Taking the problem oriented medical record forward

Peter Salmon, Ann Rappaport, Mike Bainbridge, Glyn Hayes, John Williams
Primary Health Care Specialist Group of the British Computer Society

The problem oriented medical record (POMR) has proved to be very successful in providing a structure that helps doctors record their notes about patients, and view those notes subsequently in a manner that quickly gives them a good understanding of that patients history[1]. *This approach has been validated by the American Institute of Medicine*[2]. *With the increased use of computer systems that implement the POMR by doctors, the limitations of this structure have become apparent, and there is clearly scope for developing the model further to improve the quality of the data recorded, and adding meaning to it.*

This paper describes some of the limitations of the POMR, and discusses a number of areas in which it may be extended. Crucially, this is done in a manner which is both implementable, and usable. The extensions explored include some types of entity including encounters, episodes and subproblems; and an alternative view - the Timeline. The terminology used for the extensions is clarified.

Mechanisms by which these extensions have been implemented are described. Ways in which systems can manage these extensions automatically are suggested. Such implementations are constrained by the need not to allow the demands of the computer to intrude into the patient encounter. They are also constrained by the requirements for reporting by professional and governmental institutions, and by what is pragmatically feasible in software and hardware.

INTRODUCTION

There are many reasons doctors have in choosing to collect the patient medical record (PMR) electronically. One is to reflect it back to them during the patient encounter in ways which help them to quickly assimilate the key aspects needed to properly treat their patients. In this way, at its best, the computer directly improves the quality of patient care[3]. A good electronic PMR (EPMR) will provide different "views" of the data depending on the context the doctor wishes to see. This is best done using a **structured** medical record.

The POMR was first described in 1968 by Larry Weed[4]. In various forms it has had a widespread uptake by suppliers and users of medical computer systems because it provides a readily understandable structure for recording and viewing the patient record.

In the systems some of the authors have designed the patient record is divided into a series of sections, each one given a heading which broadly matches what may be perceived as a problem - either by the patient or by the physician. The label associated with the heading may change as the understanding of the problem develops, though the history of the heading should be available. Items in the patient record ("entries" - which may be of various types from simple notes to medication records, pathology reports, images etc.) are grouped under one or more of these problem headings. As the author of the record adds entries into it, the appropriate heading is chosen to put the new entries under. When the record is viewed only that information pertinent to the currently selected problem is viewed. In order to view all the entries for all problems in the order in which they were entered, an alternative "journal" view is provided.

As a result of the work done on the POMR, the "SOAP" note was developed (Subjective, Objective, Assessment, Plan)[5]. This extends the problem oriented model, by introducing some further structure. It provides a standard approach to recording information under a problem. More work has been done in this area, such as the ASOP method[6].

Other work has taken place since the POMR was first described, and in many ways the complex anatomy of the PMR has been carefully dissected. Rector et al[7] have taken the understanding of the PMR forward. Many of the principles laid down in this work have become integral parts of computerised systems. In the process they and Purves described their views of the limitations of the POMR (prescriptive vs. descriptive medical records[8,9]). However there is an elegant simplicity about the POMR which is appreciated by many of those who use it, and on which we would like to build.

0195—4210/96/$5.00

LIMITATIONS OF THE POMR

1. Many people find it quick to pick up and understand, but it can be complex to maintain.

2. Not all headings are problems. An example may be the displaying of all entries relating to the prevention of pregnancy under the problem heading of Contraception.

3. Patients often consult the doctor with more than one problem. The fact that many problems were discussed during a single consultation is not apparent in the pure POMR. However it may well be relevant to the overall interpretation of the patient record.

4. Some information properly belongs under more than one problem. For instance a blood pressure measurement may belong both under a "Ischaemic heart disease" heading, and under a "Hypertension" heading.

5. Sometimes it is useful to link entries across problem boundaries, for instance when one problem was "caused by" another : "the pneumonia was caused by the need to stay in bed following the operation on the broken hip". The POMR doesn't provide a natural way to express this.

6. The POMR provides only crude measures of the state of problems - they are either current, dormant, or resolved. However problems may go in cycles. They get better, then get worse again. The patient may not experience trouble from them for long periods. It is important to be able to look at a patient record and get an understanding of the frequency, length and timing of problems.

7. Different members of the health care team need to see the record in different levels of detail or "granularity". For instance, for the problem of diabetes, the diabetic specialist may want to see a different sort of detail to the diabetic liaison nurse.

8. Problems vary in size enormously. Those with few entries are easily read; large problems less so. The physician can find himself "drowning in data"[10].

When considering improvements to the EPMR, both for quality and usefulness, there are two conflicting pressures. There is a great reluctance among doctors to reduce the amount of time spent interacting with the patient, in favour of entering data into the computer. In fact in one project a computer generated extension to the encounter of only one minute, prevented its implementation[11]. The extensions to be discussed have in practice to either require no extra time for data entry, or by improving the understanding of the patients record, to reduce the time spent elsewhere. This principle underlies all the discussion that follows.

THE TIMELINE

Classically the view of a patients POMR involves two parts, a list of all the patients problems, and a list of all the entries in the patient record that are grouped under one selected problem. The problem list may be filtered by the status - current, dormant or resolved They will be sorted into chronological order - either date of entry, or date applies to. Two of us have implemented this as in figure 2.

The problem view generally allows one problem to be considered at a time. In order to provide a chronological comparison of problems that the patient has had, a timeline may be created. The timeline has been described by Cousins and Kahn[12]. We introduce it here to better consider some of the subjects which follow. On the timeline the horizontal axis represents time, with earlier dates on the left, and the present (or even the future) on the right. Each problem exists as a bar, along the time axis. The left end of the bar is at the time the problem started (or the physician was aware of it), and the right end either stops where the condition resolved, or continues to the present.

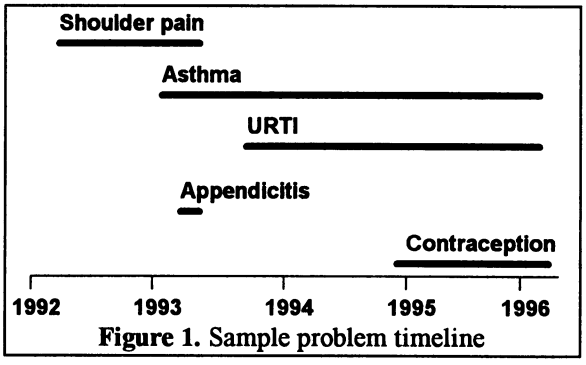

Figure 1. Sample problem timeline

ENCOUNTERS

An encounter is a concept which can be difficult to accurately define. For the purpose of this paper we will define it as being ***a communication about the patient between two or more individuals, at least one of whom is a member of the health care team.*** The communication may be direct such as a face to face or telephone conversation with the patient, or indirect such as a letter or report received from the hospital.

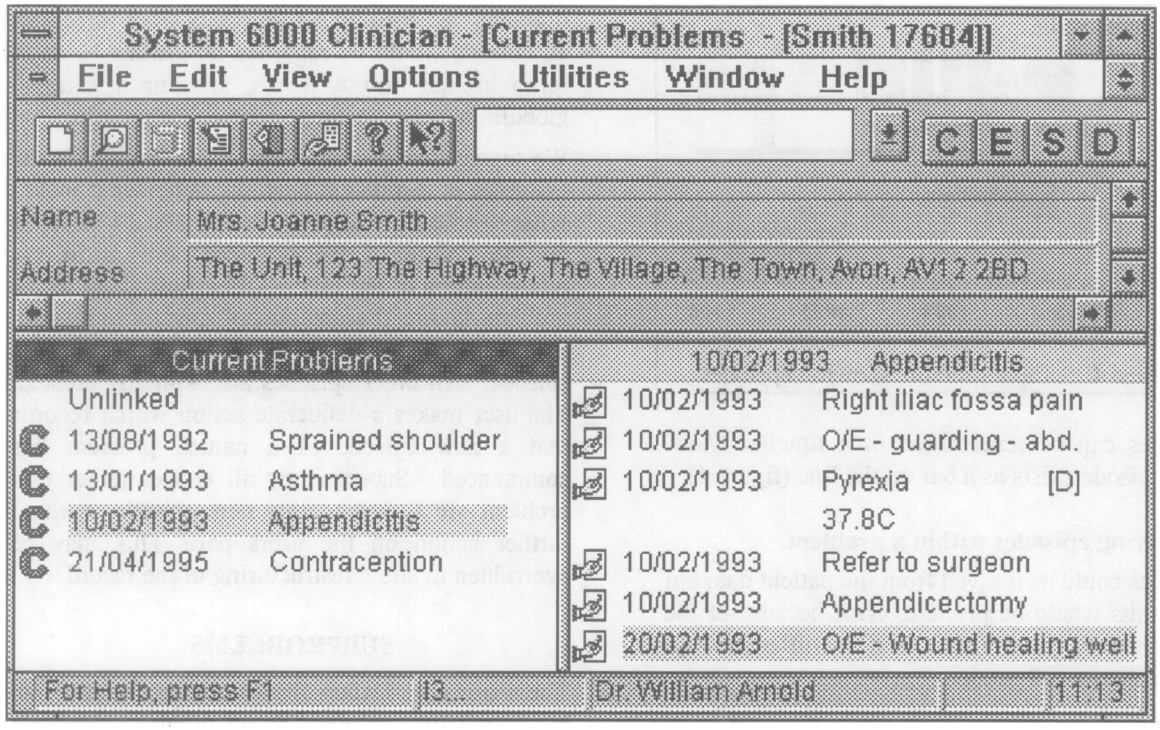

Figure 2. Sample implementation of a POMR

Encounters could superficially be derived by collecting all the entries with a common notedate together for one user. This method is certainly possible, but it is crude, with significant risk of error, and encounters are more than just notedates.

In the context of a doctor-patient encounter it is significant to note the date, time and reason for the encounter, the participants, and whether they were seen directly, or by phone or letter. Analysis of the number, frequency and purpose of encounters is then possible. During an examination of the POMR an encounter may involve several of the patients problems. This may be described by extending the diagram of the timeline above to show one encounter, on 20th February 1993.

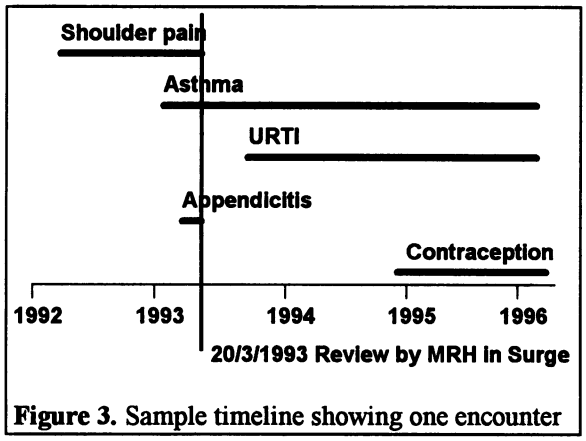

Figure 3. Sample timeline showing one encounter

As can be seen from figure 3, the encounter on the 20/2/93 may have dealt with the shoulder pain, appendicitis, and asthma. Of course, not every encounter deals with every problem and this view doesn't confirm that these things did in fact occur. For a system to be able to demonstrate what information was collected during an encounter it must collect encounters as entities within the patient record. All entries are then linked to encounters. A view is then provided which displays all the entries for a given encounter, not unlike the problem view which can display all the entries for a given problem.

EPISODES

There are many problems which are not active all the time, but come and go at intervals. For instance an asthmatic always has a problem of asthma, within which are acute attacks. ***The periods when the problem is active may be known as an "Episode" of the problem.***

Although encounters may apply to many problems, an episode only ever applies to a single problem. In the patient with asthma, he may get wheezy when he has a cold. So he has an episode of URTI, and a coincident exacerbation of his asthma. They clearly are related, but not the same. In the encounter view they would appear together.

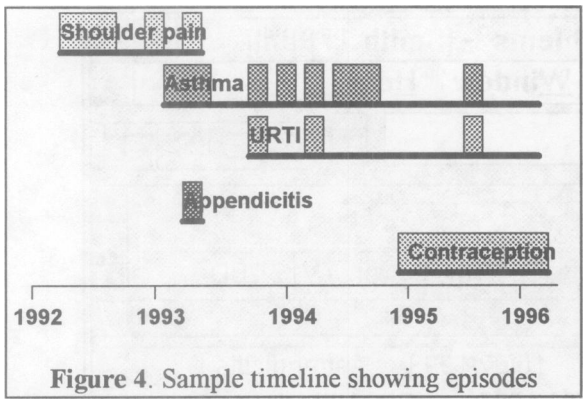

Figure 4. Sample timeline showing episodes

Episodes can be represented in a timeline view. Each episode exists as a bar on the line (figure 4).

Identifying episodes within a problem

Episodes could be derived from the patient data but the results would be prone to error because of the wide range of types of data, and the different ways people choose to store it.

A second approach is to explicitly enter episodes into the patient record. This could be done by creating entries that themselves represented an episode, or by attaching a tag to any other sort of entry, which indicated that this entry identified a new episode. The tag could be labelled to describe what sort of episode it was. Once an episode has been explicitly created, other entries can be linked to it. All entries for one episode can then be viewed together, and the episodes of a problem can be extracted and displayed without the detail of their contents.

Describing the scope of an episode

The commencement of an episode of a problem is generally straight forward to define. In secondary care the end of an episode will generally be when the patient goes home. However, in primary care the end of an episode is more difficult to establish. Often the patient will not return to report recovery. So how can the end point be established ? There are a number of strategies, which may be used in combination.

a) The end point could be manually entered if the circumstances allowed. This then would be the definitive end. Usually this will not be possible.

b) Consider the episode as the period containing patient encounters within which the problem was addressed before it was identified as inactive.

c) Automatically define the duration (and so the end point) of an episode based on the type of problem. So for upper respiratory tract infections an episode would be regarded as lasting 10 days, for a broken arm 8 weeks, and for a gamma-globulin hepatitis A immunisation - 6 months.

We suggest that an initial episode is implicit in the existence of a problem heading. Whether it makes sense to consider the problem episodically depends on the nature of the problem, and is in the interpretation of the user. All entries added to the problem before the creation of any more explicit episodes are automatically linked to this first episode. All other episodes are explicitly created. The user makes a deliberate action which records that a new episode of a named problem has commenced. Subsequently all entries under that problem are linked to this new episode, without further action on the users part. This may be overridden to allow restructuring of the record.

SUBPROBLEMS

There are times when the contents of a problem becomes very complex. Some problems have so many components that each part needs to be considered independently. Further structuring is desirable. This is supported by the concept of subproblems.

Subproblems are problems contained within a problem. In all other ways they behave just like problems. They provide a grouping mechanism for a list of related entries. They can be episodic.

One example could be the case of a patient with Crohn's disease. To represent all the different components of this disease (fever and abdominal pain, diarrhoea, perianal fistulae and abscesses, obstruction and malabsorption etc.) under a single heading is quite inadequate. Each component could however be represented as a subproblem. The detail relating to each subproblem is linked to the subproblem, not the top level problem and is only displayed by selecting the subproblem.

There may be value in having a "problem-journal view" which displays all entries under a problem, even if they are under a subproblem of the problem.

IMPLEMENTATION ISSUES.

In order to develop a computer system which includes the features mentioned above, a number of issues of implementation have to be addressed. Most of these are the scope of the development team, but we would like to make a few suggestions from our experience. The main objective was to

produce as comprehensive, and accurate a record as possible whilst not interfering with the doctor-patient encounter.

The computer must be able to make intelligent assumptions about the data entered. These assumptions must be both visible and easily corrected. This requires the system to have knowledge, and this system knowledge can be developed to enable the capture of some of the types of information, without burdening the doctor.

Encounters

We can describe a group of consequtive patient encounters, occuring during part of one day as a session. We suggest that new entries in a patient record are automatically linked to an encounter, the characteristics of which are persistent between patients within a session, characteristics which may be easily changed. The first encounter created will take its characteristics from a set of defaults for the user. This will be confirmed by the user. It will then become the default "Session encounter" for the rest of the session. Subsequent patients will have an encounter which matches the session encounter created automatically by the system, without confirmation, as the first entry in that patients record is saved.

Episodes

We have said that episodes (after the first) should be explicitly created by users. However the system could be instructed to look for specific notes within named problems and automatically generate a new episode when such an entry was created.

Consider the problem of tonsillitis. If the system is instructed to create new episodes each time the coded note "Acute tonsillitis" is created, the following may occur :

1991 New patient with sore throat. Diagnosis : Acute tonsillitis. Read code for acute tonsillitis inserted -> event hooked for new episode. System responds by searching for an existing problem of "Acute tonsillitis" - not found so a new problem with this title is created.

1992 Patient returns with sore throat. Diagnosis : new episode of acute tonsillitis. Read code for acute tonsillitis inserted -> event hooked for new episode. System responds by searching for an existing problem of "Acute tonsillitis" - and one is found so a new episode is created under this problem heading.

Acknowledgements

This work is the result of discussions of the Clinical computing special interest group of the Primary Health care specialist group of the British Computer Society.

References

1 Hayes G. Computers in the Consultation. The UK Experience. SCAMC 1993:103-106

2 Institute of Medicine. The computer-based patient record: An essential technology for health care. Washington DC: National Academy Press, 1991.

3 Purves I. The current state and future needs of the electronic medical record for primary health care: A pragmatic view for the United Kingdom. PHCSG Annual Conference proceedings 1993: 113-122

4 Weed LL, Medical records that guide and teach. New Engl J Med; 278:593-9 and 278:652-7.

5 Weed LL. Medical records, medical education and Patient care: Press of the Case Western Reserve University, 1969.

6 Gensinger RA, Fowler J. ASOP: A New method and tools for capturing a clinical encounter. SCAMC 1995;19:142-146

7 Rector AL, Nowlan WA, Kay S, Goble CA, Howkins TJ, A framework for modelling the Electron Medical Record. Methods Inf Med, 1993; 32 : 109-119

8 Rector AL, Nowlan WA and Kay S. Foundations for an Electronic Medical Record. Methods Inf Med 1991; 30:179-86

9 Purves I. The current state and future needs of the electronic medical record for primary health care: A pragmatic view for the United Kingdom. PHCSG Annual Conference proceedings 1993: 113-122

10 Rogers J & Solomon D: Putting the clinical into Clinical Information Systems. PHCSG Annual Conference proceedings 1995: 20-35

11 Warner H, Mason C, Livingstone J, Bray B. Enroute toward a computer based patient record. The ACIS Project. SCAMC 1995:152-156

12 Cousins S, Kahn M. The visual display of temporal information. AI in Medicine 3 (1991): 341-357