Bridging Encounter Forms and Electronic Medical Record Databases: Annotation, Mapping, and Integration

Yuan An
iSchool at Drexel
yan@ischool.drexel.edu

Ritu Khare
Drexel School of Medicine
Ritu.Khare@drexelmed.edu

Xiaohua Hu
iSchool at Drexel
thu@ischool.drexel.edu

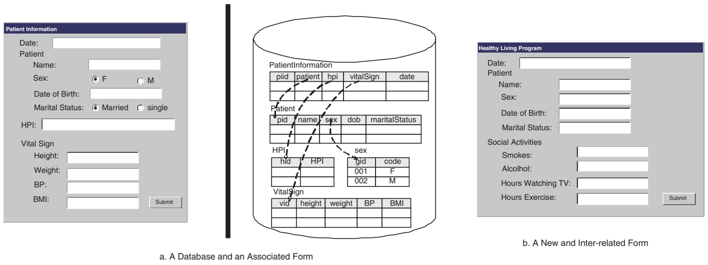
Il-Yeol Song
iSchool at Drexel
isong@ischool.drexel.edu

***Abstract*—Forms are a major source of input for getting data into the underlying medical databases of electronic health/medical record (EHR/EMR) systems. Standardizing encounter forms and integrating data collected from different forms into a single database would greatly reduce heterogeneity. In this paper, we describe a framework, the `fEHR-plus` system, that annotates, maps, and integrates user-specified encounter forms into a single database. The development of the framework incorporates machine learning, standard medical terminology, and the principles of database design. We conduct an empirical study with 52 forms collected from 6 medical institutions for evaluating the performance of the fEHR-plus system. The overall results show that the system is promising towards improving interoperability among electronic health record systems.**

I. Introduction

Forms are a major source of input for getting data into the underlying medical databases of electronic health/medical record (EHR/EMR) systems. The elements on different encounter forms are often specified by different users who may use different terms for the same medical concept or the same term with different meanings. Such heterogeneous terms are directly associated with the elements in the underlying database schemas and instances. Standardizing encounter forms and integrating data collected from different forms into a single database would greatly reduce heterogeneity. In this paper, we describe a framework, the `fEHR-plus` system, that annotates, maps, and integrates user-specified encounter forms into a single database. The following example illustrates the integration process in an EHR system.

Example 1. *Figure 1a shows an EHR application comprising a form and an associated back-end database. The application maintains a mapping between the form and the database. Suppose a new form as in Figure 1b, reflecting a new data collection need, is proposed. A technical developer would first link the Name, Sex, Date of Birth, and Marital Status items on the form to the existing Patient table in the database. She would then extend the existing database properly to collect the new data items under the Social Activities group on the new form. Materialization of the integration process entails: (i) the building of new forms, wherein a technical developer collaborates with the domain experts (i.e., the clinicians) in order to understand the new needs; (ii) the integration of new forms over the existing back-end database, wherein a technical developer directly accesses the database system; studies the existing, possibly complex, schema; and writes the appropriate application code.* ■



a. A Database and an Associated Form

b. A New and Inter-related Form

Figure 1. A Typical Integration Situation

In this paper, we describe a highly automated system that assists a technical developer in the process of integrating a new form into an existing database (as illustrated in Example 1). There are several challenges in developing such a system. The challenges include: (a) discovering semantic correspondences, (b) standardizing the terms on forms, and (c) resolving term and structure heterogeneity.

We have partially addressed some of the challenges in our previous work [1], [2], [3]. In this paper, we describe an integrated framework, `fEHR-plus` (fEHR stands for **f**lexible **E**lectronic **H**ealth **R**ecord), that extends the previous work. We thoroughly study the performance of the `fEHR-plus` framework by conducting a set of comprehensive experiments. The `fEHR-plus` framework receives a form as input and integrates the form into an existing database through a pipeline of functionality as described in Figure 2. In particular, the system first generates a formal model of the form called *form tree*, then annotates the form terms with appropriate SNOMED CT concepts. Subsequently, the system discovers the correspondences between the form tree elements and the existing database elements, generates a new high-quality database, and finally, merges the new database with the existing database.

The rest of the article is organized in the following manner. Section 2 presents the `fEHR-plus` framework,

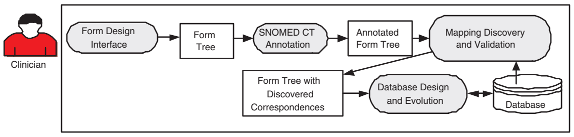

Clinician
Form Design Interface
Form Tree
SNOMED CT Annotation
Annotated Form Tree
Mapping Discovery and Validation
Form Tree with Discovered Correspondences
Database Design and Evolution
Database

Figure 2. The Pipeline of Functionality of fEHR-plus

Section 3 presents the empirical study, Section 4 presents the related work, and Section 5 concludes the paper.

II. The fEHR-plus Framework

A user can design a form through a graphical user interface in the fEHR-plus system. The system automatically converts a form to a tree structure call *form tree* [2].

Form Term Annotation Using SNOMED CT. The system automatically annotates the form terms with a standard medical terminology, SNOMED CT[1]. The key to the proposed annotation is to identify the SNOMED CT semantic category appropriate for a given term. The identification of a term's semantic category requires the knowledge of the *context* in which the term has been specified. The term context can be derived from the formal structure of the form, i.e., the form tree. The implicit relationship between the term context and the desired semantic category can be formally captured into a statistical model. We have devised a machine-learning model that classifies a given term into a semantic category based on the structure of the form tree. In sum, the proposed approach works in the following manner. (1) Determine the SNOMED CT semantic category of a given form term using a structure-based model. (2) Map the term to a linguistically matching concept within the determined semantic category.

Mapping Discovery and Validation. Once the form tree is annotated using SNOMED CT concepts, the next step is to discover the semantically matching elements between the form tree and the existing database. A set of "initial correspondences" between atomic elements in the form and the database are discovered by linguistic or concept matching. After the correspondences are discovered, it becomes a must to further validate them, particularly because a form element may be discovered to correspond to multiple database elements, or a linguistically matching element may not be semantically matching. We have designed a validation algorithm that encodes certain heuristics to validate the discovered correspondences.

Integration of Form and Database. The next step is to physically integrate the annotated form tree into the database based on the validated correspondences. The framework accomplishes this in the following manner. The form tree is translated into an equivalent new database, using a *Birthing* algorithm. Next, the new and the existing databases are merged together based on the correspondences using a *Merging* algorithm.

The birthing algorithm translates a given form tree into an equivalent database. The technical significance of the algorithm is that the evolved database is aligned with the desirable properties of databases [4]. We redefined these properties in terms of the form semantics and presented some high-quality principles such as *correctness*, *completeness*, *normalization*, *compactness*, and some optimization principles [4] such as minimization of NULL values in table columns. We focus on two properties relevant to merging: (1) *compactness*: which stipulates that each form element occurs only once in the database, and that semantically matching elements are merged together in the related database and (2) *optimization*: which stipulates that the mappings should minimize the possibility of having NULL values in a foreign key column or a descriptive column of the database.

The task of merging is to decide whether (i) to keep the matching columns separately in different tables, or (ii) to merge the corresponding columns into the existing table and link the two tables through foreign key references. This decision reflects a trade-off between the compactness and the optimization principles. While the former option violates the compactness principle, the latter is likely to violate the optimization principle. We define the following two terms to establish the trade-off: (1) *Compactness Factor (cf):* a user-configurable value that indicates the weightage to be given to the compactness property and (2) *Null Value Ratio (nvr):* a calculated value that indicates the potential of having NULL values as a result of the merger. In this situation, the nvr denotes the possibility of having NULL values in the columns of the extended table, if the merger does happen; the nvr is defined as the ratio of the number of non-matching columns to the total number of columns in the existing table. Due to the limited space, we do not present other situations in this paper. To summarize, each merger situation involves a trade-off between the compactness and the optimization principles, and the nvr calculation is customized as per the situation.

III. Empirical study

We conduct an empirical study with 52 patient encounter and data-entry forms developed by users from 6 medical institutions. The datasets are described in Table I. The forms from each institution were inter-related, i.e., had overlapping elements. The forms contain 4235 crude terms (or phrases) supplied by the users at the time of form creation. We could manually identify the appropriate SNOMED CT concepts for only 59.17% (i.e., 2506) of all the terms. This mappability metric is shown in the last column of the table.

For training the annotation module, we use cross-validation across the terms belonging to a particular dataset. For the merging algorithm, we arbitrarily set the compactness factor to 0.7. We first designed an experiment to measure the performance of the annotation module. Next, we designed the experiments to evaluate the entire framework.

[1] http://www.ihtsdo.org/

Table I
EXPERIMENT DATASET DESCRIPTIONS

No.	No. of Forms	Avg. Label	Avg. Input	Total Terms	Mappability SNOMED CT
1	3	32.33	49.33	161	75.77
2	6	17.17	33	261	63.98
3	7	16.14	37.29	294	56.80
4	18	47.83	65.22	1603	56.20
5	13	82.61	100.46	1519	59.38
6	5	53	67.4	397	62.21

To test each dataset, we start with an empty database, and incrementally map forms in a particular order to the existing database, thereby evolving the same. We design 3 versions of the mapping experiments by altering the correspondence discovery mechanism: (i) the **linguistic discovery** version, wherein correspondences are discovered by matching the linguistic properties between form terms and database element names; (ii) the **concept discovery** version, wherein correspondences are discovered by matching the concept identifiers of the annotated form terms and annotated database elements; and (iii) the **hybrid discovery** version, wherein correspondences are discovered by performing the concept-based discovery method, and if no matching concept is found, then the linguistic-based discovery method is used.

A. Experiment Results

For the annotation experiments, we calculate precision, i.e., the number of correct annotations over the total number of system annotations; and recall, i.e., the number of correct annotations over the number of gold (manual) annotations. The experiments conducted with the 52 forms resulted into a precision of 0.89 and a recall of 0.76. We find that the semantic structure helped in improving the average precision by 43% and the average recall by 29%, over the approach that does not leverage semantic structure. These results reinforce our earlier finding that the structural knowledge has the ability to improve the overall annotation performance. The dataset size used is almost twice as that used in our earlier work [3], thus implying the scalability of the annotation module.

To measure the performance of the integration process, we measure two aspects of the framework: (i) the compactness of the evolved database, i.e., what percentage of the semantically matching elements were merged together?; (ii) the number of user interventions required to carry out the integration process. Providing a quantitative account of the compactness of a given database, with respect to a given set of forms, was challenging. Given the large scale of both the forms and the databases, a manual analysis of the databases was not possible. We thus created an approximate universal set of various merging situations that consists of the "union" of the situations encountered by the three versions of the experiments during the correspondence discovery stage. With this universal set containing 1,875 situations in all, for each experiment, we categorized every encountered merging situation into one of the three classes: (i) when the situation was turned into an actual merger; (ii) when the situation was turned into duplication of elements; (iii) when the situation remained undetected.

For a given evolved database, we calculate the compactness as the number of mergers over the number of identified merging situations. For the linguistic discovery version, 4 databases had at least 75% compactness. The outlier databases, i.e., 4 and 6, had at least 20% of the form elements duplicated in the database. The forms contributing to these databases had some peculiar characteristics such as, format diversity, e.g., the column Gender appears a textbox format in one form and as a radiobutton group with options *Male* and *Female* in another form; or had section scattering, e.g., different aspects of the same semantic concept were scattered in different forms leading to a higher value for the null value ratio, and hence rejected mergers. The undetected situations (avg. 18%), represent the ones involving the terms that required SNOMED's rich descriptions for identification, e.g., "O" ("Objective"), "HPI" ("History of Present Illness").

For the concept discovery version, only half of the databases had more than 70% compactness. The main outliers are databases 5 and 6 that had at least 33% undetected situations. These situations represent the unannotated correspondences and hence were never discovered by the concept-based discovery method. The hybrid discovery version performed really well in terms of compactness generating 80% compactness for 4 databases. The exceptions were databases 4 and 6, again due to the peculiar form characteristics. We also measure the extent of annotation of the generated databases, i.e., the number of annotated elements over the total number of elements in the database. We achieved an average annotation of 39% and 43% for the concept discovery and the hybrid discovery methods, respectively.

To give an account of user interventions, we made multiple measurements as summarized in the Table II. We first measured the general impact of the framework in controlling the user interventions. For this, we conducted experiments with and without using the validation algorithm. The third column depicts the percentage reduction in interventions upon using the validation algorithm. For each, the number of screens reduced by at least 50%. Dataset 3 in the concept discovery version is an exception wherein very few validation patterns were encountered.

Next, we made some absolute measurements such as the number of interventions required, and more information about those interventions, such as the number of options presented to the user in those interventions, and the relevance of the interventions. The fourth column shows the average number of interventions required for integrating a form into the existing database. Here, we make two observations: (i) the number of screens generated for the datasets 4 and 5 is relatively larger than the rest. This is because of the

Table II
INTERVENTION RESULTS (OUTLIERS IN BOLD OR ITALICS)

Version	Dataset	Reduced Screens (%)	Avg. Screens	Options Screen	Screen Relevance (%)
Linguistic	1	50	4	2	15.39
	2	77	2	5	42.86
	3	69	2	5	**50.00**
	4	55	*10*	3	39.79
	5	76	*21*	1	**94.18**
	6	62	5	4	32.14
Concept	1	77	1	1	**75**
	2	62	3	1	**68.75**
	3	*18*	5	1	46.87
	4	54	*8*	2	**45.45**
	5	65	*15*	5	73.57
	6	65	4	*9*	42.86
Hybrid	1	52	4	2	15.38
	2	75	3	3	50
	3	57	4	2	29.63
	4	51	*13*	4	43.29
	5	69	*27*	2	86.04
	6	59	8	3	**45**

larger size of these datasets and hence more possibilities of mergers; (ii) the number of screens is greater for the hybrid version. Since this version helped in identifying more merging situations, it required more correspondences to be validated by the user. The next column denotes the average number of options presented in a screen. This varies from 1 through 5 for most cases, which is manageable for any user to process.

The last column denotes the relevance of validation screens presented to the user, i.e., the percentage of screens wherein the user proposed a merger. Across all the experiments, this value followed no fixed pattern. So, we identified the winning method for each dataset. Thus, we could conclude that in most cases, the relevance of the screens is lesser for the hybrid version. This is because it combines the irrelevant correspondences from both the linguistic-based discovery and the concept-based discovery methods. The experiments with dataset 1 led to a very low screen relevance (15%) in at least 2 methods. This denotes the prevalence of linguistically matching and semantically differing correspondences. The screen relevance was particularly higher (94%) for the linguistic version for dataset 5. In these forms, the linguistically matching, and yet semantically differing terms were not very prevalent.

IV. RELATED WORK

Schema integration has been a long-standing problem [5], [6]. In this work, we focus on integrating forms to a single EHR database. The integration task needs to merge both schema and data elements [2]. We also incorporate a semi-automated approach for discovering correspondences [7]. Standardization of clinical data has received a lot of attention in the past. Several existing works [8], [9], [10] address the problem of standardizing the clinical notes written for human processing and understanding using SNOMED CT. The proposed fEHR-plus is based on a context-based annotation method that exploits the semantic structure of forms. Conceptually, this method is similar to *MoST* [11] in that we perform the mapping of clinical meta data as opposed to data. Technically, it differs as the contextual information used by *MoST* is limited to the SNOMED CT semantic categories. Our method relies on the context of the form term; it is similar to the clinical section classification method [12] that assigns standard labels to the sections of clinical notes by exploiting the organizational structure of the clinical documents.

V. FINAL REMARKS

A limitation of this study is that it assumes that the user verified correspondences are 100% valid. In the future, we intend to conduct a user study to verify the assumption. Moreover, we believe that clinical forms are still quite under-explored. The algorithms employed by fEHR-plus can be further improved by leveraging past form mappings, frequency of form usage, domain expertise of the designer (e.g., physician, nurse, patient, data-entry staff, etc), and form category such as encounter form, admission form, data-entry form, etc.).

REFERENCES

[1] R. Khare, Y. An, I.-Y. Song, and X. Hu, "Can clinicians create high-quality databases? a study on a flexible electronic health record (fehr) system," in *Proceedings of 1st ACM International Health Informatics Symposium (IHI)*, 2010.

[2] Y. An, R. Khare, I.-Y. Song, and X. Hu, "Automatically mapping and integrating multiple data entry forms into a database," in *In the proceedings of 30th International Conference on Conceptual Modeling*, 2011.

[3] R. Khare, Y. An, J. J. Li, I.-Y. Song, and X. Hu, "Exploiting semantic structure for mapping user-specified form terms to snomed ct concepts," in *Proceedings of 2nd ACM International Health Informatics Symposium (IHI)*, 2012.

[4] R. Ramakrishnan and M. Gehrke, *Database Management Systems (3rd ed.)*. McGraw Hill, 2002.

[5] C. Batini, M. Lenzerini, and S. B. Navathe, "A comparative analysis of Methodologies for database schema integration," *ACM Computing Surveys*, vol. 18(4), pp. 323–264, 1986.

[6] R. J. Miller, Y. E. Ioannidis, and R. Ramakrishnan, "The Use of Information Capacity in Schema Integration and Translation," in *Proceedings of International Conference on Very Large Data Bases (VLDB)*, 1993.

[7] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *VLDB JOURNAL*, vol. 10, p. 2001, 2001.

[8] S. B. Henry, K. E. Campbell, and W. L. Holzemer, "Representation of nursing terms for the description of patient problems using snomed iii." *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pp. 700–704, 1993.

[9] R. C. B. Jr., J. J. Cimino, and P. D. Clayton, "Mapping clinically useful terminology to a controlled medical vocabulary," in *Proceedings of Annual Symposium of Computing Applications in Medical Care*, 1994, pp. 211–215.

[10] J. Patrick, Y. Wang, and P. Budd, "An automated system for conversion of clinical notes into snomed clinical terminology," in *In Proc. of HKMD-07, volume 68 of CRPIT*, 2007, pp. 219–226.

[11] A. R. Rahil Qamar and, "Most: A system to semantically map clinical model data to snomed-ct," in *In the proceedings of Semantic Mining Conference on SNOMED-CT*, 2006, pp. 38–43.

[12] Y. Li, S. Lipsky Gorman, and N. Elhadad, "Section classification in clinical notes using supervised hidden markov model," in *Proceedings of the 1st ACM International Health Informatics Symposium*, ser. IHI '10, 2010, pp. 744–750.