Supporting the Adaptive Generation of Learning Game Scenarios with A Model-Driven Engineering Framework

Pierre Laforcade, Youness Laghouaouta
Computer Science Laboratory of Le Mans University, France

EC-TEL 2018, Leeds (UK)

Outline

Research context & problem : the *Escape It!* project

Research proposition: an MDE approach for dealing with the generation of adapted scenarios and co-design process for the serious game

Application: the *Escape It!* serious game

Conclusion and future work

Research context: the *Escape it!* project

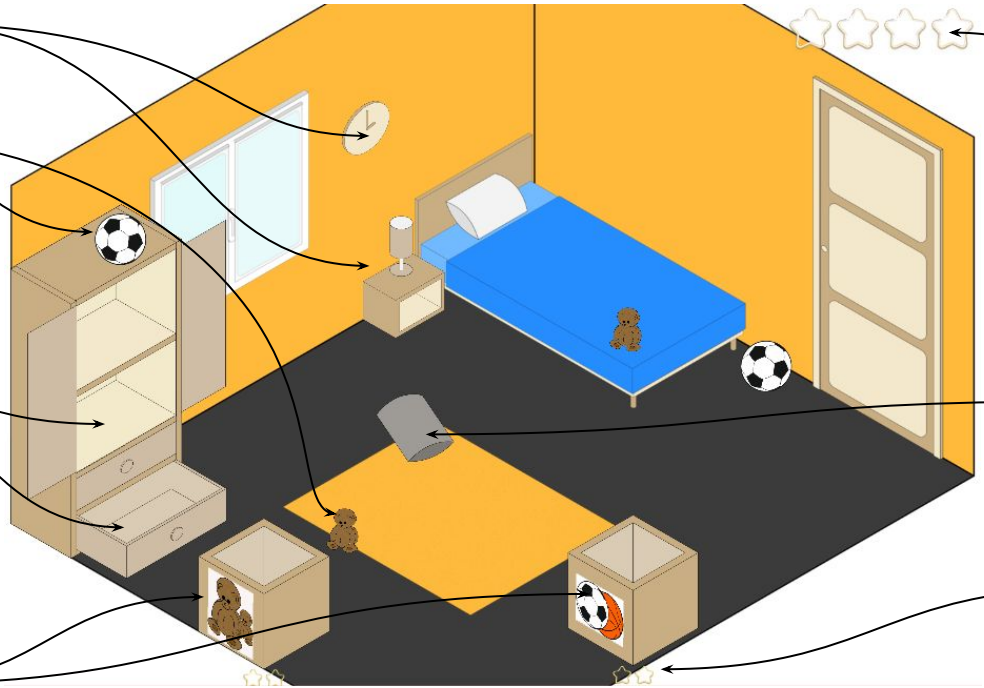
- **Objective**
  - To Develop a mobile **serious game** to train **visual skills** for children with Autism Spectrum Disorder
    - B3 skill = matching an object to an identical object,
    - B8 skill = sorting several objects into different categories
    - ...
  - Mechanics from ”escape-room” games (opening a locked door to escape a room)

children have to solve numerous puzzles requiring observation and deduction

game scenario = ordered sequence of *scenes*

scene = description of components / locations / other informations required for the *game engine* at *runtime*

Research context: Anatomy of a scene



additional decor object

objects to move to the good solution place

additional decor object with interactive areas to hide objects

Solution objects with information according to the targeted skill

feedback about the progression for achieving the scene

disturbing movable objects
(not visible within this example)

interactive hideout objects possibly hidding an object

local feedback about completion of a solution object

From Design problem to Research problem

1. Every new *game session* has to be **adapted** to the **child's profile**

Learner Model variables
- *skills to consider*
- *level difficulties for every skill*
- *preferences/dislikes,*
- ...

2. The *game* has to propose a large variety of scenes **settings** in order to support the **generalization** of skills

Scene variables
- *additional decors*
- *hideouts*
- *possible locations for movable objects*
- *Movable/hideable objects from several categories*
- ...

Design issue

Cost to design and implement all combinations

Design problem

How to dynamically generate new game sessions, adapted to the child, using the available game components?

Research problem

How to generate learning scenarios that are adapted to children' profiles (learner model) and take into account the game knowledge?

Research problem: revised

How to generate learning scenarios adapted to children' profiles (learner model) and taking into account the game knowledge?

R1 How to **make explicit and well defined** the domain components (skills, game knowledge, learner model), mapping and generation rules?

R2 How to use these information to **drive the generation** of adapted learning scenarios?

R3 How to **involve domain experts** in the design and the validation of the serious game?

Outline

Research context & problem : the *Escape It!* project

Research proposition: an MDE approach for dealing with the generation of adapted scenarios and co-design process for the serious game

Application: the *Escape It!* serious game

Conclusion and future work

Research proposition: rationale

- **General idea**

Adapt the GOALS architecture from [Sehaba & Hussan, 13]

+

Focus on a Model-Driven Engineering (MDE) framework

a research domain promoting an active use of models throughout the software development process, leading to an automatic generation of the final solution

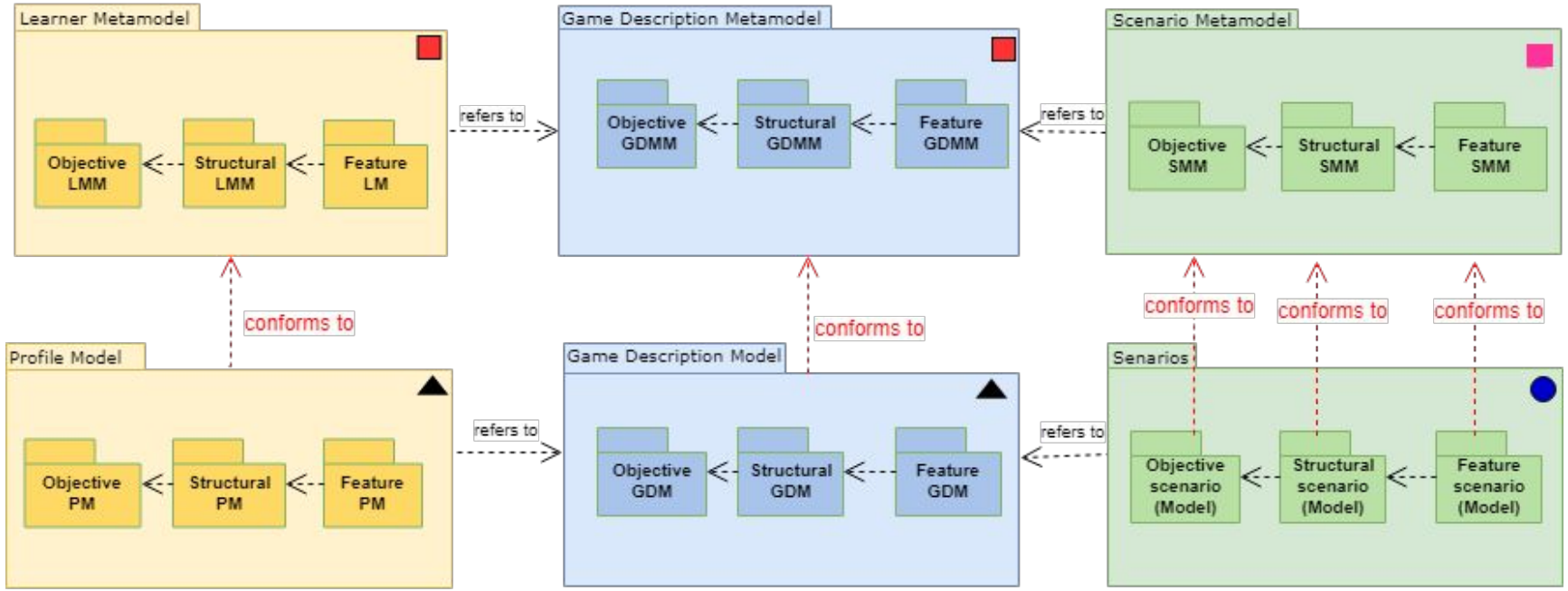
- **Research hypothesis**

- The game knowledge (skills, components, etc.), the learner profile can be expressed as active models. → R1
- Learning scenarios can be automatically generated by means of model transformations → R2
- Models in high level of abstraction for being conjointly managed by computer scientists and domain experts. → R3
- Achieve quick feedback thanks to the automatic propagation of changed. → R3

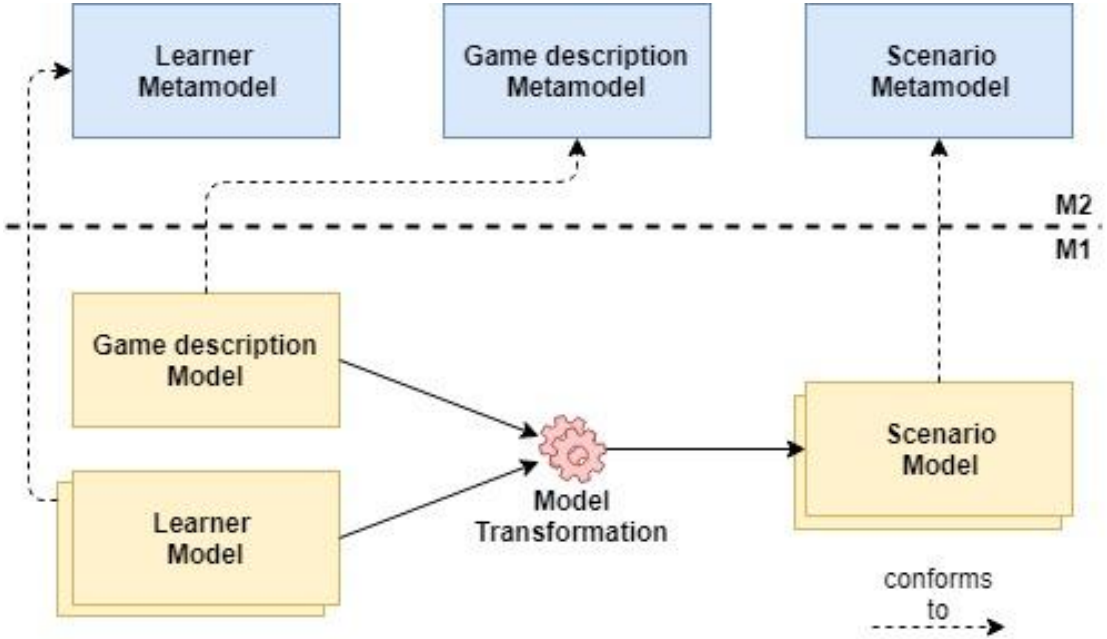
Research proposition: overview

- A 3x3 metamodel-based specification architecture
  - 3-incremental perspectives on the resulting scenarios:
    - objective scenario = selection of the targeted learning objectives according to the user's profile
    - structural scenario = selection of learning game exercises or large game components
    - features scenario = selection of the inner-resources/fine-grained elements
  - 3-dimensions specifications of domain elements to be managed
- Automatic generation of adapted scenarios
- A dedicated MDE based process to co-design the serious game
  - Involve the domain experts in the design and validation activities
  - Rapid prototyping support to immediately test playable version of the game and give relevant feedback

The metamodeling architecture: 3 dimensions



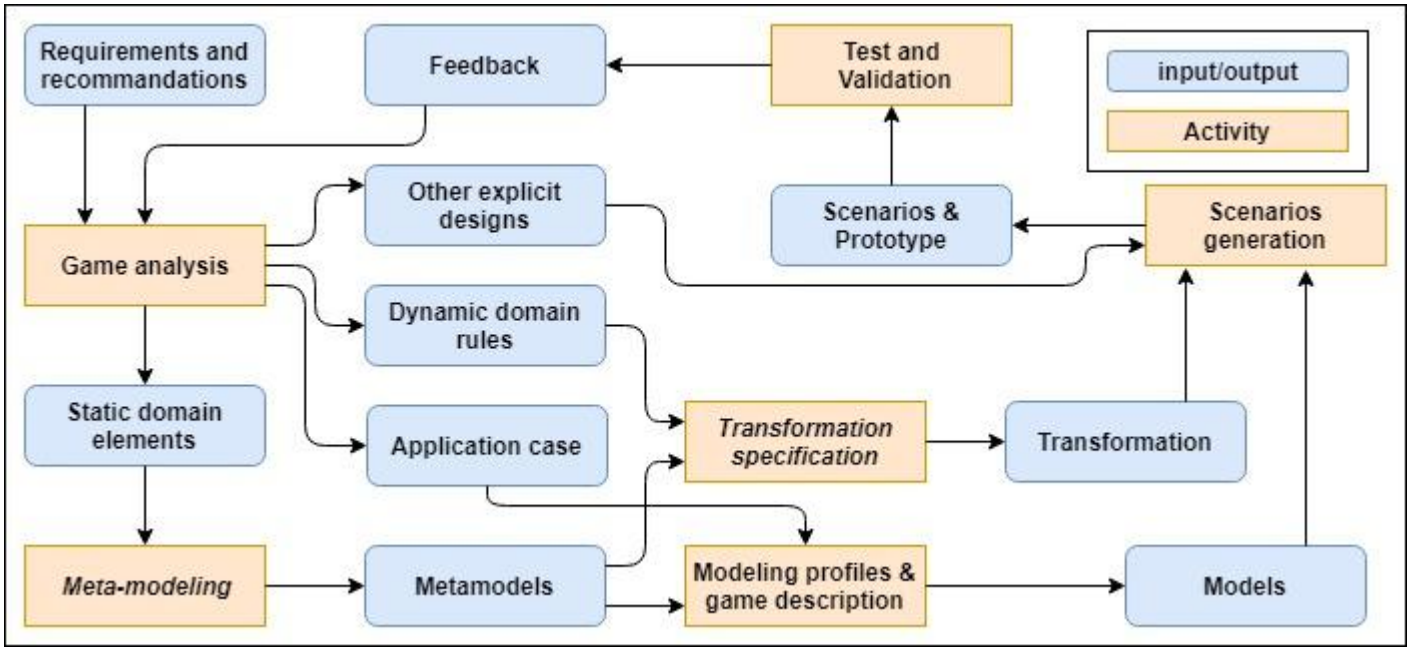
Generation of adapted scenarios



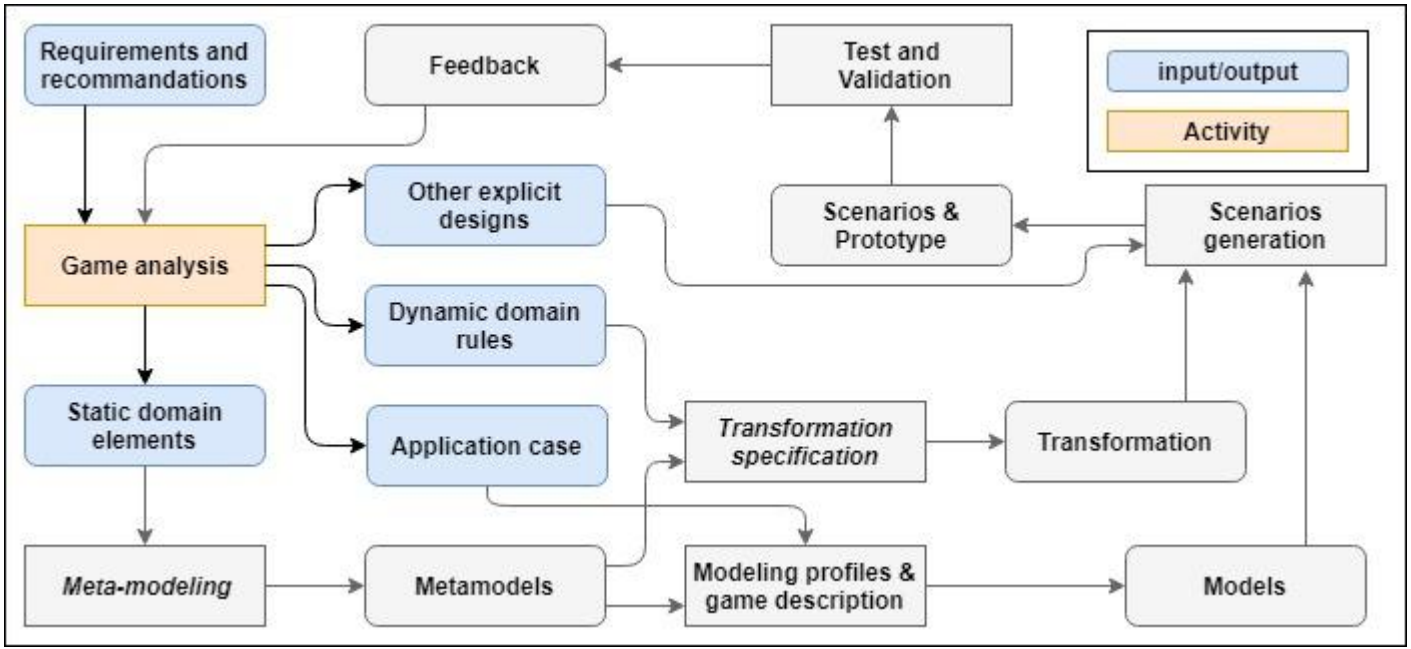
Generation of adapted scenarios: characteristics

- **Online**:
  - the scenario is generated during the runtime
- **Necessary**:
  - the content has to be correct
- **Parameterized**:
  - the model transformation takes as an input the game description model
- **Stochastic**:
  - randomness is used when several combinations are possible
- **Constructive**:
  - it never produces broken content

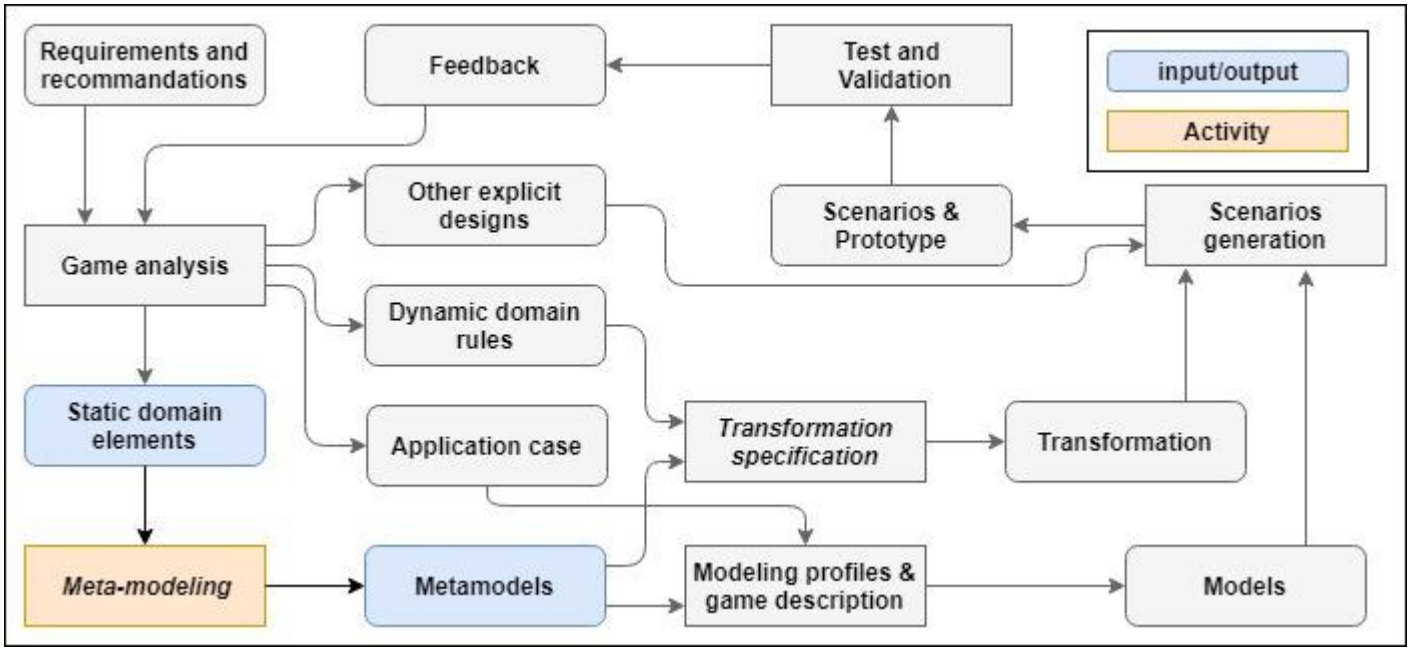
The co-design process



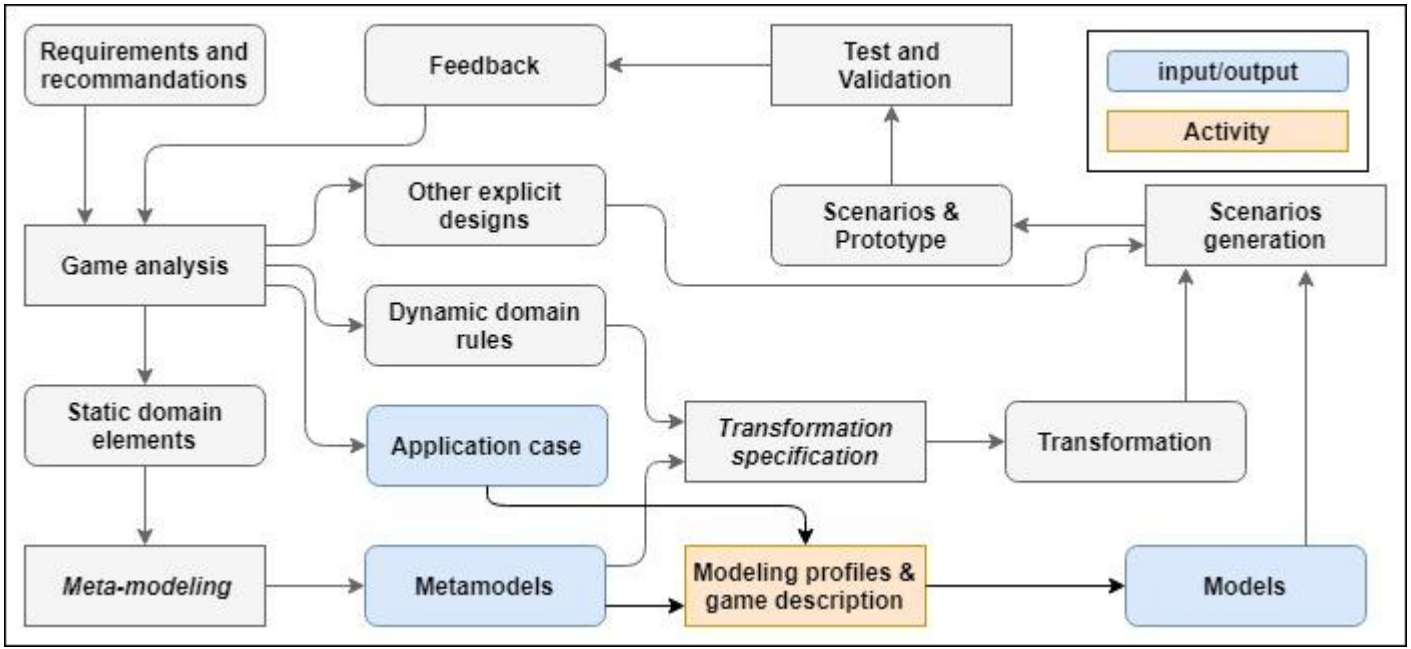
The co-design process



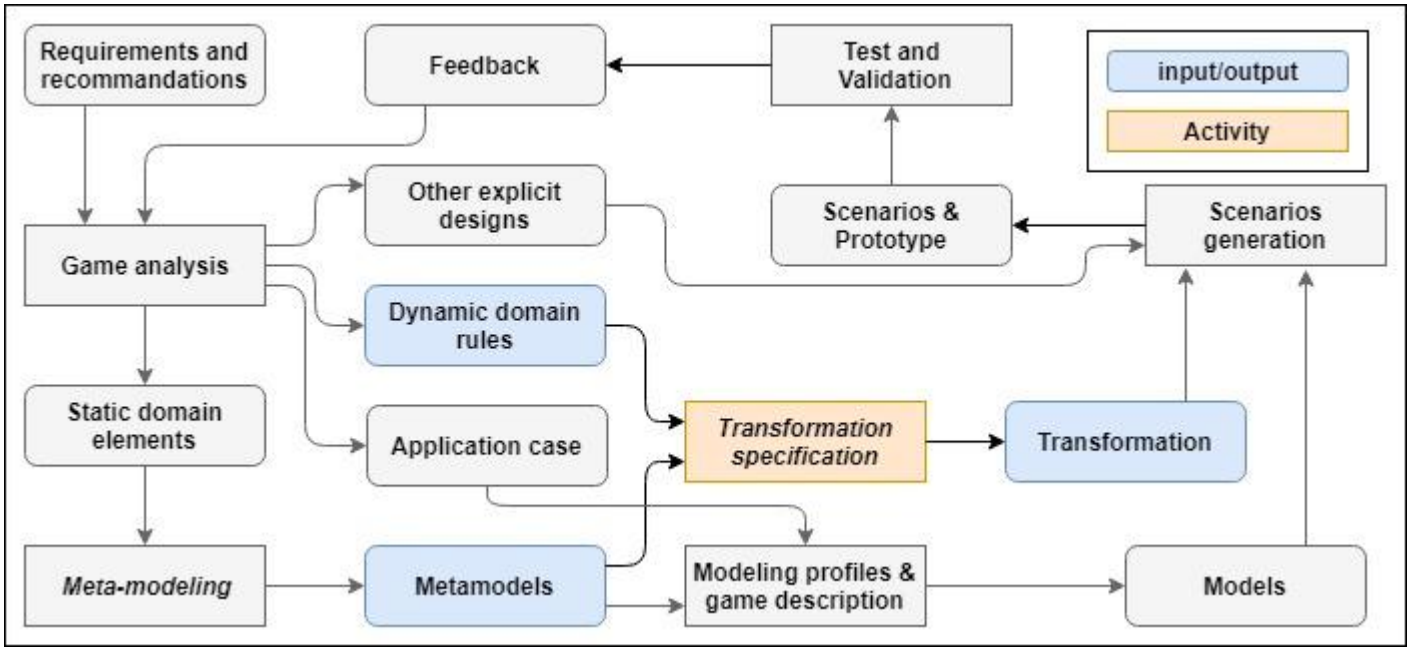
The co-design process



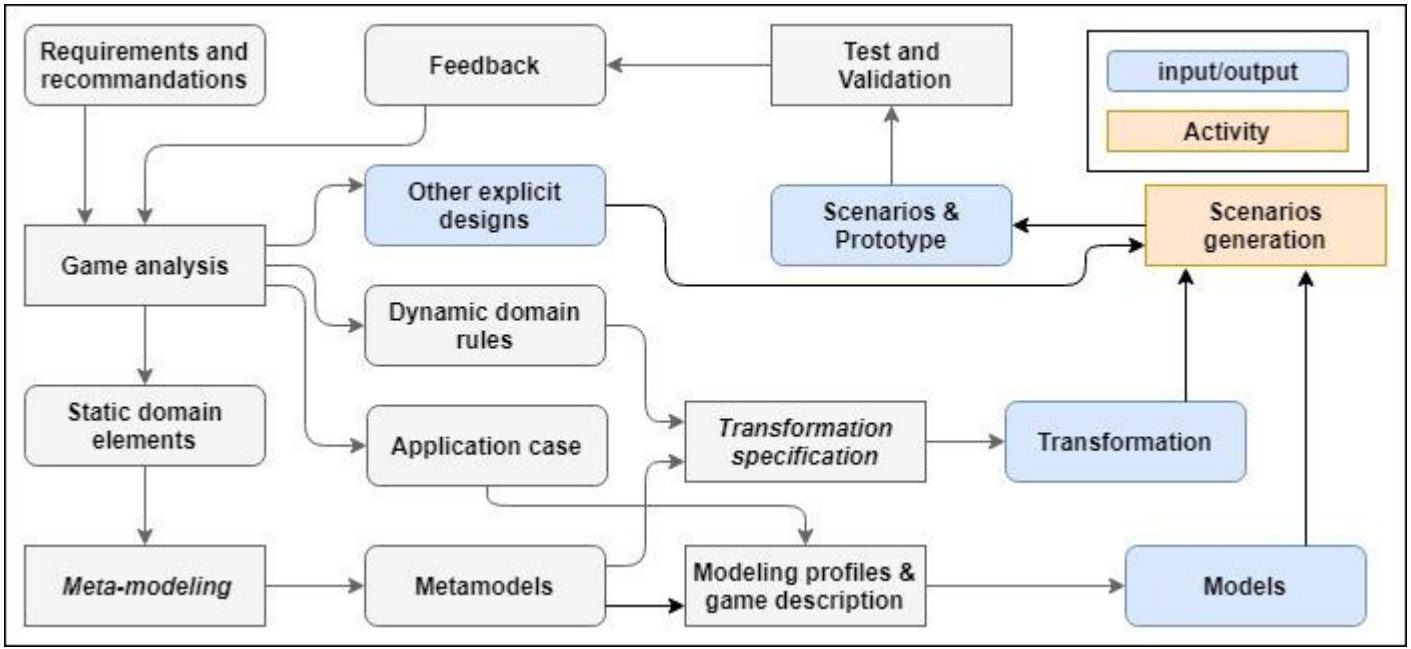
The co-design process



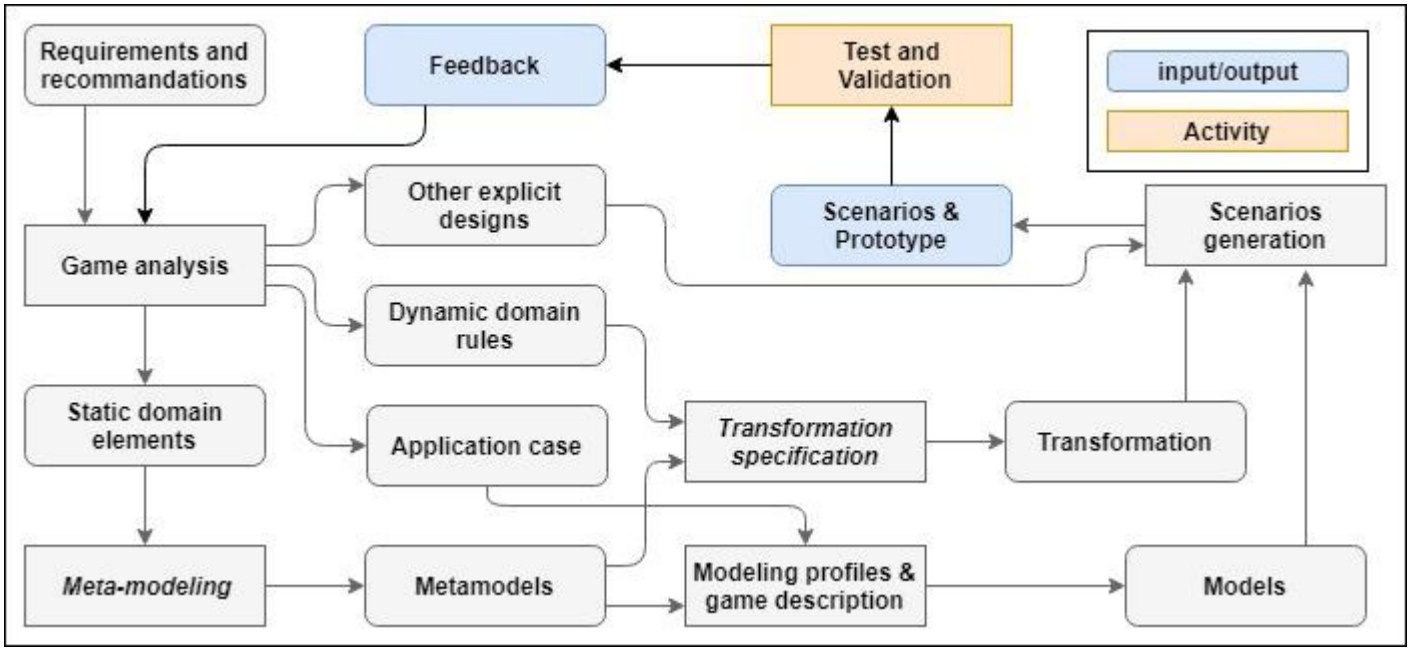
The co-design process



The co-design process



The co-design process



The co-design process : Iterations

- Basic interactions with respect to the three incremental scenarios:
  - Objective scenario
  - Structural scenario
  - Game Scenario
- Re-engineering iterations based on analysis of the generated scenarios (essentially the game scenario)
  - Misunderstandings within the interdisciplinary team
  - Misconceptions about the mapping and generation rules

Outline

Research context & problem : the *Escape It!* project

Research proposition: an MDE approach for dealing with the generation of adapted scenarios and co-design process for the serious game

Application: the *Escape It!* serious game

Conclusion and future work

Application: overview

- Concerns one co-design iteration of the *Escape it!* game
- Structured according to the different co-design activities
- Focuses on the co-design activities rather than how model transformations are implemented

Application: Game Analysis

- Detailed description of each supported scene
  - Objects to place, hiding elements, solution objects...
- Domain rules to apply when generating a scenario

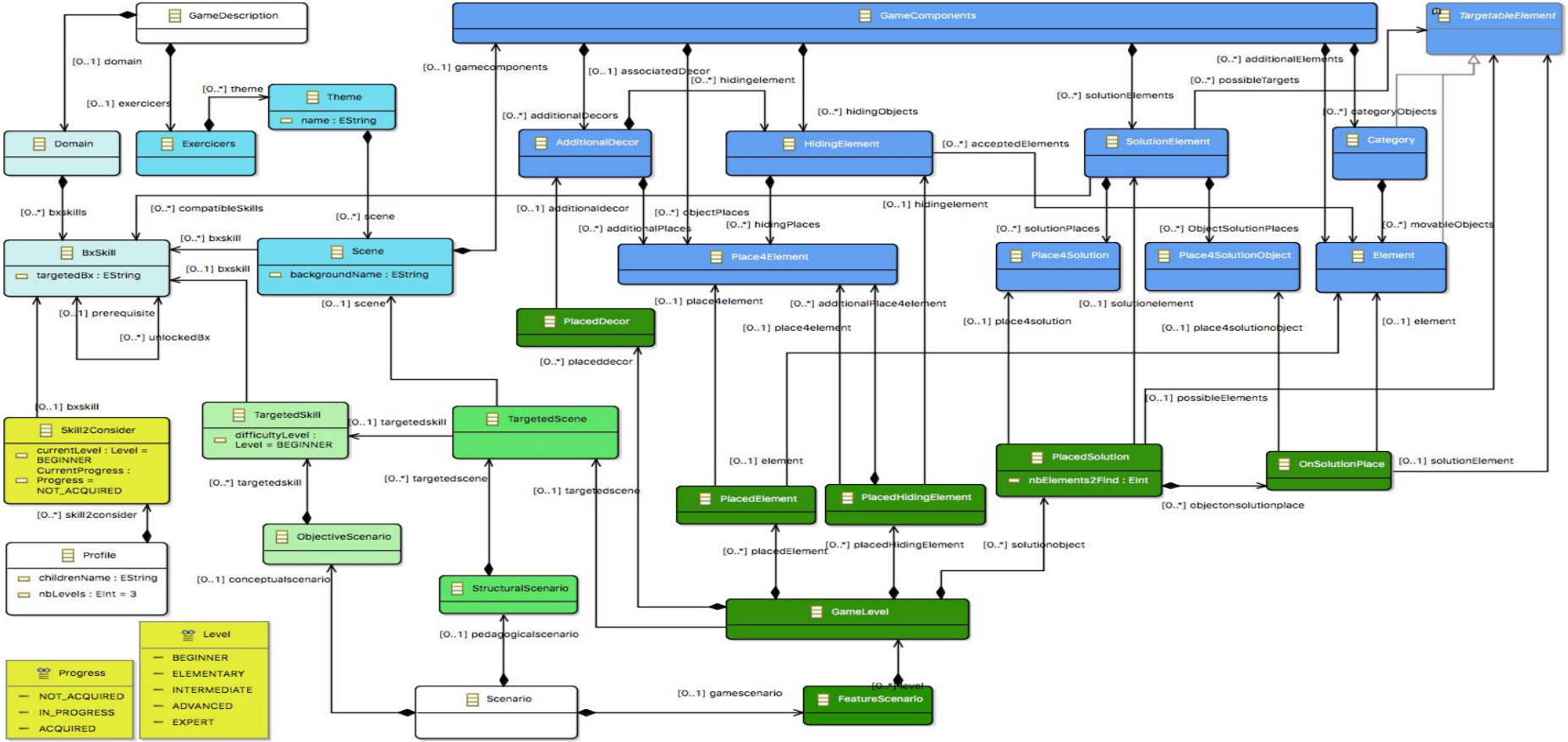
	Game description	User profile	Generation rules for scenarios
Objective scenario	-visual skills to acquire. -*dependency* relations between skills.	-acquired or in progress skills. -their difficulty level. -number of levels to generate.	-only skills with *parents* at '*Intermediate*' level or higher are eligible. -80% of targeted skills with a difficulty level less than '*Intermediate*'.
Structural scenario	-themes and associated scenes. -skills targeted by each scene.	-themes/scenes to exclude/favor according to child's preferences/dislikes. -history of proposed scenes.	-generate different scenes from the same theme.
Feature scenario	-background elements, hiding objects, available object places of each scene.	-scene objects to exclude/favor according to child's preferences/dislikes. - objects involved in previous sessions.	-mappings between each difficulty level and the objects to select and place into the scene.

Application: Game Analysis

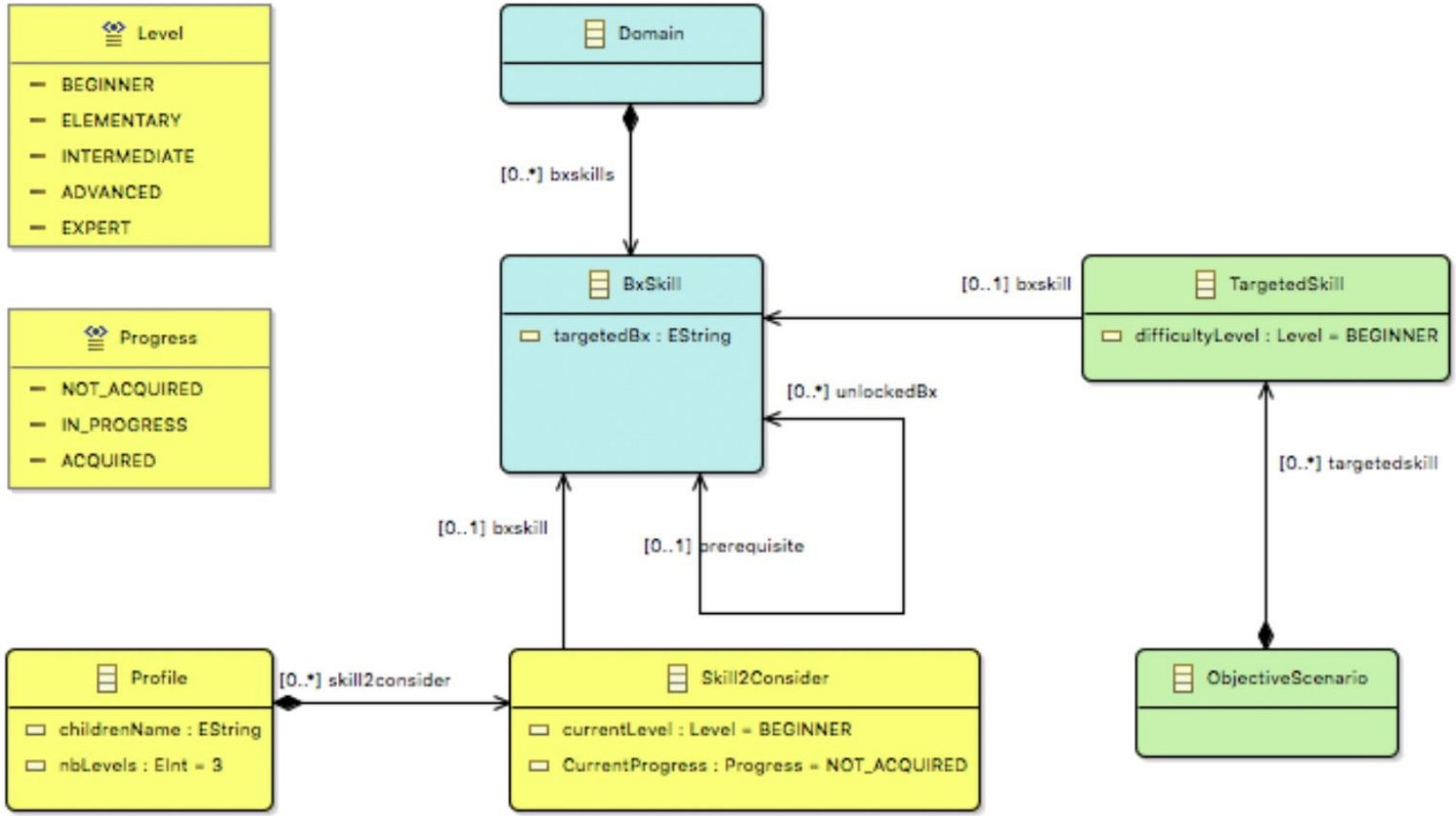
- Mapping rules to guide scenes construction according to the difficulty level

	Additional decors	Hidding objects	Number-of-objects-to-place indicator on solution objects	All movable objects are parts of the solution	Range of objects to find
Beginner	no	no	yes	yes	low
Elementary	yes	no	yes	yes	low
Intermediate	yes	yes	yes	yes	medium
Advanced	yes	yes	no	yes	medium
Expert	yes	yes	no	no	large

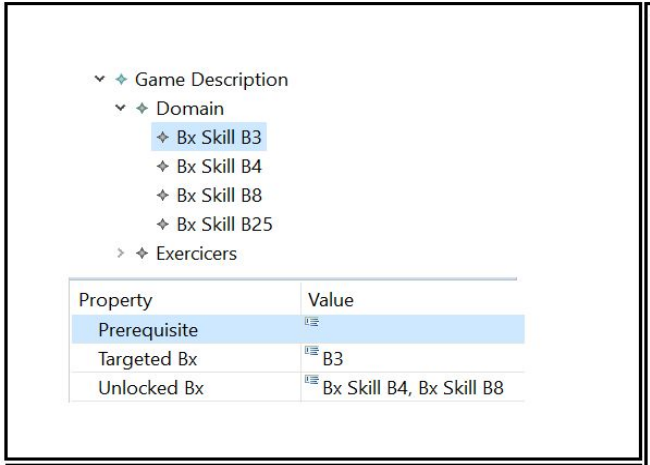
Application: Metamodeling

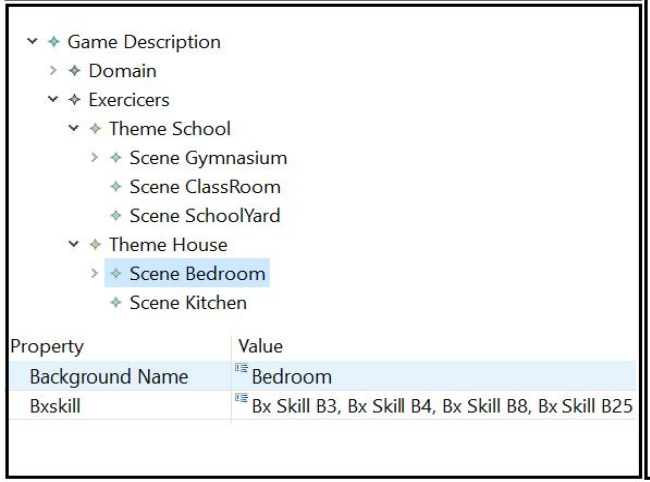


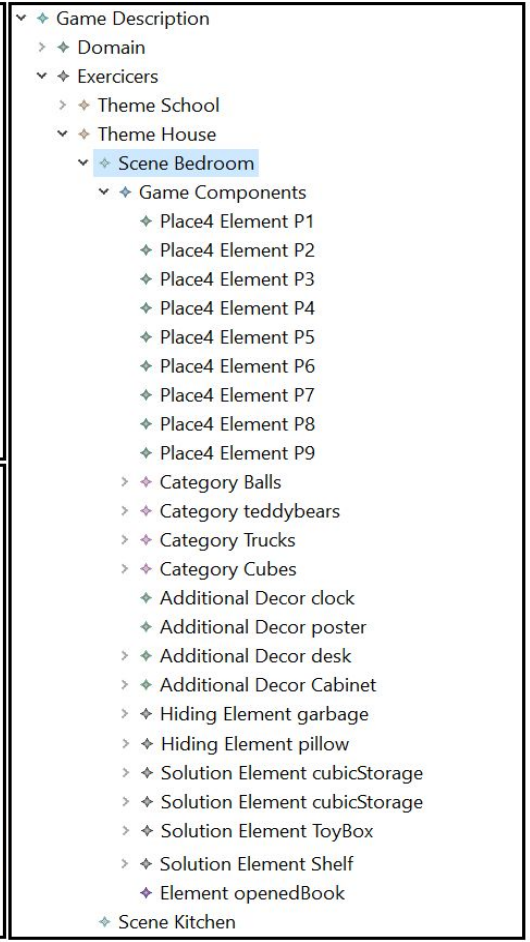
Application: Metamodeling



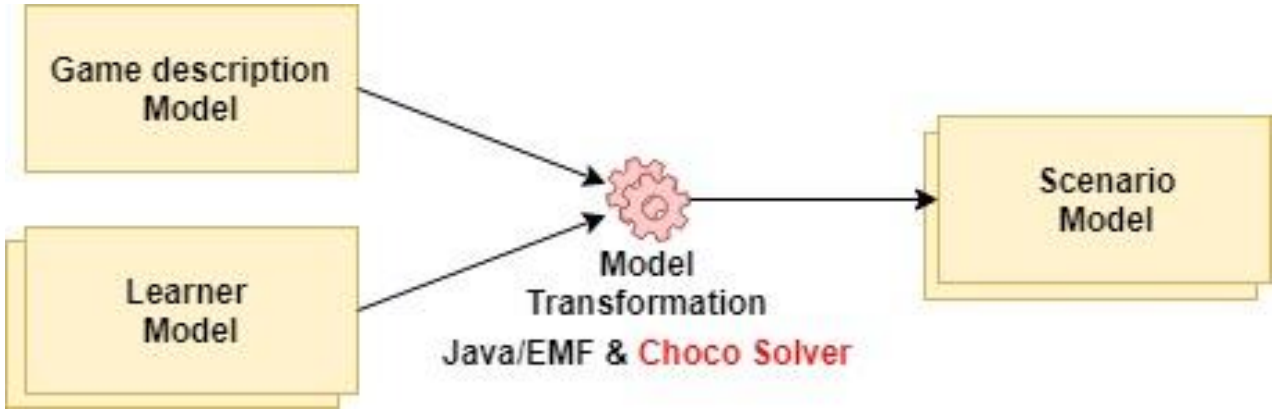
Application: Modeling profiles and game description







Application: Transformation specification



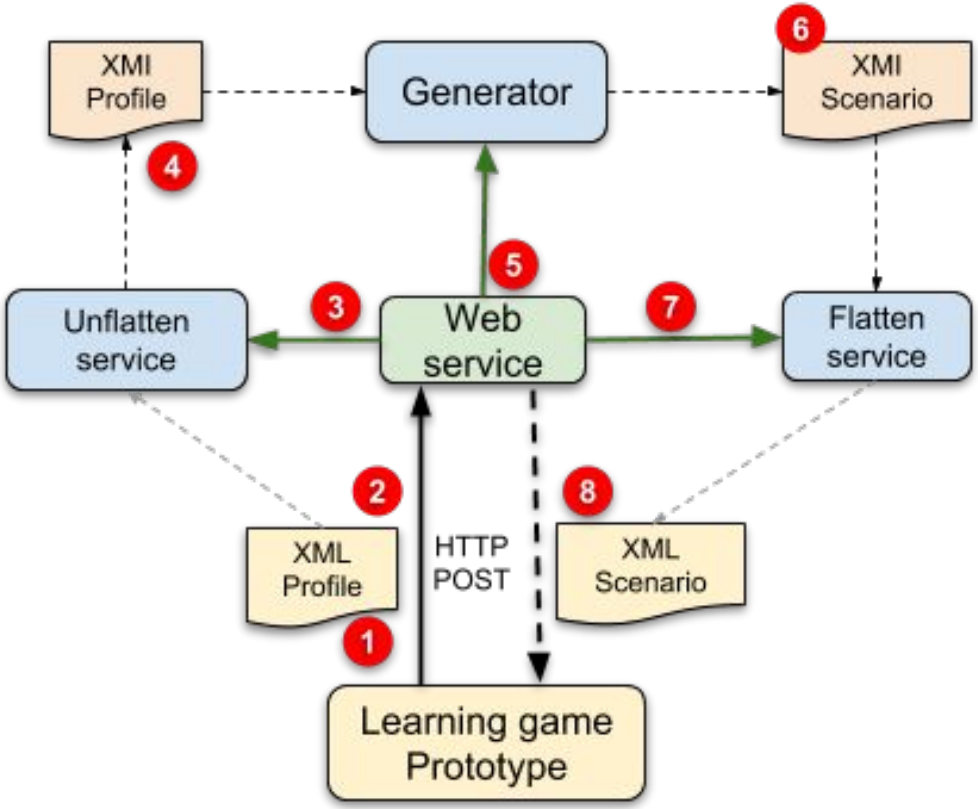
Techniques

- ➔ ***Java/EMF*** to implement the model transformation
- ➔ ***Choco solver*** to solve constraint satisfaction problems

Example of generation rules

- ➔ **IF N** (number of levels to generate from child profile) **< NBS** (nb of skills from game desc. model) **THEN** pick up randomly N different skills **ELSE** pick up randomly N different skills, when no more skills start again with initial set of skills

Application: Transformation execution



Application: Test and validation

- Collective validation session with 2 ASD experts
  - Exploit different generated scenarios and examine them at the three proposed levels (objective, structural and feature)
  - Use the game prototype integration support to test the corresponding playable scenes
- Feedback:
  - Disregard the 80/20 generation rule
    - The rule cannot be satisfied in all possible cases (e.g. children not familiar with the game, children at an advanced stage…)
  - Diversify the scenes offered to the child while using the same theme:
    - All scenes must be different and belong to the same theme
    - All scenes must belong to the the same theme. In addition, two successive scenes must be different
    - All scenes must be different (no constraints on themes)
    - Two successive scenes must be different (no constraints on themes)

Outline

Research context & problem : the *Escape It!* project

Research proposition: an MDE approach for dealing with the generation of adapted scenarios and co-design process for the serious game

Application: the *Escape It!* serious game

Conclusion and future work

Conclusion

- **A co-design process of the Escape it! game:**
  - It allows ASD experts and computer scientists to express and validate the domain elements and dynamic rules for generating adapted learning scenarios
  - It relies on MDE and rapid prototyping
  - The feedback of ASD expert can be considered in the following iteration
- **Model driven engineering:**
  - Automatic generation of adapted learning scenarios
  - Varying the situation proposed to domain experts without significant effort
- **Rapid prototyping:**
  - Based on the automatic integration of learning scenarios in Unity
  - It allows simulating a real exploitation of the game under-development

Future work

- **Rapid generation of the new prototype related to an expressed feedback:**
  - Managing traceability links between the experts recommendations/requirements and the scenarios generation mechanisms
  - Expressing the model transformation responsible for generating scenarios in a more structural and modular manner

Supporting the Adaptive Generation of Learning Game Scenarios with A Model-Driven Engineering Framework

Pierre Laforcade, Youness Laghouaouta
Computer Science Laboratory of Le Mans University, France

EC-TEL 2018, Leeds (UK)

About us

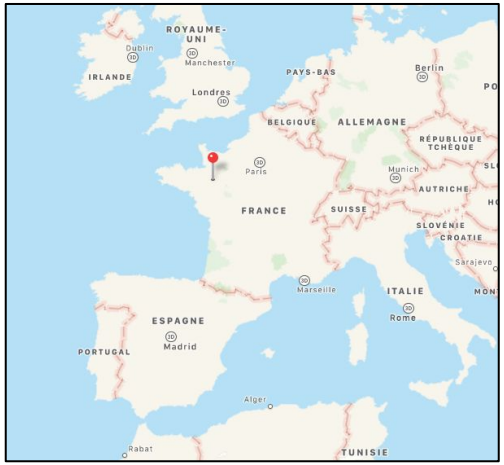
Computer Science Laboratory
Of *Le Mans University*, France

Team about
Engineering of TEL environments

Design, Operationalization, Adaptation of Pedagogical Situations

Observation Modeling & Learning Analytics

Advanced and Collaborative Interactions for Learning



State-of-the-art

- Adaptive serious games usually have specialized, *ad hoc*, approaches, where game components are adjusted to encourage training of a specific skill => **rare generic or reusable design framework**
- Adaptivity research results depend on
  - the **targets**: game mechanics, AI, narratives, content, etc.
  - the **methods**: bayesian networks, ontologies, neuronal networks, rules-based systems, procedural algorithms, …
- Very close and interesting results from CLES project [Sehaba & Hussan, 13]
  - a **generic architecture** (GOALS) for **personalizing** a **serious game scenario** according to **learners' competencies** and interaction traces
    - 3-layers architecture & process to generate 3 successive scenarios: conceptual, pedagogical and serious game scenarios
  - focus on techniques to update learner profile using interaction traces [Hussaan and Sehaba, 2016]
    - Does not tackle the definition of game components, generation rules, ...
  - experts involved during requirements specification and the validation of generated scenarios
    - not during the design stage