

Application of Information Technology ■

A Voice-enabled, Structured Medical Reporting System

DAVID F. ROSENTHAL, PHD, JOANNE M. BOS, MS, RACHAEL A. SOKOLOWSKI, JENNIFER B. MAYO, KERRY A. QUIGLEY, ROGER A. POWELL, MA, MARY-MARSHALL TEEL

Abstract Kurzweil Applied Intelligence received a research grant from the National Institute of Standards and Technology (NIST) Advanced Technology Program to develop a prototype voice-enabled, structured medical reporting system. In typical usage, the physician dictates to the system, which then uses automatic speech recognition and medical knowledge bases to produce a structured report. This report can then be formatted and viewed on a computer screen, stored in databases of patient information, transmitted to other systems, used to support outcome studies, or viewed on a Web browser. The output reports are structured according to two standard, platform-independent formats: SGML and CORBA. These formats represent the data in a way that can be read by both computers and humans, and efficiently communicated to a wide range of databases and communications protocols.

■ *J Am Med Inform Assoc.* 1997;4:436–441.

Producing and managing clinical reports represents a significant cost to the health care enterprise. It is, first of all, a serious and thankless burden on physicians to produce the reports. In a 1989 survey of internists, for example, 63% of the respondents agreed that patient records are becoming increasingly burdensome without improving the quality of patient care.¹ Second, the records, once produced, are much less useful than they could be.² One might hope that clinical reports stored on computers would not only more efficiently fill their traditional role of documenting the health issues of a particular patient, but that they could also form the basis of a database of information that would support institutional quality control, epidemiological studies, and other kinds of cross-patient

queries. Unfortunately this is not the case. Medical reports are typically stored as unstructured ASCII files (or worse, as paper documents). For practical purposes, one cannot write a program that will query the document as to what the patient's temperature was at the time of the visit, or whether the patient had a sore throat. The information is only readable by humans.

The technical solutions to these problems which have been implemented in recent years generally reflect a trade-off between ease of use and usefulness of the resulting document. Pen-based structured systems such as Physix Pocket Doc produce highly structured (and thus useful) output, but are more difficult and time-consuming to use than transcription-based systems. Automatic transcription systems such as IBM Medspeak are easier to use, but provide correspondingly less structured output. Kurzweil AI markets a system called Kurzweil Clinical Reporter, which employs voice recognition and can serve as the front end to a hospital information database. However, Clinical Reporter's output is not structured according to industry standards (no such standards currently exist) and connecting to hospital information databases, while possible, is cumbersome.

We have received a grant from the National Institute for Standards and Technology Advanced Technology Program (NIST ATP) to develop a new system that addresses problems of usability and usefulness in a comprehensive manner. The system, called OSSIM

Affiliation of the authors: Kurzweil Applied Intelligence, Inc. Waltham, MA.

Presented in part at the AMIA Spring Conference, San Jose, CA, May 1997.

This research is supported by a grant from the National Institute of Standards and Technology Advanced Technology Program.

Correspondence and reprints: David Rosenthal, PhD, Kurzweil Applied Intelligence, Inc., 411 Waverley Oaks Road, Waltham, MA 02154.

E-mail: daver@kurzweil.com

Received for publication: 5/15/97; accepted for publication: 7/16/97.

(*Open Systems Structured Information Manager*) is a voice-enabled, structured reporting application. The physician speaks to a microphone connected to laptop or desktop computer, which, in response to the physician's utterances, produces a structured clinical report. The reports are structured according to two widely adopted, platform-independent structuring schemes: SGML and CORBA, both of which are discussed further below. In parallel with the development of the system, Kurzweil is participating in the development of industry standard CORBA and SGML-based representations for clinical data. The representations allow specific clinical information in the document to be accessed easily and automatically; to use the example above, one can determine the patient's temperature at the time of the last visit, or indeed any other information contained in the patient's record which might be of interest. The report can easily interface to a variety of hospital information systems or communications protocols such as HL7. The report can also be viewed in a number of different formats, displayed on a Web browser, or edited on a word processor (Fig. 1).

Although our experience building Kurzweil Clinical Reporter has been invaluable for understanding the issues surrounding the construction of voice-enabled, structured reporting systems, OSSIM is not an extension of Clinical Reporter; rather, it embodies an entirely new architecture. An important result of this re-architecting effort is much greater modularity of design in the new system. There are independent modules, for example, corresponding to voice recognition, the medical knowledge base, and the CORBA-based connection to hospital information databases. If voice recognition is not present, the system is still entirely functional as a mouse and keyboard-based system. The medical knowledge base can be replaced with other knowledge bases, even non-medical ones,

allowing the system to be used for structured reporting in other domains. This design reflects both an engineering interest in maintainability and modular design, as well as Kurzweil's long term strategy for component-based software.

Standards for Structuring Reports

It is increasingly clear that digital information is useful only to the extent that it is represented in a standard format which can be accessed by a range of systems on possibly different platforms. Accordingly, we selected two non-proprietary, platform-independent report structuring methods: SGML and CORBA.

What is SGML?

SGML (*Standard Generalized Markup Language*) is an ISO standard means of defining *markup languages* which are ways of making the structure of text documents computer-readable.³ This is accomplished through the use of *tags*, which delineate structurally interesting parts of the document. A tag might delineate a few words, as in:

```
<NAME>Julius Rosenthal</NAME>
```

Here <NAME> and </NAME> are start and end tags, respectively, which delineate a name. Tags may also be used to delineate paragraphs, sections, chapters, or any of the structural elements of documents.

The set of tags, and the rules for their usage, which are appropriate for defining a document is defined in a *document type definition*, or DTD. A DTD thus defines a particular *markup language*. Typical examples of DTDs are those which define the structure of a book chapter or a memo. The most well known and widely used SGML-defined markup language is HTML (hy-

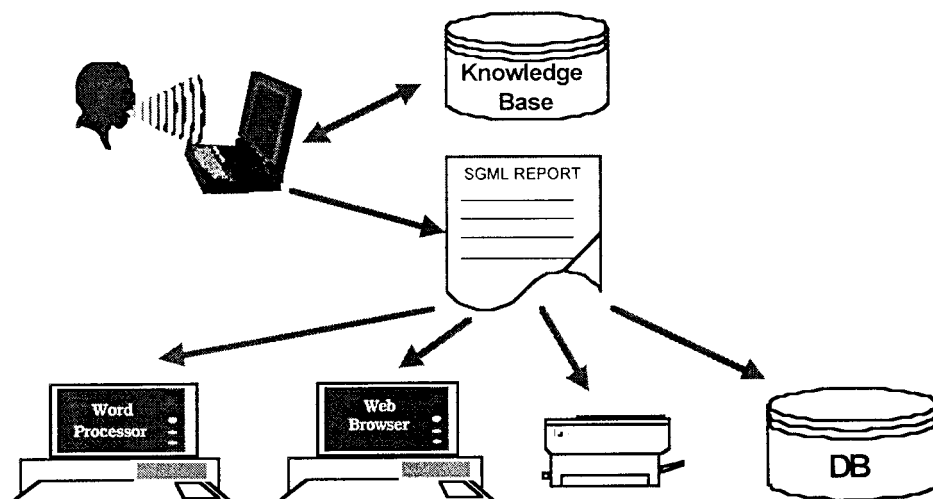


Figure 1 Overview of OSSIM report generation and data flow.

```

<SYMPTOM>There has been
  <SEVERITY>moderate</SEVERITY>
  <ANATLOC>ear</ANATLOC>
  <SYMPTOMNAME>pain</SYMPTOMNAME>
  <DURATION>for 3 days</DURATION>.
  <ONSET>The onset was
    <ONSET_TYPE>gradual</ONSET_TYPE>.
  </ONSET>
</SYMPTOM>

```

Figure 2 Content Tagging Example

pertext markup language), which defines the documents that constitute the World Wide Web. Many industries have agreed on standard DTDs for document types which are of interest to them. Examples of industries with standard DTDs are the semiconductor, pharmaceutical, and defense industries. Kurzweil AI has been working with other health care companies and organizations to produce standard DTDs for primary care visit notes, prescriptions, demographics, and other healthcare-related documents. The framework for these interorganizational efforts is the HL7 SGML Initiative.⁴ As the name implies, the HL7 SGML Initiative is a Special Interest Group in the HL7 organization; it is expected that the group will eventually become a Technical Committee. In a future version of HL7, there will be defined, standard relationships between HL7 fields and SGML elements of a standard DTD. Currently, most clinical information transmitted on the HL7 protocol is either unstructured or structured according to special, nonstandard agreements between the sender and receiver.

Figure 2 is an example of tagged output from a primary care visit note. The SGML tags are specified in the knowledge base of clinical information (described in further detail later in this paper).

What is CORBA?

The second standard form in which we present structured clinical reports is as CORBA objects. CORBA (*Common Object Request Broker Architecture*) is an industry standard for *distributed object computing*.⁵ Object-oriented computing is a style of software engineering in which the fundamental constructs are *objects* and *methods*. The objects and methods are usually thought of as corresponding, respectively, to the “nouns” and “verbs” of an aspect of the world that is modeled by the software. *Distributed object computing* refers to object-oriented systems in which the objects in question can reside on different platforms, and methods can be invoked across network connections.

In our case, the objects in question are reports, and the methods are accessors for structural elements of the report. Typical report-object methods might access

the patient’s name, the subjective section, the diagnosis, and so on. The elements which are available through the CORBA interface are the same as those which are tagged in the SGML representation. An appropriate use of CORBA is as a standard way of communicating with patient record databases. We have implemented a connection to a sample Oracle database of patient information running at a remote site. We expect to have a CORBA–HL7 connection in the near future.

Like SGML, CORBA is not itself a structuring language, but rather a standard means of defining such a language. In the case of CORBA, a specific language, or object interface definition, is specified in an IDL (*Interface Definition Language*). IDLs play the same role in the context of CORBA that DTDs play in the context of SGML.

If we consider the ultimate goal to be the ability to express our results in a language that is easily understood by any patient record database or communication protocol, then the fact of using CORBA (or SGML) gets us halfway there. The second half is accomplished when the IDL (or DTD) is itself standardized. As with the case of SGML, Kurzweil AI has been participating in an effort to produce such a standard. We are active participants in CORBAMED, an organization whose aim is to produce a standard IDL for clinical data.⁶

SGML and CORBA in OSSIM

As the report is dictated, OSSIM maintains an internal structure called the Primary Medical Document, or PMD. The PMD can be thought of as the runtime representation of the SGML document that will constitute the report at the end of the dictation session; the SGML document is essentially a “dump” of the PMD. The CORBA object, in turn, is generated from the SGML document. We ensure consistency between the CORBA objects and the SGML documents by declaring certain elements to be fixed for a given revision. Those elements are basically the elements that correspond to fields in the database with which CORBA is communicating.

Knowledge Base

The physician’s interaction with the system is guided by an object-oriented knowledge base (KB) of clinical information.

Object-oriented Representation

The object-oriented approach enables us to structure information in a content-oriented manner which im-

proves our ability to tag medical information. It provides inheritance, which allows uniform organization of data, consistency within and across knowledge base domains, and easier maintenance. For example, symptoms have attributes in common such as severity, duration, and onset, which are inherited by instances (e.g., earache, nausea, fever) of the symptom class. Another example is anatomy, which can be divided into subclasses (e.g. limb) with attributes that are inherited by instances (e.g., arm, leg) of those subclasses. Also, since objects are self-contained, the more common objects, such as anatomy, can be used by knowledge bases in many domains.

The objects in our system relate directly to SGML syntax. The SGML structure is contained in and constrained by the object structure so that data is tagged appropriately. There are two main objects: elements and concepts. Element objects represent SGML elements; they provide a content model and define the tag name. The content model defines the element structure, the set of elements that may be nested within the element. For example, the symptom element may have a content model of severity, duration and onset, each of which are elements and have their own tags. This hierarchical structure is reflected in the DTD. Concept objects are specific instances of element objects, i.e., they are derived from elements in that they are constrained by the element's content model and inherit its SGML tag. They also may contain SGML attributes, and they store report text as well as context for what can be entered or spoken by the user. The level of detail of data tagged and the SGML tags themselves are specified when the knowledge base is constructed. It is possible to derive a DTD from the knowledge base structure, or conversely, construct a knowledge base to conform to an industry standard DTD.

Object-oriented technology allows for relationships between objects. This improves our ability to reuse data and prevent inconsistency. For example, if the physician indicates that the chief complaint is earache, the system recognizes that the related anatomy is ear, and asks the user which ear. Information related to the ear is stored in the ear concept rather than in each knowledge base concept that may refer to ear. Rules also are an important component of the knowledge base. The system provides the ability to create and execute rules at various points during report creation. Rules may be used to validate input, or to verify that a report meets HCFA or practice guidelines. The combination of relationships and rules also allows the system to prompt for appropriate sets of symptoms, physicals, findings, etc., based on information such as chief complaint or patient sex.

Objects are a natural medium for representing for real world data such as medical information and an object-oriented database allows for the storage and retrieval of such objects. An object-oriented database provides efficiencies in that it enables the application to easily navigate through complex relationships without having to create resource-intensive joins which would be required using a relational database. Also, there is no programmatic mapping required to go between application objects and database structure.

What is Stored in the Knowledge Base?

Kurzweil AI has created structured knowledge bases to provide clinical reporting capability for primary care, as well as five other medical specialties. Using primary care as an example, the object-oriented knowledge base stores the following:

- Overall report structure in standard SOAP (Subjective-Objective-Assessment-Plan) format, including appropriate SGML tags
- Medical concepts, such as symptoms, physicals, chief complaints, and diagnoses, again including appropriate SGML tags
- The words or phrases (medical and general English terms) that can be spoken and the context in which the term can be said. Context includes: when a particular term may be spoken, where that term may exist in the report structure, what to prompt or to question the user for, what to do when something is said. We are using data from the National Library of Medicine's Unified Medical Language System (UMLS) as the starting point for our terms.
- Relationships between medical concepts, which can be used to create sets of typical symptoms, physicals, and plans based on the chief complaint or other relevant medical information
- Rules that operate at the individual concept level, e.g., for default values and validation; at the section level, e.g. to provide context specific data filtering based on patient age or sex; and at the report level, e.g., to verify government, insurance, and individual practice guidelines
- Report text output that may be generated from a single voice utterance: i.e., a quick way for the user to incorporate reusable text. For example, saying 'normal ears' will generate report text that describes the findings associated with a normal ear exam.
- Lists of medications, procedures, and diagnoses
- Grammars that encode the structure of notions such as duration, frequency, size, severity, etc., and provide possible responses at each step

Physician–Knowledge Base Interaction

As the physician responds to the prompts that the system presents, the knowledge base uses the data that has been collected along with the primary care concepts, rules and relationships to guide the user in completing the report. For example, suppose the physician says “earache” when prompted for the chief complaint. This will cause a number of things to happen:

- The system will ask about the onset, duration, severity, and location of the earache.
- The physician will be presented with a list of symptoms typically associated with earache (fever, sore throat, cough, etc.). For each of these, there will be additional questions (severity and description of sore throat, degree of fever).
- The system will note, for later reference, that the “ear” anatomy is involved in the chief complaint; this may be used later to filter or modify other prompts.

Customization

Customization is an important requirement for a medical reporting system. The knowledge base may be customized to incorporate individual and institution guidelines and preferences, as well as government and insurance guidelines. Customizations may range from simply changing the wording or the extent of the report text that is generated by an utterance, to changing the relationships and rules that affect report structure and default values. The SGML tags in the knowledge base may be customized to be consistent with site specific or industry standard reporting requirements or existing lexicons as desired.

Current Status and Evaluation

We are currently about 18 months into the project, which is scheduled for 2 years. To date we have developed a detailed object-oriented design and analysis of the system, including extensive *use cases*: i.e., formal scenarios describing ways in which the system is used, and how the subsystems interact. We have implemented and tested several successively more elaborate prototypes, which we have demonstrated at conferences and trade shows. The current version encompasses the major elements of the final system; a user can speak to the system, use it to generate reports, and access information in a sample patient record database. The knowledge base, however, is still incomplete, and handles only a few patient complaints. We are currently installing an alpha version

of the system at a local hospital, where, during the next few months, the system will be used to document actual patient encounters. We expect that this trial will be useful for improving the usability and robustness of the system.

Although OSSIM has not yet been formally evaluated, there have been several studies of the commercial voice-enabled clinical reporting systems that preceded it. These studies are relevant to understanding usability issues for OSSIM, since the existing commercial systems represent a baseline for OSSIM’s performance and capabilities. The most dramatic improvement that structured, voice-enabled systems represent over transcription-based systems is in the area of throughput. In one study, the average time from dictation to finished report was reduced from a matter of days to less than one minute.⁷ Gains were also realized in transcription error rates; the same study found that the voice recognition based system had a greater than 50% improved error rate compared with manual transcription. It is worth noting that Kurzweil VoiceRAD, the subject of the study, was a DOS-based system running on previous generations of microprocessors (386’s and 486’s). OSSIM runs on Windows NT and Windows 95, and uses improved recognition technology.

Discussion

One of the differences between OSSIM and its predecessors is that OSSIM depends much more critically on third-party tools that have to work together. VoiceRAD, the subject of the study mentioned in the previous section, consisted largely of a C program running in DOS. OSSIM, on the other hand, must successfully integrate C++, rapid application development tools such as Visual Basic, Object Design’s object-oriented database, various front ends that users may desire (MS Word, MS Access, Netscape Navigator), middleware such as IONA Orbix and MS ActiveX, and design tools such as Platinum’s Paradigm Plus and Microstar’s Near and Far. This list is far from exhaustive. All of these tools (in successive versions) must run on the versions of the Windows NT operating system that have been extant during the project’s lifetime, and many of them must integrate with (successive versions of) the source code control system (Atria’s Clearcase) which Kurzweil uses. Needless to say, managing all of the versioning problems that can arise in such an environment can be frustrating and time consuming.

In addition to problems of conflicting versions, we have had to deal with problems that arise because of immature technology. For example: we elected to use

(relatively new) object database technology for our knowledge base for the reasons outlined above. While we are satisfied that this was the best decision, we were hampered by the fact that there are no third-party data entry systems for our database which met our needs, and we were forced to build our own.

In general, we found that working with CORBA and SGML, in the limited sense of developing and revising DTDs and IDLs, was not particularly difficult. However, it was necessary to balance the flexibility required by the users of medical reporting tools with the requirements of adhering to the hierarchical structure inherent in our implementation of these standards.

Summary

The current pressures on the medical field make improvements in reporting and data gathering imperative. There is also a growing realization that information is useful to the extent that it is represented according to a widely adopted standard format. The voice-enabled, structured medical reporting system described here uses emerging technologies such as au-

tomatic speech recognition and object-oriented representation, as well as industry standards such as SGML and CORBA, to enable fast, efficient production of clinical reports that can easily communicate with a variety of output media and data repositories.

References ■

1. Hershey C, McAloon M, Bertram D. The new medical practice environment: Internists view of the future. *Arch Intern Med.* 1989;149:1745-9.
2. Dick R, Steen E. *The Computer-Based Patient Record: An Essential Technology for Healthcare.* Washington, DC: National Academy Press, 1991;14-23.
3. Goldfarb C. *The SGML Handbook.* Oxford: The Oxford University Press, 1991.
4. The HL7/SGML Special Interest Group Homepage: <http://www.mcis.duke.edu/standards/HL7/committees/sgml/>
5. Mowbray T, Zahavi R. *The Essential CORBA: Systems Integration Using Distributed Objects.* New York: John Wiley, 1994.
6. The CORBAMED Homepage: <http://www.omg.org:80/corbamed/>
7. The DMR Group. *A Professional Benefit Study of the Use of VoiceRAD within a Hospital Radiology Department.* Waltham: Kurzweil AI, Inc. 1991.