Using Weakly Structured Documents to Fill in a Classical Database

F. LAFOREST, Information Systems Laboratory, France
A. FLORY, Information Systems Laboratory, France

Electronic documents have become a universal way of communication due to Web expansion. But using structured information stored in databases is still essential for data coherence management, querying facilities… We thus face a classical problem known as “impedance mismatch” in the database world: two antagonist approaches have to collaborate. Using documents at the end-user interface level provides simplicity and flexibility. But it is possible to take documents as data sources only if helped by a human being: automatic documents analysis systems have a significant error rate. Databases are an alternative as semantics and format of information are strict: queries via SQL provide 100% correct responses. The aim of this work is to provide a system that associates document capture freedom with database storage structure. The system we propose does not intend to be universal. It can be used in specific cases where people usually work with technical documents dedicated to a particular domain. Our examples concern medicine and more explicitly medical records. Computerization has very often been rejected by physicians because it necessitates too much standardization, and form-based user interfaces are not adapted to their daily practice. In this domain, we think that this study provides a viable alternative approach. This system offers freedom to doctors: they would fill in documents with the information they want to store, in a convenient order and in a more free way. We have developed a system that allows users to fill in a database quasi automatically from document paragraphs.

The database used is an already existing database, that can be queried in a classical way for statistical studies or epidemiological purposes. In this system, the document fund and the database containing extractions from documents coexist. Queries are sent to the database, answers include data from the database and references to source documents.

INTRODUCTION

Information capture is an important barrier for end-users’ software acceptance. In domains where the end-user is not compelled to use a computer or is demanding because of activity constraints, classical computerized systems have difficulty being accepted and widely used. Difficulties are more accurate when documents are the paradigm used to manipulate information. One can find many domains of this type: lawyers, doctors… use technical documents to store the information they need. These are domains where computerization is particularly little used. The example of the medical domain is obvious. Doctors are not satisfied by today’s systems and prefer using paper-based medical records. Many trials have been and are still conducted in this field, but success has not completely come. The main barrier concerns information capture speed and facilities compared to computerized systems advantages. Capture forms have been chosen by computer scientists because they have the great advantage to provide important querying capacities, as they are most often easily related to a database. Capture forms do not satisfy physicians as they cannot adapt to each case encountered. Forms impose data to be captured, the order in which to capture, and a strict format for each data.

With the current prevalence of the Web and consequently of electronic documents, the next idea that comes is to use electronic documents as a basis for the system. This idea has the first advantage to remove the mismatch between paper-based documents and capture forms. Information is captured in electronic documents and queries on documents can be made using a dedicated document querying language. To go forward in this idea, we have to notice that one can find three different types of documents:

- Free text documents only contain information and formatting instructions (these instructions are also known as physical structure). These documents are very easy to write, but very difficult to query. Free text analysis is still a research domain, the results are not yet satisfying enough to be used in sensitive domains. Compared to paper-based documents, systems based on free text documents still do not provide enough precision to be widely used.

- Strongly structured documents contain information and
semantics guides (also known as logical structure). These guides are most of the time represented by tags circling information pieces. SGML (ISO, 1986) documents are good examples of such documents. These documents set a structure that the user has to follow. This structure is defined in a Document Type Definition (DTD) which provides tags, composition rules of tags, compulsory tags, attributes for tags. This structure is not as rigorous as forms structure: no format is imposed for data. Moreover, only effectively captured information appears in each document. In forms all fields are present even if not filled for the current case. Queries on strongly structured documents can be made using dedicated query languages like SgmQL (Lemaître, 1998) or UnQL (Buneman, 1996). These languages are currently under the form of prototypes and answers lack precision.

In systems that link strongly structured documents to a database, each information to be stored in the database is tightly tagged so that there is a one-to-one relationship between data in the database and tagged information in documents. The database stores a representation of the tree structure of the document, without any treatment on information pieces: filling a database is thus rather easy but does not provide the same facilities as a real information system database filled through a form: queries are not as precise as queries on atomic data. This approach still does not satisfy end users, as writing such documents is time consuming and constraining. As each stored information piece has to be tightly tagged, the DTD looks much like a disguised form.

- Weakly structured documents may be seen as an intermediate level between free text documents and strongly structured documents. Weakly structured documents use DTDs containing optional tags (compulsory tags are defined only to contain the identification of the document subject e.g. patient and doctor IDs). Most tags delimit paragraphs rather than data. A paragraph contains free text which may include many data. For example, a prescription paragraph contains one sentence in which one can find a medication name, a dose, a frequency and a duration. That is the type of documents our system manages. To query data stored in such documents, we link the DTD to a relational database. In each paragraph one can find information which should belong or not to the database. These documents have a double advantage: (1) to be easily captured and read by the end-user and (2) to be semantically more precise than free text, thus easier to analyze automatically.

In the remainder of this article, we use the medical record for illustrations. Here is a rapid tour of our system. The end-user captures information in a weakly structured document. He tags paragraphs according to a DTD conforming to his application domain. For example, tags can be <prescription>, <past history>, <diagnosis>… Once a document is validated by the end-user, an analyzer extracts data from the tagged paragraphs, builds SQL queries and sends its results to a database. At the opposite of usual document databases, the database contains a classical information system database (tables like “patient” and “prescription”, no “paragraph” nor “keyword” table). For each tagged paragraph the analyzer can extract many data to be stored in many tables. Relationships between paragraphs and the database are stored under the form of patterns to be recognized and links to the database. These patterns are defined a priori by the application designer at the same time as the DTD.

The medical record domain seems a good candidate for this system for many reasons. The first is that medicine is still an art that cannot be strongly structured, so that no form-based user interfaces are adequate. Documents are used in the daily work. For chronic diseases, records rapidly become thick: searching information looks like mining. Doctors are interested in automated systems that could help in information retrieval for the care of one patient or for statistical purposes. They also appreciate security improvement (prescriptions validation…). The second reason is that the language used by physicians has already been classified. There exists many classifications, ontologies and thesauri referencing the medical language. This is an advantage for the analyzer, that has referentials to identify data. Classifications of diseases are written for years, medications thesauri can be found in any country, acts are listed in social security files… The third reason is that many paragraphs have standard ways of writing and patterns can be defined for a quite large use. For example, the prescription paragraph should include a medication name (from a thesaurus), a duration (always followed by a duration unit and very often preceded by a word like “during”), a frequency (always followed by a frequency unit) and so on… Each sub-part of the prescription sentence can be represented by a few patterns to search.

This article presents in a deeper way the system we defined. The next section gives position of our work compared to the literature. Then, a system overview presents the differences between a classical documents database and our system and presents answers to the main questions one can think of. Following this, we give the architecture of our system using the UML formalism. A description of the first prototype we have written and an analysis example based on a medical record is given. We then conclude with the current progress state of this project.

RELATED WORKS

Much has been done on documents in the literature. Querying free-text documents is a difficult task that has been studied for years. Recall and precision are difficult to ensure and balance (Salton, 1986). Research activities in this domain are still numerous. A lot of studies are dedicated to a specific application domain, like medicine (Blanquet, 1999), law (Jackson, 1998), or automobile ads (Embley, 1998a). Passage
Related Content

Semantic Heterogeneity in Multidatabase Systems: A Review and a Proposed Meta-Data Structure
www.igi-global.com/article/semantic-heterogeneity-multidatabase-systems/3321?camid=4v1a

A Review of Experiments on Natural Language Interfaces
Hock Chuan Chuan and John Lim (2003). Advanced Topics in Database Research, Volume 2 (pp. 54-71).
www.igi-global.com/chapter/review-experiments-natural-language-interfaces/4341?camid=4v1a

Transformations Between UML Diagrams
www.igi-global.com/article/transformations-between-uml-diagrams/3298?camid=4v1a

Multi-Level Modeling of Web Service Compositions with Transactional Properties
www.igi-global.com/chapter/multi-level-modeling-web-service/74392?camid=4v1a