Horizon 2020 Societal challenge 5
Climate action, environment, resource
Efficiency and raw materials

D4.5: Serious Game tool final version

LEAD AUTHOR: Lluís Echeverria

OTHER AUTHORS: Gabriel Anzaldi, Xavier Domingo, Mehdi Khoury, Marcel Ortiz, Nuria Nievas

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement NO 689150 SIM4NEXUS

PROJECT	Sustainable Integrated Management FOR the NEXUS of water-land-food-energy-climate for a resource-efficient Europe (SIM4NEXUS)
PROJECT NUMBER	689150
DELIVERABLE	D4.5 Serious Game tool final version
WP NAME/WP NUMBER	Serious Game development and testing / WP4
TASK	Tasks 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7
VERSION	4.0
DISSEMINATION LEVEL	Public
DATE	23/11/2020
LEAD BENEFICIARY	EURECAT
RESPONSIBLE AUTHOR	Lluís Echeverria (EURECAT)
AUTHORS	Gabriel Anzaldi (EURECAT), Xavier Domingo (EURECAT), Mehdi Khoury (UNEXE), Marcel Ortiz (EURECAT), Aitor Corchero (EURECAT), Nuria Nievas (EURECAT)
REVIEWER	Lydia Vamvakeridou-Lyroudia
COMMENTS	

DOCUMENT HISTORY

VERSION	INITIALS/NAME	DATE	COMMENTS-DESCRIPTION OF ACTIONS
1.0	LLUIS ECHEVERRIA MARCEL ORTIZ	10/12/2019	FIRST VERSION
2.0	MEHDI KHOURY	17/12/2019	GUI CONTRIBUTIONS
2.1	LLUIS ECHEVERRIA	16/01/2020	REVIEW AND CONTENTS REORDERING
2.2	XAVIER DOMINGO	30/01/2020	FINAL REVIEW
2.3	LYDIA VAMVAKERIDOU LYROUDIA	31/01/2020	REVIEW
2.4	LLUIS ECHEVERRIA	31/01/2020	FINAL REVIEW
3.0	MARCEL ORTIZ	15/06/2020	UPDATE
3.1	NURIA NIEVAS	20/06/2020	RL AGENTS
3.2	XAVIER DOMINGO	28/06/2020	REVIEW
3.3	LLUÍS ECHEVERRIA	29/06/2020	FINAL REVIEW

4.0	LLUÍS ECHEVERRIA	23/11/2020	FINAL VERSION SOLVING COMMENTS

Addressing revision comments	
Comment	Response
Also include the direct link to the Global Case Study demo tool in Executive Summary and Short Summary of results	An explanation of the Global CS demo tool has been introduced in the Executive Summary. The link has also been included.
2.5.7 Game Page - update needed "Currently, only "Easy" game mode is active and able to test, where the User can play the Game without any restriction. The addition of new modes will be taken into account during the last part of the project."	Section 2.5.7; Pg24. Update GamePage has been updated with all available game modes. A specific table, with game modes by CaseStudy, has been introduced.
Pg 38 – link to Global CS system diagram https://seriousgame.sim4nexus.eu/global/systemdiagram.html - is this operational – the diagram on the website looks different from Fig 26 in D4.5? It should also be explained on the website how this should be used and interpreted, as already done for the Global CS physical flow diagram	Global CS tool links have been validated and a further explanation of the tool functionalities has been included, both in the deliverable and in the UI (guided tour). In addition, two buttons have been added in the UI to enable the navigation between the S4N Main Page and the Global tool views

Table of Contents

Figures

Tables

Executive summary

The development of the SIM4NEXUS Serious Game (SG) tool has reached an important achievement, and it is considered to be finished, thanks to the implementation and integration of its four main components: i) the Graphical User Interface (GUI), ii) the Knowledge Elicitation Engine (KEE), iii) the SIM4NEXUS Database and iv) the System Dynamic Models Engine (SDM Engine). Each one of these elements has an essential role in the Game system and its crucial for the correct behaviour of the tool.

Task 4.3 (Setting-up the project database and metadata ontology) has focused the definition and implementation of a specific database structure taking into account T4.1 (Learning goals definition), detailed in the deliverable D4.1 and its updated versions D4.8 and D4.9, T4.2 (Game logic definition), use cases from WP5, and persistence necessities of WP1, WP2 and WP3. The database schema is divided into two main components: i) the Semantic Repository (introduced in D4.4), which stores all use cases information in a standardised and interoperable way, and ii) a relational database where another type of data is persisted to be used or analysed in the future.

The KEE has been implemented under T4.4 (Development of Knowledge Elicitation Engine). It represents the hearth of the SIM4NEXUS Serious Game tool and works as a link between all the other components. Among its many tasks, it has to deal with all the requests sent by the GUI, manage the data flows between the database and the other components, execute the SDM Engine to simulate each Game turn and give advice to the players to guide them during the Game.

At the top of the SG structure, the GUI has an important role, it is the visual layer of the tool. It has been built under T4.5 (Development of the visualization and interaction tool) to be an online interactive User Interface, accessible from anywhere through the use of a web browser. Over it, the player will be able to: i) interact with the proposed Case Studies, ii) understand the complex connections between water, food, climate, energy and land, and iii) learn about the applicability of specific policies and its direct and long term effects.

Last but not least, the SDM Engine, which is in charge of integrating the SDMs (provided by WP3) to the KEE and manages their execution to simulate the different Game turns, has been developed under T4.4 (Development of Knowledge Elicitation Engine).

All of these components have been fully and successfully interconnected making possible the first complete version of the complete SIM4NEXUS Serious Game tool which, from now on, will be publicly accessible through the following URL: https://seriousgame.sim4nexus.eu/.

Five Case Studies, Greece, Azerbaijan, Latvia, the Netherlands and the southwest of the UK, are deployed and ready to be played through the SG. Finally, the Global Case Study has developed a demo tool which can be accessed through the S4N SG platform (https://seriousgame.sim4nexus.eu/global/). This demo tool offers an interactive guided tour to the exploration of the interlinkages between sectors in the water-land-energy-food-climate Nexus, at the global level, and how future scenarios, with or without policy assumptions, may result in different situations.

Changes with respect to the DoA

The SIM4NEXUS Serious Game tool has been successfully implemented and is ready to be accessed by the players. The present report describes in a detailed way the development of this tool and the efforts devoted during the project.

No changes.

Dissemination and uptake

This report is public, so it is accessible for everyone. However, the specific targeted audience of this report are the beneficiaries of the project.

Short Summary of results

The SIM4NEXUS Serious Game tool has been successfully implemented and is ready to be accessed by the players. Five Case Studies, Greece, Azerbaijan, Latvia, the Netherlands and the southwest of the UK , are deployed and ready to be played through the S4N SG GUI. Finally, the Global Case Study has developed a demo tool which can be accessed through the S4N SG platform.

Evidence of accomplishment

The latest version of the Serious Game is available and free to play at this URL: https://seriousgame.sim4nexus.eu/

Glossary / Acronyms

TERM	EXPLANATION / MEANING
API	Application Programming Interface
CS	Case Study
DSS	Decision Support System
GUI	Graphical User Interface
IE	Inference Engine
JWT	JSON Web Token
KEE	Knowledge Elicitation Engine
ML	Machine Learning
OGC	Open Geospatial Consortium
RFC	Request For Comments
RL	Reinforcement Learning
S4N	Sim4Nexus
SDM	System Dynamic Model
SOA	Service Oriented Architecture
SR	Serious Game
SR	Semantic Repository
UI	User Interface
UK	United Kingdom
URL	Uniform Resource Locator
WP	Work Package
XML	Extensible Mark-Up Language

1. Introduction

This document describes the SIM4NEXUS Serious Game (SG) tool. The SG is composed of four main elements (Figure 1): i) the Graphic User Interface (GUI), ii) the Knowledge Elicitation Engine (KEE), iii) the S4N Database and iv) the System Dynamic Models Engine (SDM Engine).

The GUI is the visual part of the tool and aims to create a realistic virtual environment where the players can interact with the proposed Case Studies and learn about the complex connections between the nexus elements and the impact of applying different policies.

The KEE is the core of the SG. It acts as a central connector between the other SG components and implements all the Game logic based on the outputs from other WPs such as the SDMs (WP3) or the policy definitions (WP2).

All SIM4NEXUS data, either generated during the project, such as the Learning Goals (T4.1) or the policies (WP2), or by the players during the execution of the Game, are stored in the SIM4NEXUS database. It is divided into two components: i) the Semantic Repository (SR) (D4.4) and a relational database. Depending on the source, type and utility of the data, it will be stored in one of these two databases.

Finally, the SDM Engine, a specific key interface which has two main functionalities. First, it is in charge of integrating the SDMs (provided by WP3) to the KEE and, second, it manages their execution to simulate the different Game turns.

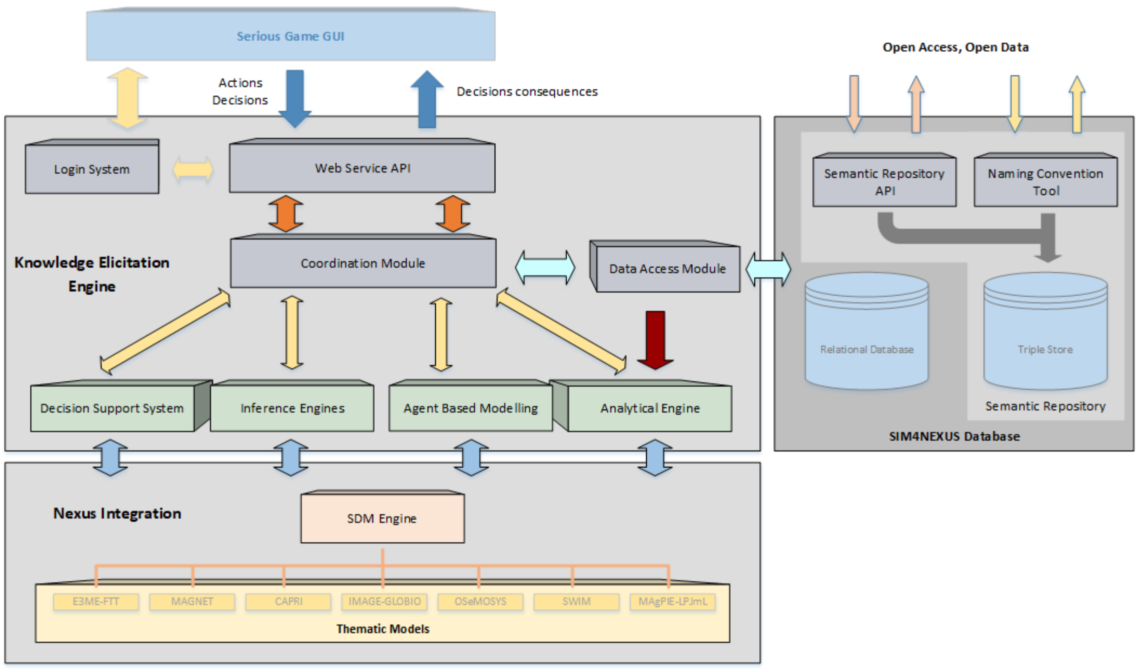


Figure 1. Serious Game tool Architecture.

At this stage, all the components have been successfully implemented and interconnected and the tool (SG) is ready to be played. Finally, five Case Studies have been fully integrated, Greece, Azerbaijan, Latvia, the Netherlands and the southwest of the UK. In parallel, the Global Case Study has developed a demo tool which can be accessed through the S4N SG platform.

As proof, the latest version of the Serious Game GUI and the underlying connected KEE, S4N database and SDM Engine are available and free to play at this URL: https://seriousgame.sim4nexus.eu/.

2. Graphical User Interface (GUI)

2.1 Introduction

The User Interface of the Serious Game is built (under task "T4.5 Development of the visualization and interaction tool") to be an online interactive visualization accessible from anywhere through the use of a web browser. Presently, it is optimised for the use of the widely popular Chrome web browser so as to guarantee both ease of access and good performance when displaying in an effective manner a complex model using a wide range of dynamically changing variables. To this end, the User Interface involves a number of technologies (Table 1. Sim4Nexus Serious Game UI technologies) on which its proper functioning depends.

Table 1. Sim4Nexus Serious Game UI technologies

jQuery	JSON Web Tokens	d3	turfJS
Dynamic control all across the HTML files (events, editions, collections, ...).	Compact and self-contained way for securely transmitting information.	Manipulate documents based on data. Brings control over the HTML thanks to the Game data.	GeoJSON management integrated with JavaScript to bring powerful methods.
TopoJSON	threeJS	mathJS	tweekJS
Extension over geoJSON which brings topology control over it.	Manages the creation and display of animated 3D graphics over web browsers	Extension over JavaScript which brings support for symbolic, complex and matrices computation.	Provides tweaking and animating control over HTML5 and JavaScript properties.

Currently, the available language is English, but the UI is able to introduce additional ones. In this line, native partners will be asked to contribute to the translation process.

On this basis, the current version of the Serious Game User interface involves the following main views/pages (and its sub-pages), which conform the Website Client and define the entire Application Flux that can be followed by a User:

- Login Page.
- Main Page.
- Profile Page.
- Game Page.

The UI has been implemented in such a way the addition of a new Case Study, and all its related data, don't explicitly need extra implementations and only a few processes common for all cases are necessary, such as the creation of a new map. Specifically, five Case Studies have been fully integrated into the Game Page, Greece, Azerbaijan, Latvia, the Netherlands and the southwest of the UK.

At this point, the different views will be introduced by a brief explanation followed by the view schema. That schema defines all the routings and interactions which can be done in the Client side of the view. The view definition has the following sections contained:

- Routes: Routing options which can be taken by the User at any time. The only restriction can be applied will correspond to the User roles and their status (Authenticated User or Guest User).
- Form interactions: Every single interaction with a formulary will be pointed out to know where is located and which data expects to get.
- Definition: Summary of the event which can be done in the view. It also points out where the routing options and form interactions are located in terms of User interactions.

2.2 Authentication system flux specification

The authentication system involves all the Serious Game modules related to User profile management, security, and customization. The Game takes control of the Users sessions by classifying them into two types of User:

- Guest User: By definition, this type of User corresponds to all the Game session where the User did not want to Sign in. The process to play as a Guest User only require to select the "Play as Guest" option in the *LoginPage* section. From that point in advance, the user will be tracked as a Guest for the System thanks to a Guest Identification value.
- Authenticated User: Authenticated Users play the game involving a previous identification of themselves in the *LoginPage* section. That identification requires to provide the email linked to their Sim4Nexus Serious Game account and the corresponding password of that account. Once the form is filled correctly they'll enter the game system as an Authenticated User. The browser private cookies in hand with JSON Web Tokens (JWT) will take care of the security of the User's account and speed-up any request to the Game Logic Manager. Thanks to that, the system avoids the continuous identification requirement.

2.3 Game system flux specification

Regarding the Game System, the User roles do not make any difference in the Game experience. Both types of Users are allowed to see and interact with the same options and modules.

The only difference will be allocated in the Game Logic Manager which corresponds to the server (KEE) part and does not affect the User experience at any moment. The Login Manager will always differentiate the actions performed by an Authenticated User or by a Guest User. That point is vital to keep track of the User's behavioural and logical evolution in-game.

A Guest User is linked to a Session ID (Session identification number) which will tell the Game Logic Manager who is performing the actions at any moment. This Session ID will expire once the Guest User leaves the Serious Game application. As a consequence, we will lose any type of correlation between that Guest User and their future Game session as a new Session ID will be attached to him the next time he/she decides to lay the game.

At this point is when the importance of the Authenticated User arrives. Once we have an account linked to a User we can keep the evolution of that User over time. That tracking over time will lead to obtaining a Behavioral Profile for that Authenticated User which will tell us how the users evolve and experience.

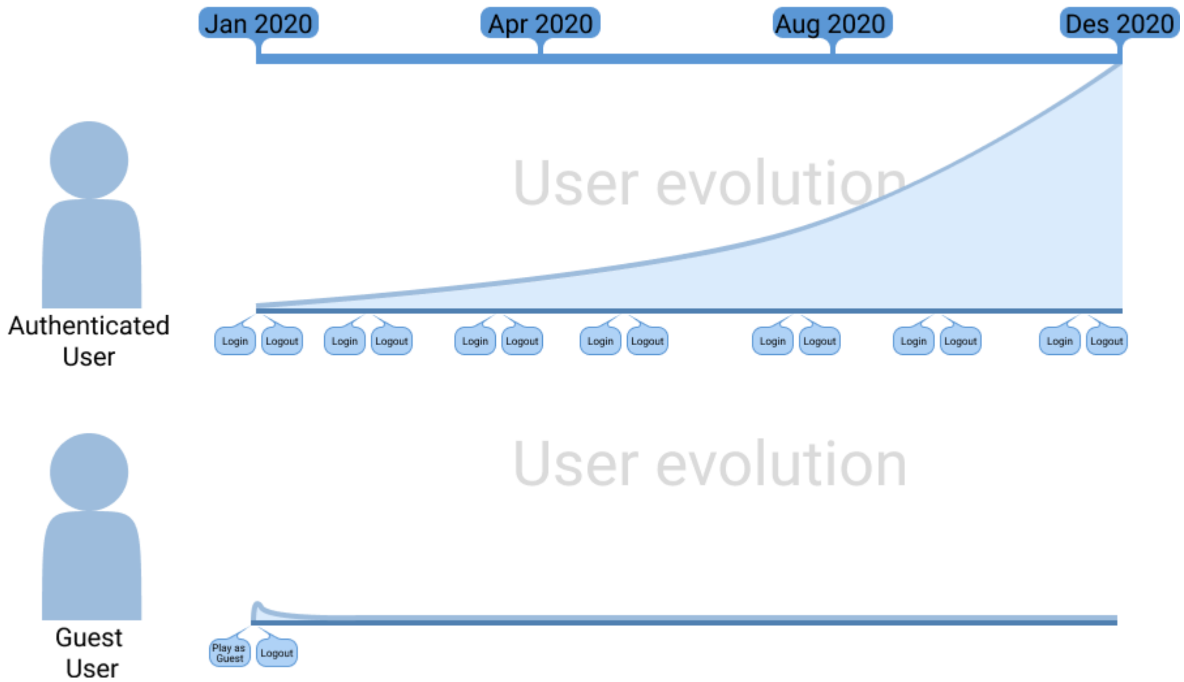


Figure 2: Users progress comparison regarding the Authentication procedure.

2.4 Application flux canvas

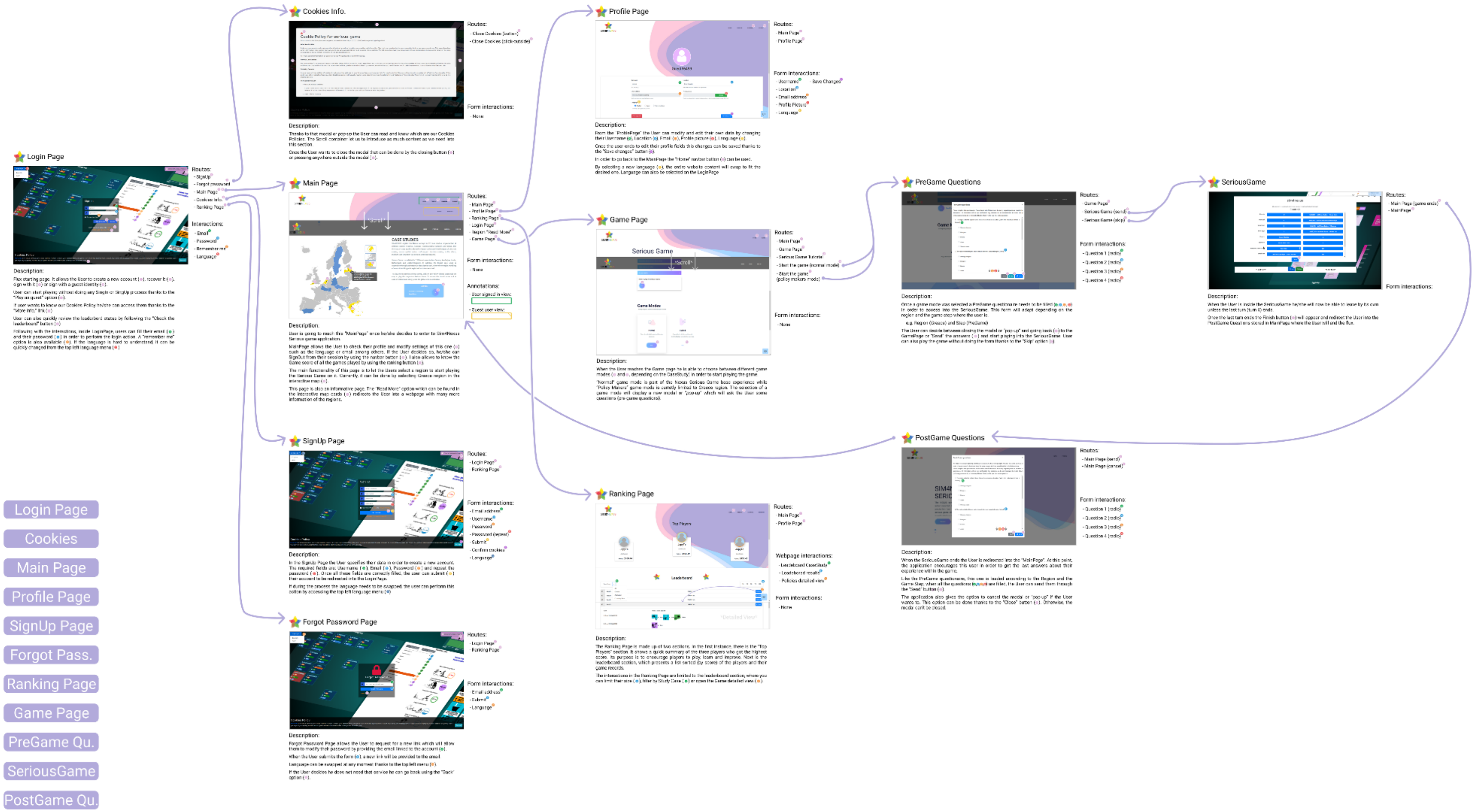


Figure 3: Full flux interaction among Sim4Nexus Serious Game website application views .

2.5 User Interface analysis

2.5.1 Login Page

Definition

Starting page of the entire Application flux. It allows the User to create a new account (1.), recover it (2.), sign with it (3.) or sign with a guest identity (4.).

User can start playing without doing any *SingIn* or *SingUp* process thanks to the “Play as guest” option (4.).

If the User wants to know our Cookies Policy, he/she can access them thanks to the “More info.” link (5.).

Schema

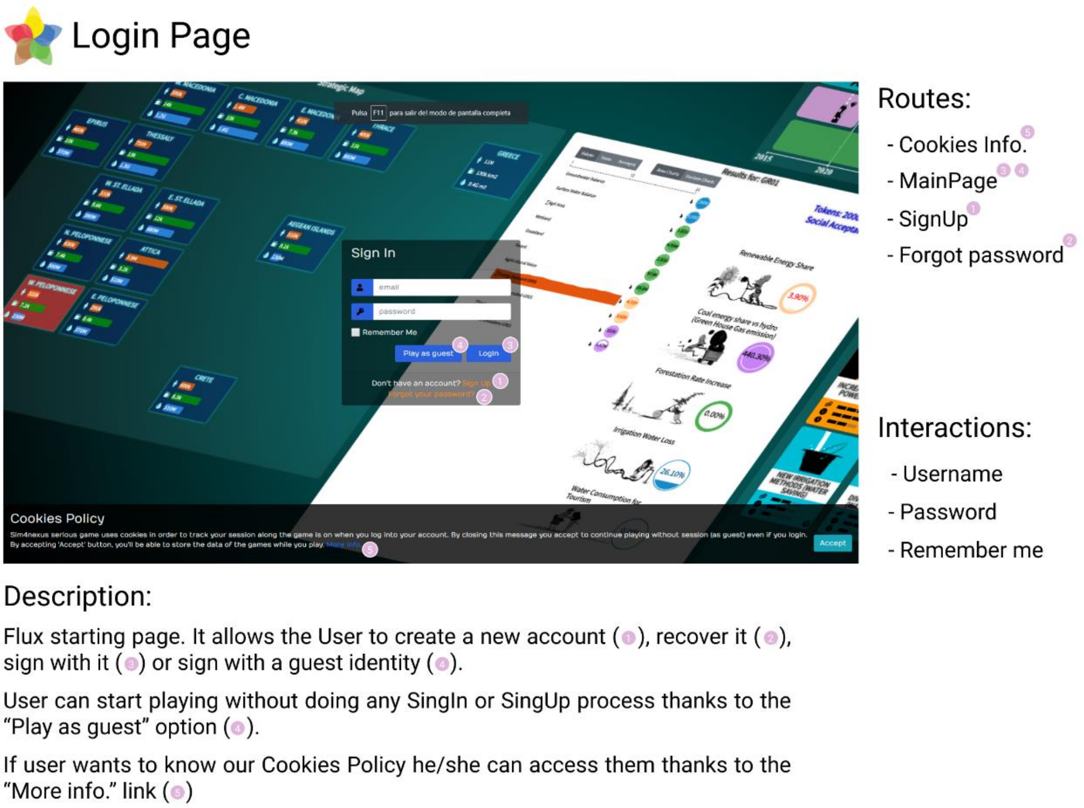


Figure 4: LoginPage graphical representation and interaction.

2.5.2 Cookies Policy

Definition

Thanks to that modal or pop-up the User can read and know which are our Cookies Policies. The Scroll container let us to introduce as much content as we need into this section.

Once the User wants to close the modal that can be done by the closing button (1.) or pressing anywhere outside the modal (2.).

Schema

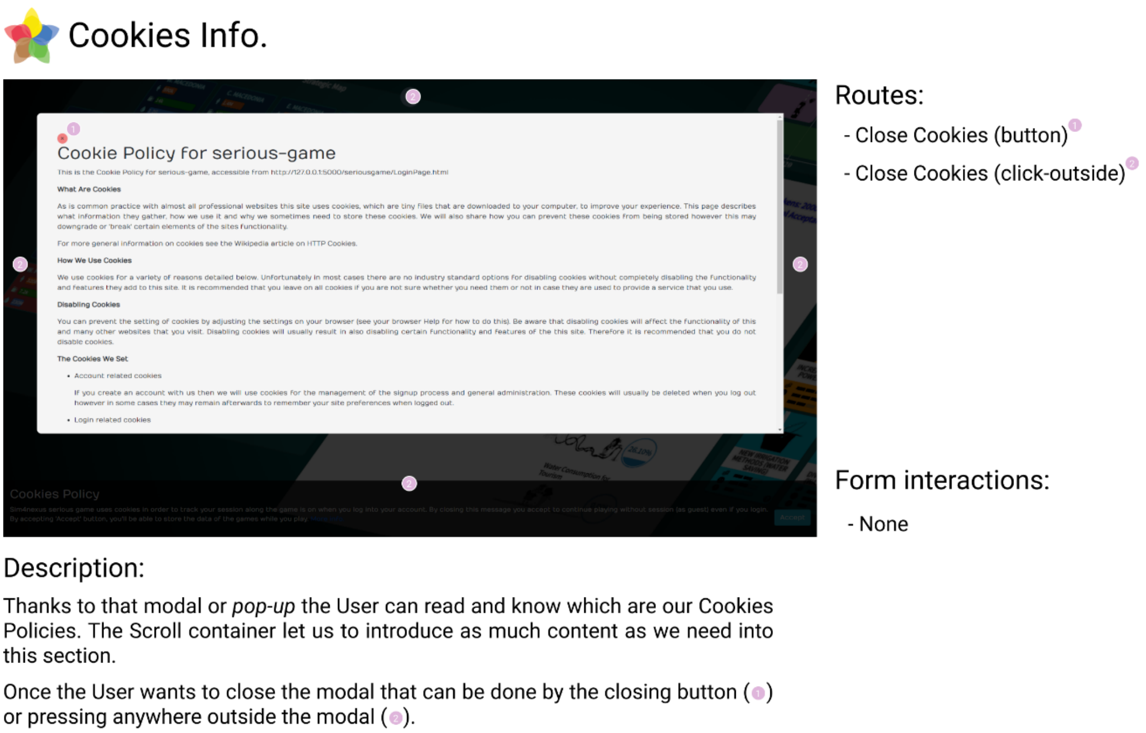


Figure 5: Cookies Information page graphical representation and interaction.

2.5.3 Main Page

Definition

User is going to reach this *MainPage* once he/she decides to enter to Sim4Nexus Serious game application.

MainPage allows the User to check their profile and modify settings of this one (2.) such as the language or email among others. If the User decides so, he/she can sign out from their session by using the navbar button (3.).

The main functionality of this page is to let the Users select a CS to start playing the Serious Game on it (4.).

This page is also an informative page. The "Read More" option which can be found in the interactive map cards (5.) redirects the User into a webpage with much more information of the CS.

Schema

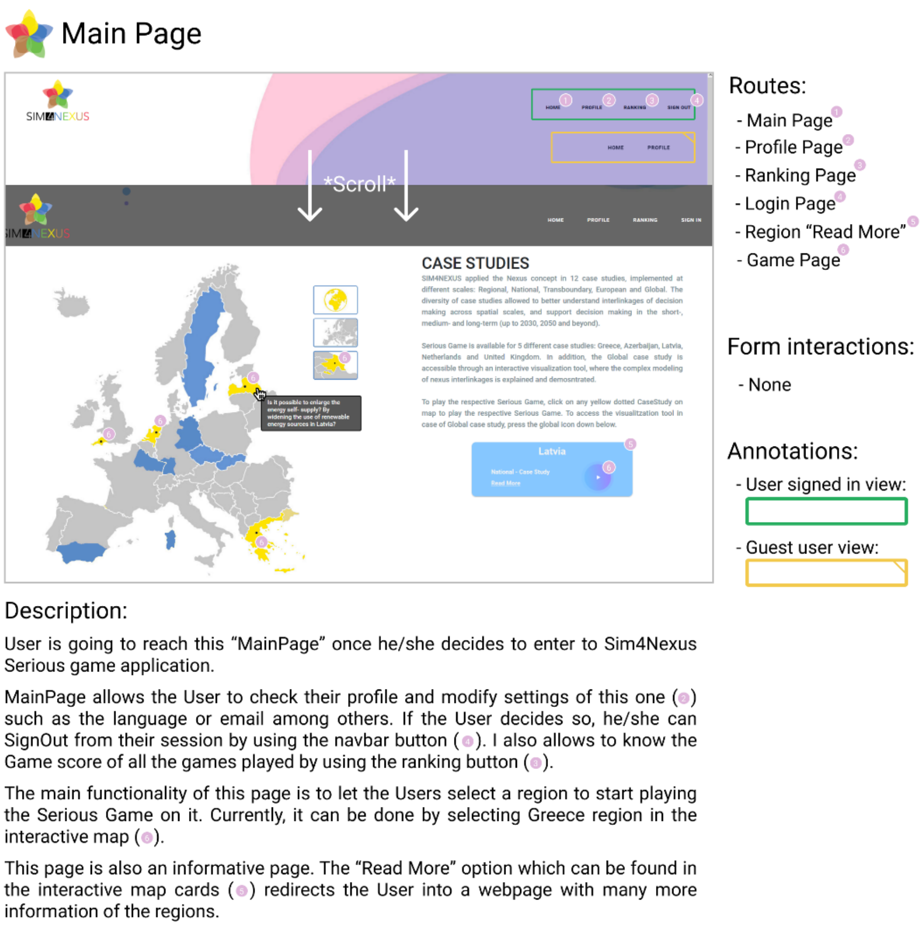


Figure 6: MainPage graphical representation and interaction.

2.5.4 Profile Page

Definition

Description:

From the *ProfilePage* the User can modify and edit their own data. Currently, it is possible to specify the Username (A.), Location (B.), Email (C.), Profile picture (D.), Language (E.).

Once the user ends to edit their profile fields this changes can be saved thanks to the "Save changes" button (F.).

In order to go back to the *MainPage* the "Home" navbar button (1.) can be used.

By selecting a new language (E.), the entire website content will swap to fit the desired one. Language can also be selected on the LoginPage. Once the language is selected, the change will persist among the different pages enabling the translation in real time.

The S4N SG has been developed in English language and translated to Azeri. The I18N system implements a dynamic mechanism to translate the SG in other languages (such as Latvian or Greek) will be added in the near future.

Schema

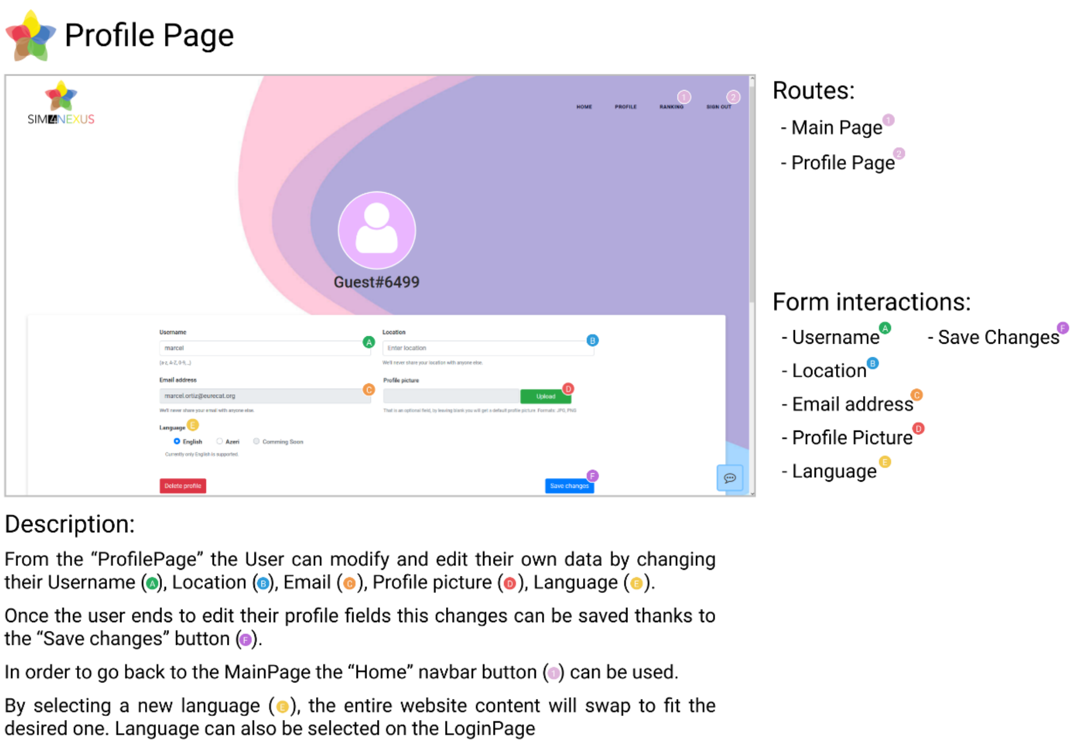


Figure 7: ProfilePage graphical representation and interaction.

2.5.5 Signup page

Definition

In the *SignUp* Page the User specifies their data in order to create a new account. The required fields are: Username (A.), Email (B.), Password (C.) and repeat the password (D.). Once all these fields are correctly filled, the user can submit (E. & 1.) their account to be redirected to the *LoginPage*.

Terms of service and privacy policy must be accepted to successfully create an account.

Schema

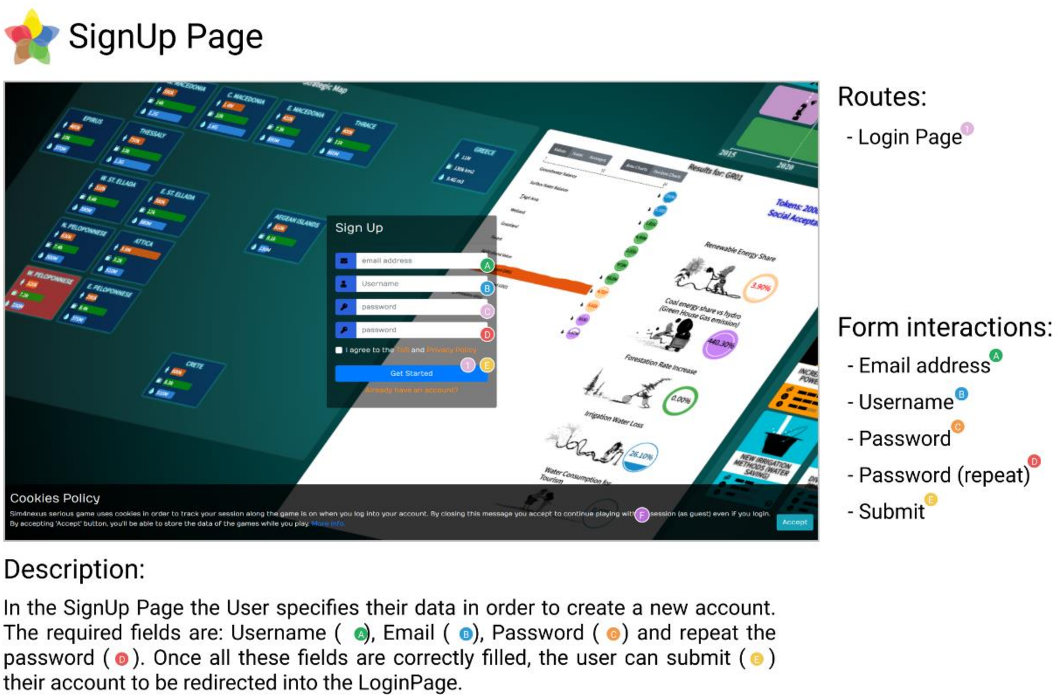


Figure 8: Signup page graphical representation and interaction.

2.5.6 Password recovery page

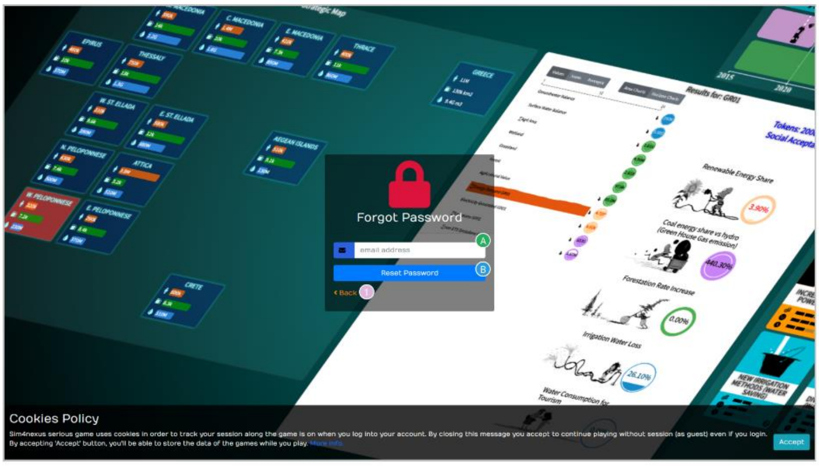
Definition

ForgotPassword Page allows the User to request for a new link which will allow him to modify their password by providing the email linked to the account (A.).

When the User submits the form (B.), a new link will be provided to the email. If the User decides he does not need that service he can go back using the "Back" option (1.).

Schema

Forgot Password Page



Routes:

- Login Page[1]

Form interactions:

- Email address[A]
- Submit[B]

Description:

Forgot Password Page allows the User to request for a new link which will allow them to modify their password by providing the email linked to the account (A).

When the User submits the form (B), a new link will be provided to the email.

If the User decides he does not need that service he can go back using the "Back" option (1).

Figure 9: Forgot Password page graphical representation and interaction.

2.5.7 Game Page

Definition

When the User reaches the Game Page, he can choose between different game modes (3.) to start playing the game.

The selection of a game mode may display (depending on the CS) a new modal or "pop-up" which will ask the User some questions (pre-game questions).

Game modes are divided into three different options depending on the CS selected to play with:

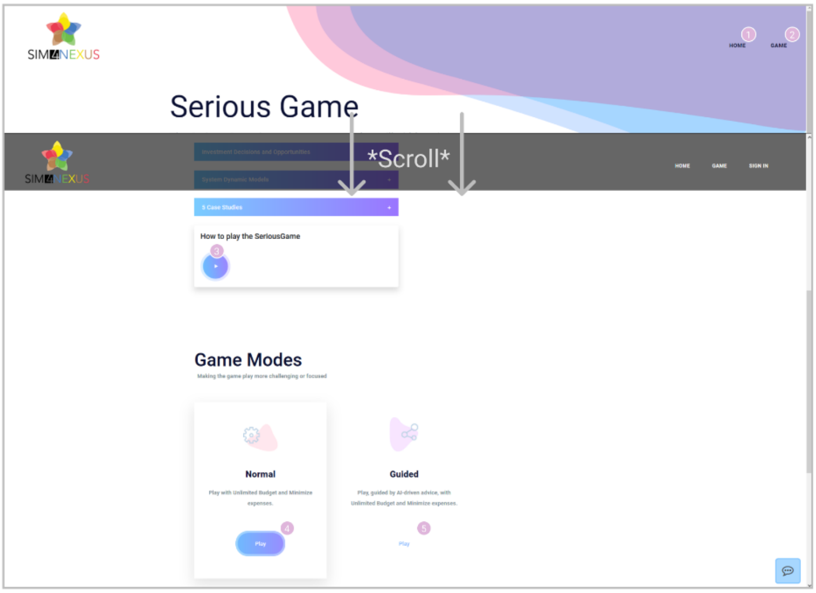
- Normal: Play the base experience of the Nexus Serious Game. It allows performing all the possible interactions over the Nexus thanks to the PolicyCards and an unlimited budget which allows experimenting with each possible outcome. This game mode is meant to be played to minimize the expenses since these take an important part in the final score.
- Policy Makers: Policy Makers is a sub-module born from the Nexus Serious Game Normal game mode. It is intended to be a more realistic approach to the game. In this scenario, resources are limited, forcing the User to improve its strategy to reach the most efficient outcome.
- Guided: The S4N DSS (section 3.2.7) will provide advice to the User in order to guide him to achieve the best 2050 scenario based on the current game status. To this end, the S4N DSS is based on the ABM (section 3.2.5) and IE (section 3.2.6), both modules introduce AI-driven methodologies to enable smart advice.

Table 2: SeriousGames gaming modes per Case Study

Serious Game / Game mode	Azerbaijan	Greece	Latvia	Netherlands	United Kingdom - Southwest
Normal	Yes	Yes	Yes	Yes	Yes
Policy Makers	No	Yes	No	No	No
Guided	Yes	No	No	No	No

The ABM engine is a critical area of research for Eurecat and new scenarios will be supported soon (e.g. the NL CS)

Game Page



Routes:

- Main Page[1]
- Game Page[2]
- Serious Game Tutorial[3]
- Start the game (normal mode)[4]
- Start the game[5] (policy makers mode)

Form interactions:

- None

Description:

When the User reaches the Game page he is able to choose between different game modes (4 and 5, depending on the CaseStudy) in order to start playing the game.

"Normal" game mode is part of the Nexus Serious Game base experience while "Policy Makers" game mode is curretly limited to Greece region. The selection of a game mode will display a new modal or "*pop-up*" which will ask the User some questions (pre-game questions).

Figure 10: GamePage graphical representation and interaction.

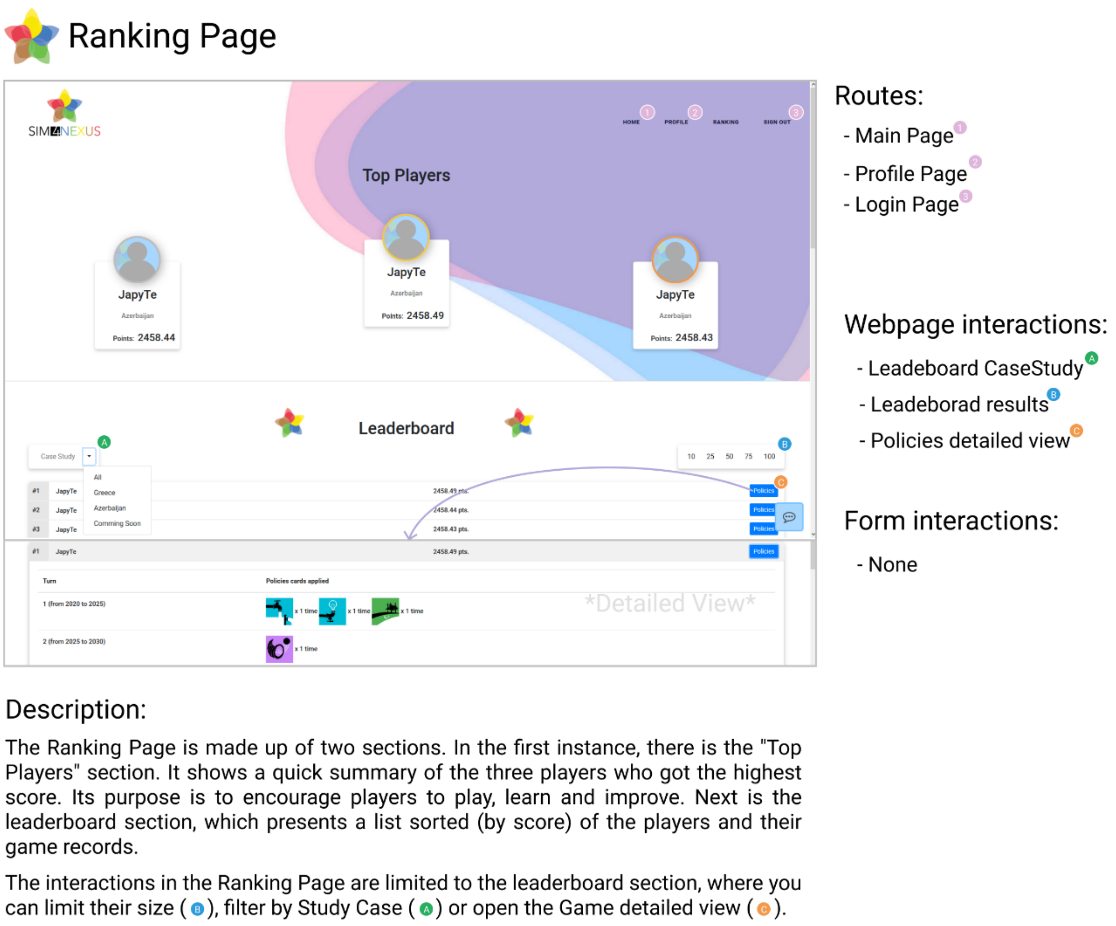
2.5.8 Ranking Page

Definition

The Ranking Page is made up of two sections. In the first instance, there is the "Top Players" section. It shows a quick summary of the three players who got the highest score. Its purpose is to encourage players to play, learn and improve. Next is the leaderboard section, which presents a list sorted (by score) of the players and their game records.

The interactions in the Ranking Page are limited to the leaderboard section, where you can limit their size (A.), filter by Study Case (B.) or open the Game detailed view (C.).

Schema



2.5.9 PreGame Questions

Definition

Once a game mode was selected a pre-game questionnaire needs to be filled (A., B., C., D.) in order to access into the SeriousGame. This form will adapt the questions depending on the Case Study and the game step where the user is.

e.g. CS (Greece) and Step (*PreGame*)

The User can decide between closing the modal or "pop-up" and going back (1.) to the *GamePage* or "Send" the answers (2.) and start playing into the SeriousGame. User can also play the game without doing the form thanks to the "Skip" option (3.).

Schema

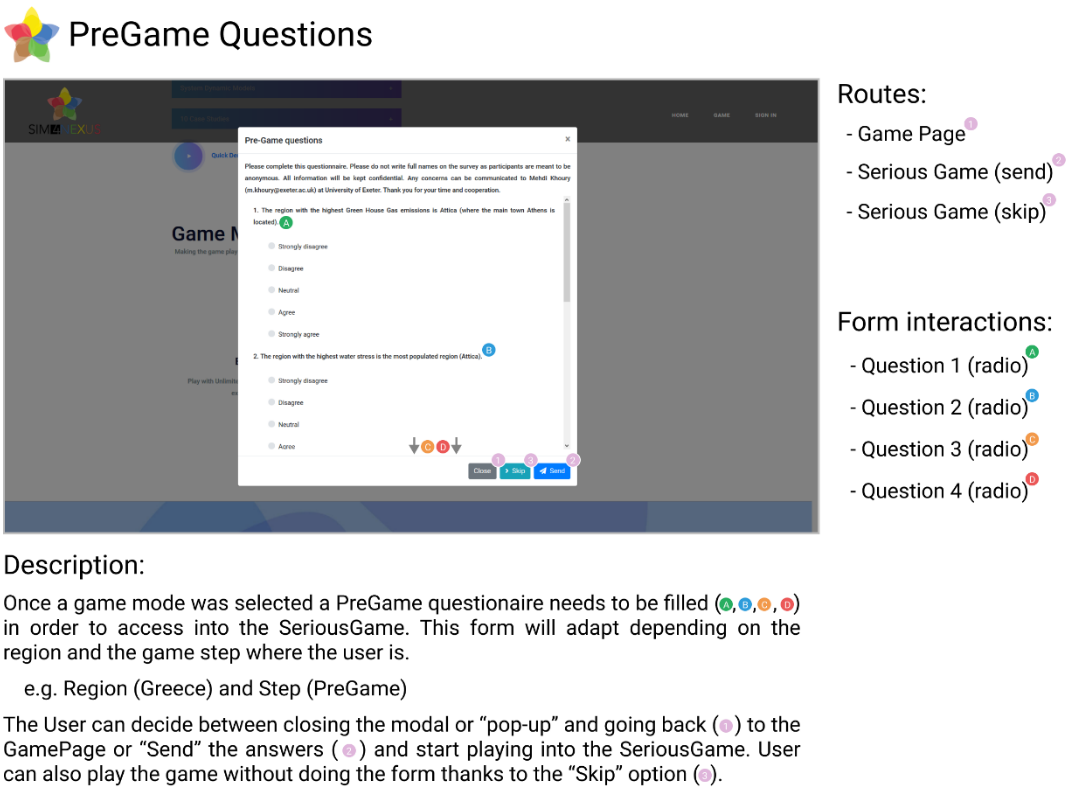


Figure 11: Pregame questions page graphical representation and interaction.

2.5.10 Serious Game

Definition

The Serious Game is using a ground-breaking design that allows the participants to easily explore a complex model of the Nexus of interactions between Water, Land, Energy, Food and Climate that can easily involve thousands of variables intricately linked together via multiple feedback loops. Players will be usually guided by having to first answer a list of simple questions (PreGame Questions) targeting isolated aspects of the Nexus i.e. make Energy affordable - and then later more open-ended questions combining several aspects of the Nexus i.e. make energy affordable while limiting the consumption of water and the emission of Green House Gas. The game interface is allowing players to switch between two different view modes: a generic default view that allows the user to drag and drop policies and see results on the Nexus health, and a detailed view where the user can conduct an in-depth analysis of the model while playing the game. The different components inside the user interface can all be zoomed in and out, and the user can focus and zoom in on any component using either mouse keyboard or touchpad, or zoom out to see the whole system. Because this kind of model can have thousands of variables, the ability to visualize the system as a whole or to focus on sub-components at any moment is a vital part of the design.

Schema

GENERIC VIEW

As shown in the figure below (Figure 12. Generic view of the Serious Game interface.), the Generic view contains a Nexus health score display, a strategic map that shows measures of stress or performances for different regions, a card panel that displays different types of policy cards available, and a timeline where players can drag and drop policies and watch the resulting combination of interventions being applied throughout time.

Figure 12. Generic view of the Serious Game interface.

THE NEXUS HEALTH SCORE PANEL.

The Nexus health score panel allows the player to understand how well the system performs from a Nexus health perspective.

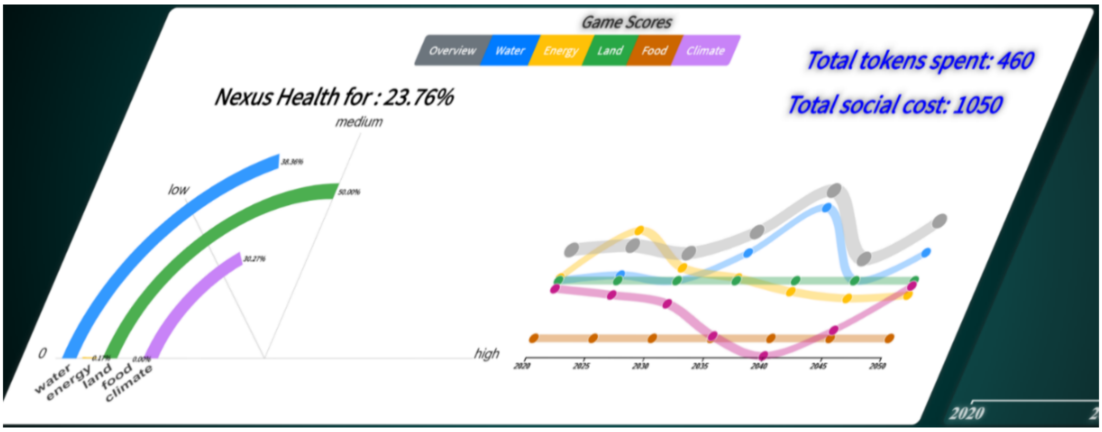


Figure 13. The Nexus Health Score panel.

The general overall nexus health score in the present game turn is displayed at the top of the panel. It is the mean of the five health scores from the different areas of the Nexus: Water, Energy, Land, Food, and Climate as detailed individually in the rainbow arc-graph showed in the lower left side. The time series line charts in the lower right side shows how the different Nexus health scores have been changing for every five years turn from 2020 to 2050. The resources spent so far after applying policies are displayed in the upper right corner. Hovering the mouse over the time series displays the percentages, while right clicking on one of the bar charts of the rainbow gives a detailed line chart showing the evolution of the related health score. Through this setup, the user can (a) appreciate the current overall health of the overall Nexus but also the different five areas, (b) find out how the health scores have evolved with time due to some external constraints or policy based interventions. Finally, the buttons at the top allows the user to see how the health score was computed for each area of the Nexus. When clicking on the Water button, for example, one can see the detail of policy goals scores and how they were obtained from policy objectives.

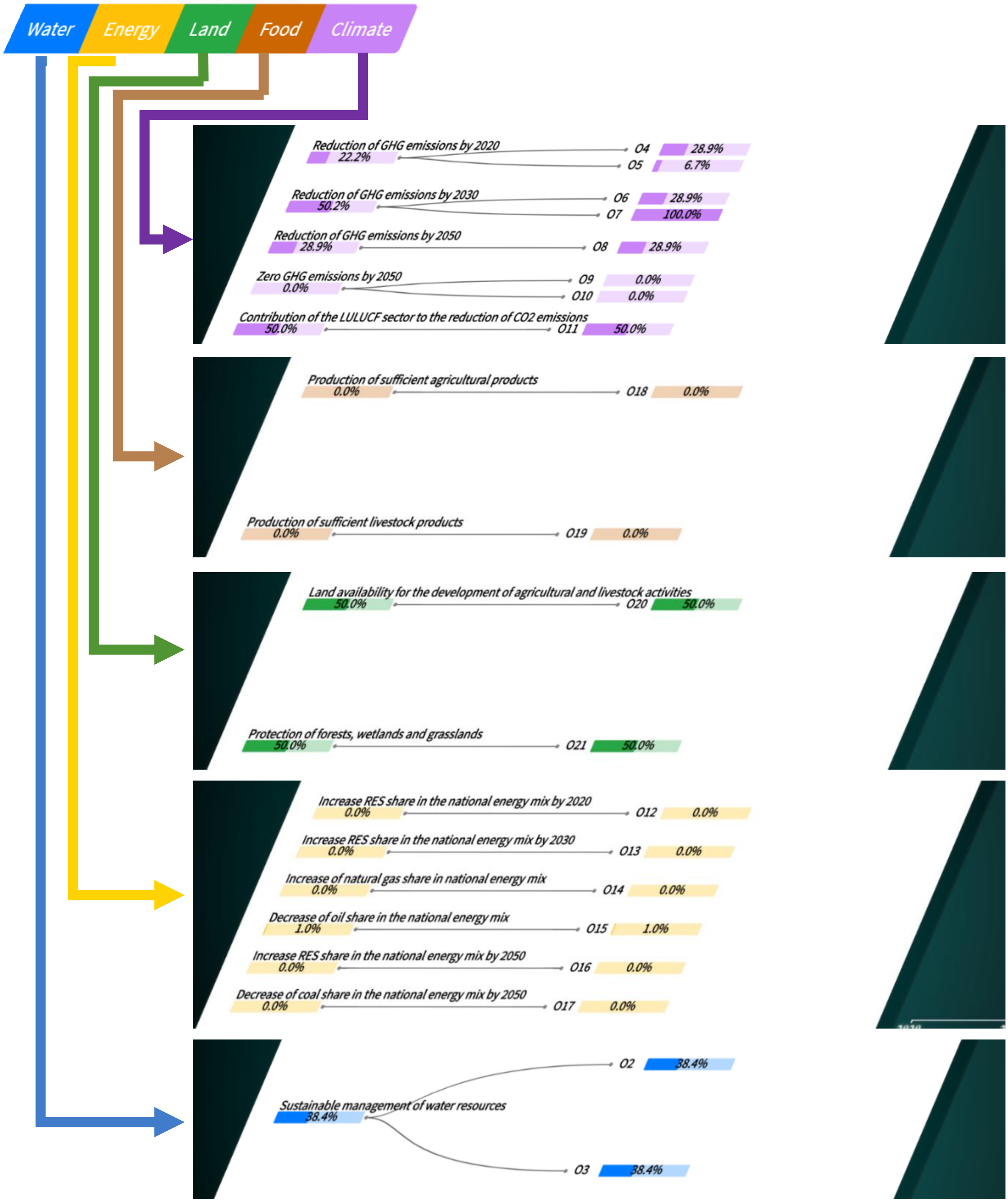
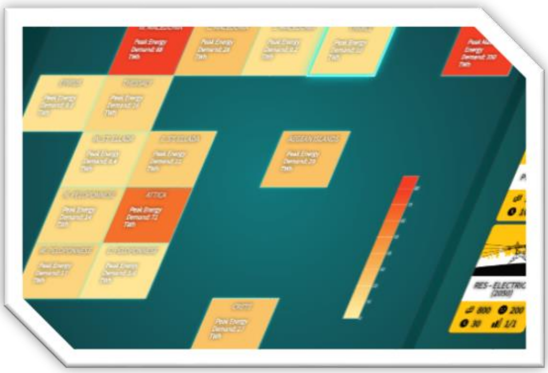


Figure 14. Looking at the different policy goals and policy objectives scores for each area of the Nexus by clicking on the corresponding button.

THE STRATEGIC MAP.

The strategic map allows the user to identify which regions are doing well and which ones are doing poorly regarding indicators of stress in the Nexus area of Water, Energy, Land, Food, and Climate. The stress indicators vary depending on the case study considered, although water stress and Green House Gas emissions tend to be the preferred indicators for the Water and the Climate Nexus areas. Decisions to address problems can then be taken by prioritizing the regions identified as problematic and applying specific policies chosen from the policy card panel.

Standard strategic map view modes for all regions (Water. Energy, Land, Food and Climate)	Corresponding GIS views
	
	
	

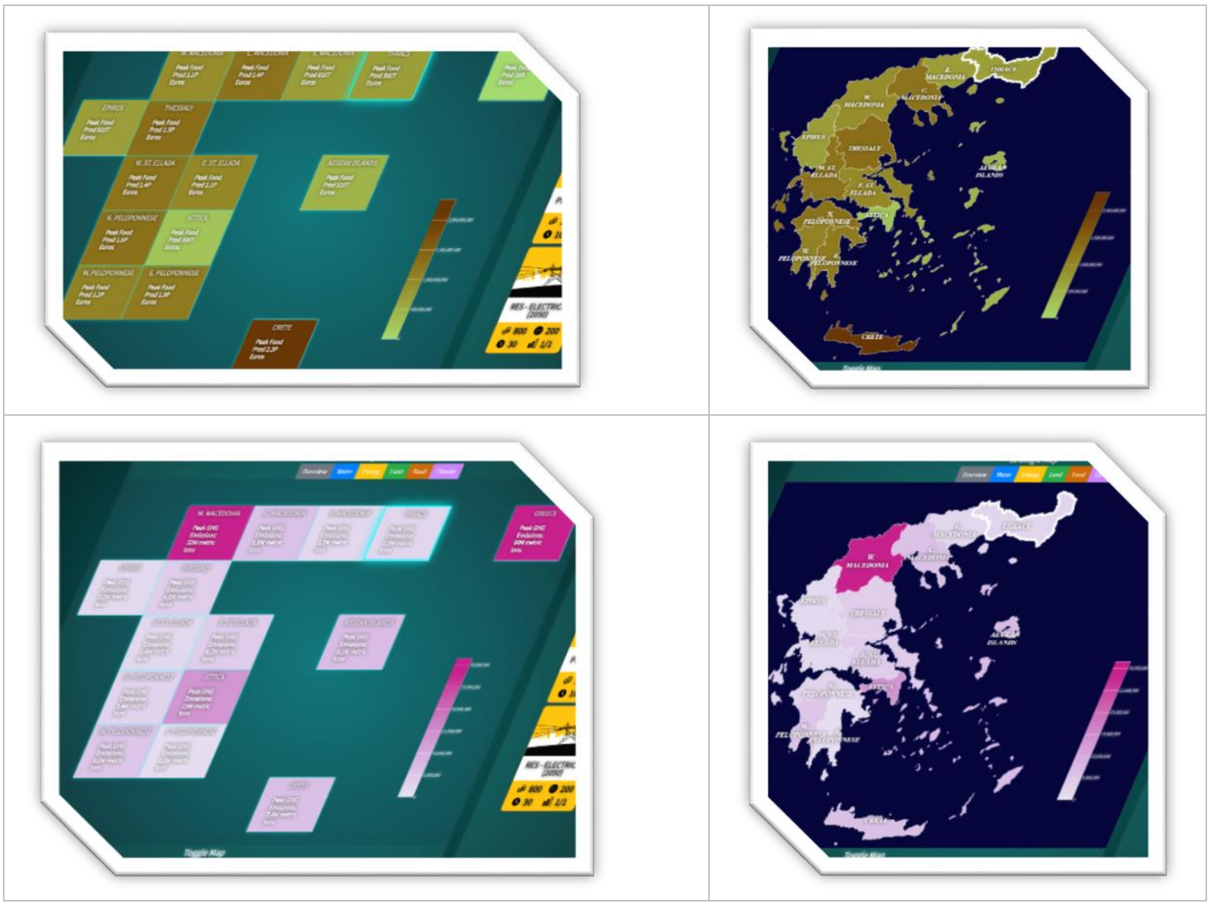
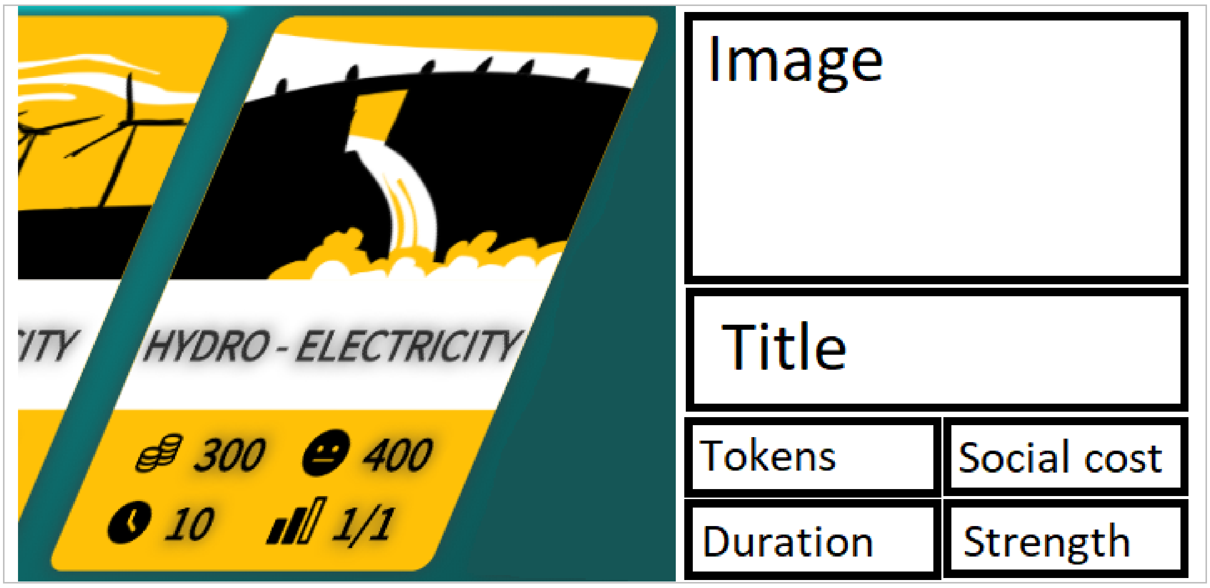


Figure 15. Looking at the different strategic map views modes corresponding the different areas of the Nexus.

THE POLICY CARDS PANEL.

Each case study has a different set of policy cards that relevant to the five areas of the Nexus. Yellow cards are energy policies, blue are water policies. Green cards are land related policies while brown cards are food related. Finally, purple cards are climate related.



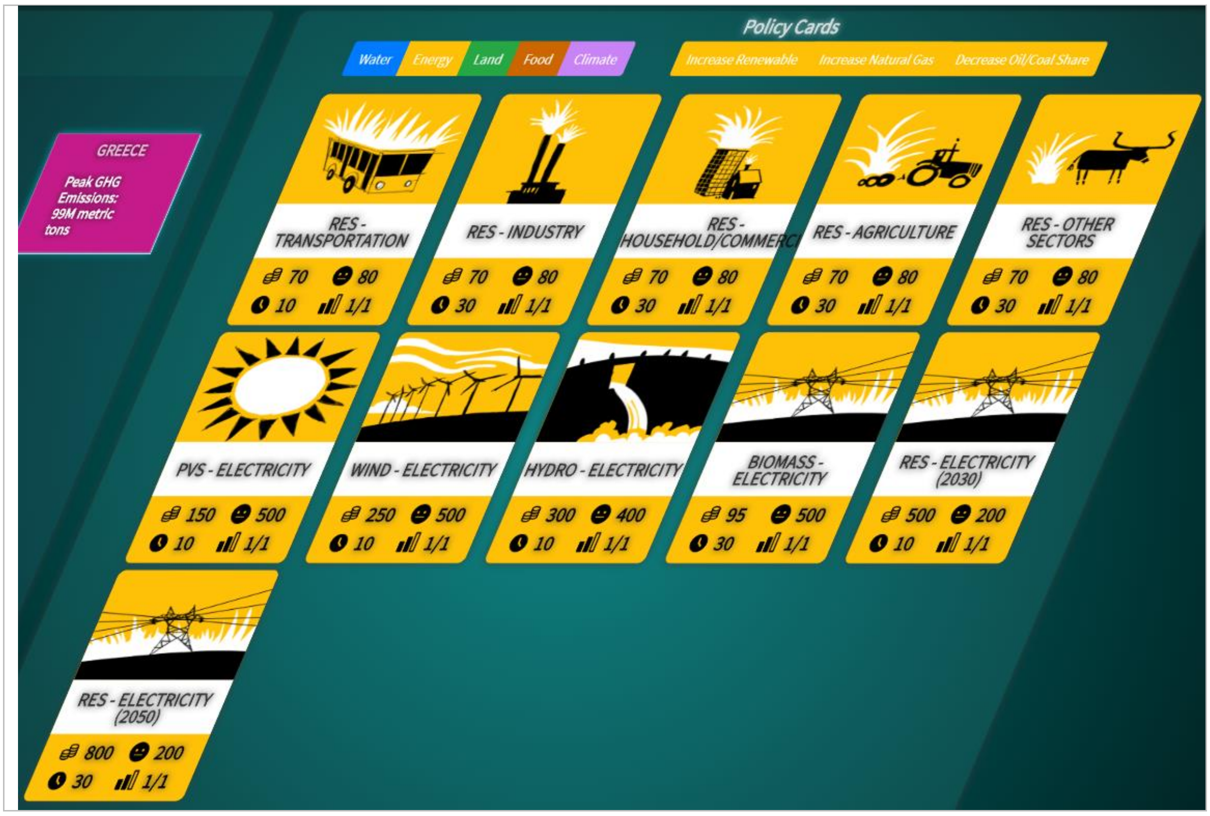


Figure 16. Individual policy card design at the top - and overall policy card display in game at the bottom.

Cards can be dragged and dropped inside the time line, and the policy will then be applied to the region selected in the strategy map.

THE TIME LINE.

The time line shows the policies that are being applied between 2020 and 2050. It is a dynamic adaptive Gantt chart that grows when the user adds different policies, and display a different content depending on the region selected.

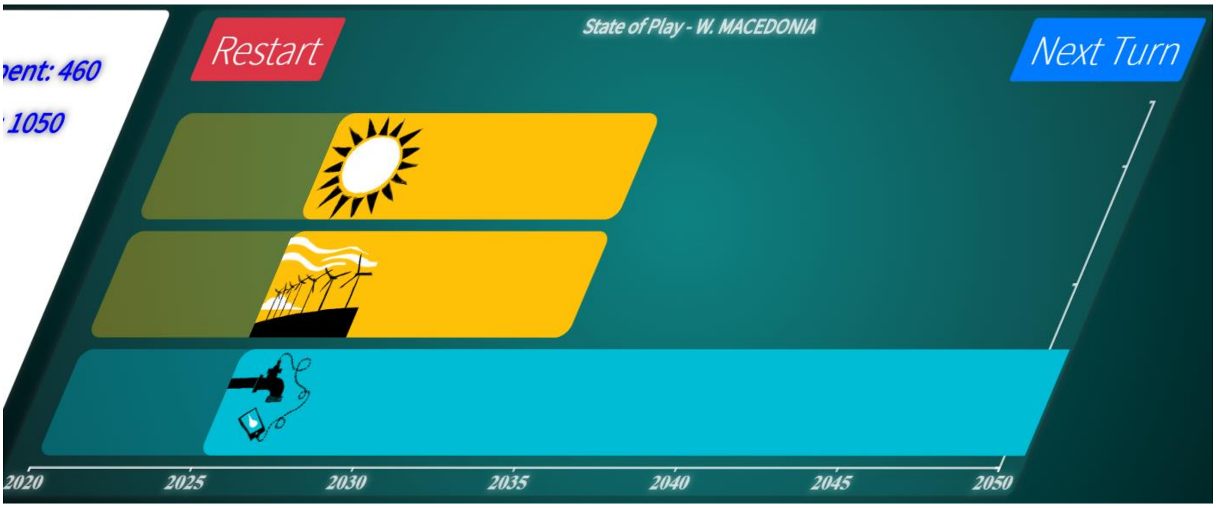


Figure 17. The timeline as a dynamic Gantt chart showing applied policies.

DETAILED VIEW

The detailed add accessible by pressing a button, adds extra components to the view. An analytic tree-map used for checking the composition for each region, a result table with time series for relevant variables, and a model view that shows the hierarchy of variables for the whole model and how policies can impact individual variables.

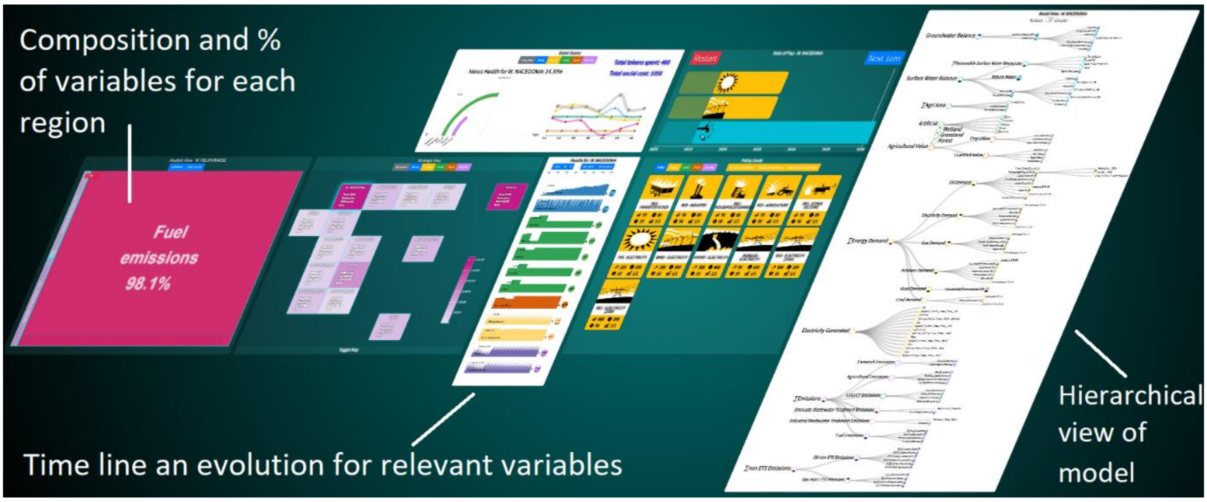


Figure 18. Detailed view mode of the Serious Game interface.

THE RESULT VIEW PANEL

The user can peruse in detail the time series of the most important variables. When clicking on a chart, one can then see the graphs of the "children" variables used to compute the "parent" variable. For example, if one click on the Groundwater balance graph, you will then access two graphs: the Groundwater demand, and the water inflow resulting from rainfalls. The user can zoom on the time series and hover on the area chart to see displayed the exact values. A button allows the user to switch between standard area chart and a horizon graph that emphasizes better vertical variations by overlaying darker colours.

Figure 19. The user can switch between area charts and horizon charts to investigate the time series variations.

The user can also investigate inside each region selected via the strategy map thing like the relative importance of certain variables i.e. if in one region emissions are mostly due to Oil rather than Gas, or perhaps due to both equally.

THE ANALYTIC VIEW PANEL

Once the user selects a region, he can investigate for example the relative share of different types of emissions in the case of climate. All five areas of the Nexus have a different tree map that can be explored. It can be useful to check that region A has mostly ETS fuel emissions while region B has mostly non-ETS emissions. In that case, it is better to apply ETS-emissions based policy cards to region A and the other type to region B. This allows the player to understand what policies will work better for which region.

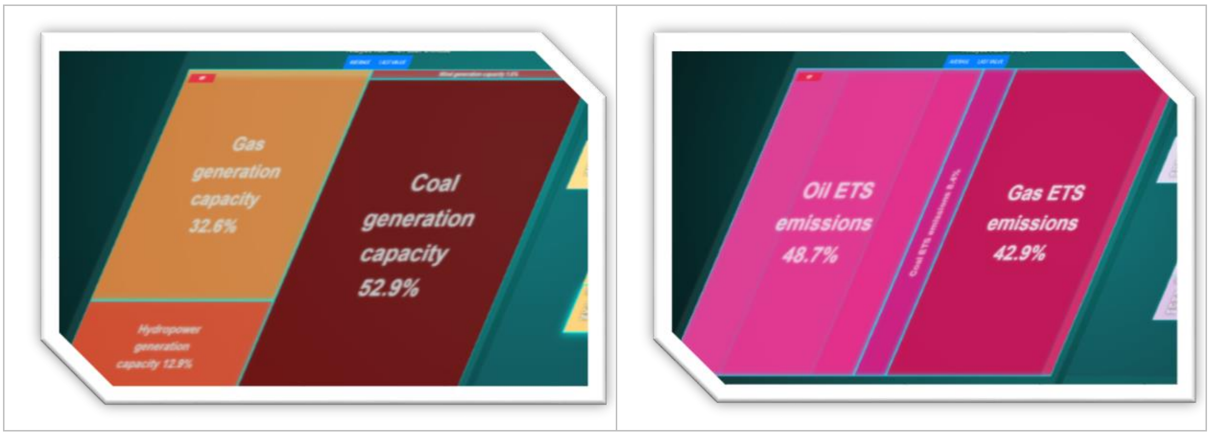


Figure 20. The treemap compares the shares of the different types of electricity generation on the left, and the shares of the different types of fuel emissions on the right for one region.

THE HIERARCHICAL MODEL VIEW PANEL

This panel display all the variables of the model behind the game. These variables are arranged in a hierarchical manner as connected nodes. The "top" nodes on the left of the tree represent the most important variables such as for example the total energy demand. The children nodes linked to these variables are the ones that were used to compute the parent nodes i.e. the energy demand requires the following children nodes to be calculated: oil demand, electricity demand, gas demand, biomass demand, and coal demand. Some of the children end nodes are hidden by default increase the readability of the tree. The user can click on grey node to reveal the full extent of the sub trees contained in their children nodes. Right clicking on any node shows an area chart of the time series of the associated variable. In short, this view allows participants to see the whole chain of variables contained in the model with complete transparency.

Another feature is that when clicking on one of the policy cards in the policy card panel, the variables directly impacted by that policy as well as the chain of parent variables linked to it inside the Nexus are highlighted in blue. This way, the user can see the impact of a particular policy at the Nexus level. The layout of the hierarchical tree can be changed from vertical to circular as shown in the two pictures below. Although the vertical layout is the most readable, the circular layout literally illustrates the concept of Nexus as it shows it at to the centre of the structure when linking the influence of a policy on different areas of the Nexus to the centre piece.

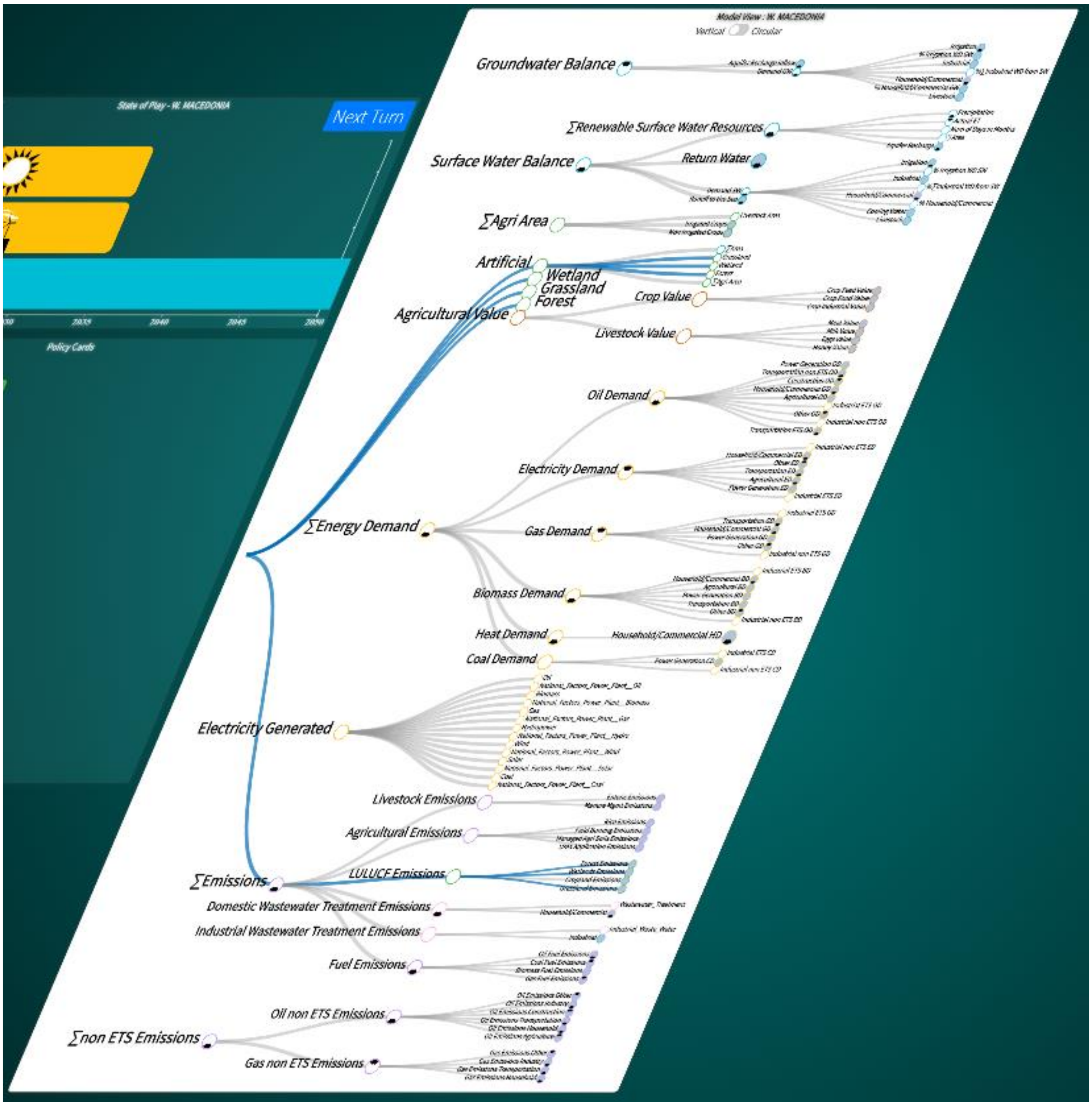
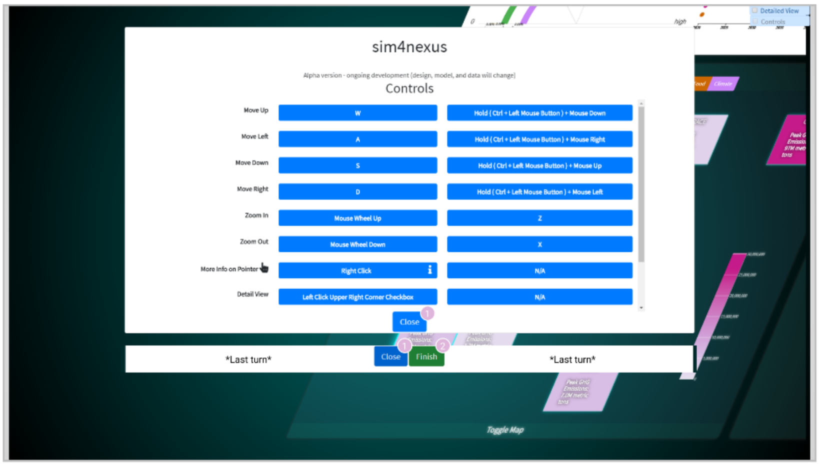


Figure 21. The vertical layout of the hierarchical model of the nexus. The paths highlighted in blue correspond to the chains of variables impacted by a land use/biodiversity type of policy.

Figure 22. The circular layout of the hierarchical model of the nexus shows the Nexus at the centre of the paths highlighted in blue representing the chains of variables impacted by a land use/biodiversity type of policy.

Finally, once the last turn ends (year 2050, turn 6) the Close button (1.) which has been closing all the modals or "pop-up" will redirect the User into the post-game Questions stored in *MainPage* where the User will end the flux.

SeriousGame



Routes:

- Next turn / See results
- MainPage

Form interactions:

Description:

When the User is inside the SeriousGame he/she will now be able to leave by its own unless the last turn (turn 6) ends.

Once the last turn ends the Finish button ( ) will appear and redirect the User into the PostGame Questions stored in MainPage where the User will end the flux.

Figure 23: SeriousGame page graphical representation and interaction.

2.5.11 PostGame Questions

Definition

When the SeriousGame ends the User is redirected into the *MainPage*. At this point, the application encourages the User in order to get the last answers about their experience within the game.

Like the pre-game questionnaire, this one is loaded according to the Case Study and the Game Step, when all the questions (A., B., C., D.) are filled, the User can send them through the "Send" button (1.).

The application also gives the option to cancel the modal or "pop-up" if the User wants to. This option can be done thanks to the "Close" button (2.). Otherwise, the modal can't be closed.

Schema

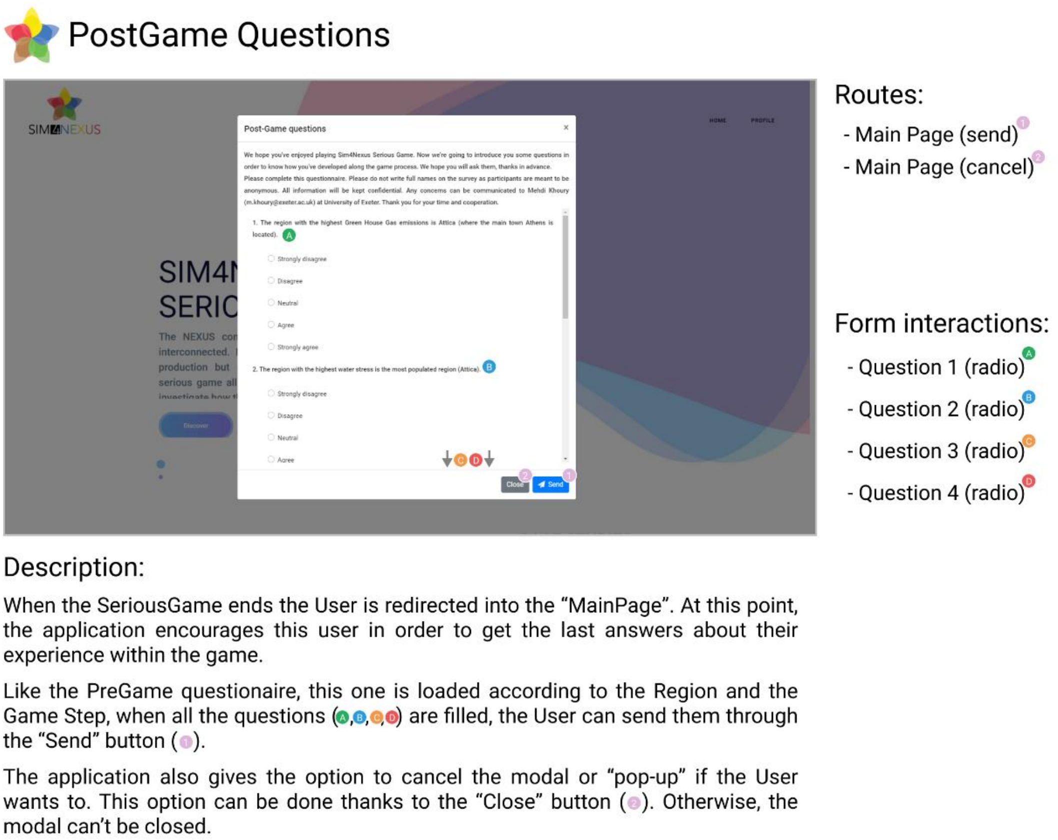


Figure 24: Postgame questions page graphical representation and interaction.

2.5.12 Global CS demonstration tool

The Global CS has implemented a demonstration and interactive visualization tool of scenarios to share results with the public.

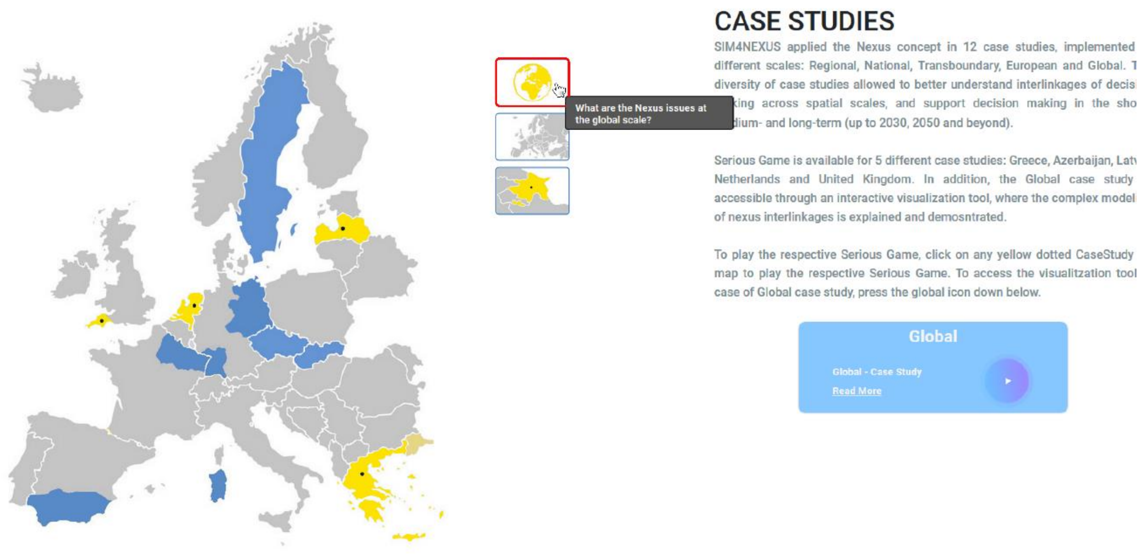


Figure 25. Main Page, map detail, Global CS tool selection

The tool is accessible from the scenario map by selecting the global CS which redirects the user to the following web sites:

- Global CS physical flow diagram: https://seriousgame.sim4nexus.eu/global/physical-flow-diagram.html
- Global CS system diagram: https://seriousgame.sim4nexus.eu/global/system-diagram.html

Global CS physical flow diagram

The global diagram displays an interactive guided tour to the exploration of physical flows associated with the Nexus at the global level. Once clicks to confirm that he wants to follow the guided tour, he is redirected to a diagram showing the current situation of flows between sectors in the Nexus of water, land, energy, food and climate. Next, the user is shown that hovering over the sector nodes in the diagram can reveal how the different sectors are connected.

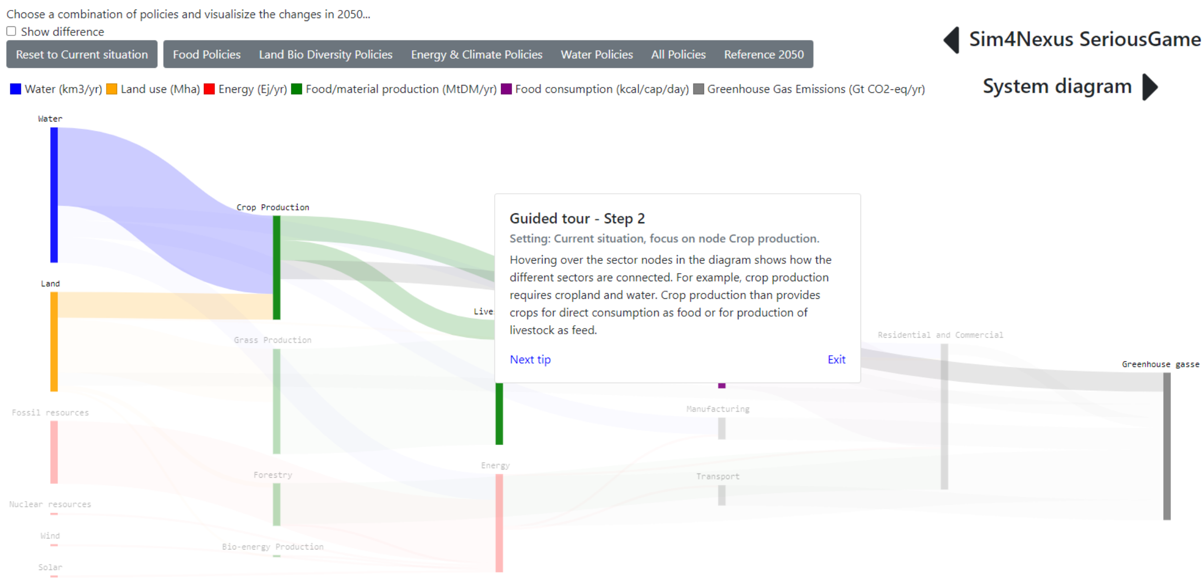


Figure 26. Global CS physical flow diagram – Guided tour - step 2

In another example, the residential and commercial sector is emphasized in a similar manner. The next step shows the user how clicking on a different scenario called "Reference 2050", allows a visualisation of the major flows in the Nexus after a change from the year 2010 to 2050 under a business as usual situation.

Next, by pressing a button, the user can see the major changes of flows where negative and positive changes of flows are emphasized respectively in gradients of blue and red.

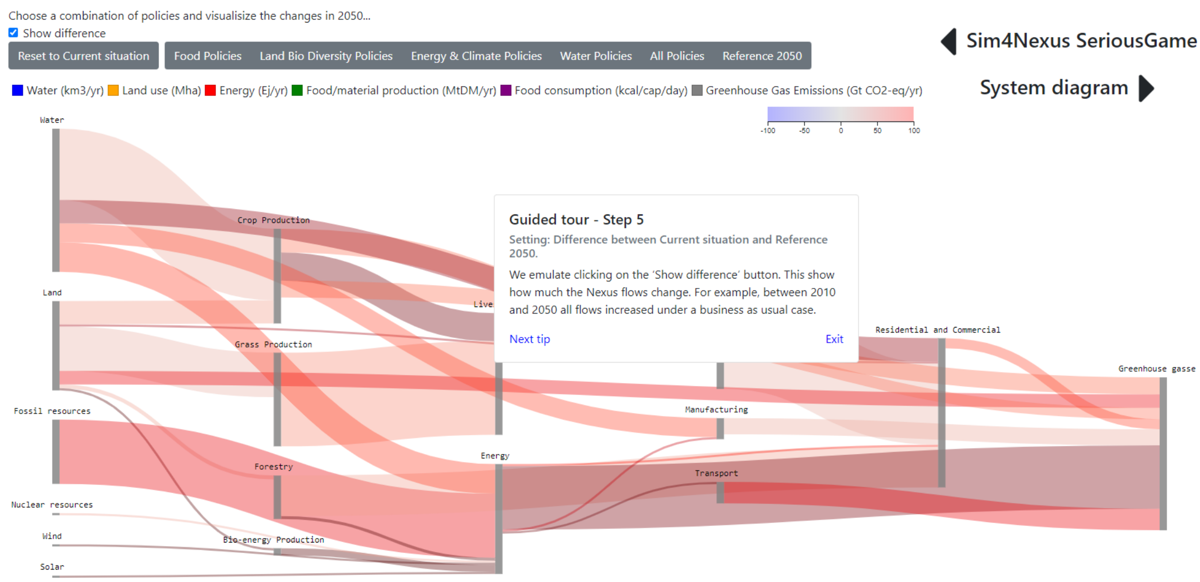


Figure 27. Global CS physical flow diagram – Guided tour - step 6

Next, the guided tour focuses on the difference between "Current situation" and "Reference 2050" scenarios for Greenhouse gasses emissions, then compares the change beween a "Reference 2050" situation and an "energy and climate policies" , and then a "flood policies" scenario.

The user is then free to explore the different changes of flows between scenarios on his own.

Global CS system diagram

The system diagram is a novel type of interactive visualisation graph that mixes arc diagrams and adjacency matrices to produce directional flows between different areas of the Nexus depending on which area the user hover on. It lends itself to telling a story that explores the system dynamic interactions by nexus area. A guided tour presents its main functionalities.

For example, the user starts with an overview of the system and notice that one blue node is bigger than the others because it is more connected. This is the "Global Mean Temperature Change" node.

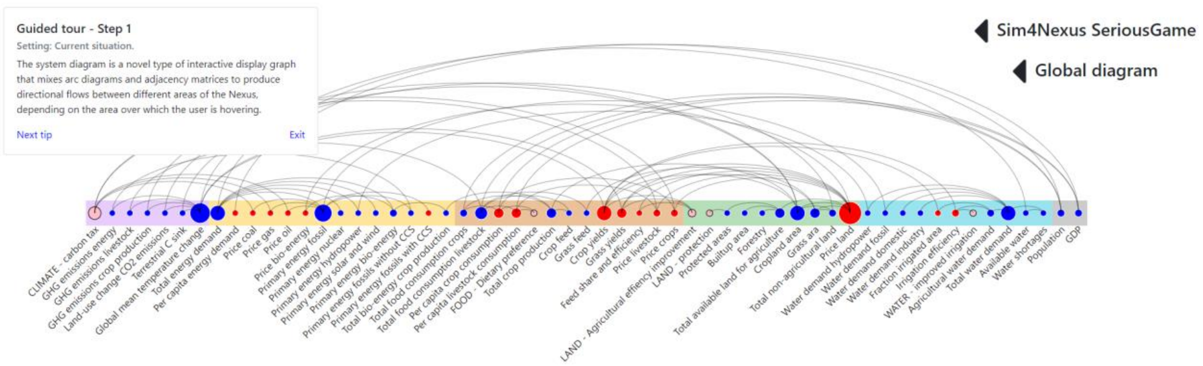


Figure 28. Global CS system diagram

If the user desires to explore what story there is to tell about the "Global Mean Temperature Change", he can then use the mouse to hover around that node. The following diagram as shown in the figure below appears.

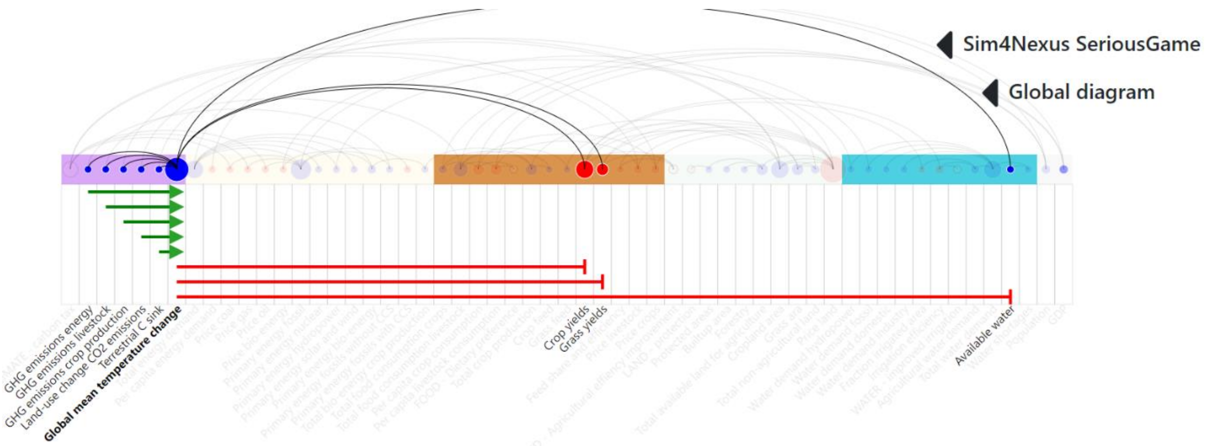


Figure 29. Global CS system diagram detail

Firstly, one can tell at a glance that the "Global Mean Temperature Change" has links to other nodes situated in general areas of the Nexus with distinct colours: the purple area (climate), brown area (Land), and blue area (water).

Second, when looking at the green arrows that express a positive correlation relationship, on can say that when GHG emissions from energy, livestock, crop production, land use and terrestrial C-sink increase, so does the global mean temperature.

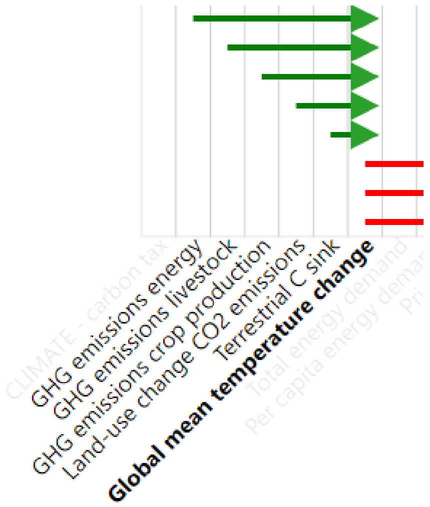


Figure 30. Global CS system diagram - green arrows detail

When looking at the red connections that end in a T shape (negative correlation or inhibition), one can observe that an increase in global mean temperature decreases crops yield, grass yield, and the available water.

Figure 31. Global CS system diagram - red connections detail

A similar process can be used to extract different stories for every single node in the diagram, allowing a complete and meaningful exploration of the relationships between nexus components.

3. Knowledge Elicitation Engine (KEE)

3.1 Introduction

The Knowledge Elicitation Engine (KEE) is being developed under "Task 4.4 Knowledge Elicitation Engine", in "WP4 Serious Game development and testing".

The current version of the KEE is fully operative and five Case Studies are already integrated: Greece, Azerbaijan, Latvia, the Netherlands and the southwest of the UK. The integration of a new Case Study is totally automatic, provided that the following requirements are met: case study information (name, description and other metadata), learning goals, policy cards, policy objectives, policy goals, nexus health and other case study specific information which will be shown to the players during the Game experience, and the System Dynamic Model (SDM).

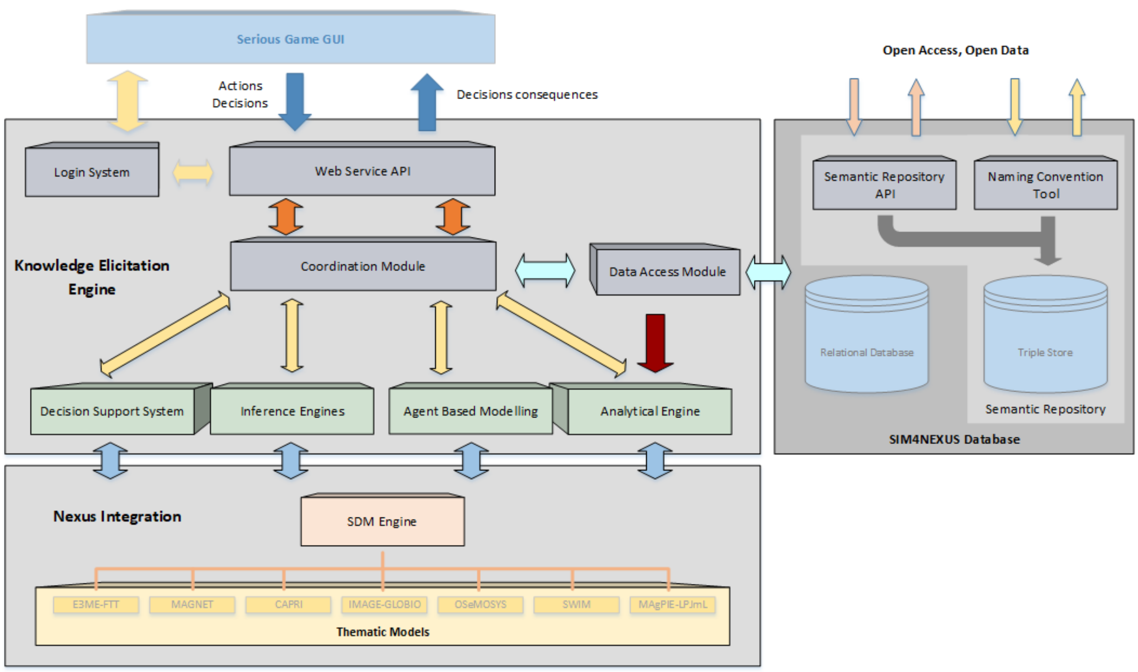


Figure 32: Serious Game Architecture, including the KEE

The main responsibilities for the KEE are:

- Provide the knowledge stored in the Semantic Repository to the Serious Game user interface (learning goals, scenario setups, available policies, restrictions, etc.)
- Embed, deploy and run the SDM models coming from WP3 considering the current game status and adapting the different parameters accordingly
- Compute Game performance indicators at each turn
- Provide the contents to be displayed to the user in the Serious Game user interface

- Collect user actions and reactions while facing different scenarios and situations in the game
- Learn from these actions to provide advice also considering knowledge in the Semantic Repository and Sim4Nexus databases
- Provide virtual players based in agents (ABM)

3.2 Knowledge Elicitation Engine

The KEE (Figure 32) includes the Web Service API, the Login System, the Coordination Module, the Data Access Module, the Game Decision Support System (DSS), the Inference Engine (IE), the analytical engine and the Agent-Based Modelling (ABM).

The KEE relies on the HTTP Apache Server to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. Thanks to the characteristics of this tool, together with the way the KEE has been designed and implemented, the KEE has the required capacities to isolate any problem and continue working while ensuring other basic characteristics such as availability, capacity, interoperability, performance, reliability, robustness, safety, security (HTTPS), and usability.

3.2.1 Web Service API

The Web Service API provides the communication between the SG UI and the KEE, dealing with all the requests and responses. Its main processes (the key processes of the Game flow) are accessible through a Service Oriented Architecture (SOA), using XML, which implements OGC® Standards[1] for the information exchange. A RESTful approach has been implemented to facilitate the data exchange in secondary Game processes, such as the gathering of the initial case study information shown in the main pages of the Game.

Details of the available web service endpoints and routes are listed in the following sections.

[1] https://www.opengeospatial.org/standards/owc

CASE STUDY SYSTEM

Table 3: Case study route system analysis.

Location	/kee/case_ studies
Methods	GET, POST
Headers	-
Body	{}
Response	Message

	[200, OK]:
	{
	  “policy_id1”: {
	    “case_study”: “2”,
	    “nexus_sector”: “Water”,
	    “name”: “Water savings in the household/commercial …”,
	    “type”: “@CaseStudy”,
	    “leader”: “Maria Papadopoulou”,
	    “summary”: “Water saving in households by establishing …”,
	    “affected_nexus_comp”: {“Climate”: {“Food”}, “Water”: {…}, …},
	    “learning_goals”: [LearningGoal1, LearningGoal2, …],
	    “decision_making_actors”: {“National energy agency”: False, …},
	    “policy_goals”: {“Food security”: False, …},
	    “indicators”: {“Population”: True, …},
	    “eurbdcode”: “ZXY1”,
	    “rbdname”: “ZXY1”,
	    “surface”: 123213,
	    “coordinates”: {},
	    “regions”: {1: Region, 2: Region, …}
	  }
	}

	Cookies
	-
Description	Every Case Study represents a study area selected for the Sim4Nexus SeriousGame project. Each new area will be treated in a unique way and with its respective values. Case Study parameters are what makes them different from one each other. Operations which can be performed over the location of this system can be ‘GET’ or ‘POST’ requests. Thanks to the ‘GET’ request we can obtain the different Case Studies allocated in the Triple Score DB. Meanwhile, the ‘POST’ operation allows us to introduce more Case Studies to the already existent set of them.

LEARNING GOALS SYSTEM

Table 4: Learning Goals route system analysis.

<table>
<tr><td>Location</td><td>/kee/learning_goals</td></tr>
<tr><td>Methods</td><td>GET, POST</td></tr>
<tr><td>Headers</td><td>-</td></tr>
<tr><td>Body</td><td>{}</td></tr>
<tr><td rowspan="4">Response</td><td>Message</td></tr>
<tr><td>[200, OK]:
{
"id1": {
"description": "You will learn how national policies in the domains of water management, renewable power production, and land use affect each other and result in changes in food production, tourism, greenhouse gas emissions, and quality and quantity of water resources."
},
"id2": {…},
"id3": {…}
}</td></tr>
<tr><td>Cookies</td></tr>
<tr><td>-</td></tr>
<tr><td>Description</td><td>Learning Goals system provides all the Learning Goals available in the Semantic Repository and defined in T4.1 Learning Goals definition.
Method 'POST' is also allowed but it will require to declare a JSON containing one or more Learning Goals in order to be stored as new ones.</td></tr>
</table>

POLICY CARDS SYSTEM

Table 5: Policy Cards route system analysis.

<table>
<tr><td>Location</td><td>/kee/policy_cards</td></tr>
<tr><td>Methods</td><td>GET, POST</td></tr>
<tr><td>Headers</td><td>-</td></tr>
<tr><td>Body</td><td>{}</td></tr>
<tr><td rowspan="4">Response</td><td>Message</td></tr>
<tr><td>[200, OK]:
{
“policy_id1”: {
“case_study”: “2”,
“nexus_sector”: “Water”,
“name”: “Water savings in the household/commercial …”,
“short_name”: “Water savings at homes/hotels”,
“description”: “Water saving in households by establishing …”,
“level”: 1,
“permanent”: “Yes ”,
“applied_times”: “Multiple”,
“effectiveness”: 0,
“pre_init_10_to_15”: 0,
“pre_init_15_to_20”: 0.0,
“building_time”: 5,
“active_time”: 30,
“cost_qualitative”: “Medium”,
“cost_value”: 60,
“cost_generated_qualitative”: “Medium”,
“social_cost_qualitative”: “Medium”,
“social_cost_value”: 50,
“social_cost_generated_qualitative”: “Medium positive”,
“social_cost_generated_value”: 90,
“included_in_thematic_model”: “No”,
“model_input_translation”: “Decrease of water demand by the …”,
“comments”: “Column H: From 2020 - Until 2050”,
“sub_class”: “”,
“image”: “WaterSavingSmartTaps.png”
},
“policy_id2”: {…}
}</td></tr>
<tr><td>Cookies</td></tr>
<tr><td>-</td></tr>
<tr><td>Description</td><td>Policy Cards generic system provides the entire set of data which the KEE manages. PolicyCard endpoint also allows to upload PolicyCards by performing a ‘POST’ request into it. That request must have a LearningGoals looking like JSON. The SeriousGame uses Policy Cards in order to modify the game state while adding progression and value to a game session.</td></tr>
</table>

POLICY GOALS SYSTEM

Table 6: Policy Goals route system analysis.

Location	/kee/policy_goals	
Methods	GET, POST	
Headers	-	
Body	{}	
Response	Message	

	[200, OK]:
	{
	"id1": {
	"case_study": "2",
	"nexus_sector": "Water",
	"name": "Water savings in the household/commercial …",
	"description": "Water saving in households by establishing …",
	"thresholds": {
	"low": 0.33,
	"medium": 0.66,
	"high": 1.0
	},
	"weights": {
	"O2": 0.5,
	"O3": 0.5
	}
	},
	"id2": {…}
	}	

	Cookies	
	-	
Description	Policy Goals system allow the upload and download of Policy Goals. Policy Goals contain all the information related to the different Goals of a given Study Case. Goals are the last point of the SeriousGame iterations.	

POLICY OBJECTIVES SYSTEM

Table 7: Policy Objectives route system analysis.

Location	/kee/policy_objectives
Methods	GET, POST
Headers	-
Body	{}
Response	Message

	[200, OK]:
	{
	"id1": {
	"case_study": "2",
	"name": "Water savings in the household/commercial …",
	"description": "Water saving in households by establishing …",
	"national_formula": "initial = get_first_year('National_Demand_SW…",
	"regional_formula": "initial = get_first_year('RBD_W_GRXX_Demand…"
	},
	"id2": {…}
	}

	Cookies
	-
Description	Policy Objectives represent a partial goal which needs to be reached. Once the game evolution reaches one of these Policy Objectives computation can be done in order to know the game performance for a specific game session.

LOGIN SYSTEM

The Login system is part of the Web Service API and provides authentication and authorization to the Game, validating all the KEE incoming communications. It is based on Json Web Tokens [2](JWT), which are an open, industry standard RFC 7519[3] method for representing claims securely between two parties.

Table 8: Login route system analysis.

Location	/kee/login	
Methods	POST	
Headers	Content-Type: application/json	
Body	{ "email": "johndoe@example.com", "password": "guestw" }	
Response	Message	
	[200, OK]: Successfully logged in [401, Unauthorized]: Invalid login request. Tip: content: application/json, fields: email, password	
	Cookies	
	[200, OK]: access_token_cookie: eyJ0eXAiOiNiJ9. […] B3PgPqdbZtSoBCyBgI refresh_token_cookie: eyJ0eXAiOiJKVOiJIUzI1NiJ […] H40Lz25Tcu2KzzOg	
Description	Kee version of /login ('/kee/login') only allows 'POST' method to be used. The request body must contain the User email and password. As a response, it will provide the authentication credentials inside response Cookies which are located in response headers. There will be two tokens ("acces_token_cookie" and "refresh_token_cookie"). These two parameters can be used in order to know who is doing the request to the back-end. From now in advance, users will use these parameters in their request headers in order to verify their session and identity. This functionality can be tested using '/whoami' endpoint.	

LOGOUT SYSTEM

[2] https://jwt.io/
[3] https://tools.ietf.org/html/rfc7519

Table 9: Logout route system analysis.

Location	/kee/logout	
Methods	GET	
Headers	Cookie: "access_token_cookie=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9. […] B3PgPqdbZtSoBCyBgI; refresh_token_cookie=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ […] H40Lz25Tcu2KzzOg; session=.eJwNyNEKgyAUANBfudznGFnTXG […] C39dYBNaWd0bwGW6sqCuZJahNSA "	
Body	{}	
Response	Message	
	[200, OK]: Successfully logged out	
	Cookies	
	-	
Description	By performing a GET over the logout endpoint the User can sign out from their session and clean the cookies. The browser client will remain as no one has signed in before.	

EMAIL SYSTEM

Table 10: Email route system analysis.

Location	/kee/changeEmail
Methods	POST
Headers	Content-Type: application/json Cookie: "access_token_cookie=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9. [...] B3PgPqdbZtSoBCyBgI; refresh_token_cookie=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ [...] H40Lz25Tcu2KzzOg; session=.eJwNyNEKgyAUANBfudznGFnTXG [...] C39dYBNaWd0bwGW6sqCuZJahNSA "

Body	{
	"email": "johnDoe@mail.com"
	}

Response	Message

	[200, OK]
	{
	"email": "johnDoe@mail.com",
	"username": "johnDoe",
	"role_id": 1,
	"confirmed": False,
	"name": "John",
	"location": "Sample City"
	}
	
	[400, Bad Request]
	Bad request, missing "email" field

	Cookies
	-
Description	Thanks to the "ChangeEmail" endpoint, the client is able to change the email of an authenticated User. As a response, it will return the User instance updated with the new email on it.

QUESTIONS SYSTEM

Table 11: Questions route system analysis.

<table>
<tr><td>Location</td><td colspan="2">/kee/questions?tag=<tag:str>&caseStudy=<case_study:int></td></tr>
<tr><td>Params</td><td>Tag:
- PRE_GAME
- MID_GAME
- POST_GAME</td><td>CaseStudy</td></tr>
<tr><td>Methods</td><td colspan="2">GET</td></tr>
<tr><td>Headers</td><td colspan="2">Content-Type: application/json</td></tr>
<tr><td>Body</td><td colspan="2"><pre>{
  "title": "Sim4Nexus WS answer schema",
  "type": "object",
  "properties": {
    "question_id": {
      "type": "integer",
      "minimum": 0
    },
    "answer": {
      "type": "string"
    },
    "extra_question_answer": {
      "type": "string"
    },
    "user_id": {
      "type": "string"
    },
    "session_id": {
      "type": "string"
    }
  }
}</pre></td></tr>
<tr><td>Response</td><td colspan="2">Message

[200, OK]

[400, Bad Request]

Cookies

-</td></tr>
<tr><td>Description</td><td colspan="2">Questionnaires KEE endpoint provides the questions which relate the case study to the situation in which the user finds himself respect the Game progression.</td></tr>
</table>

ANSWERS SYSTEM

Table 12: Answers route system analysis.

Location	/kee/answers?multiple=<multiple:Boolean>
Methods	POST
Headers	-
Body	{}

Response	Message
	
	[200, OK]:
	{
	"CaseStudy": "Greece",
	"Description": "Description",
	"Id": 3,
	"Questions": [
	{
	"Id": 27,
	"Options": [
	{
	"Id": 21,
	"Value": "Strongly disagree"
	}
	]
	}
	]
	}
	
	Cookies
	
	-

Description	Answers KEE endpoint allows the Sim4Nexus SeriousGame to upload the answers related to a question formulary previously downloaded, which can be done thanks to the "Questions" endpoint. The "multiple" parameter lets the client answer the questions one by one or all at the same time.

Coordination Module API

INITIALIZATION STEP

This KEE WS endpoint manages the initialization of an S4N SG session, it is accessed by the UI to obtain the case study, initialize the game (year 2020), initial values for each parameter, load case study data, policy cards, policy objectives, policy goals, nexus health, etc.

Table 13. Initialization step endpoint

<table>
<tr><td>Location</td><td colspan="2">/kee/wps?service=<service:str>&request=<request:int>&identifier=< identifier:str>&datainputs=<input:str></td></tr>
<tr><td rowspan="5">Parameters</td><td colspan="2">Service:
WPS</td></tr>
<tr><td colspan="2">Request:
Execute</td></tr>
<tr><td colspan="2">Version:
1.0.0</td></tr>
<tr><td colspan="2">Identifier:
sim4nexus_initialization_step</td></tr>
<tr><td>Datainputs:
input={
SessionID: 9aba-e880aff102,
ScenarioId:6,
PlayerID:0,
Language:en-US,
SessionDateTime:1/1/2020 00:00:00,
ReceiveSDMOutputs:1
}</td><td>SessionID: str
ScenarioId: int
PlayerID: str (by default is 0, guest)
Language: str
SessionDateTime: Datetime dd/MM/yyyy_hh:mm:ss
ReceiveSDMOutputs: Boolean (1 yes, 0 no)</td></tr>
<tr><td>Methods</td><td colspan="2">GET</td></tr>
<tr><td>Headers</td><td colspan="2">Content-Type: application/json</td></tr>
<tr><td>Body</td><td colspan="2">{}</td></tr>
<tr><td>Response</td><td colspan="2">Message</td></tr>
</table>

<table>
<tr><td rowspan="3"></td><td>[200, OK]: {
"Budget": 1000,
"CaseStudyName": Object <CaseStudy>,
"CurrentTurnDateTime": "01/01/2020 00:00:00",
"Interventions": [
Object <PolicyCard>
],
"InterventionsHistory": {},
"InterventionsHistoryOrder": {},
"Language": "en-Gb ",
"NexusHealth": {
<nexus>_formula: "",
<nexus>_desc: ""
},
"NexusHealthScore": {
<nexus>: 0
},
"PlayerID": 0,
"PolicyGoals": [
Object <PolicyGoal>
],
"PolicyGoalsScore": {
"pg_id": 0
},
"PolicyObjectives": [
Object <PolicyObjective>
],
"Regions": [
Object <Region>
],
"SDMOutputs2020": [[]],
"SDMVars": [
"Climate__GHGb",
...
],
"SessionDateTime": "22/4/2020 13:21:53",
"SessionID": "4506-fc4e-a30050d3342e",
"SocialAcceptance": 1000,
"StateVars": {
"Water__wd_Irr": [],
...
},
"Stocks": {
"Climate__GHGb": [],
...
},
"TotalScore": 1.5410385470252457
}</td><td>- <nexus> references the components:
o Water
o Energy
o Climate
o Food
o Forest</td></tr>
<tr><td colspan="2">Cookies</td></tr>
<tr><td colspan="2">lang: <en,es,aze,…>
gameId: A000x001
access_token_cookie: eyJ0eXAiOiNiJ9. […] B3PgPqdbZtSoBCyBgl</td></tr>
</table>

Description	By sending this GET request, the KEE starts the Game initialization process. This process consists in preparing the CaseStudy context in order to start playing. The main action is to initialize and create the year 2020 context, this one involves the process of loading all the PolicyGoals, PolicyObjectives, PolicyCards, variables and regions for a specific CaseStudy given as a dictionary element "ScenarioId" inside URL parameter "datainputs". Once the context is loaded, it is sent to the User Interface (object Response) to interact with it. In which refers to the cookies: - **lang**: It is used to keep tracking of the language in which the user wants to see the interface. - **gameId**: Different param as the "sessionId". A SessionId involves a user and a game in a specific time. Meanwhile, a gameId is the reference of a game in the database. - **access_token_cookie**: It is used to retrieve a User that is logged into the application and attach the game to it to track their evolution.

SIMULATION STEP

This KEE WS endpoint manages the simulation of the S4N SG session turn, it is accessed by the UI to run each time step based on the current Game status and provide the next one.

Table 14. Simulation step endpoint

<table>
<tr><td>Location</td><td colspan="2">/kee/wps?service=<service:str>&request=<request:int>&identifier=<
identifier:str>&datainputs=<input:str></td></tr>
<tr><td rowspan="5">Parameters</td><td colspan="2">Service:
WPS</td></tr>
<tr><td colspan="2">Request:
Execute</td></tr>
<tr><td colspan="2">Version:
1.0.0</td></tr>
<tr><td colspan="2">Identifier:
sim4nexus_initialization_step</td></tr>
<tr><td>Datainputs:
input = {
SessionID: caf7-4588-9aba-eaff102,
ScenarioId: 6,
PlayerID: 0,
Language: en-US,
SessionDateTime:1/1/2020 00:00:00,
ReceiveSDMOutputs: 1,
CurrentTurnNumber: 1,
PolicyGoals: [],
Budget: 1000,
SocialAcceptance: 1000,
Interventions: {
New: [],
History: {},
Order: []
},
Stocks: {},
StateVars: {}
}</td><td>- SessionID: str
- ScenarioId: int
- PlayerID: str (by default is 0, guest)
- Language: str
- SessionDateTime: Datetime dd/MM/yyyy_hh:mm:ss
- ReceiveSDMOutputs: Boolean (1 yes, 0 no)
- CurrentTurnNumber: int
- PolicyGoals: [object<PolicyGoal>]
- Budget: int
- SocialAcceptance: int
- Interventions: Stores objects <PolicyCard>. User can only add PolicyCards into “New”. KEE will handle the rest.
- Stocks: Key-Value structure, stores the evolution of Stock variables.
- StateVars: Key-Value structure, stores the evolution of StateVars variables.</td></tr>
<tr><td>Methods</td><td colspan="2">GET</td></tr>
<tr><td>Headers</td><td colspan="2">Content-Type: application/json</td></tr>
<tr><td>Body</td><td colspan="2">{}</td></tr>
</table>

Response	Message	

	[200, OK]:
	{
	Budget: 10028000
	CurrentTurnDateTime: "01/01/2030 00:00:00"
	CurrentTurnNumber: 2
	InterventionsHistory: {
	"PolicyCardID": {
	"National": [<year:int>]
	}
	},
	InterventionsHistoryOrder: [
	{
	"Id": "2",
	"Region": "National",
	"Year": 2030
	}
	],
	"NexusHealthScore": {
	<nexus>: 0
	},
	"PolicyGoalsScore": {
	"pg_id": 0
	},
	"Regions": {
	Object <Region>
	},
	"SDMOutputs2020": [[]],
	SessionDateTime: "22/4/2020 14:46:40",
	SessionID: "45c2-4900be8f3b39",
	SocialAcceptance: 10028000,
	"StateVars": {
	"Water__wd_Irr": [],
	...
	},
	"Stocks": {
	"Climate__GHGb": [],
	...
	},
	TotalScore: 1.5541678719103018
	}
		- <nexus> references the components:
		o Water
		o Energy
		o Climate
		o Food
		o Forest

	Cookies	
	lang: <en,es,aze,…> gameId: A000x001 access_token_cookie: eyJ0eXAiOiNiJ9. […] B3PgPqdbZtSoBCyBgI	

Description	As can be seen, the Response for the Game Simulation System is quite similar to the one seen previously for the Game Initialization System. That is because both systems share a trivial similarity. Game Simulation System can be defined as a Game Initialization System which is capable of loading the CaseStudy context only taking the data sent by the User Interface and their local files. Following this definition, this system will take the modified context by the User, will interpret it, simulate the variations involved in the next 5 years, and return a new CaseStudy context (object Response) to the User interface. As the game was already initialized, this Response will not be as long as the one seen in the Game Initialization System. Referring to cookies: - **lang**: It is used to keep tracking of the language in which the user wants to see the interface. - **gameId**: Different parameter as the "sessionId". A SessionId involves a user and a game in a specific time. Meanwhile, a gameId is the reference of a game in the database. - **access_token_cook**ie: It is used to retrieve a User that is logged into the application and attach the game to it to track their evolution.

3.2.2 Coordination Module

The Coordination Module, the core of the KEE, implements all the Game logic and the main data flows, monitors all the infrastructure status, manages the different KEE modules and load and persist the SG data.

The Game logic is managed by the **Game Logic Manager**, which implements two kinds of Game steps or processes, which correspond to the main entry points accessible through the Web Service API:

Initialization: To select the case study, initialize the game (year 2020), initial values for each parameter, load case study data, policy cards, policy objectives, policy goals, nexus health, etc.

Simulation: To run each time step based on the current Game status and provide the next one.

The Game status is represented by:

The Case Study being played.

The current turn number: turn 1 corresponds to the simulation of the years 2020 to 2025, turn 6 corresponds to the simulation of the years 2045 to 2050.

Policy history: List of policies previously applied by the player.

Stocks: within the System Dynamic Model are variables that can either increase or decrease over time depending on their inflows and outflows. They are used as a starting point to simulate the next turns.

Initialization and Simulation processes share their main steps and only differ at the starting point. Initialization loads the base data (base status) from the SIM4NEXUS databases (policies applied before 2020) and the Simulation uses the previous turn state and the new actions chosen by the users.

For example, the steps executed in the Simulation process are the following:

1. From the UI a request is sent to the KEE to execute a Simulation step based on the current status and the user decisions (new applied policies).
2. Request parameters are validated:
    - New selected policies: check if the policies correspond to the selected CS and there are enough resources to apply them (budget and social acceptance).
    - Game Status:
        - Check if the Stock variables correspond to the selected CS.
    - Other inputs
3. New budget and social acceptance are calculated based on selected policies costs.
4. Policy history structure is updated with the new policies.

5. For each policy in the policy history structure, its efficiency is computed, based on its parameters, for the current simulation.
6. Specific policy structure is built to be sent to the SDM engine (SDM Manager):
7. Initial month and final month are calculated based on the current turn number.
8. Initial month, final month, SDM policy structure and previous stocks (from Game status) are sent to the SDM Manager to run the simulation.
9. The corresponding SDM is executed by the SDM Manager and the outputs (stocks and other variables) are sent back to the Game Logic Manager to compute derived indicators.
10. Policy Objectives, Policy Goals and Nexus health are calculated based on the SDM outputs.
11. Other data structures, used in the UI, are built from the SDM outputs.
12. User decisions (new policies) are persisted.
13. All the data/information is compressed and sent back to the UI.

This process can take from a few milliseconds to one or two seconds depending on the selected case study (some SDMs are more complex than others).

The **SDM Manager** (introduced in next section) is in charge of the execution of the SDMs (step 9). It takes the initial and final month, the previous stocks and the specific policies structure and then runs the SDM for the selected time slot and catch the outputs.

The Coordination Module implements other secondary functionalities that will allow the UI to create a complete serious game able to provide to users the most interesting game capabilities, such as the I18N module, which enables the translation of the S4N SG to other languages.

3.2.3 SDM Engine

This section of the document details how the SDMs, implemented in WP3, are mixed with the policies, implemented in WP2, to enrich their functionalities and how are finally integrated to the KEE to simulate the game flow through the SDM Engine.

In WP3, the SDMs are built using a complex and specific modelling software, called Stella, which is able to represent the models in a readable format where, basically, all data and equations are listed in a logic way.

Due to its format, it cannot be directly executed by the KEE, thus it has to be previously translated to Python and integrated to the game architecture through the SDM Engine, which will manage their translation and execution.

In the translation process, the engine takes as inputs the base SDM (implemented in Stella software), the Policy Cards (stored in the SR) and other Case Study specific metadata and builds a complex logic structure for each CS that enables the corresponding simulation of each Game turn, including the selected policies (user's decisions), which modify the SDM behaviour and consequently the Game status.

This process is managed by the 'Conversion script' which define and automatic flow that goes through different steps, each one in charge of a specific task in the translation logic:

1. Variable names are translated to Python nomenclature.
2. Constant data is extracted and loaded into a specific and isolated data structure.
3. Time series data is extracted and loaded into a specific and isolated data structure.
4. Equations are extracted and loaded into a specific and isolated structure.
5. Initial stocks are extracted and loaded into a specific and isolated data structure.
6. Based on the previous data structures, the Python SDM is defined.
7. Policies are included in the SDM code.
8. The Python SDM is executed from the first year till 2050 to check its correctness.
    a. The outputs are validated monthly against a validation data set.
9. The Python SDM is executed in chunks of 5 years from 2020 till 2050 to check its correctness.
    a. The outputs are validated monthly against a validation data set.

Figure 33 shows the flow of steps 1 to 7 in relation to the conversion process. This process is carried out in two parts. The Baseline Python Model is the model that reflects that which was created in STELLA. This model is executed from the first year till 2050 whereby the Python model outputs are validated against the STELLA outputs. Once validated the Policy Builder process is implemented to incorporate the Policy Card information to ready the Python model for use within the SG. The model is then executed again though this time in 5-year intervals from it's initiation date to 2050 to and validated again against the STELLA model outputs. Finally, when this process is successfully finished, the Python SDM is added to the SDM Engine to support a new Case Study and any request of execution from the KEE is redirected to this environment.

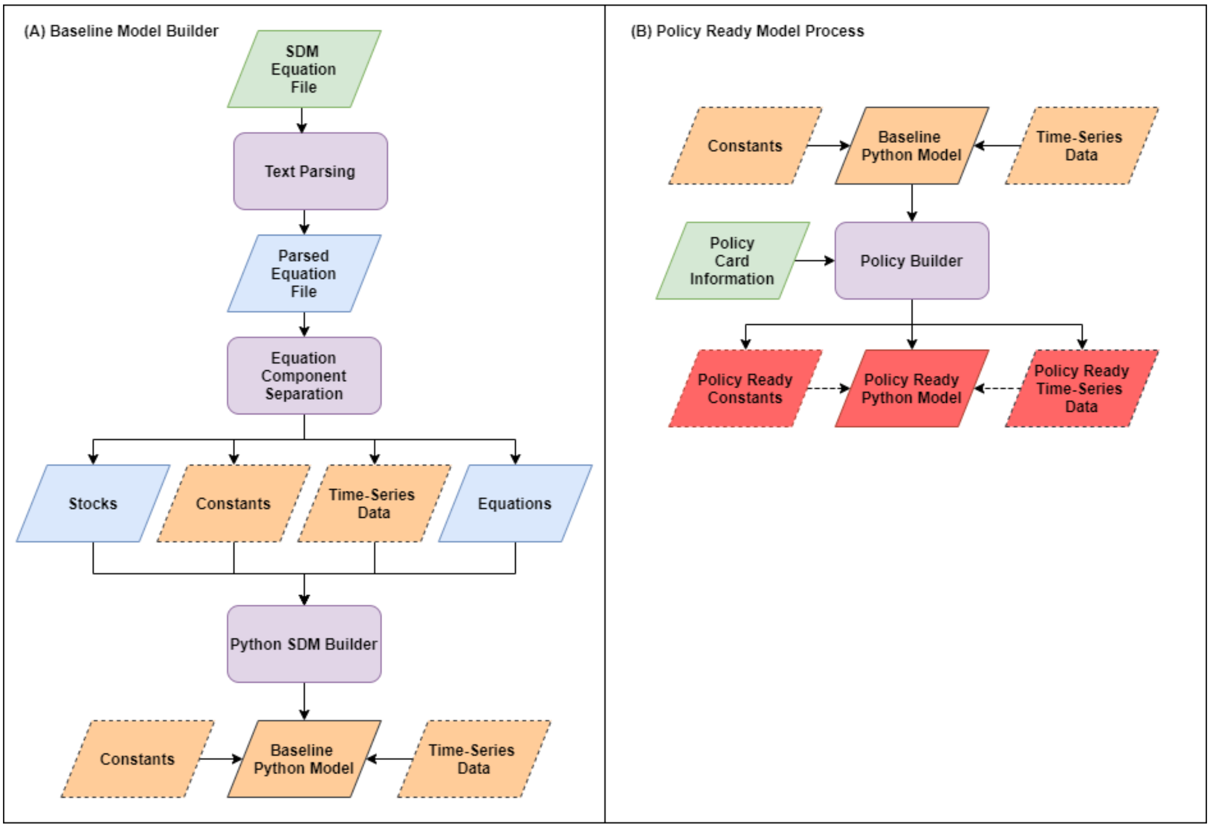


Figure 33. SDM conversion process flow chart

During a SG session, when the Game is initialized, the initialization step is executed and the initial stocks and policies are loaded to simulate (through the SDM Engine) the year 2020 (game starting point). During the rest of the SG session, the simulation step is used to compute the following turns, where the previous game status and the current user's decisions (selected policies) are sent through the UI request to the KEE and the SDM Engine uses them as inputs to obtain the next turn values, which are sent back to the UI to be presented to the player.

3.2.4 Data Access Module

The Serious Game data are persisted in the S4N databases, defined and implemented under T4.3 'Setting-up the project database and metadata ontology', which are deployed along with the KEE in the S4N servers.

There are two kinds of databases which store the data based on their type, source and utility.

All that information that has to be public and open data is available in the Semantic Repository (SR), and can be accessed via the KEE, the Semantic Repository API or the Naming Convention Tool. For example, the Case Study information, Learning Goals, Policy Cards and Policy Goals are stored in the SR.

On the other hand, other data that don't fill these requirements is persisted in a relational database. The Data Access Module acts as a bridge between the KEE and the S4N databases to simplify and isolate the data exchange between them. Apart from that, the Data Access Module is initially used to fill the databases with the raw information provided by the different Case Studies.

Unified Modelling Language Diagram

The following UML diagram (Figure 34) will introduce how the data models work referring to the Sim4Nexus Relational Database.

Starting from the core class of the diagram we get the User class. That class enables the action among the rest of the database models.

Regarding the first relation, Users can interact with questionnaires not only because they provide information about the Questions included among their sections but also because these Questions contain certain Options. These elements will help the User to perform an Answer related to this Question. The generation of these Answers will be through a formulary. Once this formulary is sent or skipped, the User will be inside the general SeriousGame flux.

Introducing SeriousGame flux, the action starts once the User has selected the game region and the game mode (MainPage and GamePage, respectively). At this point, the User is going to fall into the creation of a new Game instance. Regarding this instance, exists and is ready, Users start interacting with Policies as they think it would be the best for their region state. The selection of Policies leaf into UserDecisions over the Game timeline. Every single one of these UserDecisions interacts with the general Game progression by generating a GameStep. The accumulation of GameSteps will carry the game into their ending state (the year 2050).

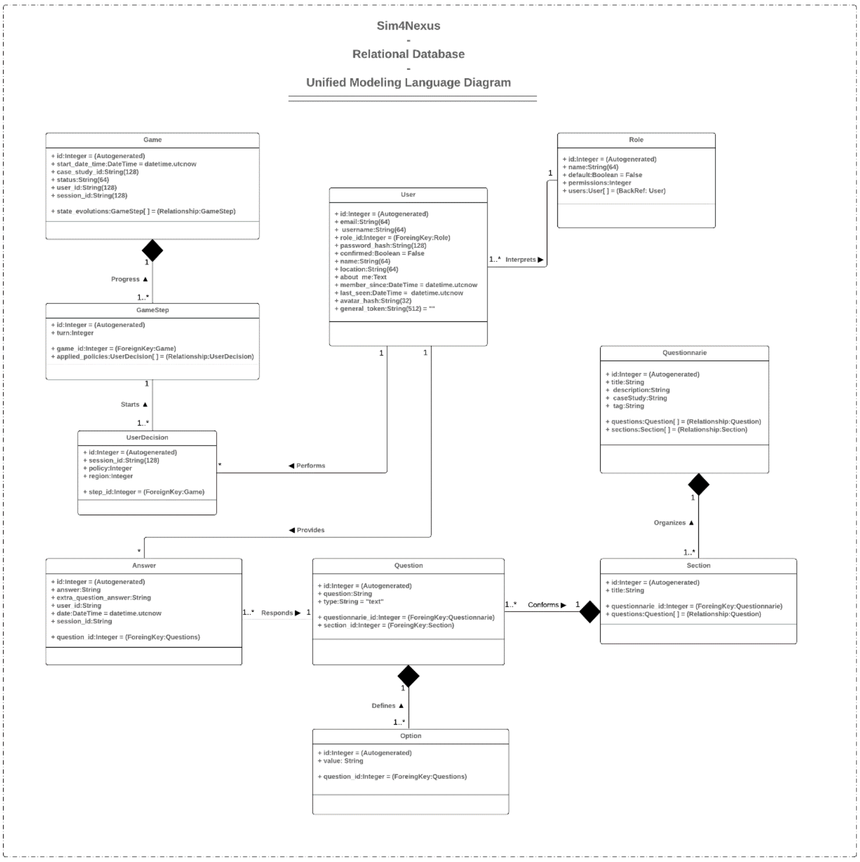


Figure 34: Sim4Nexus Relational Database Unified Modelling Language diagram (UML)

Representation regarding the Semantic Repository was previously defined in the Deliverable 4.4 Semantic Repository. As well as it was said on this deliverable, that structure can be iterable along within project progression.

Database models correlation

Figure 35: Database general routes analysis.

Method	Route	DB Model implicated	Data Source
POST	/kee/register	User	Relational DB
GET	/kee/confirm/<token>	User (confirmed)	Relational DB
GET	/kee/login	User (session)	Relational DB
GET	/kee/logout	User (session)	Relational DB
POST	/kee/changeEmail	User (email)	Relational DB
GET	/kee/questions	Question	Relational DB
POST	/kee/answers	Answers	Relational DB
GET/POST	/kee/policy_cards	PolicyCard	Semantic Repository
GET/POST	/kee/policy_goals	PolicyGoal	Semantic Repository
GET/POST	/kee/policy_objectives	PolicyObjective	Semantic Repository
GET/POST	/kee/case_studies	CaseStudy	Semantic Repository
GET/POST	/kee/learning_goals	LearningGoal	Semantic Repository

3.2.5 Agent-based Modelling

This is a reduced version because these results have been/will be submitted for publication in a peer reviewed journal. For more information please contact the authors - Lluís Echeverria Rovira (Eurecat) lluis.echeverria@eurecat.org

In order to simulate artificial players that interact with human ones and to run agent-based simulations of possible consequences of policy scenarios that directly or indirectly influence actors' behaviours a set of intelligent software agents is implemented based on Artificial Intelligence technics.

The key technology is Reinforcement Learning (RL), a specific area of Machine Learning (ML). RL is a computational approach to understanding and automating goal-directed learning and decision making. It is distinguished from other computational approaches by its emphasis on learning by an agent from direct interaction with its environment, without requiring exemplary supervision or complete models of the environment. The environment provides the learner (the agent) the needed training information: the global system status and the reward, the information on how well the system performed in the respective turn. Based on that, the agent has to uncover which actions generate the best results by trying instead of being told.

The main differences between RL and Supervised Learning algorithms are that in RL:

- There is no supervisor, only reward signals.
- Feedback is (usually) delayed, not instantaneous.
- Time matters (data is sequential, non-independent identical distributed data).
- Agent actions affect the subsequent data it receives.

Reinforcement Learning

In RL, the primary goal of the agent is to learn a method to choose its actions based on its current state, i.e., to learn a policy mapping states to actions, which optimizes a performance metric such as the expected average reward received per time-step[i].

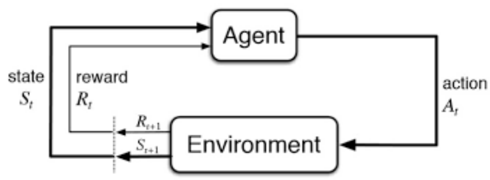


Figure 36. Reinforcement Learning schema

RL is built upon the concept of Markov Decision Process (MDP), a sequential decision-making problem based, defined by a 5-tuple: A set of states and actions (S, A), a reward model R, state transition probability matrix P (from all states s to all their successor s') and discounted factor $\gamma \in [0,1]$, which allows giving more importance to recent rewards compared to future rewards. A discount factor of 0 only considers current rewards and a discount factor of 1 considers all rewards in time with the same weight.

The reward R is a scalar feedback signal and the agent's job is to maximise the expected cumulative reward G. Formally, the cumulative reward is the discounted sum of reward accumulated thorough an episode of T steps.

Finally, the Policy π: is the agent's behaviour function, mapping an action to a given state. The final goal of the RL agent is to learn the most optimal policy to operate through the environment while maximizing the long-term reward.

Deep Reinforcement Learning

Traditional Reinforcement Learning uses a lookup table to store states and actions, which is too slow, since it learns the value of each state individually, and it is memory consuming, especially when it deals with a large number of states. The solution is to estimate value function, instead of using a lookup table, by using differentiable function approximations for the value function, trained using supervised learning algorithms. Given enough such samples of V, machine learning techniques can learn an approximation of V, say a linear combination of state-space features, giving a cheaper way to conduct policy iteration approximately in large state spaces[ii].

Deep Reinforcement Learning (DRL) is the result of combining deep neural networks (DNN) together with Reinforcement Learning, which make possible for RL not only act but to be totally autonomous and learn to see and act. For an n-dimensional state space and an action space containing m actions, the neural network is a function from R^n to R^m.

There are three main types of DRL algorithms, as in RL:

- Value optimization: the algorithm optimizes the value function V or Q, or the advantage function A.
- Policy optimization: the algorithm optimizes the policy directly function representing the neural network.
- Actor-critic: incorporates the advantages of each of the above, by learning value function with implicit policy:
    - Policy gradient component "Actor", which calculates policy gradients.
    - Value function component "Critic" that observes the performance of the actor and decides when the policy needs to be updated and which action should be preferred.

S4N ABM

Since the SG has an initial state (at the year 2020) and a set of final/terminal states (corresponding to the year 2050), which will be reached based on the policies (not to be confused with RL Policy) applied during the SG session, the environment is considered to be episodic.

As the S4N SDMs are deterministic, given an input (a combination of stocks and policies) they will always return the same output stocks, the state space of an RL agent can be obtained from all the possible sub-set of policies through the six turns of the S4N SG (from 2020 till 2050 with a 5-year timestep).

The state, the information sent by the environment to the Agent or, looked at another way, the information the Agent has to define its own representation of the environment, is defined by the SDM stocks and the current turn.

The Markov Decision Process can be directly extracted, since the process is stationary and the transition probabilities between states are deterministic (due to the deterministic SDMs previously described).

Finally, the Reward function has been specifically designed to allow the RL agent the ability to learn how to maximize the Nexus Health at the end of the session (year 2050).

Implementation and development of the S4N ABM

The library TF-Agents[4], from the TensorFlow[5] framework, provides an implementation of the DQN algorithm based on "Human-level control through deep reinforcement learning", Mnih et al., 2015 (https://deepmind.com/research/dqn/).

This algorithm, used to develop the S4N ABM module, is represented by an Agent which interacts with the environment and learns from it. The training process is based on two main tasks: collect data from the environment and use the data collected to train the Agent's Neural Network. TF-Agents provides an implementation of a DQN Agent, that can be used in any environment which has a discrete action space. The Neural Network that is trained during the process is the core of the DQN Agent and it learns to predict the expected returns for all actions given an observation from the environment.

Once trained, the DRL agents are serialized and persisted in order to be used on-demand.

3.2.6 Inference Engine

The Inference Engine analyses the user decisions to obtain derived potential knowledge which can be used to study and understand users' behaviours. It takes into account all the previous game sessions to classify and improve the results of the analysis.

[4] https://github.com/tensorflow/agents
[5] https://www.tensorflow.org/

The current S4N SG platform doesn't ask for specific user information (nationality, gender, age, …) when a new player is registered in the system. If this functionality is implemented in future versions, the Inference Engine system can make use of this detailed information to segment the users, study their behaviour based on this segmentation and finally provide more customized recommendations through the decision support system.

Different Statistic and Machine Learning data-driven algorithms are used to generate this inferred knowledge through the implementation of intelligent models. Since the used methodologies are based on historical data (user decisions) to train the models, first it is necessary to arrange enough information and, once enough information is available, due to the users playing the game, the models will be finally trained. From this point on, models can be automatically and periodically trained to include recent user sessions (new user's decisions) to continually improve their knowledge.

The Inference Engine is connected to the Analytical Engine, which provides the raw data directly extracted from the S4N databases.

3.2.7 Decision Support System

The Decision Support System gives advice, at each time step, to users while taking their decisions. To define this advice, there are two sources of knowledge which are combined to help the user:

- ABM: the current status of the Game is processed by the ABM to obtain the optimal actions to achieve the best long term goals.
- Inference Engine: previous decisions from other users (that faced a similar scenario) are taken into account to suggest new actions.

The advice corresponds to a specific policies combination. From the ABM system, the optimal policies (those policies that, base on the current state of the game, will lead on the maximal Nexus Health at the year 2050) are identified. In parallel, the Inference Engine is also consulted to identify which are the most common policies applied by other users in similar situations.

Once identified, these policies are sent to the UI through the request response and the corresponding policy cards are highlighted in the Policy Cards panel.

The ABM engine is a critical area of research for Eurecat and new scenarios will be supported soon (e.g. the NL CS)

4. Enhancements over the Semantic Repository

This section is mainly devoted to the description of the latest version of the ontology (Section 4.1) and the semantic repository (Section 4.2) to reflect and document the latest changes performed since the delivery of the D4.4 in August 2019. During this period, the main improvements has been performed to link policies and goals aligned with the evolvement of the game interface and knowledge elicitation engine.

4.1 Nexus Ontology

The nexus ontology is focused on the linkage between the nexus variables and instruments with specific policies and goals. The elaborated ontology supposes an advancement in **the interrelation of the nexus variables under a common instrument (policy) that affects case-study variables with a corresponding impact (indicator). This interrelation between cross-domain variables serves to demonstrate the affection of nexus in policy at long-term situations**.

The final version of the ontology is graphically represented in the Figure 40. Considering the enhancements, this version of the ontology has been focused on two main objectives: (i) align the nexus ontology with the structure of SAREF4WATR to provide an extension of this ontology towards the nexus and their indicators; and (ii) reinforce the interrelation between the defined case-studies and the nexus variables and instruments through the policy-making procedures.

Considering the alignment of the ontology with SAREF and SAREF4WATR (Figure 37), the main work performed has been focused on the re-design and adaptation of the "Measurement" part of the nexus ontology. In this regard, the current version of the ontology contains the reference to SAREF in the concepts of "Property" (general types of measures), "Measurements" (specific values) and geospatial elements (location of the measurements). Over these values represented in grey colour, we extended the ontology with the specific features of interest of S4n like "Regions" or "Case-Study" and specific property types (economic, Food, LandUse, Climate and Energy). This property types will enclose a representation of all variables involved in the demo cases (e.g. CAPEX/OPEX, Land Recovery, Carbon production, etc.)

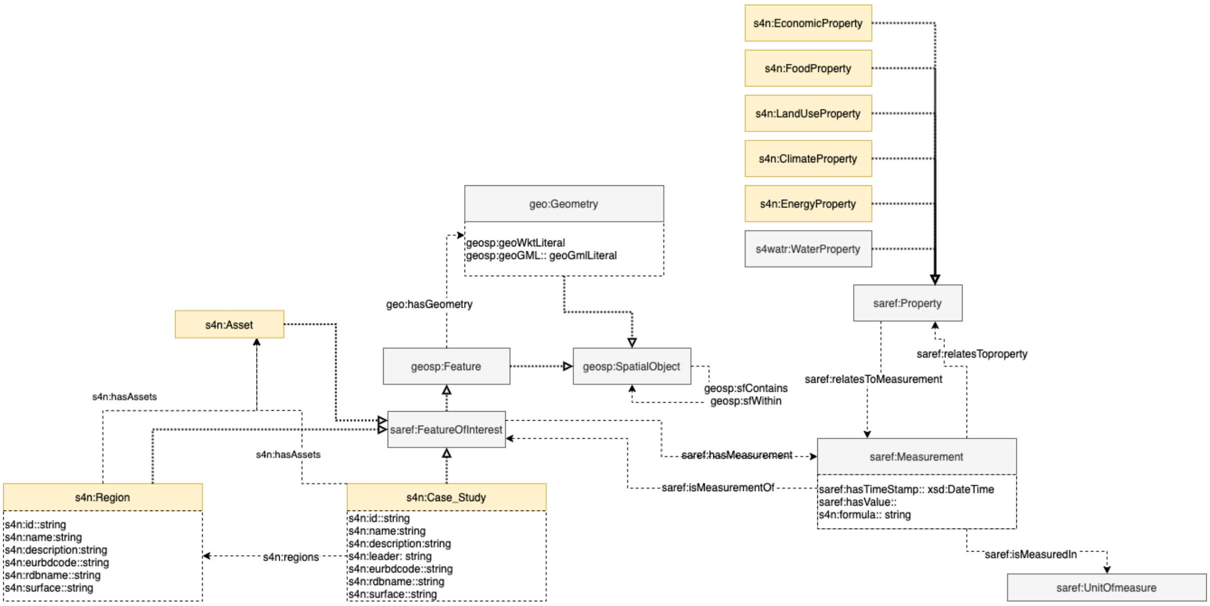


Figure 37. Alignment with SAREF4WATR

Moreover, another important improvement performed over the ontology relies in the interrelation between the case-studies and the corresponding goals and policies (Figure 38). In this case, each case study the case studies are formed by several regions where the actions are performed. Each of the case studies have defined learning goals, that means, the envisioned aspects to be learnt while playing the game. Moreover, the case studies also contain policies goals defined as the main states to be reached at the end of the game (e.g. Demand reduction in X%). To achieve and measure these goals, there are defined the "policy objectives" that are being represented as a key performance indicators at cross-domain level. To achieve the policy objectives, several instruments (Policies) are implemented in the case studies (e.g. "installation of next generation of smart meters").

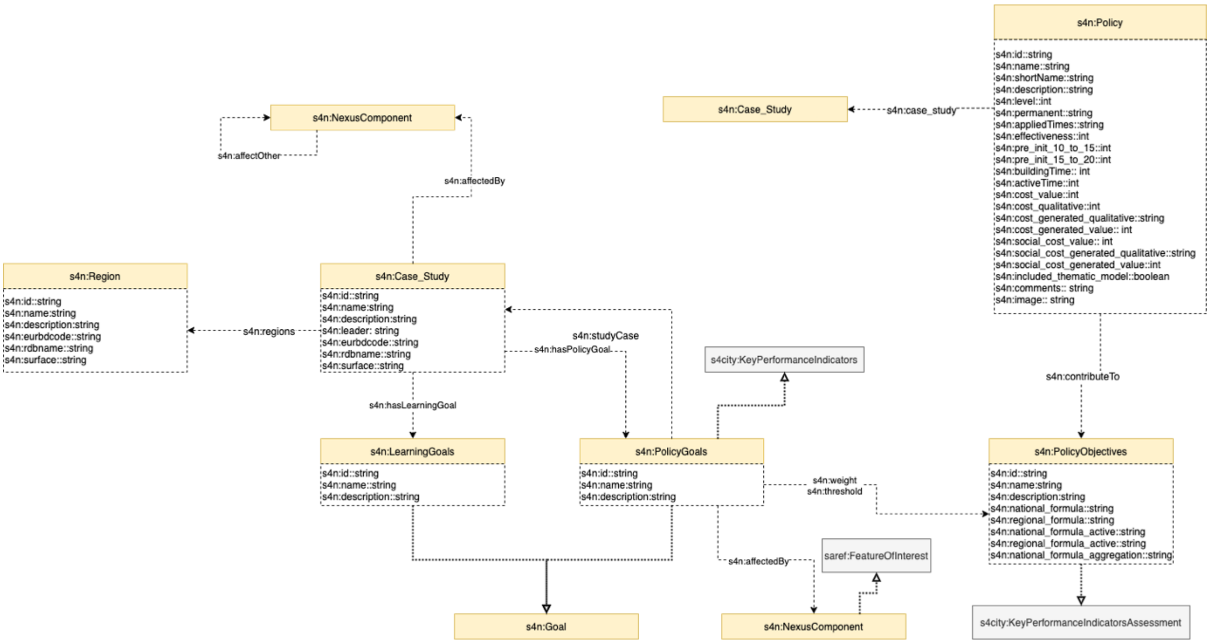


Figure 38. Interrelation between case-studies and policies

Considering these newer improvements over the ontology, we documented the ontology using WIDOCCO tool[6]. A publishable version of the ontology is deployed under the following link (Figure 39): http://seriousgame.sim4nexus.eu/ontology

[6] WiDOCO webpage: https://zenodo.org/record/2576182#.XV6ORy2B1TY

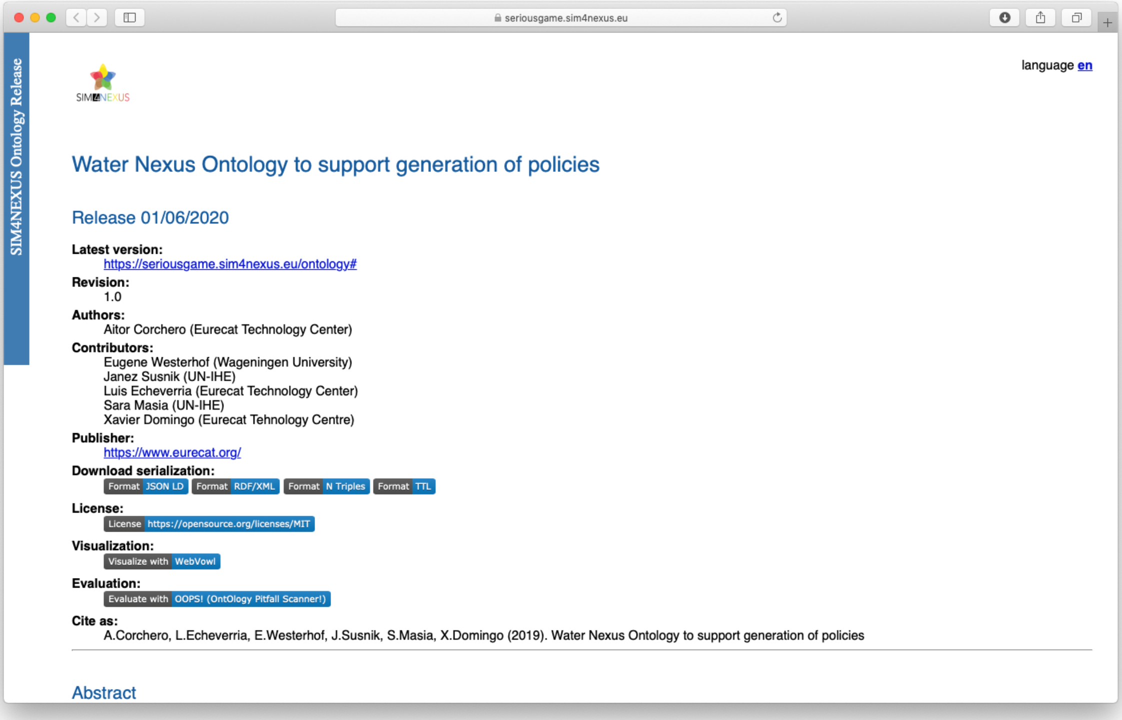


Figure 39. Nexus Ontology in the online version

As a conclusion of this section, we have concluded the development lifecycle of the ontology and it will be maintained by EUT beyond the SIM4NEXUS execution in order to contribute to the semantic interoperability of the nexus variables and their impacts at cross-domains.

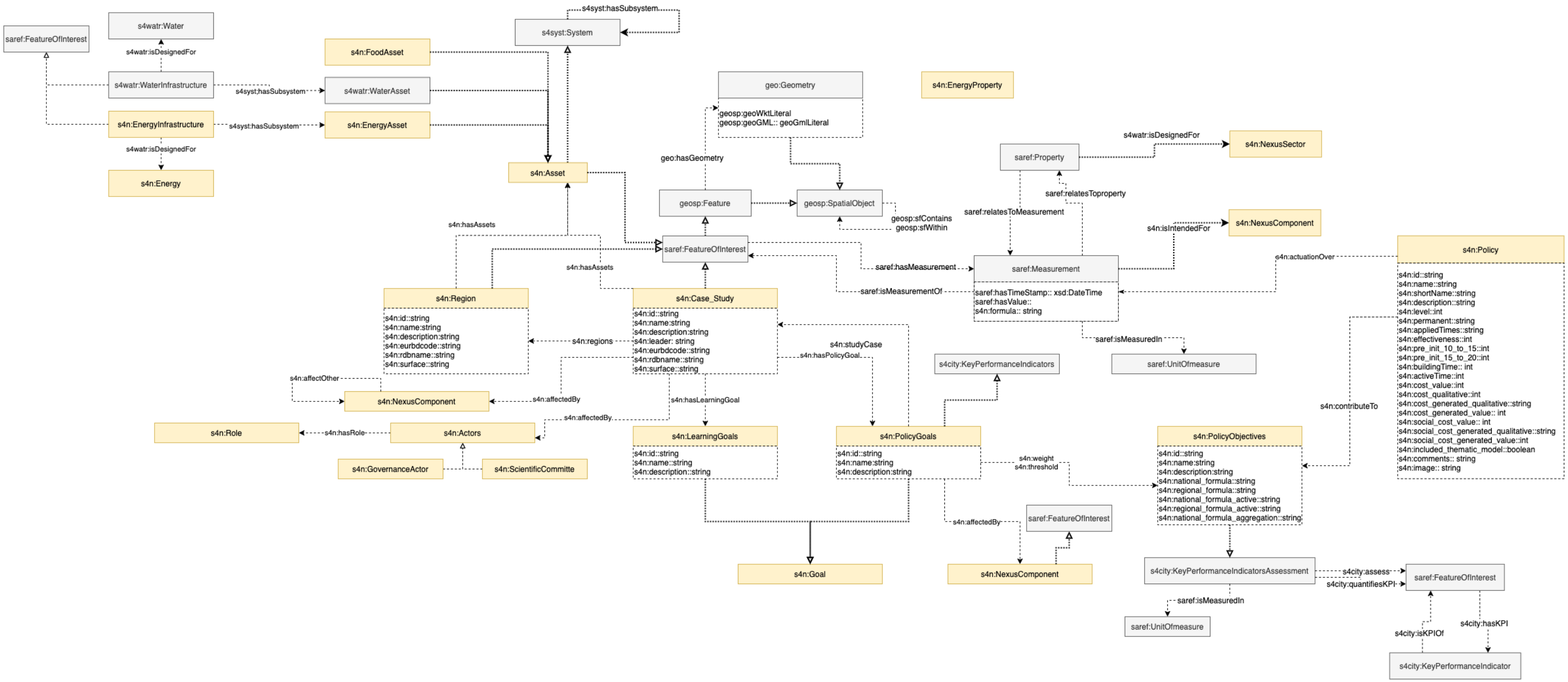


Figure 40. Overview of the Nexus Ontology

4.2 Semantic Repository

In relation to the semantic Repository improvements, the main work performed within this time has been focused on updating the data models in relation to the improvements of the ontology. In these regards, we redefine the following data models into a JSONLD (see Table 15).

Table 15. Improved Data models inside the semantic repository

Data Model	Description
Nexus health	Data model related to the definition of the formulas to compute the policy goals.
Policy Card	Data model referring the description of a policy card
Policy Goal	Data model referring the description of a policy goal
Policy Objective	Data model referring the description of a policy objective
Learning Goal	Data models used to define the learning goals.
Region	Definition of the geospatial information about a region.

The main work has been focused on defining "rules" to automatically convert JSON files (as receive the KEE) into the corresponding JSONLD documents (linked data graphs) to store correctly the information inside the semantic repository. For the elaboration of this mapping, we defined and published several context files. This context files (see Table 16) contains the needed information to transform the information from the incoming JSON into the ontology format:

Table 16. Context transformation file

```
{
 "@context":{
   "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
   "xsd": "http://www.w3.org/2001/XMLSchema#",
   "s4n": "http://seriousgame.sim4nexus.eu/ontology#",
   "rioter": "http://w3id.org/rioter#",
   "geo": "https://purl.org/geojson/vocab#",
   "s4watr": "https://saref.etsi.org/saref4watr#",
   "@version": 1.1,
   "@vocab": "http://seriousgame.sim4nexus.eu/ontology#",
   "@base": "http://seriousgame.sim4nexus.eu/ontology#",
   "id": "@id",
   "type": "@type",
   "lang": {
     "@type": "xsd:string"
   },
   "name": {
     "@id": "s4n:name",
     "@container": "@language"
   },
   "leader": {
     "@type": "xsd:string",
     "@container": "@set"
   },
```

```
    "summary": {
      "@id": "s4n:description",
      "@type": "xsd:string",
      "@container": "@set"
    }
 [...]
```

For the construction of the context files, we used the JSONLD playground[7] in order to test the correctness of the conversions. Moreover, the context files have been make openly available for their reference under the following links:

https://github.com/aolite/nexus_context/blob/master/geoContext.jsonld

https://github.com/aolite/nexus_context/blob/master/s4nContext.jsonld

The “geoContext” mainly refers to the conversion rules of the geospatial information into GeoJSON. Complementary, the “s4nContext” refers to the conversion rules for the serious games, the instruments and the corresponding interlinks between case studies and nexus variables.

As a conclusion of this section, the improvements performed in the elaboration of the context will serve to make consistent the incoming data-models inside the semantic repository. A part of this, the generic rules presented will serve to make agnostic the conversion between JSON and JSONLD and then, make more interoperable the interrelation between the different S4N systems.

[7] JSONLD playground: https://json-ld.org/playground/

Conclusions

Following the planning depicted in the Grant Agreement, the current version of the Serious Game tool implements all the defined modules (GUI, KEE, S4N Database and SDM Engine) and provides all the requirements needed to play the SIM4NEXUS Game and achieve one of the main SIM4NEXUS project goals.

In addition, five Case Studies have been already fully integrated, Greece, Azerbaijan, Latvia, the Netherlands and the southwest of the UK, thus making possible the interaction with them through the GUI and the final test to validate the expected and correct behaviour of each one. The inclusion of new Case Studies is simple and no new developments should be necessary (only in the case that new CS need new policy behaviours).

In parallel, the Global Case Study has developed a demo tool which can be accessed through the S4N SG platform.

As proof, the latest version of the Serious Game GUI and the underlying connected KEE, S4N database and SDM Engine are available and free to play at this URL: https://seriousgame.sim4nexus.eu/.

References

[i] Sutton, R. S., & Barto, A. G. Reinforcement learning: An introduction. MIT press, 2018
[ii] Roy, N., & How, J. P. A Tutorial on Linear Function Approximators for Dynamic Programming and Reinforcement Learning, 2013.