

Automatically extracting view models from component-based web applications

Andreas Tielitz, 25.07.2016

Software Engineering for Business Information Systems (sebis)
Department of Informatics
Technische Universität München, Germany

www.matthes.in.tum.de

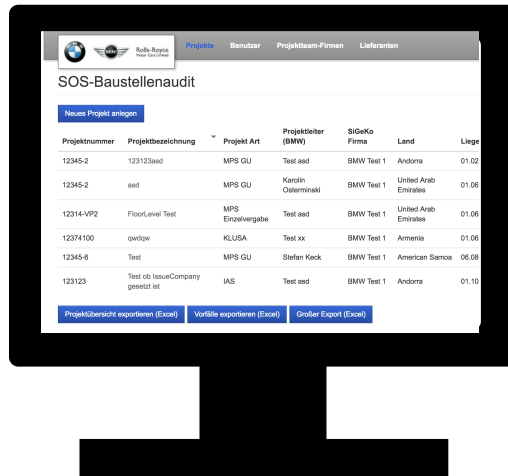
1. Introduction

- Research Questions

2. Proposed Process

3. Process for React

4. Next Steps & Timeline

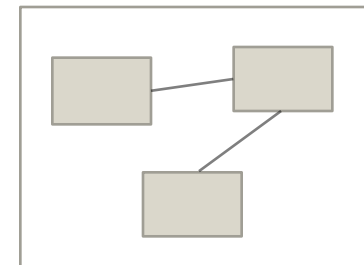
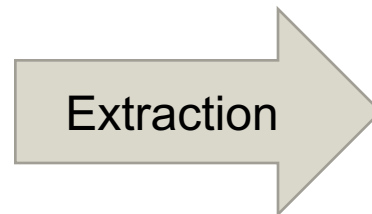


- Existing component based web application
- No documentation
- Migration / Reengineering

- Retrospectively extracting a model from the source code



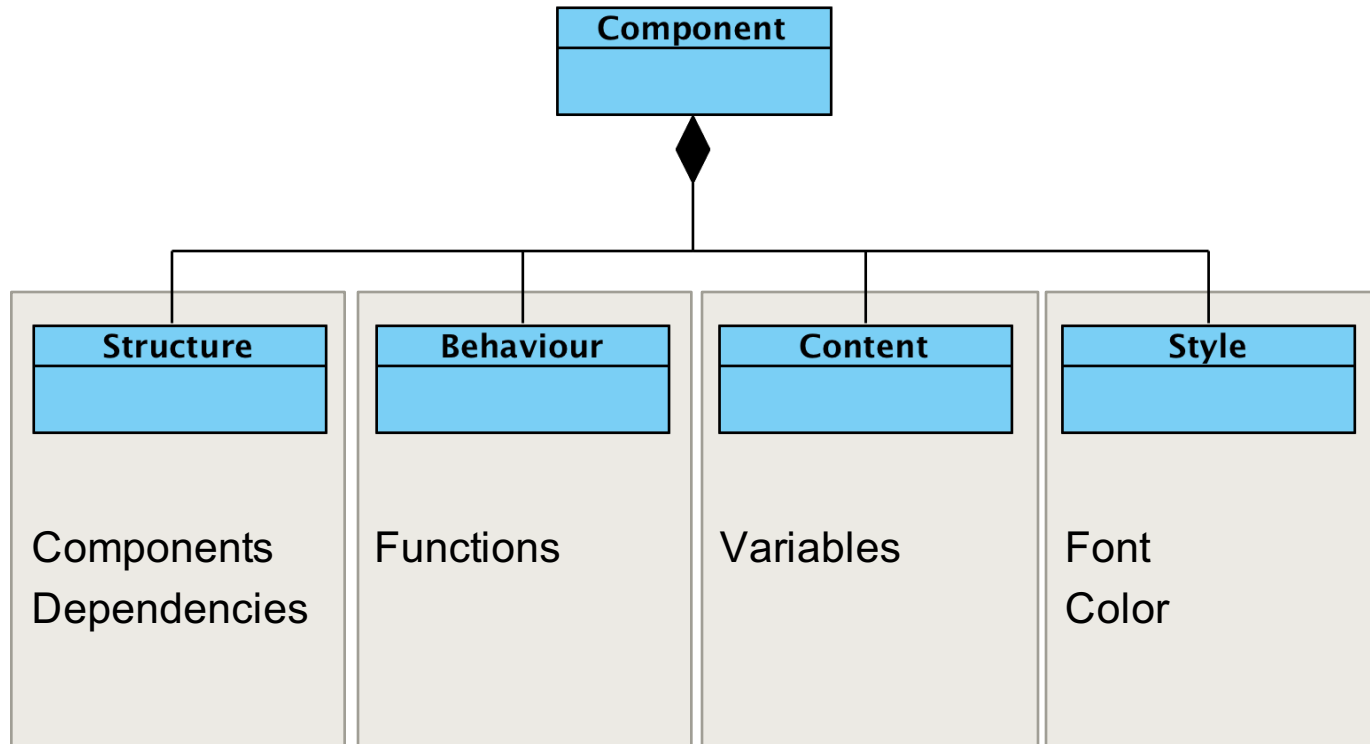
Source Code



View Model

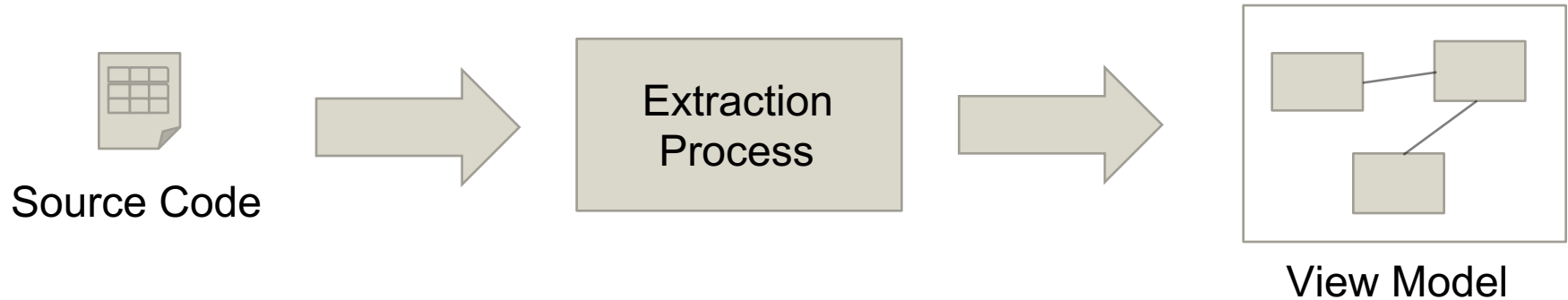
1. What is a view model of a web component?
2. How can a model be extracted from existing web components code snippets?
3. How does existing technology support the model extraction process?

What is a View Model of a Web Component

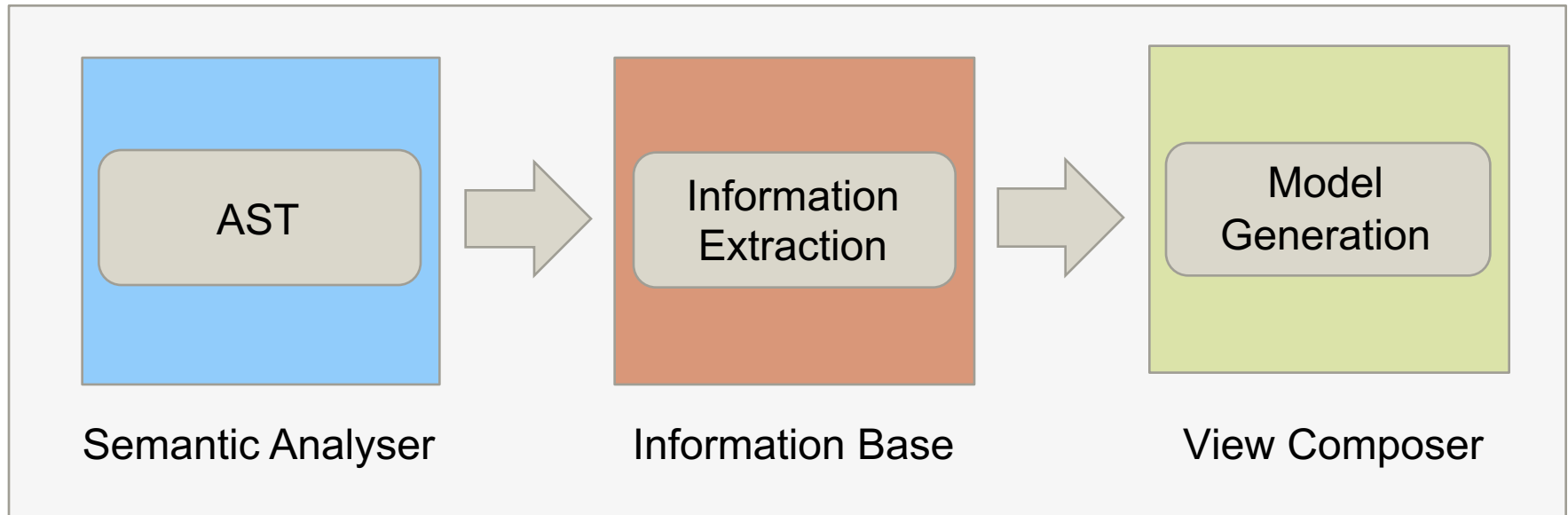


Adapted from UIML 4.0 specification, <http://docs.oasis-open.org/uiml/v4.0/cd01/uiml-4.0-cd01.pdf>, 2008

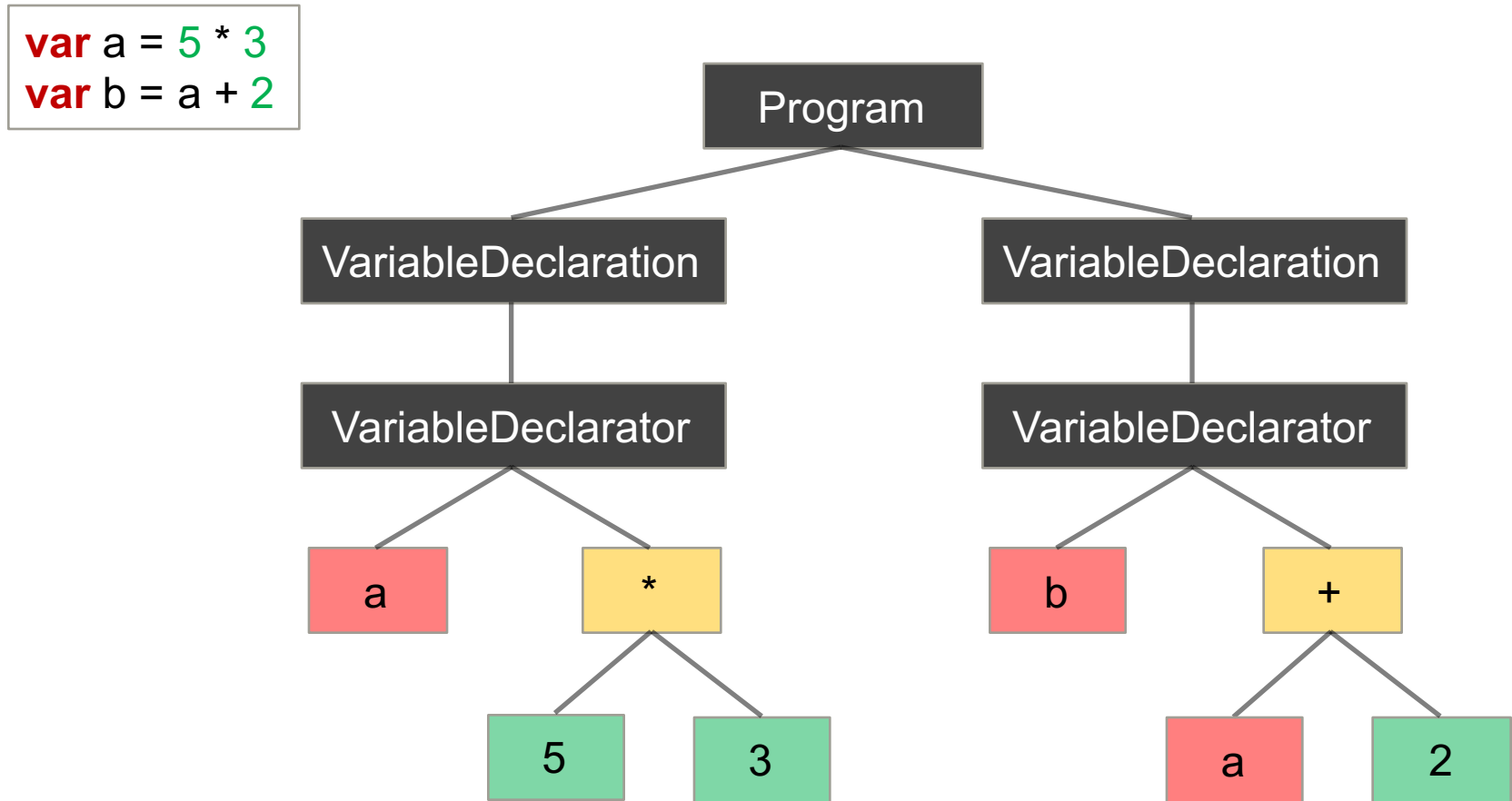
How can a Model be extracted?



Extraction Process



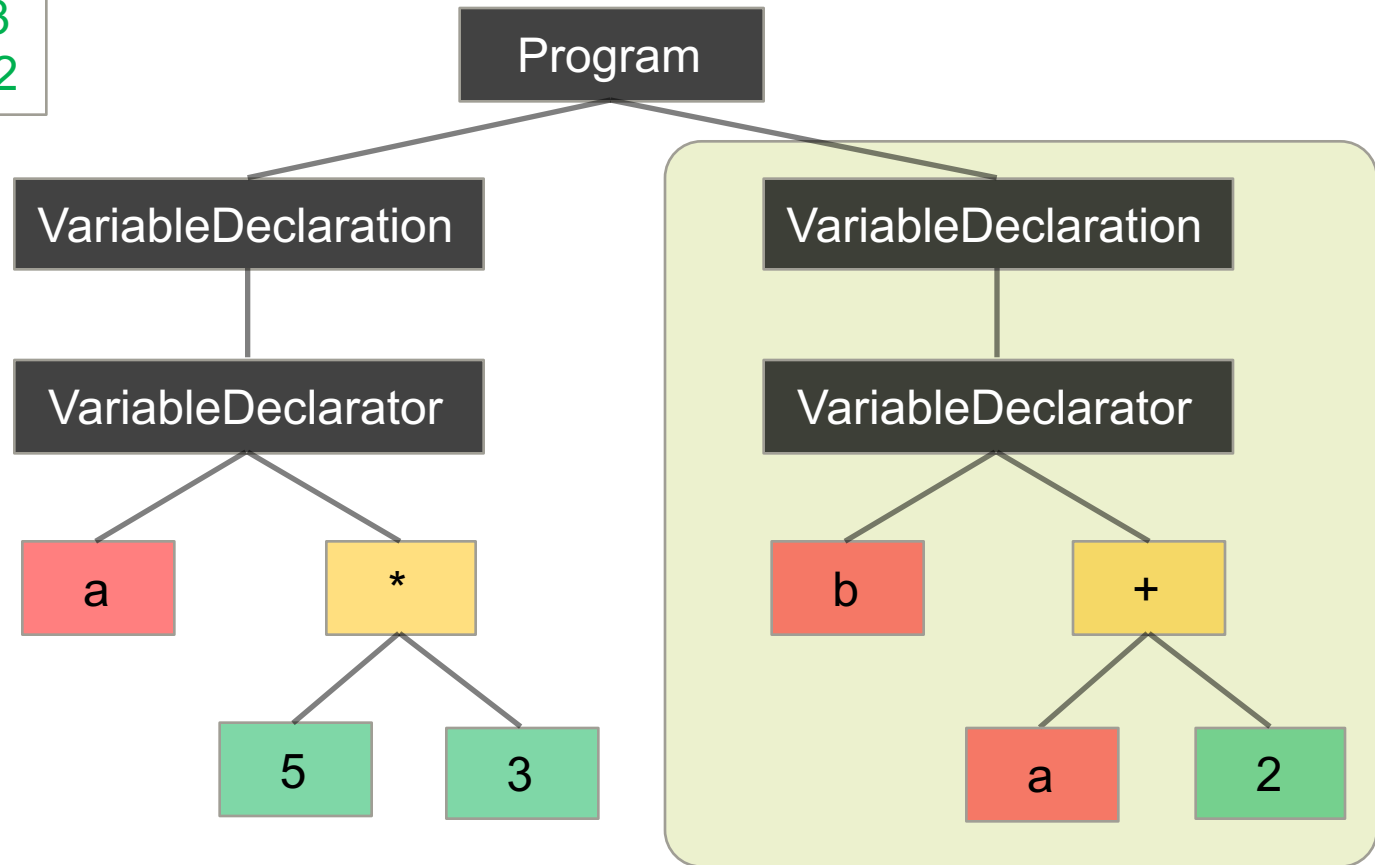
- Tree that represents structure of source code





- Challenges
 - What are the components?
 - Which functions are declared in each component?
 - Does this component depend on other components?
 - ...
- **Information is contained in AST**
- Use rules to describe extraction process

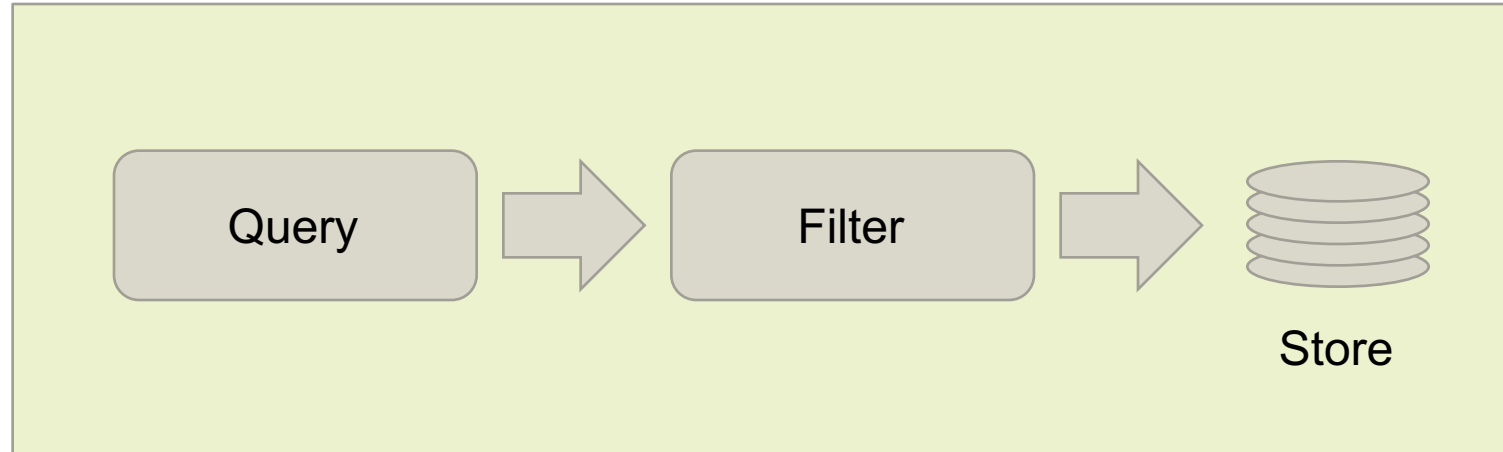
```
var a = 5 * 3  
var b = a + 2
```

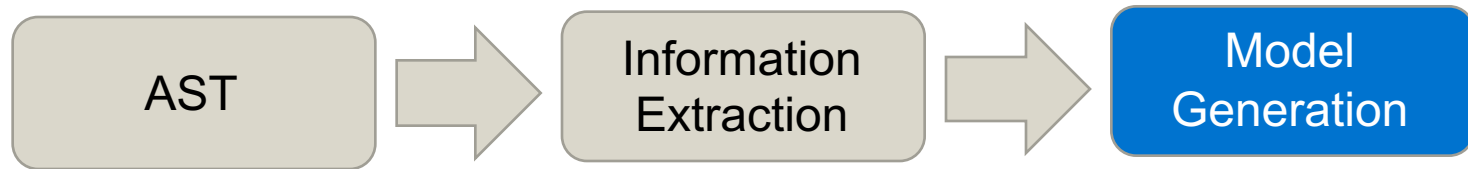


Find all **VariableDeclaration** that contain an **Identifier** on the right-hand side

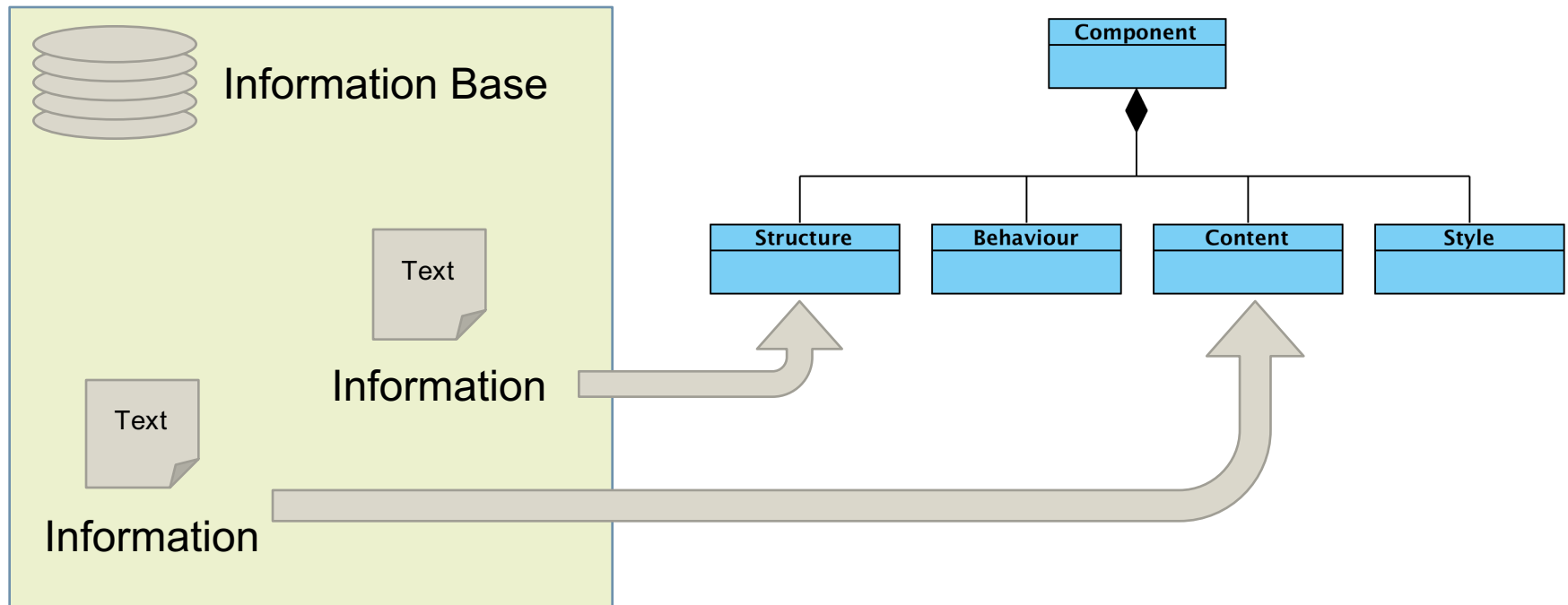


Rule





- Matching extracted information with web component view model



1. Introduction

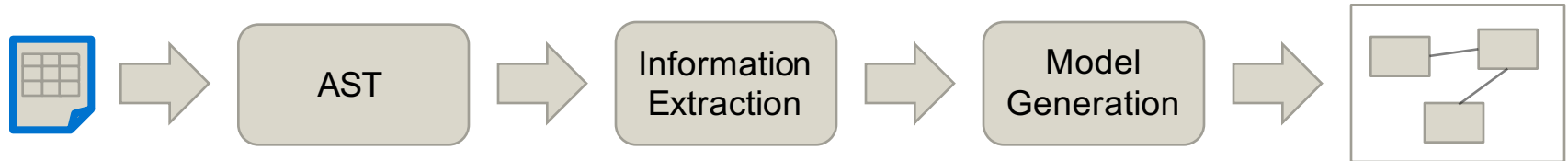
- Research Questions

2. Proposed Process

- AST
- Information Extraction
- Model Generation

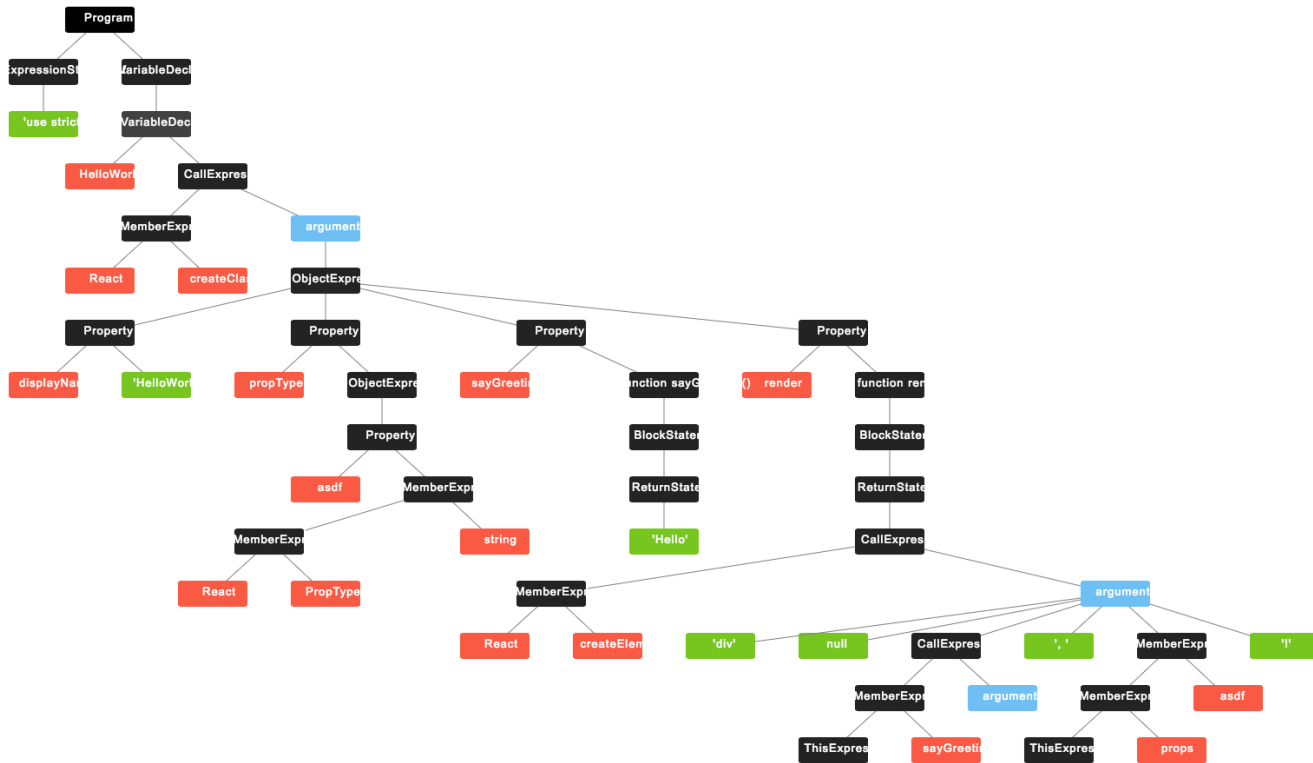
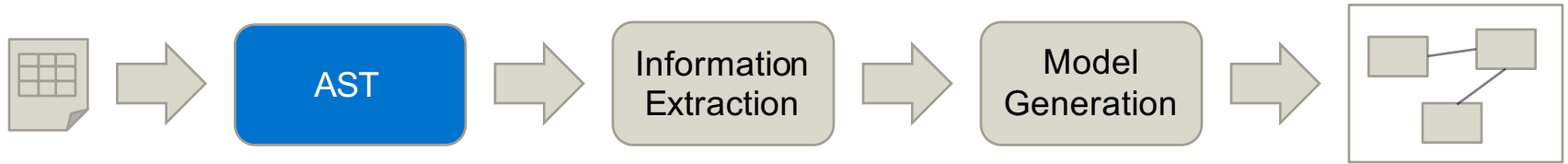
3. Process for React

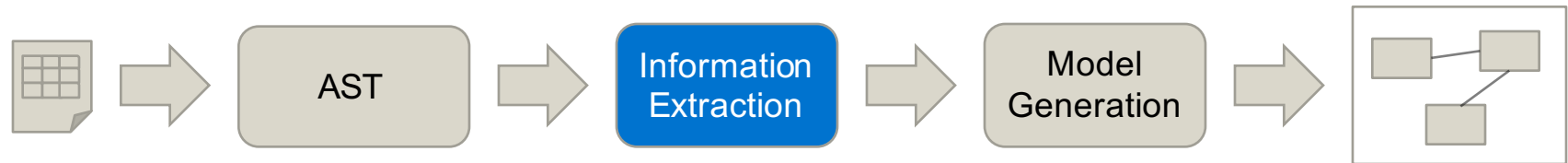
4. Next Steps & Timeline



```
var HelloWorld = React.createClass({
  propTypes: {
    name: React.PropTypes.string
  },
  sayGreeting: function () {
    return 'Hello';
  },
  render: function() {
    return (
      <div>
        {this.sayGreeting()} {this.props.name}!
      </div>
    );
  }
});
```

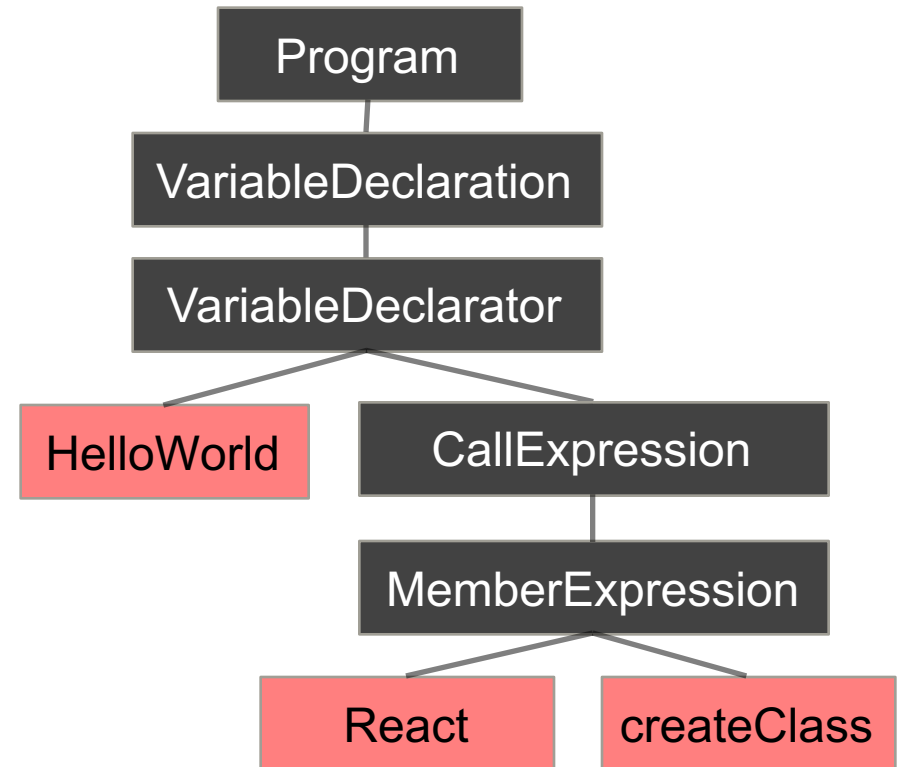
Example React

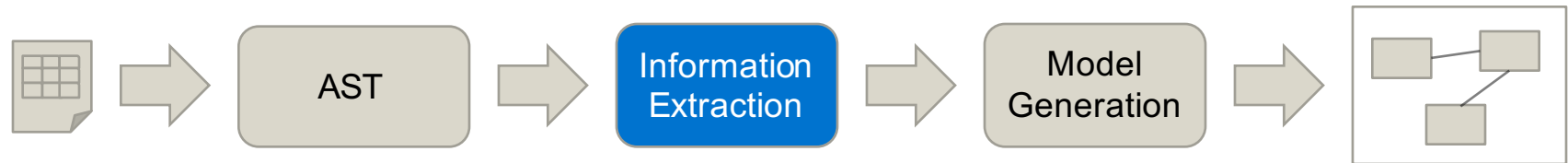




```
var HelloWorld = React.createClass({
```

Select all **VariableDeclaration** which
are assigned a **CallExpression**
of **React** and **createClass**





```
var HelloWorld = React.createClass({
  propTypes: {
    name: React.PropTypes.string
  },
  sayGreeting: function () {
    return 'Hello';
  },
  render: function() {
    return (
      <div>
        {this.sayGreeting()} {this.props.name}!
      </div>
    );
  }
});
```



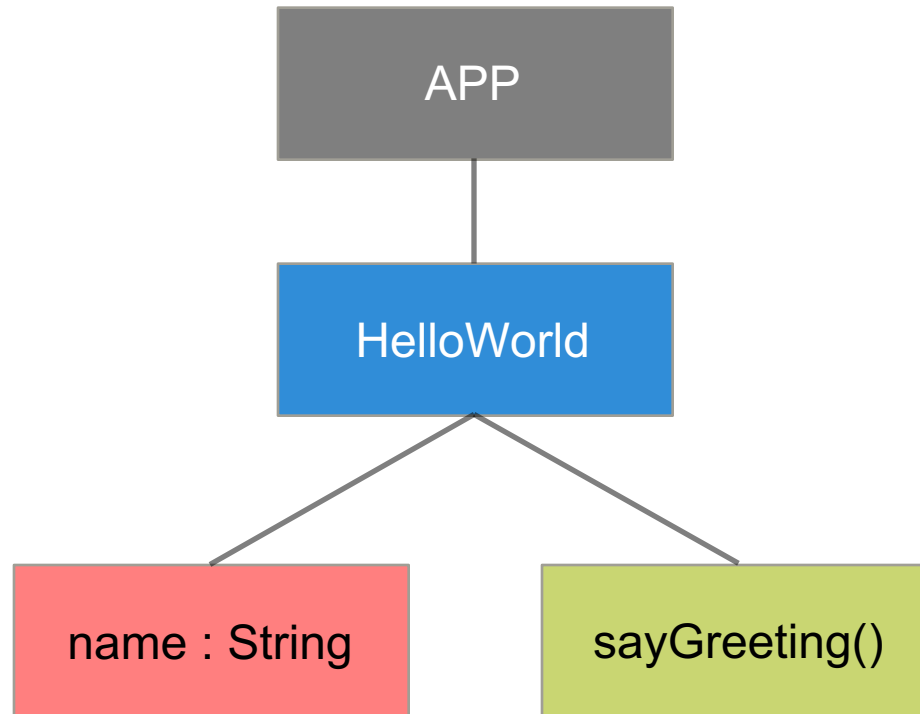
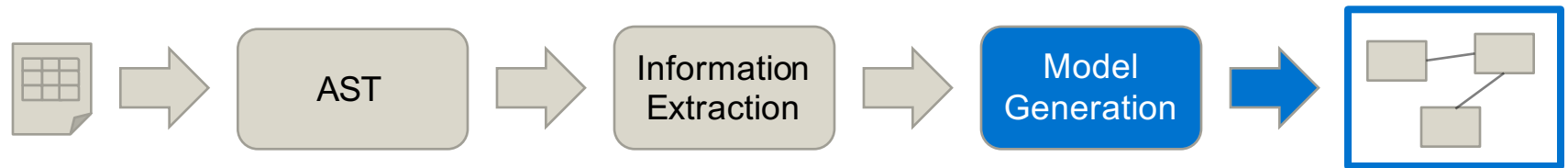
Information Base

Components

- *HelloWorld*

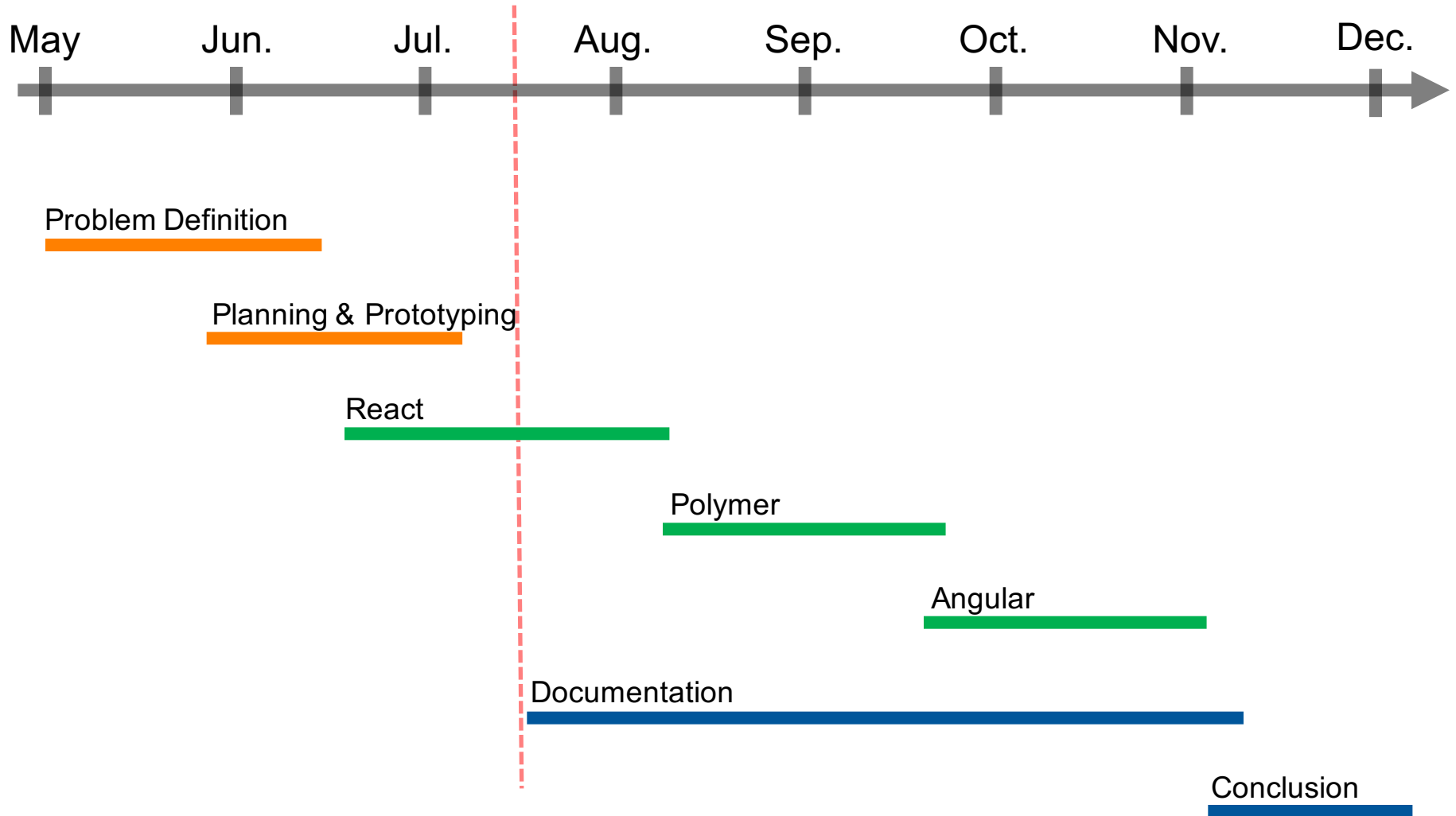
Functions

- *sayGreeting*
- *render*



- More frameworks
 - Polymer
 - Angular
- Model generation
- Include HTML & CSS







Thank you



Andreas Tielitz
B. Sc.



Technische Universität München
Department of Informatics
Chair of Software Engineering for
Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

tielitz@in.tum.de
www.matthes.in.tum.de