TECHNISCHE UNIVERSITÄT DRESDEN

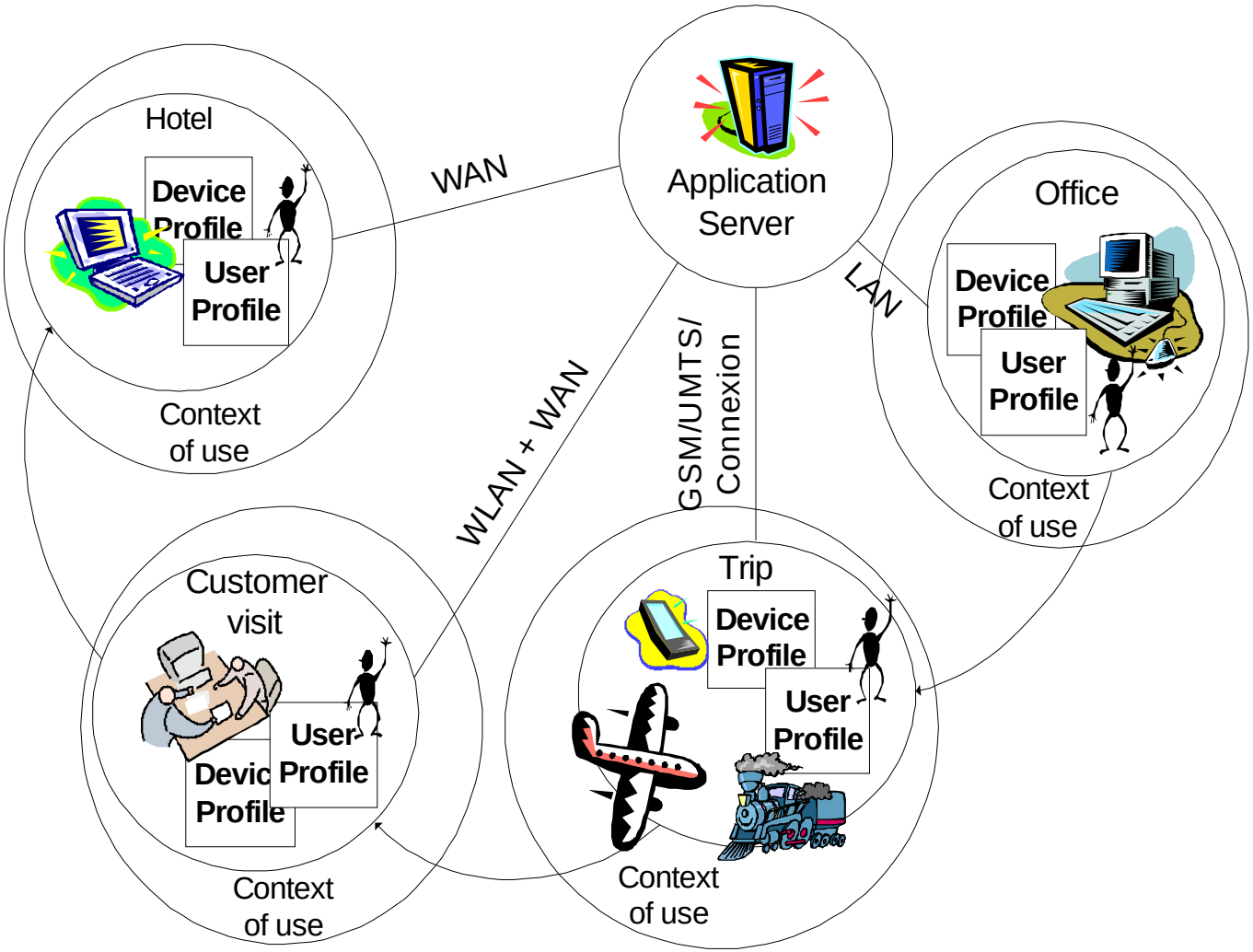
# An Integrated Approach for Platform-Independent Web Applications

W3C Multimodal Interaction Workshop
Sophia-Antipolis, 19.-20.7.2004

Gerald Hübsch and Thomas Springer
Dresden University of Technology
{huebsch,springet}@rn.inf.tu-dresden.de

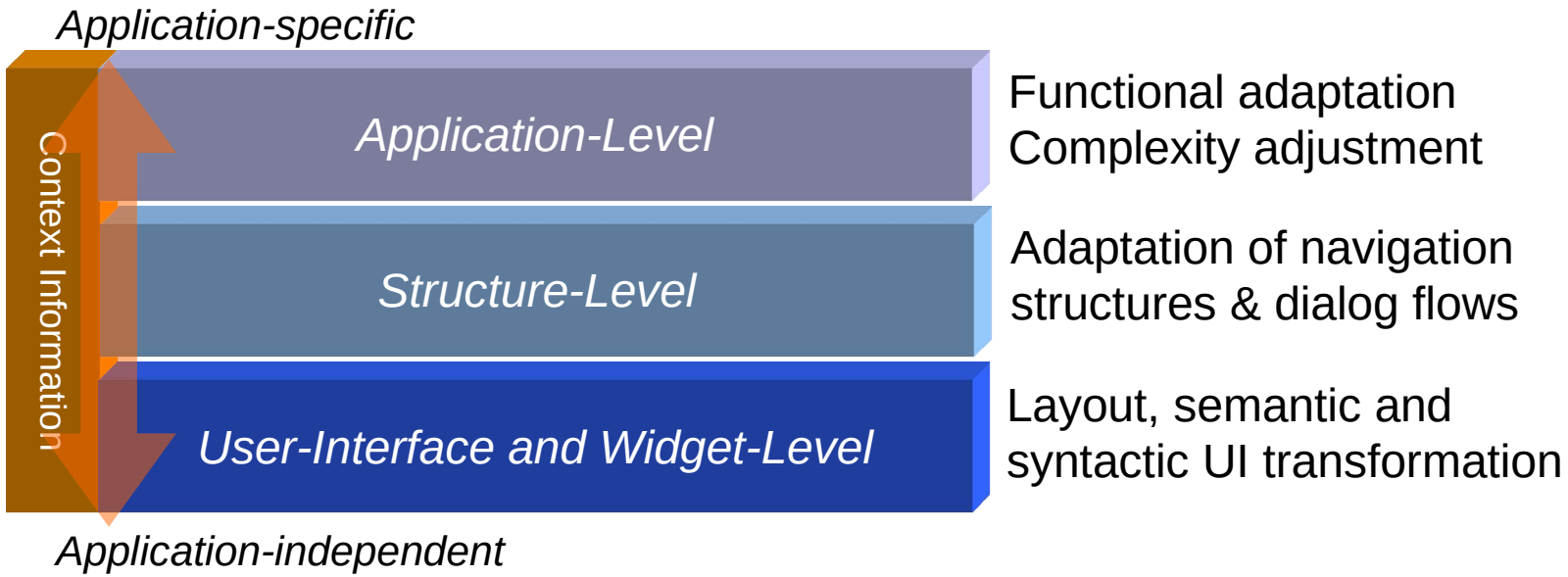
# Motivation

- Existing and upcoming wireless network infrastructures enable ubiquitous web-based applications
- State-of-the-Art web application design: desktop-style target platforms for fully-fledged browsers
- Today requirements imposed by the heterogeneity and limitations of mobile devices are neglected



# Adaptation as a promising solution

Adaptation is a promising solution to overcome the increasing differences in requirements imposed by mobile device technology




Today's adaptation approaches focus on User-Interface-level and Widget-level adaptation and support single modalities

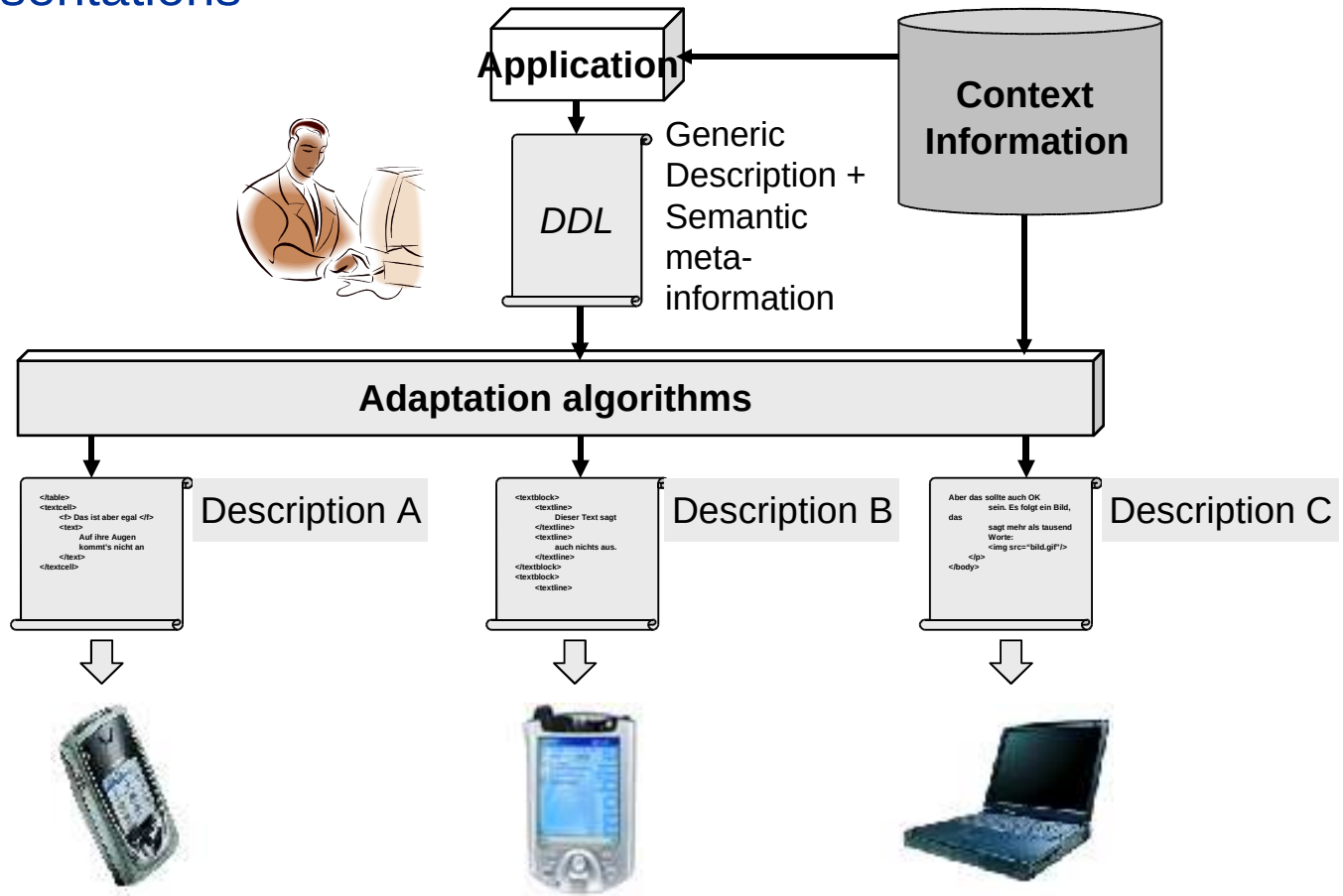
TECHNISCHE UNIVERSITÄT DRESDEN

# Dialog Description Language

**G. Hübsch, T. Springer –** Dresden University of Technology

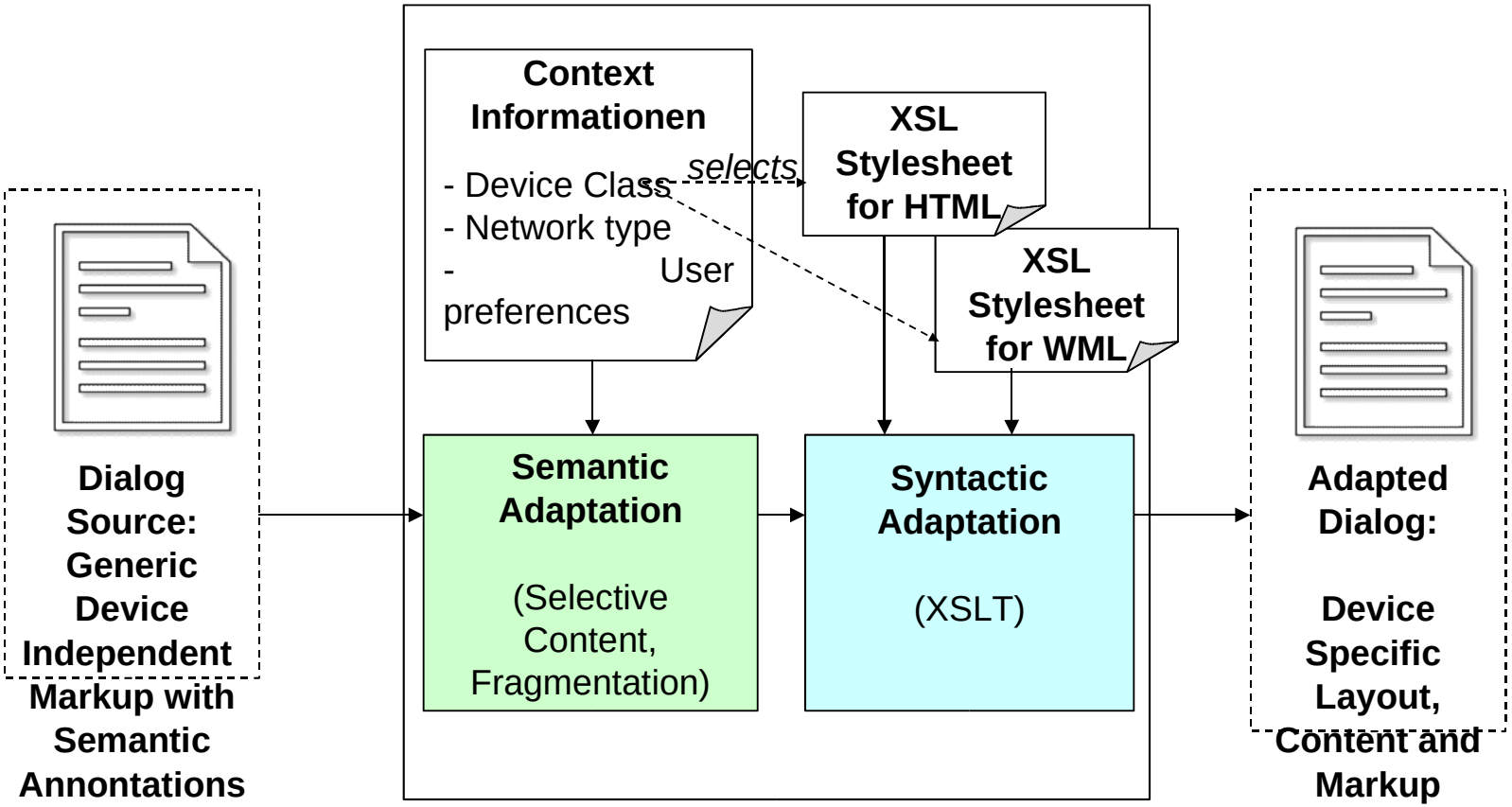
# Single Source Authoring

- Single Source Authoring uses generic dialog descriptions and applies dedicated adaptation algorithms parameterised by context information to generate device-specific content representations



# Adaptation Concepts

## Semantic and Syntactic Dialog Adaptation (UI & Widget Level Adaptation)



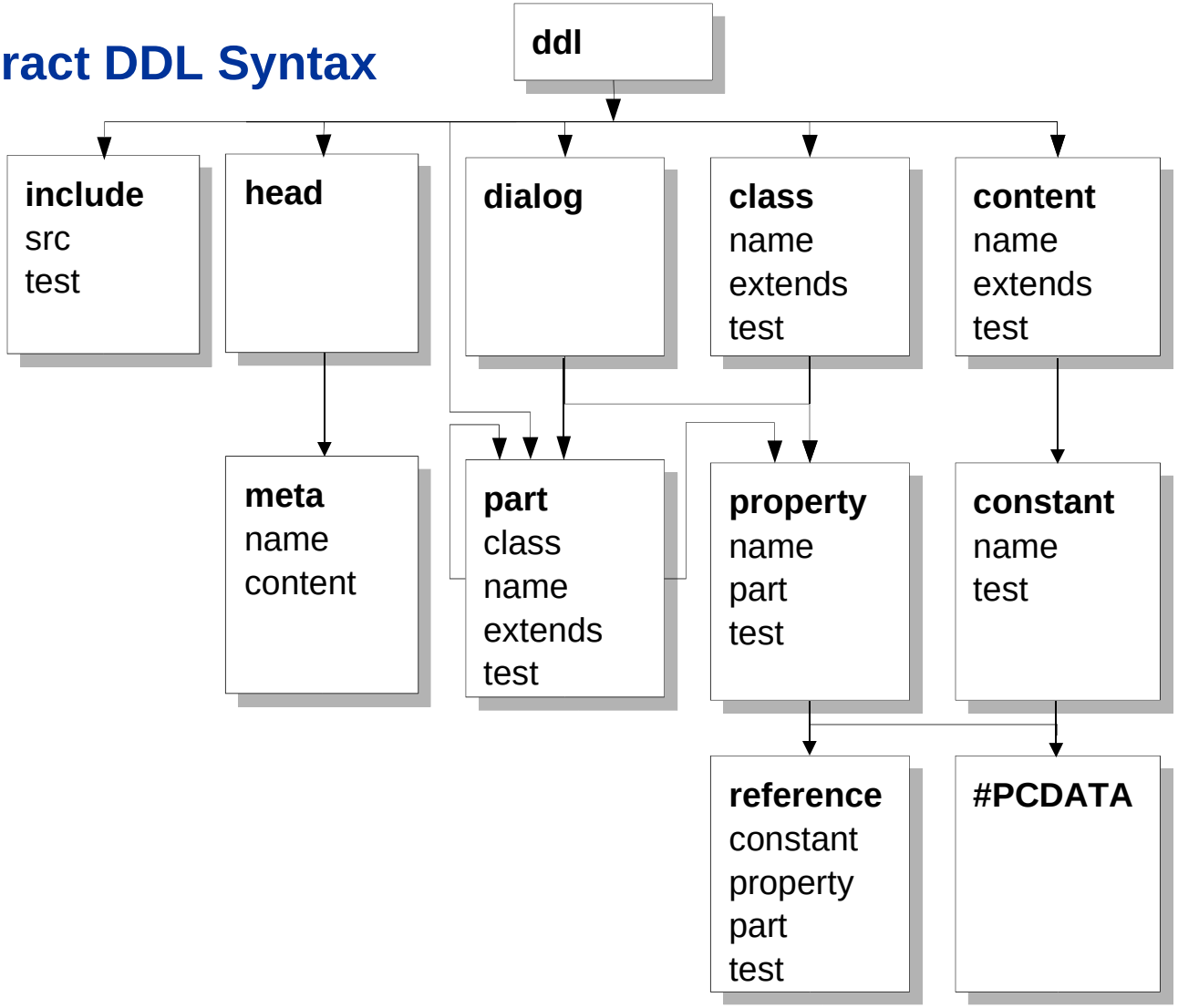
# Dialog Description Language

- XML-based device independent markup language for Single Source Authoring of web content and web application UIs
- **Separation of structure, content and presentation**
- Inheritance concept for content reuse
- **Inline meta-information governs adaptation**
- **Language concepts support semantic and syntactic adaptation**
- Support for dynamic content and interaction with application logic (MVC)
- Modular adaptation pipeline implements adaptation algorithms

```
<ddl>
  ...
  <dialog>
    <part>
      <property name="type">label</property>
    </part>
    ...
  </dialog>
  ...
</ddl>
```

# Dialog Description Language

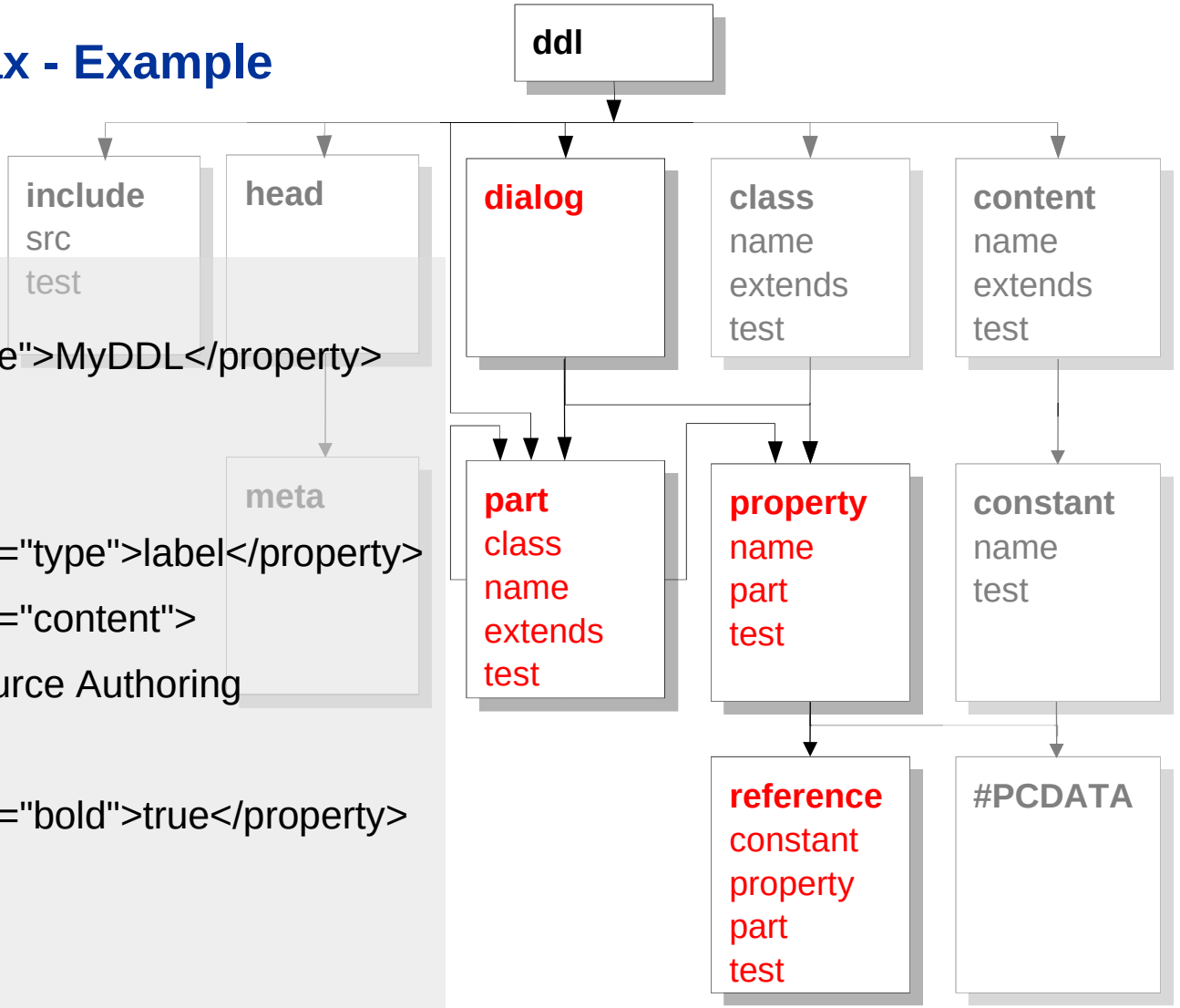
## Abstract DDL Syntax



# Dialog Description Language

## DDL Syntax - Example

```
<ddl>
<property name="title">MyDDL</property>
  <dialog>
    <part>
      <property name="type">label</property>
      <property name="content">
            Single Source Authoring
      </property>
      <property name="bold">true</property>
    </part>
  </dialog>
</ddl>
```

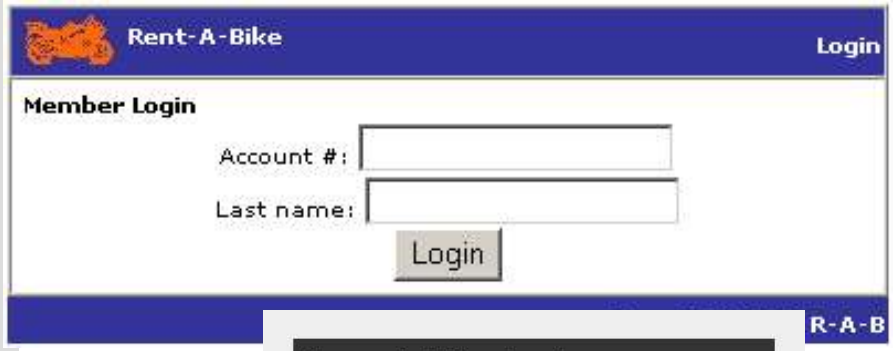


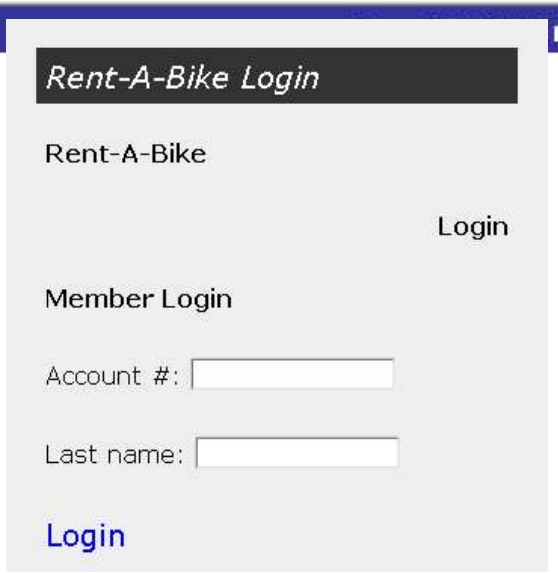
# Semantic Adaptation

## Selective exclusion / Alternative content based on context information

Context information query

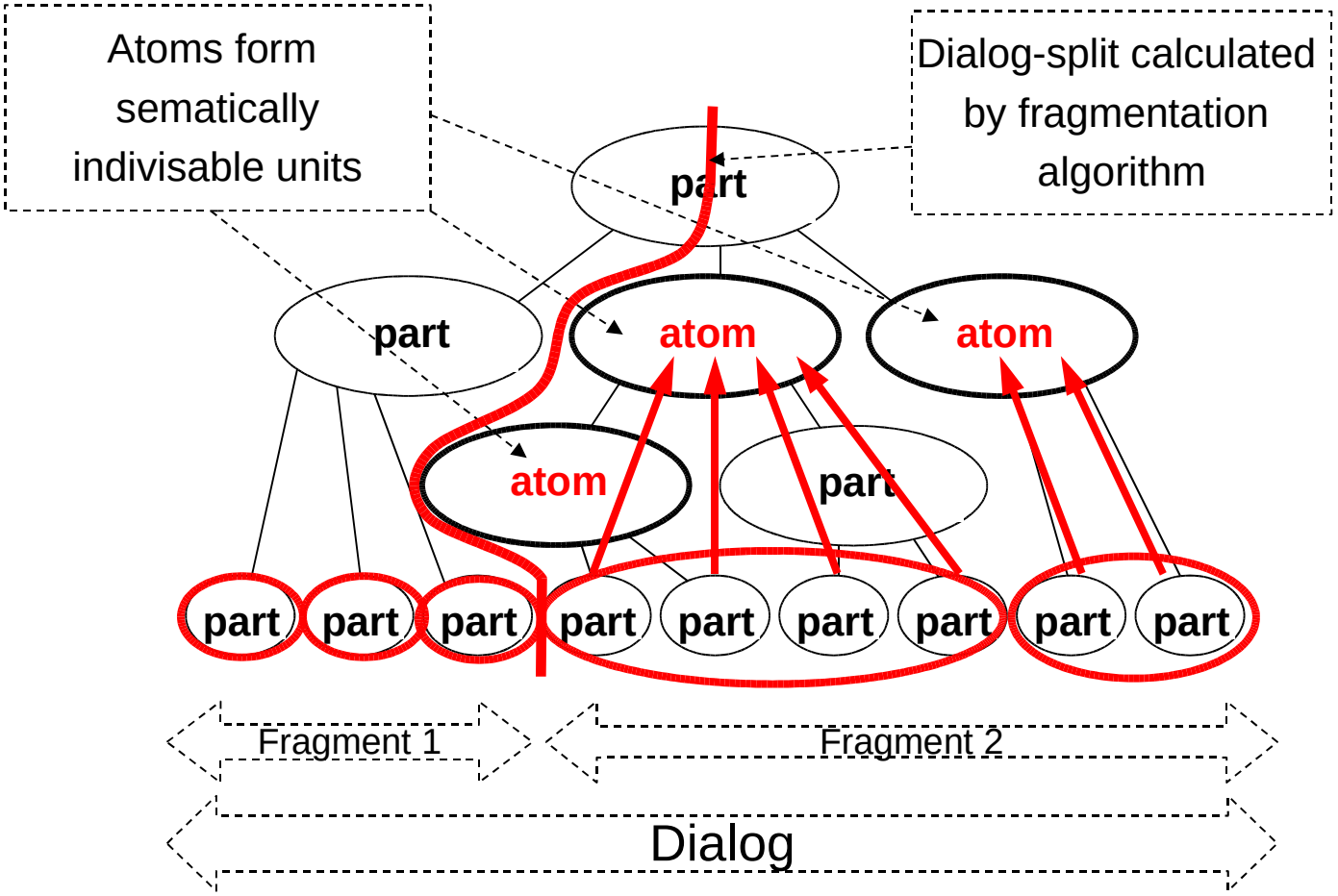
```
<part test="class='pda' or class='desktop'">
    <property name="type">image</property>
    <property name="src">logo.jpg</property>
</part>

<part test="class='wap'">
    <property name="type">label</property>
    <property name="content">
      Rent-A-Bike Login
    </property>
</part>
```



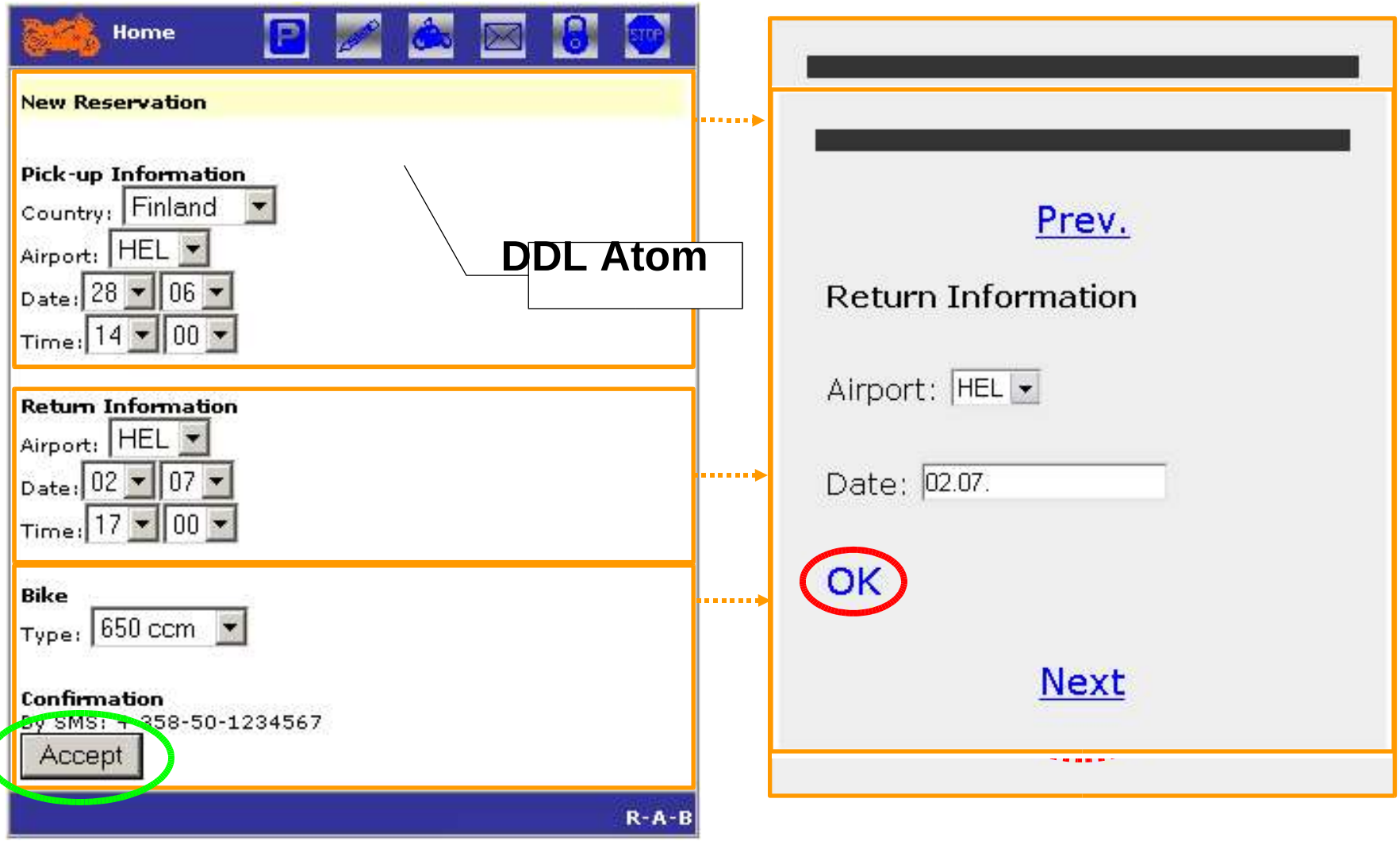
# Semantic Adaptation

**Fragmentation:** Approach to meet size and usability constraints imposed by the target platform and the network protocol through semi-automatic dialog splitting



# Semantic Adaptation

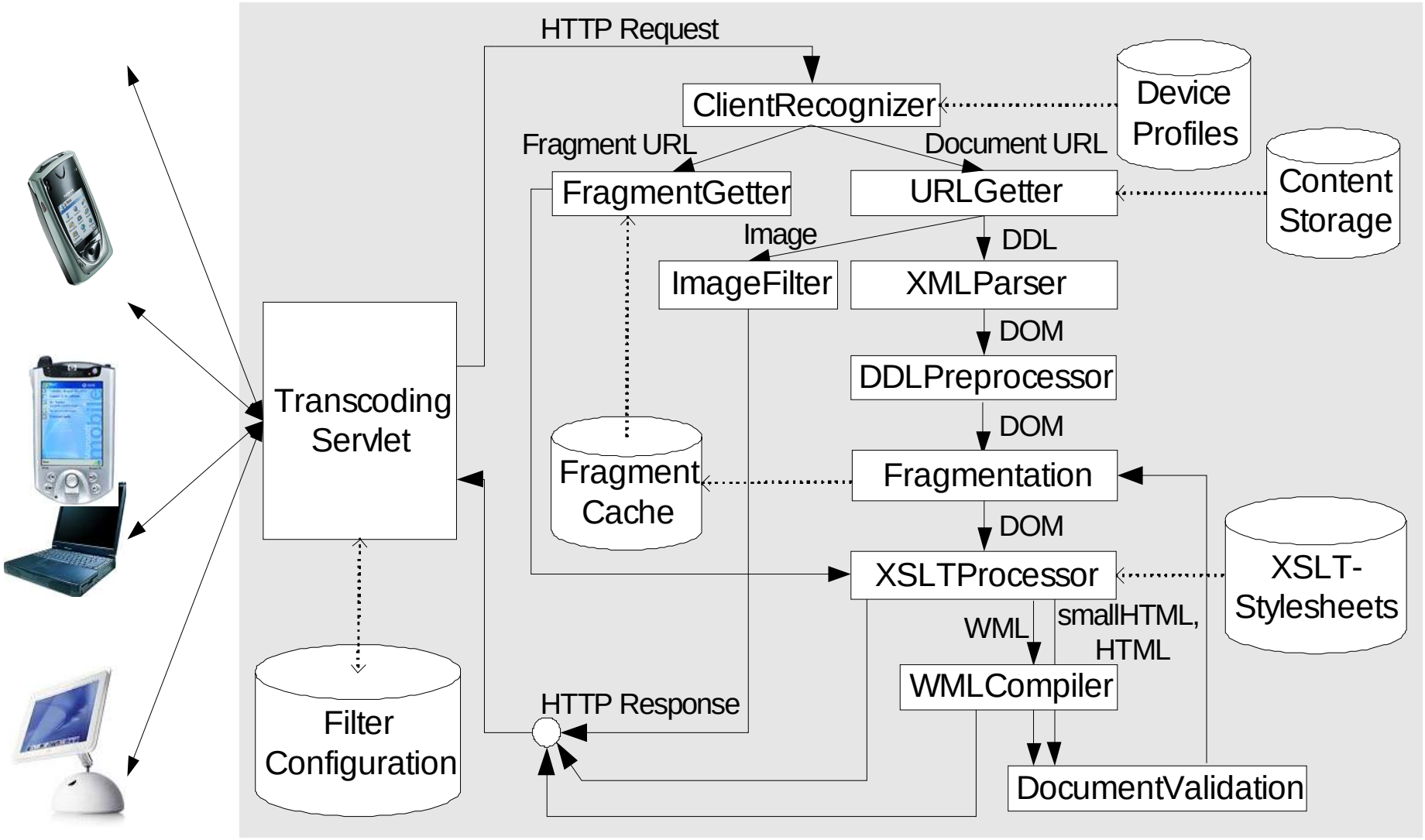
## Example: Dialog fragmentation



# Adaptation Engine

# Dialog Adaptation Architecture

## DDL Adaptation Engine Architecture (Simplified)



# Application Logic and Backend Integration

# Application Logic and Backend access

## Application logic interaction and Backend Integration

- Event-driven MVC pattern
- Dedicated language extensions for resource interface declaration, dynamic content sections and event listeners
- Integration with adaptation concepts and Adaptation Engine

DDL Event Handler Resource

```
<resource name="LoginHandler"
class="tud.eventhandler.RabHandler">


   <method name="validatelogin">
      <param name="lastname"/>
      <param name="account"/>
   </method>


</resource>
```

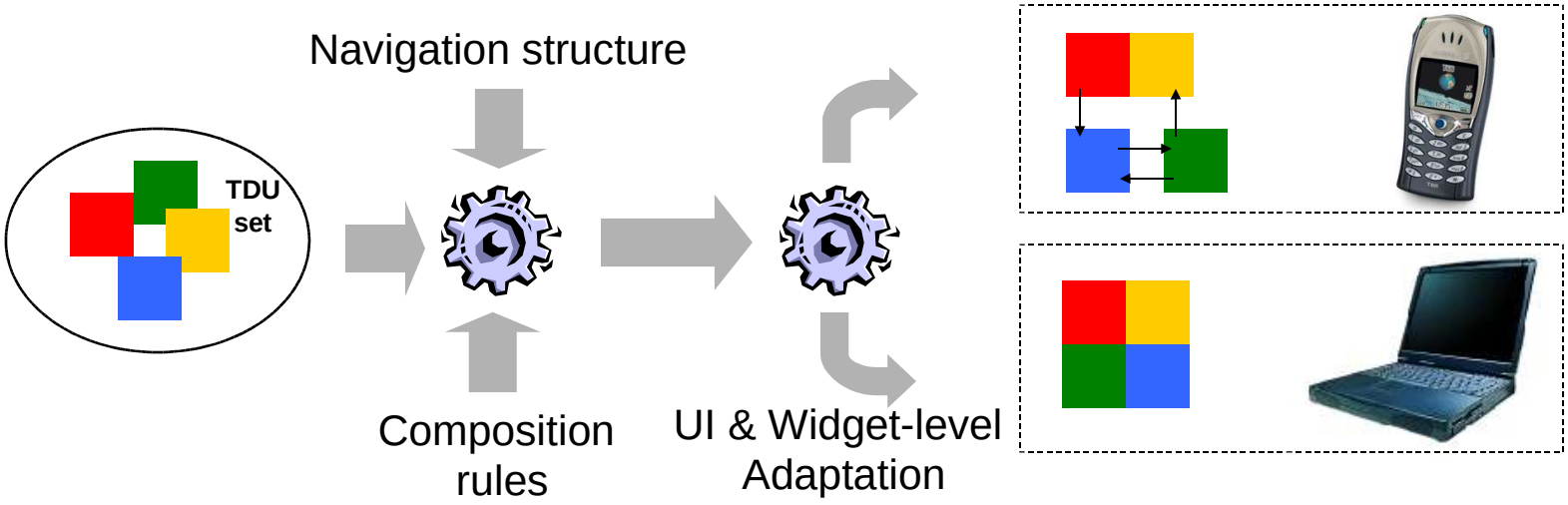
DDL Backend Access Resource

```
<resource name="RAB_Data"
 class="sql.RABConnection">


   <method name="getreservations">
      <field name="__label" />
      <field name="__rid" />
   </method>


</resource>
```

# Structure-Level Adaptation

# Structure-Level Adaptation

- Semi-automated adaptation of web applications with complex user interfaces is important for high user acceptance and low development costs
- Different application dialog flows for one task
- Serialisation of complex dialogs and user interactions
- Adaptation through dialog modularisation
  - Application user interfaces are described as a set **Transactional Dialog Units (TDU)**, e.g. authored in DDL



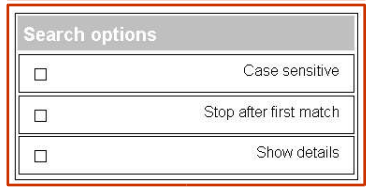
# Structure-Level Adaptation: Example

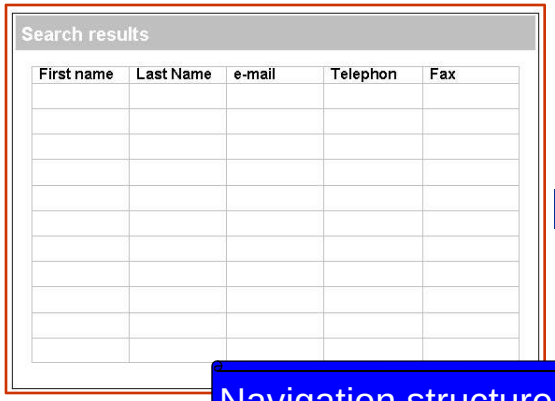
## Example: Phonebook Application

TDU 1: Search for…

TDU 2: Search options

TDU 3: Search results

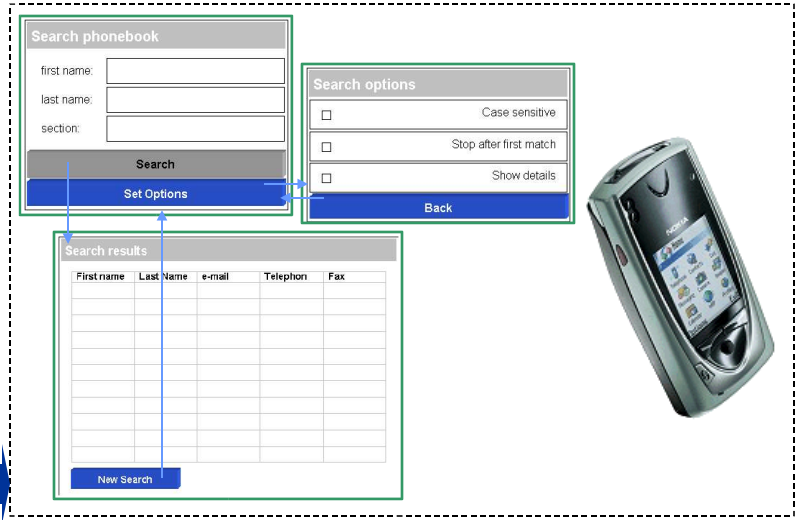



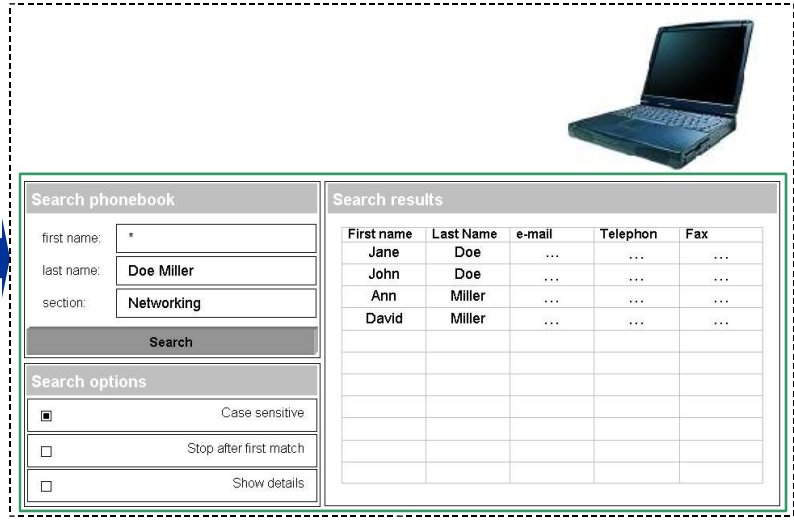




Navigation structure

Composition rules

Structure-level Adaptation

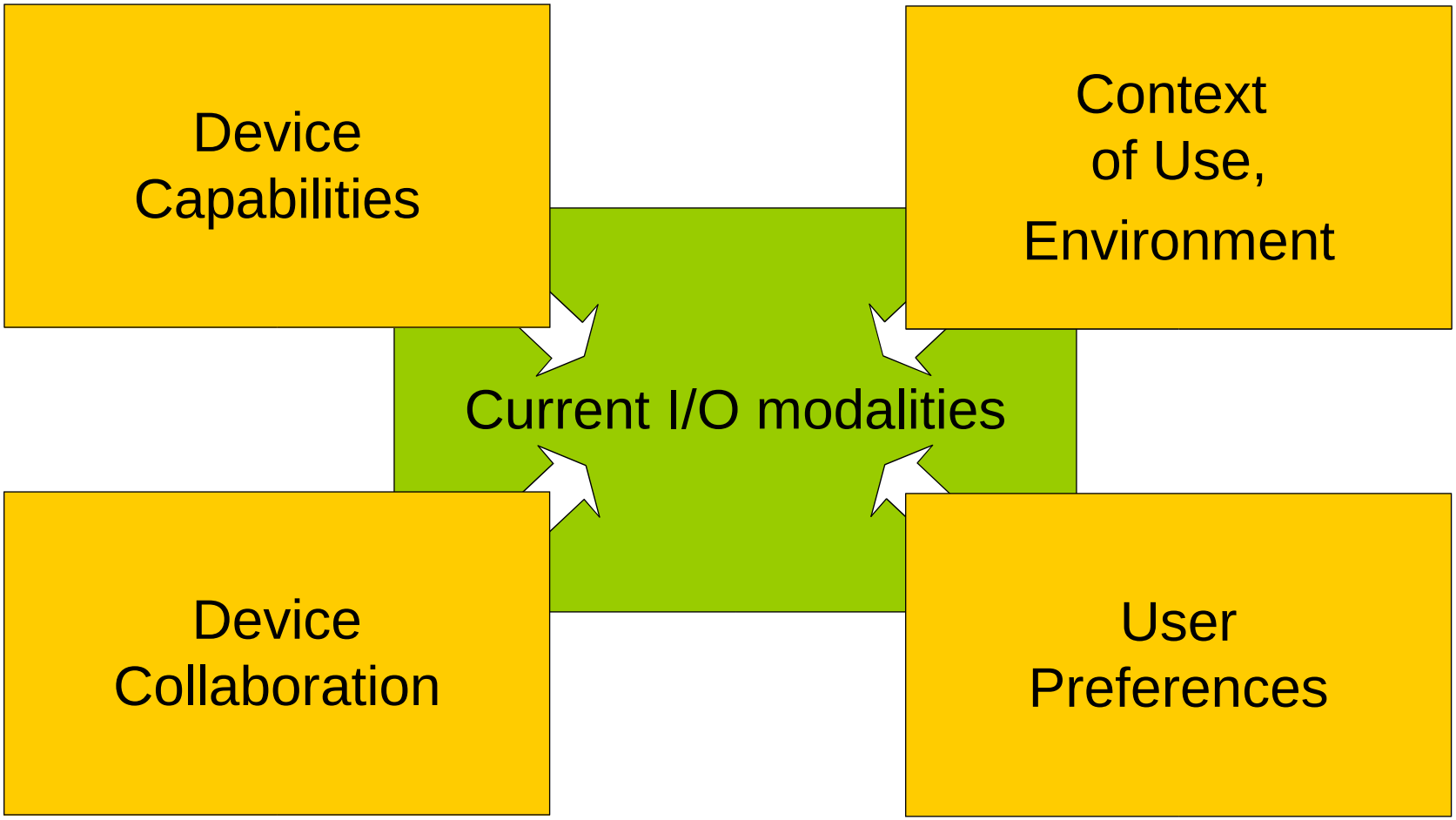


# Future Research

## Adaptation & Multimodal Interaction

# Multimodal user interaction + Adaptation



# Use Cases: Adaptive Multimodal Interfaces for industry and business applications

- Multimodal Maintenance Documentation (EU FP6 SNOW): Display + Voice + AR + Gesture + Robotic vision
  - Hands-free access to maintenance documentation and procedures in complex work situations
  - On-site capture of repair and maintenance data
  - Dynamic selection of interaction channels
- Multimodal Business Applications
  - Adaptive multimodal access to corporate applications: device *and* modality independent description of UI
  - Adaptive interaction channel selection
  - Dedicated platform services for multimodality support

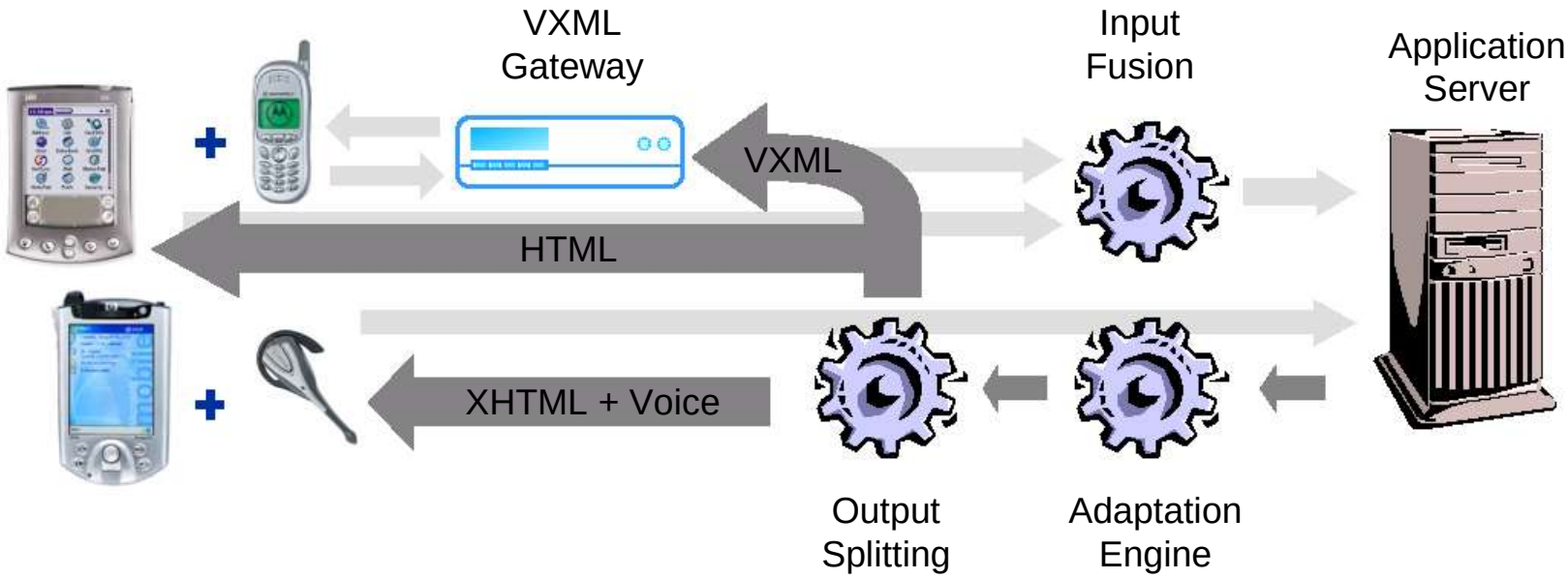
# Multimodal user interaction

- Adaptive multimodal applications can provide dynamic support for different modalities
- Novel adaptation concepts are required for modality-independent application description, interaction channel selection and channel utilization

# Adaptation and Multimodality

## Adaptation architecture for multimodal applications

- Additional components for output splitting and input fusion
- Client supports voice recognition, visual rendering and speech synthesis (e.g. XHTML+Voice Profile capable browser)
- Client supports visual rendering only, voice recognition and speech synthesis on VXML gateway



# Adaptation and Multimodality

## Adaptation in all phases of the software engineering process

### Analysis

- Mobile scenarios in Requirements Engineering
- Mobile computing in Business Processes
- Context-aware applications

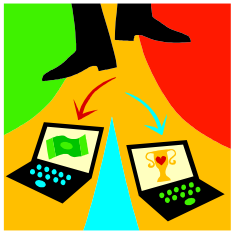
### Design

- Modality annotations
- Ambient-aware software architectures
- Adaptation infrastructures
- Design of adaptive UIs

### Implementation

- Weaving context into application logic and user interface
- Interaction with adaptation engines
- Implementation of multimodal and adaptive UIs

### Test & Deployment



- Deployment on heterogeneous platforms
- Integration with adaptation services and multimodal infrastructures
- Complex test cases

# Q & A