USIXML
USer Interface eXtended Markup Language

UsiXML – General Description

François Beuvens, Vivian Genaro Motti, Jérémie Melchior, Benoît Michel
Ricardo Tesoriero, Jean Vanderdonckt
Université catholique de Louvain
on behalf of UsiXML Consortium
http://www.usixml.eu/view-all-partners

USIXML User Interface eXtended Markup Language

Fact sheet

- UIDL Name

  User Interface eXtensible Markup Language

- Version

  V2.1 (15 November 2011, 12th version)

- Availability

  Open and free (LGPL 3.0 – no commercial license)

- Collaboration

  The UsiXML Consortium consists of 21 organisations from 7 countries (FR, BE, DE, PT, ES, GR, RO). They voted unanimously to agree on current UsiXML version.

USIXML
USer Interface eXtended Markup Language

The current consortium

End User Club members

I am Observer of USIXML

Observer

Express their interest for the project, its goals, scientific results, methods, tools or demonstrators.

I am Supporter of USIXML

Supporter

Express their interest for specific results of the project (from meta-models to validators) and wish to receive information.

I am Promoter of USIXML

Promoter

Express their interest for UsiXML goals and plan to create demonstrator using the UsiXML language and tools.

USIXML USer Interface eXtended Markup Language

Fact sheet

- Backgrounds

  The UsiXML Consortium combines various competences: software engineering, web services, business process engineering, HCI, web engineering, language engineering, distributed computing, reverse engineering

- Websites

  www.usixml.org, www.usixml.eu, facebook.com/usixml, usixml.postano.com

- Funding

  FP5 Cameleon, FP6 Similar, FP7 Human, FP7 Selfman, FP7 Serenoa, Initiatives III Salamandre, ITEA2 Call 4, …

- Other projects

  Other projects that used and extended UsiXML (not just downloaded) are FP7 Veritas, GENIUS project, IMPEQ project, Univ. of Hokkaido, Univ. of Fulda, AIR, ReQuest, Destine, MulPlex, …

USIXML User Interface eXtended Markup Language

Fact sheet

- Semantics

  Defined in UML 2.0 class diagram, MOF-XMI, OWL-Full 2.0

- Syntax

  Defined in XML Schema

- Stylistics

  Defined in GMF

- Coverage

  Task, domain, abstract user interface (AUI), concrete user interface (CUI), context of use, adaptation, quality of use, transformation, mapping, workflow, process, interactor

- Validation

  With OCL and Eclipse

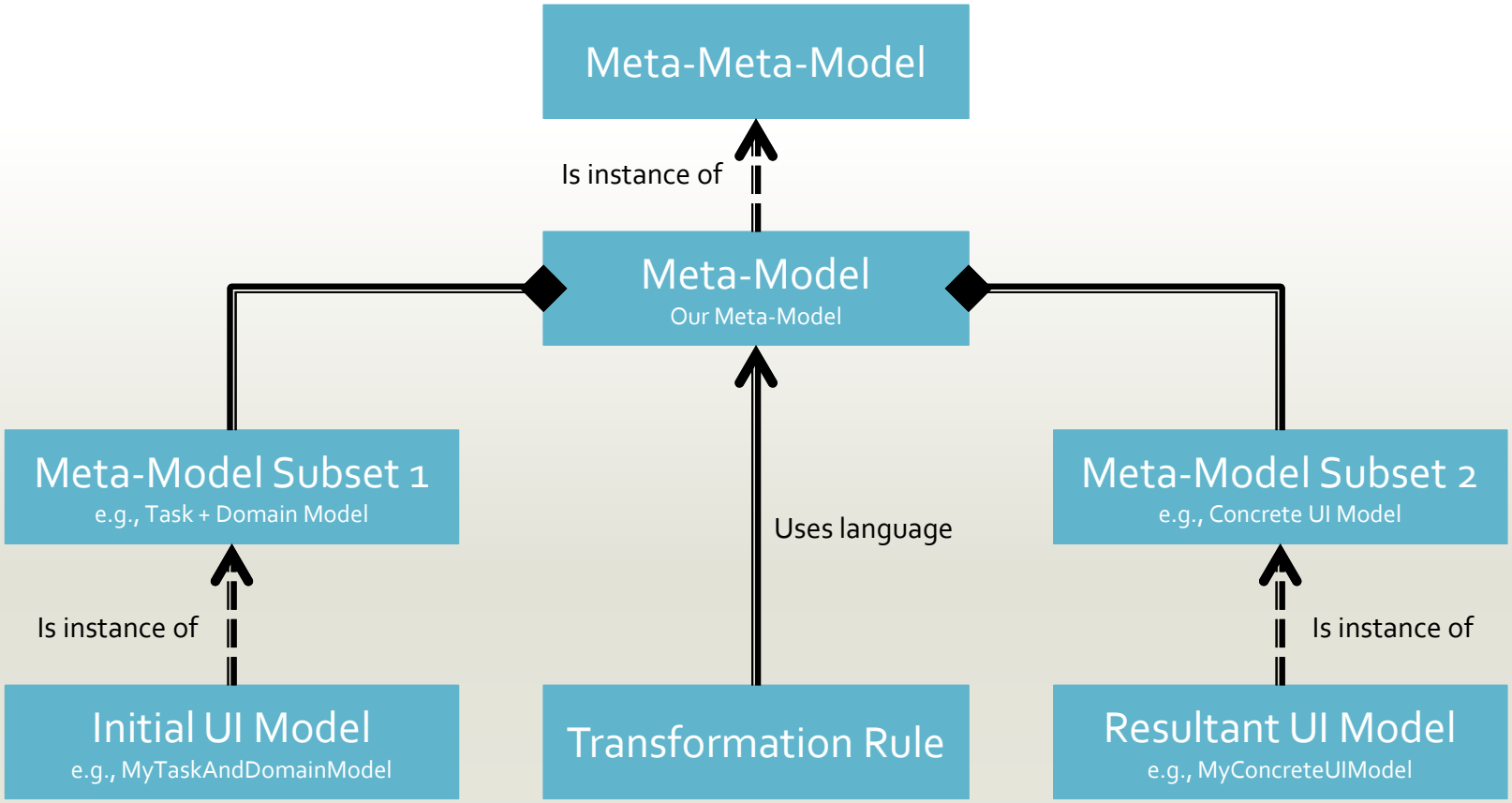
USIXML User Interface eXtended Markup Language

What is UsiXML?

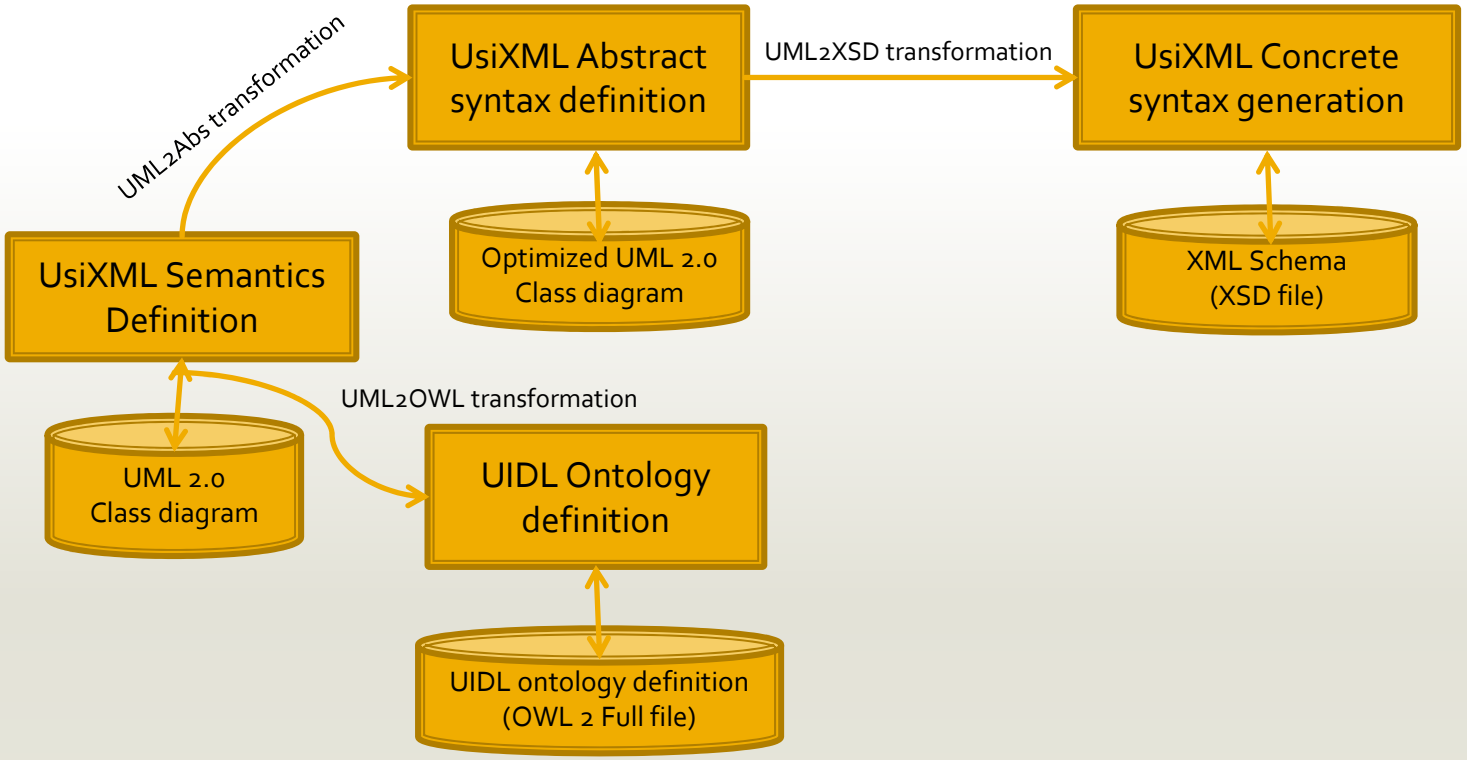
- User Interface eXtensible Markup Language (UsiXML) is a formal Domain-Specific Language (DSL) used in Human-Computer Interaction (HCI) and Software Engineering (SE) in order to describe any user interface of any interactive application independently of any implementation technology.
- A user interface may involve variations depending on: the context of use (in which the user is carrying out her interactive task), the device or the computing platform (on which the user is working), the language (used by the user), the organization (to which the user belongs), the user profile, the interaction modalities (e.g., graphical, vocal, tactile, haptics).

USIXML User Interface eXtended Markup Language

Fact sheet

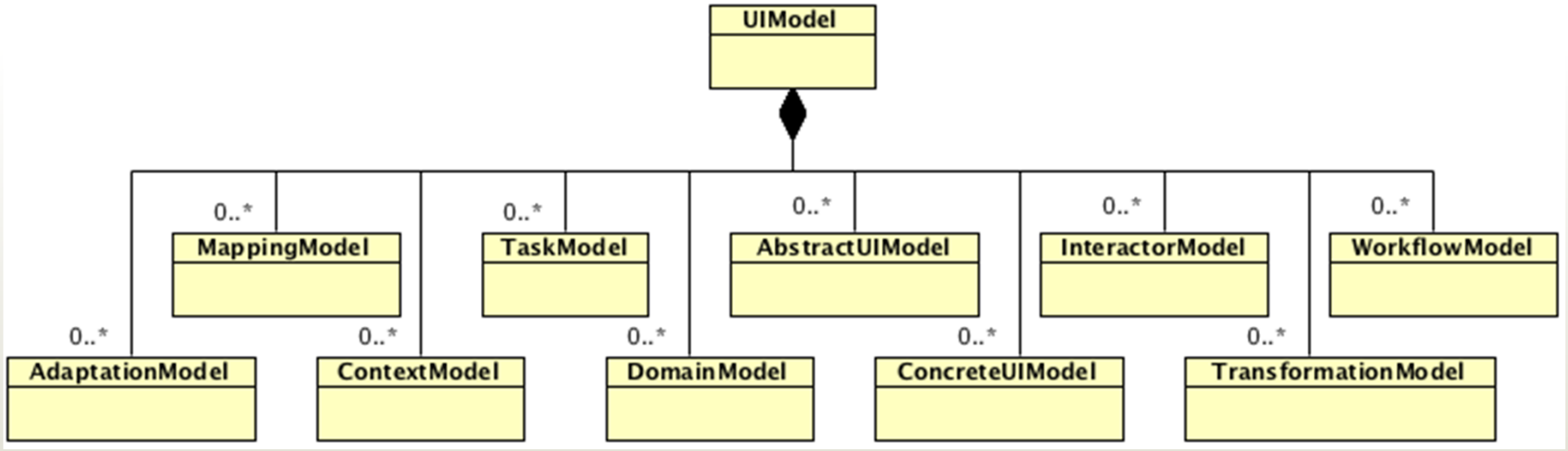


The definition process



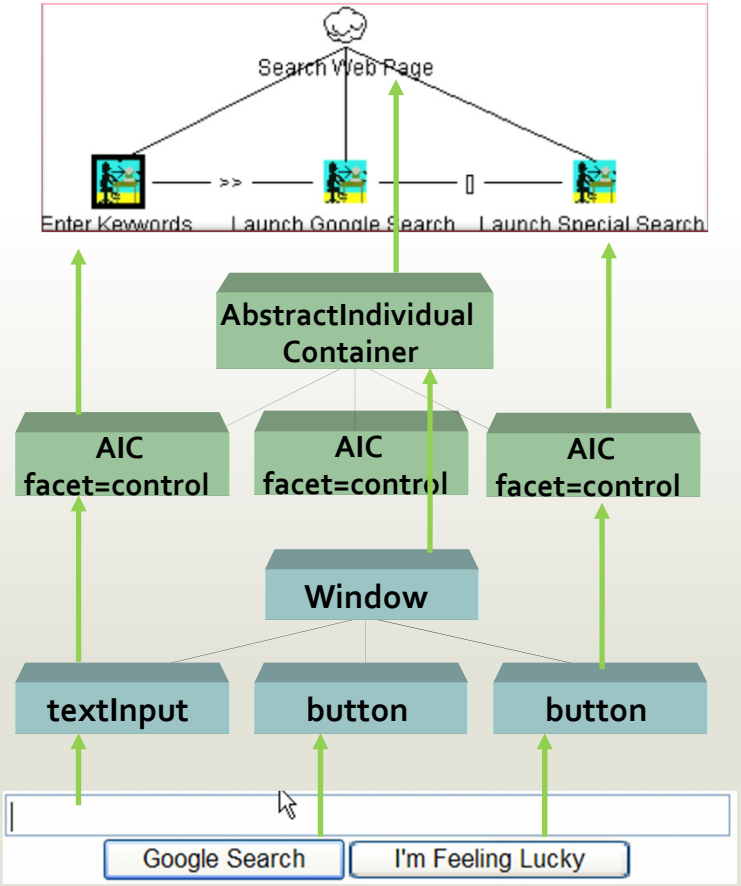
USIXML User Interface eXtended Markup Language

UsiXML Meta-Models



UsiXML – Processes

Cameleon Reference Framework



Task & Domain (T&D)

Abstract User Interface (AUI)

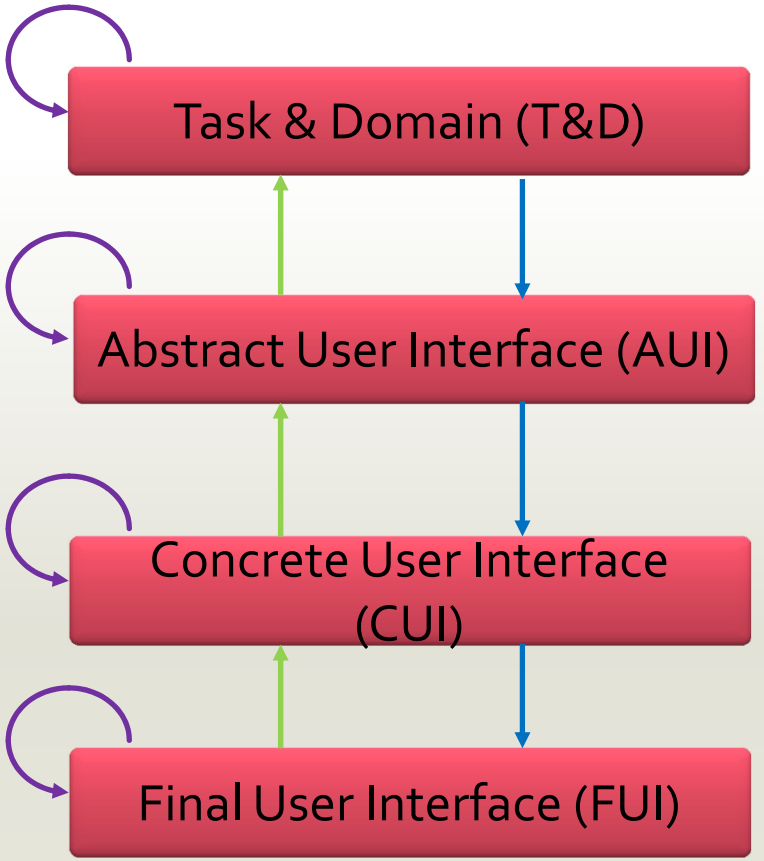
Concrete User Interface (CUI)

Final User Interface (FUI)

USIXML USer Interface eXtended Markup Language

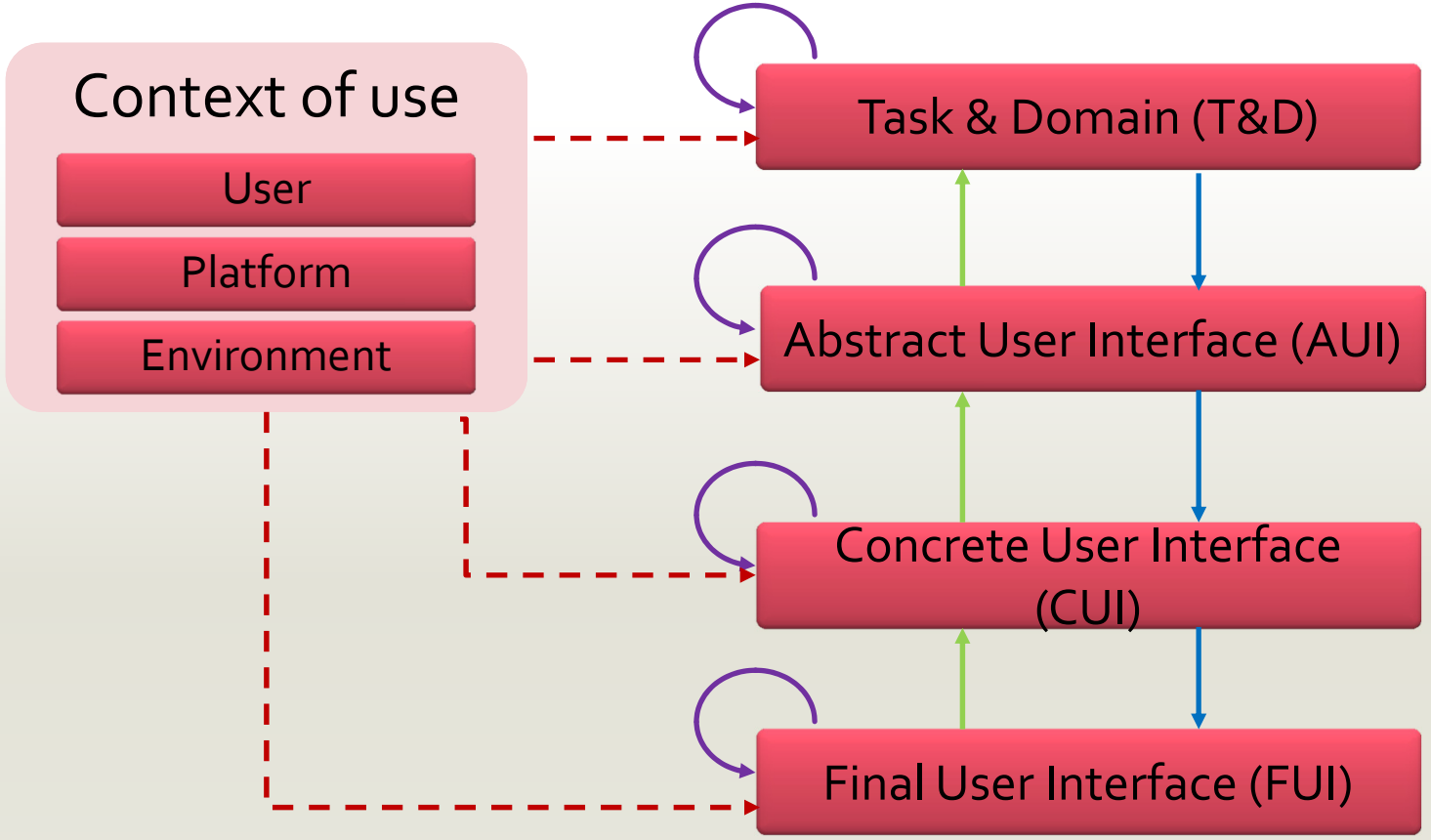
The different relationships

- Reification: from high to lower-level
  - T&D → AUI
  - AUI → CUI: e.g., a GUI, a vocal UI
  - CUI → FUI: e.g., HTML, VB for GUI VoiceXML, X+V for Vocal UI
- Abstraction: from low to higher-level
  - FUI→ CUI: e.g., reverse HTML
  - CUI → AUI: e.g., for changing modality
  - AUI → T&D: e.g., for recovering task
- Reflexion: at the same level
  - FUI: e.g., transcoding
  - CUI: e.g., graceful degradation
  - AUI: e.g., restructuring
  - T&D: e.g., for retasking



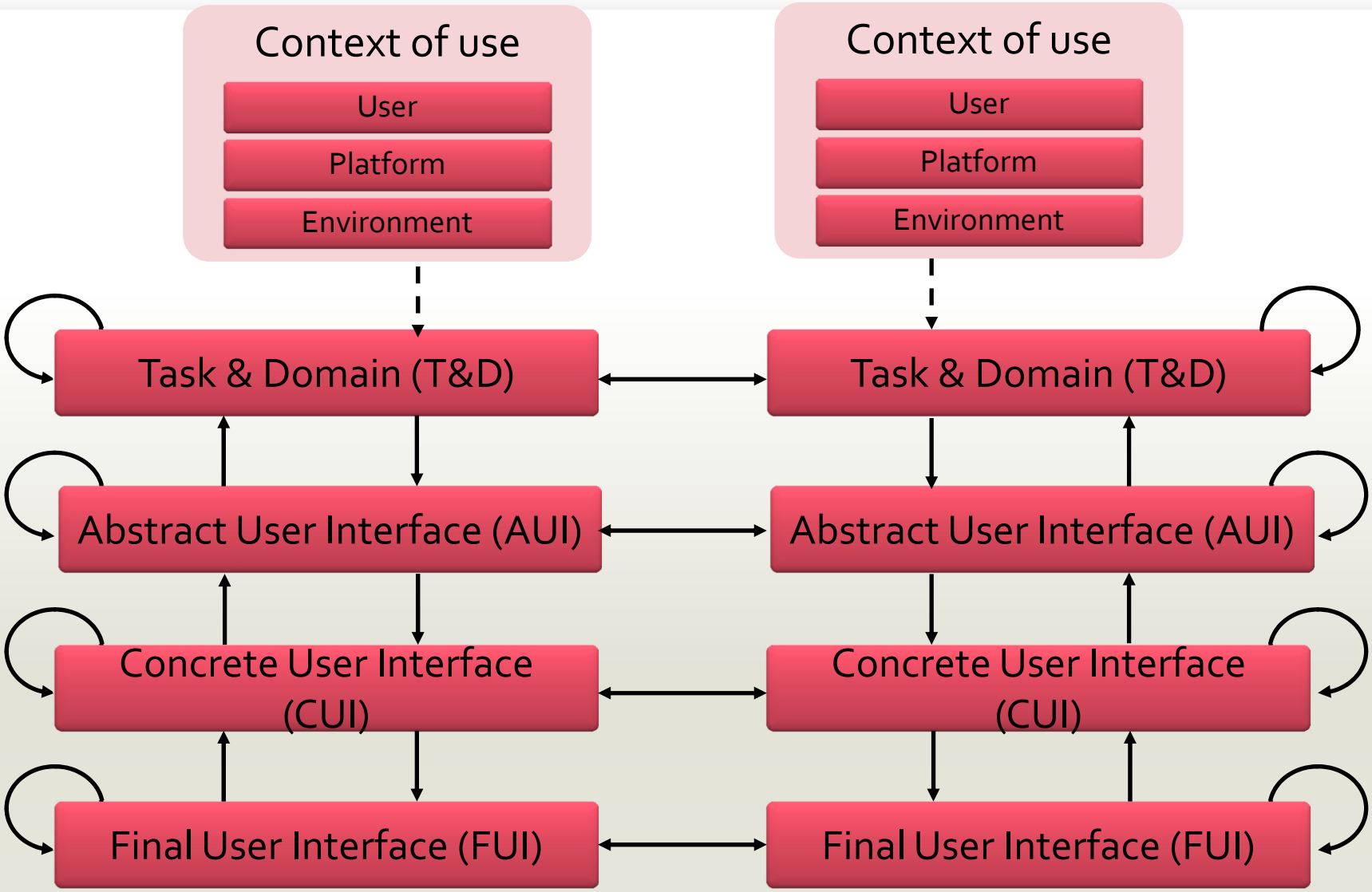
USIXML User Interface eXtended Markup Language

The context of use (user, platform, environment)



USIXML USer Interface eXtended Markup Language

The Cameleon Reference Framework (CRF)



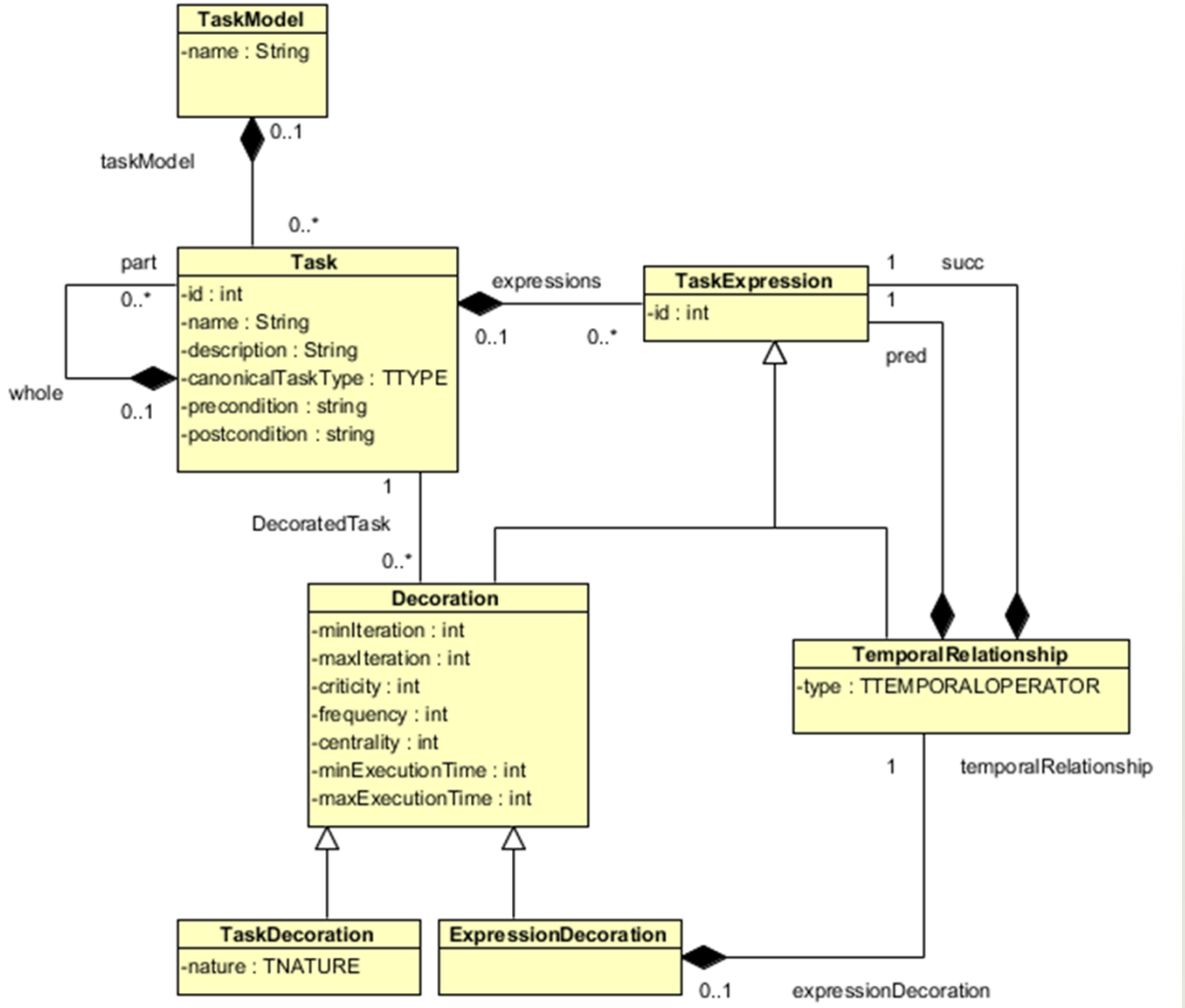
Task&Concepts: Task Meta-Model, model and an example

Requirements

- Several notations exist: HTA, GTA, TKS, CTT, ANSI, MAD, MAD+,…
  - Req1: independence of any notation
- Several levels of details: formal vs informal, static vs dynamic, data-centric vs process-centric
  - Req2: separation of static aspects from dynamic aspects
  - Req3: separation of hierarchical structure from other aspects
- Several expressivities: single user vs multiple users, single platform vs multiple, different contexts
  - Req4: context-awareness (user, platform, socio-organisational environment) and independence wrt context
- Several usages: design, analysis, model checking, requirements
  - Req5: support both informal and formal expression
- Several interpretations: freedom vs guidance, manual vs automation
  - Req6: provide a taxonomy of canonical task types (optional usage)

Task Meta-Model

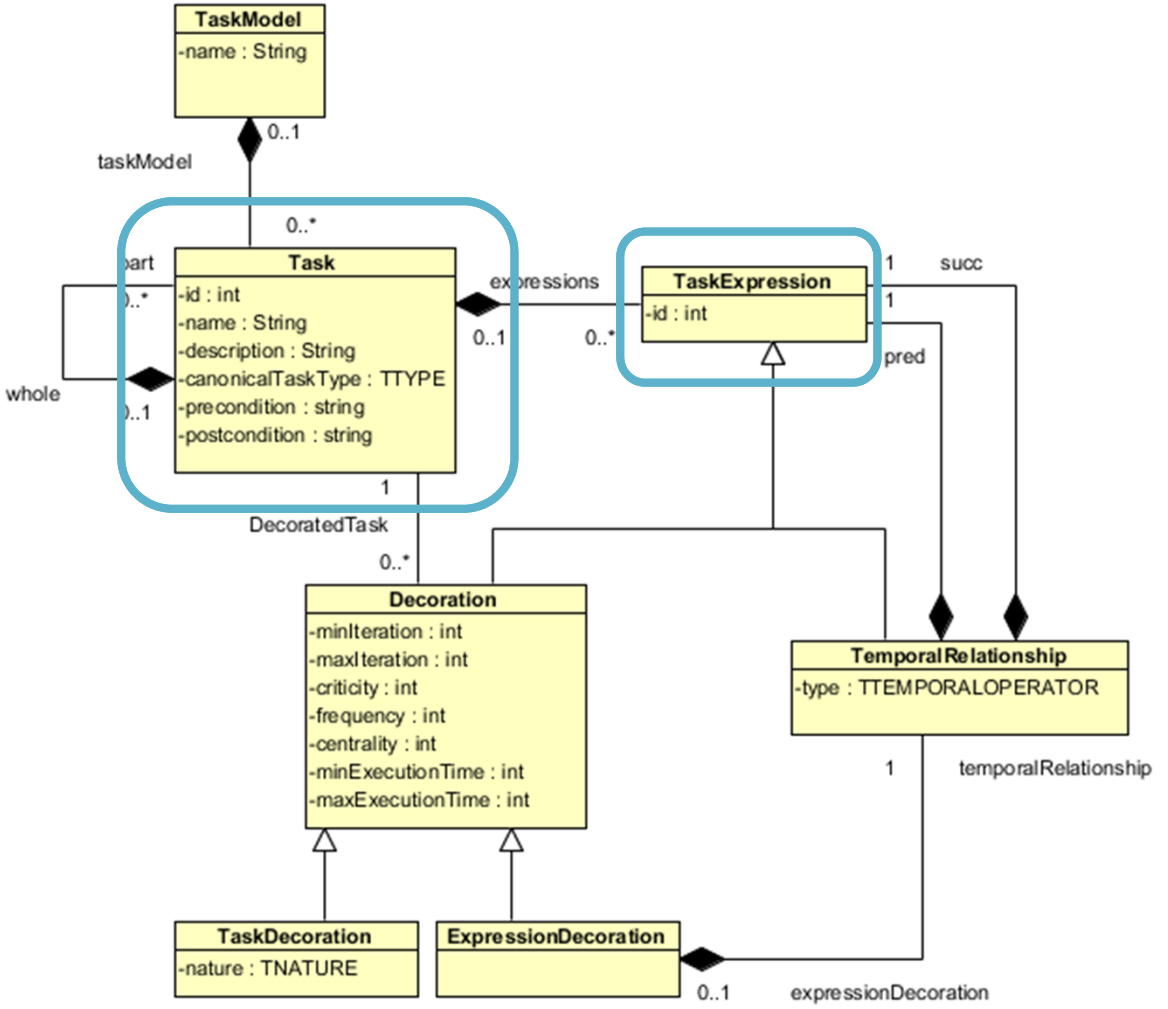
Overview:



Task Meta-Model

Key classes:

- Task
- TaskExpression



USIXML User Interface eXtended Markup Language

Task Meta-Model (Key classes 1/2)

Task:

- The Task meta-class describes a task performed by one or more system entities.
- Example:
  - SignGuestbook
    - id = 1
    - name = “SignGuestbook”
    - description = “Sign the guest book”
    - canonicalTaskType = MODIFY

Task Meta-Model (Key classes 2/2)

TaskExpression:

- The TaskExpression is an abstract meta-class that defines expressions on tasks.
- Example:
  - Decoration of a task (see Decoration class)
  - Expression between two tasks through their TaskExpression

USIXML User Interface eXtended Markup Language

Task Expressions

2 types:

- Unary: Decoration
- Binary , n-ary: TemporalRelationship

2 types of Decoration:

- TaskDecoration
- ExpressionDecoration (inheritance of the ExpressionDecoration to all the tasks involved in the TemporalRelationship)

USIXML User Interface eXtended Markup Language

Task Expressions (1/2)

Decoration:

- The Decoration is an abstract meta-class that allows developers to define attributes that vary according to the situation they are performed (i.e. criticity, frequency, etc.).

2 types of Decoration:

- TaskDecoration
- ExpressionDecoration (inheritance of the ExpressionDecoration to all the tasks involved in the TemporalRelationship)
-

Task Expressions (2/2)

TemporalRelationship:

- The TemporalRelationship meta-class that allows developers to define the temporal relationship among TaskExpressions.
- TemporalRelationships are temporal operations defined as LOTOS operators on TaskExpressions.

USIXML User Interface eXtended Markup Language

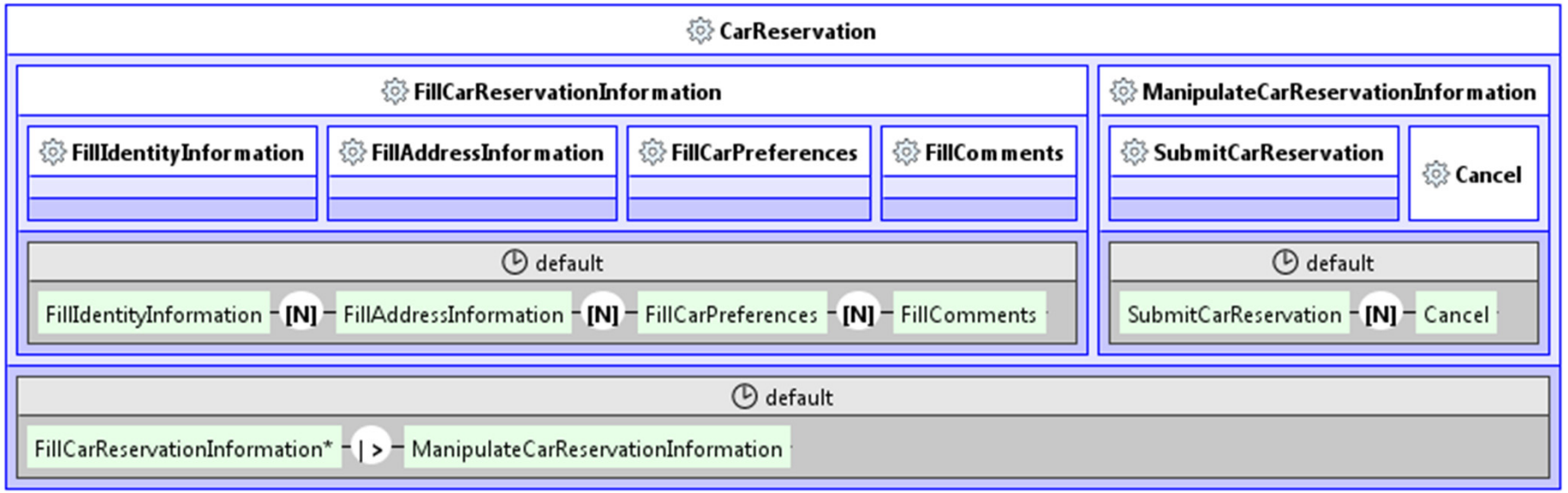
Task Model

- Task Model: an instantiation of the Task Meta-Model
- Use of an example: a reservation of a car:
  - Filling information for the reservation:
    - Identity, address and car preferences
    - Add some comments
  - Manage the information:
    - Modify
    - Submit the reservation
    - Cancel the reservation

USIXML User Interface eXtended Markup Language

Example: car reservation

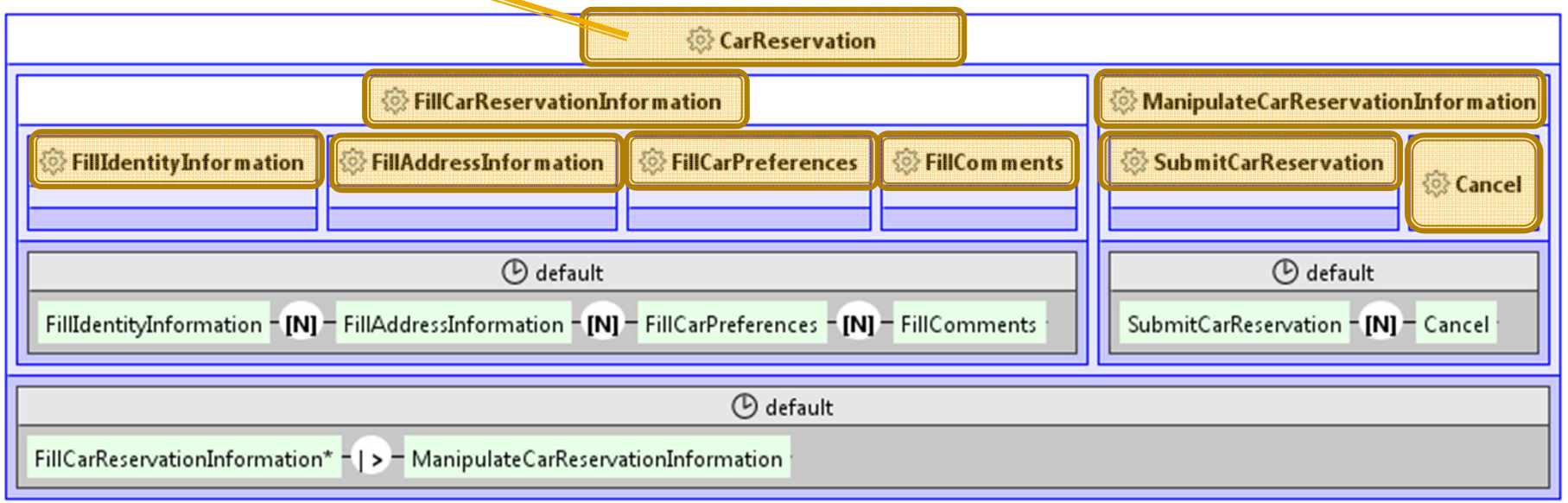
Task Model:



USIXML User Interface eXtended Markup Language

Example: car reservation

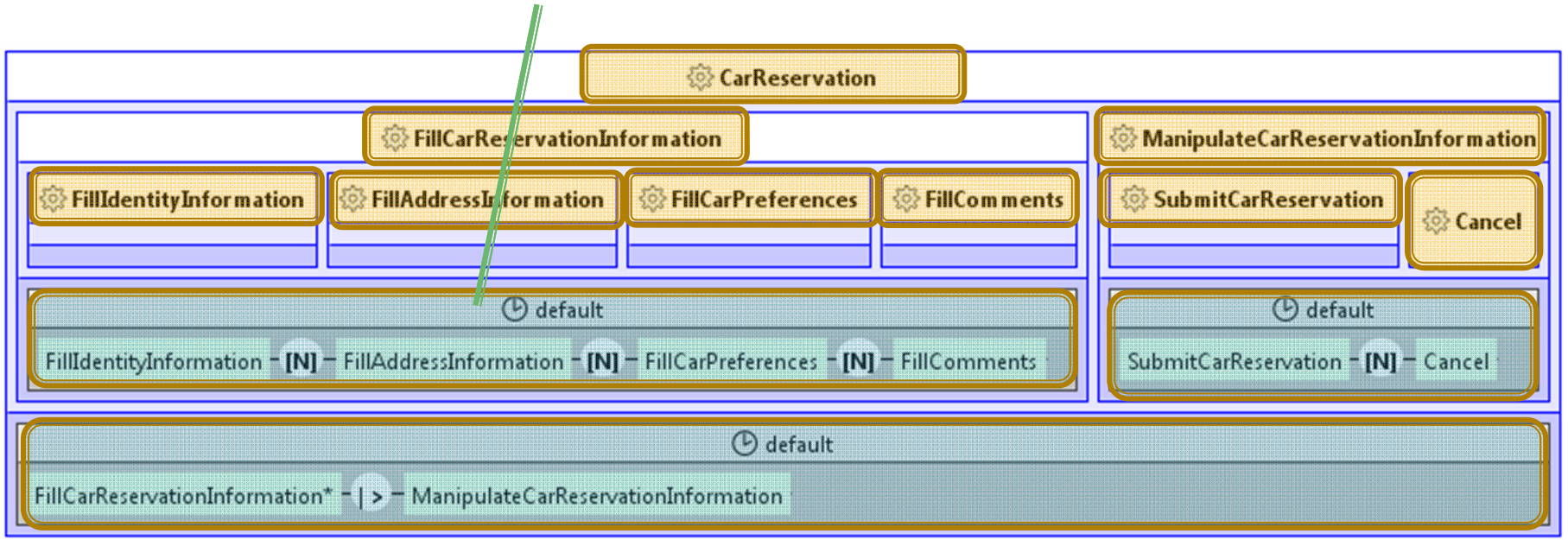
Tasks:



USIXML User Interface eXtended Markup Language

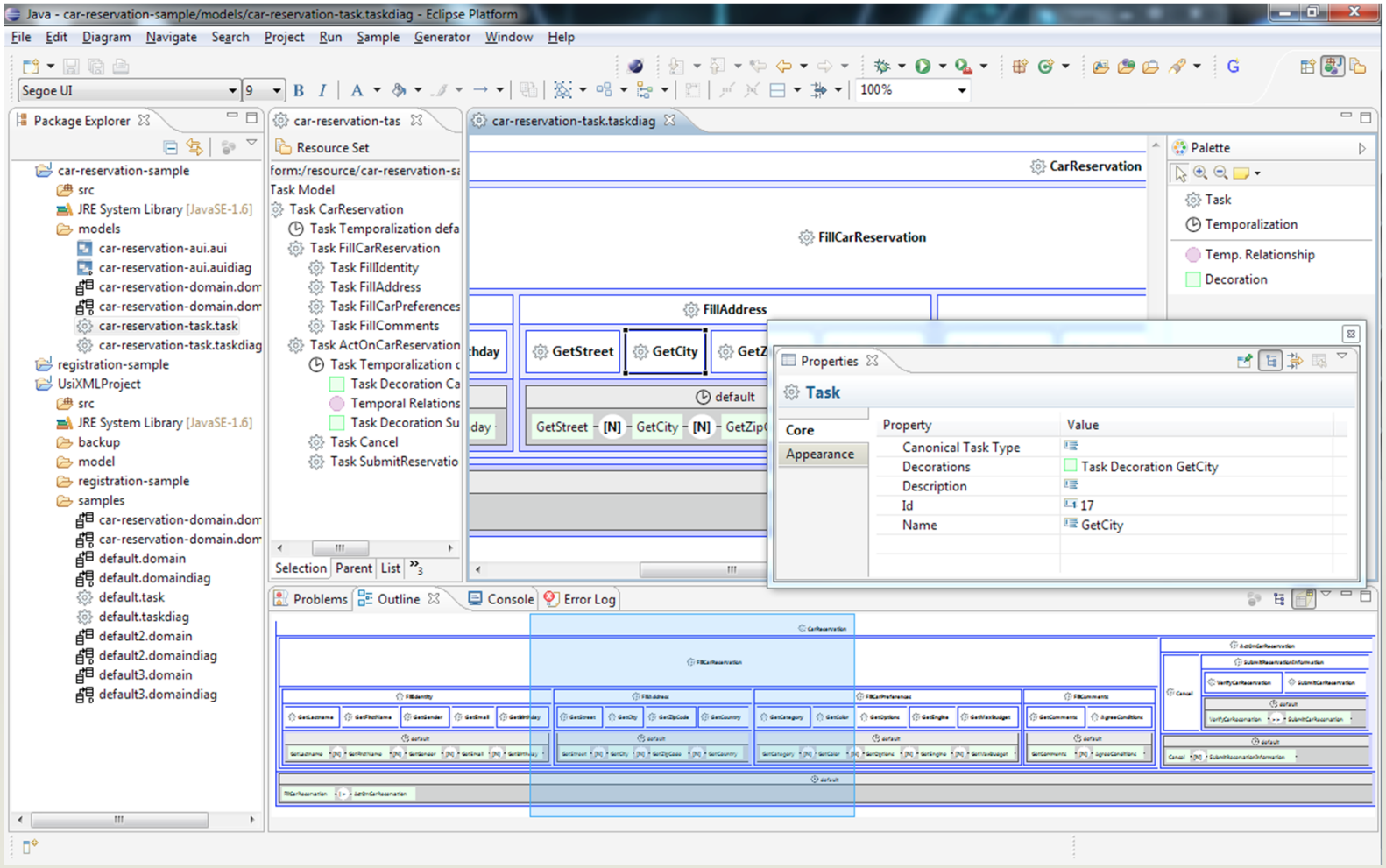
Example: car reservation

TaskExpressions:



USIXML User Interface eXtended Markup Language

Eclipse tool



USIXML User Interface eXtended Markup Language

Videos

- Car Reservation: structure
- Car Reservation: Behavior