



# ProFoUnd: Program-analysis-based Form Understanding

**Andreas Savvides**

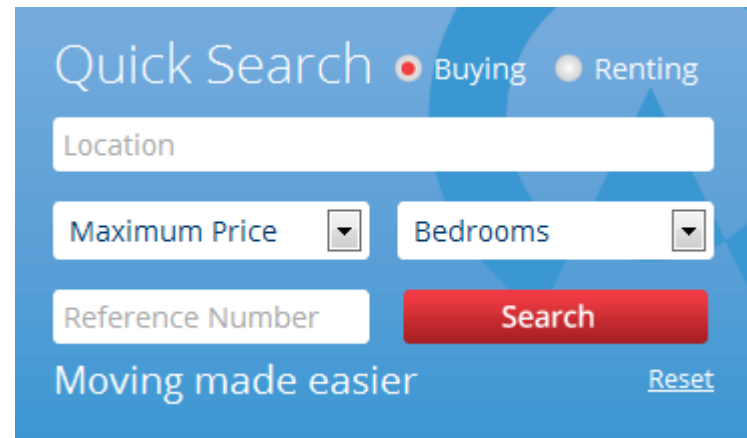
(work done with M. Benedikt,  
T.Furche and P. Senellart)

# Looking deep under the surface

- Surface Web:
  - Bread & butter of **traditional** search engines
  - All about **hyperlinks**
- Deep Web:
  - Search engines **struggle at surfacing** hidden content
  - Plethora of **valuable information** behind Web services and **HTML forms**
  - Valuable data for information extraction systems, e.g. **DIADEM**

# Dealing with a deep Web search interface

1. Find relevant websites
2. Identify a search interface
3. Deduce input fields of the search interface and corresponding meta-data
4. Query a hidden database



Quick Search  Buying  Renting

Location

Maximum Price  Bedrooms

Reference Number

Moving made easier [Reset](#)

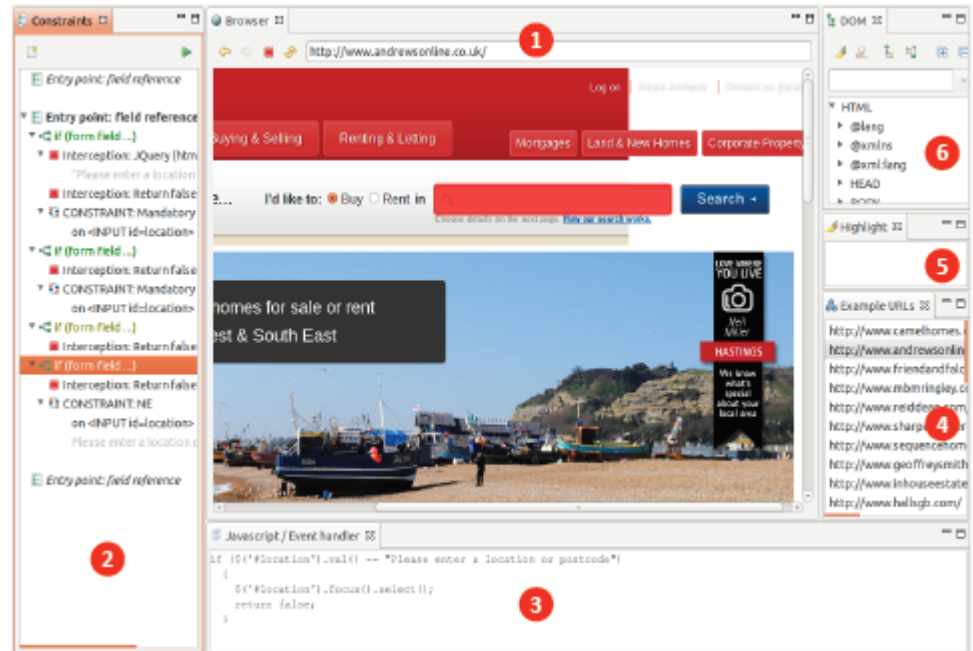
# JavaScript and the Deep Web

- Client-side **integrity constraints** enforced using JavaScript

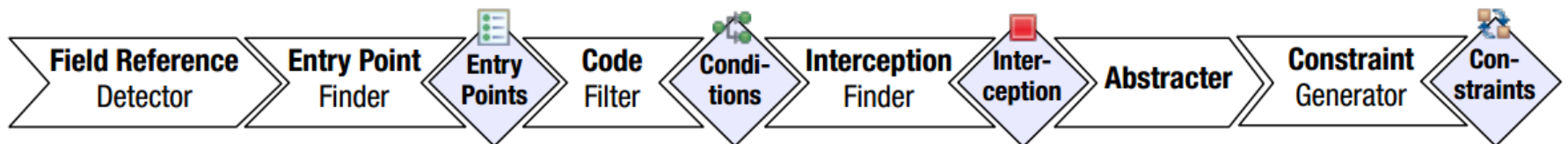
```
// Do not submit unless form is valid
$j("#searchForm").submit(function(event) {
    $j("#searchFormLocationClue").val($j("#searchFormLocationClue").val().trim());
    if ($j("#searchFormBusinessClue").val().isEmpty()) {
        alert('Help us help you\nWe need more information to
            complete your search.\n\n- Please enter a Search Term');
        return false;
    } else {
        return true;
    }
});
```

# ProFoUnd

The first system to provide **integrity constraint identification** for deep Web search interfaces based on **JavaScript analysis**.



ProFoUnd's Interface



ProFoUnd's Architecture

# Evaluation & Moving Forward

- 70 randomly selected real-estate websites
  - 100% **precision** and 63% **recall**
- Where did we fall short?
  - eval, obfuscation, system limitations
- What about...
  - Runtime execution?
  - Ajax?
  - Server-side constraints?

# Demonstration?

Come find us at our demo booth **all day long on Friday** for a showcase of the system or simply to learn more!

The screenshot displays a web browser window with the URL `http://diadem.cs.ox`. The browser content includes a form with the following elements:

- Keyword:
- Price (USD) Minimum:
- Price (USD) Maximum:
- Submit Query button

Overlaid on the browser is a 'Constraints' tool window. A red circle with the number '2' is positioned near the top of this window. The tool lists several constraints:

- Entry point: field reference
- if (form field ...) with Interception: Return false
- CONSTRAINT: LE on <INPUT id=min> and on <INPUT id=max>
- if (form field ...) with Interception: Return false
- CONSTRAINT: NE on <INPUT id=max>
- if (form field ...) with Interception: alert "Please enter a keyword" and Interception: Return false
- CONSTRAINT: NE on <INPUT id=product>

At the bottom of the screenshot is a 'Javascript / Event handler' panel. A red circle with the number '3' is positioned near the bottom right of this panel. The code in this panel is:

```
if (min > max)
{
  $(".label_price").css("color", "red");
  return false;
}
```