CapView

Functionality-Aware Visual Mashup Development for Non-programmers

CapView is a functionality-aware development view on running composite applications. It abstracts from interface and wiring details. Centered on component capabilities, CapView enables users to perceive mashup development as a process of assembling component functionalities.

Motivation and Problem

- The mashup paradigm and end user development (EUD) complement each other well.
- Hurdles for non-programmers:
  - limited understanding of technical concepts
  - no experience on development practices
  - mapping domain problems to specific combinations of components
- Requirements for non-programmers EUD:
  - avoid technical details and terminology
  - provide user guidance and automation
  - provide immediate feedback and a task-oriented user interface
- Contributions: CapView enables...
  - non-programmers to realize “components” as task solving entities,
  - explore mashup functionalities,
  - and visually manipulate a mashup.

CapView Basics

- Assumption: Mashups offer a set of functionalities, which deliver outputs to or consume inputs from other functionalities. Properties represent attributes of components.
- CapView is part of the EDYRA platform enabling “live sophistication” of mashups.
- CapView is a view overlaying a running mashup
  - utilization of natural language to abstract from technical details of a mashup.
- Semantic component description (SMCDL)
- User guidance via hybrid recommendation

Visual Exploration

- Natural language labels, derived from semantic annotations, represent capabilities
- Color coding:
  - capabilities according to need of user interaction, e.g., blue (yes) or orange (no)
  - properties are colored uniformly green
- Clustering equal capabilities per component
- Ports represent inputs and outputs and can be connected
- Ports can be selected
  - all representations r within the 1-layer S0 and T0 gets relabeled according to r0 to form human readable phrases
  - related connections are transitively highlighted

Context Specific Labels

- A generic rule set enables context sensitive label generation for capabilities based on annotated activities and entities
- The generation process distinguishes two cases, if nothing is selected:
  - properties are labeled with their ontology concepts and instance data, if available, e.g., current location (Dresden)
  - capabilities are displayed via human-readable phrase, e.g., search a route
- If a user selects a specific r0, labels of connectable capabilities are adapted to clarify cause and effect
  - thereby all ri and rj are treated differently
- Trailings or leading data helps to clarify the reading sequence

Reconfiguration of Mashups

- Connectable capabilities can be combined using drag & drop
  - options in case of ambiguous mapping
- Automatic and transparent implementation of connections in terms of composition model concepts by the platform
- Recommendation menu lists capabilities of components not part of the mashup yet