



HAL
open science

LittoSIM-GEN: A generic platform of coastal flooding management for participatory simulation

Ahmed Laatabi, Nicolas Becu, Nicolas Marilleau, Marion Amalric, Cécilia Pignon-Mussaud, Brice Anselme, Elise Beck, Xavier Bertin, Amélie Monfort, Clarisse Hayoun, et al.

► To cite this version:

Ahmed Laatabi, Nicolas Becu, Nicolas Marilleau, Marion Amalric, Cécilia Pignon-Mussaud, et al.. LittoSIM-GEN: A generic platform of coastal flooding management for participatory simulation. *Environmental Modelling and Software*, 2022, 149, pp.105319. 10.1016/j.envsoft.2022.105319 . hal-03640456

HAL Id: hal-03640456

<https://hal.science/hal-03640456v1>

Submitted on 18 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Highlights

LittoSIM-GEN: a generic platform of coastal flooding management for participatory simulation

Ahmed Laatabi, Nicolas Becu, Nicolas Marilleau, Marion Amalric, Cécilia Pignon-Mussaud, Brice Anselme, Elise Beck, Xavier Bertin, Amélie Monfort, Clarisse Hayoun, Frédéric Rousseaux

- We describe LittoSIM-GEN, a generic model and platform for conducting workshops on flooding risk management.
- We discuss the genericity problem in participatory simulation models.

LittoSIM-GEN: a generic platform of coastal flooding management for participatory simulation

Ahmed Laatabi^a, Nicolas Becu^a, Nicolas Marilleau^b, Marion Amalric^c,
Cécilia Pignon-Mussaud^a, Brice Anselme^d, Elise Beck^e, Xavier Bertin^a, Amélie Monfort^a,
Clarisse Hayoun^a and Frédéric Rousseaux^{a,*}

^aUMR 7266 LIENSs - CNRS, 2 rue Olympe de Gouges, 17000 La Rochelle, France

^bUMI 209 UMMISCO - IRD, 32 Av. Henri Varagnat, 93140 Bondy, France

^cUMR 7324 CITERES - CNRS, 33 Allée Ferdinand de Lesseps, 37204 Tours, France

^dUMR 8586 PRODIG - CNRS, 2 rue Valette, 75005 Paris, France

^eUMR 5194 PACTE - CNRS, 14 bis avenue Marie Reynoard, 38100 Grenoble, France

ARTICLE INFO

Keywords:

Participatory simulation
Agent-based model
Genericity
LittoSIM-GEN
Coastal flooding
Risk management

ABSTRACT

Coastal flooding is among the most significant disasters that threaten littoral populations in today's world. Climate change and global warming increase this risk and require implementing efficient strategies to anticipate the hazard and reduce human and material damage. This paper describes a generic platform called LittoSIM-GEN (a coupling of agent-based and hydrodynamic models), which aims to involve stakeholders in a participatory simulation of land use management to minimize the risks of coastal submersion. An application to the southwest coast of France (Rochefort) demonstrates the features of this platform. We propose several recommendations to improve the generic conception and implementation of models dedicated to participatory simulation.

1. Software and data availability

The LittoSIM-GEN model source code and data preparation scripts are available on GitHub (https://github.com/LittoSim/LittoSim_model/tree/LittoDev).

LittoSIM-GEN is developed by the LittoSIM consortium, a group of multidisciplinary researchers (<https://littosim.hypotheses.org/>).

LittoSIM-GEN is coded in GAML language under the Gama platform (<https://gama-platform.org/>).

Software and hardware requirements to run LittoSIM-GEN are detailed in the "*Implementation details*" section of this paper.

2. Introduction

Since the 1960s, the gaming simulation approach that combines system modeling and gaming interactions (Duke, 2014; Dukes et al., 2011; Stadskev, 1975) was used mainly for two purposes: to foster learning and improve awareness (learning games) (Sorace et al., 2018) or to support collective and deliberated decision-making processes (policy games) (Mayer et al., 2014). With the evolution of research and practices, complex systems modelers have developed the participatory modeling approach (Gray et al., 2016; Voinov and Bousquet, 2010). In parallel, video games have extended their application domain to the rapidly growing serious games since the 2000s (Michael and Chen, 2005; Sawyer and Rejeski, 2002). Participatory simulation (PS) is a gaming simulation approach that takes root more on the side of participatory modeling¹. It uses gaming and models to develop frameworks for groups of participants

*Corresponding author

✉ laatabi44@gmail.com (A. Laatabi); nicolas.becu@cnrs.fr (N. Becu); nicolas.marilleau@ird.fr (N. Marilleau); marion.amalric@univ-tours.fr (M. Amalric); cecilia.pignon-mussaud@univ-lr.fr (C. Pignon-Mussaud); brice.anselme@univ-paris1.fr (B. Anselme); elise.beck@univ-grenoble-alpes.fr (E. Beck); xavier.bertin@univ-lr.fr (X. Bertin); amelie.monfort@univ-lr.fr (A. Monfort); clarissehayoun@orange.fr (C. Hayoun); frederic.rousseau@univ-lr.fr (F. Rousseaux)

ORCID(s):

¹Some authors use the terms "participatory simulation" and "serious game" interchangeably, assuming that the serious game is multiplayer and serves a participatory purpose. In this paper, we use the first terminology.

that share control over decisions (Bakhanova et al., 2020; Becu, 2020). These participants collectively build ideas, abstractions (concepts), strategies, and plans that are useful for the game. PS is closely tied to the companion modeling approach (Étienne, 2013), as they share the same participatory and empowerment objectives (Becu and Crookall, 2020).

Multiple research works have investigated the use of PS among different environmental disciplines (Hedelin et al., 2021). Van der Wal et al. (2016) implemented a model of water management, where the game participants receive feedback depending on their chosen measures (embankment, changing land use, evacuation, etc.). Ornetsmüller et al. (2018) used an approach based on board games and cards to encourage farmers to change their cultivation strategies. Adam et al. (2018) propose a PS to examine the role of communication in reducing casualties during wildfires. Briot et al. (2017) developed a game for park managers to explore the impact of democratic practices and local governance on the biodiversity in protected areas. In flooding management, several applications are dedicated to various aspects of submersion risk. SPRITE (Taillandier and Adam, 2018) is a game designed for students to explore diverse policies of coastal flooding management. VigiFlood (Adam and Andonoff, 2019) is a game for crisis communication during floods. ANYCaRE (Terti et al., 2019) is a role-playing simulation that investigates how decision-makers anticipate and respond to weather-related events. Houry et al. (2018) propose a PS to understand flood prevention and improve resilience in mixed urban-rural environments. Tanwattana and Toyoda (2018) used gaming simulation as a learning tool to build community-based disaster risk management. Some online games are also available on the internet, such as STOP DISASTERS!² developed by the UN Office for Disaster Risk Reduction.

Participatory modeling and simulation, which involve both models and stakeholders, are becoming popular in environmental studies (Sterling et al., 2019; Van Bruggen et al., 2019; Gray et al., 2018; Le Page et al., 2013; Mayer, 2009) that use agent-based modeling to approach social, economic, and natural problems. This increasing use of participatory modeling and simulation has revealed many issues concerning their methods and results. Some gaps highlighted by researchers and related to complex models are lack of comprehension, dissemination, and replication (Laatabi et al., 2018). These limitations affect the genericity of PS models, as stressed in the concluding chapter of the Companion Modeling book (Étienne, 2013). Therefore, existing models and games are not usually reused or replicated to address similar questions in different contexts.

To investigate the issue of genericity in participatory modeling and simulation, we use the LittoSIM (Becu et al., 2017) kit³, a flooding risk management PS involving decision-makers. This toolkit is an agent-based model initially developed for the territory of Oléron Island in France. Thus, it faces considerable limitations since it is highly contextualized and hard to reuse in another context or even in the same context when the risk process or the field data are updated. Therefore, we edited LittoSIM-GEN (GENeric LittoSIM), an extended version to make the model generic, reusable, and adaptable to other territories. This work describes the architecture of the LittoSIM-GEN platform as an attempt to address the lack of genericity in PS models.

The paper is structured as follows: the following section presents the context of flooding risk management and its process in the French case and introduces the original version of the LittoSIM model. After that, we describe the new generic version of the model following the structure of a standard protocol. In the fourth section, we present an example of an application of LittoSIM-GEN to the estuary of Rochefort (France). Then, we discuss the changes made to generalize LittoSIM, and we draw up a set of recommendations to develop generic and reusable participatory simulations. The paper ends with a conclusion and some perspectives of this work.

3. Context and background: flooding risk management in France

Global warming and climate change increase the risk of natural disasters such as coastal flooding (Wong et al., 2014). Large waves and storms can lead to catastrophic losses in coastal areas (Paprotny et al., 2018), where populations are growing worldwide (Neumann et al., 2015). Low coastal zones become extremely dangerous for human activities, especially when the sea-level rise in concomitance with continental and pluvial flooding. Such scenarios amplify the risk and cause severe damage to exposed zones. Hence, decision-makers and urban planners must deal with this new socioeconomic challenge requiring them to mitigate the vulnerability and reinforce the resilience of their coastal cities.

In France, efforts to address the problem of coastal flooding risk are getting more attention, particularly after the Xynthia storm struck the Atlantic coast in 2010 (Bertin et al., 2014). The management of this risk is the responsibility

²<https://www.stopdisastersgame.org/>

³The kit is composed of: (i) an agent-based model coupled with a hydrodynamic model; (ii) a set of operating instructions to organize, conduct, and facilitate a workshop; and (iii) a list of equipment to run the workshop.

of local structures (formerly municipalities, district councils since 2018) that collaborate with the State services to elaborate plans and strategies to reduce flooding damage. The Risk Prevention Plan (PPR) is one of the tools used by municipalities to make decisions about land use and urbanization in areas exposed to natural hazards such as flooding. The PPR zoning is associated with the Local Urbanization Plan (PLU) to provide decision-makers with urban planning and management guidelines. However, depending on the territorial configuration, risk prevention may not be managed at the micro-scale and may require strategies that exceed the territory and the capacity of municipalities. The Inundation Prevention Action Program (PAPI) is defined to allow an integrated risk management program at the macro-scale (inter-district). This program gathers several municipalities to develop collective and more efficient plans that are difficult to elaborate on the micro-scale (district).

Flood damage may highly depend on individual choices and awareness of flood management (Dawson et al., 2011; Filatova et al., 2011). This culture concerns local authorities and decision-makers who manage strategic policies. Urban planners, for example, determine habitable and risk areas, build coastal defenses and encourage or discourage residential choices and preferences of coastal populations. Hence, developing a risk culture requires the implication of stakeholders in coastal flooding management. Training, teaching, and producing manuals are popular ways of learning and consolidating personal experience. However, modern technical developments allow more innovative and virtual methodologies to spread culture and improve awareness. Participatory simulations offer a simulated environment to put players in real-world situations (Bakhanova et al., 2020). In flooding risk management, these tools foster social learning about coastal flooding by experiencing different scenarios to reduce the potential damage of submersion events.

LittoSIM model (Becu et al., 2017) is one of these computer simulation tools dedicated to local authorities to learn new and alternative strategies by testing different scenarios of preventive measures before the submersion event. This toolkit focuses on participation and collaborative management of risk to rethink actual policies and overcome obstacles preventing the efficient handling of flooding problems. LittoSIM was developed for the territory of Oléron island in southwest France and experienced in several workshops between 2015 and 2018 with the community of Oléron districts (Amalric et al., 2017). The simulation environment was designed in collaboration with local stakeholders and technicians according to their insights about the flooding problem. Therefore, LittoSIM includes a set of concepts, rules, and constraints that are specific to the original territory. These specifications do not apply to other contexts (e.g., the restrictions on building outside a defined urban ring, the prohibition of changing urban areas to agricultural areas, etc.). Besides, the model design includes some behaviors and data types that may be relevant only in Oléron (e.g., types of land-use units, types of coastal defenses, installing sand fences process, costs and delays of player actions, etc.). These concepts are implemented through complex modeling aspects and data structures that are not generic. Thus, understanding the model and adapting its parameters is challenging and makes LittoSIM extremely complicated to reuse in a different territory. Hence, generalizing LittoSIM requires revising its structure to externalize independent components, data inputs, and simulation parameters.

4. LittoSIM-GEN: a generic version of the LittoSIM model

In this section, we follow the structure of the ODD+2D protocol (Laatabi et al., 2018) to describe all relevant concepts and necessary components to understand the model architecture and reproduce its results in different contexts. For the sake of understanding, we introduce the *Theoretical and empirical background* element as a part of the *Overview* block since describing general concepts, theories, and empirical basis underlying the model is necessary to understand the following blocks of the description.

4.1. Overview

4.1.1. Purpose

LittoSIM-GEN is a coupling of an agent-based model and a hydrodynamic model and implemented as a gaming simulation of submersion risk management. Participants of a LittoSIM-GEN workshop, whom we call players, manage their territories and test alternative strategies on land use planning and coastal defenses to reduce flooding damage. The game offers the possibility to experience different realistic management situations (e.g., deciding to invest in habitat densification or protecting populations, anticipating work delays due to regulatory constraints, etc.). Multiple actions (e.g., building dikes, withdrawing people, adapting buildings, etc.) are experienced and combined to assess their efficiency for protecting coastal populations, activities, and assets. Hence, the principal aim of LittoSIM-GEN is to help stakeholders and urban planners (decision-makers) improve their risk culture and awareness by discovering new ways to optimize their decisions and collaborate with other stakeholders implied in the flooding risk question.

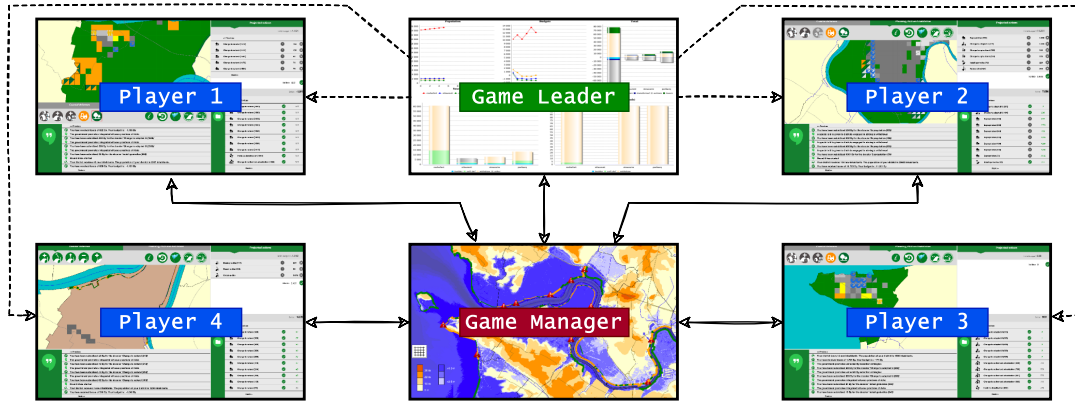


Figure 1: The architecture of the LittoSIM-GEN platform: A game leader (the risk agency) and four players (districts) interact in a shared space managed by the game manager. Solid and dashed lines symbolize bidirectional and unidirectional communications. Each role is implemented as a separate agent-based model.

The current version of LittoSIM-GEN proposes three territorial archetypes: overflow coast, estuary coast, and cliff coast. A territorial archetype is the typical example of a situation of inhabited space defined according to physical, socio-economic, historical, and governance characteristics (Rocle et al., 2020). New archetypes can be implemented in LittoSIM-GEN by including relevant data and updating parameter files.

4.1.2. Theoretical and empirical background

LittoSIM-GEN is composed of three agent-based models representing the three roles of the game (Figure 1). *Manager* is the model representing the study area and manages game rounds, launches submersion events, and communicates with the two other roles. *Player* represents decision-makers of the districts of the study area that administer their territories through actions on land-use units and coastal defenses. Players do not interact directly between them, and their number is limited to four in each game for fluency reasons. *Leader* represents the role of a hypothetical Risk Agency⁴ that provides policy advice to players and manages meetings between them to set out collective strategies. The Leader participates in the game through financial (giving and taking money) and information transactions (orientation, expertise). These interventions can alter the expected process of the game by changing conditions and constraints governing the application of player actions.

LittoSIM-GEN uses empirical data from the case study. These data are loaded into the model through several agents to implement different theoretical concepts that we introduce in the following paragraphs.

Districts. The study area may be composed of multiple districts (Figure 2b). Each district determines the territory that a player team can manage. A game is limited to four players.

Risk areas. The high-risk zones are more subject to submersion during flooding events (Figure 2h). The boundaries of these areas are determined by competent authorities and do not evolve during the game.

Protected areas. Natural and special protected areas (SPA) are forbidden from urban activities (Figure 2i). Non-urban projects (agriculture, managing coastal defenses) are subject to temporal and financial constraints due to the applied laws. The boundaries of these areas are determined by authorities and do not evolve during the game.

Soil cells. Soil cells are the micro-level of the spatial scale (usually $20 \times 20 \text{ m}^2$, or $10 \times 10 \text{ m}^2$ for high-resolution grids) in LittoSIM-GEN (Figure 3). The soil cells grid plays three distinct roles: the DEM (Digital Elevation Model) grid storing the soil altitude at each cell (Figure 2a), the rugosity (water resistance) grid saving the corresponding Manning value to the land cover of each cell (Figure 2j), and the flooding grid displaying within each cell the water height during

⁴The Risk Agency collaborates with districts and local authorities to encourage collaborative management and initiate collective strategies. The Agency funds a part of the proposed projects and supervises their implementation to allow their continuity under governance practices and alternative approaches despite financial and political conflicts between various stakeholders.

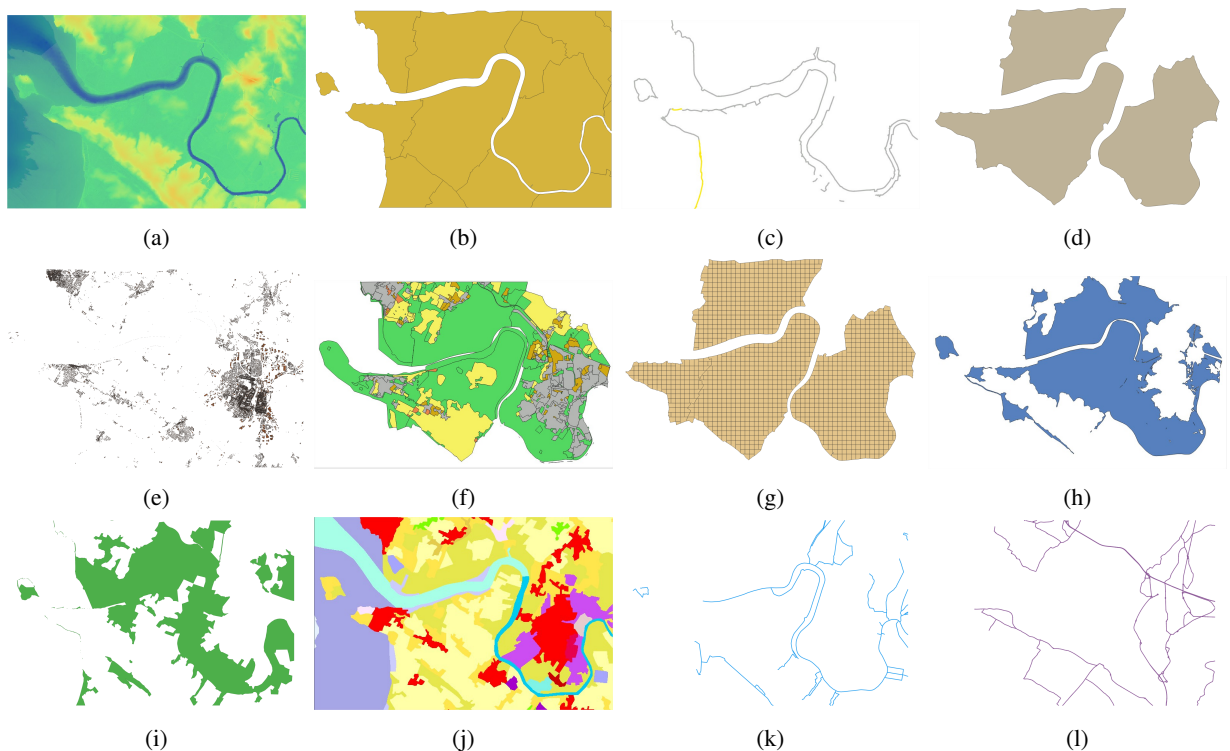


Figure 2: Areas and spatial components of the Rochefort version of LittoSIM-GEN: (a) digital elevation model (DEM); (b) districts; (c) coastal defenses; (d) inland dikes area; (e) buildings; (f) local urban planning (PLU); (g) land use grid; (h) risk areas (PPR); (i) natural protected areas; (j) land cover/rugosity; (k) rivers; (l) roads.

submersion events. The altitude and rugosity of soil cells are not changed directly by players but updated indirectly through actions on land-use units and coastal defenses (Figure 3).

Land-use units. A land-use unit is a principal element of planning management in LittoSIM-GEN (Figure 3). These macro-cells are represented by regular squares or by irregular polygons when located at the boundaries of districts (Figure 2g). According to the applied rules, players may change the type of a land-use unit (Figure 4). Six types are currently implemented:

Natural (N). This type corresponds to green zones like forests and meadows. Urbanizing in these areas may be allowed or prohibited when in protected areas. Non-protected natural units can become agricultural or authorized-for-urbanization.

Agricultural (A). These zones are open for human activities related to agriculture but not urbanizing. To build in these areas, they first need to become authorized-for-urbanization. Players can also transform them into natural areas.

Authorized-for-Urbanization (AU). This type corresponds to previously natural or agricultural areas open to urbanization. They can become urbanized (U) or adapted/special AU (AUs).

Authorized-for-Adapted-Urbanization (AUs). Authorities enhance infrastructures and subsidize flood-adapted buildings. This type is turned afterward to adapted-urban (Us).

Urban (U). This type contains residential buildings and can receive new populations. It may become natural through expropriation (N), densified to attract more population (Ui), or transformed to an adapted-urban (Us) area through the adaptation process.

Adapted-Urban (Us). It represents adapted urban areas with buildings that are more resistant to flooding risks.

Urban-in-Densification (Ui). This transition state occurs when authorities take measures to encourage populations to settle in these zones. Densification continues until the unit reaches the next level of population density.

LittoSIM-GEN: a generic platform of coastal flooding management

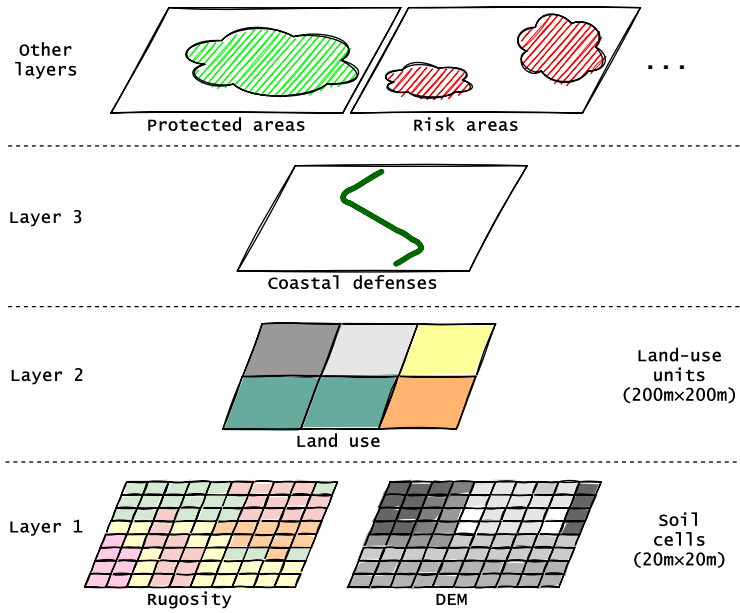


Figure 3: Players manage spatial objects in layers 2 and 3 (land-use units and coastal defenses). Layer 1 contains soil cells used to parameterize and display the submersion. Elements of the other layers visualize additional information that may affect player actions. Changing a land-use unit type adjusts the water-resistance of the intersecting soil cells (rugosity grid). Changing coastal defenses updates their altitude (DEM grid). Water heights of the submersion are displayed at the level of soil cells.

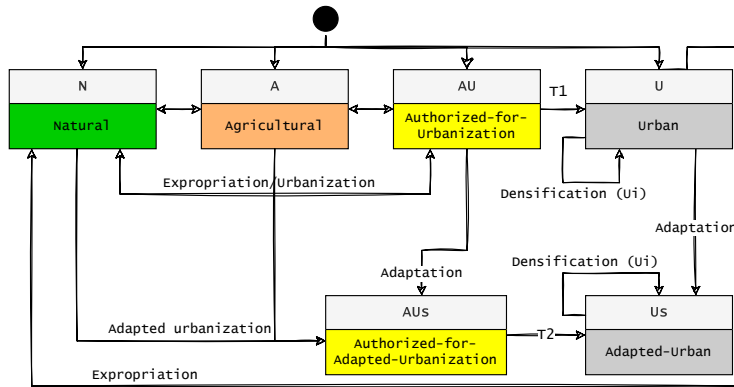


Figure 4: The life cycle of a land-use unit: player actions trigger all transitions except turning AU/AUs to U/Us that automatically occurs after a specified time T_x , and the transition state U_i that continues until the densified unit reaches the next population-density class.

Coastal defenses. Coastal defenses are spatial lines created to protect littoral areas from flooding when the sea level exceeds the land elevation after storms or large waves (Figure 2c). Building a new coastal defense increases the altitude of the soil by adding the height of the created object. Modifying this height (raising) affects the corresponding soil cells. Dismantling an object resets the soil altitude. A coastal defense has a state (good, medium, or degraded) that degrades in time but can be upgraded with a repair action, installing sand fences, or loading pebbles. During a flood event, a coastal defense has a probability of rupture that depends on its state. Figure 5 depicts the state diagram of coastal defenses in LittoSIM-GEN. State dynamics depend on whether the object is a dike, a dune, or a pebble dike.

Dike. A dike is a human-engineered barrier built with stones or other building materials. Dikes are littoral when installed directly on the coast or inland when constructed away from the coastline with a specified distance (inside

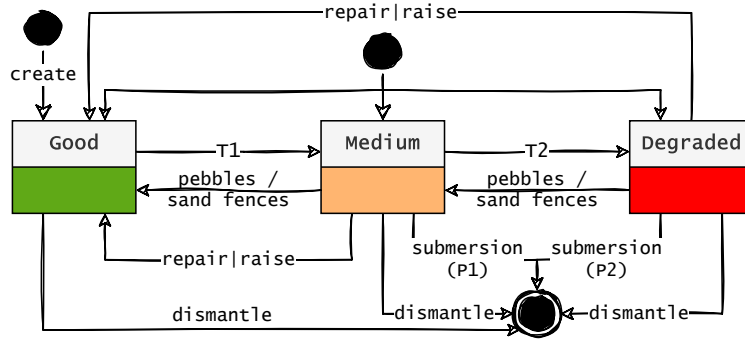


Figure 5: The life cycle of a coastal defense: created objects are always of a good state. Status degradation takes time (number of rounds) T_x . Status is upgraded by a player action depending on the object's type (raise, repair, installing sand fences, or loading pebbles). A submersion can breach an object with a probability P_x depending on its status (an object in a good state does not break). A player can dismantle his coastal defenses at any time.

the area of Figure 2d). Dikes are detrimental and non-natural structures, but inland dikes are less harmful to the environment.

Dune. A dune is a hill of sand constructed naturally or built by humans through, for example, installing sand fences. A sand fence is a wooden barrier made of slats that regenerates a dune around it. Dunes are natural barriers and preferred to dikes for ecological reasons.

Pebble dike. A pebble dike is a natural or human-engineered (but without any other building materials) dike of pebbles (small or large pieces of rocks). This barrier is also eco-friendly as dunes and is often used to reinforce existing solid dykes.

Player actions. Players can take several measures to reduce the submersion risk. These actions are of two categories depending on their target object: a land use (LU) or a coastal defense (CD) action. Each of these actions has a cost and a delay (number of years/rounds) before being applied and may belong to different strategies depending on its application conditions (see Table 1).

Strategic profiles. The principal aim of LittoSIM-GEN is to allow decision-makers to experience alternative strategies of flooding risk prevention. The experience with earlier workshops of LittoSIM (Becu et al., 2017) has revealed three different player profiles based on their actions:

Builder. Builders build coastal dikes and initiate urbanization and densification projects even in risk areas. The builder profile opts for solid coastal defenses to protect coastal populations and assets.

Soft defense. Players opt for more natural solutions to protect their coasts, such as building and maintaining dunes and creating inland dikes. For urbanization and densification projects, this profile favors risk-adapted areas.

Strategic withdrawal. This profile regroups retreat actions such as dismantling coastal defenses, expropriating risk urban areas, and urbanizing and densifying out of littoral and risk areas.

These three strategies classify a player that can be, for example, 50% builder, 40% soft defense, and 10% withdrawal based on his actions. Associating a player to a strategy depends on a specified threshold (e.g., a player has a builder profile when he is at least 40% builder). The classification of some actions within a strategy depends on the player profile (e.g., classifying urbanization out of littoral and risk areas as a strategic withdrawal requires that the player already has a withdrawal profile). This dynamic profiling anticipates player strategies. An action may be categorized as "Other" if it does belong to none of the three profiles (e.g., densification out of risk areas but without being neither withdrawal nor soft defense). Table 1 depicts the implemented actions in the current LittoSIM-GEN with their classification among different strategies. Each of these player actions may trigger one or several levers depending on a set of conditions and parameters.

Levers. A lever is a bonus or penalty intervention that rewards or penalizes a player based on his actions during the game. Different levers can be triggered after a certain threshold and applied (if not canceled) on player actions (Figure 6). There are two types of levers. Cost levers add (to penalize) or subtract (to reward) an amount to an action

Table 1

Player actions in LittoSIM-GEN: profiling an action in each strategy depends on the action itself, the area where it takes place, and the profile of the player. **CD.** coastal defense; **LU.** land use; **RA.** risk area; **LA.** littoral area; **OUT.** out of risk and littoral areas; **B.** builder; **S.** soft defense; **W.** strategic withdrawal.

Player action	Type	Strategy			Remarks
		RA	LA	OUT	
Create a dike	CD	B/S	B/S	B/S	Building a new littoral (B) or inland dike (S)
Raise a dike	CD	B/S	B/S	B/S	Increasing dike's height (B if littoral, S if inland)
Repair a dike	CD	B/S	B/S	B/S	Repairing turns the dike's status to "Good" (B if littoral, S if inland)
Dismantle a dike	CD	W	W	W	Dismantling a dike is always a strategic withdrawal
Load pebbles	CD	S	S	S	Reinforcing a pebble dike with new pebbles
Create a dune	CD	S	S	S	Creating a new dune elevates the corresponding cells altitude
Install sand fences	CD	S	S	S	Installing sand fences to regenerate dunes (raising their height)
Enhance accretion	CD	S	S	S	Enhancing the natural accretion also regenerates dunes
Maintain a dune	CD	S	S	S	Preventing the deterioration of dunes (keep the same status)
Change to AUs	LU	S	S	S	Changing a unit to Authorized-for-Adapted-Urbanization
Change to AU	LU	B	B	S/W	Changing to AU when being soft defense (S) or withdrawal (W)
Change to Us	LU	S	S	S	Changing an Urban unit (U) to Adapted-Urban (Us)
Change to Ui	LU	B	B	S/W	Densifying a U unit when being soft defense (S) or withdrawal (W)
Change N to A	LU	B	-	-	Changing a Natural unit to Agricultural
Change AU to A	LU	-	-	-	Changing an Authorized-for-Urbanization unit to Agricultural
Change U to N	LU	W	W	W	Expropriating an Urban unit
Change AU to N	LU	W	W	W	Expropriating an Authorized-for-Urbanization unit
Change A to N	LU	W	W	S	Changing an Agricultural unit to Natural

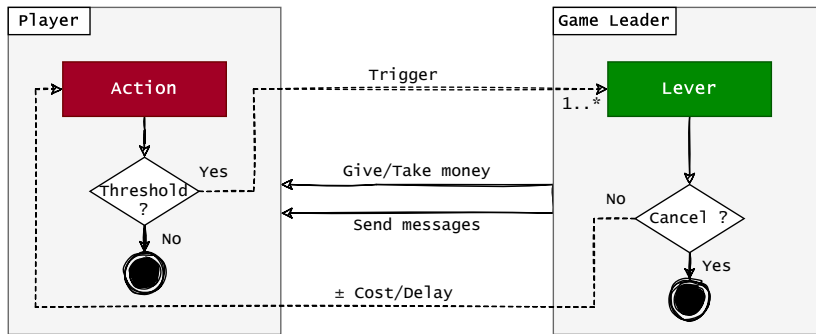


Figure 6: Several interactions are possible between the game leader and the players in LittoSIM-GEN. Besides information messages and direct financial transactions, levers allow reorienting player strategies. When an action exceeds a threshold, the corresponding automatic levers are triggered to reward or penalize the player. The leader might cancel some levers to balance the game.

cost. For example, constructing a new dike after exceeding a fixed threshold (penalty: the action cost is marked-up) or installing sand fences on a dune (bonus: the action cost is marked-down). Delay levers add or deduce a duration from an action delay. For example, creating an inland dike (bonus: the action delay is accelerated) or urbanizing a land-use unit in a coastal area (penalty: the action delay is retarded).

Levers are implemented in LittoSIM-GEN to encourage or discourage player actions depending on their ecological and virtuous strategies. These levers fit into the three previous strategic profiles: builder levers (B) concern dike building and urbanization actions; soft defense levers (S) are applied on dunes, inland dikes, and adapted urbanization; and withdrawal levers (W) concern actions of strategic withdrawing of urban populations and dismantling of dykes. Builder levers penalize, and soft defense and strategic withdrawal levers reward players. Table 2 lists implemented levers in the current LittoSIM-GEN.

Table 2

Lever in LittoSIM-GEN are triggered when their threshold is crossed and act by adding or removing an amount of the action cost or delay. **C.** cost; **D.** delay; **RA.** risk area; **LA.** littoral area; **B.** builder; **S.** soft defense; **W.** withdrawal.

Lever	Cate	Thres	Impact	Description
	gory	hold	C D	
Dike created	B	0.2	+0.25	Adds 25% of the action cost once new dikes exceed 20% of the initial length
Dike raised	B	0.2	+0.25	Adds 25% of the cost once raised dikes exceed 20% of the initial dikes' length
Dike repaired	B	0.2	+0.25	Adds 25% of the cost once repaired dikes exceed 20% of the initial length
AU or Ui in LA	B	2	+2	Adds 2 rounds to the action delay once the number of new actions exceeds 2
AU or Ui in RA	B	1	+0.5	Adds 50% of the action cost once the number of new actions exceeds 1
Dune created	S	0.1	-0.25	Takes 25% of the cost once new dunes exceed 10% of the initial dunes
Accretion enhanced	S	0.1	-0.25	Takes 25% of the cost once accreted dunes exceed 10% of the initial length
Sand fences installed	S	0.1	-0.25	Takes 25% of the cost once installed fences exceed 10% of the initial dunes
Dune maintained	S	0.1	-0.25	Takes 25% of the cost once maintained dunes exceed 10% of the initial length
Pebbles loaded	S		-0.5	Activated manually by the game leader and takes 50% of the action cost
Inland dike created	S	0.01	-1	Takes 1 round from the delay once new dikes exceed 1% of the initial length
Us out of LA & RA	S	2	-0.25	Takes 25% of the action cost once the number of new actions exceeds 2
Us in LA	S	2	-0.5	Takes 50% of the action cost once the number of new actions exceeds 2
Us in RA	S	2	-0.5	Takes 50% of the action cost once the number of new actions exceeds 2
No dike created	W	2	-0.5	After 2 rounds, it takes 50% from the cost of one action of type S or W
No dike raised	W	2	-0.5	After 2 rounds, it takes 50% from the cost of one action of type S or W
No dike repaired	W	2	-0.5	After 2 rounds, it takes 50% from the cost of one action of type S or W
A to N in LA or RA	W	2	-0.5	Takes 50% of the action cost once the number of new actions exceeds 2
Ui out of LA & RA	W	2	-0.25	Takes 25% of the action cost once the number of new actions exceeds 2
Expropriation	W	1	-0.25	Takes 25% of the action cost once the number of new actions exceeds 1
Dike dismantled	W	0.01	-0.5	Takes 50% of the cost once dismantled dikes exceed 1% of the initial length

4.1.3. Entities, state variables and scales

LittoSIM-GEN simulates the territory evolution over a set of rounds (usually 10). Each game round represents one year in the real world. The model is composed of three separate agent-based models representing the three roles of the game. We describe their conceptual structures separately. Each UML entity describes an agent.

Game Manager. Figure 7 depicts the structure of the game manager model as a class diagram with only relevant attributes. It is composed of three categories of classes. Six core entities representing principal agents allow managing coastal defenses and land-use units by applying received actions from players and activated levers from the game leader. Two network entities to communicate (send and receive data) with players and the game leader. Six display entities add additional information to the simulation, such as roads, rivers, natural protected areas, and flooding risk areas.

Game Leader. Three entities of the manager model (District, Player Action, and Activated Lever) are implemented here to represent information that the two models share. An activated lever is a Lever triggered by an action and accepted (not canceled) by the leader. According to its type, a lever may change the delay of an action (Delay Lever) or change its cost (Cost Lever). Several entities extend these two abstract classes to represent all levers introduced in Table 2. One network agent performs communications with both the game manager and players. Figure 8 depicts the UML class diagram of this model with relevant attributes of each entity.

Players. Player Action is extended to represent the two types of player actions (Coastal Defense Action and Land Use Action). Communications with the game manager are handled by Network Player and with the game leader by Network Listener To Leader. Additional classes from the game manager provide players with all the information required to play the game and optimize their actions. Three new classes help manage the game interface on the player side. Basket allows storing current player actions (name, cost, delay) with the possibility of canceling. History lists already validated player actions with their information (final cost after taking into consideration applied

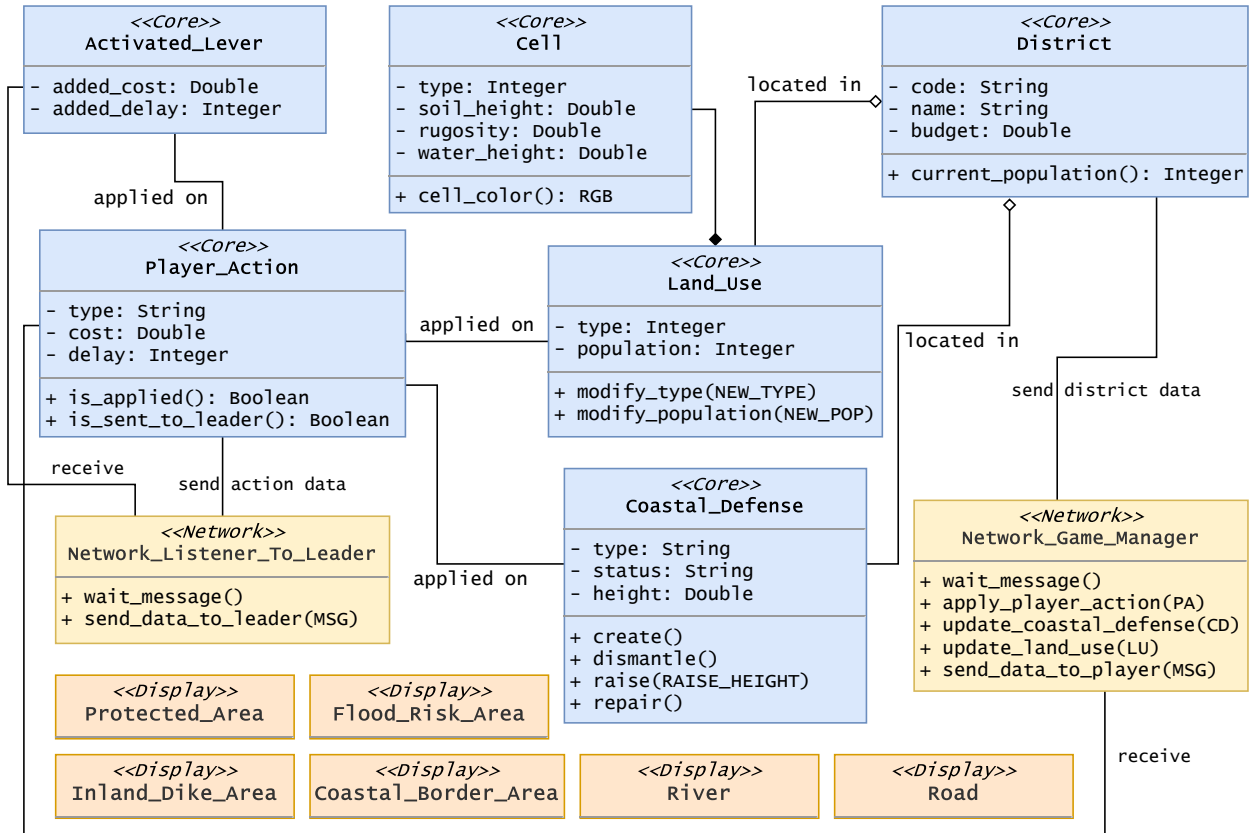


Figure 7: The class diagram of the game manager model: Cell represents soil cells (DEM and rugosity in Figures 2a and 2j); District (2b); Land_Use (2g); Coastal_Defense (2c); Protected_Area (2i); Flood_Risk_Area (2h); Coastal_Border_Area represents an area of 400m around the coastline to identify littoral areas; Inland_Dike_Area represents a 100m inside districts' area to identify inland dikes (2d); Road (2l) and River (2k) display roads and rivers of the study area. Player_Action represents an action on a land-use unit or a coastal defense. Activated_Lever represents levers triggered by player actions. Network agents allow exchanging information about the study area, player actions, and activated levers.

cost levers, remaining delay after considering applied delay levers). Message displays received information concerning the game (rounds, population, budget, messages from the leader), validated actions, and activated levers.

4.1.4. Process overview and scheduling

A LittoSIM-GEN game takes place over three phases. At first, the workshop animator explains game rules to the four playing teams. In the second phase, the game begins and players start to manage their territories in about ten rounds, during which three or four submersion events occur. The third phase is devoted to debriefing and discussing the results.

During the game, the game manager launches submersion events and controls different components of the game (rounds, player actions, and leader commands). Players (decision-makers representing the districts) execute their planning actions (on coastal defenses and land-use units) and submit them to the game manager. The game leader validates or cancels the triggered levers and exchanges information with players (financial transactions and text messages). The activity diagram in Figure 10 shows the three actors of LittoSIM-GEN with their different interactions and activities within the model.

4.2. Design concepts

4.2.1. Individual decision making

Levers are activated automatically based on player actions and activation parameters (Figure 6 and Table 2). Other agents do not implement any automatic decision-making. All decisions are made by humans implied in the game.

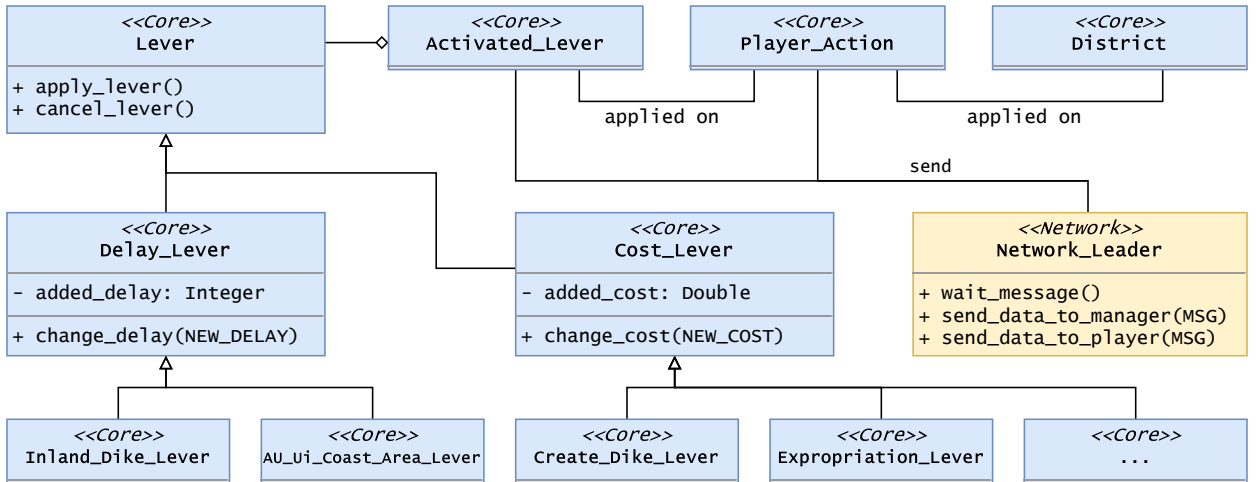


Figure 8: The class diagram of the game leader model: Levers of Table 2 are all implemented as agents that inherit delay or cost lever depending on their type. In the current LittoSIM-GEN, there are two delay levers ("AU or Ui in Littoral Area" and "Inland Dike creation") and 19 cost levers.

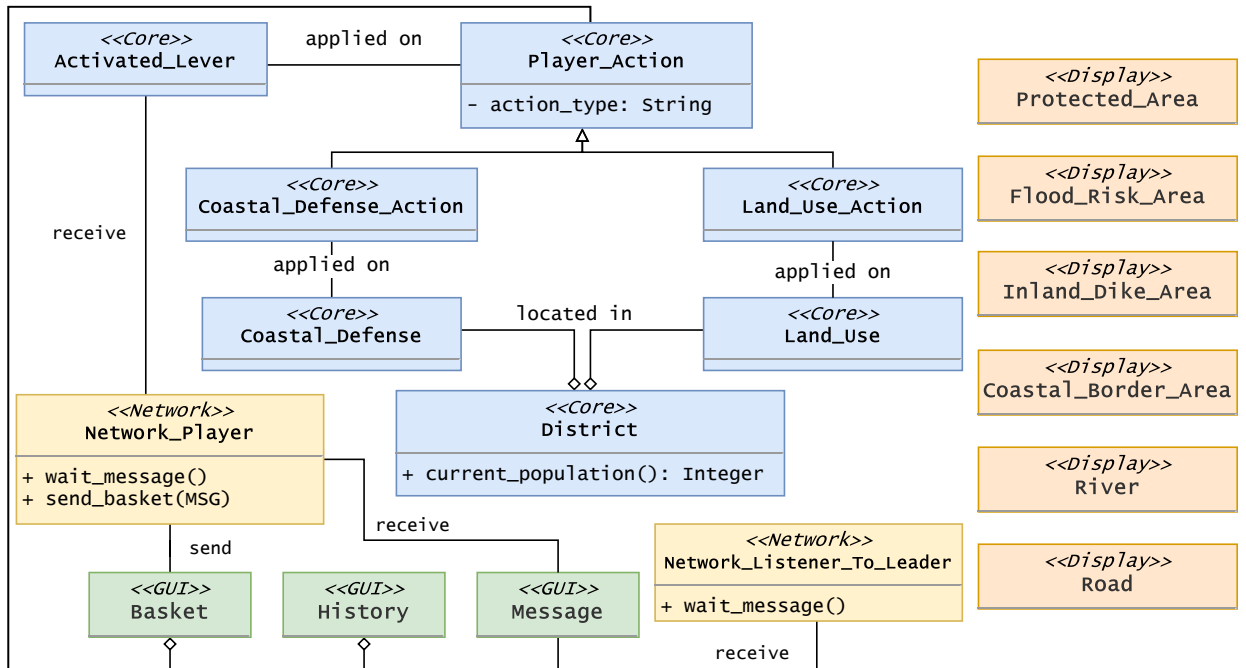


Figure 9: The class diagram of the player model: Player_Action represents the two types of actions. The Basket agent registers player actions that may be canceled or transmitted to the game manager later. History lists sent actions. The Message agent displays messages received from the game manager and the game leader.

Game Manager. Based on a predefined scenario, the game manager decides when to switch rounds and launch submersion events. Flooding events are planned according to the workshop time and the number of game rounds. This scheduling should not be told to the playing teams.

Game Leader. The risk agency has a set of indicators to help make proper decisions depending on the adopted strategies (builder, soft defense, strategic withdrawal, or others), the budget of districts, and the shared financial plan

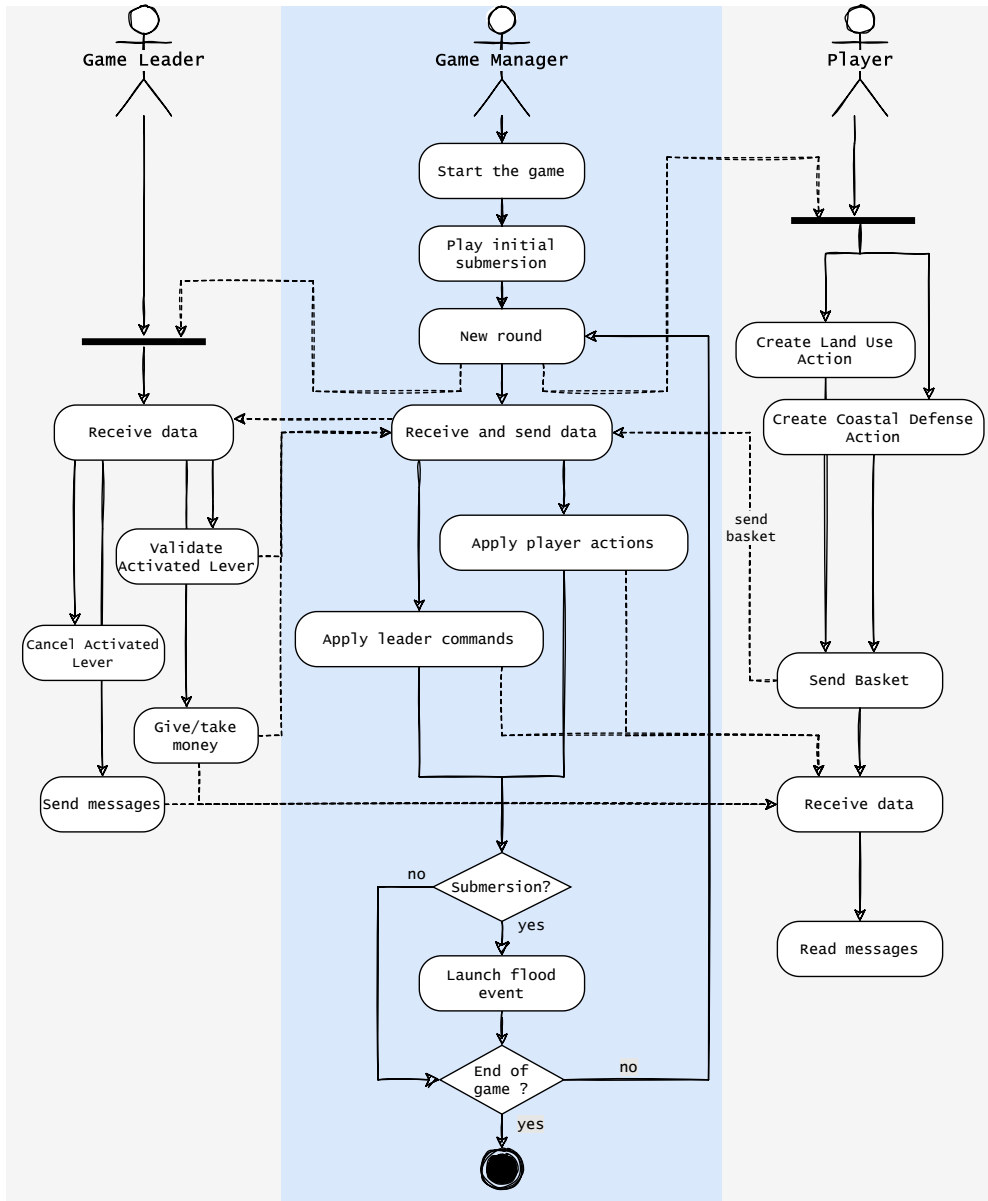


Figure 10: The activity diagram of the LittoSIM-GEN model uses the agents of previous class diagrams (Figures 7, 8, and 9). The game manager is the central actor playing the role of synchronization between the other actors. Players manage their territory, and the game leader supervises executed actions and takes proper decisions.

that players may want to set up. These decisions are critical and can alter the game process. Hence, the game leader must take the following actions with vigilance:

Validating/canceling levers. A player can take actions that are too penalized or rewarded by automatic levers. The game leader may intervene to correct this situation and apply or cancel some levers.

Giving/taking money. The game leader can give or take money from some players to balance their budgets, reorient their strategies, or reward or penalize them if the impact of levers is biased.

Sending messages. Information and warnings are sent to players to inform them about decisions concerning their budgets, populations, and strategies.

Players. Each playing team makes decisions individually or with the risk agency and the other players. Some decisions may concern many districts and must be subject to a collective discussion that leads to a shared financial plan. Recommendations of the risk agency may affect the decision-making process of players.

4.2.2. Learning

Model agents do not implement any form of automatic learning. To optimize their decisions, human agents, namely the game leader and the players, learn from their past experiences and the feedback of their actions. Discussions and face-to-face role-playing reinforce social and horizontal learning (Solinska-Nowak et al., 2018; Voinov et al., 2018).

4.2.3. Individual sensing

The game manager and the game leader have a global perception of the game and use several indicators to optimize their decisions. Each player team has a partial perception limited to its environment and has no access to the actions of the game manager (submersion events) and the game leader (levers and transactions). Players may get more information through collaboration exchanges (with the game leader and other players) and at the shared space while displaying a submersion event.

4.2.4. Individual prediction

The model does not implement any form of automatic prediction. Only the human intelligence of the game participants predicts future events and the consequences of current decisions.

4.2.5. Interaction

Interactions among model components follow the conceptual model and relationships between agents (Figures 7, 8, and 9). Network entities of each model allow to exchange data through remote communications (Figure 10). All player actions and leader commands (financial transactions and levers) pass through the game manager, except textual messages from leader to players. At each game round, the game manager updates the environment and sends back the current state to the game leader and players.

Players of different districts can collaborate and interact to develop collective strategies. Global interactions take place at the shared space when visualizing submersion results. Additional exchanges can also occur between players and the game leader. These interactions aim at reorienting strategies through advice, technical expertise, and initiating collaborations.

4.2.6. Collectives

Agents (land-use units and coastal defenses) of the same district form collectives and are affected by the same decisions (budget limits and applied levers). Players of the same district form a human group that deliberates to make decisions. Different playing teams may also set an inter-district collective while discussing collaborative projects and strategies.

4.2.7. Heterogeneity

Agents differ by their state variables. Each district has its geographical configuration, budget, and population that evolve differently. Game players are also heterogeneous and make decisions differently based on their objectives, perceptions, and personal experiences.

4.2.8. Stochasticity

Simulation parameters are read from predefined configuration files (see *Initialization* block under the *Details* section). Only three parameters are randomly attributed:

- Each coastal defense object (dike or dune) has an initial number of game rounds before updating its state. This counter parameter is randomly initialized between 0 and the maximum number of rounds to change the state (STEPS_DEGRADE_STATUS_DIKE or STEPS_DEGRADE_STATUS_DUNE).
- Each coastal defense object has a probability of breach (rupture) that is assessed randomly and according to the predefined values for each state.
- After each flooding event, the game manager sends a random number (between 1 and 5) of submerged land-use units to each district as flood marks. These landmarks give partial information about the extent of the submersion.

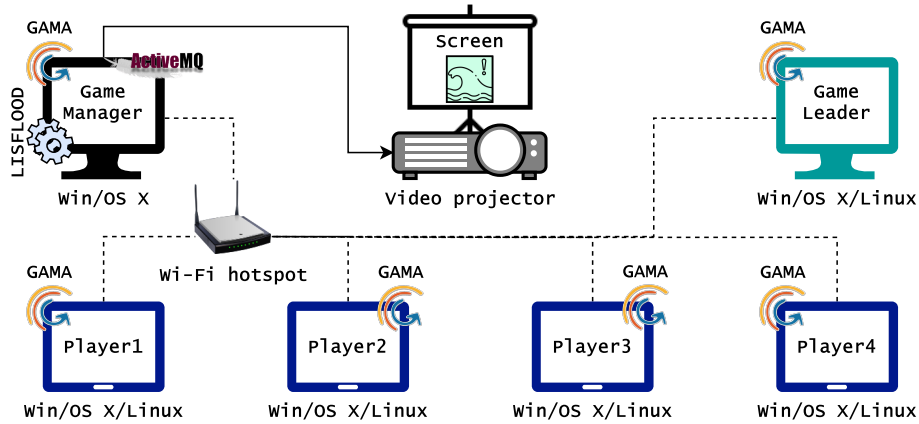


Figure 11: A LittoSIM-GEN workshop needs several material components. A powerful computer to host the manager model (this machine runs GAMA, ActiveMQ, and LISFLOOD that requires a strong CPU). A video projector connected to the game manager to display the submersion in the shared space. A computer (or tablet) to play the role of the risk agency (game leader). Four tablets (or computers) to play the districts. A Wi-Fi to connect all the components to the server (game manager). All machines can be under Windows, OS X, Linux, or any other OS supporting Java 1.8, except the game manager that hosts LISFLOOD, which requires Windows or OS X.

4.2.9. Observation

The game manager has a set of real-time indicators used at the end of the game for discussion and debriefing. Some of these indicators are also available for the game leader to optimize his decisions. Players have a limited view of the environment and cannot see the global information until they are in the shared space with the game manager.

4.3. Details

4.3.1. Implementation details

LittoSIM-GEN includes several distributed tools combined to allow playing the game between different actors implied in coastal flooding management. The agent-based model is implemented under the GAMA platform (Taillandier et al., 2019) and uses empirical data to display realistic views of the study area. Flooding simulation uses the LISFLOOD-FP (Neal et al., 2011) model to compute actual flood extent based on real submersion events. Network connections use Apache ActiveMQ wrapper⁵.

GAMA platform. GAMA (Taillandier et al., 2019) is an Eclipse-based modeling platform for developing spatial agent-based simulations. It allows manipulating vector and raster files and provides many features and libraries to ease coding and implementing large spatial agent models. All devices implied in the game must install GAMA (Figure 11). The current LittoSIM-GEN requires GAMA1.8 with Java version 1.8 or later.

ActiveMQ. LittoSIM-GEN is a distributed game that can be played over the network using Apache ActiveMQ broker. Models have networking agents that use a shared mailbox to exchange data through the MQTT protocol. The ActiveMQ server is usually installed on the game manager (Figure 11) but can be in any other accessible machine on the network.

LISFLOOD-FP. LISFLOOD (Neal et al., 2011) is a 2D hydrodynamic model that simulates inundations by calculating water elevation on a grid space according to a predefined flood event. It must be on the manager (Figure 11) under Windows or OS X systems on a machine with powerful calculation capabilities⁶. LISFLOOD needs a set of parameters about the reference event and the study area to submerge. A ".bdy" file stores the event scenario as a time series of water elevations. A second ".bci" file specifies to LISFLOOD a set of geographical coordinates where the submersion will start to discharge (inflow points). Finally, LISFLOOD needs a territory representing the study area where the submersion occurs. Two raster grids represent the territory: a DEM file as a grid of soil altitudes and a

⁵<https://activemq.apache.org/>

⁶LISFLOOD is better under OS X. With a MacOS Mojave 10.14.3, MacBook Pro 2018, 2.6 GHz Intel Core i7, 16 Go 2400 MHz DDR4, it needs 4 minutes to calculate the submersion on a 631×906 grid of 20×20 m².

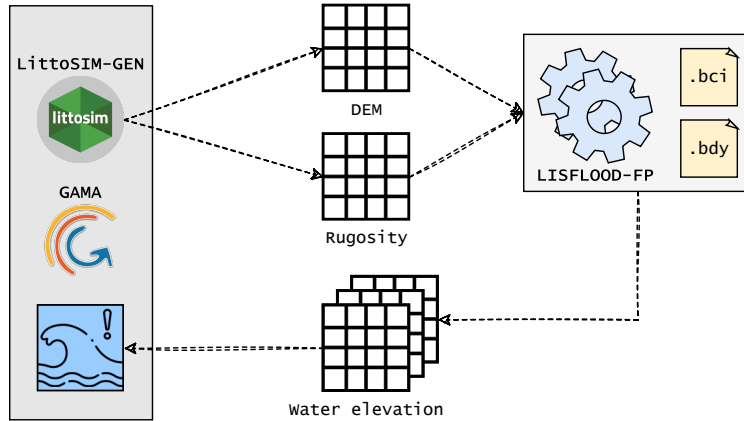


Figure 12: Coupling LittoSIM-GEN and LISFLOOD-FP. The hydrodynamic model uses the configuration files (".bdy" and ".bci") and the updated grids (DEM and rugosity) to calculate flood propagation, then returns a set of water elevation grids to display under GAMA.

rugosity (roughness) file as a grid of Manning coefficients (resistance to water flow). LittoSIM-GEN produces these two grids each time the manager launches a flood event. Generated grids incorporate all player actions that change the nature of the soil. Therefore, changing coastal defenses by creating, raising, or destroying dikes and dunes affects soil altitudes (DEM), while land use management (changing the type of a unit) affects the rugosity grid since the Manning coefficients depend on the nature of the land cover. As a result, LISFLOOD returns a set of grids (14 files) with water elevation at each iteration. The platform uses these grids to display the propagation of the submersion event in 14 timesteps (Figure 12).

4.3.2. Initialization

LittoSIM-GEN reads simulation data and parameters from five configuration files:

littosim.conf. This is the main configuration file and contains general parameters such as network server address, default language, LISFLOOD path, and paths towards the other configuration files.

study_area.conf. It is the configuration file storing all specific parameters to each study area (territorial archetype), such as paths of shape and raster files, simulation parameters, actions and levers details, etc. Applying LittoSIM-GEN to a new case requires configuring a new study area configuration file.

actions.conf. This file is also specific to the study area and contains parameters (cost, delay, description) of available player actions (build a dike, change a land-use unit from N to A, etc.). Each player can have a different configuration of actions. Player actions implemented in the current LittoSIM-GEN are shown in Table 1.

levers.conf. It is a specific file for each study area and includes available levers with their parameters (threshold, cost, delay) that can be enabled or disabled for each district in the game. Table 2 lists the currently supported levers.

langs.conf. This file gives each displayed message the translation in different languages supported by LittoSIM-GEN (currently: French, English, and Vietnamese). Each model can have a distinct language.

4.3.3. Input Data

This paper describes the data of the case study of Rochefort. Data preparation for LittoSIM-GEN is detailed in Laatabi et al. (2020).

Data overview. Data represent the districts of Rochefort estuary in southwest France. Administrative and topographical data correspond to the IGN (National Institute for Geographic and Forest Information) BD TOPO® database. Land cover data are from the European Union Corine Land Cover (2018), and Local Urban Planning (PLU) data are provided by local authorities. Protected natural areas and Risk Prevention Plan (PPR) files are from the local French State. Coastal

Table 3

LittoSIM-GEN input data files. GAMA uses shapefiles to display different aspects of the study area (Figure 2). The two raster files are used by GAMA and by LISFLOOD. Text files (".bdy" and ".bci") are the configuration files to LISFLOOD.

File	Type	Description
districts	vector	The districts of the study area (Figure 2b)
convex_hull	vector	A rectangular buffer enveloping the study area
buffer_in_100m	vector	A buffer of 100m inside the districts' shape (Figure 2d)
land_use	vector	A grid of land use and planning cells (Figure 2g)
urban_plan	vector	The local urban plan (PLU: Plan Local d'Urbanisme) specifying the town planning (Figure 2f)
land_cover	vector	The Corine Land Cover (CLC) specifying the land cover type (Figure 2j)
buildings	vector	The buildings of the study area (Figure 2e)
roads	vector	Lines representing principal roads (Figure 2l)
rivers	vector	Lines representing principal rivers (Figure 2k)
coastline	vector	A line representing the coast
coastal_defenses	vector	dunes and dikes protecting the coast (Figure 2c)
spa	vector	Natural and special protected areas (Figure 2i)
rpp	vector	The risk prevention plan (PPR) of flooding risk areas (Figure 2h)
dem	raster	The digital elevation model (DEM) grid (Figure 2a)
rugosity	raster	The roughness (resistance to water) grid (Figure 2j)
file.bdy	timeseries	A timeseries of the water elevation scenario
file.bci	boundaries	Geographical boundaries of the domain

defenses data are from the local authorities and the Artelia group engineering office, and LISFLOOD data scenarios are based on the Xynthia storm (Bertin et al., 2014). Geospatial data are reprojected into the RGF93/Lambert-93 coordinate reference system (EPSG:2154).

Data structure. Table 3 lists data files required by the LittoSIM-GEN model. These files are recorded in the *study_area.conf* configuration file and then loaded into the simulation as input data. Table 4 depicts the data structure of each file (attributes and their corresponding values). Besides these attributes, all files have a default auto-generated numeric ID to identify each spatial object in the shapefile.

Data mapping. The mapping scheme in Figure 13 represents entities: (i) in blue, the original structure of Rochefort data; (ii) in yellow, the target data structure. Mapping patterns (gray ovals) summarize data transformations to the LittoSIM-GEN structure. Basic patterns are self-explaining (e.g., Rename is a simple operation of copying and renaming the original data). Complex transformation patterns are identified with their output name (file or attribute) and explained in the next section (*data patterns*). More details about data mapping and transformation patterns in LittoSIM-GEN are in Laatabi et al. (2020).

Data patterns.

- `districts`. represents the districts of the study area (estuary of Rochefort).
 - `dist_area`. generates the area of a district by calculating the area of its polygon.
- `buffer_in_100`. represents a bounding zone of 100 meters inside the unified area of districts and is used to identify a coastal defense (dike) as littoral or inland (retro-dike).
- `convex_hull`. generates a convex hull specifying the global study area for the simulation as a rectangular closure (envelope) around districts.
- `buildings`. represents the buildings of the study area.
 - `bld_type`. Since only residential buildings are relevant, this attribute contains two values (`{Residential, Other}`) created by a pattern of replacement by replacing "*résidentiel*" with "*Residential*" and all other values "*" with "*Other*".

Table 4

The structure of LittoSIM-GEN input data: besides these attributes, all shapefiles have a unique feature 'ID' of type integer to identify different spatial objects.

File	Attribute	Type	Values	Description
districts	dist_code	string		A unique code of the district
	dist_pop	integer		The total population
	dist_area	double		Total area of the district (m ²)
buildings	bld_type	string	{ <i>Residential, Other</i> }	The type of the building
	bld_area	double		The building area (m ²)
urban_plan	unit_code	integer	[1,5]	The PLU type of the area
land_cover	cover_type	integer	{111, ..., 523}	The CLC type (44 classes)
coastal_defenses	dist_code	string		The district code of the object
	type	string	{ <i>Dike, Dune, Pebble</i> }	The type of the object
	status	string	{ <i>Good, Medium, Degraded</i> }	The status of the object
	alt	double		The altitude of the object (m)
land_use	height	double		The height of the object (m)
	unit_code	integer	[1,5]	The type of the cell
	sub_type	integer		The sub-type of the cell
	unit_pop	integer		The population of the cell
	dist_code	string		The district code of the cell
	unit_area	double		The area of the cell (m ²)
	expro_cost	double		The expropriation cost of the cell

- `bld_area`. calculates the area of the building based on its polygon.

- Topology. The topological verification of `buildings` file consists in splitting polygons intersecting with several land-use units and deleting buildings in natural areas.

- `urban_plan`. represents the land use planning (PLU) that determines the exploitation type of local areas.
 - `unit_code`. specifies the type of each area in urban planning based on the original file with a type conversion (text to numeric) and replacement of values as follows:

$$\{N, Nh, Na\} \rightarrow 1, \{U, Us\} \rightarrow 2, \{AUc, AUb, AU, AUs\} \rightarrow 4, \{A, Az, Ax\} \rightarrow 5.$$

- `land_cover`. determines the land cover type of the area based on the European CLC database.
 - `cover_type`. specifies the type of each zone in the study area. This attribute can take 44 standardized values (111, ..., 242, ..., 523) representing various land covers (forests, airports, rice fields, etc.).
- `land_use`. is generated as a grid cell of a specified size (200×200 m² for Rochefort) covering the area of the active districts. This grid is the principal component allowing planning and land use management (through editing cell units) in LittoSIM-GEN.
 - `unit_code`. is created through an intersection between the land-use unit (one cell of land use) and the urban planning file (`urban_plan`). It specifies the planning type of the cell and initially takes four values: 1 (Natural N), 2 (Urban U), 4 (Authorized-for-urbanization AU), and 5 (Agricultural A). This variable may change during the game and take other values: 6 (Adapted-Urban Us), 7 (Authorized-for-Adapted-urbanization AUs).
 - `sub_type`. this attribute is created through an intersection with the `land_cover` file to specify the sub-type of a land-use unit. For example, in some cases, agricultural areas may have various sub-types.
 - `dist_code`. affects the district code to each land-use unit through an intersection with the `districts` file. The model splits boundary cells to fit within districts.

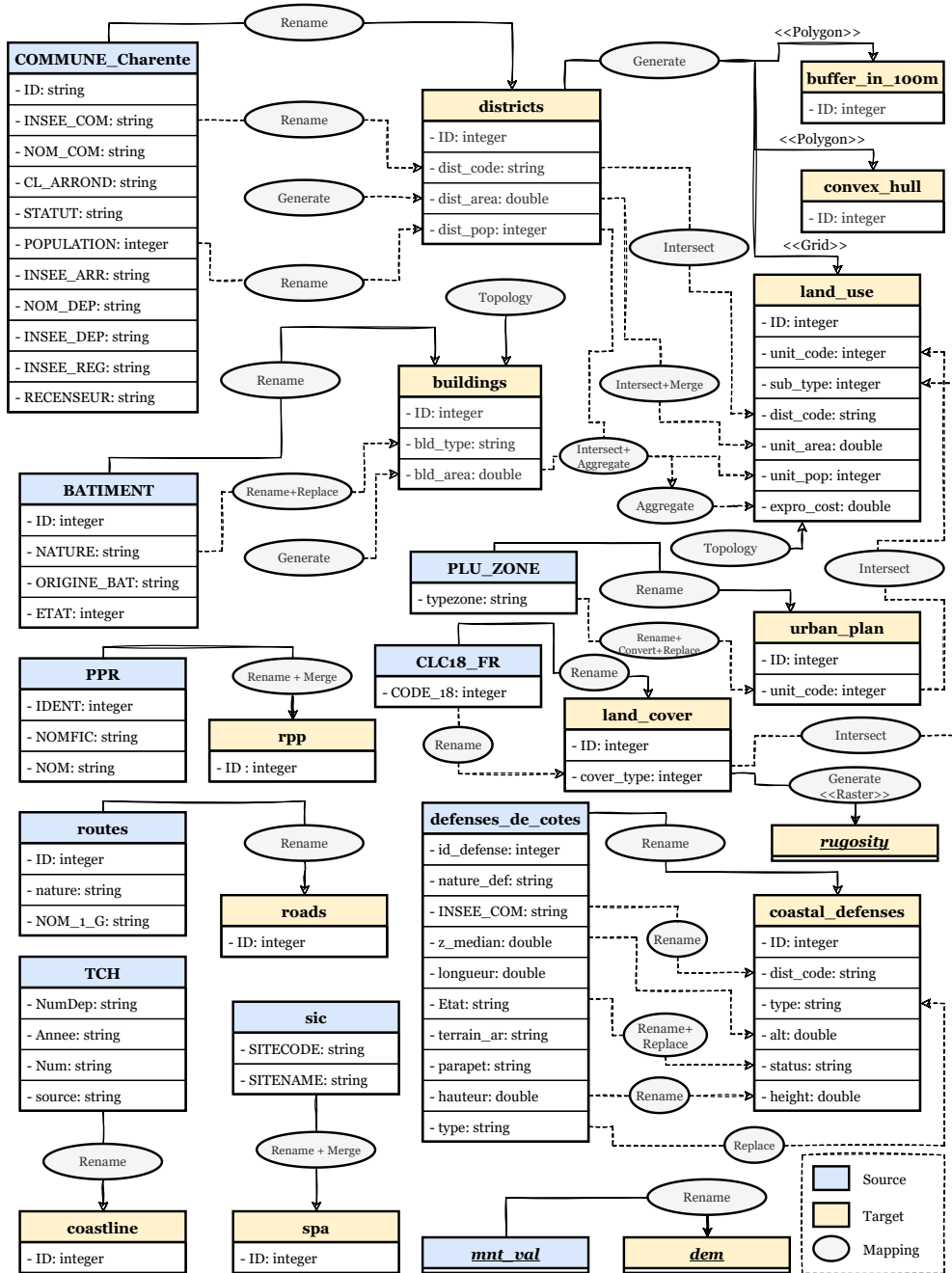


Figure 13: Mapping diagram of LittoSIM-GEN-Rochefort data. Blue entities: sources; Yellow entities: target data; Solid lines (—): file patterns; Dashed lines (- - -): attribute mapping.

- `unit_area`. calculates the area of each `land_use` unit. For ergonomic purposes (players cannot select small-scale cells), a cell smaller than a specified minimum size (20.000 m² for Rochefort) is merged with the neighbor that shares the longest border with it.

- `unit_pop`. uses aggregate and intersect patterns to calculate the population of each land-use unit. It takes the sum of the area of residential buildings intersecting the cell (`bld_area`), divides it by the sum of the areas of residential buildings of the district, and multiplies the result by the total population of the district (`dist_pop`). LittoSIM-GEN considers that the population rate is the proportion of residential buildings within

each cell (equation 1).

$$\begin{aligned} \text{resid_blids} &\leftarrow \text{buildings}[\text{bld_type} = \text{"Residential"}] \\ \text{unit_pop} &\leftarrow \frac{\sum(\text{bld_area}, \text{resid_blids} \cap \text{land_use})}{\sum(\text{bld_area}, \text{resid_blids} \cap \text{districts})} \times \text{dist_pop} \end{aligned} \quad (1)$$

◦ *expro_cost*. The expropriation (transforming to natural) cost of an urban cell depends on its population. Empty cells (*unit_pop* = 0) take a predefined parameter (*empty_expro_cost*) in the *study_area.conf* file. Populated cells are expropriated with a cost that follows the next function (equation 2). This cost increases less as the cell population goes up. The number 400 represents the expropriation cost of a cell with one inhabitant.

$$\text{expro_cost} \leftarrow \frac{\text{unit_pop} \times 400}{\sqrt{\text{unit_pop}}} \quad (2)$$

◦ *Topology*. The topological verification of the land use grid consists of fixing cell merging issues and avoiding that a natural cell contains buildings or that an agricultural cell has populations (agricultural cells may contain non-residential buildings).

- *coastal_defenses*. represents the dunes and the dikes that protect the coastline of districts from flooding.
 - *type*. specifies the type of a coastal defense among the two possible values. This attribute is created with a replacement of original values: "*Naturel, Cordon*" become "*Dune*" and all other values "*" become "*Dike*".
 - *status*. determines the quality of a coastal defense that affects its probability of rupture during flood events. This attribute is renamed and replaces old values as follows:

"bon" → "Good", "moyen" → "Medium", "" → "Degraded"*

- *rugosity*. The rugosity grid has the same dimensions as the dem grid (732×472 of 20×20 m² cells for Rochefort). It is created based on a predefined set of rugosity coefficients (Manning) depending on the land cover of the area covering each cell. The initial rugosity grid is generated with the detailed *land_cover* (CLC) data. Cells updated during the game take simplified coefficients corresponding to the land-use cell type. In the case of Rochefort, we use the following coefficients: Natural N (0.11), Urban U (0.05), Authorized-for- Urbanization AU (0.09), Agricultural A (0.07), Adapted-Urban Us (0.09), Authorized-for-Adapted-Urbanization AUs (0.09). Therefore, each updated standard *land_use* unit (200×200 m²) contains 100 rugosity cells (20×20 m²) with the same coefficient.
- *spa*. The protected areas file is renamed, and its polygons merged to form a unified polygon. This merge prevents potential geometry errors (intersections).
- *rpp*. The risk areas file is also renamed and merged for the same reason as the *spa* file.

4.3.4. Submodels

We describe the principal dynamics implemented implicitly in the LittoSIM-GEN model, namely the evolution of populations, budgets, and coastal defenses. Explicit player actions trigger all other dynamics.

Population dynamics. At each round, LittoSIM-GEN calculates the number of individuals to dispatch among different districts. The model supports population growth and decline (the dispatched number of people may be positive or negative). This dynamic is computed as depicted in Equation 3. First, it calculates the population growth based on the *ANNUAL_POP_GROWTH_RATE* parameter, then adds an attraction factor to attract people according to the number of land-use units in the densification process (*U_i* state).

$$\begin{aligned} \text{pop_to_dispatch} &\leftarrow \\ &[\sum (\text{District } \textit{accumulate} \text{ each.current_population}) \times \text{ANNUAL_POP_GROWTH_RATE}] + \\ &[\# (\text{Land_Use } \textit{where} \text{ each.is_in_densification}) \times \text{POP_IMMIGRATION_IF_DENSIF}] \end{aligned} \quad (3)$$

The number *pop_to_dispatch* is then distributed randomly to urban land-use units of the study area until it reaches 0. Each urban unit takes a number according to its area and type: AU that becomes U takes *POP_FOR_NEW_U* people, *U_i* takes *POP_FOR_U_DENSIFICATION*, and U/*U_s* take *POP_FOR_U_STANDARD* individuals.

Budgets. The budget is first calculated based on the initial population and predefined tax table rates (IMPOT_UNIT_TABLE), then it is increased with a fixed bonus rate (INITIAL_BUDGET_BONUS). During the game and at each new round, a tax is calculated based on district population and the previous tax rates, then this tax is added to the total budget. Equation 4 shows this dynamic for each district.

$$\begin{aligned} & \text{TAX_UNIT} \leftarrow \text{IMPOT_UNIT_TABLE} [\text{district_name}] \\ \textit{At initialization} : & \text{budget} \leftarrow \text{current_population} \times \text{TAX_UNIT} \times (1 + \text{INITIAL_BUDGET_BONUS}) \\ \textit{Then} : & \text{budget} \leftarrow \text{budget} + (\text{current_population} \times \text{TAX_UNIT}) \end{aligned} \quad (4)$$

Coastal defenses. At each game round, a counter is increased by one, and the model checks all coastal defenses and updates the status of objects that reached one of the defined thresholds. At each submersion event, a coastal defense object has a probability of breaking depending on its state. The study area configuration file is used to initialize all parameters.

Dikes. Each dike has a counter specifying the number of remaining rounds before it degrades to the down status (Algorithm 1). The degradation occurs when this counter reaches STEPS_DEGRADE_STATUS_DIKE.

Dunes. Each dune has a counter specifying the number of remaining rounds before changing its status (Algorithm 2). A dune degrades to down status after STEPS_DEGRADE_STATUS_DUNE game rounds if no sand fences are installed or when its accretion is not enhanced. Otherwise, it upgrades to the upper status after STEPS_UPGRADE_STATUS_DUNE rounds. A maintained dune does not update its status for STEPS_MAINTAIN_STATUS_DUNE rounds.

Pebble dikes. Each pebble dike has several slices and loses NB_SLICES_LOST_PER_ROUND at each game round. Two parameters determine the thresholds for status degradation depending on the number of remaining slices (Algorithm 3). The player action "loading pebbles" triggers the upgrade of pebble dikes.

5. An application of LittoSIM-GEN

In this section, we apply LittoSIM-GEN to the case study of Rochefort (estuary coast) in southwest France, where the Xynthia storm (Bertin et al., 2014) caused considerable damage in 2010. Applying the LittoSIM-GEN model to a territory is a simplified process that requires only two tasks. The first one consists in collecting raw data from various sources and transforming them to the input data structure of LittoSIM-GEN (tables 3 and 4). Laatabi et al. (2020) explain the data preparation process performing the automatic transformation based on XML/R scripts, and the sources are in Appendix A.1. Processed data files are included in the new project folder containing the related configuration files. The second task is to parameterize those configuration files to use the newly prepared data. The study_area.conf configuration file must include all general parameters concerning the study area, such as the names of districts, the resolution of raster grids, and the paths towards source data files. The actions.conf must contain only relevant actions, and relevant levers are in levers.conf. After configuring these files with the right parameters, LittoSIM-GEN is ready to run the new project.

5.1. LittoSIM-GEN platform with Rochefort data

The LittoSIM-GEN platform offers multiple graphical interfaces for the three agent models (game manager, game leader, and player). We demonstrate the tool's features through examples from the Rochefort case study.

In the game manager model, the game control interface offers a set of buttons to manage various aspects of the simulation (switch rounds, pause and resume the game, launch submersion events, etc.). The flooding display (Figure 14a) uses the DEM grid to show the altitude map of the territory where the submersion extent and ruptures of coastal defenses are displayed. Figure 14b shows the area of flooded cells by land use type and water height. The blue scale color represents the three levels of water heights. Additional displays show the evolution of populations and budgets, player actions and strategies, land use and coastal defenses, and the flooded area at each submersion event.

In the game leader model, the main interface (Figure 15a) allows the risk agency to follow the activated levers for each district, send messages, and make financial transactions. An activated lever can be validated, canceled, or disabled (to prevent its activation until it is enabled again). The actions' interface (Figure 15b) details the strategic tendency of each playing team by providing the number of each player action by strategy ([Builder, Soft defense, Strategic withdrawal]). Other interfaces allow the game leader to follow the principal indicators representing the evolution of populations and budgets of each district and the profiling of player actions. The game leader uses these graphs to decide about levers, financial transactions, and other decisions used to orient and interact with players.

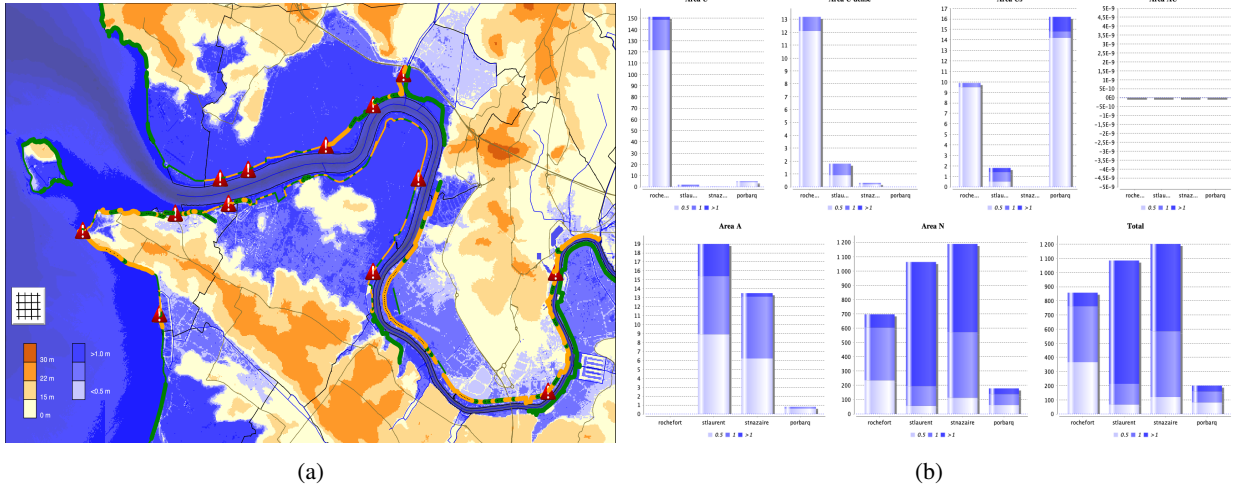


Figure 14: (a) Flooding display with maximum water heights and ruptures at the end of a submersion event. (b) Flooded areas by land-use type.

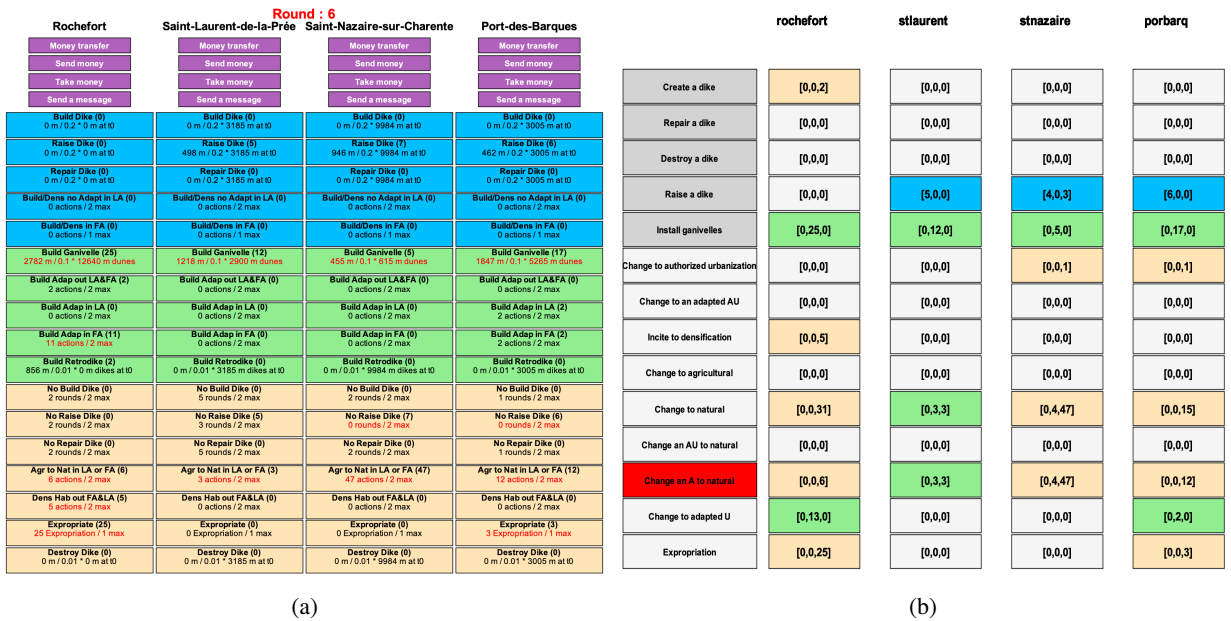


Figure 15: (a) The game leader's main control panel to supervise activated levers. (b) The number of executed player actions by strategy.

In the player model, two different displays (Figure 16) allow players to manage the land use and coastal defenses. The top-left part of each interface shows the area where players can select spatial objects. Buttons on the top-right display additional information on the map, such as risk and protected areas, landmarks of submersion events, and the history of previously executed actions. Buttons on the top-left present the available player actions. At the top-right side, the Basket lists all selected player actions that may be validated or canceled. The History component at the bottom-right of the window displays validated player actions. Received messages at the bottom provide players with information about their territory and about the current state of the game.

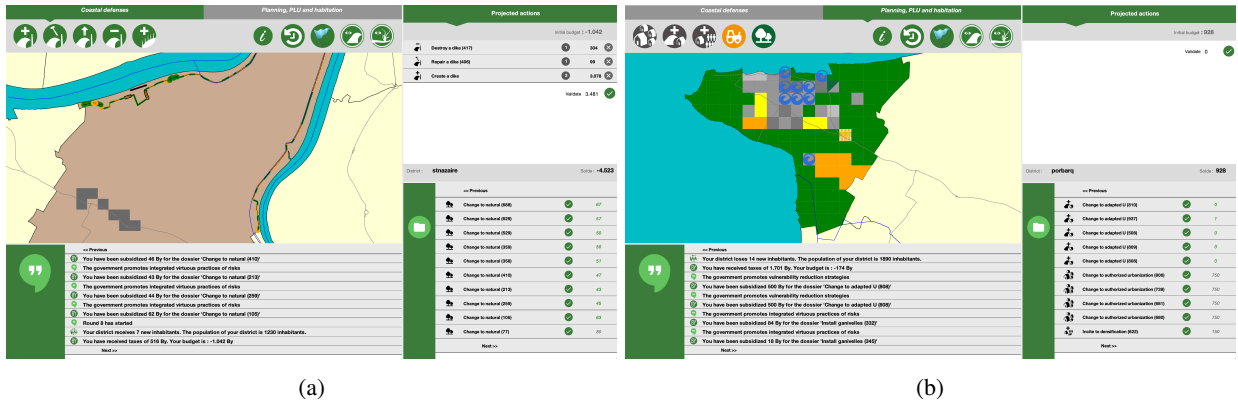


Figure 16: (a) Coastal defenses display. (b) Land use planning display.

5.2. LittoSIM-GEN workshops and results

LittoSIM-GEN is designed to organize participatory simulations during workshops involving local stakeholders and decision-makers (Figure 17a). Playing teams are usually composed of two or three individuals (technicians and members of the district council). Larger groups may generate extended discussions and hamper the planning of the workshop. Each district team takes place on a separate table from the others, and meetings with the risk agency are controlled by a timekeeper (who also facilitates the workshop) to avoid losing much time in discussions to the detriment of playing the game (Van Hardeveld et al., 2019). The platform collects data about the study area (land use and coastal defenses), flood events, player actions, and game leader activities. In parallel, an observer team watches the game and collects additional data about participants' actions, interactions, and behaviors. Further analysis using collected data and pre-/post-workshop surveys allow obtaining more results about learning and adaptation strategies. Conclusions are communicated to all participants in the game through reports or debriefing.

The platform is implemented to be user-friendly without neglecting principal aspects that give realistic results (Amalric et al., 2017). The hydrodynamic model used by LittoSIM-GEN (LISFLOOD (Neal et al., 2011)) calculates the submersion extent according to the reference event and the configuration of the supplied territory. Figure 17b shows, for example, a comparison between the actual and the simulated extent of the Xynthia storm (Bertin et al., 2014) in the estuary of Rochefort. This realism enhances the simulation results (Voinov et al., 2018), particularly the relationship between player actions and the flooding damage. Discussions and debriefing during these workshops foster the social learning of participants (Van Hardeveld et al., 2019) by simulating scenarios where collaborative projects and collective management of the risk may be the efficient way to reduce flood damage and protect coastal populations and activities.

6. Discussion: Genericity in participatory simulation models

This section discusses the experience of developing the generic version of LittoSIM. Generic models are transparent and easier to extend to other contexts and use cases, and their development can be carried out by teams other than the initial developers. We structure this experience as recommendations to practitioners of participatory modeling and simulation.

6.1. Participatory conception

Local stakeholders must be involved in the conception and development of participatory simulations (Voinov et al., 2018). Their internal view of the problem, external to the solution, helps develop contextualized models easily applicable to the research question. Involving decision-makers and domain experts guarantees the link with empirical observations representing real phenomena (Laatabi et al., 2018) and makes the model observable for the local stakeholders (Klabbers, 2009). This engagement allows the model to include various potential needs of final users in terms of features and functionalities. Stakeholders consider themselves as participants in the model creation (ownership), which may push them to take more responsibilities in its implementation and application to real problems (motivation). In the case of LittoSIM-GEN, stakeholders have contributed to various aspects of the contextualized

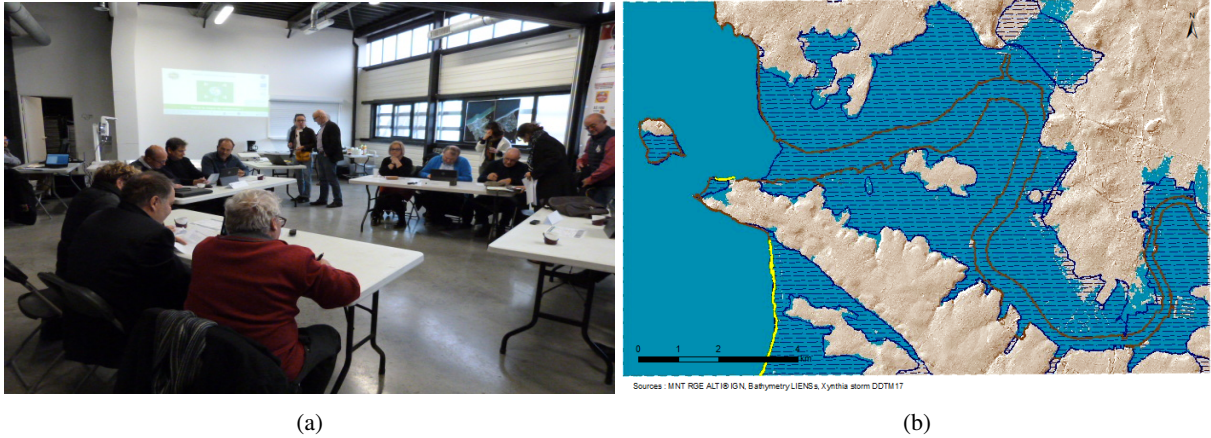


Figure 17: (a) Playing teams during a LittoSIM-GEN workshop (Normandie, North France). (b) The true extent of the Xynthia storm (dashed areas) vs. LittoSIM-GEN simulation (blue areas).

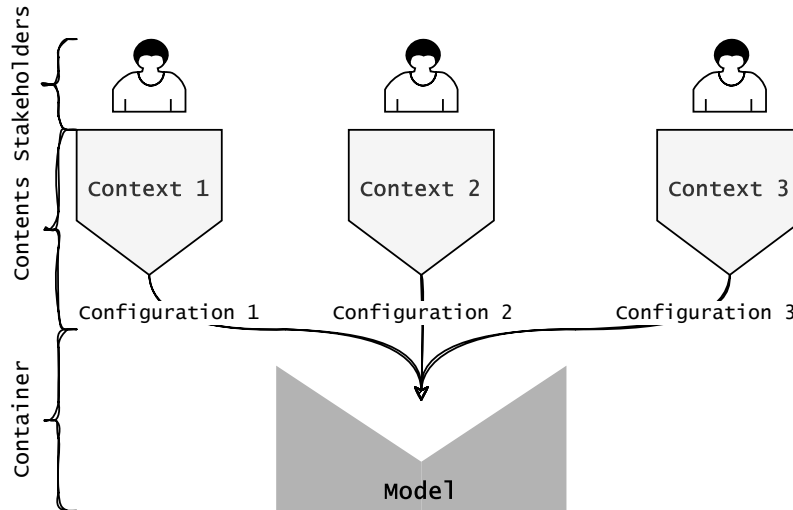


Figure 18: A generic model can take several contexts/contents in the same model/container with a simple reconfiguration (context specification).

model: player actions and their details, applied levers and their parameters, game rules, decision-making process, the evolution of populations and territories, submersion scenarios, empirical data, etc. These contributions concern the context of the model and its components that also shape its conception. The generic architecture separates the context as an independent element from the model itself.

6.2. Model architecture

The context is the set of elements that compose a case study. It includes empirical data, model dynamics, and simulation parameters. The context is independent of the model that may contain various contexts with different configurations (Figure 18). Therefore, a context is loaded dynamically into the model through external tools such as configuration files. The separation between the context and the model allows one to apply the same model to different case studies with a simple reconfiguration and to edit the rules and the data (context) without modifying the model itself (container).

A generic model is a container reusable with several contexts, and it is the context specifications that define the model specificity. However, if a model is too generic, it becomes difficult to be reused as more adaptation and configuration are needed to load new content. Developers must maintain the model within the reusability domain (Figure 19).

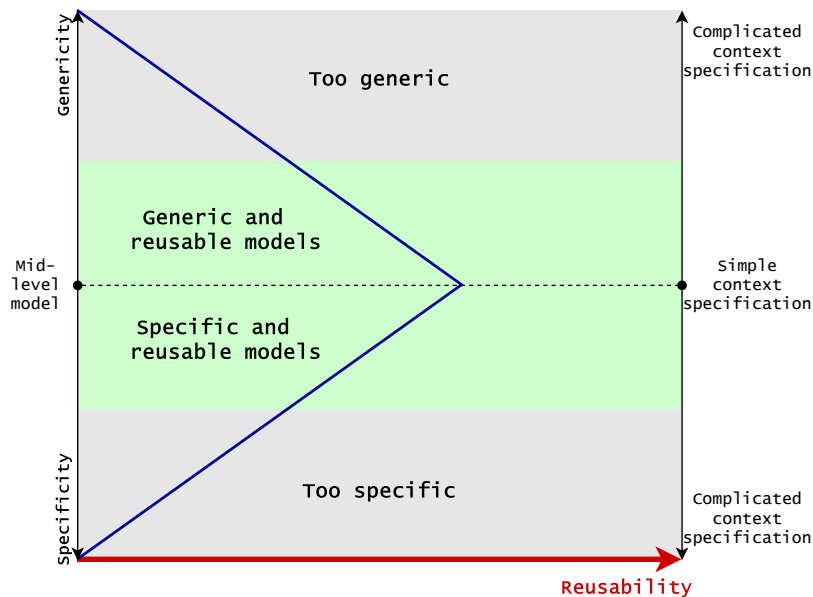


Figure 19: The blue line represents the evolution of reusability according to the genericity/specificity of the model and the simplicity/complexity of the context specification. Models in the intermediate zone (in green) are generic and reusable in other contexts without much adaptation. Models with a high level of genericity lose their descriptive capabilities and require more configuration efforts to be reused (complex context specification). Specific models with a high level of description require much adaptation as well. The green zone represents the ideal equilibrium for models between genericity/specificity and reusability.

This domain is the equilibrium between the genericity/specificity of the model and the simplicity/complexity of the context specification. In this zone, the generic model is reusable with other contexts without being complicated to configure. Very generic models need complex context specifications (data, configuration) that hinder their reusability. Hence, the complexity of the context specification (adaptation and setting) is a good measure of reusability. This trade-off between genericity/specificity is similar to the one proposed by Sun et al. (2016) to develop mid-level models as an intermediate between theoretical and empirical modeling, and the one proposed by Klabbers (2009) to adopt a game design that balances between generic, specific, and observable qualities.

The principal components of the model must be separated to avoid monolithic models and build a resilient model composed of many independent and related parts. Such models guarantee a high level of flexibility and fault tolerance. Developers can modify or replace a single component or add a new one independently from the rest of the model. Flexible links couple various parts with each other but keep their independence. Those coupling links manage the compatibility between model components and allow reorganizing the model's architecture without much effort.

LittoSIM-GEN uses configuration files to load the study area (context). The externalization of data and parameters allow playing the same model as a different game in different territories. The configuration of player actions and levers can differ between case studies or between players of the same game. The three agent models (game manager, game leader, player) are implemented separately, and the cessation of one component does not affect the others. The connection between these components through the exchange of text messages allows any new role to retrieve exchanged data by connecting to the shared mailbox. The multi-component architecture also offers the possibility of playing the game with any number of players. The hydrodynamic model (LISFLOOD) is coupled with the agent-based model through a grid-based link (Figure 12) that may be modified to use any other compatible model. This architecture allows a fully modular design of the simulation (Jaxa-Rozen et al., 2019).

6.3. Model and data description

Genericity and reusability require other users and developers to understand the model. Therefore, a detailed description explaining the structure and the dynamics of the model is needed. A standard protocol is preferred to organize and disseminate model descriptions. The ODD protocol (Grimm et al., 2020), for example, presents the

necessary information to understand, replicate, and reuse agent-based models. By explaining the model structure, such descriptions address the model genericity-complexity issue and drive any new use or adaptation to other contexts. But to fully bridge the gap between the model and the context, descriptions should also describe how data are used and transformed to suit the model structure. The ODD+2D protocol (Laatabi et al., 2018) adds this feature to the standard ODD and extends the description to the data-model relationship. When transforming raw data to the model structure is complicated, more tools may be needed to foster the reusability of models, for example, algorithms or automated scripts that perform data transformation.

In the case of LittoSIM-GEN, this paper provided an ODD+2D description to help readers understand the structure of the model in terms of theoretical concepts, agent dynamics, process scheduling, and empirical data. An automated script transforms data to the LittoSIM-GEN format based on a mapping configuration file (Appendix A.1). This data compiler saves the time and effort necessary to use LittoSIM-GEN with data from other territories.

6.4. Simulation use and handling

Participatory simulations designed for stakeholders or other non-expert public must guarantee a high simplicity of use (Bakhanova et al., 2020). Game participants often highlight the ergonomics of the gameplay as a weakness of computer-based participatory simulations (Becu et al., 2014). Facilitation capabilities (Voinov et al., 2018) should target intermediate users (workshop animators) and final users (game players). Hence, control panels and outputs should use simple and user-friendly interfaces to display only relevant information. User manuals (for both intermediate and end-users) are also a useful tool to disseminate platforms and ease handling them. These manuals should include information about required material configuration, installation, setting, and a user guide to manipulate the tool and get full access to its inputs/outputs.

In LittoSIM-GEN, user interfaces give access to the functionalities of different models. The control panel of the game manager allows to switch game rounds and launch submersion events. Multiple interfaces display additional game details (Figure 14). The game leader uses several indicators to supervise the game (Figure 15). Players have a user-friendly interface to manage their territory and trace their actions to optimize their future decisions (Figure 16). Game players are initiated to the game during a warm-up lap at the beginning of the workshop. An ergonomist participates in the project to improve the efficiency of graphical interfaces. Finally, user manuals and training courses serve to train future workshop animators.

6.5. Knowledge extraction and learning

Developed tools must collect data during games and workshops to fulfill the objective of participatory modeling and simulation (Hedelin et al., 2021). Such data are essential for post-analysis using statistical and machine learning methods to extract knowledge and hidden patterns that cannot be observed in real-time. This task is complementary to the learning process initiated at the game time (Van Hardeveld et al., 2019; Van der Wal et al., 2016) and helps to understand how the participants have responded to different situations. The analyzed data contain quantitative and qualitative records collected by the platform, remarks and notes taken by observers, and potential pre/post surveys (Voinov et al., 2018). Such data give an idea of how players have evolved and learned during the game.

In the case of LittoSIM-GEN, all the previous features are combined (saved data, observer notes, surveys), and a set of R scripts analyze collected data and visualize their principal aspects. An additional quantitative analysis estimates how LittoSIM-GEN contributes to learning and raising awareness about coastal risk management. The structure of collected data and the protocol of data analysis must be maintained to keep the generic aspect of the platform and to be able to compare the results of different games with diverse territorial archetypes.

7. Conclusions and perspectives

In this paper, we described the LittoSIM-GEN agent-based model and platform. This tool allows conducting workshops to foster learning about coastal flooding risk management. Externalized configuration parameters make LittoSIM-GEN a generic and reusable model with new case studies. New users can edit only three configuration files to reuse the platform with new data. The resulting environment is a set of graphical interfaces that display the evolution of the study area based on player actions and scenarios of flooding events. Several game outputs provide necessary information to improve game efficiency and engage discussions and exchanges between participants. In parallel, the platform saves all results and game states for further analysis. Analyzing outputs of LittoSIM-GEN is essential to detect latent aspects and hidden patterns.

LittoSIM-GEN is a realistic gaming simulation that uses empirical data, involves stakeholders, and simulates actual submersion events. The flooding model (LISFLOOD) calculates the true extent of submersion based on the state of the territory and the predefined event scenario. Integrating hydrodynamic models into risk management platforms is not straightforward and makes accessing results difficult for non-experts (Jaxa-Rozen et al., 2019; Zenger and Wealands, 2004). The architecture of LittoSIM-GEN simplifies this task by using a grid approach. The model currently uses LISFLOOD, but due to the modularity of its architecture, it may integrate any other flooding model that fits into the coupling scheme (Figure 12).

From this experience to generalize the LittoSIM model, we draw a set of recommendations to develop generic and reusable participatory simulation models: separating model and context specifications, multi-component architecture and modularity, model and input data description, ergonomics of user interfaces and simulation handling, and output data collection and parsing. These recommendations foster the reusability of models and save time and effort when developing similar tools and solutions.

The future aim of the LittoSIM-GEN project is to offer a generic and documented toolkit easy to reuse for coastal flooding simulation and risk management. The model is currently applied to four different French territories: Oléron and Camargue (overflow coasts), Normandie (cliff coast), and Rochefort (estuary coast). Implementing the model by other users for different case studies is essential for:

Validating. Applying LittoSIM-GEN with different data, users, and experiences is a validation criterion of its genericity. The validity of LittoSIM-GEN concerns multiple components: agent-based (GAMA) and flooding (LISFLOOD) models, operating instructions and equipment, and the participatory process related to the organization of workshops.

Extending. The flooding model (LISFLOOD) simulates specific types of submersion and may not be suitable for all case studies. Hence, other contexts such as continental or overtopping flooding may need a different hydrodynamic model. LittoSIM-GEN can be used in these contexts since its architecture allows to couple the platform with any other grid-based submersion model.

Improving. Testing the model by new teams other than its developers may lead to improvements and changes in its conceptual model, architecture, building blocks, and the implication of stakeholders in the game.

Popularizing. By diffusing the model among the scientific community interested in coastal flooding management, LittoSIM-GEN will be reused and therefore validated, extended, and improved by third parties.

Future developments of LittoSIM-GEN will focus on more dynamical aspects, such as the extensibility of the model components. These features will add new functionalities and game rules through external configuration files without modifying model sources.

Acknowledgements

This research was conducted with the financial support of Fondation de France and the Nouvelle-Aquitaine region.

A. Appendix

A.1. LittoSIM-GEN sources

The LittoSIM-GEN model source code and data preparation scripts are available on GitHub (https://github.com/LittoSim/LittoSim_model/tree/LittoDev).

A.2. Algorithms

References

- Adam, C., Andonoff, E., 2019. Vigiflood: a serious game for understanding the challenges of risk communication.
- Adam, C., Bailly, C., Dugdale, J., 2018. Communication during bushfires, towards a serious game for a serious matter. *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)* 10, 79–105.
- Amalric, M., Anselme, B., Bécu, N., Delay, E., Marilleau, N., Pignon, C., Rousseaux, F., 2017. Sensibiliser au risque de submersion marine par le jeu ou faut-il qu'un jeu soit spatialement réaliste pour être efficace? *Sciences du jeu*.

Algorithm 1 Dike status degradation at each game round.

```

counter_status ← counter_status + 1
if counter_status > STEPS_DEGRADE_STATUS_DIKE then
  counter_status ← 0
  if status = STATUS_MEDIUM then
    status ← STATUS_DEGRADED
  else if status = STATUS_GOOD then
    status ← STATUS_MEDIUM
  end if
end if

```

Algorithm 2 Dune status evolution at each game round.

```

if sand_fence = true then
  counter_status ← counter_status + 1
  if counter_status > STEPS_UPGRADE_STATUS_DUNE then
    counter_status ← 0
    if status = STATUS_MEDIUM then
      status ← STATUS_GOOD
    else if status = STATUS_DEGRADED then
      status ← STATUS_MEDIUM
    end if
  end if
else
  if maintained = true then
    maintain_status ← maintain_status + 1
    if maintain_status > STEPS_MAINTAIN_STATUS_DUNE then
      maintain_status ← 0
      maintained ← false
    end if
  else
    counter_status ← counter_status + 1
    if counter_status > STEPS_DEGRADE_STATUS_DUNE then
      counter_status ← 0
      if status = STATUS_MEDIUM then
        status ← STATUS_DEGRADED
      else if status = STATUS_GOOD then
        status ← STATUS_MEDIUM
      end if
    end if
  end if
end if

```

- Bakhanova, E., Garcia, J.A., Raffe, W.L., Voinov, A., 2020. Targeting social learning and engagement: what serious games and gamification can offer to participatory modeling. *Environmental Modelling & Software*, 104846.
- Becu, N., 2020. Les courants d'influence et la pratique de la simulation participative: contours, design et contributions aux changements sociétaux et organisationnels dans les territoires. Ph.D. thesis. La Rochelle Université.
- Becu, N., Amalric, M., Anselme, B., Beck, E., Bertin, X., Delay, E., Long, N., Marilleau, N., Pignon-Mussaud, C., Rousseaux, F., 2017. Participatory simulation to foster social learning on coastal flooding prevention. *Environmental modelling & software* 98, 1–11.
- Becu, N., Bommel, P., Botta, A., Le Page, C., Perez, P., 2014. How do participants view the technologies used in companion modelling?, in: *Companion modelling*. Springer, pp. 189–209.

Algorithm 3 Pebble dike status degradation at each game round.

```

if nb_slices > 1 then
  nb_slices ← nb_slices – NB_SLICES_LOST_PER_ROUND
  if nb_slices ≤ NB_SLICES_PEBBLE_DEGRADED then
    status ← STATUS_DEGRADED
  else if nb_slices ≤ NB_SLICES_PEBBLE_MEDIUM then
    status ← STATUS_MEDIUM
  end if
end if

```

- Becu, N., Crookall, D., 2020. Companion modelling and participatory simulation: a glimpse.
- Bertin, X., Li, K., Roland, A., Zhang, Y.J., Breilh, J.F., Chaumillon, E., 2014. A modeling-based analysis of the flooding associated with xynthia, central bay of biscay. *Coastal Engineering* 94, 80–89.
- Briot, J.P., de Azevedo Irving, M., Vasconcelos Filho, J.E., de Melo, G.M., Alvarez, I., Sordoni, A., de Lucena, C.J.P., 2017. Participatory management of protected areas for biodiversity conservation and social inclusion: Experience of the simparc multi-agent-based serious game, in: *Multi-Agent-Based Simulations Applied to Biological and Environmental Systems*. IGI Global, pp. 295–332.
- Dawson, R.J., Peppe, R., Wang, M., 2011. An agent-based model for risk-based flood incident management. *Natural hazards* 59, 167–189.
- Duke, R.D., 2014. Gaming: the future’s language. *wbv*.
- Dukes, R.L., Fowler, S.M., DeKoven, B., 2011. R. garry shirts: Simulation gaming exemplar. *Simulation & Gaming* 42, 545–570.
- Étienne, M., 2013. *Companion modelling: a participatory approach to support sustainable development*. Springer Science & Business Media.
- Filatova, T., Mulder, J.P., van der Veen, A., 2011. Coastal risk management: how to motivate individual economic decisions to lower flood risk? *Ocean & coastal management* 54, 164–172.
- Gray, S., Paolisso, M., Jordan, R., Gray, S., 2016. *Environmental Modeling with Stakeholders*. Springer.
- Gray, S., Voinov, A., Paolisso, M., Jordan, R., BenDor, T., Bommel, P., Glynn, P., Hedelin, B., Hubacek, K., Introne, J., et al., 2018. Purpose, processes, partnerships, and products: four ps to advance participatory socio-environmental modeling. *Ecological Applications* 28, 46–61.
- Grimm, V., Railsback, S.F., Vincenot, C.E., Berger, U., Gallagher, C., DeAngelis, D.L., Edmonds, B., Ge, J., Giske, J., Groeneveld, J., et al., 2020. The odd protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation* 23.
- Hedelin, B., Gray, S., Woehlke, S., BenDor, T.K., Singer, A., Jordan, R., Zellner, M., Giabbanelli, P., Glynn, P., Jenni, K., et al., 2021. What’s left before participatory modeling can fully support real-world environmental planning processes: A case study review. *Environmental Modelling & Software* 143, 105073.
- Jaxa-Rozen, M., Kwakkel, J.H., Bloemendal, M., 2019. A coupled simulation architecture for agent-based/geohydrological modelling with netlogo and modflow. *Environmental Modelling & Software* 115, 19–37.
- Khoury, M., Gibson, M.J., Savic, D., Chen, A.S., Vamvakieridou-Lyroudia, L., Langford, H., Wigley, S., 2018. A serious game designed to explore and understand the complexities of flood mitigation options in urban–rural catchments. *Water* 10, 1885.
- Klabbers, J.H., 2009. *The magic circle: Principles of gaming & simulation*. Brill Sense.
- Laatabi, A., Becu, N., Marilleau, N., Pignon-Mussaud, C., Amalric, M., Bertin, X., Anselme, B., Beck, E., 2020. Mapping and describing geospatial data to generalize complex models: The case of littoSIM-GEN. *International Journal of Geospatial and Environmental Research* 7, 6. URL: <https://dc.uwm.edu/ijger/vol17/iss1/6/>.
- Laatabi, A., Marilleau, N., Nguyen-Huu, T., Hbid, H., Ait Babram, M., 2018. Odd+2d: An odd based protocol for mapping data to empirical abms. *Journal of Artificial Societies and Social Simulation* 21, 9. URL: <http://jasss.soc.surrey.ac.uk/21/2/9.html>, doi:10.18564/jasss.3646.
- Le Page, C., Bazile, D., Becu, N., Bommel, P., Bousquet, F., Etienne, M., Mathevet, R., Souchere, V., Trébuil, G., Weber, J., 2013. Agent-based modelling and simulation applied to environmental management, in: *Simulating social complexity*. Springer, pp. 499–540.
- Mayer, I., Bekebrede, G., Warmelink, H., Zhou, Q., 2014. A brief methodology for researching and evaluating serious games and game-based learning, in: *Psychology, pedagogy, and assessment in serious games*. IGI Global, pp. 357–393.
- Mayer, I.S., 2009. The gaming of policy and the politics of gaming: A review. *Simulation & Gaming* 40, 825–862.
- Michael, D.R., Chen, S.L., 2005. *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade.
- Neal, J., Schumann, G., Fewtrell, T., Budimir, M., Bates, P., Mason, D., 2011. Evaluating a new lisflood-fp formulation with data from the summer 2007 floods in tewkesbury, uk. *Journal of Flood Risk Management* 4, 88–95.
- Neumann, B., Vafeidis, A.T., Zimmermann, J., Nicholls, R.J., 2015. Future coastal population growth and exposure to sea-level rise and coastal flooding—a global assessment. *PloS one* 10, e0118571.
- Ornetsmüller, C., Castella, J.C., Verburg, P., 2018. A multiscale gaming approach to understand farmer’s decision making in the boom of maize cultivation in laos. *Ecology and Society* 23.
- Paprotny, D., Sebastian, A., Morales-Nápoles, O., Jonkman, S.N., 2018. Trends in flood losses in europe over the past 150 years. *Nature communications* 9, 1985.
- Rocle, N., Rey-Valette, H., Bertrand, F., Becu, N., Long, N., Bazart, C., Vye, D., Meur-Ferec, C., Beck, E., Amalric, M., et al., 2020. Paving the way to coastal adaptation pathways: An interdisciplinary approach based on territorial archetypes. *Environmental Science & Policy* 110, 34–45.

- Sawyer, B., Rejeski, D., 2002. Serious games: Improving public policy through game-based learning and simulation.
- Solinska-Nowak, A., Magnuszewski, P., Curl, M., French, A., Keating, A., Mochizuki, J., Liu, W., Mechler, R., Kulakowska, M., Jarzabek, L., 2018. An overview of serious games for disaster risk management—prospects and limitations for informing actions to arrest increasing risk. *International journal of disaster risk reduction* 31, 1013–1029.
- Sorace, S., Quercia, E., La Mattina, E., Patrikakis, C.Z., Bacon, L., Loukas, G., Mackinnon, L., 2018. Serious games: An attractive approach to improve awareness. *Community-Oriented Policing and Technological Innovations* , 1–9.
- Stadsklev, R., 1975. *Handbook of Simulation Gaming in Social Education: Textbook. volume 1.* Institute of Higher Education Research and Services, University of Alabama.
- Sterling, E., Zellner, M., Jenni, K., Leong, K., Glynn, P., BenDor, T., Bommel, P., Hubacek, K., Jetter, A., Jordan, R., et al., 2019. Try, try again: Lessons learned from success and failure in participatory modeling. *Elementa* 7.
- Sun, Z., Lorscheid, I., Millington, J.D., Lauf, S., Magliocca, N.R., Groeneveld, J., Balbi, S., Nolzen, H., Müller, B., Schulze, J., et al., 2016. Simple or complicated agent-based models? a complicated issue. *Environmental Modelling & Software* 86, 56–67.
- Taillandier, F., Adam, C., 2018. Games ready to use: A serious game for teaching natural risk management. *Simulation & Gaming* 49, 441–470.
- Taillandier, P., Gaudou, B., Grignard, A., Huynh, Q.N., Marilleau, N., Caillou, P., Philippon, D., Drogoul, A., 2019. Building, composing and experimenting complex spatial models with the gama platform. *GeoInformatica* 23, 299–322.
- Tanwattana, P., Toyoda, Y., 2018. Contributions of gaming simulation in building community-based disaster risk management applying japanese case to flood prone communities in thailand upstream area. *International journal of disaster risk reduction* 27, 199–213.
- Terti, G., Ruin, I., Kalas, M., Láng, I., Cangros i Alonso, A., Sabbatini, T., Lorini, V., 2019. Anycare: a role-playing game to investigate crisis decision-making and communication challenges in weather-related hazards. *Natural Hazards & Earth System Sciences* 19.
- Van Bruggen, A., Nikolic, I., Kwakkel, J., 2019. Modeling with stakeholders for transformative change. *Sustainability* 11, 825.
- Van Hardeveld, H., Driessen, P.P., Schot, P.P., Wassen, M.J., 2019. How interactive simulations can improve the support of environmental management—lessons from the dutch peatlands. *Environmental Modelling & Software* 119, 135–146.
- Voinov, A., Bousquet, F., 2010. Modelling with stakeholders. *Environmental Modelling & Software* 25, 1268–1281.
- Voinov, A., Jenni, K., Gray, S., Kolagani, N., Glynn, P.D., Bommel, P., Prell, C., Zellner, M., Paolisso, M., Jordan, R., et al., 2018. Tools and methods in participatory modeling: Selecting the right tool for the job. *Environmental Modelling & Software* 109, 232–255.
- Van der Wal, M.M., De Kraker, J., Kroeze, C., Kirschner, P.A., Valkering, P., 2016. Can computer models be used for social learning? a serious game in water management. *Environmental modelling & software* 75, 119–132.
- Wong, P.P., Losada, I.J., Gattuso, J.P., Hinkel, J., Khattabi, A., McInnes, K.L., Saito, Y., Sallenger, A., et al., 2014. Coastal systems and low-lying areas. *Climate change* 2104, 361–409.
- Zerger, A., Wealands, S., 2004. Beyond modelling: linking models with gis for flood risk management. *Natural Hazards* 33, 191–208.