Adaptive Entry System for Electronic Medical Record

龍安靖	An-Jim Long	陽明大學公共衛生研究所、衛生資訊與決策研究所
劉建財	Chien-tsai Liu	台北醫學大學醫學資訊研究所副教授
張博論	Polun Chang	陽明大學衛生資訊與決策研究所副教授
許明輝	Min-Huei Hsu	台北市立忠孝醫院神經外科主任
李友專	Yu-Chuan Li	台北醫學大學醫學資訊研究所教授
簡文山	Wen-Shan Jian	台北醫學大學醫務管理研究所講師
劉碩琦	Shuo-Chi Liu	陽明大學公共衛生研究所、衛生資訊與決策研究所

ABSTRACT

Managing, analyzing, and reusing of medical knowledge have been limited by the absence of non-structure narrative medical records. The authors in this paper present methods to simplify data entry and management of narrative electronic medical records (EMR) using self-exzplanatory, hierarchical structure and content components as data entry templates. The authors implemented the data entry templates in a prototype of an Electronic Medical Records system for surgical Intensive Care Units. The feedback from physicians showed the data entry templates did not only accelerate the input and compilation of medical records but also increased the ability to reuse the stored medical knowledge.

Keyword: Electronic Medical Record, free-text narrative record, adaptive entry system, template medical phrases, Script-based method

:

1. BACKGROUND

Medical records provide a view of a patient's health history, a method of communication among medical professionals for patient care, and a supporting document for health claims to payors.[13] Narrative medical records are a form of medical records that have free and non-structure text. Examples of these can include: Admission notes, discharge notes, progressive notes, on/off service notes, diagnosis reports, consultation notes, nursing care notes etc. Currently in hospital settings most of computerized physician workstations can usually only open free-text editing windows for compilation of medical records thus only producing electronic narrative patient records. They can hardly be analyzed and indexed for clinical decision making as well as academic research efficiently. In order to facilitate data entry and the efficient use of the electronic narrative medical records, the authors implement an adaptive EMR entry system that can meet the following requirements:

1. To provide an easy user interface for quickly and efficiently entering, editing, and compiling data.
2. To maximize the reusability of structural medical records via template construction.[1, 2]
3. To provide efficient tools for management, administration, analysis and search of the narrative medical records.

2. METHOD

To accomplish this difficult task, we first proposed an adaptive, hierarchical content and structure architecture for medical records. This architecture classifies components from the entire range of medical records into a taxonomic system with a hierarchical of subsets including classification, title, and content. The path from root to leaf is present as: document -> paragraph -> sentence -> phrase. The architecture is a dynamic one, adaptive to any kind of narrative medical record. We then designed an algorithm to identify hierarchical components and transform them into natural language via a script-based approach. The algorithm presents an approach to combine content components and script into a human readable document. Finally a script-generating assisting application is proposed to assist the primary script editor to generate the script easily.

Script was defined as[12]: a program or sequence of instructions that indicate a series of action, designate file or other resources, and their interaction.

2.1 Document component architecture

Presently, narrative medical records can differentiate varieties of documents by purpose and category. Documents can thus be classified into paragraphs, sentences, and phrases. As shown in figure 1, there were at least (but not limited to) four levels of the document structure. Level 1 was the root of the system. Branching on was level 2 was the first sub-folder of the document e.g. nursing, general, or surgery. Level 3 classified the purpose, e.g. on/off service note, discharge summary, admission summary, and operative note. Level 4 (with white ground and gray border) classified the free-text subcomponents of the document or paragraph. The level of structure could also expand to customize and subdivide infinitely (e.g. sentence level and phrase level.)

The highlighted area of figure 1 shows an example of how an admission note divided into several components. The method could adapt other free-text records such as on/off service notes, discharge summaries, and operative notes with the same approach.

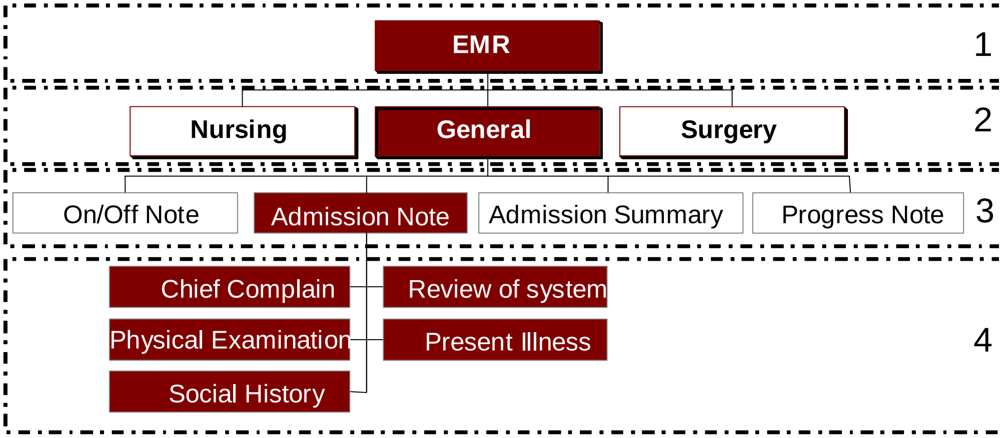

Figure 1 Architecture for narrative medical record

In order to identify components of all levels for further processing, three major data entities were used, their elements are listed as:

1. Template component: Stored the entire structure of all components (Cuboids from Figure 1) of figure 1 architecture and form template, which could be used by other users or purposes. All stored components were self-indexed by attribute – <u>ComponentID</u> and <u>ParentID</u>.
2. Result Report: Stored edited narrative record.
3. Script: Stored the code of sentence components (presentational style.)

Table 1 presents physical schema transformed from the entity-relationship diagram (ERD) shown in Figure 2. There were three entities after normalization.

Entity	Attribute	Description
Template component	1.CID	Component Identifier, consisting parent's identifier and starting from A1 to Z9 in hexadecimal.
	2.CName	Full name of the component.
	3.Category	Component category with three values of C, V and I, which stand for Category, Value, and Input.
	4.Level	The level of the component (As Figure 1).
	5.Type	Data type, which could be String, Integer, or Date Time, of the component if V (Value) is selected for the attribute Category.
	6.ParentID	Identifier of component's parent node.
	7.ResultID	Identifier of presenting which component's identifier to be used in the script.
	8.ScriptID	Identifier of the script that characterizes the role of this component.
Result report	1.ChartNo	Combination with sequential number as the Primary key for the charts.
	2.SeqNo	Sequential number of the charts.
	3.CID	All identifiers clicked by the user.
	4.Result	The narrative text (Natural Language) after processing.
Script	1.ScriptID	Identifier of the script.
	2.Script	Script for generating natural language.

Table 1 Physical Schema Transformed From ERD

Table 1 uses the matrix to present the schema that actually transformed the structure into the database management system (DBMS). The entity column shows the table name of three entities. The attribute column shows the field names (or attribute names) of each entity. The description column explains all attributes in detail.

The category attribute (attribute 3 of template component entity) activates the application interface to a dynamic one. C values pop-up with a user click based List-Box. I values popup an input dialogue-box when user clicks.

The Result ID attribute (attribute 7 of template component entity) plays an important role and replaces script's identifier with real/full name when script generation is processed. Two closeness identifiers usually have the same result identifier.

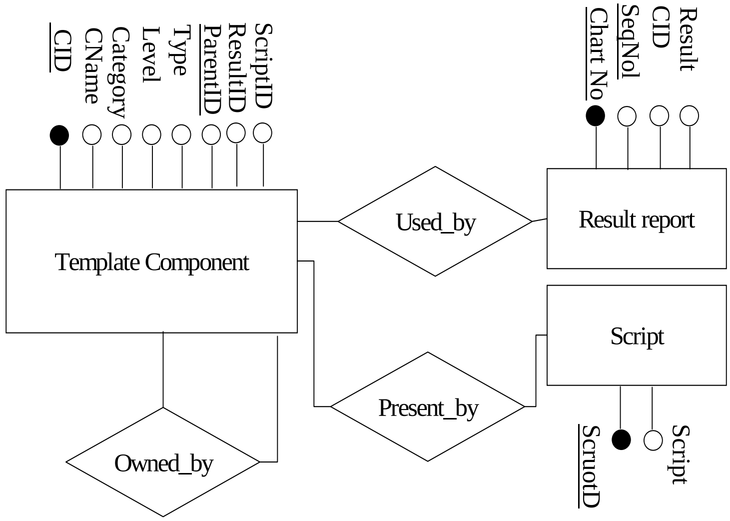

Figure 2 ERD of narrative medical record architecture

Figure 2 shows the ERD of how this research transformed the logical architecture into physical schema. The diamond "Owned by" shows the self-indexed relationship of template component entity. The full circle means the primary key of the entity, the hollow circle with underline name means the foreign key of the entity.

2.2 Combination model for template components and script

To simplify the generating process from template components to natural language, the script representative method was proposed. The rules of script (as shown in table 2) are shown as followed:

1. Script used to be (but not limited to) defined components below level 3 (e.g. Chief Complain - A1A1A1 of Level 4.)
2. Symbols used in script:
   - a). A square bracket […]: Bracket the component identifier, e.g. [A1], to represent terminology, which maps values to database in the script.
   - b). A bracket (…): Component in brackets could be optional, may or may not be needed, e.g. (at [[A2A2A1] [A2A2]].)
   - c). ~: Repetition, e.g. ~[A2A2A1] to represent the components which could be selected multiply.
3. Category of the component:

According to table 1, the category could be C, V, and I. C means "Category" and popup a menu when user clicks the 'C' component. V means Value. I means Input and will popup an input dialog when user clicks 'I' component.

4. Words or phrases (e.g. at, and) could fill anywhere in the script except above-mentioned rules.

The above part of Table 2 shows the original style of script, the below part shows the logical meaning of the original style of script.

Script ID	Script
A1A1A1	[~A1A1A1A1] (at [~[A1A1A1A2A1] [A1A1A1A2]]) (for [A1A1A1A4]) [A1A1A1A5] (at [A1A1A1A6]) (due to [A1A1A1A7]) (, refer from [A1A1A1A8]) (and send by [A1A1A1A9].)
Logic Meaning	Logical Meaning
Chief Complain	[sign and symtomps](, [sign and symptoms]….and.) (at [site] [location](, [site] [location]…and.)) (for [duration]) (since [since_time1 since_time2 since_time3]) (at [place describe]) (due to [reason describe]) (, refer from [refer describe]) (and send by [send]).

Table 2 The comparison of script and its logical meaning

2.3 Method of script generating

The script generator loads the computer system with codes before the selection of phrases, and generates a computer understandable content from script via the following steps.

1. Dividing bracketed content and non bracketed content into a computer array,
2. Adding repetition flags to every elements with repetition symbols,
3. Terminating normalization when every square bracketed content is unable to be divided again,
4. Reprocessing step 1 to 3 in recursive function when every element in an array unable to divide.

The system used two 2-dimensional arrays (see Table 3, the simulation of the array) to process script generation. One was read only while system process, while another was a temporary array and used to be replaced by phrases (i.e. CName of Table 1)

Sentence processing array	Flag
[A1A1A1A1]	~
at	
[A1A1A1A2A1]	~
[A1A1A1A2]	
for	
[A1A1A1A4]	
[A1A1A1A5]	
at	
[A1A1A1A6]	
due to	
[A1A1A1A7]	
, refer from	
[A1A1A1A8]	
and send by	
[A1A1A1A9]	

Table 3 Array for sentence processing

The system went to the status of While-Loop after the arrays generated. The steps of the process were shown as followed:

1. User clicks to select the component, and the components selected enter into the arrays from the application interface,
2. The system finds the relative script for presentation, and then processes the script as a "method of script generating" (as mentioned before), and compare identifiers with arrays of processed script, and replace square bracketed identifiers with components' name,
3. Reiterate step 1 and 2 until user quits the paragraph editing screen,
4. Combine all elements of array to form natural language.

Table 4 below illustrates the steps previous mentioned. Components which user selected enters into the array, are compared with script, and finalized for the following sentence: Nasal Pain at frontal area for 2 hours yesterday at home due to accident, refer from LMD and send by private car.

Component ID	Real Value	ID to be replaced in the Script
A1A1A1A1A2	Nasal Pain	A1A1A1A1
A1A1A1A2A4	Frontal area	A1A1A1A2
A1A1A1A4A3	2 hours	A1A1A1A4
A1A1A1A5A1	Yesterday	A1A1A1A5
A1A1A1A6A1	Home	A1A1A1A6
A1A1A1A7A5	Accident	A1A1A1A7
A1A1A1A8A2	LMD	A1A1A1A8
A1A1A1A9A1	Private car	A1A1A1A9

Table 4 examples of inputted components and the ID to be replaced in the script

3. RESULT: SYSTEM IMPLEMENTATION

In order to test the accessibility, and usability of the methodology, the authors of this research implemented the system at the 544-bed Chung-Hsiao Municipal Hospital, Taipei, Taiwan during June 2001. The system was expected to solve typing efficiency problems and problems related to the poor structure of the current EMRs. The purpose for this study was to enable the generation of narrative medical records by just using a mouse click.

3.1 Main Screen

As shown in figure 3, level 2 components are presented on the screen in a black and gray folder. The general doctor and patient information is displayed at the top of the screen. All level 3 medical notes are included in the folder. After the user section has been selected, blank text boxes appear below the folder section allowing the input of level 4 information in free format. All elements in figure 5 are generated from the database table including the folder category, note selection, and the text input area.

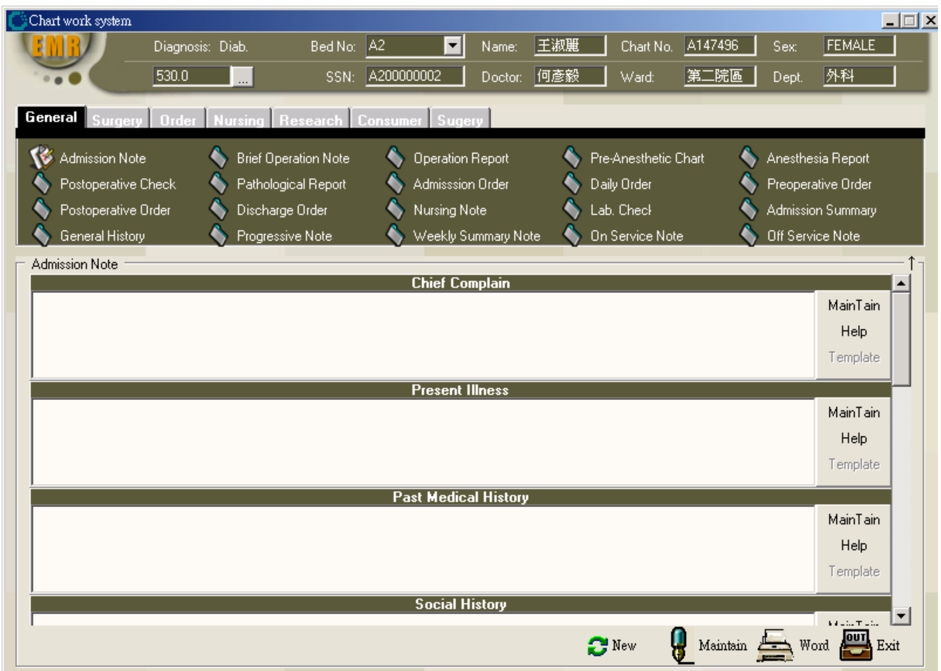

Figure 3 Main screen of the system

3.2 Text-based editing screen

The user can edit in the blank area of the screen (The input area under Chief Complaint as shown in Figure 3), or click the help button (On the bottom right-hand side of Figure 3) to enter notes in the text based editing screen (Figure 4).

Figure 4 shows the screen, which automatically enables the generation of title, components and other interfaces. The screen shown in figure 4 allows an inexperienced user to write either from left to right or top to bottom and to finalize the medical record by just clicking the mouse to check off the appropriate contents, show in the component hierarchy.

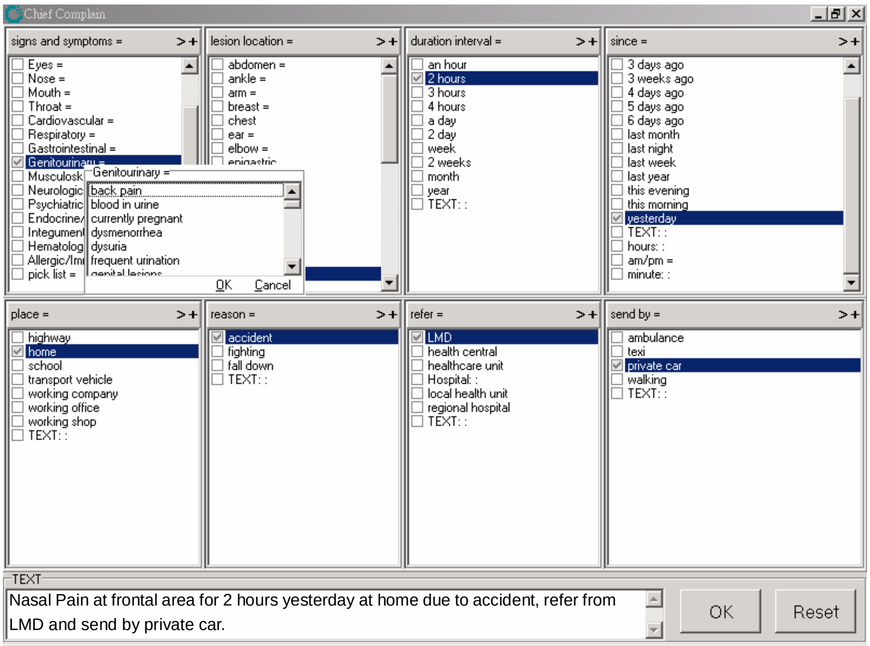

Figure 4 paragraph-based editing screens

3.4 Component maintaining screen

To assist maintenance of the structure and hierarchy of the content, the authors implement the maintenance screen to provide a tree-like presentation (left side of figure 5) and allow user to load template from formatted ASCII textual file or Excel file into any node.

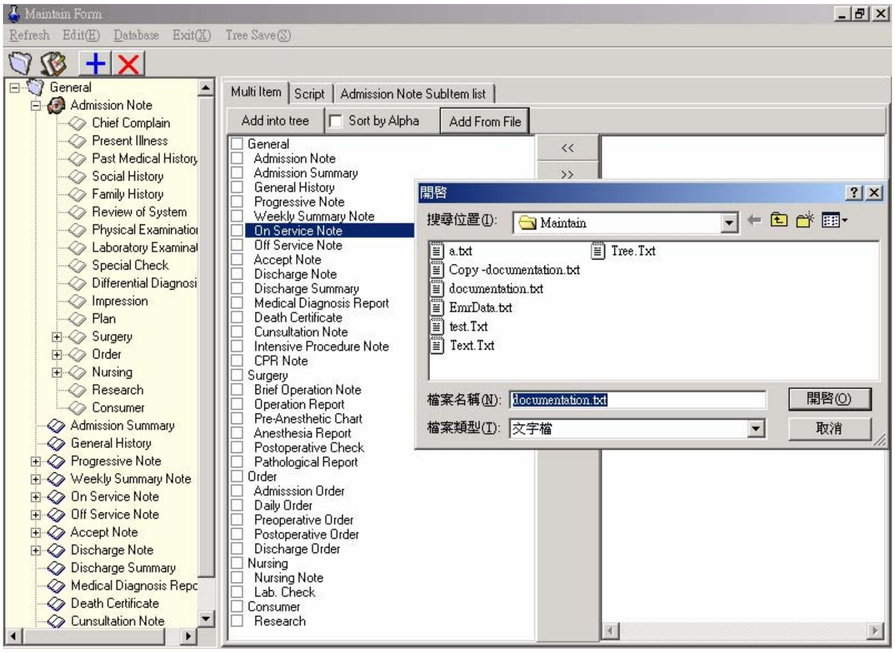

Figure 5 Maintaining application for template components

3.5 Script-generating assisting application

Since the script writing was very difficult for common users and any tiny mistake might cause compiling error, we developed the script-generating assisting application as shown in Figure 6. The script-generating assisting application was composed of three sections: the area showing the tree structured components (right side of Figure 6), the script editing area, and the natural language testing functions area. Users could also customize the style of personal-based script.

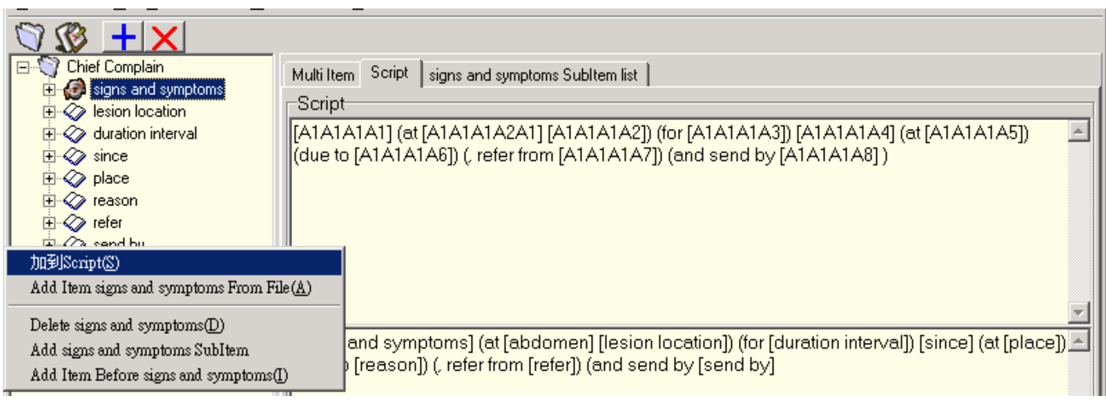

Figure 6 Script generating assistant

3.6 Information retrieval

As a result of replacing a free-text record with one based on terminology, the physician can use the editing screen to search for patient information via a mouse click. An example of a query command could be “Search for Patient who has a Nasal Pain caused by an accident and refer from the LMD”. Instead of using a full-text search, this type of search provides search results with higher accuracy and quality.

4. DISCUSSION

The authors have designed an architecture for entering EMRs that enables the users to take advantage of the reusability and accessibility of narrative electronic medical records. Other architectures for archiving EMRs also proposed:

4.1 Form-based

Form-based approach follows the guidelines on coding schema while entering data via a variety of user interfaces[3]. Webster (2001) [4] proposed a program that coordinates pre-set screens for detailed history, physical examination, treatment and prescription modules. It also presents "pick lists" that allows further customization and individualization of data inputs. Form-based method standardizes the entry style and the working model. The disadvantages about this method are the difficulties to customize the way that the physician works.

4.2 Natural language processing

The techniques of natural language processing can be applied to transform medical narrative into a form more suitable for information processing and management. CAROL (1999) [5] used the MedLEE system to process a narrative chest radiograph report into problem, body location, degree, and certainty. GEORGE (1997) [6] used the pulmonary tuberculosis guidelines published by the Center for Disease Control (CDC) to determine the positives or negatives of the patient by processing their medical records. The results showed higher recognition ability. Natural language processing can allow the user to write medical records in their normal way of working and to output the data in a structured form, but the accuracy must be increased by the support of a complete knowledge engine and by limiting the process target.

Other methods presented as semi-structures were proposed, such as structuring data via XML[9], HL7 [11] (American National Standard Institute now approved the method with the name of Clinical Data Architecture) or customized syntax and divided into several levels. The four-level XML based CDA standard provides the clinical document with persistence, stewardship, potential for authentication, wholeness, and human readability. Kahn (1999) proposed [7] standardized data elements and predetermined data-entry formats to record observations via XML tags. The weakly structured medical record gave users flexible methods for data entry and adapted to the user's way of working. Laforest (2000) [8] proposed semi-structured documents as a user interface for a database-based system. The captured documents are analyzed so that information is extracted and used to fill in a database. The results not only match the physician's way of working but also satisfying querying performance. The composition method combines the advantages mentioned previously, but also the disadvantages, including an unsatisfied structure, a limitation of natural language understanding, and a computer’s low accuracy to distinguish correctly.

4.3 Lessons Learned during Preliminary Implementation

Firstly, when we implemented the structure into the real hospital, we found if we chose the level 2 to point the primary elements of the medical records, the user wouldn't find the required note easily. Therefore, we suggest classifying the real document to the level 3.

During the preliminary implementation, we chose the admission summary to be the primary testing document. Twelve residents were invited to participate in this research. Seven of them considered that the system could reduce the editing costs of admission summaries after adjusting the template to their way of typing. The rest of the residents didn't want to join the project due to the difficulty of adjusting the template; two of the residents were unfamiliar with using word processors to edit admission summaries. The response of 11 physicians was that the difficulty of coding script would be the biggest entry barrier to promoting this system.

This research proposed a practicable methodology to implement an adaptive EMR system to structure narrative text into semi-structured data. Feedback from physicians of Chung-Hsiao municipal hospital showed the following advantages of the system:

1. The system could reduce the total time spent editing medical records through provision of appropriate templates and by selecting phrases with just a mouse click.
2. Medical records were semi-structured after being edited by the system and the coded phrases increased the reusability through advance searching, summarization, and statistical analysis,
3. The completed reports could accumulate as templates and were effective in accumulating knowledge about written medical records,
4. The system could be applied to most of the general narrative medical records.

After implementation, we also found the limitations of this research:

1. Medical records sometimes needed to be presented and combined with pictures, but an effective solution still unavailable.
2. The total size of leaf nodes expanded too fast and lacks reusability.

The future work of this research will focus on introducing a terminology database (e.g. UMLS[10]) to standardize the coding schema, automatically script generation, and enhancement of the reusability of components and the programmability of the script.

REFERENCE

[1] Liu CT, AG Long, YC Li, KC Tsai, HS Kuo. Sharing patient care records over the World Wide Web. International Journal of Medical Informatics. 2001(April). 61(2-3): 189-205.

[2] Liu CT, PJ Huang, AG Long, KC Tsai, SS Kuo. A Web-based Referral Information System. New Taipei Journal of Medicine. 2000. 2:39-50

[3] Friedman C. Towards a comprehensive medical language processing system: methods and issues. Proc AMIA Annu Fall Symp 1997; 595-9.

[4] Webster C, Copenhaver J. , Structured data entry in a workflow-enabled electronic patient record, J Med Pract Manage 2001 Nov-Dec;17(3):157-61

[5] Carol F, PhD, George H, MD, etc, Representing Information in Patient Reports Using Natural Language Processing and the Extensible Markup Language; J Am Med Inform Assoc 1999 May-Jun;6:76-87

[6] George H., MD, Charles A. Knirsch, MD, etc. Automated Tuberculosis Detection; J Am Med Inform Assoc. 1997;4:376–381.

[7] Kahn CE Jr, Standard Generalized Markup Language for self-defining structured reports. Int J Med Inf 1999 Feb-Mar;53(2-3):203-11

[8] Laforest F, Flory A., The electronic medical record: using documents for information capture. Stud Health Technol Inform 2000;77:617-21

[9] Dolin RH, Alschuler L, Behlen F, Biron PV, Boyer S, Essin D, Harding L, Lincoln T, Mattison JE, Rishel W, Sokolowski R, Spinosa J, Williams JP HL7 Document Patient Record Architecture: An XML Document Architecture Based on a Shared Information Model; Proc AMIA Symp 1999;:52-6

[10] Betsy L. Humphreys, MLS, Donald A. B. Lindberg, MD, Harold M. Schoolman, MD, G. Octo Barnett, MD; The Unified Medical Language System; ; J Am Med Inform Assoc; 1998 Jan-Feb:5:1-11

[11] HL7 website. Available at http://www.hl7.org. Accessed September 20, 2002.

[12] Technology term what is. Available at http://whatis.techtarget.com/. Accessed September 20, 2002.

[13] Definition of the Health Record for Legal Purposes. Available at http://www.ahima.org/journal/pb/01.10.2.html. Accessed September 20, 2002.

龍安靖　An-Jim Long，專長電子病歷與電子簽章法研究、醫療資訊交換標準、醫院資訊系統、軟體工程、物件導向分析與設計。

劉建財　Chien-tsai Liu，台灣健康資訊交換第七層協定協會理事長，目前致力於 HL7、LOINC 等標準之研擬，本土化與推廣。

張博論　Polun Chang，專長資訊科技應用與評估、機動資訊、社區保健資訊、護理資訊、專家系統、智慧型代理人系統等。

許明輝　Min-Huei Hsu，忠孝醫院神經外科主任，並就讀於台北醫學大學醫學研究所博士班，致力於資訊於加護病房應用研究。

李友專　Yu-Chuan Li，爲台灣醫學資訊學會理事長，萬芳醫院副院長，目前致力於病患安全與知識擷取之研究。

簡文山　Wen-Shan Jian，目前任教於台北醫學大學醫務管理系所，，並就讀於國立陽明大學公共衛生研究所博士班。

劉碩琦　Shuo-Chi Liu，目前任教於私立元培科學技術學院醫務管理系，並就讀於國立陽明大學公共衛生研究所博士班。