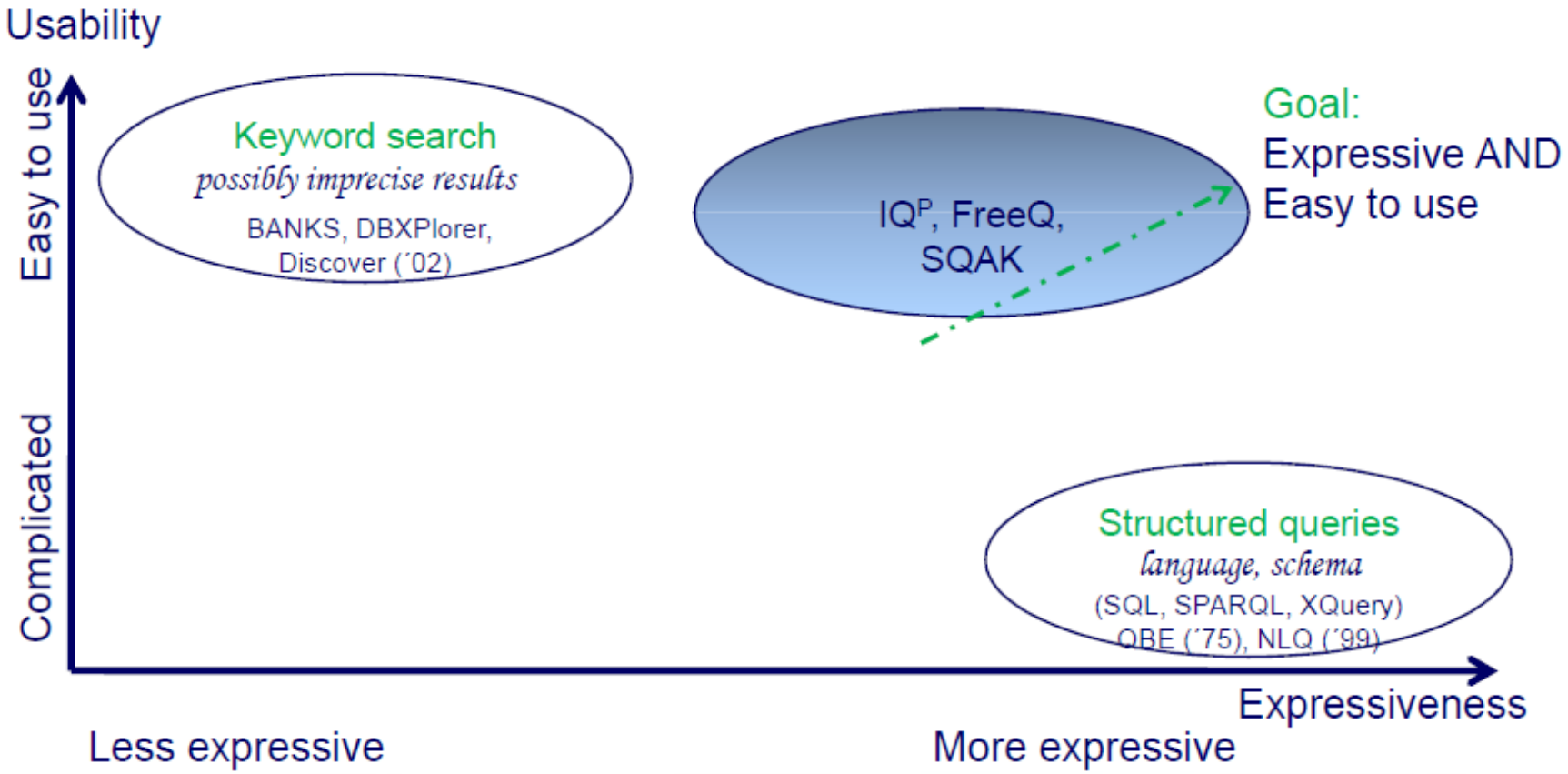
Database Query Approaches to overcome the Tradeoff between Usability and Expressiveness

Simon Gottschalk
Wadim Ortlieb

10/07/2014

Problems and Motivation

Tradeoff between expressiveness and usability



Problems and Motivation

- Users don‘t know
  - database schema
  - query language

- But users are familiar with
  - Google‘s search field
  - forms

Approaches

(1) combine keyword search and form-based interfaces

(2) provide a form that the user can modify

(3) iteratively ask the users questions to find out what information he is searching for

Example database

- Flight
  - flightNumber
  - origin
  - destination
  - duration
  - distance
  - airlineName

- Seat
  - seatNumber
  - flightNumber
  - price
  - fareClass
  - available

- Airline
  - name
  - url

Keyword Search - Example

paris london seats

- Which seats are in which flights
  - Basic Form
  - Aggregation
- Flights with their airline

Flight		Seat	
origin		price	
destination		fareClass	
duration		available	

Skeleton Template Generation

- Generate a set of *skeleton templates* with short descriptions

1) entities without foreign keys
   - "Details about an airline"

$$\sigma_{\text{name } op \text{ value} \wedge \text{url } op \text{ value}} \text{AIRLINE}$$

2) entities referencing other entities
   - "Which seats are in which flights"

$$\sigma_{\text{price } op \text{ value} \wedge \ldots \wedge \text{origin } op \text{ value} \wedge \ldots} (\text{SEAT} \bowtie FLIGHT)$$

3) non-foreign key equi-joins

Form Generation

- generate SQL queries from skeleton templates and map them to forms
- four query classes
  - Simple `SELECT`
  - Aggregation
  - `GROUP BY` (+ Aggregation)
  - `UNION/INTERSECT`

Example of Form Generation

Skeleton template

$$\sigma_{\text{price } op \text{ value} \wedge \ldots \wedge \text{origin } op \text{ value} \wedge \ldots}(\text{SEAT} \bowtie \mathit{FLIGHT})$$

SQL template – query class „Aggregation“

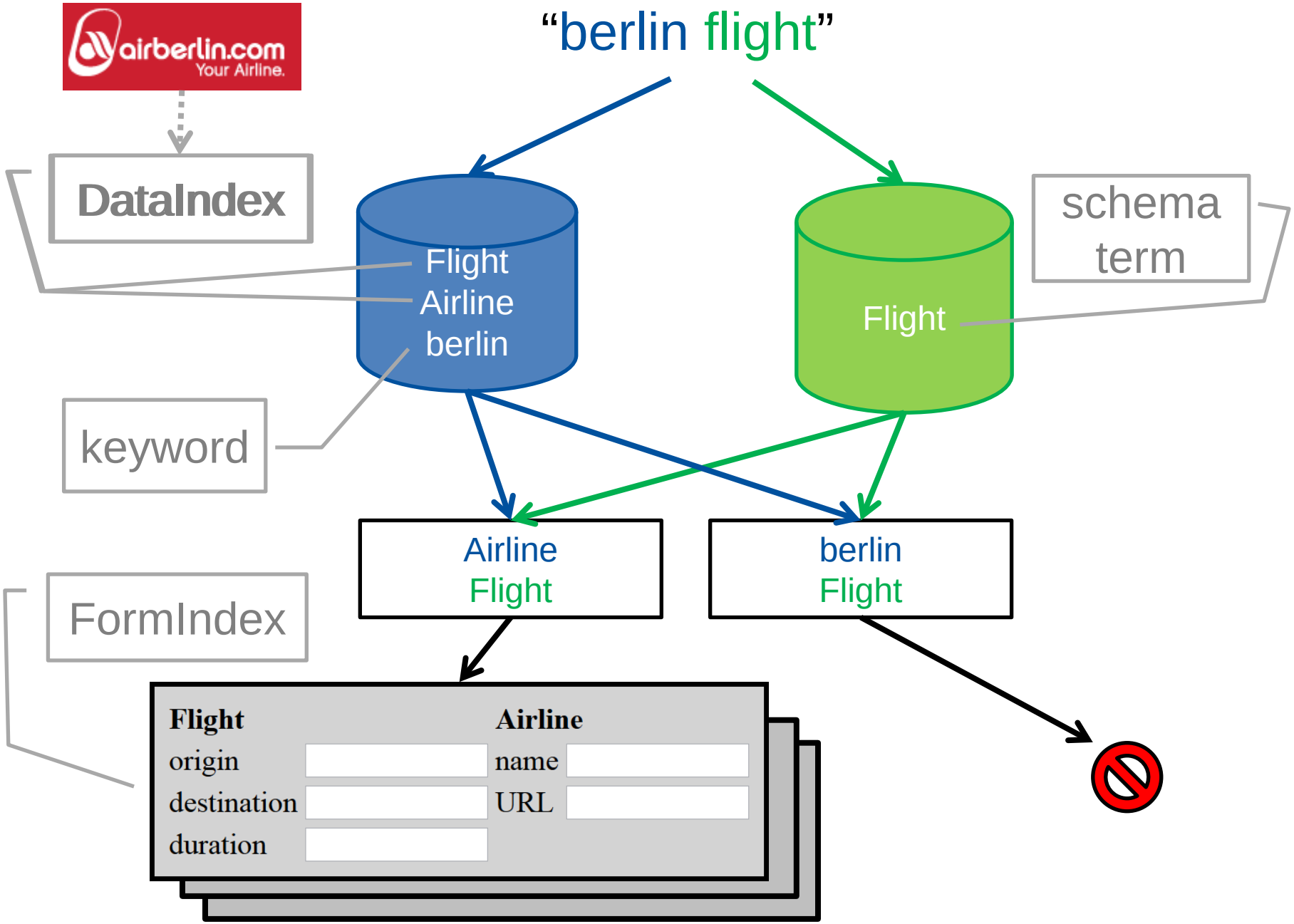
```
SELECT COUNT(*)
FROM   Seat s
       JOIN Flight f USING(flightNumber)
WHERE  f.origin op value AND …
       AND s.price op value AND …
```

Flight		**Seat**	
origin		price	
destination		fareClass	
duration		available	☐

Finding relevant Forms for Keyword Query I

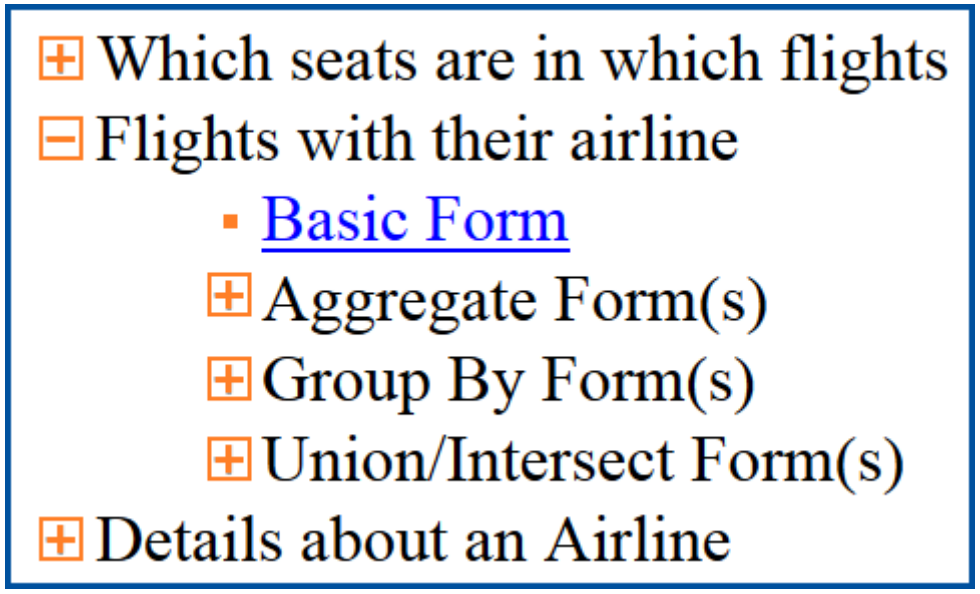
- keyword query terms:
  - *schema term*: term is name of table
    - "flight" → table `Flight`
  - *data term*: term is a value in a table
    - "london" → value in `Flight`.origin

- two inverted indexes
  - DataIndex: term → <tuple-id, table>
    - "london" → <243, `Flight`>, <547, `Flight`>
  - FormIndex: term → form-ids
    - search within the form's descriptions, schema terms, …


airberlin.com
Your Airline.
"berlin flight"
DataIndex
Flight
Airline
berlin
Flight
schema
term
keyword
Airline
Flight
berlin
Flight
FormIndex
Flight
origin
destination
duration
Airline
name
URL

Ranking and Grouping

- Rank forms
  - interprete forms as „documents“ and use well-known TF/IDF scores
- Group forms
  - Problem: forms based on the same skeleton template get the same score
  - Solution: Group them together



Keyword Search - Result

- set of 700 forms
- 7 information needs
- 7 users

} 49 keyword queries

percentage of returned queries (average)	27.8%
ranks of correct form (medians per query)	1 or 2
interaction time (medians per query)	from 24.6 to 106.9 seconds
response time (average)	84.57 ms

Approaches

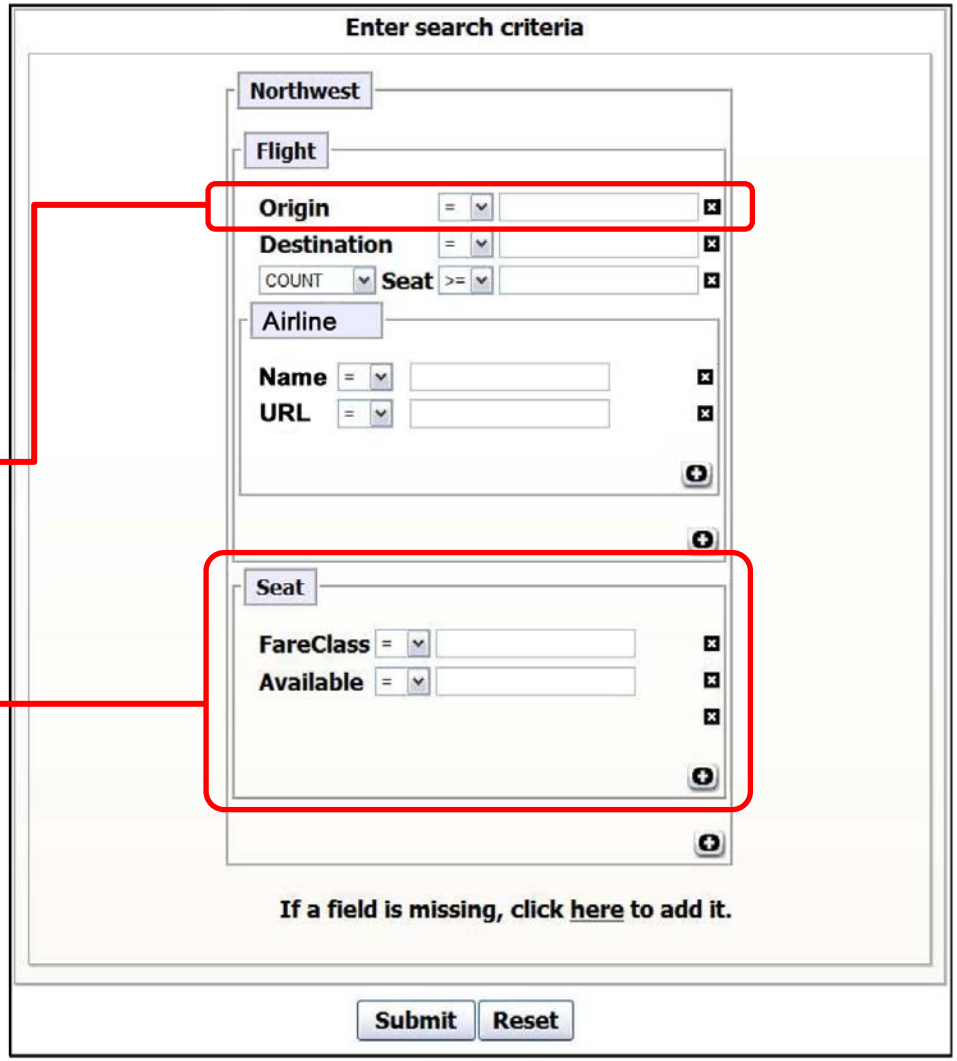
(1) combine keyword search and form-based interfaces

(2) provide a form that the user can modify

(3) iteratively ask the users questions to find out what information he is searching for

Form Definition

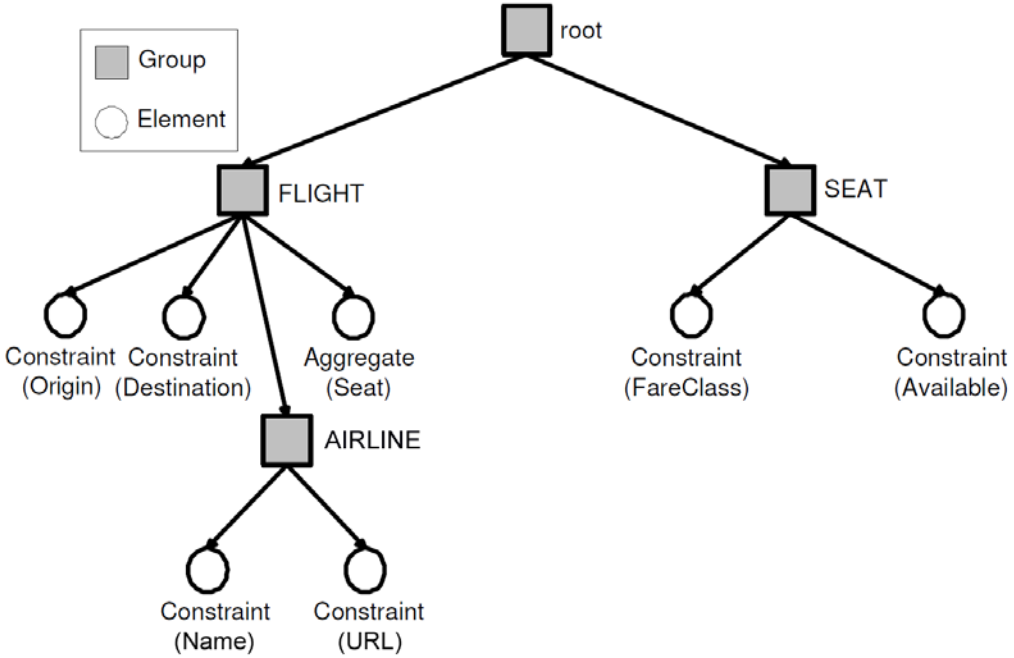
- Form
- Form Element
  - set of form controls
- Form Group
  - set of form-elements



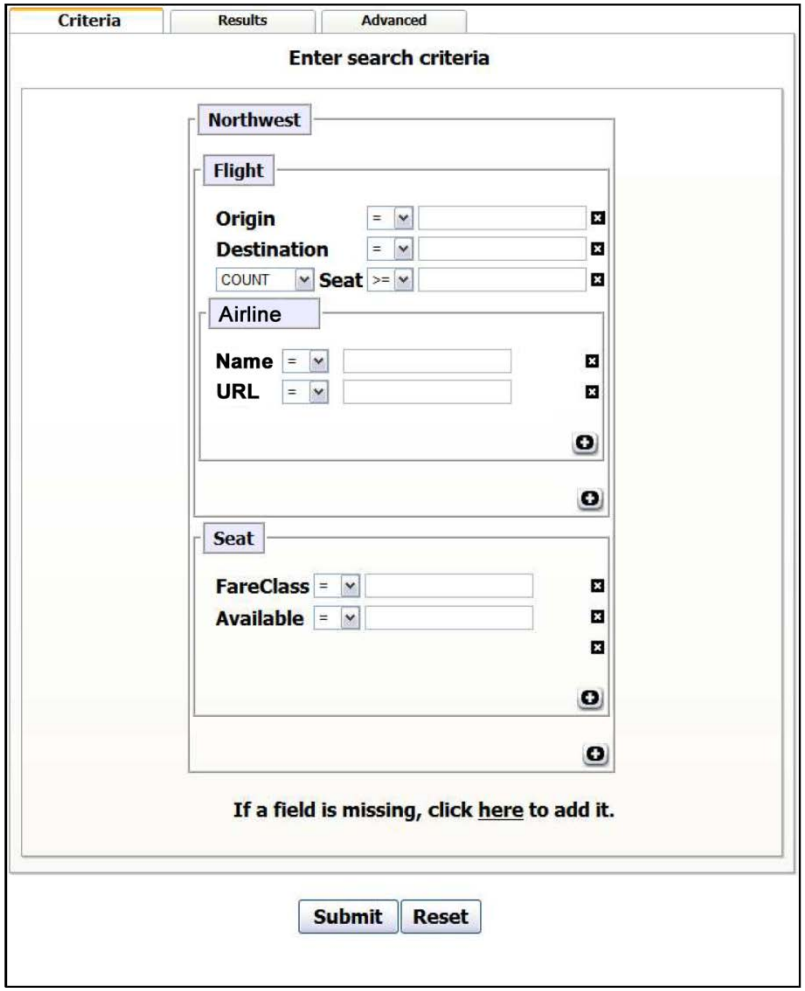
Form Components

- Main components
  - Inputs
  - Outputs
  - Relationships
- Each component can be represented as a tree
- Form as 3-tuple: <IT, OT, RT>
- Modifying a form
  - using predefined set of form modification operators
  - leads to modification of the corresponding tree

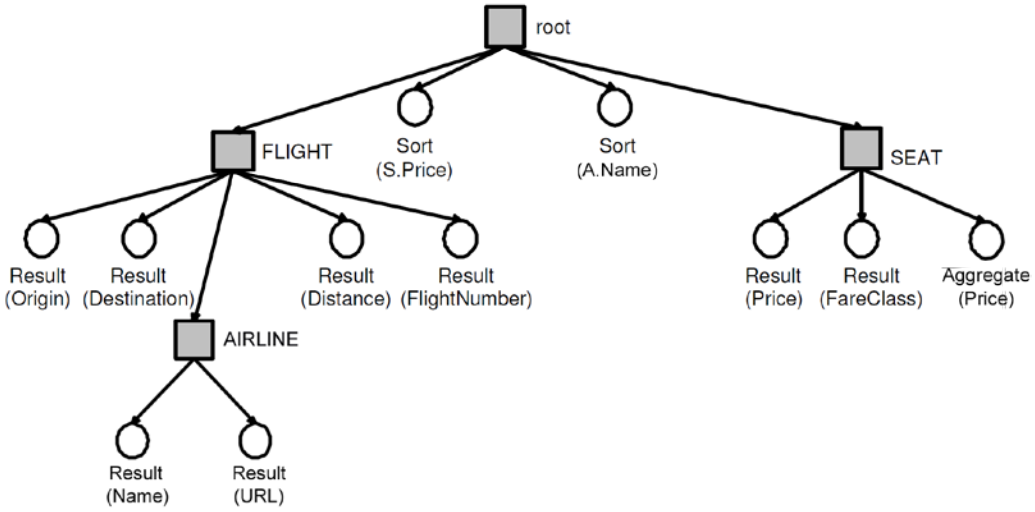
Input Tree



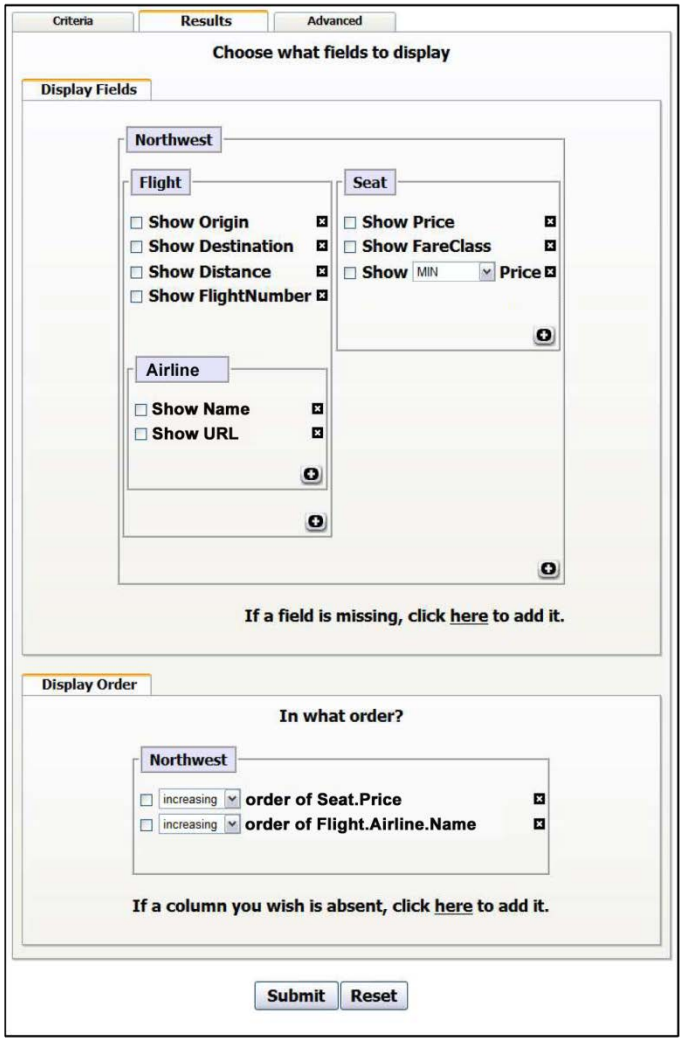
Criteria Pane



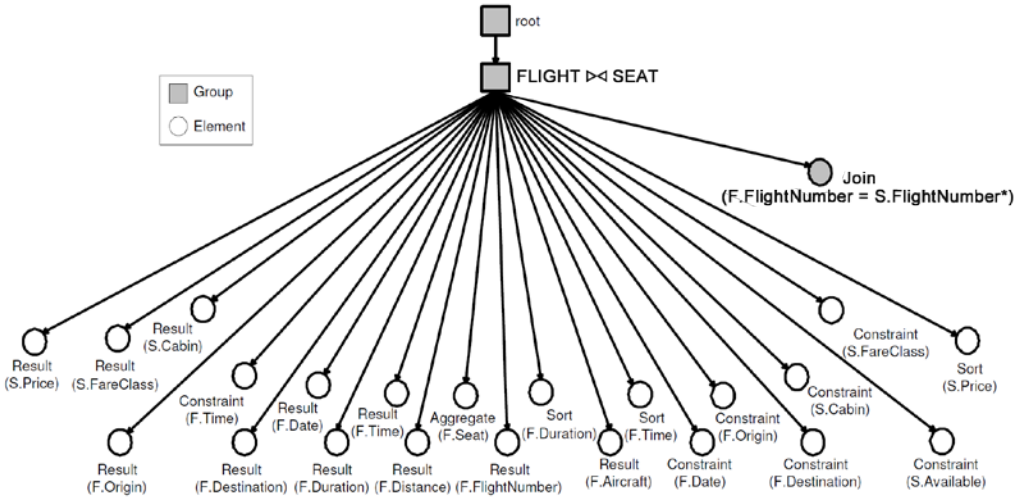
Output Tree



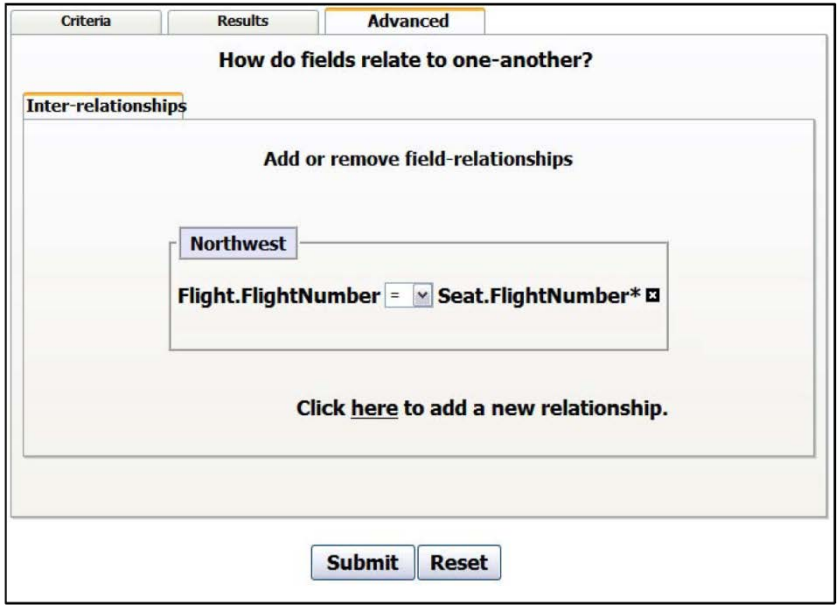
Result Pane



Relation Tree



Advanced Pane



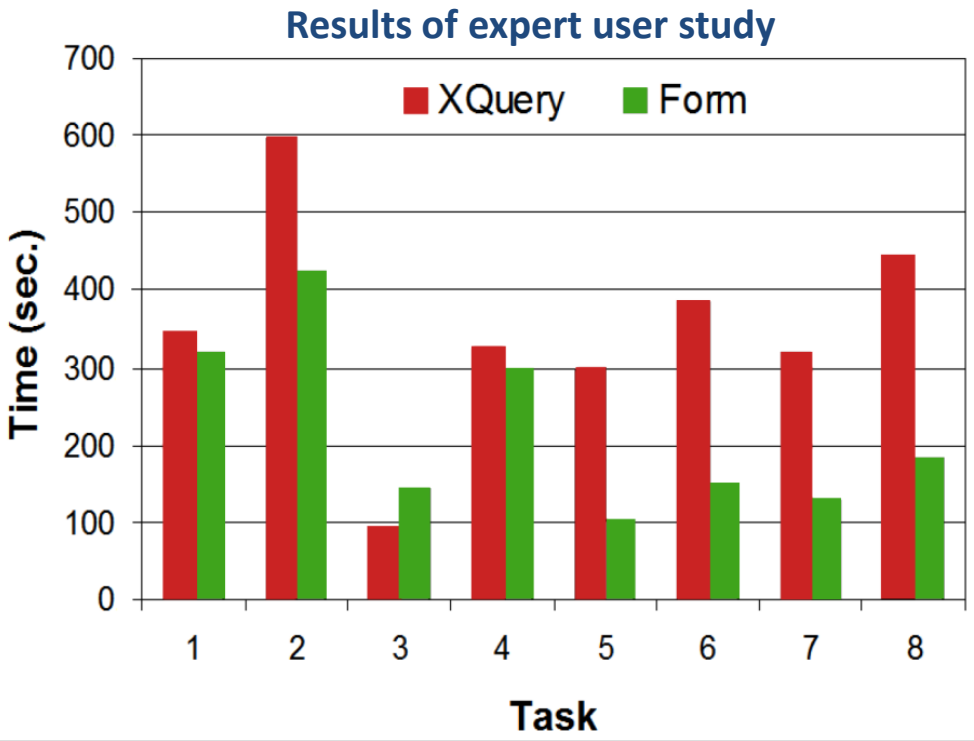
Query Generation

- Translation procedure
  - dynamically formulate the query
  - in contrast to static forms and predefined queries

- Input Tree
  - `WHERE`
- Output Tree
  - `SELECT`
  - `ORDER BY`
- Relation Tree
  - `FROM`
  - `JOIN`

Form Customization - Result

- Usability studies
  - casual and expert users
  - given forms and a set of not fully supported queries
  - measured time taken to solve querying tasks using the interface
- Results
  - customizable forms are effective regardless of queryig expertise



Approaches

(1) combine keyword search and form-based interfaces

(2) provide a form that the user can modify

(3) iteratively ask the user questions to find out what information he is searching for

Interactive Query Construction - Example

- Keyword query: london paris 2 15

- Select the first correct option from the top:
  - ☒ Flight.origin: [london]
  - ☐ Flight.destination: [london]

Interactive Query Construction - Example

- Keyword query: london paris 2 15

- Select the first correct option from the top:
  - ☐ Departure.time: [2, 15]
  - ☒ Flight.duration: [2, 15]
  - ☐ Flight.distance: [15]
  - ☐ Departure.date: [2, 15]
  - ☐ Seat.price: [2]

Comparison

- Expressivity
  - (2) only approach with result ordering
  - (3) no aggregation

(1) keyword search
(2) form customization
(3) incremental query construction

- Usability
  - (1) problems with the overview of the retrieved forms
  - (2) fits best in the workflow (just one form)

- Scalability
  - (1) all the forms have to be created off-line
  - (2) best for mono-thematic databases
  - (3) well scalable using ontologies

Result

- no perfect solution
- decision criteria
  - topic
  - scale
  - wished expressivity
- combined approach
  - start with keyword query and retrieve customizable forms

Summary – Three Approaches

(1) combine keyword search and form-based interfaces
- skeleton templates
- query generation algorithm

(2) provide a form that the user can modify
- form definition: panes, groups, elements
- tree modification: Input, Output and Relation Tree

(3) iteratively ask the users questions to find out what information he is searching for

References

- [Jayapandian et. al 2008] Magesh Jayapandian and H. V. Jagadish. 2008. Expressive query specification through form customization. In Proc. of the EDBT 2008.
  - http://doi.acm.org/10.1145/1353343.135339 5
- [Chu et. al 2009] Eric Chu, Akanksha Baid, Xiaoyong Chai, AnHai Doan, and Jeffrey Naughton. 2009. Combining keyword search and forms for ad hoc querying of databases. In Proc. of the 2009 ACM SIGMOD
  - http://doi.acm.org/10.1145/1559845.1559883
- Elena Demidova. 2014. Refinement of keyword queries over structured data with ontologies and users
  - Lecture Slides for “Advanced Methods of IR”, SS 14
  - Source of the image in slide 2