

Automatically Mapping and Integrating Multiple Data Entry Forms into a Database

Yuan An, Ritu Khare, Il-Yeol Song,
Xiaohua Hu

iSchool at Drexel University, USA
in ER 2011

Motivation: Populating Databases

- Form is a popular way for gathering data into database...

Healthy Living Program

Date:

Patient

Name:

Sex:

Date of Birth:

Marital Status:

Social Activities

Smokes:

Alcohol:

Hours Watching TV:

Hours Exercise:

Motivation: Querying Databases

- Or for querying databases.

Books Search

Keywords

Author

Title

ISBN(s)

Publisher

Subject

Condition

Format

Reader Age

Language

Pub. Date Month Year

Sort Results by:

Motivation: Behind the door

Patient Information

Date:

Patient Name:

Sex: F M

Date of Birth:

Marital Status: Married single

HPI:

Vital Sign

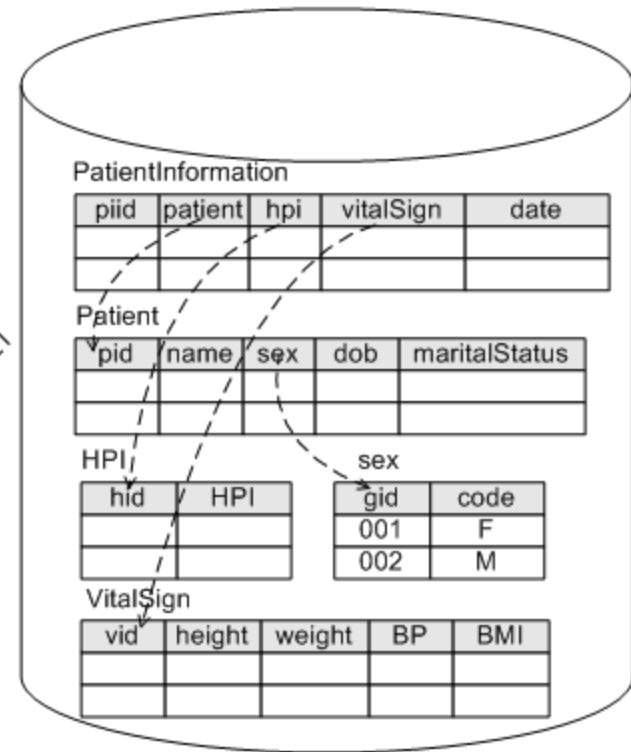
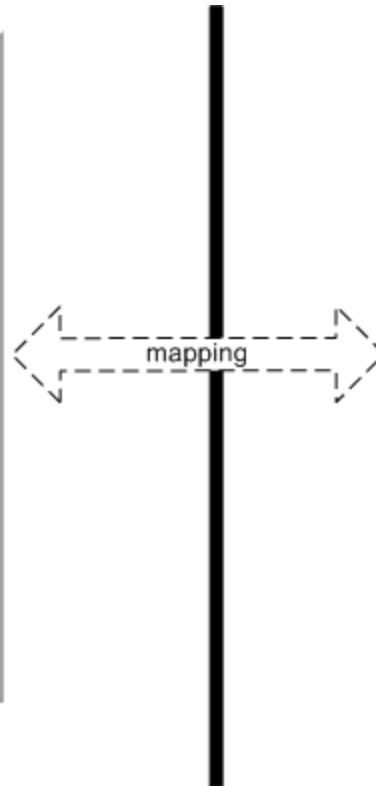
Height:

Weight:

BP:

BMI:

HPI: History of Present Illness
 BP: Blood Pressure
 BMI: Body Mass Index



Motivation: Evolving Requirements and Multiple forms over a Database

Patient Information

Date:

Patient Name:

Sex: F M

Date of Birth:

Marital Status: Married Single

HPI:

Vital Sign

Height:

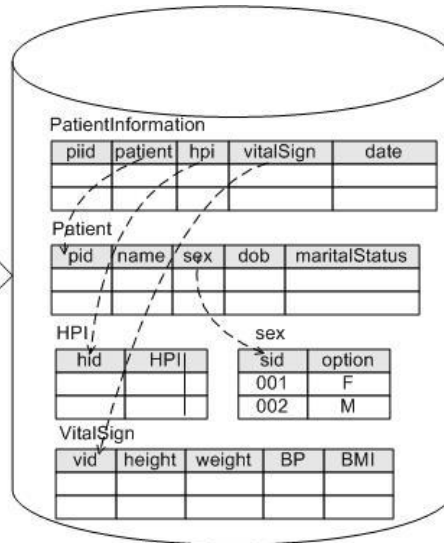
Weight:

BP:

BMI:

HPI: History of Present Illness
 BP: Blood Pressure
 BMI: Body Mass Index

(a)



Healthy Living Program

Date:

Patient

Sex:

Date of Birth:

Marital Status:

Social Activities

Smokes:

Alcohol:

Hours Watching TV:

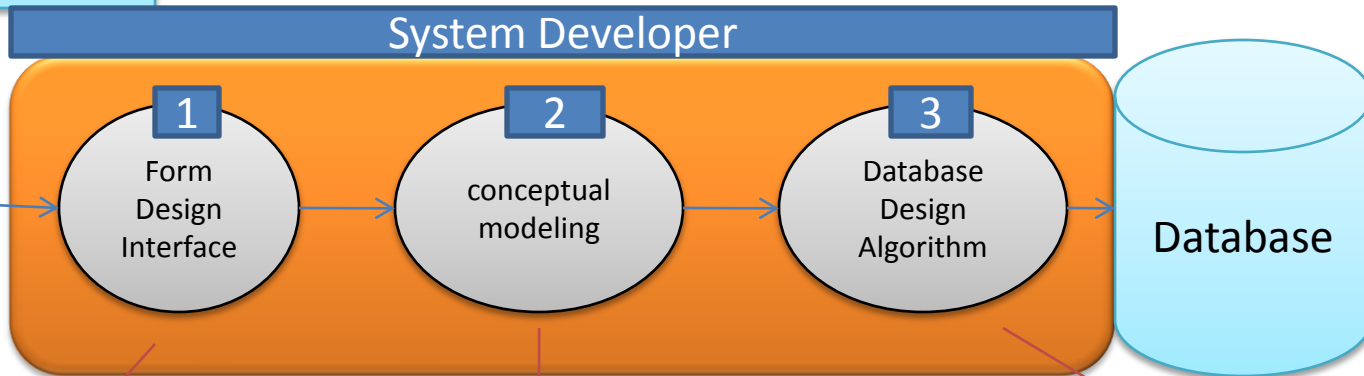
Hours Exercise:

(b)

Motivation: Costly and Inefficient in Building Systems

I want to collect patient's information, personal and vital signs, etc

System Developer



Patient Information

Date:

Patient

Name: Sex: F M

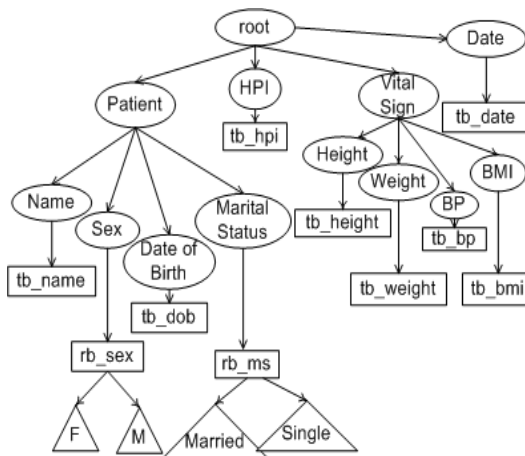
Date of Birth: Marital Status:

HPI:

Vital Sign

Height: Weight:

BP: BMI:



PatientInformation

piid	patient	hpi	vitalSign	date

pid	name	sex	dob	maritalStatus

hid	HPI	gid	code
		001	F
		002	M

vid	height	weight	BP	BMI

Aim: Supporting Non-Technical Users to Use Structured Databases

- Automatically map and integrate user-created forms to structured databases.
- Reduce the barrier of efficient data gathering and analysis for non-technical users.

Patient-Physician Form

Patient

Name

Age

Address

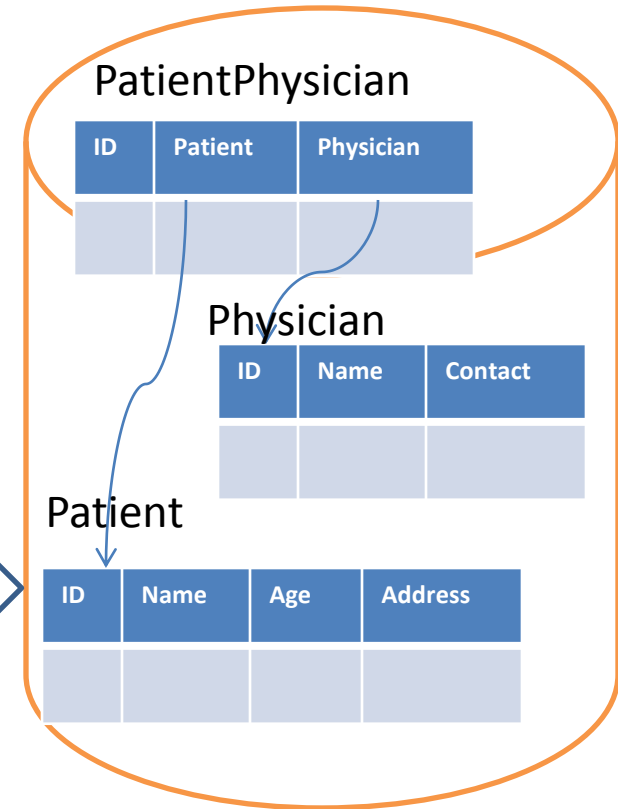
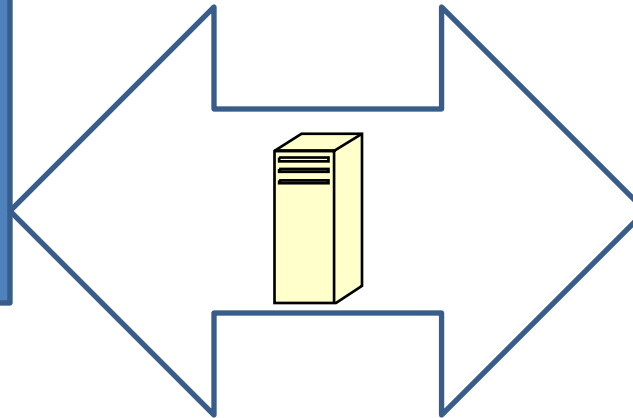
Physician

Name

Contact



user



The *form2db* Problem

- To map and integrate multiple forms into an existing structured database:
 - to allow non-technical users to express their information needs through graphical forms.
 - to automatically extract a formal structure from the user-created forms.
 - to automatically create databases or maps form elements to existing databases.

The FormMapper System

- A fully automatic solution that accepts user-created data entry forms, and maps and integrates them into an existing database:
 - accepts sophisticated forms as input,
 - automatically captures the semantic relationships among form elements,
 - automatically links form elements to the elements in the hidden database,
 - automatically extends the hidden database for unmatched form elements.

Challenges

- As a user interface, a form consists of a collection of form elements :
 - text label, text box, radio buttons, and check boxes
 - organized based on the user's visual preference
 - different form layouts may correspond to the same set of entities and relationships.
- The first challenge is *how to automatically extract a formal structure from a form (or multiple forms)* for creating a database or linking the form elements to an existing database.

Challenges

- The second challenge is *how to automatically link form(s) to existing databases.*

Patient-Physician Form

Patient

Name

Age

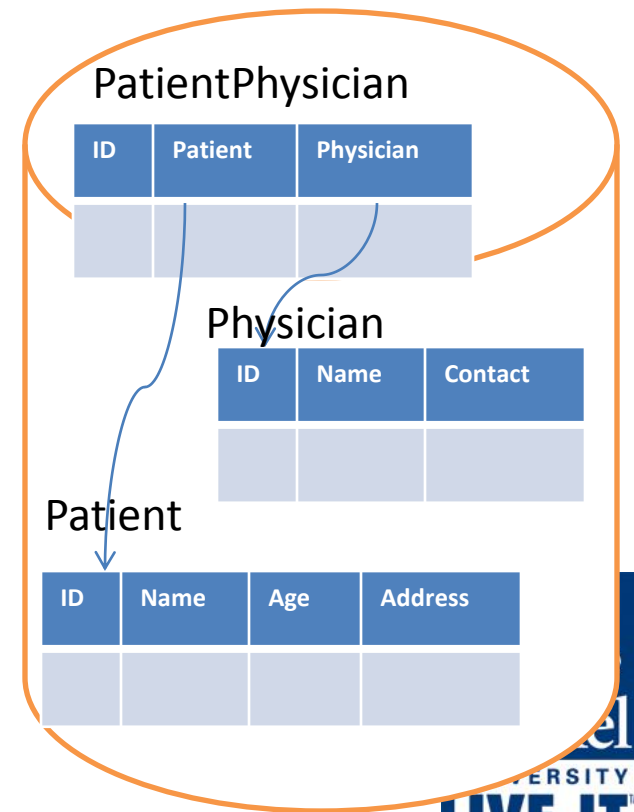
Address

Physician

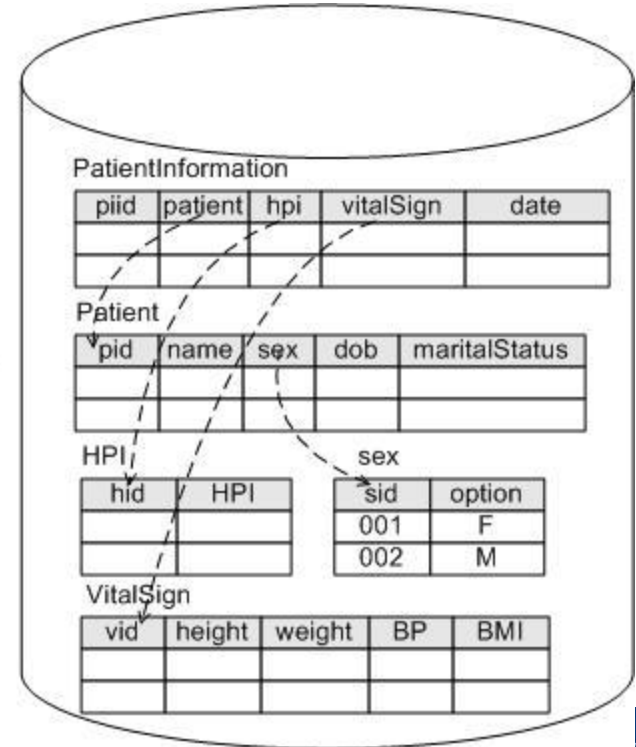
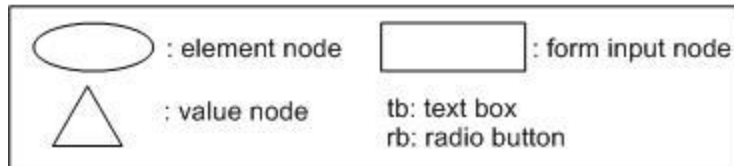
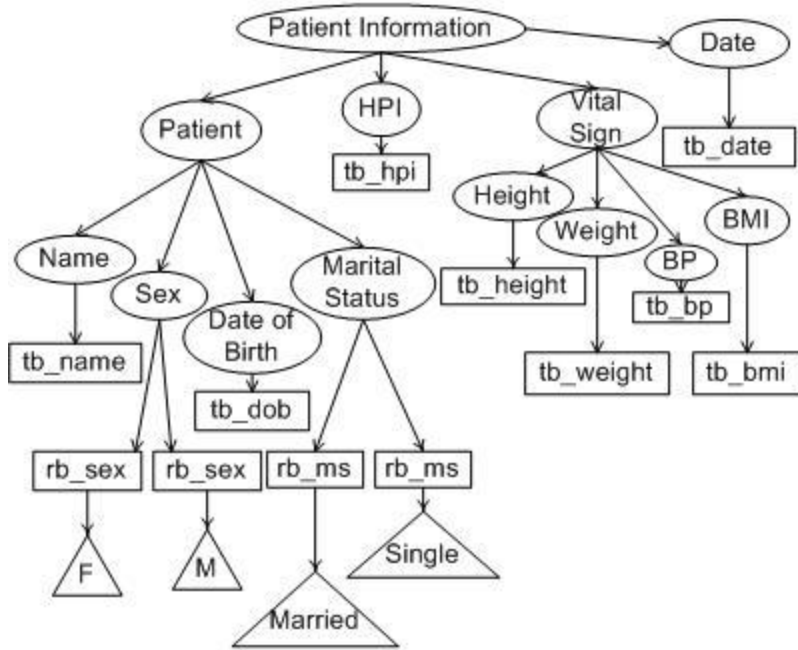
Name

Contact

- Schema mapping, more generally, model mapping is a long-standing problem
- Current solutions for database schema matching and mapping are semi-automatic and require human judgment and intervention.

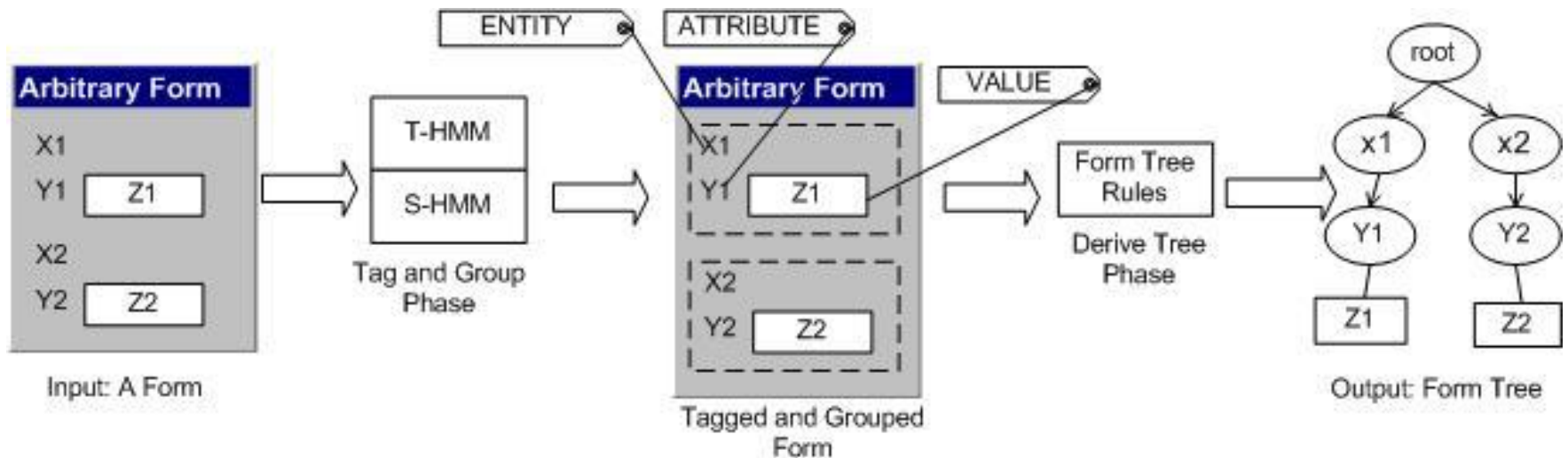


Tree Extraction: Form Tree

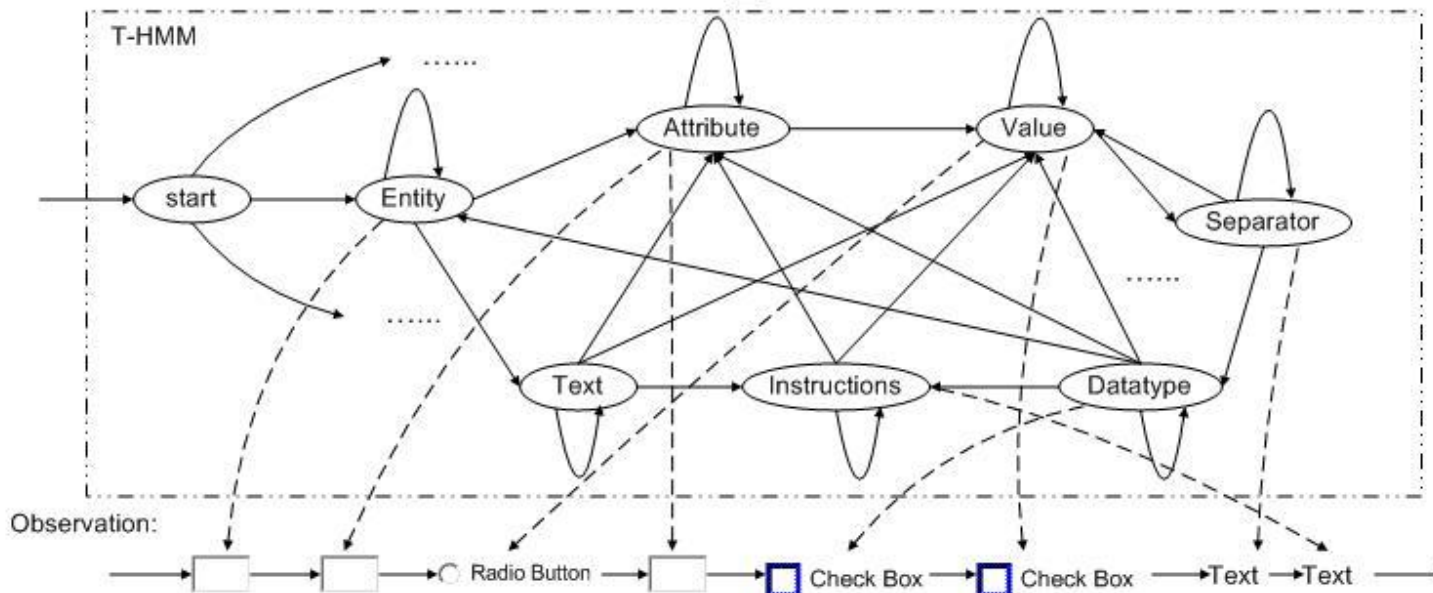
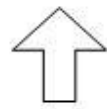
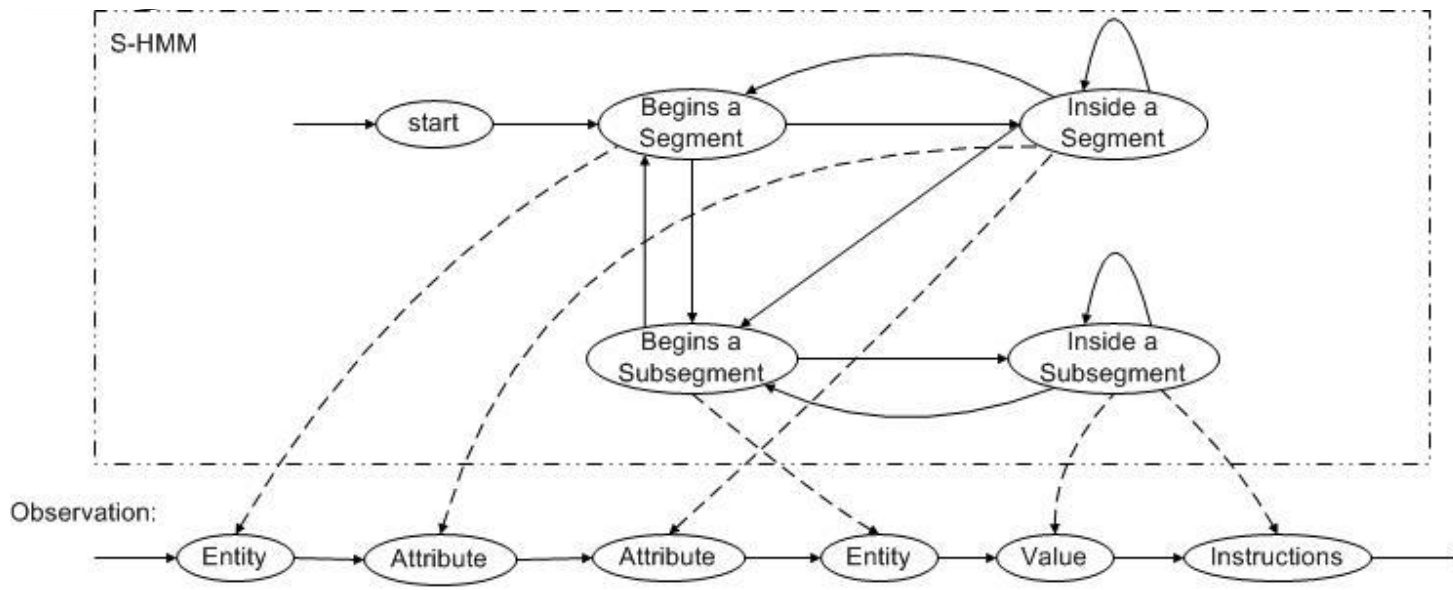


Tree Extraction

- A machine learning technique for automatically extracting a form tree.



Using HMMs: T-HMM and S- HMM



Integrating Forms to Database

- Rely on syntactic matches between the elements in forms and the database.
 - discover a set of initial element correspondences.
 - derive a new database from the given form tree.
 - clean up the initial correspondences.
 - extend the existing database for the unmapped elements.

Birthing Algorithm: Issues in complex forms

Health Information

Date:

Health Status
(Please enter accurate information)

Heart Rate*: bpm

Blood Pressure before Exercise:
Systolic: Diastolic:

Health Problems
Diagnosed (Elaborate in the given space)

Obesity

Depression

None

Symptoms Concerned:

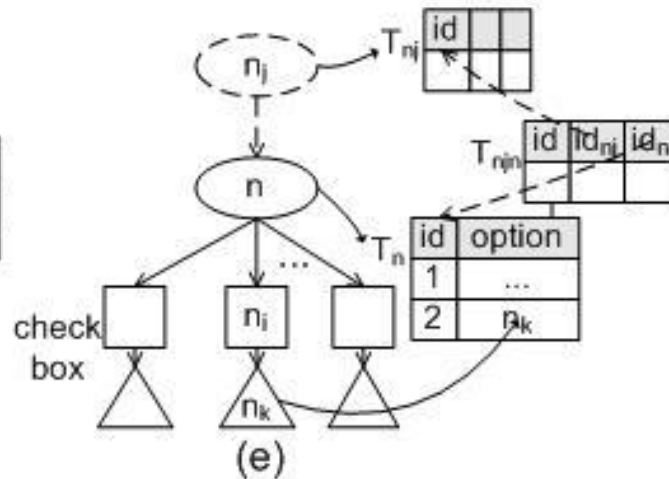
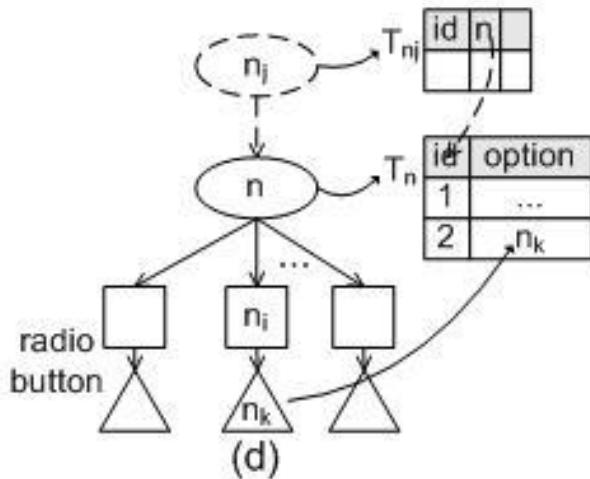
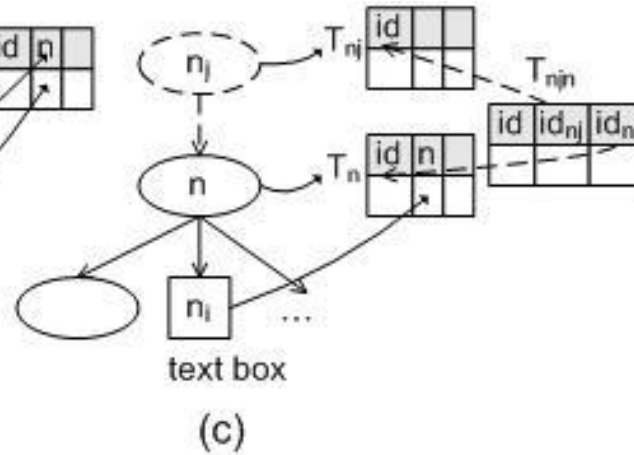
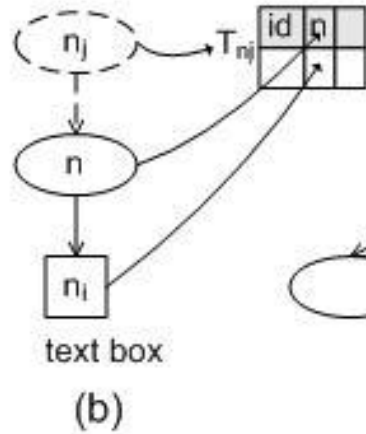
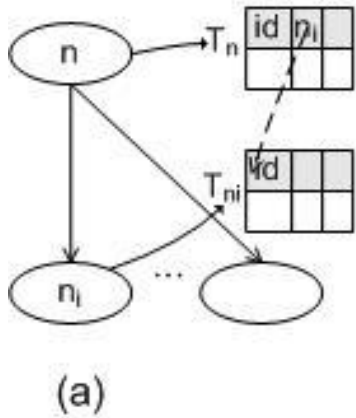
Do you smoke?
 Yes No

If yes, how many times a week?

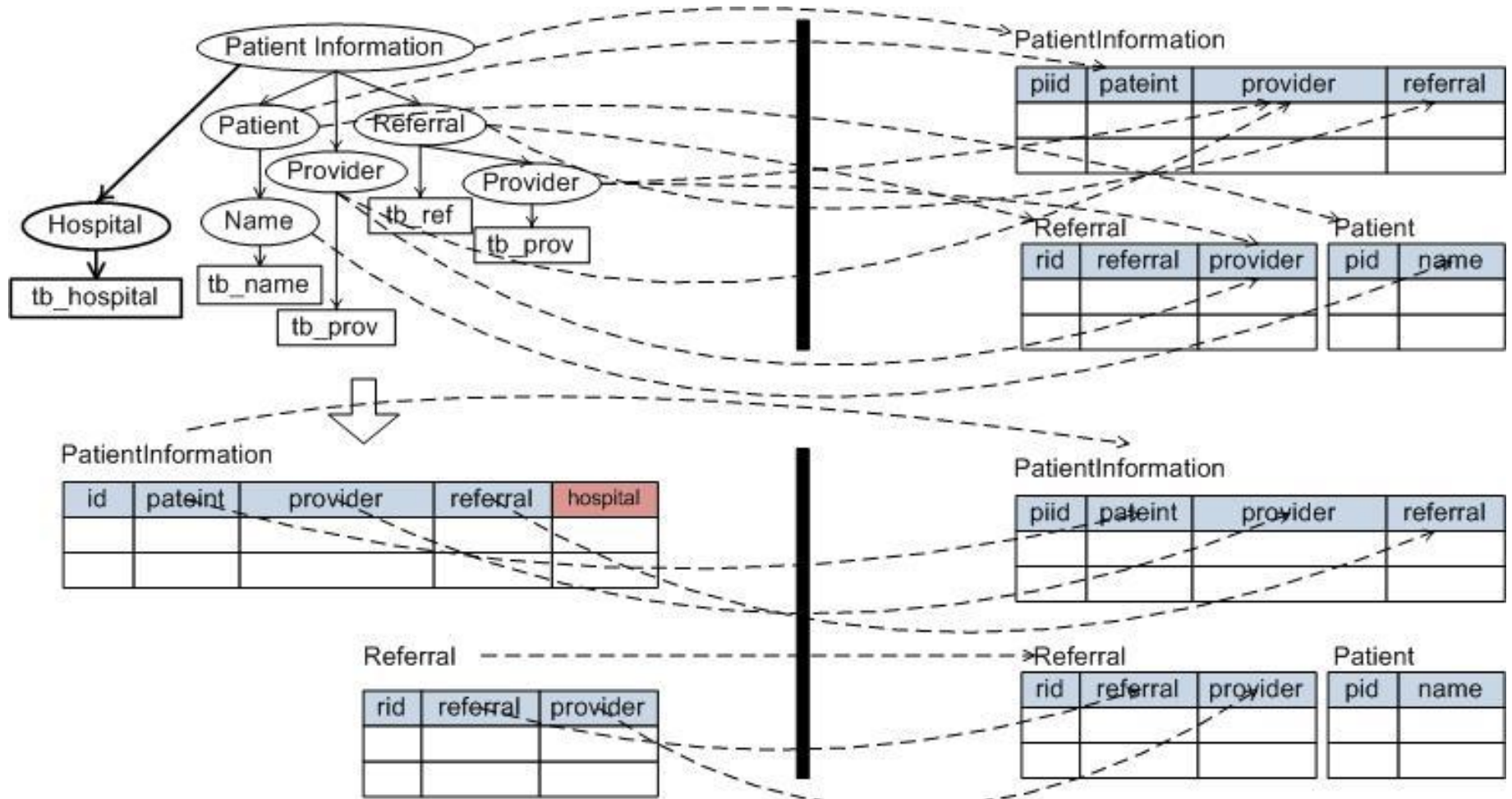
Birthing Algorithm: Requirements

- Requirement 1:
 - All the user-specified elements and values should be captured in the database.
- Requirement 2:
 - All the hierarchical relationships among logical and value elements should be maintained in the database.
- Requirement 3:
 - Requirements 1 and 2 considered, tables should be merged for query efficiency.

Birthing Algorithm: Patterns



Merging Algorithm



Merging Algorithm: optimization

- Adding a column to an existing table
 - a lot of NULL values to the table.
- Creating a new table for new columns
 - the number of tables is increased.
- To balance the trade-off between reducing the number of tables (joins) and reducing NULL values
 - a user-defined quality tuning factor qf : ($0 \leq qf \leq 1$).
 - $qf = 0$ indicates a high preference to reducing NULL values,
 - $qf = 1$ indicates a high preference to reducing the number of tables.

Merging Algorithm: optimization

- Compare qf with a numeric metric null value ratio (nvr) to make decision:
 - $nvr = ((m-h)+(n-h)) / (n+(m-h))$
 - m and n are the numbers of columns of tables; h is the number of shared columns.
 - If the nvr is lower than qf , we merge the two tables,
 - otherwise, we create a new table.

Experiments:

HMM

- Data set:
 - 52 encounter forms in the healthcare domain from three healthcare organizations.
 - form sizes vary from 50 elements to 183 elements.
 - manually labeled with semantic tags as training examples.
- HMM training: k-fold cross-validation for training and testing.
- Measurement:
 - extraction accuracy = the percentage of the correctly extracted parent-child relationships.
- Results:
 - average accuracy was 96%.
 - average time for generating a tree structure was within one second.

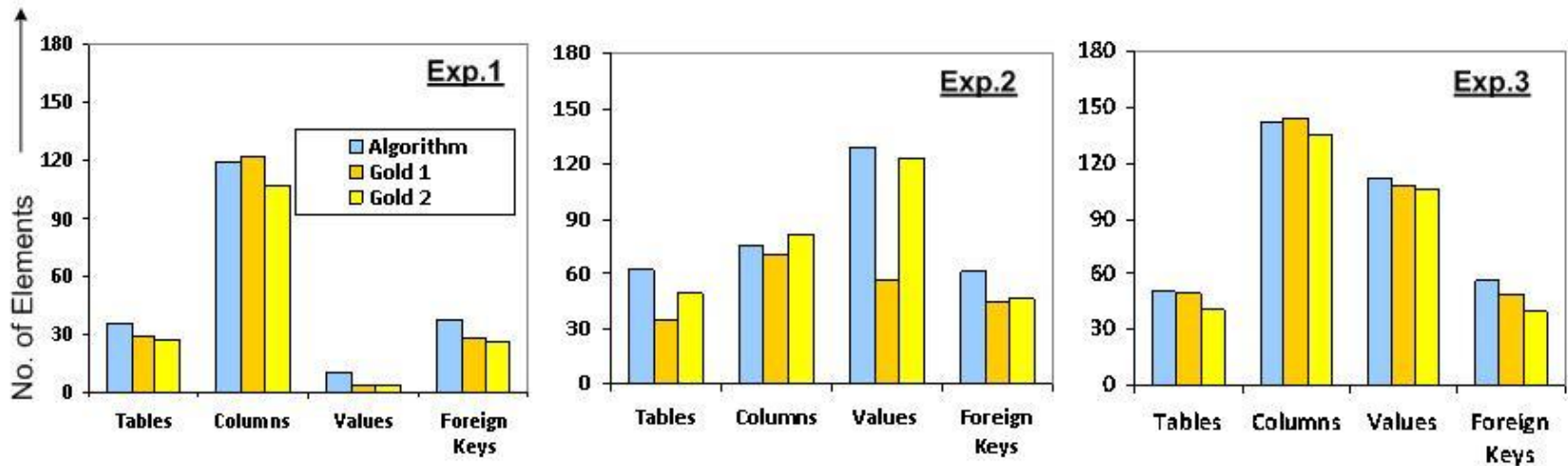
Experiments:

Merging

- Gold standard:
 - databases created by human experts.
- Experimentation:
 - a set of forms is put in a random order.
 - applied the system to them one by one starting with an empty database.
- Results:
 - on an average, 87% of system generated databases are considered to be “matched” to the “gold standard” databases.
 - average time for creating or integrating dbs is
 - 3 seconds

Experiments: Discrepancies

- Missing correspondences.
- Human judgment based on a personal understanding of the domain semantics:
 - for example, many-to-one relationships from the category-subcategory relationships on forms based on personal domain knowledge.



Conclusions and Future Work

- Alleviate the birthing pain.
- With the fully automatic solution, users do not need a clear knowledge of the final structure of a database.
- The structure of the database grows automatically, however, in a principled way with certain characteristics.
- Generate databases with “good” properties by only taking as input the forms – promising and have the potential to replace human developers.
- In future, to exploit a wide variety of information and large-scale sophisticated machine learning models to tackle semantic heterogeneity.