A Modular Approach for ABM/LMM Models: Specification of Reusable Building Blocks Centred on the Economic Concepts of WTA and WTP

Eric Innocenti *, Dominique Prunetti *, Marielle Delhom †, Corinne Idda *
*UMR CNRS 6240, LISA Research Laboratory, University of Corsica Pasquale Paoli, Corte, France
e-mail:{eric.innocenti | prunetti_d, | idda_c}@univ-corse.fr
†UMR CNRS 6134, SPE Research Laboratory, University of Corsica Pasquale Paoli, Corte, France
e-mail: marielle.delhom@univ-corse.fr

***Abstract*—Agent-Based Models focusing on land markets provide a computational framework to simulate socio-economic dynamics in land and real-estate markets. In this paper, we introduce the 5-Step Simulation Iterative Modelling Process method, an iterative, five-step modelling and simulation decomposition approach specifically designed to structure the development of Agent-Based Land Market Models. We describe how implementing reusable building blocks—conceptual, computational, and executable—enhances modularity and fosters reusability of both theoretical concepts and software code. An illustrative example, applied to land and real-estate markets in Corsica, concretely demonstrates the application of the method and the creation of these reusable components. The integration of the economic concepts of Willingness To Accept and Willingness To Pay into the design of an Agent-Based Land Market Model exemplifies how these building blocks contribute to market dynamics formation. Finally, we highlight the potential of this approach to strengthen computational simulation, support socio-economic analysis, and promote sustainable land management.**

Keywords-Agent-Based Modelling; Land and Real Estate Markets; Reusable Building Blocks; Socio-Economic Dynamics; Sustainable Land Management.

I. Introduction

This article presents an extended version of the international conference paper titled "*Reusable Building Blocks for Agent-Based Simulations: Towards a Method for Composing and Building ABM/LUCC*", which was presented during SIMUL 2024 [1]. Computer simulations of *Agent-Based Models of Land Use and Cover Change* (*ABM/LUCC*) constitute a relatively recent interdisciplinary research domain within computational economics, positioned at the intersection of computer science, software engineering, economics, geography, and social sciences. *ABM/LUCC* models represent an increasingly specialised branch of *Agent-Based Models* (*ABM*), originating from *Multi-Agent Systems* (*MAS*) within the broader field of *Distributed Artificial Intelligence* (*DAI*). In this work, we argue that modelling *complex socio-economic systems* must follow a progressive and iterative approach, characterised by an iterative *composition-decomposition* process, extending from the formulation of a *conceptual model*—defining key study elements—to the development of executable, *modular*, and *adaptive* computer code. The *5-Step Simplified Iterative Modelling Process* (*5-SSIMP*) guides this modelling procedure from initial concept formulation to the execution of computer-based simulation experiments. The process is structured around producing intermediate models (Phase A), categorised as abstract and concrete, with the ultimate objective of delivering a modular and adaptive executable model. This approach involves specifying *Reusable Building Blocks* (*RBB*) [1], [2]. [2] advocates structuring agent-based models around atomic, clearly documented, and context-specific *RBB*s. Building upon their approach, our work expands this concept by specifically defining and integrating economic building blocks centred on *Willingness To Accept* (*WTA*) and *Willingness To Pay* (*WTP*). These model entities integrate theoretical concepts (conceptual model), computational components (computational model), and executable code modules (executable model). Such entities must be designed generically to ensure their reusability across diverse computational simulation projects. Ultimately, this methodology aims to reduce development costs and time, ensure code reliability, and facilitate integration with artificial intelligence systems. The *modelling phase A* of the *5-SSIMP* specifically targets the creation of three types of *RBB*: conceptual-RBB (*conRBB*), computational-RBB (*comRBB*), and executable-RBB (*exeRBB*). In this article, we illustrate how *RBB* are produced according to phase A of the *5-SSIMP* methodology. We detail the construction of *RBB* using the generic economic concepts of *WTA* and *WTP*, commonly employed in *ABM/LUCC* of the *Agent-Based Land Market Model (ABM/LMM)* type. These represent specialised socio-economic *MAS* designed to study and analyse spatial-economic phenomena driven by agent interactions within market frameworks [3]–[5]. The *WTA* and *WTP* concepts are indispensable for characterising agent decision-making processes within *ABM/LMM*.

In the second part, we clarify the theoretical and methodological framework that structures this interdisciplinary research. We present in detail the generic economic concepts of *WTA* and *WTP*, specifying their roles and integration within the conceptual model.

The third part provides a concise overview of the *5-SSIMP* method, briefly revisiting its primary phases and steps. We define the three types of *RBB*s—conceptual (*conRBB*), computational (*comRBB*), and executable (*exeRBB*)—, which must be specified during phase A of the modelling process. We also describe the fundamental characteristics of these blocks, along with their structural roles in progressively developing intermediate abstract and concrete models.

In the fourth part, we practically illustrate this methodological approach through an application to land and real estate markets in Corsica. We demonstrate how the economic concepts of

WTA and *WTP* can be specified and implemented as *RBB*s within an *ABM/LMM*, tailored to the unique socio-economic characteristics of this tourist region. This case study also serves as an illustrative example of modular structuring within the computational model, highlighting methodological advantages associated with the use of objects, design patterns and *RBB*s.

The fifth part engages, in an in-depth discussion of the essential roles played by *WTP* and *WTA* concepts within agent decision-making processes in the *ABM/LMM*. Using preliminary simulation results, we illustrate how this modular approach, based on *RBB*s, effectively explores simulation data, addressing specific sustainable land management challenges in Corsica.

Finally, the sixth part concludes by summarising the major contributions of this work and outlines future research perspectives. We envisage further refinements of the Corsican *ABM/LMM* and its integration into dedicated software Python infrastructure, facilitating broader generalisation and reusability across larger different territorial and thematic contexts.

II. Theoretical foundation

A. Willingness To Pay and Willingness To Accept

First introduced over a century ago for the former concept [6], *WTP* and *WTA* are fundamental in economics for understanding consumer and producer decision-making processes. *WTP* is defined as the maximum amount a consumer is willing to pay to acquire a good or service. Conversely, *WTA* represents the minimum amount an individual is willing to accept to give up a good or forego a service. These concepts are grounded in the *theory of subjective value*, where the value of a good is determined by the perceived utility it provides. *WTP* is often employed in market studies to estimate potential demand and set optimal prices. It can be measured using various methods, such as *contingent valuation* or *conjoint analysis*, though these techniques are sometimes biased by strategic or hypothetical factors [7]. In theory, *WTP* and *WTA* should be equivalent according to the *standard model of neoclassical economics*. However, in practice, they often differ, with this gap known as the *endowment effect*. This effect highlights that individuals tend to place a higher value on what they already own. The discrepancy is attributed to psychological factors such as loss aversion, perceptions of relative value, and cognitive biases [8], [9]. Practically, *WTP* helps determine consumer surplus, i.e., the difference between what a consumer is willing to pay and the actual price. Meanwhile, *WTA* assists in estimating producer surplus, i.e., the difference between the price received and the minimum acceptable amount. These indicators are essential for evaluating economic welfare, whether in contexts such as differential pricing, environmental valuation, or public policy analysis [10]. Ultimately, *WTP* and *WTA* are crucial tools for specifying agent decision-making processes and pricing behaviours in an *ABM/LMM*. During the *simulation data analysis phase*, the comparison of *WTP* and *WTA* serves as a *key indicator*, reflecting the potential for a convergence of interests between buyers and sellers within the markets of an *ABM/LMM*. These concepts thus provide a rigorous framework for studying socio-economic dynamics in spatially explicit environments, such as those modelled in *ABM/LMM*s.

B. Context of the Case Study

The primary empirical application of our work concerns the land and housing markets in Corsica, which, due to their dynamics and the scarcity of land available for residential use, are characterised by significant pressure on prices and conflicts over land use. Corsica is a geographically defined territory, surrounded by the Mediterranean Sea. It is one of the least densely populated Mediterranean regions, with the majority of its population concentrated along the coastline, with only the town of Corte, located inland, standing as an exception due to the presence of the island's sole university within its territory [11]. Since the 1970s, Corsica has experienced the development of a tourism-based economy, which today represents the island's main economic sector. This tourism expansion has led to the construction of infrastructure for mass tourism, as well as a significant rise in the number of second homes, a trend that continues to grow. By 2015, there were over 90,000 second homes in Corsica, accounting for more than 37% of the regional housing stock [12], a proportion nearly four times higher than the national average. Consequently, in cities such as Ajaccio and Bastia, as well as other tourist areas, securing affordable accommodation has become increasingly difficult. These findings raise important questions, which we aim to address through the development of a virtual computational simulation environment based on an *ABM/LMM* model. Our objective is, first, to determine how investments in properties intended for short-term tourist rentals affect the availability of housing at affordable prices for local residents. Secondly, we seek to examine, from a public policy perspective, which measures could be implemented to address these issues (e.g., zoning regulations, investment taxes, or the creation of a permanent resident status).

III. Method

A. 5-Step Simulation Iterative Modelling Process

To structure this computational modelling and simulation work within an interdisciplinary context, we employ the iterative composition-decomposition method known as the *5-Step Simulation Iterative Modelling Process* (*5-SSIMP*)[1]. This approach ensures a logical and coherent progression, from the concepts formulated in intermediate models to the validation of results obtained through simulation experiments. Its iterative nature allows for continuous adjustments to the modelling components. As illustrated in Figure 1, the *5-SSIMP* method consists of two main phases (*A.-Modelling, B.-Simulation*), and five stages (*A.1-Conceptualisation*, *A.2-Integration*, *A.3-Implementation*, *B.1-Experimentation*, and *B.2-Data Analysis*. This ensures a smooth transition from conceptual theory to operational simulation. The principle of the *5-SSIMP* method is illustrated in Figure 2. During the *A.1-Conceptualisation* stage, modellers simplify the real-world system by formulating a *conceptual model*, identifying key components and their relationships. In the *A.2-Integration stage*,

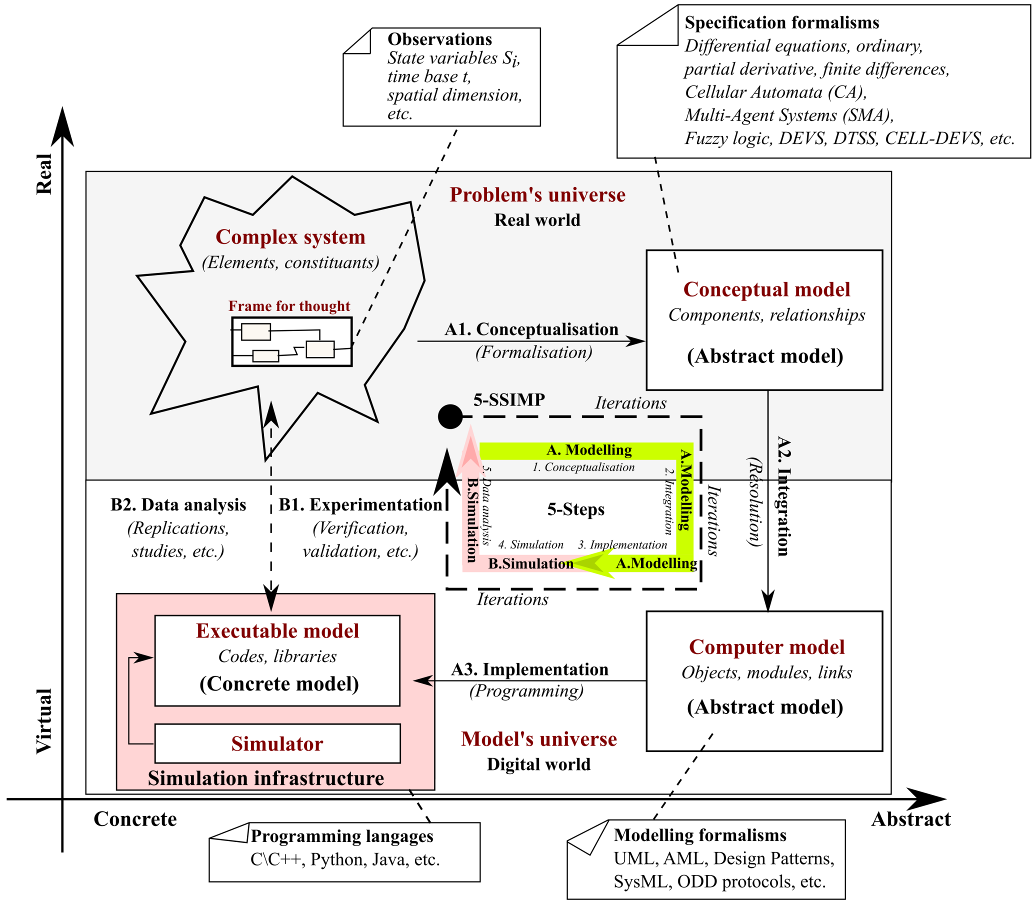

Figure 1. The *5-SSIMP* method is an iterative composition-decomposition approach for the modelling and simulation process.

modellers translate the *conceptual model* into a *computational model*, defining *objects* (classes, interfaces), *modules* (groups of objects), and appropriate *links* (relationships). In the *A.3-Implementation* stage, the *computational model* is transformed into an *executable model* (executable code modules). This process is based on *Object-Oriented Programming* (*OOP*) and *Generic Programming* (*Template Programming*) techniques [13], as well as *Design Patterns* (*DP*) [14]. During the *B-Simulation* phase, in the *B.1-Experimentation* stage, modellers must validate and verify the *executable model* through rigorous testing to ensure the accuracy and reliability of the simulated data obtained, comparing it, where possible, with real-world data. In the *B.2-Data Analysis* stage, the executable model is finalised for routine practical use, ensuring that it remains reliable, robust, and efficient over time and across evolving conditions. As illustrated in Figure 1, the *RBB* components of the *executable model* are continuously refined and validated through the iterative cycle of continuous adjustments within the *5-SSIMP* method. These iterations enable feedback loops and the integration of new elements derived from field survey data and questionnaire responses. The *5-SSIMP* method effectively structures the iterative phases of development, testing, and refinement involved in constructing *ABM/LUCC* models.

B. Reusable Building Blocks

TABLE I
LINKS BETWEEN INTERMEDIATE MODELS (ABSTRACT AND CONCRETE) AND REUSABLE BUILDING BLOCKS (RBB) IN THE 5-SSIMP METHOD.

Intermediate model	RBB	Specification Tools
A1. Conceptual	*conRBB*	Mathematics, ODEs, PDEs, etc. Formalisms, DTSS, DEVS, etc.
A2. Computational	*comRBB*	Standards, norms, UML, AML, ODD, SysML, etc.
A3. Executable	*exeRBB*	OOP, Template programming, DP, etc.

The *5-SSIMP* method is based on the creation of *Reusable Building Blocks* (*RBB*), which ensure the modularity and reusability of various modelling components. Each of these components is designed independently, facilitating its evolution from conceptual specification to implementation in an object-oriented programming language. Within this framework, generic building blocks (*conRBB*, *comRBB*, *exeRBB*) are defined, integrated, and implemented in a way that allows them to be reused and adapted for other simulation projects. The *5-SSIMP* method significantly reduces development time and associated costs by optimising collaboration among team

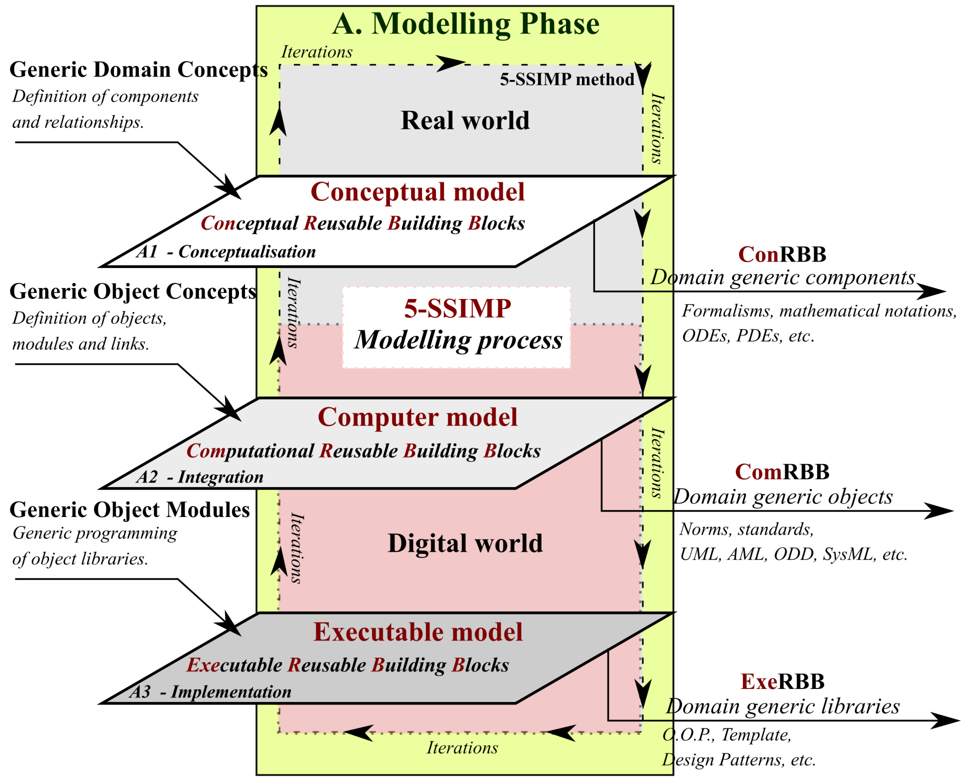

Figure 2. The core concept of *Reusable Building Blocks* (*RBB*) in the *A-Modelling* phase of the *5-SSIMP* method.

members involved in the modelling process. In this paper, the concepts of *WTP* and *WTA* serve as representative examples to illustrate the relevance and efficiency of the *5-SSIMP* method. These concepts are specified in the form of *RBB* to structure their modular integration into the intermediate models of the *5-SSIMP* method. Furthermore, this approach facilitates their future application in various other contexts related to land and housing markets. The production of *RBB* is embedded within the *A.1-Conceptualisation*, *A.2-Integration*, and *A.3-Implementation* stages of the *A–Modelling* phase in the *5-SSIMP* method. During these stages, modellers identify and isolate key elements characterising the socio-economic system under study, defining corresponding intermediate models, both *abstract* and *concrete*. These models incorporate both generic and specific characteristics, which must be characterised separately. Elements with generic properties are grouped into *Reusable Building Blocks* (*RBB*) within the intermediate models, serving to modularly organise structures and functionalities that can be reused in other *ABM/LUCC* models. In this context, each *RBB* represents a generic aspect of the intermediate model (either *abstract* or *concrete*), such as generic agent behaviours, generic social interactions, or generic economic dynamics. The structuring of intermediate models into *Reusable Building Blocks* (*RBB*) facilitates the iterative process by enabling:

- updates, replacements, maintenance, and evolution of the executable model (*modularity*);
- the creation and updating of code libraries, allowing validated code to be shared and reused across different simulation projects (*reusability*);
- collaboration between modellers in economics and computer science, supporting the development and evolution of new generic modules tailored to specific requirements.

C. Conceptual Reusable Building Block

A *Conceptual Reusable Building Block* (*conRBB*) is a type of *RBB* produced in the *A.1-Conceptualisation* stage of the *5-SSIMP* method. *conRBB* are formulated using specification formalisms (e.g., rules, notations, formal languages, etc.), mathematical notations, ordinary differential equations (ODEs), and system specifications (e.g., DEVS [15]). These *conRBB* structure and organise generic key components and their relationships within abstract conceptual models in a clear and structured manner.

D. Computational Reusable Building Block

A *Computational Reusable Building Block* (*comRBB*) is another type of *RBB*, produced in the *A.2 Integration* stage of the *5-SSIMP* process. *comRBB* ensure the consistency, standardisation, and reusability of generic objects or object groups across different simulation projects. They can be reused in various fields of study. *comRBB* should adhere to recognised computational norms and standards, such as *Unified Modelling Language* (*UML*) [16], *Agent Modelling Language* (*AML*) [17], *Overview*, *Design concepts*, and *Details* (*ODD*) [18], or *Systems Modeling Language* (*SysML*) [19], among others. To achieve this, modellers rely on *Object-Oriented Programming* (*OOP*) [20], *Template Programming* [13], and *Design Patterns* (*DP*) [14]. Recent studies demonstrate that *DP* significantly enhance the *modularity* and *reusability* of models in computational simulations [21]–[23].

E. Executable Reusable Building Block

An *Executable Reusable Building Block* (*exeRBB*) is the type of *RBB* produced in the *A.3 Implementation* stage of the *5-SSIMP* modelling process. These building blocks consolidate the generic code components of the executable model. This involves implementing, within a programming language, the object groupings defined in the computational model, incorporating *Templates* and *Design Patterns* where applicable. *exeRBB* are essential for computational replications carried out in the *B.1-Simulation* and *B.2-Data Analysis* stages of the *5-SSIMP method*. They consist of reusable code modules that can potentially be adapted for use in other computational simulation projects. These software libraries are designed generically, providing a significant advantage in computational simulation projects that require high productivity and software reliability.

IV. Implementation

A. Conceptual Model

The *ABM/LMM* presented as an example in this study comprises two economic markets: a *Land Market* (*LM*) and a *Housing Market* (*HM*). This work builds upon the research we initiated in 2021, focusing on the economic study and analysis of a tourist region, Corsica [24]. In this section, we provide a concrete example of modelling the decision-making processes and price formation mechanisms of agents within an *ABM/LMM*, following the implementation of the *5-SSIMP* method. We also present the *RBB*s that we have developed to represent the *WTA* and *WTP* formulated within the *ABM/LMM*.

1) Land Market - LM: In the *LM*, household agents who own land interact as sellers (`LndHse-agents - LM-sellers`) with real estate developer agents acting as buyers (`Dev-agents - LM-buyers`).

The concepts of *WTP* and *WTA* are employed to model the decision-making behaviours of economic agents involved in both the *LM* and the *HM*. These concepts serve to specify the price-setting decision processes and behavioural mechanisms of the agents within the *ABM/LMM* we are constructing. More specifically, in the *LM*, *WTA* represents the minimum amount that a `LndHse-agent` household owner is willing to accept to sell a plot of land on the *LM*. This *WTA* corresponds to the average past sale price of similar plots with identical characteristics. Conversely, *WTP* represents the maximum amount that a `Dev-agent` is willing to pay to acquire land on the *LM*. This *WTP* is determined by the `Dev-agent` real estate developer, based on the anticipated selling price of the house they intend to build on the land, while seeking to achieve a target profit margin on the total cost of the house. Interactions between the `LndHse-agents` as sellers and the `Dev-agents` as buyers in the *LM* are illustrated in Figure 3.

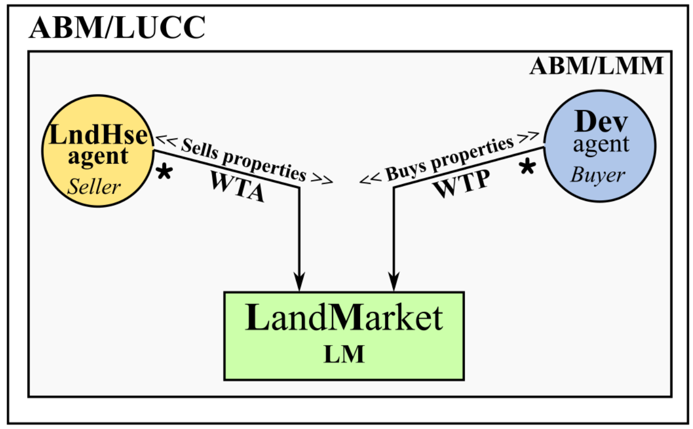

Figure 3. Interactions between household landowner agents (`LndHse-agents`) as sellers and real estate developer agents (`Dev-agents`) as buyers in the ABM/LMM (*LM*).

2) Housing Market - HM: In the *HM*, `Dev-agents` (real estate developers) resell land with newly built houses. Their *WTA* corresponds to the amount required to generate a profit margin on the total costs associated with producing the housing unit (construction costs and the price paid for the land). Within this *HM*, there are two categories of buyers. `Hse-agents` are Households seeking accommodation, whose *WTP* is determined through the maximisation of a quasi-linear utility function under budget constraints.

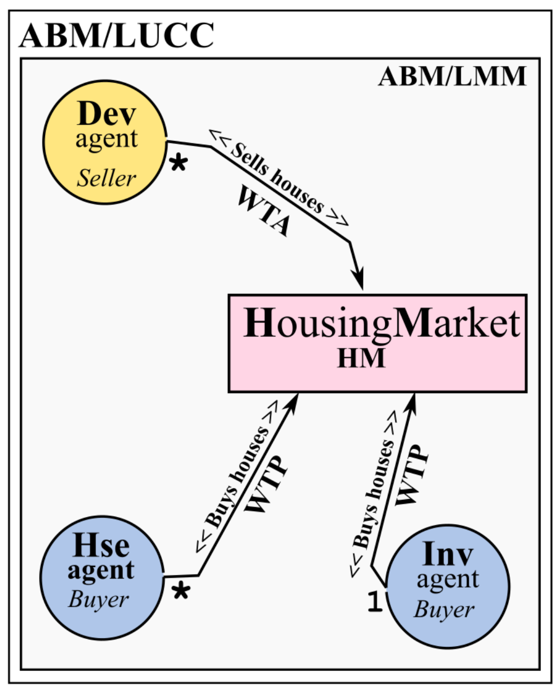

Figure 4. Interactions between real estate developer agents as sellers (`Dev-agents`), household agents as buyers (`Hse-agents`), and the single investor agent as a buyer (`Inv-agent`) in the ABM/LMM (HM).

`Inv-agent` is the single representative investor who purchases houses and rents them as tourist accommodations.

For the `Inv-agent`, *WTP* is based on the opportunity cost of investing in a tourist rental property versus an alternative financial investment of the same amount. In other words, this *WTP* follows an objective criterion whereby the expected return from purchasing a house at a given price must be equal to or greater than the return from an equivalent investment in the financial market.

3) Agent's Interaction: In terms of interactions, `Dev-agents` purchase land assets (plots) sold by `LndHse-agents`, with the objective of constructing a house. For the sake of conceptual model simplification, each real estate developer is specialised in constructing only one category of house $m = \{1, ..., M\}$ (e.g., detached house, chalet, bungalow, etc.), which is subsequently sold on the *HM*. In the *HM*, real estate developer agents who previously acted as buyers in the *LM* become sellers (`Dev-agents - HM-Sellers`) and interact with household agents as buyers (`Hse-agents - HM-buyers`), as well as with the single representative investor agent (`Inv-agent - HM-buyer`). Whether in the *LM* or the *HM*, the price formation processes of agents in the model are based on the key economic concepts of *WTA* and *WTP*. These two fundamental concepts are represented as *Conceptual Building Blocks* (*conRBB*) within *ABM/LMM* models.

4) Price Formation in the LM: Household landowner agents (`LndHse-agents`) make their decisions based on their *WTA*, represented as a *Conceptual Reusable Building Block* (*conRBB*). This decision depends on various factors, such as opportunity costs, market conditions, and the financial objectives of the agents. The *WTA* of `LndHse-agents` in the *LM* is given by Equation (1).

$$WTA_{LndHse}\left(z_i\right) = P_{t-1}\left(z_i\right) \quad (1)$$

The vector z_i represents the characteristics of the land asset being sold, while the past average sale price $P_{t-1}(z_i)$, calculated using a spatial regression model, is estimated based on the historical sale prices of similar plots with comparable characteristics z_i. Interviews conducted with real estate professionals (three estate agents and one property developer) revealed that an implicit rule prevails across various segments of the Corsican housing market: sellers and buyers tend to maintain a negotiation margin when finalising a transaction. For sellers, this margin manifests as an attempt to secure a transaction price above their reserve price, whereas for buyers, it reflects an effort to negotiate a transaction price below their maximum *WTP*.

a) $Askprice_{\text{Lwd-Agent}}$*:* According to our interviewees, this negotiation margin is approximately 7% of the sale price. We denote this margin a ψ, a value specific to each individual, which plays a role in the negotiation process between seller agents (`LndHse-agents`) and buyer agents (`Dev-agents`) during land transactions. It is incorporated into the calculation of the asking price set by sellers in the *LM*, $Askprice_{\textit{Lwd-Agent}}$, as defined in Equation (2).

$$Askprice_{\textit{Lwd-Agent}} = (1 + \psi_{\textit{Lwd-Agent}}) \times WTA_{\textit{Lwd-Agent}} \quad (2)$$

b) WTP - `Dev-agents`*:* The purchasing decisions of real estate developer agents (`Dev-agents`) in the *LM* are based on their *WTP*, represented as a *conRBB*, as defined in Equation (3).

$$WTP_m\left(z_i\right) = \frac{P_{H,i,t-1}\left(z_i, LivA_m\right)}{1 + \pi_m} - C_{LivA}\left(z_i\right) LivA_m \quad (3)$$

$LivA_m$ represents the surface area of a house belonging to category m. π_m denotes the anticipated profit margin of the real estate developer agent. $P_{H,i,t-1}$ is the average past price of a house built on a plot of land, both of which have equivalent characteristics i corresponding to z_i. $C_{LivA}\left(z_i\right)$ represents the construction cost per unit of surface area required to build the house.

c) $Bidprice_{\texttt{Dev-agent}}$*:* Taking into account the negotiation margin ψ_m of a `Dev-agent` and in an attempt to secure a transaction price lower than their *WTP*, the purchase price ($Bidprice_m$) is given by Equation (4).

$$Bidprice_m = (1 - \psi_m) WTP_m(z_i) \quad (4)$$

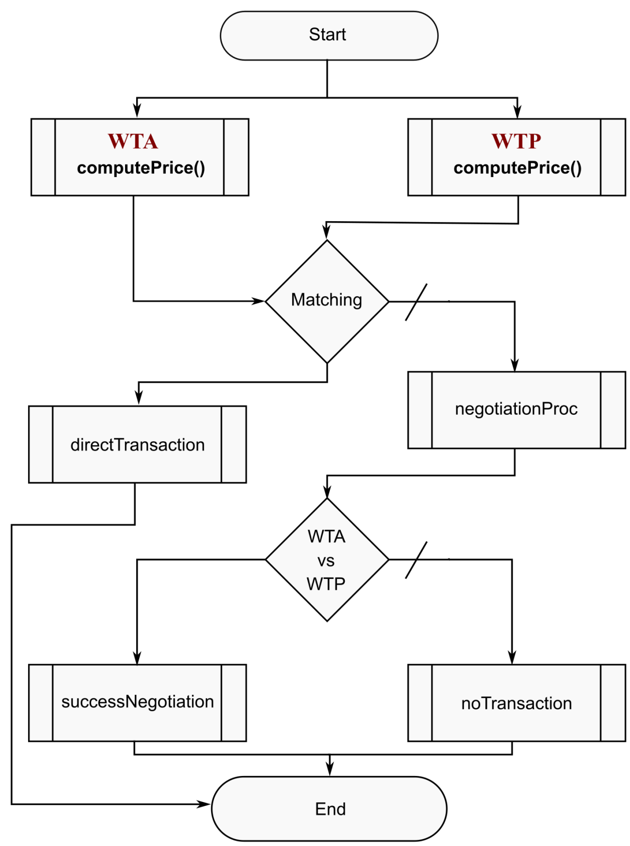

Figure 5. Flowchart of the Negotiation and Transaction Process in the LM Algorithm.

5) Negotiation and Transaction in the LM: Once all buyer agents in the *LM* have set their bid prices ($Bidprice_{m-\texttt{Dev-agent}}$) and seller agents have determined their asking prices ($Askprice_{-\texttt{Lwd-Agent}}$), the matching and exchange mechanism is executed. `Dev-agent` submit offers for available plots of land while considering their construction capacity constraints, whereas `LndHse-agents` evaluate the received offers. Seller `LndHse-agents` review all proposals and identify the highest bid ($Bidprice_{m-\texttt{Dev-agent}}$). Matching then occurs between the `Dev-agent` buyer who submits the highest offer and the corresponding `LndHse-agent` seller. Once this matching is completed, the negotiation and transaction process can begin.

a) Direct Transaction Process - DTP: If the highest $Bidprice_{max}$, submitted by the `Dev-agent`, is greater than or equal to the asking price ($Askprice_{\texttt{Lwd-Agent}}$) set by the household landowner agent, the transaction is completed immediately without negotiation. In this case, the transaction price ($\bar{P}(z_i)$) is set at the asking price ($Askprice_{\texttt{Lwd-Agent}}$) determined by the selling household agent.

b) Transaction Process with Negotiation - TDN:

$$\bar{P}\left(z_i\right) = WTA_{Hse}(z_i) + \frac{ND_i}{\chi}\left[WTP_{Max} - WTA_{Hse}\left(z_i\right)\right] \tag{5}$$

If the asking price ($Askprice_{\texttt{LndHse-agent}}$) set by the household landowner agent is strictly higher than the maximum bid price $Bidprice_{max}$ offered by the real estate developer agent (`Dev-agent`), a negotiation process may be initiated, provided that the buyer's *WTP* (`Dev-agent`) is greater than or equal to the seller's *WTA* (`LndHse-agent`). The transaction price $\bar{P}(z_i)$ is determined by Equation (5). ND_i represents the number of expressed bids for the property, while χ corresponds to the total number of `Dev-agents` for which $n_{m'} \geq 0$, with $m\prime = \{1, ..., \chi\}$. In Equation (5), the distribution of the difference between the buyer's *maximum WTP* WTP_{max} and the seller's *WTA* $WTA_{Hse}(z_i)$ depends on market conditions and the relative bargaining power of both parties. Thus, the higher the number of expressed bids (ND_i) for a property, the stronger the seller's bargaining power, leading to an increase in the transaction price. Conversely, when the number of expressed bids is low, the buyer's bargaining power increases, resulting in a lower transaction price.

c) No Transaction: Finally, if the asking price ($Askprice_{\texttt{LndHse-agent}}$) set by the household landowner agent strictly exceeds the maximum bid price ($Bidprice_{max-\texttt{Dev-agent}}$) and the *WTA* ($WTA_{LndHse}$) of the landowning household is greater than the *WTP* (WTP_m) of the `Dev-agents`, no transaction can take place for this plot.

6) Price Formation in the HM: The *HM* represents the interaction between "*real estate developer agents*" `Dev-agents`, who become sellers of their properties, "*household agents*" `Hse-agents` acting as buyers, and a single "*investor agent*" buyer `Inv-agent (HM-buyer)`.

In this market, the *WTA* (*ConRBB*) of the *Dev*-agents selling their houses is given by Equation (6):

$$WTA_{i,m}\left(z_i, LivA_m\right) = \left(1+\pi_m\right)\left(\overline{P_i}\left(z_i\right) + C_{LivA} LivA_m\right) \tag{6}$$

where $\overline{P_i}\left(z_i\right)$ represents the price paid for the parcel i with characteristics z_i.

A negotiation margin similar to that used by sellers and buyers in the *LM* is also applied by economic agents in the *HM*. Sellers initially propose an asking price above their *WTA*, attempting to achieve their individual negotiation margin. Similarly, buyers initially propose a bid price below their *WTP*, based on their individual negotiation margin.

To avoid unnecessarily complicating the exposition of the conceptual model, the explicit relationships linking sellers' *Ask prices* to their *WTA* and buyers' *Bid prices* to their *WTP* in the *HM* are not presented here.

Following the maximisation of a quasi-linear utility function under budget constraints, the *WTP* for household agents seeking accommodation is defined by Equation (7):

$$\mathrm{WTP}_c\left(\eta\right) = \frac{\left(1+\delta_c\right)\left[\left(1+\delta_c\right)^T - 1\right]}{\delta_c}\theta_c\left(\eta\right) \tag{7}$$

where δ_c is the household agent's discount rate, T is the average duration of a real estate loan, and $\theta_c\left(\eta\right)$ is the function defined by Equation (8), based on [25]:

$$\theta_c\left(\eta\right) = \frac{YD_c\left(V_c^{Max}\right)^2}{b^2 + \left(V_c^{Max}\right)^2} \tag{8}$$

where:

- YD_c represents the monetary amount allocated by the household to housing each period;
- b represents the slope of their bid function.

7) HM: Purchase Process of the `Inv-agent`*:* The purchasing decision-making process of the `Inv-agent` buyer relies on the *WTP* (*conRBB*) described by Equation (9):

$$\mathrm{WTP}_I\left(\eta\right) = \frac{1-\left(1+r\right)^{-T}}{r\left(1-\rho\right)\left(1+r\right)^{-T} + rT - 1 + \left(1+r\right)^{-T}}\gamma\zeta\varphi\left(\eta\right) \tag{9}$$

where:

- η: vector of the house characteristics;
- $r \in [0, 1]$: interest rate in the financial market;
- $\gamma \in [0, 1]$: coefficient of net rental yield (net of maintenance costs);
- ζ: number of days in the tourist season;
- $\varphi\left(\eta\right)$: average daily revenue of a tourist residence with similar characteristics;
- $\rho \in [0, 1]$: residual value coefficient of the house after T years.

The *conRBB* of the `Inv-agent` involves a limited budget, denoted as Ω, representing the total investment in the *HM* over a given period. The agent can choose between two investment types: a financial market investment or a real estate investment.

8) Negotiation and Transaction in the HM: The negotiation and transaction algorithm of the *HM* is similar to that of the *LM*. To avoid unnecessarily complicating our presentation, and due to the similarity of these two algorithms—particularly their reliance on the *WTA* and *WTP* of seller and buyer agents—this algorithm is not detailed here.

For a more detailed description of the *conRBB* used in this article's conceptual model, the interested reader can refer to [26] and [24].

B. Computer Model

In this section, we present the computational integration of the *ABM/LMM* and its *computational Reusable Building Blocks* (*comRBB*) related to the generic concepts of *WTP* and *WTA*, previously formulated as conceptual *Reusable Building Blocks* (*conRBB*).

1) Design patterns: In computational modelling, the number of objects can rapidly become very large, especially when multiple organisational levels are involved. In such contexts, modellers must address the challenge of defining numerous objects and relationships.

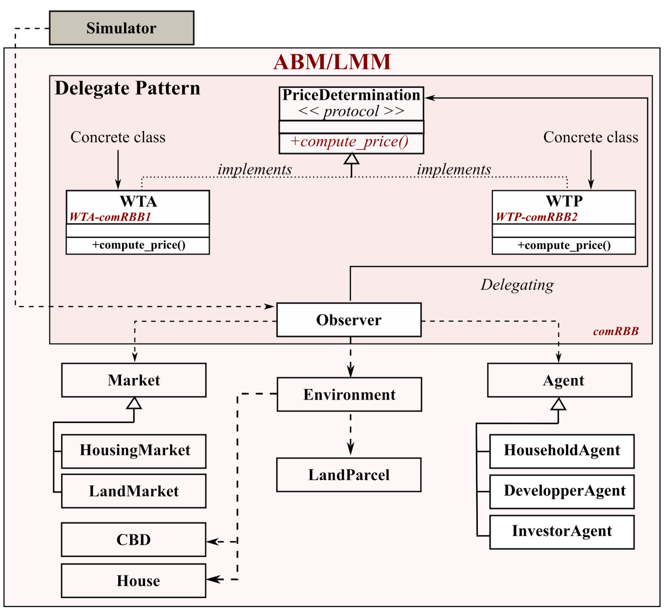

Figure 6. In the computer model, *comRBB* related to *WTP* and *WTA* are implemented following the *Delegation* design pattern.

The use of *design patterns* helps manage this complexity, allowing structurally well-defined and organised *comRBB* to be implemented and improved independently. The concept of design patterns originates from the seminal work *A Pattern Language: Towns, Buildings, Construction* [27] by architects *Christopher Alexander*, *Sara Ishikawa*, and *Murray Silverstein*. During the 1970s, these authors defined a pattern as: "*Each pattern describes a problem, which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice*". This statement defines a design pattern as a recurring problem with its associated solution, a specific context, an *architecture*, and the expression of the associated *generic solution*. Subsequently, the concept of design patterns was introduced into computer science during the 1970s with object-oriented programming (OOP), notably through the renowned *Model-View-Controller* (MVC) pattern [28]. The MVC pattern was initially proposed by *Tryve Reenskaug* as a generic solution for complex data-handling problems [29]. Towards the late 1980s, the contributions of *Reid Smith* also supported their use in computer science [30], alongside the influential article by *Kent Beck* and *Ward Cunningham*, which proposed adapting design patterns to OOP through five interface-design compositions [31].

TABLE II
CHARACTERISTICS OF A *DESIGN PATTERN* IN COMPUTER SCIENCE ACCORDING TO THE "*GoF*" FORMALISM [32].

Vocabulary Element	Meaning
Name	Name of the design pattern.
Problem	Description of the problem addressed by the *Design Pattern*.
Initial context	The context to which the pattern applies.
Forces	Description of situations where the *Design Pattern* is applicable.
Solution	Components of the solution and their relationships.
Examples	Examples of application.
Consequences	Description of how the *Design Pattern* achieves its goal.
Logic	Description of the logic implemented.
Related patterns	Closely related *Design Patterns* referenced here.

However, it was not until 1995 that the use of *Design Patterns* became widespread within OOP, particularly due to the influential work of the "*GoF*" (Gang of Four)—*Erich Gamma*, *Richard Helm*, *Ralph Johnson*, and *John Vlissides* [32]. This book standardised precise vocabulary and formalisation for design patterns, now commonly adopted in computer science literature (see Table II). Since then, *Design Patterns* have progressively promoted unambiguous modular and reusable code expression, facilitating best practices and expertise dissemination among designers and developers [28], [33], [34].

Recognising the valuable knowledge encapsulated in expert-driven design patterns, we propose using them to integrate the *comRBB* of the ABM/LMM into the computer model, ultimately aiming at implementing modular and reusable *exeRBB* in the executable model.

Additionally, using *Design Patterns* in computational simulation promotes reproducibility by precisely and completely describing generic object-based model elements. Consequently, the implemented *exeRBB* code is easily understandable and replicable by the scientific community. Using established *Design Pattern* names helps modellers clearly explain, justify, and communicate the structure of their *comRBB* and *exeRBB*, ensuring reproducibility in executable implementations from the conceptual stage onwards.

2) NetLogo Design Pattern: In an interdisciplinary scientific context, the *NetLogo* simulation platform is preferred for initial prototyping of *reusable building blocks* (*RBB*) within agent-based *ABM/LMM* models.

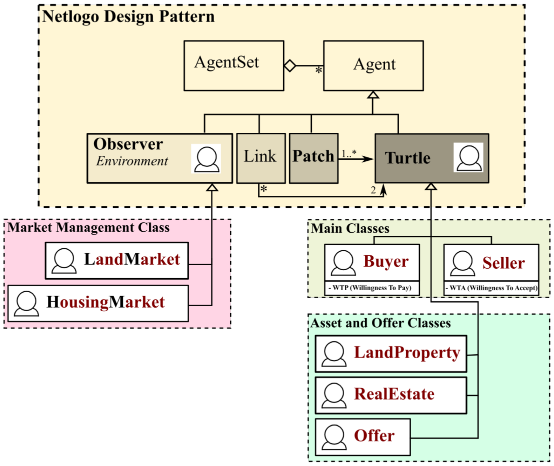

Figure 7. The *NetLogo Design Pattern* allows rapid construction of an *ABM/LMM* prototype using only four generic types of components (*comRBB*): an omniscient *observer*, *patches*, *turtles* (mobile agents), and relational links (*links*) between agents [1].

NetLogo simplifies modelling by providing four generic types of reusable components (*comRBB*): an omniscient observer (*Observer*), spatially explicit fixed cells (*Patches*), mobile agents (*Turtles*), and social connections (*Links*). These components significantly facilitate the initial modelling stage of *ABM/LMM* by enabling rapid prototyping and systematic testing of generic behaviours implemented within executable reusable building blocks (*exeRBB*). This structuring of the *NetLogo* simulation platform (cf. Figure 7), as proposed by *Seth Tisue* [35], facilitates modelling through clear modular organisation, thereby promoting rapid prototyping and verification of *RBB*s. Nevertheless, we restrict the use of *NetLogo* to the initial prototyping phase of *RBB*, given its limitations when dealing with intensive real-world data processing tasks, for which higher-performance object-oriented languages, such as Python or C++, are more suitable.

3) Delegation Design Pattern: In computational integration phases of an *ABM/LMM*, the description of numerous objects and relationships can quickly increase complexity, exacerbated by extensive inheritance use. However, inheritance creates strong coupling among objects, limiting modularity, evolvability, and reusability. To ensure optimal computer model structuring and avoid such restrictive couplings, an alternative delegation-based approach is necessary. The *Delegation Design Pattern* precisely addresses this challenge, allowing an object to delegate a specific part of its behaviour to another specialised object without direct inheritance, thus reducing strong coupling. In this work, the architectural choice of delegation, outlined in Figure 6, enables the creation of modular, maintainable, and easily evolvable *comRBB* for *WTA* and *WTP*. The characteristics of the delegation design pattern are summarised in Table III. In Figure 6, the price determination classes `WTA` and `WTP` both implement the abstract method `compute_price()` defined in the common interface *PriceDetermination*. The abstract class `PriceDetermination` specifies a common interface protocol for different price determination strategies within the *ABM/LMM* model. Concrete classes `WTA` and `WTP` represent specialised objects responsible for implementing their specific strategies to determine asset prices, thereby encapsulating the model's economic concepts of *WTA* and *WTP*. This approach notably prevents coupling between objects of the final executable model and those from the simulation infrastructure. The resulting object-oriented code is significantly more modular and reusable; dependencies between objects of the executable model and those of the simulation infrastructure are thus substantially reduced [36]. Thus, it is essential to design the components of the computer model with design patterns in mind, to ensure modularity and reusability.

TABLE III
CHARACTERISTICS OF THE *DELEGATION* DESIGN PATTERN ACCORDING TO THE *GoF* FORMALISM.

Vocabulary item	**Meaning**
Name	*Delegation*
Problem	Ensuring that an object can transfer or delegate part of its behaviour to another object without resorting to inheritance.
Initial context	When a component must execute an operation, but the processing varies depending on context. It is desirable to avoid strong coupling or a rigid class hierarchy.
Forces	- Reduction of coupling between classes. - Easier code reuse. - Improved maintainability.
Solution	Introduce a *delegate* object responsible for the desired behaviour. The primary class redirects the call to this object, allowing the behaviour to be specialised or modified dynamically.
Examples	- Implementation of multiple behaviours in agent-based simulations. - Management of different calculation strategies (pricing, behaviour, etc.) via a separate object.
Consequences	- Enables combining functionalities without multiplying subclasses. - Makes the system more flexible and modular. - Simplifies future modifications or extensions.
Logic	A primary object holds a reference to a secondary object (the delegate). Calls related to specific behaviour are transferred to this delegate rather than being implemented directly.
Related patterns	*Strategy* (for algorithm specialisation), *Adapter* (for interface compatibility), *Decorator* (for dynamic addition of responsibilities).

C. Executable Model

The *delegation* design pattern is implemented in accordance with the organisation of the *WTA-comRBB* and *WTP-comRBB* components of the computational model shown in Figure 6. The code is structured into abstract classes (`PriceDetermination`) to facilitate the implementation of the *delegation* design pattern and to implement the *WTA-exeRBB* and *WTP-exeRBB*. As a reminder, *abstract classes* are fundamental tools in object-oriented programming, allowing code to be structured in a modular and reusable way. They define a common interface that multiple subclasses can implement, thus ensuring a consistent organisation of the code and facilitating its extension. Their use in the executable model promotes separation of responsibilities, with each subclass being responsible for implementing its own behaviours while maintaining compatibility with the overall model. Abstract classes also enable the addition of new functionalities without altering the existing object structure, thereby enhancing the flexibility and scalability of the model. In this perspective, as illustrated in Figure 6, the creation of agents within the computational model follows a modular approach, externalising decision-making strategies related to *WTP* and *WTA* into specialised object codes (*exeRBB*). This design avoids direct implementation of price formation calculations within agent classes, delegating this responsibility to dedicated objects. The implemented *exeRBB* thus allows agents to adopt various pricing strategies without altering their internal structure. Furthermore, this approach facilitates the addition or modification of strategies without requiring a complete overhaul of agent class codes, while ensuring the reusability of these modules in other modelling contexts.

Listing 1. The abstract class (`PriceDetermination`) implements the *delegation* design pattern in *Python*.

```
from abc import ABC, abstractmethod
class PriceDetermination(ABC):
 """
 Abstract class defining a common interface for price
 decision strategies related to WTA and WTP.
 Abstract methods:
  compute_price(): Method to calculate the price based on
  the agent's attributes and asset characteristics.
  """
 def __init__(self,delegate, **kwargs):
  """
  Abstract constructor that accepts a variable number of
  keywords.
  arguments to provide flexibility for derived
  strategies.
  Args:
     **kwargs:  Dictionary containing named arguments
     specific to derived classes (here WTP or WTA RBBs).
  """
  self.__kwargs = kwargs
  self.delegate = delegate
  # Additional initialisation specific to subclasses can
  be added here.
  @abstractmethod
  def compute_price(self):
   """
   Abstract method to calculate price (i.e., WTP or WTA).
   Returns:
   float: Computed price according to the decision-making
   strategy implemented by subclasses.
   """
   pass
```

In order to implement this object-oriented architecture, the `PriceDetermination` module presented in Listing 1 is defined as an abstract class serving as a common interface for economic agents' decision-making processes. This interface requires implementation of the abstract method `compute_price()`, which models the price-formation calculation according to market dynamics. Thus, the `PriceDetermination` object in Listing 1 provides a common basis for implementing decision-making strategies related to WTA and WTP.

Listing 2. Implementation of the *exeRBB* `WTA` in *Python*.

```
from PriceDetermination import PriceDetermination
import math
class WTA(PriceDetermination):
 """
 Represents the minimum acceptable price (WTA) for
 selling an asset. This class computes the WTA using
 a regression model where the natural logarithm
 of the price is a function of various parameters
 and regression coefficients.
 """
 def __init__(self, **kwargs):
  """
  Initialise a WTA instance.
  Keyword arguments include:
  - past_land_price: Historical land price.
  - regressCoeff_k0 to regressCoeff_kN:
  Regression coefficients for the model.
  - parcel_surface: Surface area of the parcel.
  - beach_distance: Distance to the beach.
  - cbd_distance: Distance to the central business
  district.
  - sea_viewIndex: Index representing the quality of
  the sea view.
  - currentTime: The current time variable.
  """
  # Initialise the superclass with the provided keyword ↪
     arguments.
  super().__init__(self, **kwargs)
  # Dynamically extract regression coefficient keys
  from kwargs.
  # This ensures that the number of coefficients is
  # determined by the input.
  coeff_keys = sorted([key for key in kwargs if key.↪
     startswith("regressCoeff_")], key=lambda x:int(x.↪
     split("_k")[1]))
  # Create a dictionary mapping 'k{i}' to its
  # corresponding coefficient.
  self.coeffs = {f'k{int(key.split("_k")[1])}': kwargs.↪
     get(key, 0.0) for key in coeff_keys}
  # Retrieve additional parameters with default values if
  # not provided.
  self.past_land_price = kwargs.get('past_land_price'↪
     ,0.0)
  self.parcel_surface = kwargs.get('parcel_surface',0.0)
  self.beach_distance = kwargs.get('beach_distance',0.0)
  self.cbd_distance = kwargs.get('cbd_distance',0.0)
  self.sea_viewIndex = kwargs.get('sea_viewIndex',0.0)
  self.currentTime = kwargs.get('currentTime',0.0)

  def compute_price(self):
   """
   Compute the minimum acceptable price (WTA) using
   the regression formula: ln(price) = k0 +
   k1 * log(parcel_surface) +
   k2 * (log(parcel_surface))^2 +
   k3 / beach_distance + k4 / cbd_distance +
   k5 * sea_viewIndex + k6 * currentTime
   Returns:
   The computed price, obtained as the exponential of:
   ln(price).
   In the event of a division by zero, 0.0 is returned.
   """
   try:
    # Compute the natural logarithm of the parcel's
    # surface area.
    log_parcel = math.log(self.parcel_surface)
```

```
    # Calculate the natural logarithm of the price based
    # on the regression model.
    ln_price = (
    self.coeffs.get('k0', 0.0) +
    self.coeffs.get('k1', 0.0) * log_parcel +
    self.coeffs.get('k2', 0.0) * (log_parcel ** 2) +
    self.coeffs.get('k3', 0.0) / self.beach_distance +
    self.coeffs.get('k4', 0.0) / self.cbd_distance +
    self.coeffs.get('k5', 0.0) * self.sea_viewIndex +
    self.coeffs.get('k6', 0.0) * self.currentTime
    )
    # Return the computed price by exponentiating
    # ln(price).
    return math.exp(ln_price)
  except ZeroDivisionError as e:
    # Error message in case a division by zero occurs.
    print("Error computing WTA: division by zero.", e)
    return 0.0
```

In Listing 2, the code excerpt demonstrates how the *exeRBB* `PriceDecision` specifically implements the `compute_price()` method, which is used to calculate the minimum acceptable price for a seller agent's *WTA* in the *ABM/LUCC*.

Similarly, the `WTP` class shown in Figure 6 implements the *WTP* strategy for buyer agents by determining the maximum price a buyer agent is prepared to offer. This amount is calculated based on the agent's available budget, anticipated selling price, estimated construction costs, and desired profit margin. The modules `WTA` and `WTP` *exeRBB* inherit from the `PriceDetermination` class and implement their own computational logic through the methods `compute_price()`, tailored to the specific strategies of seller and buyer agents, respectively. Thus, agents within the model can instantiate a `WTA` or `WTP` object and delegate the price calculation to it, without directly handling the complexity of these decisions. This organisation, structured into increasingly specialised implementations of `PriceDetermination`, promotes a modular, scalable, and reusable architecture, ensuring a clear separation of responsibilities and enhanced flexibility in adapting the model to various economic scenarios. The *exeRBB*s thus defined not only contribute to the modularity and clarity of the executable model's code but also facilitate its adaptation and reuse in other *ABM/LMM* modelling contexts. This enhances the modularity, reusability, and flexibility of the model by allowing the dynamic customisation of behaviours within the *exeRBB*s.

V. Discussion and preliminary results

Owing to its modular architecture based on *RBB*s, the developed *ABM/LMM* exhibits high flexibility, enabling extensive testing, fine-tuning, and comparative evaluation of various simulation scenarios. Within this context, initial simulation experiments focus on analysing the behaviour of previously developed *RBB*s. To this end, several land and real estate policy tests scenarios were considered to evaluate how the *WTA* and *WTP RBB*s influence price formation and market dynamics. Examples of these test scenarios, preliminarily validated through prototyping in *NetLogo* [35], for the validation of *RBB*s [1], are presented and discussed in detail in Table IV.

Each scenario is based on varying configurations of the *WTA* and *WTP RBB*s, enabling a detailed analysis of economic,

TABLE IV
Example scenarios of land and real estate policies simulated in Corsica.

Scenario	Description
BAU	This baseline scenario extrapolates current trends, serving as a comparative reference to evaluate the effects of more interventionist policies.
BTRI	This scenario explores the impact of a total prohibition of new tourist rental investments, reflecting drastic yet plausible policy measures to curb tourist accommodation saturation.
TTRI	A tax varying from 0.01 to 0.5 is imposed on revenues derived from tourist rental investments. This measure aims to reduce profitability, limiting investment expansion and influencing *LM* dynamics.
CZ	This scenario mandates a minimum compulsory distance (between 1 and 50 distance units) from the coastline for any new tourist rental investment. The goal is to protect coastal zones by reducing development concentration near sensitive habitats.
CBD-Z	Similar to coastal zoning, this policy imposes a mandatory minimum distance (1 to 50 distance units) from the city centre (*CBD*) for new tourist rental investments. Its objective is to redistribute tourist accommodations across the territory, alleviating urban centre pressures and encouraging development in less exploited areas.

TABLE V
Fixed Parameters

Input	Value
Percentage of low revenue households	54%
Percentage of middle revenue households	20%
Percentage of high revenue households	26%
Percentage of households that own their home	52%
Percentage of households that own a buildable parcel of land	87%
Share of households income spent on housing	0.3
Time Horizon (year)	20
Transport cost (€/km)	0.5404
Number of buildable parcels of land	2,500
Number of resident households	5,625
Number of developers	4
Number of Rent days	220
Slope of the bid function	0.8

regulatory, and behavioural parameters influencing agents' price-setting processes. These variations allow a nuanced examination of public policy impacts on land and real estate dynamics, particularly regarding price setting, market pressures, and housing accessibility. Table V summarises the fixed parameters used throughout simulation experiments, which remain constant to provide reference points.

Table VI presents parameters randomly assigned values following a uniform distribution, introducing variability into the simulations.

Data sources for these simulation experiments include the detailed *PERVAL* database provided by the Chamber of Notaries, offering various land and property market indicators, and *AirDNA*, which supplies data on short-term holiday rentals to estimate average daily revenue from tourist properties. Additional datasets from the *INSEE* (French National Institute of Statistics and Economic Studies) provide demographic and economic parameters. Supplementary data sources include references [37] for construction sector data and [38] for transportation costs to the CBD, alongside valuable insights from discussions with real estate agents and property developers.

As an example, we present a selection of preliminary simulation results obtained using the parameters listed in Table VII.

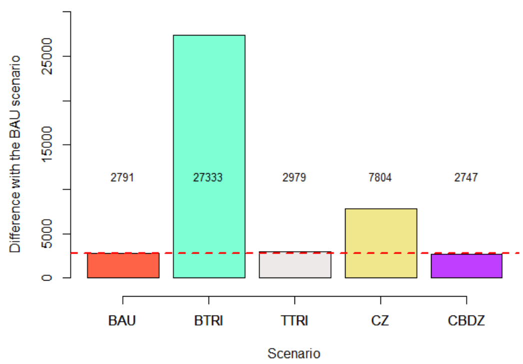

Figure 8. Number of developer bankruptcies by scenario.

Analysing these results, we extensively examine the impacts of agent interactions on Corsica's land and real estate dynamics. Each simulated policy scenario illuminates potential public policy effects on urban development and *HM* evolution. Detailed examination of developer bankruptcies, market price variations, distances to strategic interest points such as the *Central Business District* (*CBD*) and coastline, as well as the *Mean Sea View Index*, provided particularly insightful findings. These indicators offer valuable interpretations of the economic, social, and environmental consequences of each policy. Specifically, we detail developer bankruptcies and average price trends observed across both studied markets for each simulated scenario.

TABLE VI
RANDOM UNIFORM VALUE PARAMETERS

Input	Value
Land Area (m^2)	$[500, 2500[$
Sea view Index	$[0.75, 23.25[$
Disposable budget of low revenue households	$[6200, 23200[$
Disposable budget of middle revenue households	$[10000, 40000[$
Disposable budget of high revenue households	$[11500, 71500[$
Disposable budget of Investor	$[900000, 1100000[$
Interest rate	$[0.02, 0.04[$
Discount rate	$[0.01, 0.11[$
Leeway	$[0.05, 0.09[$
Net return of rental	$[0.65, 0.75[$
House loss value	$[0.08, 0.12[$
Margin rate	$[0.02, 0.024[$
Cost Parameter (€/m^2)	$[750, 1000[$
Max Building sites	$[1, 3[$
Number of Bedrooms	$[1, 6[$
House surface : $a + b(\text{N. of Bedrooms} - 1)$	$a : [18, 31[, a : [24, 33[$
Building time : $8 + BTSup$	$BTSup : [0, 4[$

TABLE VII
EXPERIMENTAL PARAMETERS AND DATA SOURCES.

Aspect	Details
Initial	Experiments start with an initial seed of 0 to ensure reproducibility.
Replications	Simulations ran for a total of 2,500 replications over 40 time steps, with each step representing a semester.
Environment	The world is a grid of 501×501 = 251,001 patches, with the *CBD* located at coordinates (20, 50).
Data	The *ABM/LUCC* heavily relies on real-world data for accuracy, including results from four spatial regressions and a hedonic model categorising households by income classes.

A. Developer Bankruptcies

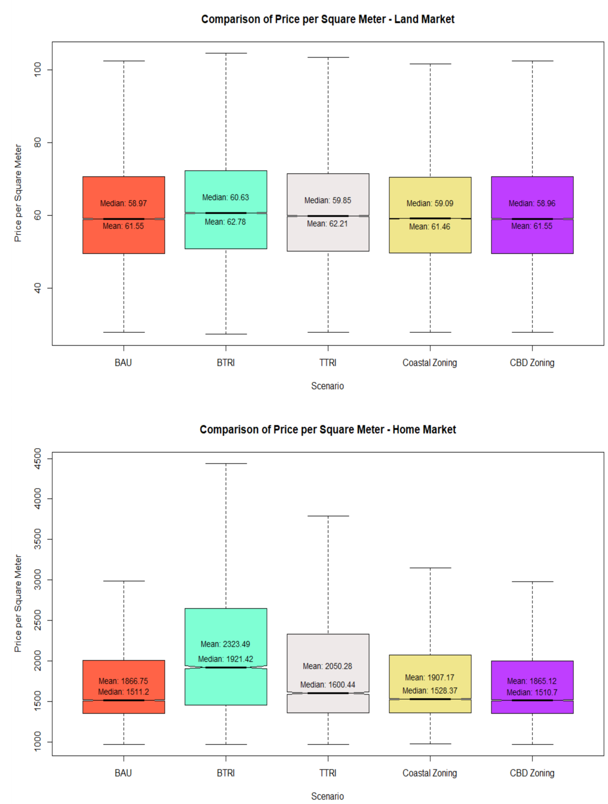

Figure 9. Dispersion of price per square meter for *LM* and *HM*.

Figure 8 reveals significant variation in developer bankruptcies across scenarios. Developer bankruptcies totalled 2,791 under the *BAU* scenario, escalating sharply to 27,333 under the *BTRI* scenario. Scenarios *TTRI*, *CZ*, and *CBD-Z* demonstrate relatively stable or mildly favourable conditions for developers, highlighting the delicate balance required between stringent regulation and market economic sustainability.

B. Market Price Variations

Figure 9 demonstrates *LM* resilience, displaying only minor price fluctuations across scenarios.

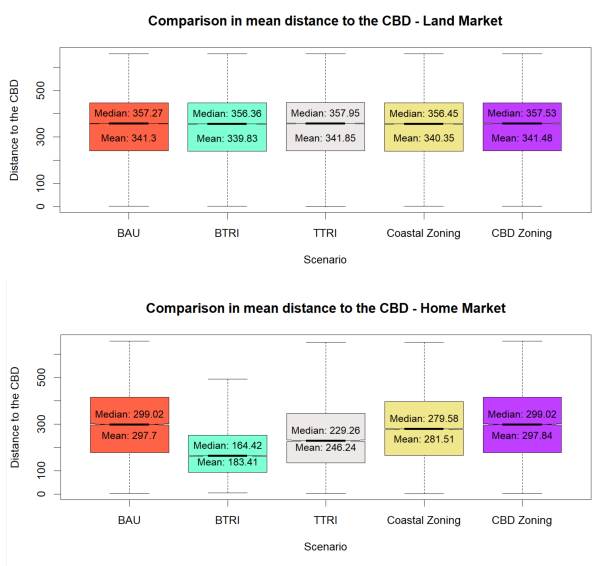

Figure 10. Comparison with the mean distance to the *CBD* in the *LM* and *HM* markets.

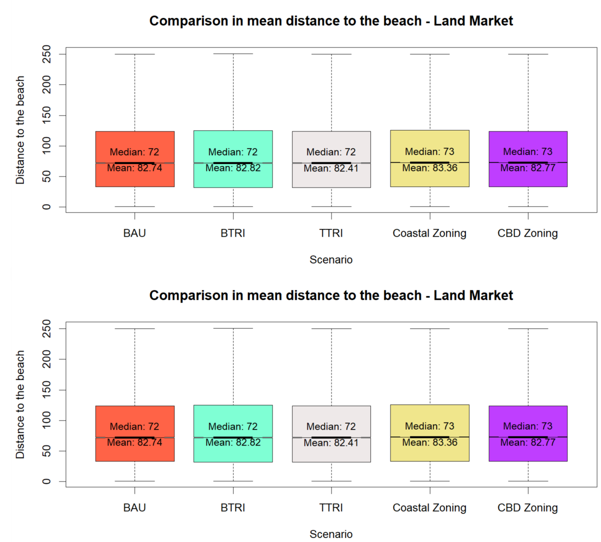

Figure 11. Comparison with the mean distance to the Beach in the *LM* and *HM* markets.

While the *LM* generally withstands simulated policies, the *HM* reacts significantly to restrictive scenarios, especially *BTRI*, with notably higher price increases. This pattern aligns with typical market responses under restrictive conditions, underscoring substantial impacts of such policies on supply-demand dynamics.

C. Influence of Central Business District (CBD)

Figure 10 assesses the scenarios' effects on average transaction distances to the CBD. The *BTRI* scenario notably induces a shift in investor-to-household purchasing, reducing average distances to the CBD, thus indicating household-driven spatial realignment towards urban centres.

Investor-driven submarkets show less sensitivity to CBD-based zoning policies, highlighting investors' resistance to such spatial regulation.

D. Distance to the Coastline

Figure 11 analyses average distance variations to the coastline across scenarios, revealing increased distances in the *CZ* scenario, especially among investors. Conversely, *BTRI* triggers decreased coastal distances in the *HM*, suggesting spatial substitution and a shift favouring household proximity to coastal areas. These observations illustrate the spatial behavioural impacts regulatory policies exert on economic agents, shaping territorial development dynamics.

VI. Conclusion and Future Work

In this article, we have presented the *5-SSIMP* modelling method, based on the design, integration, and implementation of *RBB*s. We specifically focused this study on the economic concepts of *WTA* and *WTP*. By applying the iterative composition-decomposition modelling method *5-SSIMP* to the development of an *ABM/LMM*, we demonstrated how these modular blocks (conceptual, computational, and executable) effectively structure price formation dynamics and economic agents' decision-making processes. Applying this method to the case study of *LM* and *HM* in Corsica, concretely illustrated the benefits brought by the use of the *Delegation Design Pattern* in the development of *exeRBB*s in Python. This architectural choice significantly improves the modularity, flexibility, and reusability of the executable model, while simplifying the integration of complex economic concepts such as *WTA* and *WTP*. The modular approach proposed here offers several key advantages: it reduces development time and associated costs; ensures enhanced robustness of the software code; and facilitates reproducibility of simulation experiments. Furthermore, it promotes effective collaboration between economists and software development specialists, enabling easier adaptation of models to different territorial contexts or new research objectives. In terms of future research, this work opens several promising avenues. Firstly, it would be valuable to extend experimentation by diversifying public policy scenarios further, aiming to better understand their impacts on socio-economic and environmental dynamics. Secondly, integrating *exeRBB*s as modules within a dedicated Python-based software infrastructure, would facilitate the construction, parametrisation, and simulation of these models across various geographical and thematic contexts. Lastly, this approach also opens promising interdisciplinary perspectives by fostering dialogue between economic and computer science modellers.

Our findings rely on calibrating the model to a single island—Corsica—which constitutes one of the limitations of our study. Addressing this constraint will be the focus of future work, notably by applying the model to other territories across the Mediterranean basin to assess its external validity.

References

[1] E. Innocenti, D. Prunetti, M. Delhom, and C. Idda, "Reusable Building Blocks for Agent-Based Simulations: Towards a Method for Composing and Building ABM/LUCC", in *The 16th International Conference on Advances in System Modeling and Simulation (SIMUL 2024)*, 2024, pp. 28–33.

[2] U. Berger *et al.*, "Towards Reusable Building Blocks for Agent-Based Modelling and Theory Development", *Environmental Modelling & Software*, vol. 175, pp. 1–12, 2024.

[3] D. C. Parker, "An Economic Perspective on Agent-Based Models of Land Use and Land Cover Change", in *The Oxford Handbook of Land Economics*, J. Duke and J. Wu, Eds., Oxford University Press, 2014, ch. 16, pp. 402–429.

[4] D. C. Parker, D. G. Brown, J. G. Polhill, P. J. Deadman, and S. M. Manson, "Illustrating a New "Conceptual Design Pattern" for Agent-Based Models of Land Use via Five Case Studies—the MR POTATOHEAD Framework", in *Agent-Based Modelling in Natural Resource Management*, Universidad de Valladolid, 2008.

[5] D. C. Parker *et al.*, "MR POTATOHEAD: Property Market Edition | Development of a Common Description Template and Code Base for Agent-Based Land Market Models", in *15th Annual Social Simulation Conference (SSC 2019)*, 2019.

[6] H. J. Davenport, "Proposed Modifications in Austrian Theory and Terminology", *The Quarterly Journal of Economics*, vol. 16, no. 3, pp. 355–384, 1902.

[7] J. Schmidt and T. H. A. Bijmolt, "Accurately Measuring Willingness to Pay for Consumer Goods: A Meta-Analysis of the Hypothetical Bias", *Journal of the Academy of Marketing Science*, vol. 48, pp. 499–518, 2020.

[8] D. Kahneman, J. L. Knetsch, and R. H. Thaler, "Experimental Tests of the Endowment Effect and the Coase Theorem", *Journal of Political Economy*, vol. 98, no. 6, pp. 1325–1348, 1990.

[9] W. L. Adamowicz, V. Bhardwaj, and B. Macnab, "Experiments on the Difference between Willingness to Pay and Willingness to Accept", *Land Economics*, vol. 69, no. 4, pp. 416–427, 1993.

[10] C. A. Vossler, S. Bergeron, M. Doyon, and D. Rondeau, "Revisiting the Gap between the Willingness to Pay and Willingness to Accept for Public Goods", *Journal of the Association of Environmental and Resource Economists*, vol. 10, no. 2, pp. 413–445, 2023.

[11] A. Huyssen and R. Malleville, "En Corse, 86 % de la Population Vit dans l'Aire d'Attraction d'une Ville", *Insee Flash Corse*, vol. 54, 2020, https://www.insee.fr/fr/statistiques/4808381.

[12] H. Touzani, "Des Résidences Principales et Secondaires en Forte Croissance", *Insee Flash Corse*, vol. 32, 2018, https://www.insee.fr/fr/statistiques/3571002.

[13] D. Vandevoorde and N. M. Josuttis, *C++ Templates: The Complete Guide*. Addison-Wesley Professional, 2002, ISBN: 978-0201734843.

[14] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns: Abstraction and Reuse of Object-Oriented Design", in *ECOOP'93—Object-Oriented Programming: 7th European Conference, Kaiserslautern, Germany*, Springer, 1993, pp. 406–431.

[15] M. J. Blas, S. Gonnet, and B. P. Zeigler, "Towards a Universal Representation of DEVS: A Metamodel-Based Definition of DEVS Formal Specification", in *2021 Annual Modeling and Simulation Conference (ANNSIM)*, IEEE, 2021, pp. 1–12.

[16] I. Jacobson, J. Booch, and G. Rumbaugh, *The Unified Modeling Language Reference Manual*. Addison-Wesley, 2021.

[17] R. Cervenka and I. Trencansky, *The Agent Modeling Language—AML: A Comprehensive Approach to Modeling Multi-Agent Systems*. Springer Science & Business Media, 2007.

[18] V. Grimm *et al.*, "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*, vol. 23, no. 2, 2020.

[19] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann, 2014.

[20] M. Aniche, J. Yoder, and F. Kon, "Current Challenges in Practical Object-Oriented Software Design", in *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, IEEE, 2019, pp. 113–116.

[21] M. Najdek, M. Paciorek, W. Turek, and A. Byrski, "Three New Design Patterns for Scalable Agent-Based Computing and Simulation", *Informatica*, vol. 35, no. 2, pp. 379–400, 2024.

[22] M. Rudolph, S. Kurz, and B. Rakitsch, "Hybrid Modeling Design Patterns", *Journal of Mathematics in Industry*, vol. 14, no. 1, p. 3, 2024.

[23] L. Serena, M. Marzolla, G. D'Angelo, and S. Ferretti, "Design Patterns for Multilevel Modeling and Simulation", in *2023 IEEE/ACM 27th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, IEEE, 2023, pp. 48–55.

[24] D. Prunetti *et al.*, "ABM/LUCC of a Complex economic system of land and home markets facing an intense residential development", in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2021, pp. 1–8.

[25] T. Filatova, D. C. Parker, and A. Van der Veen, "Agent-Based Urban Land Markets: Agent's Pricing Behavior, Land Prices and Urban Land Use Change", *Journal of Artificial Societies and Social Simulation*, vol. 12, no. 1, p. 3, 2009.

[26] E. Innocenti, C. Detotto, C. Idda, D. C. Parker, and D. Prunetti, "An Iterative Process to Construct an Interdisciplinary ABM Using MR POTATOHEAD: An Application to Housing Market Models in Touristic Areas", *Ecological Complexity*, vol. 44, p. 100 882, 2020.

[27] C. Alexander, *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.

[28] E. M. Buck and D. A. Yacktman, *Les Design Patterns de Cocoa*. Pearson Education France, 2010.

[29] T. M. H. Reenskaug, "User-Oriented Descriptions of Smalltalk Systems", *Byte*, vol. 6, no. 8, 1981.

[30] R. Smith, "Panel on Design Methodology", in *ACM SIGPLAN Notices*, ACM, vol. 23 (5), 1987, pp. 91–95.

[31] W. Cunningham and K. Beck, "Using Pattern Languages for Object-Oriented Programs", *Proceedings of OOPSLA*, vol. 87, 1987.

[32] R. Johnson, E. Gamma, R. Helm, and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", *Boston, Massachusetts: Addison-Wesley*, 1995.

[33] J. Hunt, *Scala Design Patterns—Patterns for Practical Reuse and Design*. Springer, 2016.

[34] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*. Wiley, 1996.

[35] S. Tisue and U. Wilensky, "NetLogo: A Simple Environment for Modeling Complexity", in *International Conference on Complex Systems*, Citeseer, vol. 21, 2004, pp. 16–21.

[36] E. M. Buck and D. A. Yacktman, *Cocoa Design Patterns für Mac und iPhone*. mitp Verlags GmbH & Co. KG, 2010.

[37] RF, "Plateforme Ouverte des Données Publiques Françaises", 2024, [Online]. Available: https://www.data.gouv.fr/ (visited on 05/27/2025).

[38] RF, "Impots.gouv.fr", 2024, [Online]. Available: https://www.impots.gouv.fr/ (visited on 05/27/2025).