 1. The terminal may allow switching by using a "hot key" to change terminal sessions, or it might display multiple applications simultaneously, using either a split screen or a windows interface. Figure 1 shows a sample of such a display. The arrows show how information is transferred between applications: the user determines which information from the source application is needed and copies it to the target application. As with information transfers in the previous classes, if translation or reformatting of the information is required, the user must perform these operations mentally prior to the transcription into the target application.

Many windowing interfaces provide a tool, often called a clipboard, by which the user selects the desired information by use of a pointing device. Using a "cut-and-paste" command, the selected information is automatically transferred to the target application. Such a transfer is easier than manual transcription; however, the information is transferred *verbatim*. Since the cut-and-paste operation is application-independent, it has no knowledge of how the information should be altered prior to transfer. It is here that a UMLS tool, if properly designed, could facilitate information transfer. If the UMLS tool is capable of translating and reformatting information from its source form to its target form, and if the user is accessing the tool along with the source and target applications, then the user can select the desired information, paste it into the UMLS application (which converts it into the target form) and then paste the result of the conversion into the target application. The UMLS is being designed for just such a scenario.[11]

Although most CPMC users are operating in a Class 2 environment today, a limited number have access to Class 3 integration (in varying degrees). In some locations, a simple hardware upgrade permits the software running on the personal computer to exercise additional features that allow multiple access to a limited list of applications. For example, the user of these machines can review clinical data, log onto a bibliographic search system, perform a literature search, and return to the results review application. This is generally accomplished through hot-key application switching, although primitive screen-splitting is available. In other cases, the operating system can provide an environment (MS-DOS Windows 3.0, Apple Macintosh, OS/2 Presentation Manager, or UNIX X-Windows) that extends the list of applications that can be accessed simultaneously and provides true windowing capability with a cut-and-paste feature. At present, this ability is limited to a relatively small number of system developers.

As previously noted, the software platform currently in use at CPMC generally provides a Class 2 environment. While a path exists to transform the environment to Class 3, the present IAIMS plan is to proceed in a direction that will allow other types of integration (see below). The UMLS does not include any translation tools that could be easily incorporated into the present clinical computing environment. Until such time as tools become available, no specific Class 3 role for the UMLS is planned at CPMC.

Class 4—Using Applications to Integrate with Each Other

In each of the integration classes described above, the task of information transfer is carried out by a human user. The classes differ with respect to the physical access to the various applications, with information transfer made easier in some. However, the user is still responsible for the basic tasks of selecting the necessary information from a source, determining the appropriate target application, translating and reformatting the information, and ferrying it (either manually or automatically) to the target application. In Class 4, the applications perform these tasks for the

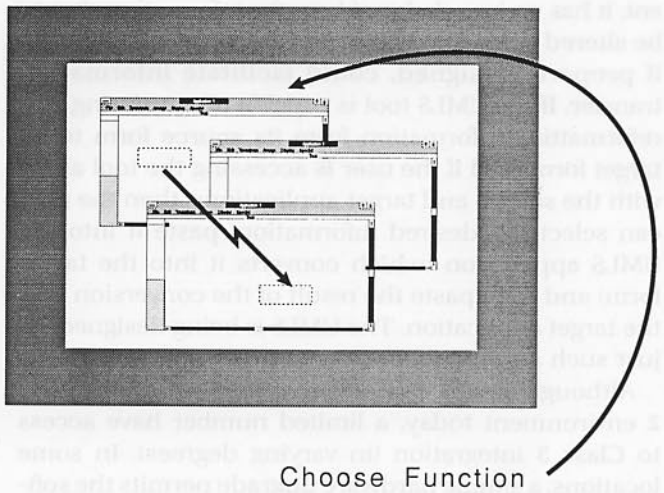

FIGURE 2. Class 4 integration. Information transfer requires the user to select a desired application function. The application then determines which information should be selected and where it should be transferred.

user because they are given this ability by design. Figure 2 shows an example of Class 4 integration. The user selects an application function that requires information be transferred to a target application. The application then selects and transfers the information, using interapplication techniques such as Dynamic Data Exchange (DDE). For example, if the application in the top window is a laboratory results review application, and the user wishes to know something about the specificity of the test, the chosen function might carry out the sequence: 1) select the test name, 2) "AND" the test name with "specificity," 3) transfer the "test AND specificity" expression to the search engine (represented by the lower window), and 4) tell the search engine to perform a literature search.

As in Class 3, the UMLS can provide a valuable service by converting the information appropriately prior to transfer. Because the transfer is being carried out by the application, which knows where the information came from and where it is going, the UMLS can be applied in precise ways. For example, since the application knows that the intended target is a bibliographic search engine that uses MeSH terms, it can add a step 1a) to the above sequence: use the UMLS Metathesaurus to translate the test name to a MeSH term. Furthermore, since the application knows that it is dealing with a laboratory test, it can make the translation step even more specific: use the UMLS Metathesaurus to translate the test name to a MeSH term that has the UMLS Semantic Type "Diagnostic Procedure" or "Laboratory Procedure." Thus, if the laboratory system choose to call a particular test "Gamma Globulin," the translation will not be to the MeSH chemical term "Gamma Globulins" but to the MeSH laboratory procedure term "Globulins." Finally, the application might make use of the Information Sources Map (ISM),[13] when it becomes available, to help direct the query to the appropriate target application. For example, the application might send the query to an application that uses the MEDLINE database, unless the ISM indicates that another NLM database, such as AIDSLINE or PDQ, would be more appropriate.

Class 4 integration (such as automated literature searching from clinical records) has been identified as a desirable goal for the CPMC IAIMS project.[18] To this end, we are exploring ways in which the controlled terms used in our applications can be translated from the source form to the target form. One approach is to integrate the various vocabularies into a single large dictionary, the Medical Entities Dictionary.[19] A clear role for the UMLS has been identified in the MED: it provides the basic framework in which the terms from the CPMC vocabularies are being integrated with each other.[20]

Class 5—Integration with Autonomous Transfer

In Class 4 integration, the applications carry out information transfers in response to a user request. However, if they can perform such functions on demand, they can also be designed to carry out such transfers for their own purposes. We refer to this final stage of integration as Class 5. This type of integration is being explored as part of the CPMC on-line decision support system, currently under development.[21] This system is evoked by occurrence of clinical events and collects information from appropriate sources as needed to process its logic. The roles for the UMLS that are described for Class 4 apply equally well to Class 5: selection of an appropriate target application and translation of terms, in context, to the vocabulary of the target. In addition, the CPMC decision support system has adopted a currently evolving (ASTM) international standard for medical logic modules known as the Arden syntax.[22] Among the topics to be discussed by the standards committee is the need for a controlled vocabulary in the syntax and the appropriateness of the UMLS to serve that function. If this terminology is indeed adopted, it will signify an additional Class 4 role for the UMLS in the CPMC IAIMS project.

Discussion

Integration in IAIMS takes on many meanings; one that is common to all sites involves the coordination of various applications to bring them to bear on problems in a concerted way. The model presented here is not meant to be comprehensive, nor meant to rank or criticize the choices made regarding application

integration in specific IAIMS implementations. It does, however, point out some of the variations in which information can be transferred between applications as part of their integration. In particular, this model highlights the functional differences between bringing applications together for the user to perform information transfers and developing applications that take advantage of the integration environment to carry out the more tedious parts of information transfer. One such tedious process is the transformation of information from its source form to its target form. The UMLS has been designed for just such a process. The integration environments that provide an opportunity to automate this process are the ones that can take greatest advantage of the resources of the UMLS.

The IAIMS project at CPMC is evolving with respect to the way in which applications are integrated. The creation of a Medical Entities Dictionary which seeks to unify the UMLS with local CPMC vocabularies is in anticipation of the development of integration Classes 4 and 5. Because the UMLS is providing much of the infrastructure of this dictionary, its primary role at CPMC will be in the task of integrating the local CPMC vocabularies with each other and, through the UMLS, with national vocabularies such as MeSH, ICD9 and SNOMED. In addition, we look forward to possible roles for the UMLS in clinical decision support through the Arden syntax and through automated application selection using the Information Sources Map.

Supported in part by a National Library of Medicine Integrated Academic Information Management Systems (IAIMS) grant. In addition, Dr. Cimino receives support from the International Business Machines Corporation.

References

1. Ball MJ, Douglas JV. Tying it all together: the Integrated Academic Information Management System being implemented at Maryland. Crit Rev Med Inform. 1988;1:311–22.
2. Barnett GO, Greenes RA, Zielstorff RD. IAIMS development at Harvard Medical School. Bull Med Libr Assoc. 1988;76:226–30.
3. Lorenzi NM, Marks EB. University of Cincinnati Medical Center: integrating information. Bull Med Libr Assoc. 1988;76:231–6.
4. Van Hine P, Pearse WH. The IAIMS project of the American College of Obstetricians and Gynecologists: using information technology to improve the health care of women. Bull Med Libr Assoc. 1988;76:237–41.
5. Stead WW. Information management through integration of distributed resources. Bull Med Libr Assoc. 1988;76:242–7.
6. Lucier RE, Matheson NW, Butter KA, Reynolds RE. The knowledge workstation: an electronic environment for knowledge management. Bull Med Libr Assoc. 1988;76:248–55.
7. Gorry GA, Burger AM, Chaney RJ, Long KB, Tausk CM. A virtual notebook for biomedical work groups. Bull Med Libr Assoc. 1988;76:256–67.
8. Broering NC, Gault HR, Epstein H. BIOSYNTHESIS: bridging the information gap. Bull Med Libr Assoc. 1989;77:19–25.
9. Paton JA, Clyman JI, Lynch P, Miller PL, Sittig DF, Berson BZ. Strategic planning for IAIMS: prototyping as a catalyst for change. In: Miller RA, ed. Proc Fourteenth Annual Symposium on Computer Applications in Medical Care. Washington, DC, November 1990;709–13.
10. Guyatt GH. Evidence-based medicine. ACP Journal Club. Ann Intern Med. 1991;114(suppl.2):A-16.
11. Lindberg DAB, Humphreys BL. Computer systems that understand medical meaning. In: Scherrer JR, Côté RA, Mandil SH, eds. Computerized natural medical language processing for knowledge engineering. International Medical Informatics Association. Amsterdam: Elsevier (North-Holland), 1989;5–17.
12. Fagan LM, Perreault LE. The future of computer applications in health care. In: Shortliffe EH, Perreault LE, eds. Medical informatics. Reading, MA: Addison-Wesley, 1990;620–44.
13. Lindberg DAB, Humphreys BL. The UMLS Knowledge Sources: tools for building better user interfaces. In: Miller RA, ed. Proc Fourteenth Annual Symposium on Computer Applications in Medical Care, Washington, DC, November 1990;121–5.
14. Lindberg DAB. The IAIMS opportunity: the NLM view. Bull Med Libr Assoc. 1988;76:224–5.
15. Tuttle M, Sherertz D, Olson N, et al. Using Meta-1—the 1st version of the UMLS Metathesaurus. In: Miller RA, ed. Proc Fourteenth Annual Symposium on Computer Applications in Medical Care, Washington, DC, November 1990;131–5.
16. Sherertz DD, Tuttle MS, Cole W, Erlbaum M, Olson N, Nelson SJ. A HyperCard implementation of Meta-1: the first version of the UMLS Metathesaurus. In: Kingsland LW, ed. Proc Thirteenth Annual Symposium on Computer Applications in Medical Care, Washington, DC, November 1989;1017–8.
17. National Library of Medicine. UMLS knowledge sources—experimental edition documentation. Bethesda, MD, September 1990.
18. Cimino JJ, Sideli RV. Using the UMLS to bring the library to the bedside. Med Decis Making 1991;11:S116–S119.
19. Cimino JJ, Hripcsak G, Johnson SB, Clayton PD. Designing an introspective, controlled medical vocabulary. In: Kingsland LW, ed. Proc Thirteenth Annual Symposium on Computer Applications in Medical Care, Washington, DC, November 1989;513–8.
20. Cimino JJ, Hripcsak G, Johnson SB, Friedman C, Fink DJ, Clayton PD. UMLS as knowledge base—a rule-based expert system approach to controlled medical vocabulary management. In: Miller RA, ed. Proc Fourteenth Annual Symposium on Computer Applications in Medical Care, Washington, DC, November 1990;175–80.
21. Hripcsak G, Clayton PD, Cimino JJ, Johnson SB, Friedman C. Medical decision support at Columbia–Presbyterian Medical Center. In: Timmers T, ed. Proc International Medical Informatics Association Working Conference on Software Engineering in Medical Informatics, Amsterdam, October 1990 (in press).
22. Hripcsak G, Clayton PD, Pryor TA, Haug P, Wigertz OB, Van der Lei J. The Arden syntax for medical logic modules. In: Miller RA, ed. Proc Fourteenth Annual Symposium on Computer Applications in Medical Care, Washington, DC, November 1990;200–4.