ConcurTaskTrees and MARIA Languages for Authoring Service-based Applications

Fabio Paternò, Carmen Santoro, Davide Spano
CNR-ISTI, HIIS Laboratory
Pisa, Italy
http://giove.isti.cnr.it

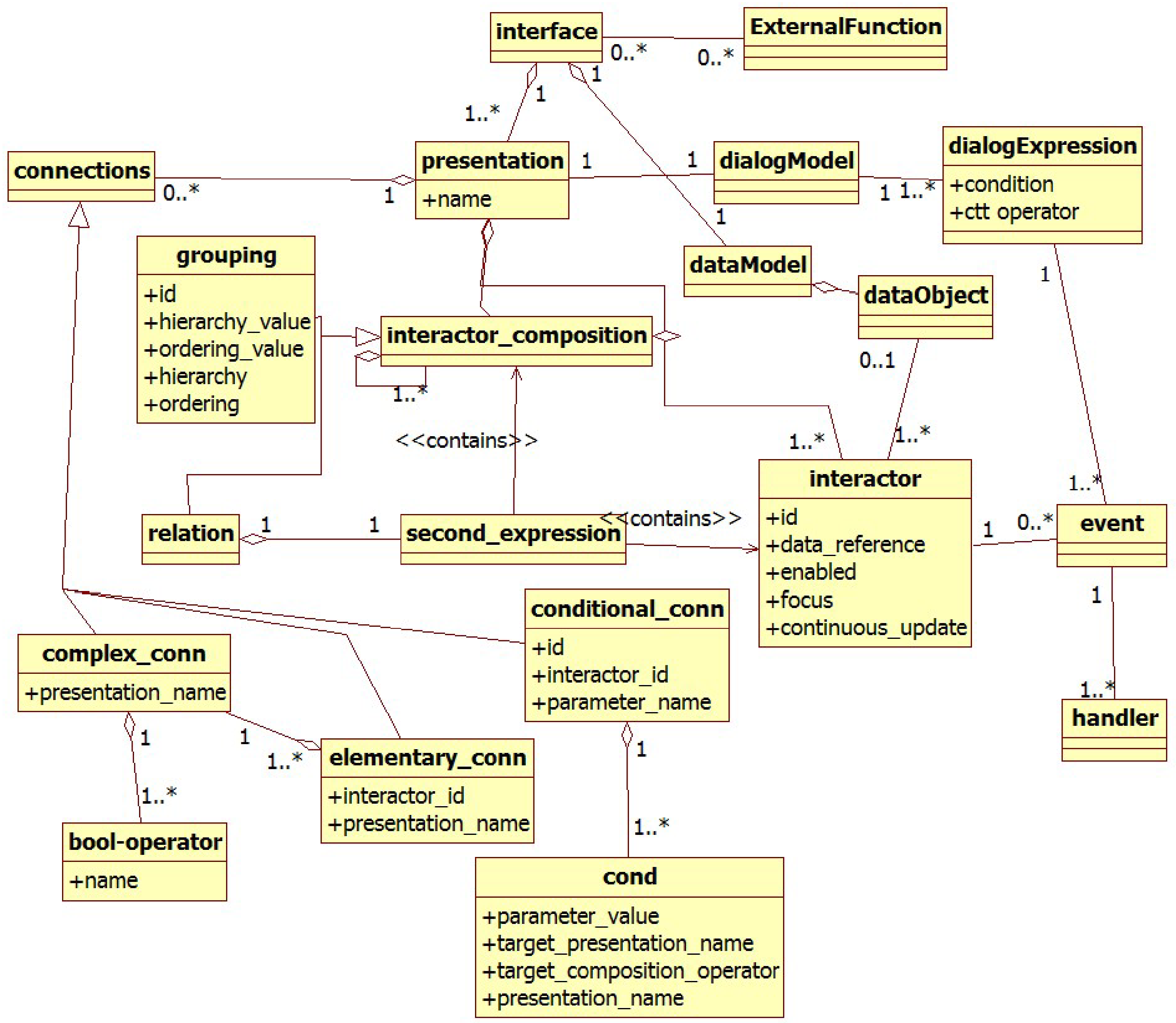
Mark-up Languages for Model-based Approaches

- UIML, XForms, TERESA, UsiXML
- ACM TOCHI Special Issue, NESSI Roadmap
- Need for novel solutions able to:
  - support access to a number of pre-existing Web services that can be distributed everywhere
  - support access to various interaction modalities (multi-touch gestures, voice, ...)
  - support ability to change the content of user interfaces dynamically

MARIA XML Features

- Support for Data Model
  - Useful for specifying the format of input values, association of various data objects to the various interactors, ..
- Events at abstract/concrete levels
  - Property change events / Activation events (e.g. access to a web service or a database)
- Extended Dialogue Model
  - Conditions and CTT operators for event handlers, including support for parallel input
- Able to support user interfaces including complex and Ajax scripts
  - Continuously updating of fields without explicit user request
- Dynamic set of user interface elements
  - Conditional connections between presentations
  - Possibility to change only a part of a UI

The AUI MetaModel



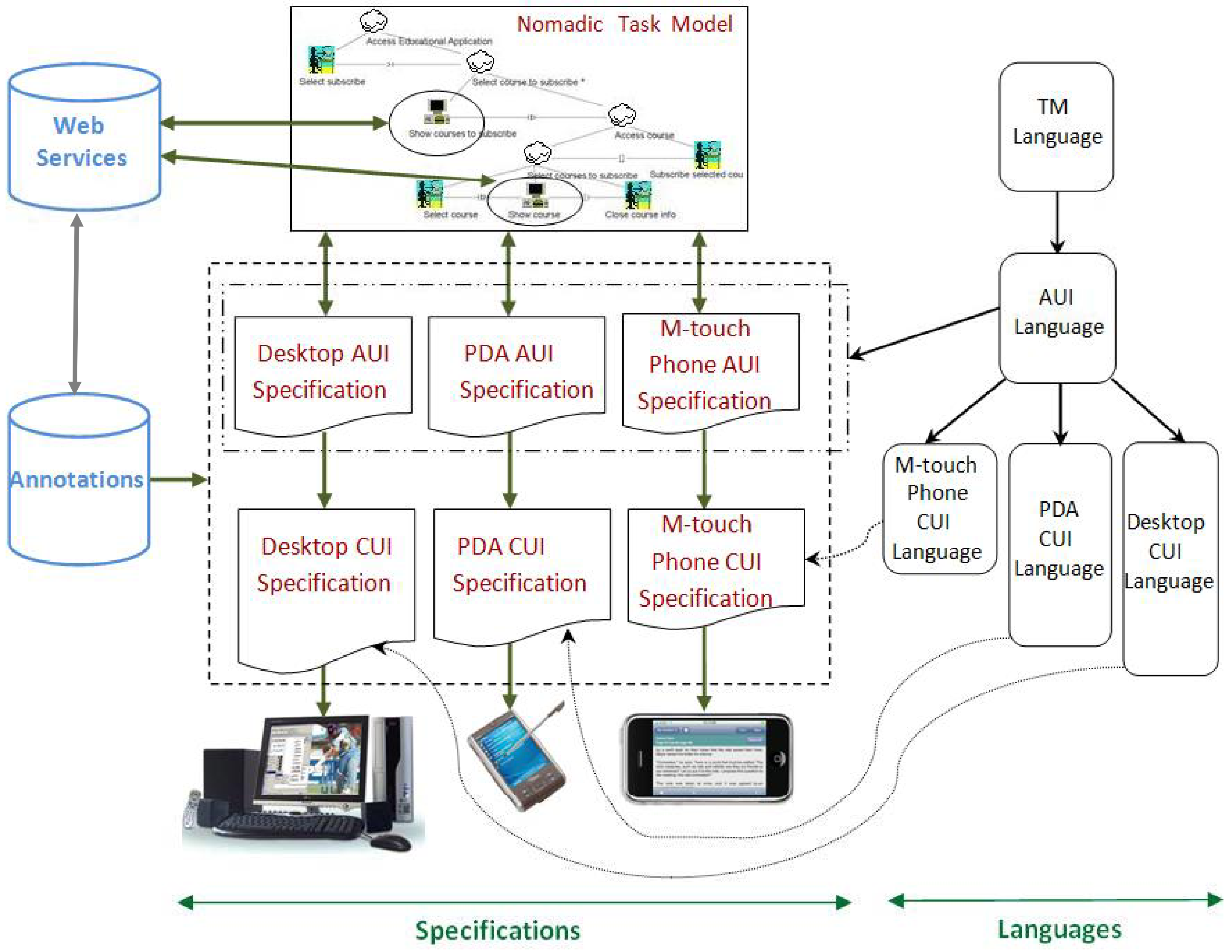
Support Service-Oriented Applications

- Web services are increasingly used to support remote access to application functionalities
- Services are defined previous to the interactive applications
- Applications are built through composing such services
- Service composition
  - at the *Service* level,
  - at the *Application* level,
  - at the *User Interface* level

UI Annotations for Web services (EU ServFace Project)

- The hints do not have the aim of defining the whole interface
- They can address various aspects (presentation, content, dynamic behaviour)
- They should be independent of the UI implementation language
- They can have various granularities
- They can abstract from the platform (if necessary)
- Examples: Labels, icons, validation data rules, optional/mandatory fields, ...

Support for Applications based on Web Services



The Proposed Design Approach

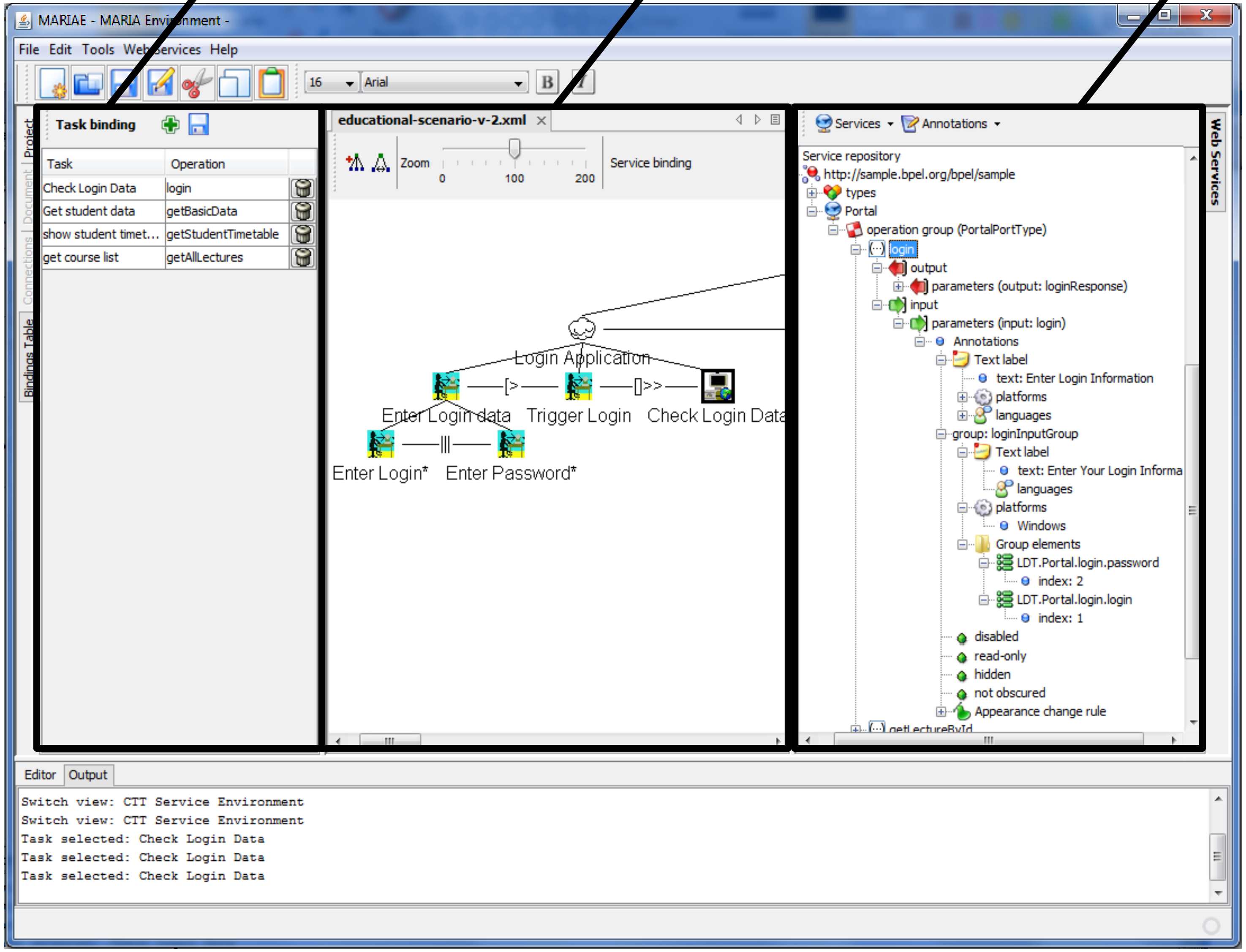
- First a bottom-up step in order to analyse the Web services providing functionalities useful for the new application to develop.
  - Analysis of the operations and the data types associated with input and output parameters is carried out in order to associate them with suitable abstract interaction objects
- Task model expressed in ConcurTaskTrees (CTT) for describing the interactive application and how it assumes that tasks are performed.
  - Design based on user requirements
  - Indicate how to compose functionalities implemented in different Web services, which are associated with system tasks.
- Level of granularity to reach in the task decomposition.
  - Associating the system basic tasks to the web services
  - Associate each system basic task with the operations of the web services. Thus, if a Web Service supports three operations, then there would be three basic system tasks.
- Transformation of Task Models into MARIA Descriptions
  - Temporal operators indicate requirements for UI dialogue model
  - Task hierarchy provides information regarding grouping of UI elements
  - Type of task indicates type of interactor
  - Use information from tasks/operation bindings
  - Inclusion of annotations in resulting Abstract/Concrete Specification
- Editing Abstract/Concrete Specification
- User Interface Generation

Service, Tasks and Annotations

Tasks/Operations binding

Task Model

Service and annotations



Task	Operation
Check Login Data	login
Get student data	getBasicData
show student timet…	getStudentTimetable
get course list	getAllLectures

Service repository
- http://sample.bpel.org/bpel/sample
  - types
  - Portal
    - operation group (PortalPortType)
      - login
        - output
          - parameters (output: loginResponse)
        - input
          - parameters (input: login)
            - Annotations
              - Text label
                - text: Enter Login Information
                - platforms
                - languages
              - group: loginInputGroup
                - Text label
                  - text: Enter Your Login Informa
                  - languages
                - platforms
                  - Windows
                - Group elements
                  - LDT.Portal.login.password
                    - index: 2
                  - LDT.Portal.login.login
                    - index: 1
              - disabled
              - read-only
              - hidden
              - not obscured
              - Appearance change rule
      - getLectureById

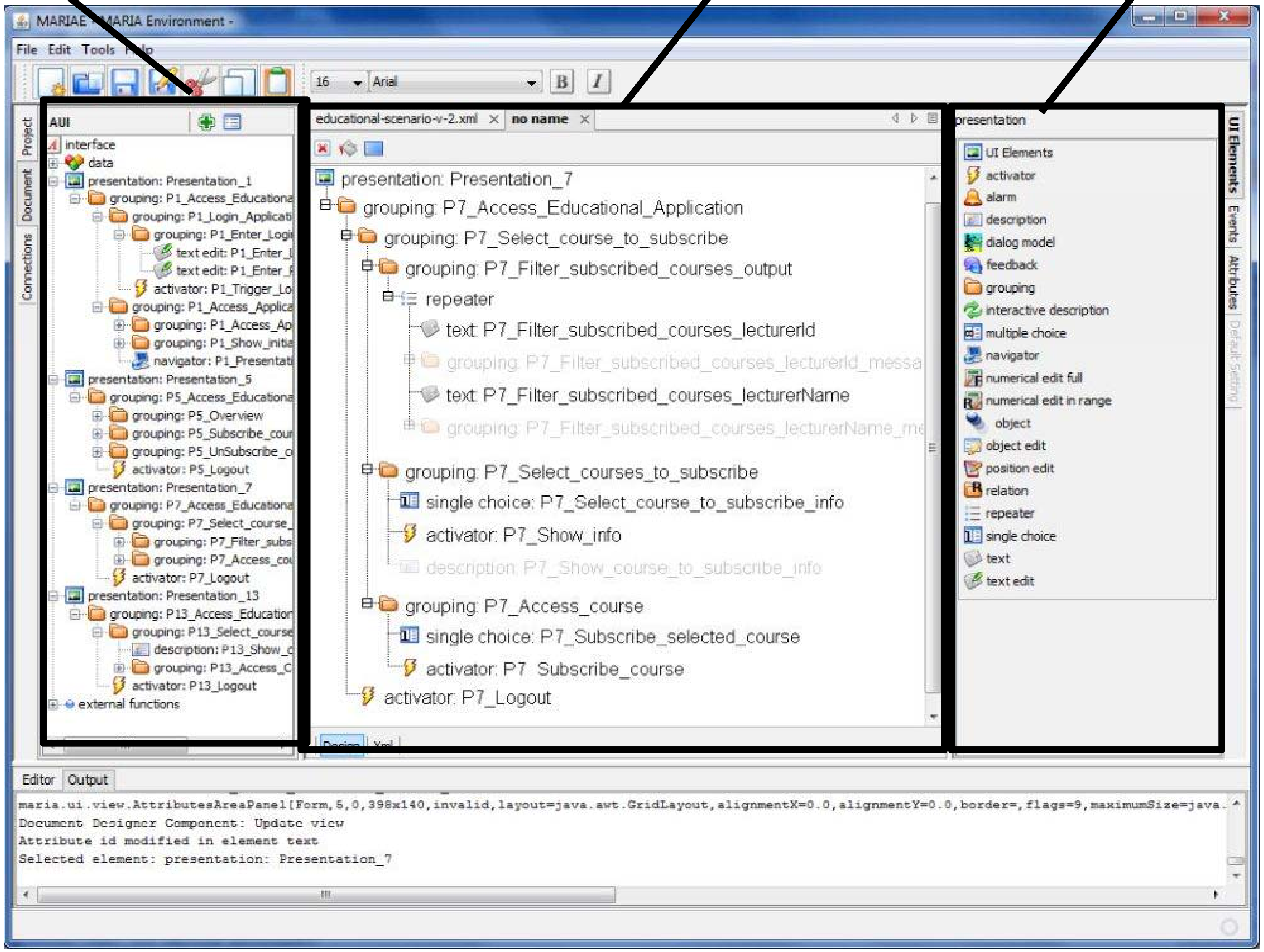
```
Switch view: CTT Service Environment
Switch view: CTT Service Environment
Task selected: Check Login Data
Task selected: Check Login Data
Task selected: Check Login Data
```

Editing Environment

Interactive Tree View

Visual Specification

Elements and attributes



MARIAE - MARIA Environment -
File Edit Tools Help
16
Arial
AUI
interface
data
presentation: Presentation_1
grouping: P1_Access_Educationa
grouping: P1_Login_Applicati
grouping: P1_Enter_Logi
text edit: P1_Enter_l
text edit: P1_Enter_F
activator: P1_Trigger_Lo
grouping: P1_Access_Applica
grouping: P1_Access_Ap
grouping: P1_Show_initia
navigator: P1_Presentati
presentation: Presentation_5
grouping: P5_Access_Educationa
grouping: P5_Overview
grouping: P5_Subscribe_cour
grouping: P5_UnSubscribe_c
activator: P5_Logout
presentation: Presentation_7
grouping: P7_Access_Educationa
grouping: P7_Select_course_
grouping: P7_Filter_subs
grouping: P7_Access_cou
activator: P7_Logout
presentation: Presentation_13
grouping: P13_Access_Education
grouping: P13_Select_course
description: P13_Show_c
grouping: P13_Access_C
activator: P13_Logout
external functions
educational-scenario-v-2.xml
no name
presentation: Presentation_7
grouping: P7_Access_Educational_Application
grouping: P7_Select_course_to_subscribe
grouping: P7_Filter_subscribed_courses_output
repeater
text: P7_Filter_subscribed_courses_lecturerId
grouping: P7_Filter_subscribed_courses_lecturerId_messa
text: P7_Filter_subscribed_courses_lecturerName
grouping: P7_Filter_subscribed_courses_lecturerName_me
grouping: P7_Select_courses_to_subscribe
single choice: P7_Select_course_to_subscribe_info
activator: P7_Show_info
description: P7_Show_course_to_subscribe_info
grouping: P7_Access_course
single choice: P7_Subscribe_selected_course
activator: P7_Subscribe_course
activator: P7_Logout
Design
Xml
presentation
UI Elements
activator
alarm
description
dialog model
feedback
grouping
interactive description
multiple choice
navigator
numerical edit full
numerical edit in range
object
object edit
position edit
relation
repeater
single choice
text
text edit
UI Elements
Events
Attributes
Default Setting
Project
Document
Connections
Editor
Output

```
maria.ui.view.AttributesAreaPanel[Form,5,0,398x140,invalid,layout=java.awt.GridLayout,alignmentX=0.0,alignmentY=0.0,border=,flags=9,maximumSize=java.
Document Designer Component: Update view
Attribute id modified in element text
Selected element: presentation: Presentation_7
```

Current State

- Concrete Languages (desktop, smartphones with gestures, mobile, vocal, multimodal combination of graphical and vocal)
- Implementation languages (XHTML, SMIL, VoiceXML, X+V, planned HTML 5 and JSP for Web service access)
- Next week remote test with people working in software companies, any volunteer?