



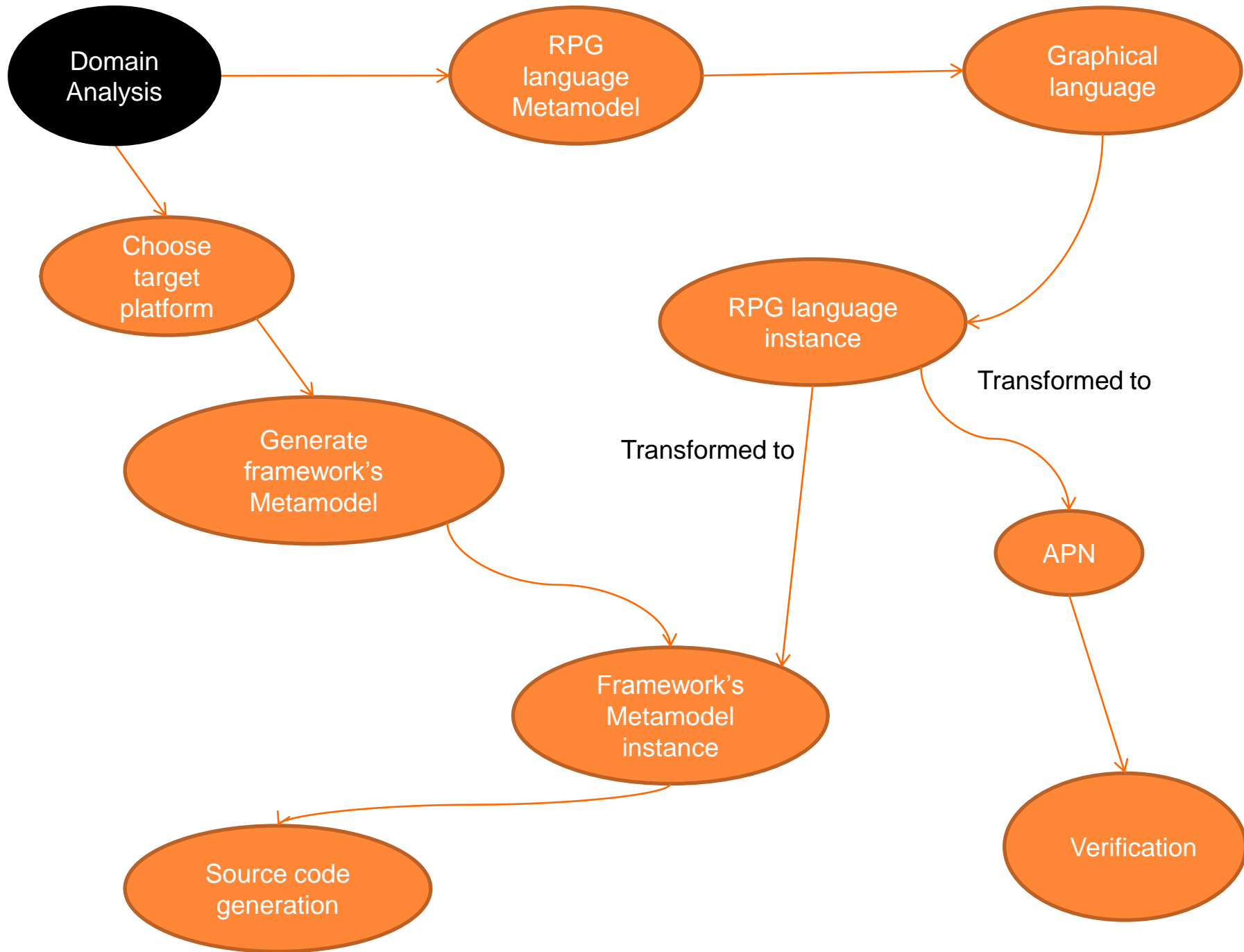
THE RPG DSL: A CASE STUDY OF LANGUAGE ENGINEERING USING MDD FOR GENERATING RPG GAMES FOR MOBILE PHONES

**Eduardo Marques , Valte Balegas,
Bruno Barroca, Ankica Barisic, Vasco Amaral**

THE PROJECT

- Development of a language to create RPG games
 - Domain analysis
 - Language Design
 - Textual/Graphical language editor
 - Generate framework code
 - Game properties verification
- Target users: domain experts





DOMAIN ANALYSIS(1)

- Identify the RPG Domain characteristics
- Restrict the domain
 - 80% approach
- Textual description
- Feature model



DOMAIN ANALYSIS(2)

- World Map
 - Scenes
 - Tile map
- Agents
 - Hero
 - Inventory
 - Hostile
 - Interaction: Fight
 - Friendly
 - Interaction: Dialogue
- Resources
 - Gold, Wood and Metal
- Magics

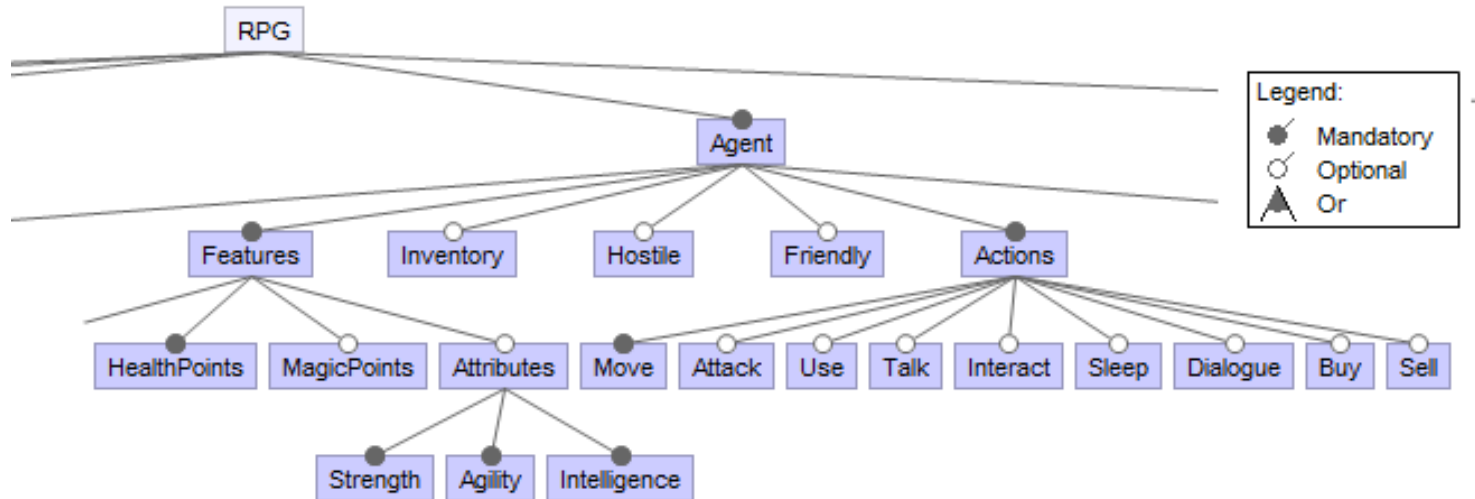


DOMAIN ANALYSIS(3)

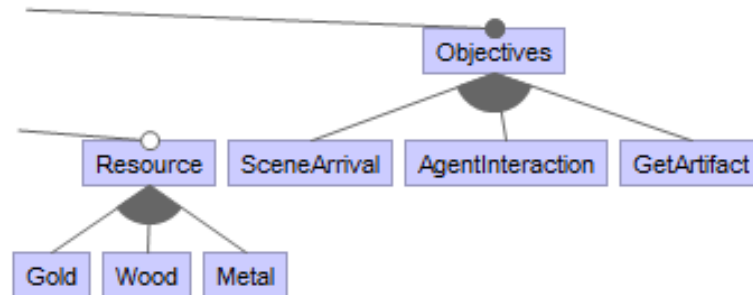
- Agent properties
 - Life points, magic points, strength, agility, intelligence, experience
- Objects
 - Artifacts, equipment, keys, doors, obstacles, traps and switches
- Objectives
 - Reach a specific scene
 - Pick an artifact
 - Interact with an agent
- Time



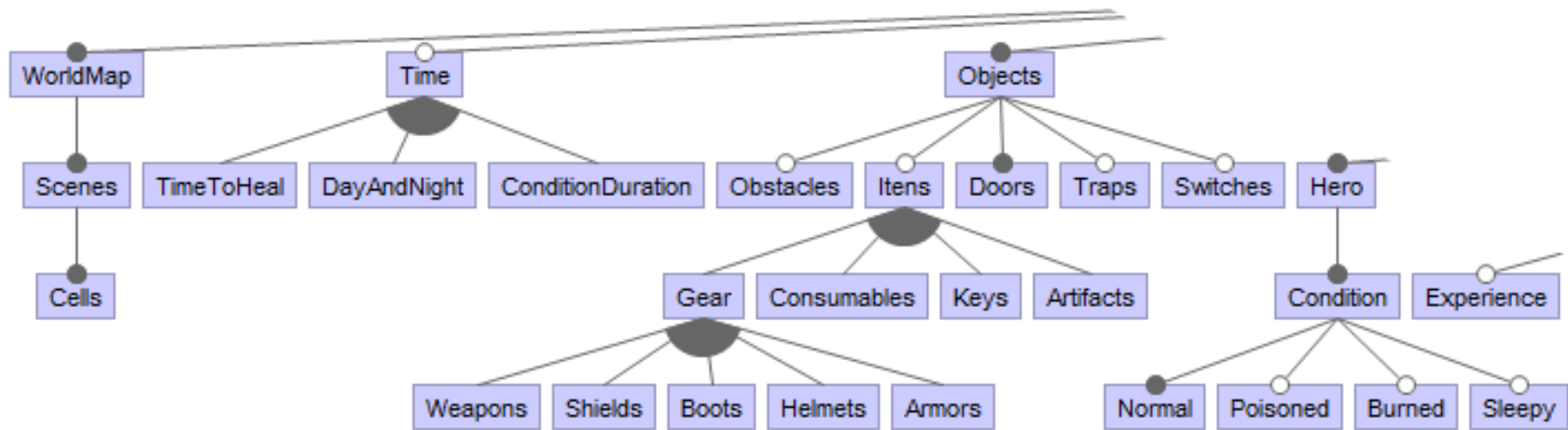
DOMAIN ANALYSIS(4)

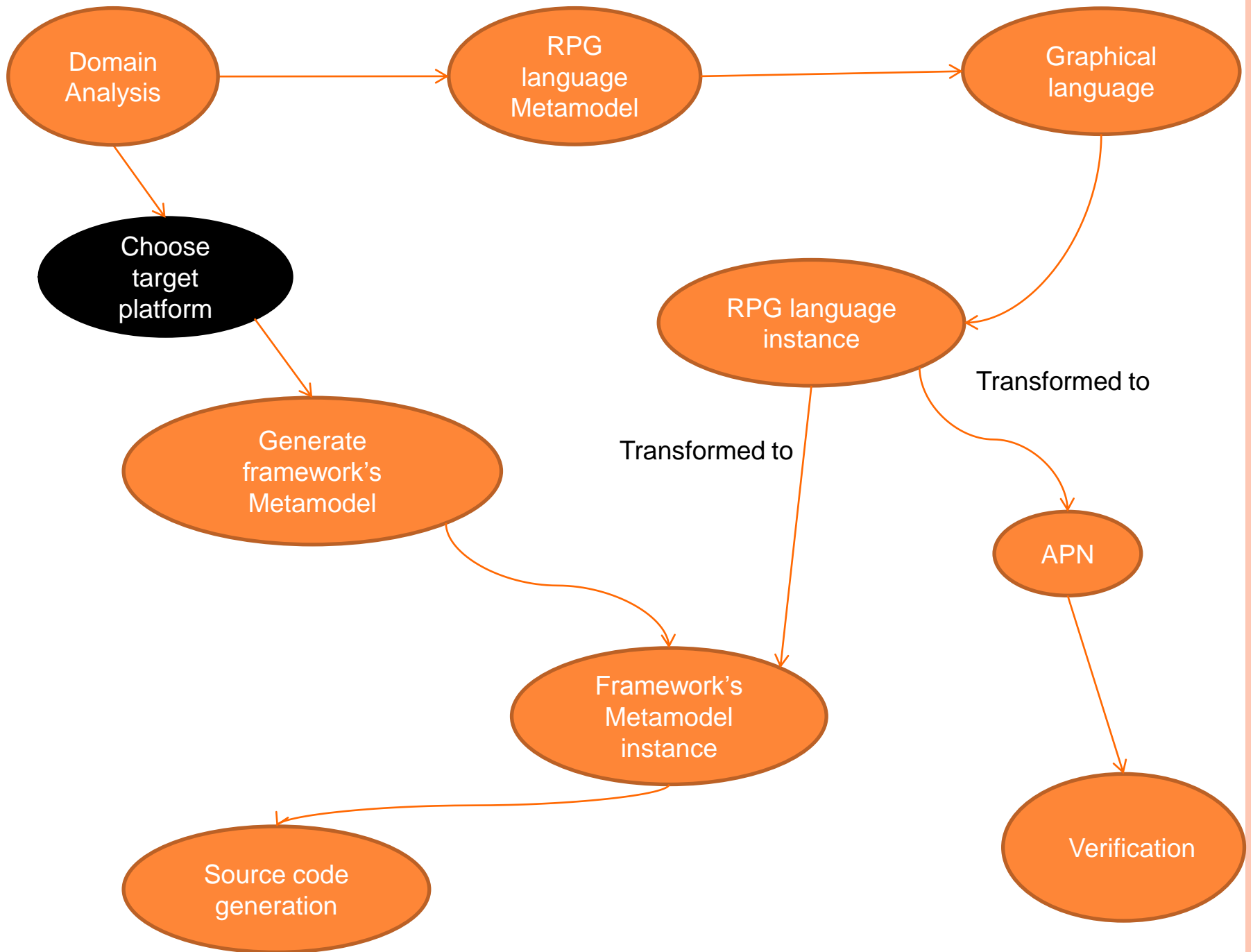


- Inventory \Rightarrow Items
- ConditionDuration \Rightarrow Condition
- Hostile \Rightarrow Attack
- AgentInteraction \Rightarrow Hostile v Friendly
- GetArtifact \Rightarrow Artifacts
- Use \Rightarrow Consumables v Keys v Artifacts
- Dialogue v Talk \Rightarrow Friendly
- Buy v Sell \Rightarrow Items



DOMAIN ANALYSIS(5)





TARGET PLATFORM(1)

- Platform analysis:
 - Advantages and drawbacks
 - Ease of learning/development

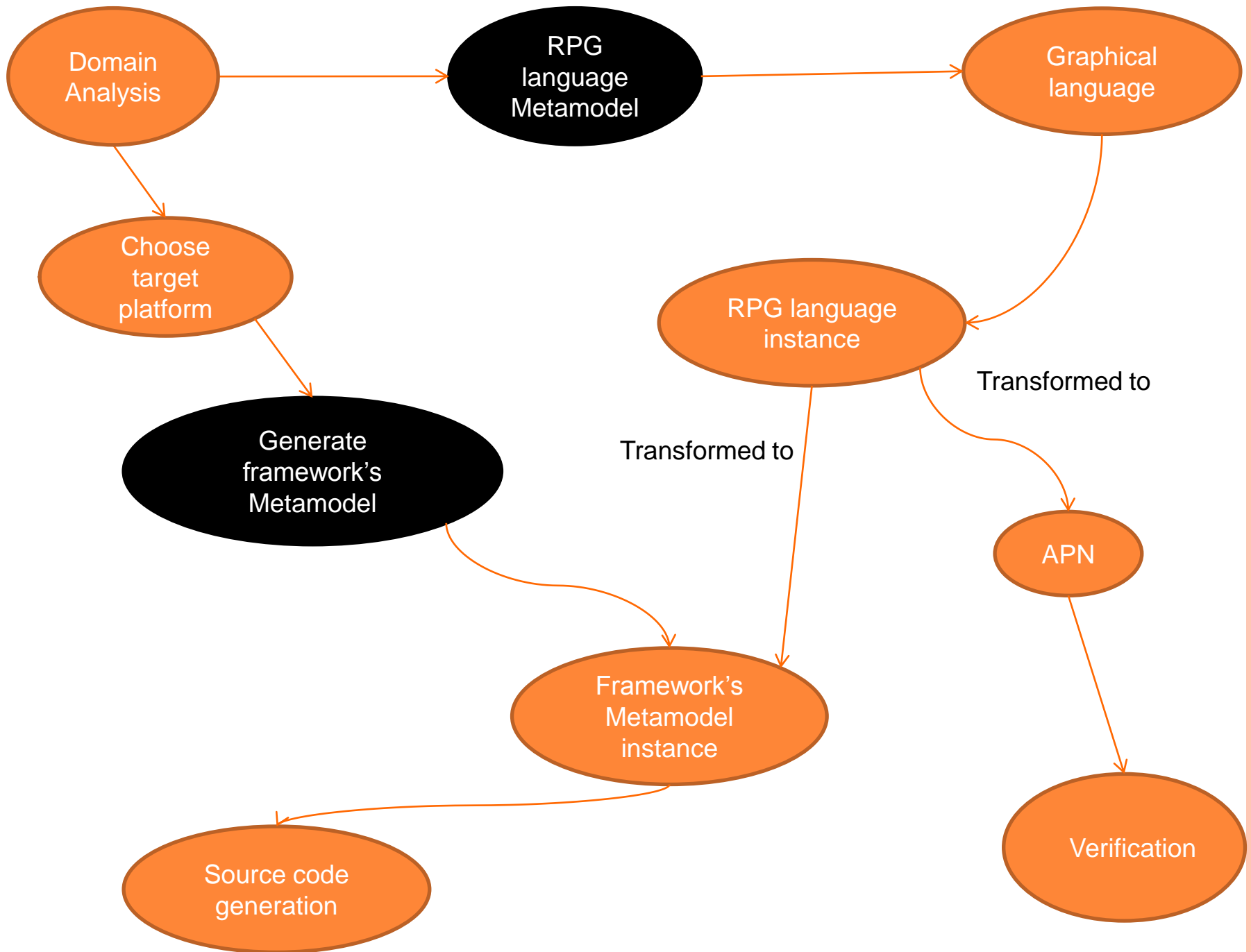
Framework	Characteristics
Slick	Java based language; Uses LWJGL
Sphere	Scripting language; Abstraction level that allows typical RPG features implementation
Corona	Scripting language; Cross-platform compilation for Android and iOS



TARGET PLATFORM(2)

- Corona *Framework*
- Development of an abstraction layer
 - Implements RPG features
 - Data structure library
 - Decision trees, graphs, menu lists
 - Simplify inter-model transformation

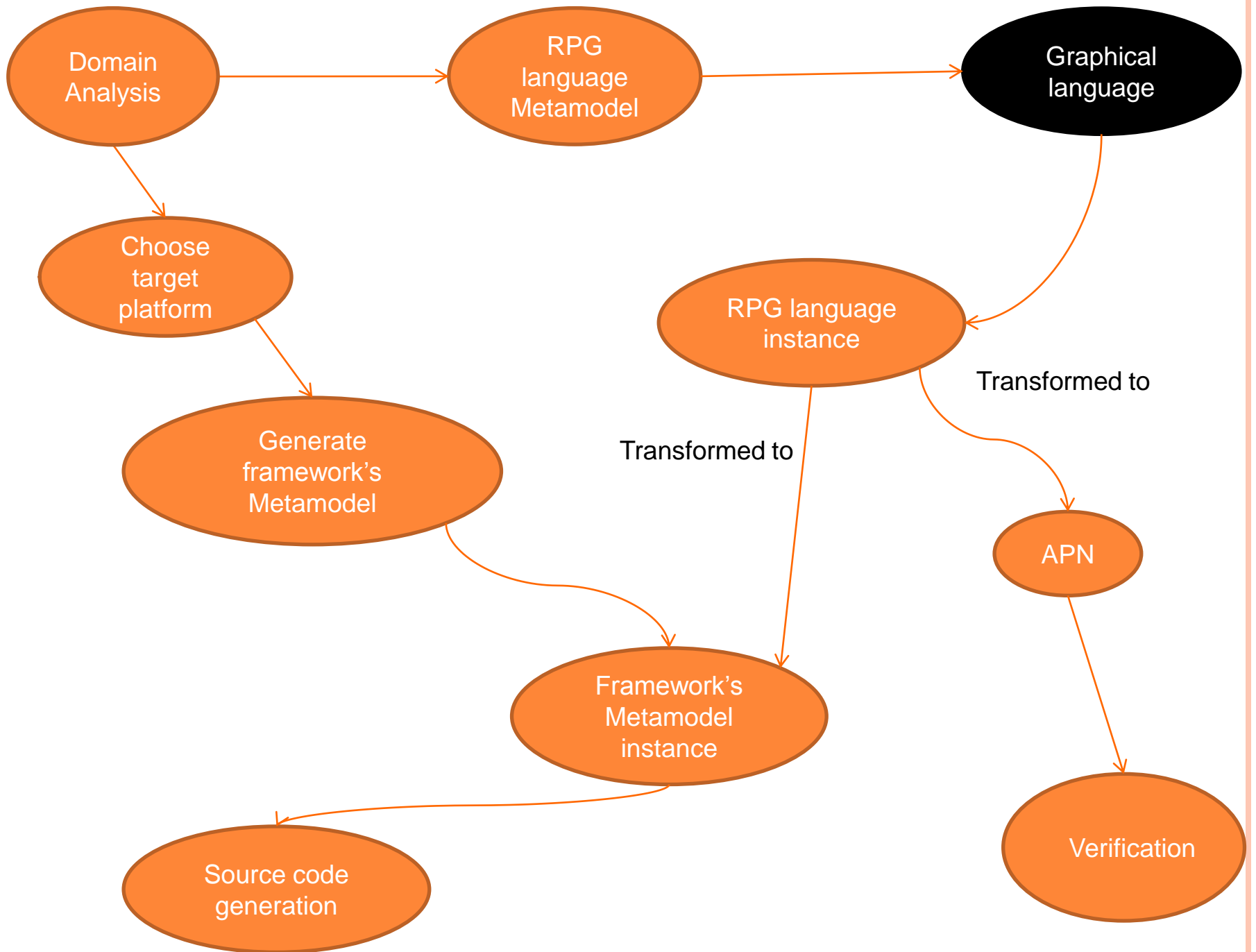




METAMODELS CREATION

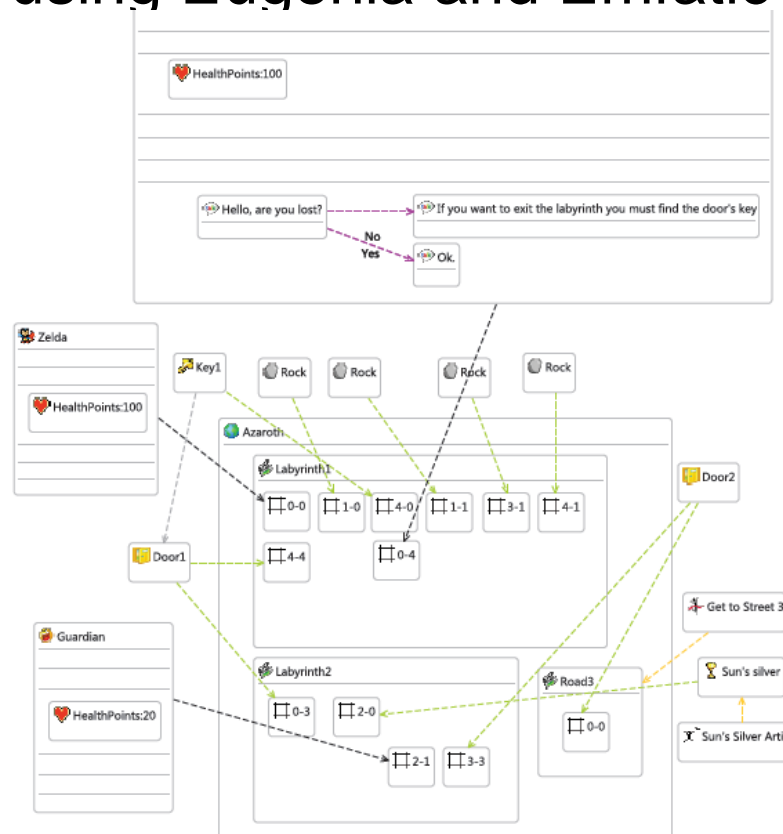
- RPG language metamodel:
 - Based on domain analysis
- Framework metamodel:
 - Based on API layer developed over the framework
- 1-1 relation between models (whenever possible)

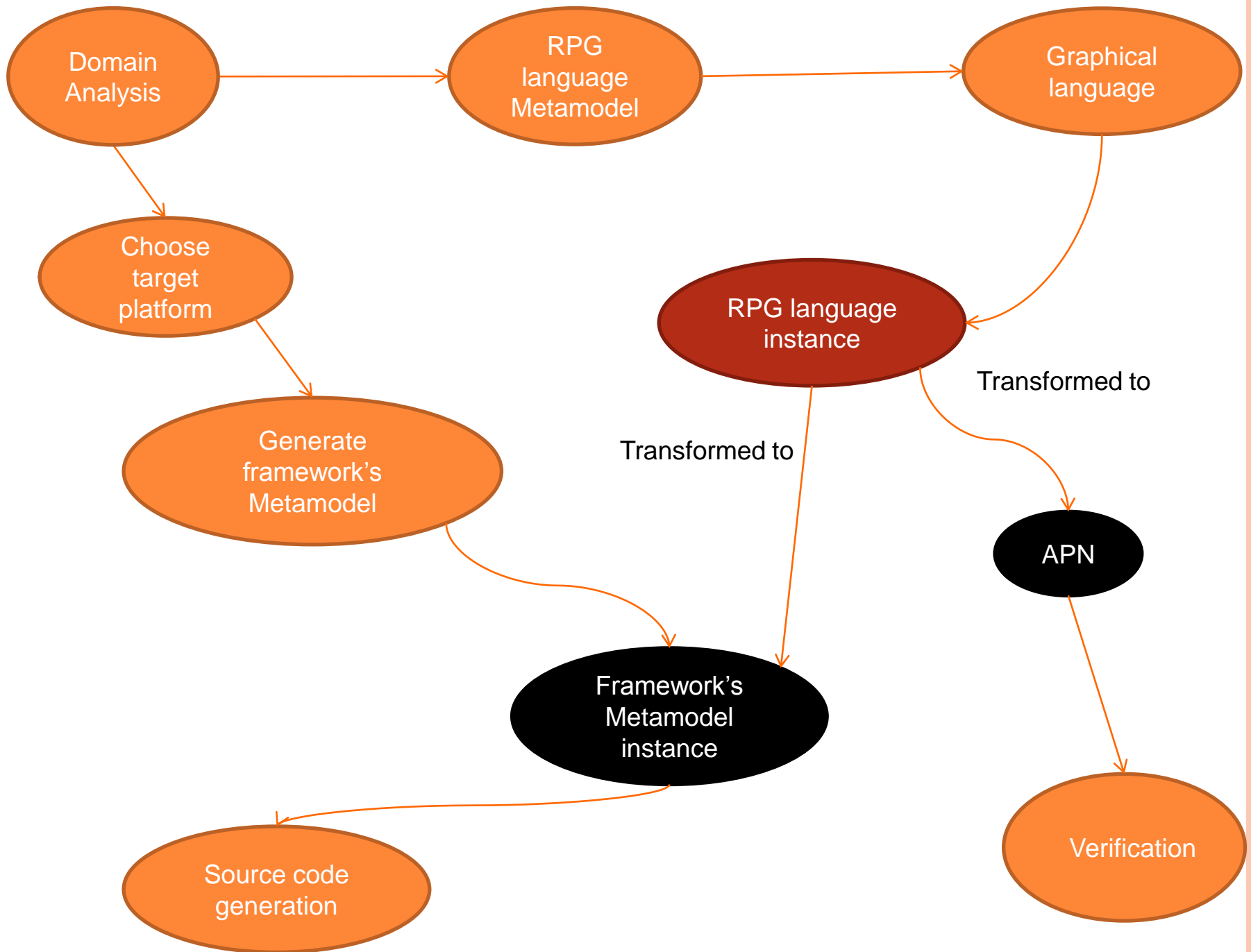




GRAPHICAL LANGUAGE

- Drag and drop approach
 - Ease of use
- Developed using Eugenia and Emfatic

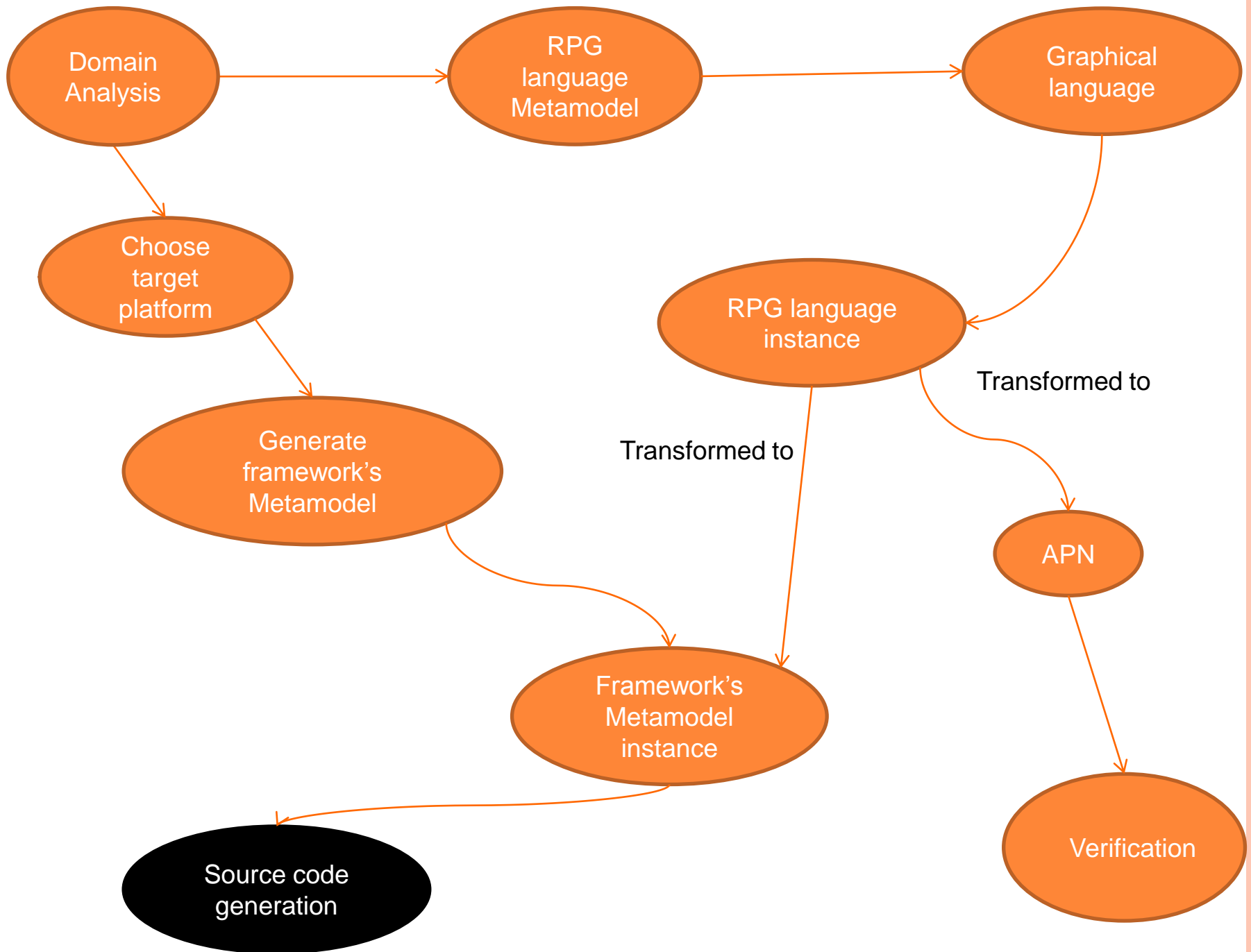




METAMODEL TRANSFORMATIONS

- Generate target platform model instances
- Generate Petri-Nets for verification
- Using ATL
 - Language and *toolkit* for model transformation

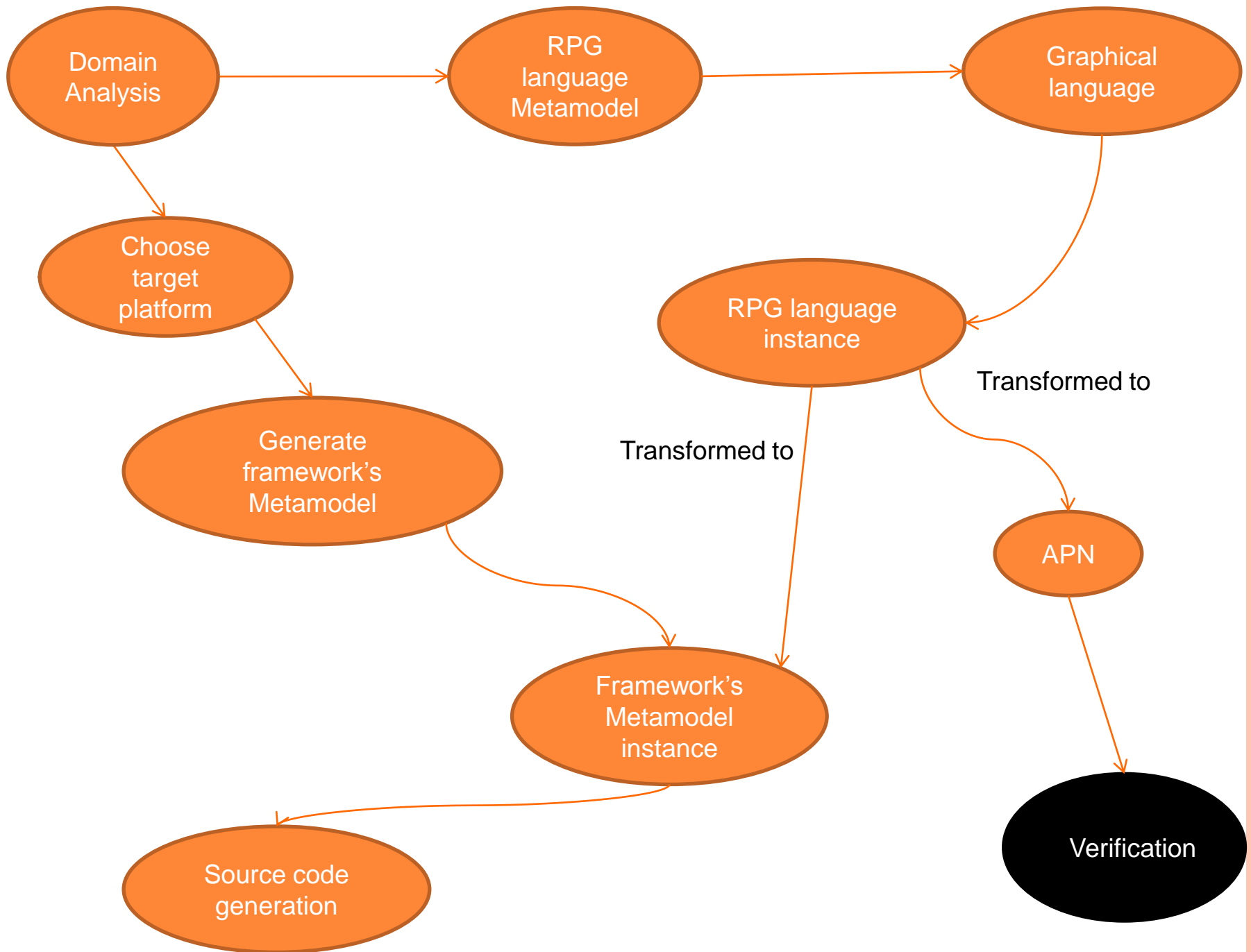




SOURCE CODE GENERATION

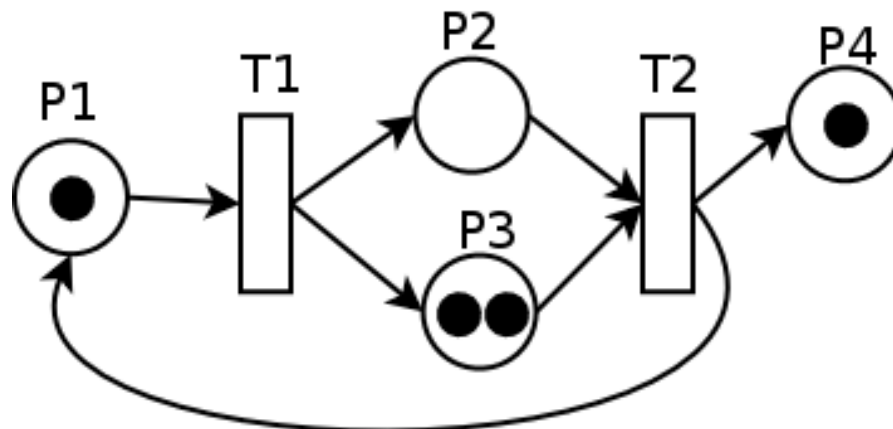
- From RPG language to target platform language
- To generate verification rules
- Using Xpand – Template based language





VERIFICATION(1)

- Create a minimal model of RPG language specific for verification purposes.
- An abstraction to simplify Petri-net generation



VERIFICATION(2)

- Verified properties:
 - User can finish game
 - Player completes the final objective
 - User can finish game with maximum score
 - Player completes all the objectives
- Player can complete an objective if he can reach the scene where it is



QUESTIONS?

