# Progress Notes Model

Houtan Aghili, Ph.D., Richard A. Mushlin Ph.D., Rose M. Williams, M.S.C.S.
Health Care Solutions Project
IBM T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532
Jeffrey S. Rose, M.D.
Director, Clinical Information Systems
Colorado Permanente Medical Group, 2550 S. Parker Road, Aurora, CO 80014

## ABSTRACT

*The largest part of the medical record today consists of notes documenting the care delivered to patients and the clinical events relevant to diagnosis and treatment. These "progress notes" serve as the repository of medical facts and clinical thinking, and are intended as a concise vehicle of communication about a patient's condition to those who access the health record. They should be readable, easily understood, complete, accurate, and concise. They must also be flexible enough to logically convey to others what happened during an encounter, e.g., the chain of events during the visit, as well as guaranteeing full accountability for documented material, e.g., who recorded the information and when it was recorded. This paper describes a model for progress notes, which addresses the above needs, and outlines the rationale and principles which led to that model.*

## INTRODUCTION

In this paper we describe a model and implementation of progress notes in an electronic patient chart, which is part of a clinical information system (CIS) designed and built for Rocky Mountain division of Kaiser Permanente Region, based in Denver, Colorado. Kaiser Permanente is a staff-model HMO, with some 400 physicians, practicing in 20 clinics, and serving a community of about 375,000 active members in the Denver area. At the time of this writing the system is installed in a clinic with 150 full function users (progress note authoring, electronic order entry and result review, etc.). Since any member can, in principle, go to any clinic, the system is currently populated with demographic and historical medical data for all members. Progress notes and clinical histories are the main features of the CIS application, which also includes order entry, results reporting, and appointment look-up. CIS includes other components for handling workflow management, ancillary system transactions, controlled medical vocabulary, and clinical data repository. The system is scheduled for rollout to all providers in all of the region's clinics in 1997.

## MOTIVATION

A patient's chart is a complex collection of all the relevant "facts" relating to the patient's health. This raw data can be quite voluminous, and has led to efforts to manage it with a computer rather than with paper [1,2]. But just keeping track of the data is not enough. To be useful and illuminating to the health care professional, the data must be organized in familiar and predictable ways [3].

There is no universal standard for the organization of a chart, be it paper or electronic. However, certain paradigms have evolved in modern medical practice using the paper chart. This can serve as the starting point for the level of standardization, which the computerized patient record demands. Two organizing principles stand out as useful approaches for making patient data more informative: histories and progress notes.

"Clinical Histories" denote the grouping of patient data by its type, *independent* of the context in which the data was recorded. Refinements can include further subclassification and sorting. For example, a medication history includes all medication data, but not lab data, even though certain medications may have been prescribed based on the results of certain lab tests. The medication history may be grouped by drug and sorted by date to make it more readable. Observe that it is only because the reader associates certain drugs with certain medical conditions that the medication history is useful for understanding the patient's problems and treatment. When several histories are considered together (e.g. medications, lab tests, procedures, family history, allergies), the experienced clinician can piece together a more complete picture of the patient's condition. The completeness of the record and the skill of the clinician combine to make the historical, data-centered view of the chart a useful clinical tool.

But the clinicians, and the health care enterprise, want more information. They want the context in which the data was generated. This context includes the relationships between medical facts and the thought processes involved in investigating, diagnosing, and treating medical conditions. The *progress note* is the vehicle for capturing that context. In practice, the degree of structure evident in progress notes spans the continuum from unstructured stream-of-consciousness text, to highly structured machine-readable forms. Although most progress notes exhibit neither of these extremes, the reason for the polarization is clear: the content must be both expressive and searchable [3]. A common approach is to record information as labeled, stylized, free text, optimized for visual scanning by clinicians, in one place, and selectively code certain information, optimized for processing by data analysts, in another place. With this scheme, neither

1091—8280/97/$5.00 

the clinician nor the analyst has easy access to the complete picture. It is a design goal of electronic progress notes to help unify this picture, resulting in a chart which is expressive, searchable, and facilitates both data-centered (historical) and process-centered (contextual) views.

## PROGRESS NOTE STRUCTURE

In light of the introductory remarks, the progress notes function of the electronic chart was designed around the following key points:

- All medical data entered into the chart should be part of a progress note, independent of their use elsewhere in the chart. The intent is to always preserve the context in which events occur.
- Progress notes should support both business and clinical analysis by storing data in coded form, while providing medically familiar textual documentation.
- Progress notes should allow for entries by multiple authors without sacrificing individual accountability. This behavior should support both the common doctor/nurse office visit scenario as well as the broader team approach to treatment.
- Progress notes should allow useful documentation patterns to be captured and later applied, with suitable modification, in similar clinical situations. The goal is to facilitate ease of use, reduce time spent entering data, and provide a framework for protocol definition and management.

The discussion which follows will describe a model to address these points. We will define the layers of data organization used for progress notes, expose the structure of the principal progress note components, and describe their behavior within, and effect on, the note.

### *Events*

A medical chart is both a transaction system and a documentation tool [4]. Some information is purely descriptive, and the documentation serves to highlight the clinically relevant aspects of the state of the patient. Other data is proscriptive, and the documentation serves to initiate the transaction as well as provide a description for future reference. It is important to distinguish between the data and the documentation. For example, an order "happens to" the patient, and is presumably for the patient's benefit, while the documentation of that order in the note is primarily for the clinician. The details, which are most important for the correct execution of the order, are not necessarily the same as those which provide the most appropriate documentation.

In our model, the data itself is captured in a unit called an event. There are many types and subtypes of events, organized into a hierarchy whose "leaves" (terminal nodes) are used as the transaction units of the system. Examples are a chief complaint, a set of vital signs, a clinical pathology result, a diagnosis, a medication order, and a patient instruction. These data structures are optimized for transactions and histories. They contain coded elements, numerical values, text descriptions, and system identifiers. Patient, author, and timestamps are among the identifiers present in every event. In addition, every event is directly or indirectly (through another event) linked to one and only one progress note.[1]

When an event is documented in a note, the event itself can provide information relevant for that purpose, including transformed or filtered information. For example, the code for a diagnosis is important in the transaction and search realms, but the name is the important documentation feature. By encapsulating functions which generate the descriptions of events, we help assure consistency of documentation throughout the chart.

### *Blocks*

The documentation requirements for a progress note, however, exceed the encapsulated descriptions of the events alone. Progress notes have evolved their own "look and feel", classifying and arranging information so that the reader can understand what the author was thinking. One common set of guidelines is the so-called **SOAP** model [5].

We believe that **SOAP** is the best model for a progress note. It is simple, yet sound and robust, offering a documentation framework that accommodates all kinds of medical information. In this idealized format, the author documents subjective (**S**) information, such as chief complaint, history of present illness, objective (**O**) data, such as vital signs, physical exam, a diagnosis or assessment (**A**), and a treatment plan (**P**), which includes orders and dispositions.[2]

In practice, the arrangement and appearance of the note can vary widely from the prototypical **SOAP** format, depending on the author's style, documentation goals, practice management guidelines, and the patient's condition.

Structures which impose a fixed arrangement of **S**, **O**, **A**, and **P** elements were rejected as unusable. Equally unacceptable were structures which label every entry with a **SOAP** letter, but arrange the entries chronologically. Our model automatically labels the entries, but allows the author to insert the entries anywhere, subject to certain restrictions for enforcing accountability.

One important accountability issue is the handling of revisions to existing documentation. Our model deals with two kinds of revisions.

---

[1] For events generated outside the system for which no clinically appropriate note can be assigned, the system could either create a new note or collect all such "orphans" in one special note.

[2] Our solution includes a knowledge base that contains a controlled medical vocabulary. Medical information, e.g., clinical findings in a physical examination, history of present illness, is stored as coded statements defined over this vocabulary. This strong support for codification makes our patient charts a valuable input for outcome analysis.

1. Recording errors, such as misspellings or entering "right" when the author means "left", are corrected by allowing the author to replace the information, but with the original data accessible for audit.
2. An evolution or refinement of clinical judgement is different, in that both the original as well as the revision are considered equally valid at the time they were entered. For example, an initial diagnosis of a sinus infection may be replaced with a diagnosis of migraine after further investigation or deliberation.

The "story" that the note tells certainly depends on the relative positions of the two judgements. Again, accountability requires that the model embody certain limitations on the arrangement of entries in the note.

To handle the labeling, arrangement, and accountability requirements of the note, our model defines a structure called a "block," which serves as the atomic unit of documentation. A block has access to the data stored in one or more events, maintains the correct labels for that data, maintains the correct position in the note, enforces limitations on rearrangement, maintains audit trails of in-place revisions, and constructs the appropriate documentation text to represent its underlying "event" data.

Both events and blocks are persistent in our model. The information relevant to transactions and histories is stored in events, while the information needed for progress notes documentation is stored in blocks, including an access path to associated events. This separation of function allows an application to change the "look and feel" of the progress note without affecting the transaction or history-building subsystems. This facilitates the integration of new event types and the retrofitting of our progress notes model to existing clinical repositories.

***Notes***

The progress note is modeled as a sequence of blocks. The order of the sequence is controlled by the authors and by the accountability model. Every block is assigned a chart-specific note identifier and a note-specific sequence position. Each event is assigned to one and only one block, but a given block may represent more than one event. This many-events-to-one-block relationship manages the "versions" used to correct recording errors.

This simple, yet elegant model of the progress note allows for a wide range of possibilities for authorship, appearance, and analysis. A noteworthy feature of our model is that notes have a well-defined beginning, when the first block is inserted, but no end. An author may add new information to an existing note, or start a new note, based on clinical judgement and practice guidelines, without *artificial* system limitations. This implies that a single note can be used to document a part of an encounter, a whole encounter, multiple encounters, or entire episodes of care, without having to strictly define "encounter" or "episode", for which there is not any standard universal meaning. Traditional paper charts usually attempt to impose a notion of closure on documentation based on the encounter concept. The encounter boundaries, however, are often determined by financial practices rather than by clinical models. An advantage of our electronic chart is that financial information can be obtained from the transaction record, while the documentation record can be better tailored for clinical use, without loss of data for either purpose.[3]

The act of "signing" a progress note takes on a special meaning in an electronic chart. In a paper chart, signing (or sometimes initialing) imparts two distinct meanings in one act. Namely,

1. It serves as a means to "authenticate" the author.
2. It attests to the author's confidence that the signed portion reflects sound medical judgement.

In the electronic note, the authentication task is accomplished automatically by the log-on procedure to the system; every block, therefore, has a specific known author. Signature, then, takes on a pure accountability meaning, which is captured with two special blocks: *Initial* and *Sign*. Together, they address the need for expressiveness, which requires that the note allow for the free arrangement of blocks, and the need for accountability, which requires that the author be able to prevent rearrangement and insertion, which in turn might change the perceived meaning of the original documentation. In our model, initialing a note prevents rearrangement above the initialing point, but allows insertion. Signing prevents both rearrangement and insertion above the signature. Correction of recording errors is always allowed for that block's author and never for another author, regardless of initials or signatures anywhere in the note. Accountability is therefore addressed by automatic identification, two levels of content locking, and controlled, traceable content versioning.

***Index***

Our electronic chart takes advantage of the high degree of internal structure and content coding present in the notes by constructing a progress notes index. The index contains attributes such as dates, diagnoses, authors, and procedures, extracted from all notes in the chart, and organized into a sortable, filterable, searchable structure. This structure serves the combined purpose of facilitating access to particular notes, while automatically maintaining a useful clinical summary, similar to the traditional "problem list." We are currently investigating alternative approaches for compiling and presenting summaries of progress notes data, which we believe will significantly enhance the value of the computerized patient record.

## BASELETS

Using the structures discussed above, together with collections of coded elements from controlled medical vo-

[3] As for storage requirements, the typical size of a patient chart in our system is O(100 kilobytes).

cabularies, an author can construct progress notes to document a wide variety of clinical situations. It would be advantageous if common patterns of documentation could be captured, tailored, and reused within the health care enterprise. This would save time and effort, as well as allow both standardization and customization of protocols. To be useful, such a system would require the right balance between very specific "templates," from which the correct instance would have to be selected from a very large library, and a small number of general purpose "guidelines", each of which would need extensive tailoring on every use.

Our system provides an intermediate mechanism, which allows progress notes to be written for a "virtual" patient, with two levels of customizability for application in specific clinical situations. We have coined the term "baselet" to describe a progress note in which coded elements from a " small view of the knowledge base" are precoordinated into "living" documentation "template" ("baselet") with the desired degree of flexibility.

A baselet is a combination of "macro" and "menu" representations of blocks within a progress note. The application provides an authoring mode in which baselet notes can be written *exactly* as they would be for "live" notes, with several special features:

- The baselet is written outside the context of a patient chart, and can be stored and retrieved for later editing.
- None of the blocks are committed, which means no transactions are generated, even when the specification of an event is complete.
- Checking for completeness of data entry is turned off, which means events need not be completely specified.
- Versions of events are not maintained, which means that editing a "baselet" block replaces its associated event.
- The coded elements, which represent the clinical data, can be assigned attributes to control their behavior during the tailoring process when the baselet is applied to a live note. This facilitates capturing *default* and *alternative* choices, deferring the selection of the exact mix until the baselet is applied. *Rare* choices, not anticipated by the baselet author, can be added by the user and seamlessly integrated with the current baselet.

Observe that the baselet notion offers significant advantages over the traditional notion of templates. Namely,

1. Templates have a "fixed" format. In order to serve a large user community with diverse needs, e.g., the various medical specialties, one has to design a large number of templates. Baselets, on the other hand, do not have a fixed format, and can be extended or abbreviated from a standard "startup kit," by each author as needed, without loss of coded information.
2. Templates, having a fixed format, document all of the fields they contain when committed. This implies that for cases when a template is sparsely populated, the entire template is still recorded, resulting in hard-to-read documentation. Baselets, on the other hand, only commit the blocks which the author determines to commit.
3. The authoring of templates is a separate and distinct process from their use. This results in having a group of template "designers," and another group of template "users." This separation of the function makes it particularly difficult to introduce new recording patterns into practice. Baselets, in contrast, are authored in exactly the same way as notes. This enables an enterprise, or an individual author, to take an existing note and convert it to a baselet for future use. That is, as the use of the system evolves, and the "useful" patterns are identified, they will be captured and reused for better documentation of care.

## PROGRESS NOTES IMPLEMENTATION

We have implemented a tool for authoring progress notes. Figure 1 illustrates a portion of a sample progress note. Depicted in this note are:

1. Two committed **P** blocks: an order for a lab test, followed by a prescription.
2. Four baselet blocks: an **S** block (Sore Throat), a reminder **O** block (prompting to take the vital signs), an **A** block (Acute Strep Throat), and a **P** block (a prescription for Amoxicillin 500 mg).

The author of the note may choose to "keep," to "drop," or to change any one of the baselet blocks, e.g., there is no need to keep the baselet **P** block. Once it is determined which blocks to keep, and which to drop, blocks may be committed to the patient chart, e.g., by signing the note.

## CONCLUSION

We described a model for progress notes, based on the **SOAP** model proposed by Dr. Lawrence Weed in 1971 [5]. We view the progress note as a logical integrator of clinical data relative to a health care contact, which should provide the caregivers with a full compliment of diverse medical data relevant to the patient care. We demonstrated that our model facilitates an expedient and yet flexible means for documenting care, provides full accountability, and produces readable, comprehensible, accurate and concise progress notes.

We have implemented a tool for authoring progress notes based on our model. This tool is a component of a clinical information system, which is under test in the Rocky Mountain Division of Kaiser Permanente. 150 full function users are currently using the system. Full deployment is expected in the spring of 1998.

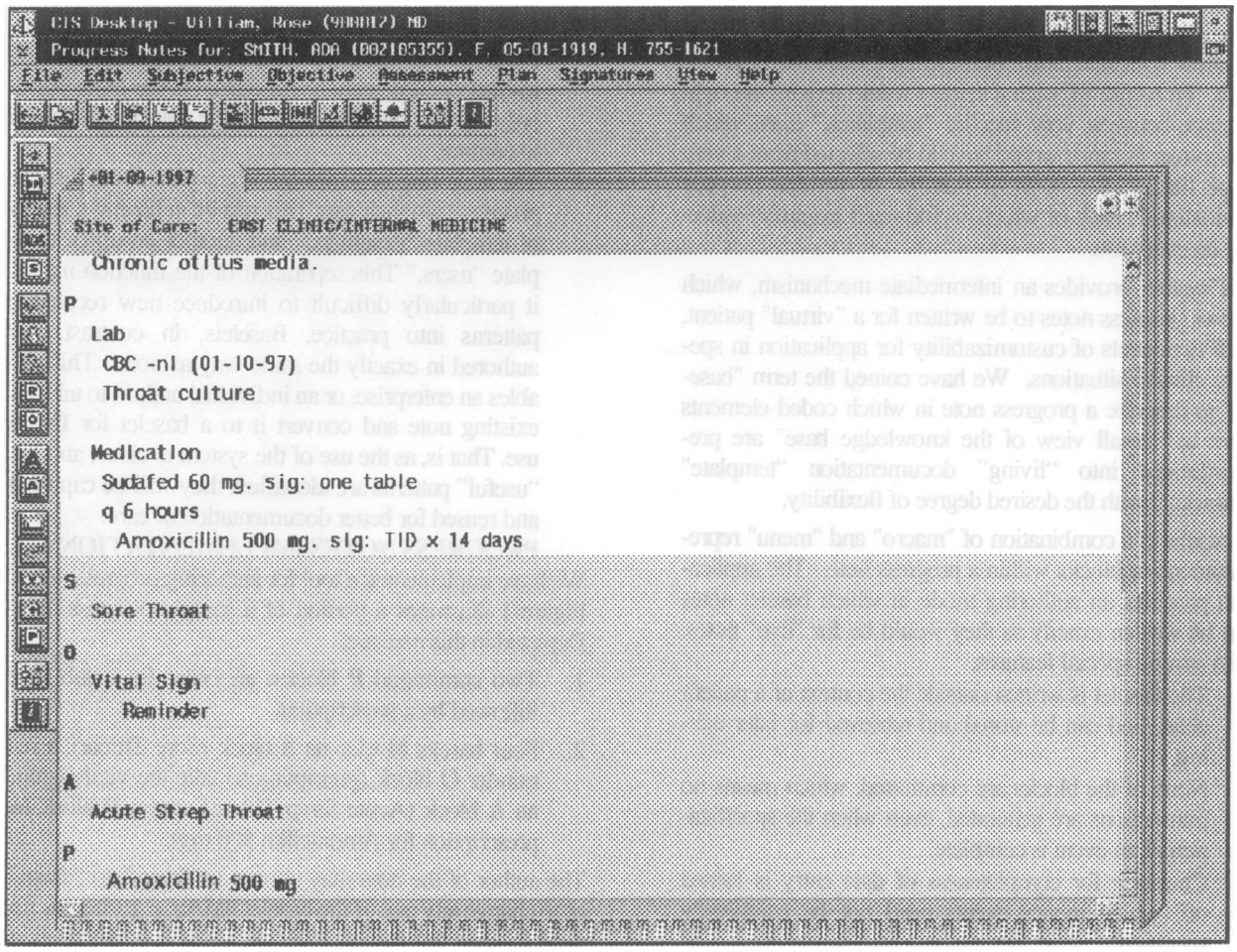


**Figure 1. A Portion of a Sample Progress Note**

## REFERENCES


1. R. Dick, E. Steen (Editors): "The Computer Based Patient Record." *Washington DC, National Academy Press, 1991.*
2. A.L. Rector, A.J. Glowinski, W.A. Nowlan, A. Rossi-Mori: "Medical Concept Models and Medical Records: An approach based on GALEN and PEN&PAD." *JAMIA, 1995, 2, 19-35.*
3. E. Nygren, P. Henriksson: "Reading the Medical Record I. Analysis of physician's ways of reading the medical record." *Yearbook of Medical Informatics, 1994, Schattauer, Germany.*
4. S.M. Huff, R.A. Rocha, B.E. Bray, H. Warner, P.J. Haug: "An event Model for Medical Information Representation." *JAMIA, 1995, 2, p 116-134.*
5. L. Weed: "The Problem Oriented Record as a Basic Tool in Medical Education, Patient Care, and Research." *Ann. Clin. Res., 1971, 3, (3).*