

### 3.3.1 Les paramètres

#### 3.3.1.1 Différents points de vue sur la définition d'un paramètre

Au plus général, le nouveau petit Robert définit un paramètre en ces mots :

- « 1. MATH. Quantité à fixer librement, maintenue constante, dont dépend une fonction de variables indépendantes, une équation ou une expression mathématique. — Variable en fonction de laquelle on exprime chacune des variables d'une équation.
- 2. FIG. et DIDACT. Élément important dont la connaissance explicite les caractéristiques essentielles de l'ensemble d'une question.
- 3. PAR EXT. Élément nécessaire pour juger, évaluer, comprendre (qqch.). »

(ROBERT, REY-DEBOVE et REY 1993, Paramètre)

Seule la première définition correspond, à grands traits, à ce que l'on attend ici, mais elle est très généraliste, bien que ne correspondant pas pour autant à tous les usages du terme employés dans la littérature.

**En mathématiques, une définition univoque.** L'acceptation mathématiques d'un paramètre est sans doute celle qui souffre le moins d'ambiguïté : il s'agit des termes fixes d'une équation simple, par opposition aux variables qui en constituent les éléments qui seront amenés à évoluer. Par exemple, dans la formulation d'une fonction affine,  $f(x) = ax + b$ , la valeur de  $f(x)$  dépend de la variable  $x$  et des paramètres fixés  $a$  et  $b$ . Quelles que soient les valeurs empruntées par  $x$ , ces paramètres demeurent constants. Dans une famille d'équations plus complexes, par exemple le modèle de croissance logistique de population<sup>28</sup>, l'accroissement de population au cours du temps ( $\frac{dy}{dt}$ ) dépend de la valeur de la variable  $y$ , population à cet instant, ainsi que de deux paramètres,  $\alpha$ , le taux de croissance, et  $K$ , la « capacité d'accueil », c'est-à-dire un potentiel maximum de population vers laquelle tend – et ne peut donc dépasser – le système modélisé. La définition mathématique est donc assez universelle et convient à la quasi-totalité des systèmes d'équations<sup>29</sup>. Il convient toutefois de noter que la différence entre variable et paramètre est une affaire de point de vue, une inversion de perspective menant à échanger les paramètres et variables, tel que décrit dans l'exemple suivant :

« For the function  $f(x) = ax^2 + bx + c$ , could we learn something by leaving the expression in terms of the symbols  $a$ ,  $b$ , and  $c$  and seeing how  $f(x)$  depends on the parameters? Maybe we could fix  $x = 2$  and look how  $f(2)$  changes as we let  $a$  vary.

If we do such manipulations and look at how the output of a function depends on varying a parameter, then we are treating the function as though the parameter were a input variable. But that's OK, as the difference between variables and parameters is really just a matter of perspective. »

(NYKAMP 2015a)

28.  $\frac{dy}{dt} = \alpha y \times (1 - \frac{y}{K})$  d'après (VERHULST 1838)

29. A l'exclusion notable des systèmes d'équation paramétriques, où, à l'inverse, le terme de paramètre désigne alors les variables indépendantes.

On peut donc en retenir qu'en mathématiques, ce qui différencie la variable du paramètre est l'aspect fixe de ce dernier, au moins pendant la durée d'exécution d'une fonction :

« A parameter is a quantity that influences the output or behavior of a mathematical object but is viewed as being held constant. [...] »

Variables are viewed as changing while parameters typically either don't change or change more slowly. »

(NYKAMP 2015b)

- On propose donc une définition sans doute plus spécifique que celle du nouveau petit Robert, mais aussi plus tournée vers l'usage : un paramètre est une variable maintenue constante durant l'ensemble de l'utilisation d'une fonction.

**Une vision duale en statistiques.** En statistiques, quand bien même cette discipline fait un large usage des formalismes mathématiques, les paramètres recouvrent un ensemble assez différent : il s'agit d'une « grandeur mesurable qui permet de présenter de façon plus simple, plus abrégée les caractéristiques essentielles d'un ensemble statistique. » (IMBS et QUEMADA 1992, Paramètre, STAT. (calcul des probabilités))

« In statistics, the most common use of « parameter » is for a characteristic of a population, or of a distribution of scores, described by a statistic such as a mean or a standard deviation. For example, the mean (average) score of on the midterm exam in Psychology 201 is a parameter. It describes the population composed of all those who took the exam. »

(VOGT 1993, p. 164)

Ainsi, pour les statisticiens, la moyenne, l'écart-type ou encore le coefficient d'asymétrie sont des paramètres, que l'on symbolise alors au moyen de lettres grecques (VOGT 1993, ibid.). On peut toutefois y retrouver une logique commune avec les paramètres mathématiques quand on décrit une loi statistique avec ces valeurs, qui deviennent alors les éléments permettant de caractériser une variable. Ainsi, pour décrire les caractéristiques de la distribution théorique – normale – d'une variable  $X$ , on fera appel aux paramètres théoriques de cette loi que sont l'espérance<sup>30</sup> ( $\mu$ ) et l'écart-type ( $\sigma^2$ ) :  $X \hookrightarrow \mathcal{N}(\mu, \sigma^2)$ .

**Une absence de consensus en informatique.** Dans le domaine informatique, la définition est bien plus floue et inclusive que dans les champs décrits auparavant, sans doute en raison d'une hétérogénéité bien supérieure dans les pratiques et construits informatiques. Comme dans les fonctions mathématiques, les paramètres sont les arguments des fonctions informatiques. Ces fonctions recouvrent toutefois un ensemble extrêmement vaste, bien plus hétérogène que dans les domaines mentionnés ci-dessus. On parle ainsi de fonction<sup>31</sup> pour toute suite d'instructions informatique ayant pour vocation – ou

30. Il s'agit de la moyenne théorique. On réserve ainsi le terme de moyenne à une valeur calculée depuis des valeurs empiriques.

31. On utilisera aussi indistinctement les termes de procédure, de (sub)routine ou encore de méthode. Notons que chacun de ces mots a normalement un sens précis. Par exemple, une

pour capacité – à être répétée, ré-exécutée. Ces fonctions appliquent ainsi, le plus fréquemment, cette suite d'instruction sur des données en entrée, et produisent ainsi des données en sortie, résultantes du traitement effectué sur les entrées<sup>32</sup>. Certaines fonctions sont donc très simples, et peuvent s'apparenter à des fonctions mathématiques, telles que par exemple la fonction arrondi (`round()` dans sa version la plus courante). Cette dernière prend en entrée un nombre décimal, et retourne un nombre entier. Le nombre décimal passé en entrée est alors nommé paramètre. Notons que cette fonction accepte le plus souvent un second paramètre, sous forme d'un nombre entier, qui permet de définir le nombre de décimales (*digits*) que l'on souhaite conserver. Par exemple, l'arrondi de 2,551 à une décimale renverra la valeur 2,6. Il est important de noter qu'en informatique, les paramètres ont souvent une « valeur par défaut », c'est-à-dire une valeur qui sera utilisée si le paramètre n'est pas explicitement spécifié. Dans le cas de la fonction arrondi, si le premier paramètre n'a pas de valeur par défaut, ce qui n'aurait aucun sens, le second paramètre est souvent proposé avec une valeur par défaut de 0. Si on ne la précise pas, le nombre renvoyé par la fonction sera ainsi un entier, soit 3 dans le cas précédent : `round(2.551) = 3`, mais `round(2.551, digits = 1) = 2,6`.

Ce premier exemple est quasiment en tout point assimilable à l'acceptation mathématique d'une fonction, et dès lors, ses paramètres ressemblent fortement à ceux que ce domaine définit, à l'exception que le premier paramètre de la fonction arrondi serait défini comme une variable en mathématiques. Pour autant, l'informatique fait usage de nombreuses fonctions bien moins comparables, car formulées de manière algorithmique et non mathématique.

Prenons l'exemple d'une fonction simple de conversion d'image, permettant par exemple de convertir une image du format JPEG au format PDF. Cette fonction, que l'on nommera `convert`<sup>33</sup>, requiert au moins deux « paramètres » : l'emplacement informatique (le « chemin », ou *path*) du fichier image (JPEG) d'origine, et le chemin du PDF en sortie. Il ne s'agit plus dès lors de paramètres numériques comme en mathématiques ou en statistiques, mais d'éléments nécessaires à une fonction pour être exécutée. De plus, cette fonction accepte aussi d'autres « paramètres », permettant entre autre de redimensionner l'image pendant cette conversion, d'en modifier la résolution, ou encore d'en transformer les couleurs, par exemple en la convertissant en nuances de gris. Ces paramètres, facultatifs, agissent alors comme autant de nouvelles fonctions. Ils ne servent plus uniquement à « paramétrier » le but premier de la fonction de

---

méthode désigne une fonction qui peut être exécutée par une instance de classe en programmation orientée objet. Certaines définitions plus précises sont toutefois souvent utilisé à mauvais escient, provoquant alors des inversions de sens. Ainsi, une procédure est le plus souvent décrit comme une suite d'instructions ne retournant pas de valeur, au contraire d'une fonction. Mais, par exemple dans le langage SAS (INSTITUTE 1990), toute fonction est nommée procédure, à l'instar de la moyenne par exemple (PROC MEANS).

32. Notons que certaines fonctions produisent des « effets de bord », c'est-à-dire ne renvoient pas de données en sortie, mais effectuent une action à partir des données en entrée. La fonction imprimer par exemple, ne renvoie aucune sortie informatique, ce sont ses effets de bord qui déclenchent l'impression d'un document.

33. Cette fonction est disponible dans le logiciel ImageMagick (IMAGEMAGICK STUDIO 2008).

conversion, mais en fait à y ajouter des fonctionnalités, des mécanismes de transformation de l'image source.

On ne peut donc plus véritablement parler de variables qui seraient affectées ou transformées par des valeurs statiques de paramètres. On utilisera donc, dans ce contexte, davantage le terme de paramètre d'entrée ou encore d'argument dans ce cas. Notons qu'en anglais, la différence entre *parameter* et *argument* est plus formalisée qu'en français : ils se définissent par le lieu de leur utilisation. Lors de la définition d'une fonction, on fait appel à des paramètres qui seront utilisés au sein de la fonction. Lors de l'utilisation de cette fonction, l'utilisateur fournira des arguments, dont les valeurs seront alors utilisés en remplacement des paramètres dans la fonction.

« The terms parameter and argument are sometimes used interchangeably, and the context is used to distinguish the meaning. The term parameter (sometimes called formal parameter) is often used to refer to the variable as found in the function definition, while argument (sometimes called actual parameter) refers to the actual input passed. For example, if one defines a function as def f(x) : ... , then x is the parameter, while if it [is] called by a = ... ; f(a) then a is the argument. »

(*Parameter (Computer Programming)* 2017)

Il apparaît donc que si les définitions mathématiques et statistiques d'un paramètre sont assez largement précises et explicites, il en est tout autre dans le champs disciplinaire informatique. Peut-être parce que ce champs est composé de bien plus de praticiens (les développeurs) que de chercheurs, on constate que les termes de variables, de paramètres, d'arguments ou encore d'entrées (*inputs*) y sont assez régulièrement intervertis. Afin de préciser l'emploi que nous ferons de ces termes dans le cadre de la modélisation à base d'agents présentée dans le chapitre précédent, il convient donc de s'intéresser plus spécifiquement aux usages de ces termes dans le domaine de la simulation informatique en sciences humaines.

### 3.3.1.2 Les paramètres dans les modèles agents

**L'approche classique.** Un premier point est à noter : nous n'avons trouvé que très peu de définitions spécifiques de ce qu'est un paramètre dans le champs de la simulation à base d'agents. C'est pourtant un terme employé dans la quasi-totalité de la littérature existante. Cela ne relèverait que d'un problème de jargon non explicité s'il y avait consensus que le sens donné à ce mot, mais au contraire, les acceptations, qui doivent être comprises par le contexte en l'absence de définitions formelles, varient fortement selon les auteurs. Par exemple, la définition que l'on peut extraire de l'un des manuels de référence en modélisation agent (TREUIL, DROGOUL et ZUCKER 2008) s'éloigne fortement de ce que l'on a pu décrire ci-dessus :

« Un modèle dynamique renferme en effet deux composants distincts : une représentation de la structure du système de référence

(exprimée dans le langage du métamodèle), et une représentation des lois régissant sa dynamique. Ces deux représentations sont habituellement pourvues de données ou d'éléments d'information souvent numériques (le minimum pour un modèle dynamique étant d'être pourvu d'un élément représentant le temps) appelés **paramètres**. La « perturbation » d'un modèle par simulation va donc signifier la modification contrôlée de la valeur de certains de ces paramètres, que l'on appellera **entrées** du modèle. Inversement, ce que l'on pourra mesurer dans une simulation sera décrit sous la forme d'autres paramètres qui seront appelé **sorties**.

Les *entrées* d'un modèle dynamique sont des paramètres dont la valeur est définie en dehors du modèle et qui représentent ce que le simulateur peut perturber. Les *sorties* d'un modèle dynamique sont également des paramètres qui expriment ce que l'on cherche à mesurer en réponse à ces perturbations. »

(TREUIL, DROGOUL et ZUCKER 2008, p. 8)

Nous trouvons plusieurs problèmes à cette définition. En premier lieu, l'acceptation très globale de ce qu'est un paramètre rappelle celle d'une variable en informatique. Il semble s'agir d'une vision plus orientée techniquement que conceptuellement. Ainsi, définir le temps — ou la variable informatique permettant de le mesurer — comme un paramètre nous semble bien trop à contrecourant des définitions de paramètres issues des autres champs scientifiques. De plus, le fait que les sorties d'un modèle soient considérées comme des paramètres est en opposition avec l'ensemble de l'usage courant de ce terme, y compris dans le domaine spécifique de la modélisation agent. Nous ne pouvons donc souscrire à cette définition, ni à la vision qu'elle dépeint de ce qu'est un paramètre.

Dans le présent ouvrage, nous donnerons donc à ces termes des sens différents, voire opposés, qui nous semblent plus fréquents dans le champ de la modélisation en sciences humaines et se retrouvent partiellement dans le schéma de Balci (figure 3.8).

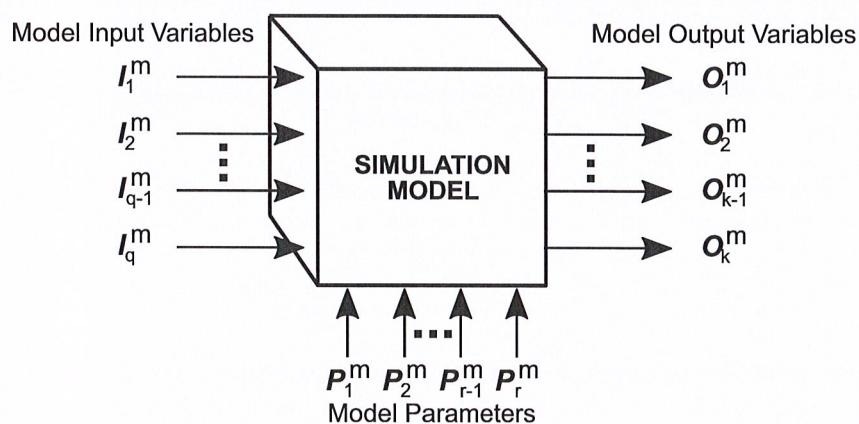


FIGURE 3.8 – Les variables d'un modèle de simulation, adapté depuis BALCI (1994, p. 122).

*Nota bene* : ce schéma est tronqué, ne présentant que la partie « modèle de simulation » alors que celle-ci est mise en miroir à une partie « système » représentant l'empirique.  
la, il émet du domaine empirique

Dans ce schéma et l'article associé, et bien qu'il n'en définisse nulle part ex-

plicitemment le sens, BALCI esquisse la composition d'un modèle de simulation. Le modèle est alimenté par des variables en entrée (*Model Input Variables*), qui sont globalement affectées par des paramètres (*Model Parameters*). Après passage dans la « boîte noire » que représente ici le modèle de simulation ( $m$ ), cela aboutit sur des variables de sortie (*Model Output Variables*).

Notons que si l'auteur du schéma symbolise les entrées, paramètres et sorties au moyen de quatre flèches à chaque fois, leur nombre est variable et sans correspondance. Il y a ainsi  $q$  variables en entrée,  $r$  paramètres et  $k$  variables en sortie. Ces nombres différents représentent bien qu'il n'y a pas de lien direct entre les variables en entrée et celles en sortie : la variable en entrée  $I_1^m$  et la variable en sortie  $O_1^m$  ne correspondent pas nécessairement. De même, BALCI n'indique pas de lien entre les paramètres et les variables (en entrée ou en sortie) : les paramètres agissent à l'échelle du modèle de simulation, sans détail des éléments affectés. *sont pris en compte*

**Paramètres, variables, indicateurs : un essai de définition graphique.**  
Dans ce travail de thèse, nous rendons compte de la construction d'un modèle complexe, doté de très nombreux paramètres, variables d'entrée et variables de sortie (selon la définition de BALCI). Nous avons donc préféré développer notre propre « ontologie » de ces termes, qui nous semble plus adaptée à la description d'un modèle tel que SimFeodal.

Comme chez BALCI, notre ontologie est organisée autour d'une opposition entre des **entrées** et des **sorties**.

*du modèle*  
**Les entrées.** La figure 3.9 représente notre proposition de définition des entrées, utilisée dans ce travail de thèse. Les entrées sont donc composées de deux types d'éléments, des variables et des paramètres. Les **variables** ( $V_n$ ) sont des variables informatiques dont la valeur évolue tout au long de la simulation. Le nombre d'agrégats, par exemple, est une variable : une valeur est assignée à l'initialisation du modèle, et cette valeur est ensuite susceptible de changer à chaque pas de temps.

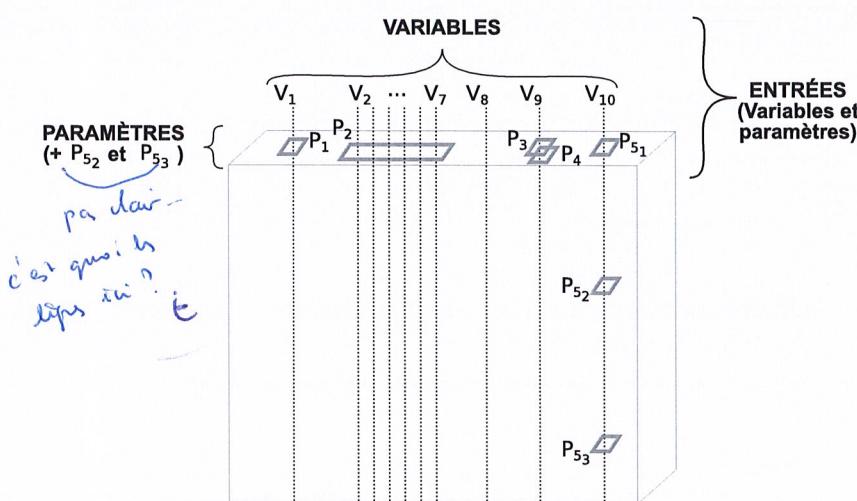


FIGURE 3.9 – Les entrées : variables et paramètres.

Ces variables sont potentiellement affectées par des paramètres ( $P_n$ ), qui peuvent avoir des effets directs sur une unique variable ( $P_1$ , qui pourrait par exemple représenter le paramètre régulant le nombre d'agrégats à l'initialisation), ou sur plusieurs variables ( $P_2$ , nombre de foyers paysans par agrégat initial par exemple). Plusieurs paramètres peuvent aussi affecter la même variable : dans la figure, la variable  $V_9$  est affectée conjointement par les paramètres  $P_3$  et  $P_4$ . On peut illustrer ce type de cas avec l'exemple du nombre de foyers paysans, qui évoluera en fonction d'une part d'un paramètre définissant la population initiale et d'autre part en fonction d'un second paramètre définissant le taux de croissance. Le dernier type de paramètre ( $P_5$ ), assez fréquent dans SimFeodal, correspond aux paramètres dont les valeurs peuvent changer au cours de la simulation. Par exemple, le besoin de protection des foyers paysans évolue au cours du temps simulé, et le paramètre correspondant aura donc plusieurs valeurs décrivant les différentes dates où appliquer le changement ( $P_{51}$ ,  $P_{52}$ ,  $P_{53}$ ), et les modalités que prendra alors la variable  $V_{10}$ .

pourquoi l'initialisation ?

variables fixe + avec variables en échelle

OK  
mais reformuler

**Les sorties.** À l'autre bout du modèle, dans la figure 3.10, on retrouve encore les variables ( $V_n$ ), mais certaines sont extraites du modèle sous forme d'indicateurs de sortie.

le résultat de la simulation utilisée pour construire les ---

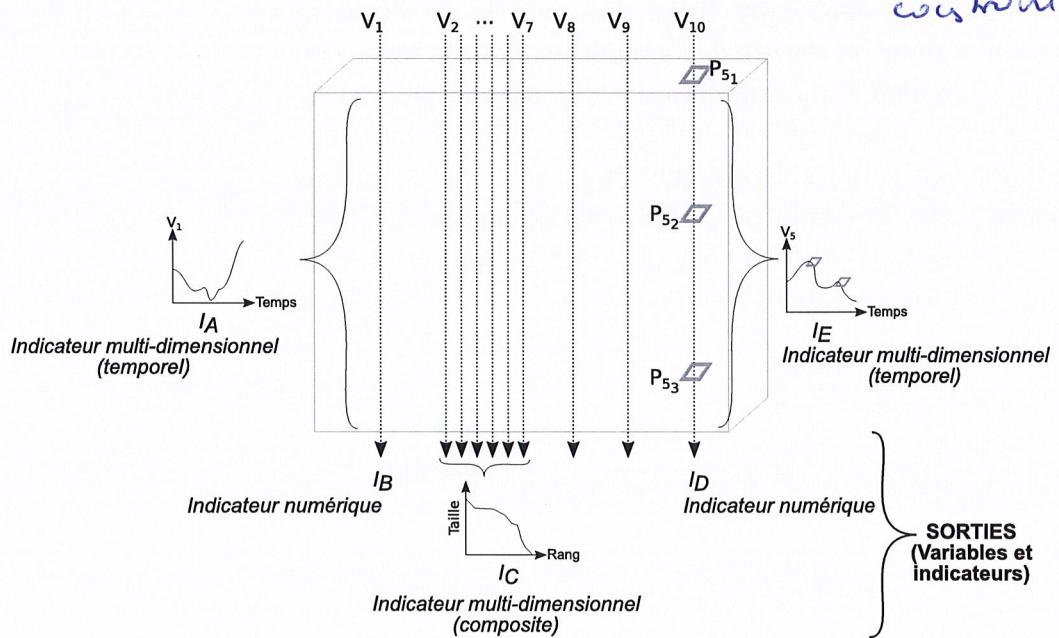


FIGURE 3.10 – Les sorties : variables et indicateurs.

Parmi ces indicateurs de sortie, on identifie là aussi plusieurs types. Certains indicateurs ( $I_B$  et  $I_D$ ) correspondent directement aux valeurs des variables correspondantes ( $V_1$  et  $V_{10}$  respectivement) en fin de simulation. Ces indicateurs sont donc d'une forme numérique (cf. figure 3.7) simple, et renseignent par exemple sur le nombre de foyers paysans ou d'agrégats de population en fin de simulation. D'autres indicateurs sont dits « multi-dimensionnels » (figure 3.7) : ils correspondent soit à la combinaison de plusieurs variables en fin de simulation (indicateur « composite »,  $I_C$ , qui peut par exemple représenter la courbe rang-taille issue des variables  $V_2$  à  $V_7$  qui seraient alors les populations des agrégats), soit à l'évolution temporelle d'une variable ( $I_A$  et  $I_E$ ). Ce sont dans

dans l'exemple de la figure 3.10

les cas de

ce cas des indicateurs de sortie multi-dimensionnels temporels, et ceux-ci sont très fortement mobilisés dans l'évaluation de SimFeodal : évolution du nombre d'agrégats, du taux de concentration des foyers paysans etc.

Notons que si toutes les variables, par définition, varient au cours de la simulation, toutes ne sont pas utilisées pour former des indicateurs de sortie ( $V_8$  et  $V_9$  dans la figure). Elles ne sont pas « inutiles » pour autant, parce qu'elles peuvent influencer/interagir avec d'autres variables par le biais des mécanismes des agents. Leur variation aura alors une influence sur d'autres variables, et donc potentiellement sur différents indicateurs de sortie.

Dans la figure 3.10, l'indicateur de sortie  $I_E$ , de type multi-dimensionnel et temporel, permet de garder à l'esprit que le lien entre variable et paramètre peut être étroit : la variable  $V_{10}$ , dont la valeur finale est extraite dans l'indicateur  $I_D$ , ne peut être étudiée dans son évolution ( $I_E$ ) sans prendre en compte l'influence forte du paramètre évolutif  $P_5$ . On retrouve ainsi de fortes inflexions dans l'évolution des valeurs de  $V_{10}$  au cours du temps simulé, quand les modalités  $P_{52}$  et  $P_{53}$  du paramètre  $P_5$  changent.

**Entrées et sorties du modèle.** La figure 3.11 constitue une synthèse de cette « ontologie » lexicale mobilisée dans cette thèse. Un modèle de simulation comme SimFeodal contient donc en **entrée des variables**, dont la valeur peut changer au cours du déroulement d'une simulation. Ces variables peuvent être initialisées ou affectées par des **paramètres**, qui agissent conjointement et selon des modalités qui peuvent varier au cours du temps. En fin de simulation, on récupère des **sorties**, sous la forme d'**indicateurs de sortie** qui correspondent aux valeurs finales des **variables**, à leur combinaison, ou encore à leur évolution, temporelle, au cours de la simulation.

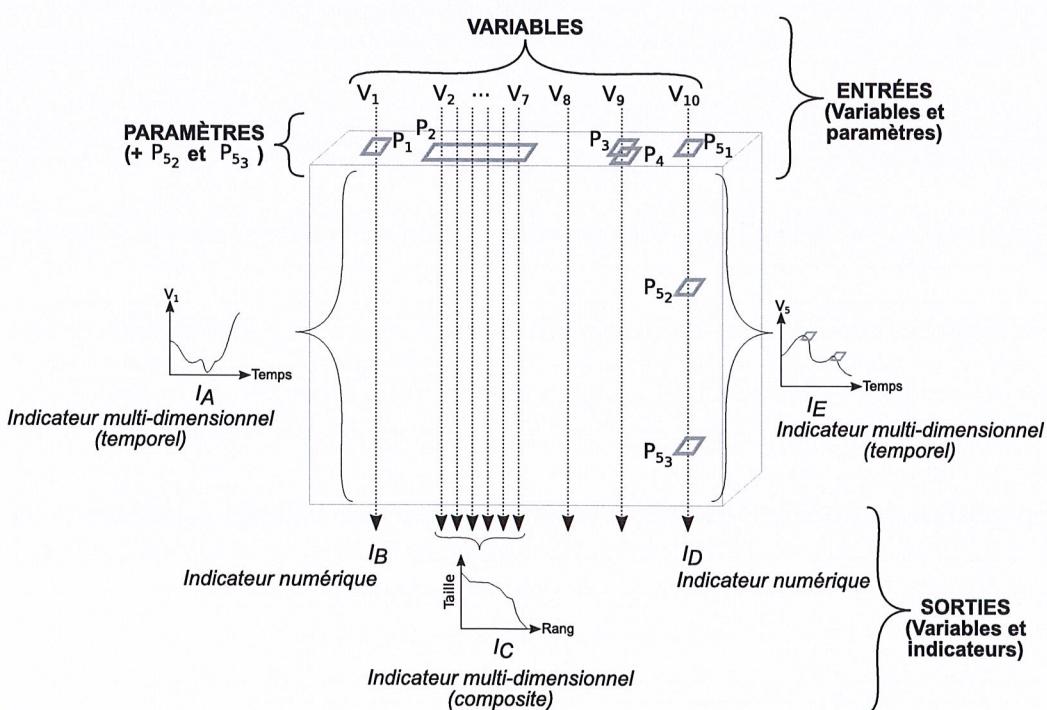


FIGURE 3.11 – Organisation des entrées et sorties du modèle.

**Paramètres et expériences.** Les paramètres ont des valeurs pré-définies, par définition, pour l'exécution d'une simulation ou de ses réplications (dans lesquelles les valeurs des paramètres sont volontairement identiques). Quand bien même leurs modalités peuvent évoluer ( $P_{5_1}$ ,  $P_{5_2}$  ou  $P_{5_3}$ ) au cours du temps simulé, c'est un choix effectué avant la simulation et les valeurs et dates d'applications de ces valeurs n'évolueront pas.

Les valeurs des paramètres sont toutefois amenées à évoluer, non pas au sein des simulations, mais au sein d'expériences (figure 3.12), dont le but est justement de faire varier ces paramètres pour en comparer les effets. Les paramètres variant ( $P_2$ ,  $P'_2$  ou  $P''_2$  par exemple), il est ainsi attendu que les variables de chacune de ces expériences soit affectées différemment, et débouchent ainsi sur la production d'indicateurs de sortie qui montreront des variations entre les expériences.

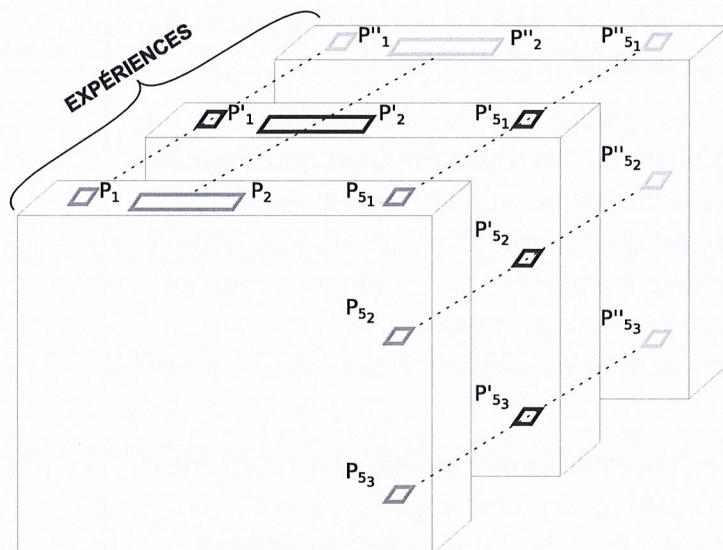


FIGURE 3.12 – Variation des paramètres entre les expériences.

### 3.3.1.3 Les paramètres dans SimFeodal

Les paramètres sont donc un sous-ensemble des entrées d'un modèle qui ont vocation à varier dans les différentes exécutions de ce modèle. Pour autant, c'est un sous-ensemble assez divers, et tous les paramètres n'apportent pas les mêmes connaissances, que ce soit par leur valeur ou par la manière dont ils varient. Il convient donc d'identifier différents types de paramètres, non selon leur caractéristique propre (c'est-à-dire leurs propriétés, qui permettent de les différencier des sorties par exemple), mais selon leur usage et donc la manière dont on pourra les mobiliser. À ce titre, la littérature en sciences humaines et sociales distingue souvent deux types de paramètres. D'un côté, des paramètres empiriques, dont la valeur a une correspondance directe dans le domaine empirique (on peut donc l'observer), et d'un autre, les paramètres techniques<sup>34</sup>, pour lesquels cette correspondance n'existe pas. Comme dans la plupart des

34. MATHIAN et TANNIER (2015) utilisent le terme de paramètre mécanique pour définir ce type de paramètres. Nous trouvons l'usage du mot « technique » plus approprié, en ce qu'il se différencie plus de l'empirique sur le plan de l'usage qui en est fait et que le terme de « mécanique » peut rappeler celui des mécanismes, alors même que les paramètres, empiriques ou non, ont une implication pour les mécanismes.

? L'c'est le cas dans l'examen sur le proj.  
3.11 des param...

} rev. formulation

typologies ayant trait à la catégorisation de valeurs numériques, de nombreux paramètres peuvent se trouver à l'interface de ces deux classes. Comme les paramètres de SimFeodal sont nombreux et seraient trop sommairement distingués dans ces deux catégories, nous avons choisi de mobiliser une typologie en quatre classes (voir encadré 3.3), relatives au sens de ces paramètres vis-à-vis du fonctionnement du modèle. Cette typologie est mixte, entre d'une part le degré de connaissance empirique sur lesquels les paramètres reposent (paramètres techniques vs les autres), et d'autre part l'usage qui en est fait dans le modèle (*input* et contexte vs les autres). Cette typologie est notamment appuyée sur celle de TANNIER (2017, p. 45).

avant ?

#### Encadré 3.3 : Quatre types de paramètres

Dans SimFeodal, nous avons choisi de différencier les paramètres en quatre « types », qui distinguent l'usage qui est fait de chacun de ces paramètres dans le modèle, en particulier vis-à-vis des connaissances expertes sur lesquelles ils reposent ou non.

**Paramètres d'*input*.** Ce sont les paramètres qui définissent l'état du monde simulé lors de son initialisation. Ils sont fortement basés sur les connaissances empiriques, liés à la région qui est modélisée (la Touraine). À ce titre, ces paramètres d'*input* ne sont donc pas amenés à varier en dehors d'application du modèle à d'autres cas d'étude, ou si de nouvelles connaissances empiriques viennent les modifier.

*Exemples : nombre de seigneurs, de villages, d'églises etc. à l'initialisation, dimensions du monde modélisé...*

**Paramètres de contexte.** Les paramètres de contexte ont un ancrage empirique moins appuyé, mais agissent de manière globale et continue sur le contexte de déroulement du modèle. Ce sont les paramètres sur lesquels on peut s'appuyer majoritairement pour l'exploration des hypothèses thématiques du modèle. Leurs variations entre des expériences différentes est nécessairement guidée par l'empirie.

*Exemples : taux de croissance démographique, puissance des communautés, période de construction des châteaux...*

**Paramètres de mécanisme.** Ces paramètres ont une assise empirique plus incertaine, s'appuyant plus sur des ordres de grandeur que les précédents. Ils agissent au niveau des mécanismes des agents, et peuvent varier, tout en restant dans les mêmes ordres de grandeur, lors du paramétrage du modèle.

*Exemples : seuils de distances aux églises et châteaux pour les foyers paysans, probabilité de dons de zones de prélèvement et de châteaux pour les seigneurs, attractivité relative des attracteurs qui composent les pôles...*

**Paramètres techniques** Il s'agit des paramètres dont les valeurs, ou même les ordres de grandeur, ne s'inscrivent sur aucune connaissance empirique. Ces paramètres ont pour raison d'être de permettre à d'autres types de paramètres de s'exprimer en valeurs compréhensibles et exploitables. Dès lors, leurs valeurs sont propres à chaque version, sous-version

types de paramètres de s'exprimer en valeurs compréhensibles et exploitable. Dès lors, leurs valeurs sont propres à chaque version, sous-version ou expérience du modèle formalisé, et une comparaison de ces valeurs entre les différents modèles n'apporte pas de connaissance. Ils sont amenés à varier d'une manière uniquement guidée par l'évaluation du modèle lors du paramétrage, sans que cela n'ait le moindre ancrage ou répercussion empirique.

*Exemples : distance de fusion entre les agrégats, pondération de la satisfaction matérielle des foyers paysans en fonction du nombre de droits acquittés, montants récupérés par les seigneurs selon les types de droits...*

version du modèle ?

Avec cette typologie des paramètres basées sur la nature de ceux-ci mais sur leur utilisation dans le modèle, nous nous inscrivons dans une vision fonctionnaliste et donc très subjective, rappelant la définition d'un modèle de MINSKY (cf. chapitre 1). Selon l'usage que l'on fait du modèle, un même paramètre pourra donc être vu comme un paramètre de contexte ou de mécanisme (par exemple selon l'état des connaissances empiriques liées à ce paramètre sur le cas d'étude traité).

### Comment choisir les valeurs de paramètres ?

Ajout de cette petite conclusion intermédiaire.

Le saut de page était présent pour en « garder la place », mais sera supprimé.

Après avoir présenté notre définition des paramètres, nous pouvons désormais revenir sur le processus qui les mobilise et a demandé cette explicitation : le paramétrage.

Cette étape, que nous allons maintenant définir et illustrer, consiste ainsi notamment à « ajuster » les valeurs des paramètres, ou plus exactement de certains des paramètres, en se basant notamment sur la typologie mise en place dans ces pages (paramètres d, de contexte, de mécanisme, techniques).

### 3.3.2 Le paramétrage

Le paramétrage d'un modèle est souvent réduit à l'un de ses aspects, le « calibrage », étape finale de la construction d'un modèle qui cherchera à reproduire autant que possible des données empirique en faisant varier les valeurs des paramètres jusqu'à ce qu'une combinaison de celles-ci soit satisfaisante.

De nombreux auteurs ont montré que le paramétrage d'un modèle ne pouvait se réduire à cette étape, chacun employant des termes différents pour désigner le processus de paramétrage, processus le plus souvent inscrit comme l'une des composantes de l'évaluation des modèles (chez NGO et SEE (2012), cf. figure 3.3 par exemple).

Le plus souvent, une fois le modèle construit, le modélisateur s'attache à son « calibrage », en cherchant pour chaque paramètre la ou les valeurs qui permettront au modèle de s'approcher, au plus près, des données empiriques devant être reproduites, c'est-à-dire la conjugaison de « valeurs optimales » de paramètres minimisant l'écart entre les données simulées et les données empiriques de contrôle. Cette étape, que l'on nomme aussi souvent calibration par anglicisme, peut se faire de manière manuelle, par approximations successives – CROOKS et al. (2019, p. 253) nomme cela « calibration qualitative » –, par semi-automatisme, par exemple en effectuant des analyses de sensibilité – THIELE, KURTH et GRIMM (2014, §2.3–2.4) mobilisent ainsi l'analyse de sensibilité, via échantillonnage, pour faire de l'estimation de paramètres –, ou encore de manière entièrement automatique – par exemple, pour HEPPENSTALL, EVANS et BIRKIN (2007) ou NGO et SEE (2012, p. 188), en suivant une méthode d'optimisation à base d'« algorithmes génétiques ».

Dans notre travail, nous souhaitons revenir sur cette approche de la modélisation, ancrant le paramétrage comme étape ultime de la construction d'un modèle, en particulier en ce que nous considérons que cette pratique de recherches de valeurs optimales est un exercice qui devrait s'effectuer tout au long de la construction du modèle, de manière plus itérative que conclusive.

#### 3.3.2.1 Définition

Le terme de paramétrage recouvre deux sens différents, dont la distinction peut se faire selon qu'on l'utilise pour définir un processus ou pour caractériser une configuration. Ici, nous emploierons plutôt le premier cas, définissant dès lors le paramétrage comme le processus, manuel ou automatique, visant à constituer cette configuration de paramètres. Dans ce deuxième cas, le paramétrage désigne un ensemble de valeurs de paramètres, par exemple quand on mentionne un paramétrage par défaut, ou un paramétrage optimal. Pour ne pas risquer de contre-sens, nous préférerons le terme de configuration de paramètres ou de « jeu de valeurs de paramètres ».

On tend à distinguer le paramétrage – passage obligé ne nécessitant pas d'être évoqué – du calibrage, processus systématique qui inscrirait le modèle comme un outil scientifique et incontestable. Nous choisissons ici de confondre ces approches, non pas en considérant le paramétrage comme un outil d'évaluation

tion du modèle, mais comme une composante inhérente à la construction d'un modèle, quelles que soient les formes et les temporalités que le paramétrage adopte. Le paramétrage est en effet une pratique utile dans la construction du modèle, car les résultats auxquels il aboutit, c'est-à-dire les valeurs de paramètres qui semblent mieux adaptés, renseignent aussi bien sur les biais des mécanismes adoptés que sur leur efficacité réelle. Par exemple, quand, après avoir ajouté un mécanisme, on se rend compte que des variations dans les valeurs de paramètres ne changent pas réellement les sorties du modèle, cela peut être l'occasion de repenser le mécanisme dans son ensemble, ou plus souvent, la manière dont le paramètre est mobilisé dans ce mécanisme. On retrouve cette logique dans l'exploration par Clara Schmitt du modèle SimpopLocal (SCHMITT 2014), qui a permis de réaliser que la variation de l'un des paramètres (*InnovationLife*) n'avait que peu d'impact sur les sorties du modèle, tout en rendant son calibrage plus complexe et instable :

« Au-dessous du seuil des 150 pas de simulation pour le paramétrage de InnovationLife, le calibrage du modèle est très difficile voire impossible. Au-dessus de ce seuil, le mécanisme associé au paramètre InnovationLife n'a plus d'effet sur le calibrage du modèle. Dans un souci de parcimonie du nombre et de la complexité des mécanismes simulés dans le modèle SimpopLocal, il est justifiable de retirer du modèle ce mécanisme qui n'est pas nécessaire à la simulation de la dynamique de croissance recherchée. »

SCHMITT (2014, p. 224)

Ajout des paragraphes de définition suivants, qui s'insèrent donc entre ce message et le paragraphe « Désambiguisation », page 55 dans l'impression.

Dans ce travail de thèse, nous reprenons ainsi le sens du terme paramétrage tel qu'initialement employé par HIRTZEL (2015)<sup>35</sup> et ensuite explicité par TANNIER (2017) :

« Le paramétrage d'un modèle consiste à fixer les valeurs des variables et paramètres de mécanisme, au moyen d'analyses spatiales ou statistiques de données empiriques, de transcriptions de dires d'experts, ou de simulations avec le modèle. Le paramétrage comprend une phase d'estimation (statistique ou autre) des valeurs des paramètres et variables, et une phase de calibrage si celle-ci est nécessaire. »

TANNIER (2017, p. 52)

Nous ajoutons à cette définition une composante d'implémentation, quand celle-ci est minime, qui consiste à adapter l'implémentation des mécanismes de

35. « La notion de paramétrage d'un modèle est souvent associée à celle de calibrage, et ces deux notions, bien que différentes, sont parfois confondues dans la littérature (Richiardi et al., 2006). Le calibrage consiste à tester plusieurs jeux de paramètres possibles pour une variable et à choisir l'un d'eux pour l'exécution des simulations, selon sa capacité à atteindre les objectifs définis. Il constitue ainsi une étape du paramétrage d'un modèle. Cette étape n'est pas forcément indispensable : si les valeurs initialement affectées permettent d'atteindre les objectifs du modélisateur, celui-ci n'a pas besoin de procéder à un calibrage. » (HIRTZEL 2015, p. 136).

La suite me  
perrait continuer.  
au fin "non pris"  
les intérêts  
répense  
calibry et  
parce que

environ une  
étape de +

manière à en rendre le résultat plus satisfaisant. Cela n'implique pas de changer le modèle conceptuel, et n'est pas non plus véritablement un changement dans l'implémentation du modèle. Par exemple, changer l'ordonnancement du détail d'un mécanisme relève à notre sens du paramétrage.

On peut illustrer cela avec l'exemple du mécanisme de définition du contour spatial des agrégats (cf. [méca, 2.7.2.1, chap2](#)) : un *buffer* est actuellement appliqué autour de l'enveloppe convexe formée par les foyers paysans membres d'un agrégat afin de fusionner d'éventuels agrégats très proches. On peut adapter ce mécanisme en appliquant ce *buffer* à la fin du mécanisme plutôt qu'au milieu, une fois que l'héritage des agrégats précédents a été transféré par exemple.

Cela n'aurait pas un impact important sur le plan conceptuel, ni même d'ailleurs lors de l'observation agrégée de tous les agrégats. Toutefois, au niveau local, cela peut avoir un impact non négligeable sur l'historique des agrégats. À notre sens, il s'agit d'une adaptation du modèle du même ordre qu'un changement de valeur de paramètre, et nous incluons ainsi ce type de modifications dans le processus de paramétrage.

*jeu*

*des termes utilisés*

**Désambiguïsation.** De nombreux termes sont utilisés dans la littérature, souvent sans réelle distinction, pour désigner cette opération qui consiste à choisir un jeu de paramètres pour un modèle. Pêle-mêle, on y retrouve le paramétrage, le calibrage, l'ajustement.... Nous proposons une représentation graphique, dans la figure 3.13, de notre usage de ces termes, et en donnons des définitions dans l'encadré 3.4.

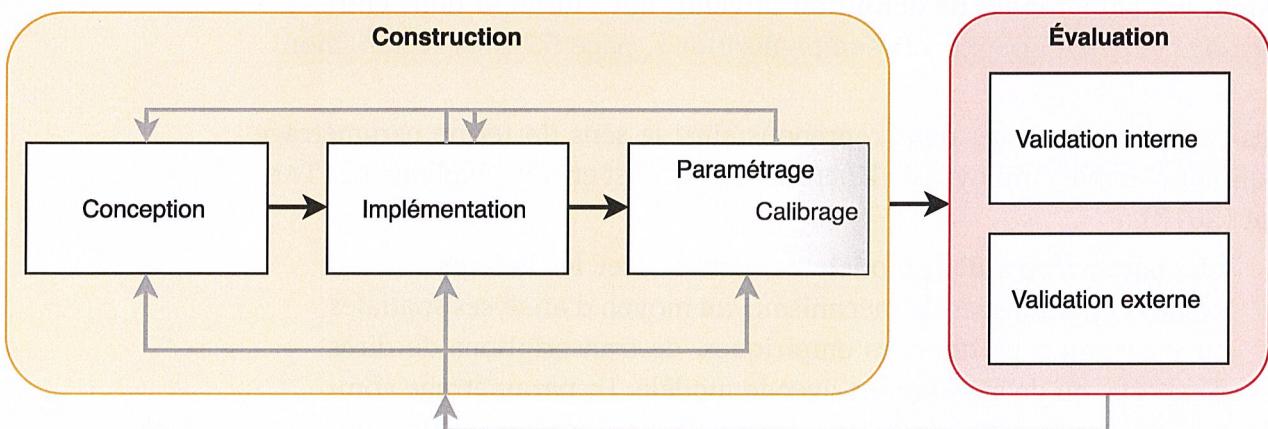


FIGURE 3.13 – Étapes du processus de modélisation.

*TB mais où est  
la désambiguïsation ?  
Je ne connaissais pas dans  
le sens que tu utilises -*

**Encadré 3.4 : Construction, paramétrage et calibrage**

La figure 3.13 représente l'enchevêtrement des différentes étapes du processus de modélisation, dont nous allons brièvement détailler les parties relatives à la construction ici.

**Construction.** Par construction, nous définissons l'ensemble des étapes relatives à la création du modèle, c'est-à-dire la conception du modèle conceptuel, son implémentation sous forme d'un modèle de simulation et le paramétrage du modèle. À force d'allers-retours entre ses composantes, la construction aboutit sur un modèle « finalisé », qui peut être évalué de manière globalement (chacune des composantes de la construction pouvant être évaluée individuellement : l'implémentation est par exemple évaluée au moyen des méthodes de validation interne).

/ a

**Paramétrage.** Le paramétrage du modèle est l'étape où l'on vérifie la cohérence empirique et simulée du modèle. Lors du paramétrage, on ajuste les valeurs de paramètres, leur utilisation par les mécanismes et le détail du fonctionnement technique de ces mécanismes. On cherche par cela à obtenir un modèle plus satisfaisant du point de vue des dynamiques et résultats qu'il produit. Une des définitions du nouveau petit Robert (ROBERT, REY-DEBOVE et REY 1993) nous semble particulièrement représentative de cela :

« Paramétrier (informatique) : « Programmer (un appareil complexe), en définissant les paramètres assurant son fonctionnement optimal. Ex. *Paramétrier une imprimante* ».

**Calibrage.** On réserve souvent, et nous nous y tiendrons, ce terme à la dernière étape dans l'aboutissement d'un modèle. Une fois les mécanismes fixés et des objectifs définis, on peut procéder au calibrage (aussi nommé, par anglicisme, « calibration »), c'est-à-dire à une exploration de l'espace des paramètres ayant pour but de stabiliser les paramètres afin de se rapprocher autant que possible de ces objectifs. Dans cette thèse, nous inscrivons le calibrage comme une des étapes, finale, du paramétrage.

B de  
redondance  
avec le test  
principal ?  
ici + clair

**Co-construction.** La co-construction, définie dans le chapitre 1, est l'approche mise en place pour réaliser chacune de ces étapes. L'approche de co-construction, dans ce travail de thèse, concerne donc aussi bien la « construction » que l'évaluation du modèle.

= un autre  
référence  
à garder ici ?

Notons que toutes ces étapes sont représentées de manière linéaire et chronologique, le paramétrage marquant la fin de la construction du modèle et l'évaluation la fin de l'« évolution » du modèle. Pour autant, les allers-retours entre ces étapes sont extrêmement nombreux (flèches grises), et la figure 3.13 aurait aussi pu être représentée (en perdant en simplicité) sous forme d'une spirale, voire d'une hélice, comme la « vis sans fin » d'Archimède.

### 3.3.2.2 Paramétrier des modèles, illustrations

Afin d'illustrer ces propos, on peut s'appuyer sur des modèles bien connus de la littérature, issus de deux champs disciplinaires différents : le modèle gravitaire (encadré 3.5) et le modèle de Schelling (encadré 3.6).

#### Encadré 3.5 : Paramétrage du modèle gravitaire

**Description.** Ce modèle statistique, dans la formulation qu'en a faite STEWART (1948), vise à prédire des flux (démographiques, marchands etc.) potentiels entre des lieux à partir d'une analogie avec la loi physique de la gravitation. Dans le formalisme le plus simple, « sans contrainte » (PUMAIN et SAINT-JULIEN 2001), on peut l'exprimer ainsi :

$$F_{ij} = \frac{k \times M_i \times M_j}{d_{ij}^\alpha}$$

Ce modèle présente trois valeurs empiriques ( $M_i$ ,  $M_j$  et  $d_{ij}$ ) et deux paramètres ( $k$  et  $\alpha$ ).  $k$  est un paramètre technique (encadré 3.3), puisque bien que basée sur l'empirique, sa valeur – et les fluctuations de celle-ci – ne donne aucune information sur le système étudié.  $\alpha$  est un paramètre de contexte : il est basé sur une logique empirique, et conditionne fortement le modèle (et notamment les valeurs du paramètre  $k$ ). En faisant varier  $\alpha$ , on explore des hypothèses thématiques relatives à l'importance de la distance dans les interactions entre les composantes d'un système. À partir des valeurs empiriques  $M_i$  et  $M_j$  qui caractérisent les masses (populations, ou stocks de marchandise par exemple) des lieux  $i$  et  $j$ , et de la distance qui les sépare  $d_{ij}$ , on peut ainsi prédire  $F_{ij}$ , le flux entre ces lieux.

**Objectif.** Afin que ce modèle donne des ordres de grandeur réalistes quand aux quantités échangées, il faut le paramétrier en définissant une valeur de  $k$ , permettant dès lors d'obtenir un rapport entre les masses d'origine et les quantités échangées. La valeur de ce paramètre est conditionnée par celle de  $\alpha$ , que l'on nomme fréquemment « frein de la distance », en ce qu'il permet de quantifier l'impact qu'aura un éloignement plus ou moins important sur la quantité de flux échangés.

**Paramétrage.** Pour « ajuster »<sup>a</sup> le modèle, il faut donc en réaliser le paramétrage en s'appuyant sur les éléments connus de l'équation (les valeurs empiriques).  $\alpha$  et  $k$  étant liés dans l'équation, la valeur de chacun de ces paramètres dépend de celle choisie pour l'autre, et un changement dans l'un des paramètres entraînera la nécessité de modifier l'autre.

Le paramétrage le plus simple consiste à réaliser une régression linéaire sur les logarithmes décimaux des distances et des flux observés,  $k$  prenant alors la valeur du logarithme de l'ordonnée à l'origine et  $\alpha$  celle du coefficient directeur de la courbe. Le modèle est alors calibré (voir encadré 3.4), mais si l'on souhaite donner une valeur spécifique à l'un des paramètres, par exemple pour utiliser une valeur classique de 2 au frein de la distance ( $\alpha$ ) (PUMAIN 2004), il faudra alors le re-soumettre à au paramétrage pour adapter  $k$ . De même si l'on modifie les lieux et/ou les masses sur lesquels il s'applique.

le sens du  
paramètre qui sont  
les valeurs  $M_i$ ,  $M_j$  et  
dans pour les lieux  
 $i$  et  $j$ .

l'intensité des

? OK

de technique la +  
simple pour paramétrier le  
modèle consiste à -

a. C'est le terme employé par PUMAIN et SAINT-JULIEN (2001), que l'on peut ici retenir comme équivalent de paramétriser. DURAND-DASTÈS (1995, p. 298) utilise pour sa part le terme de calibration à propos du même modèle.

La manière classique de mener le paramétrage du modèle gravitaire est donc directe et déterministe. Pour une configuration donnée de masses et de distributions spatiales – impliquant les distances –, un et un seul jeu de valeurs de paramètres permet un ajustement optimal, c'est-à-dire d'assurer un écart minimal entre les valeurs empiriques et les valeurs théoriques (simulées). Le paramétrage de ce modèle se résume donc à l'étape de calibrage.

#### *Encadré 3.6 : Paramétrage du modèle de ségrégation de Schelling*

**Description.** Le modèle de Schelling<sup>a</sup> est un modèle de simulation décrit par l'économiste SCHELLING (1971) qui vise à montrer comment un espace peut passer d'un état intégré à un état ségrégué ethniquement à travers une succession de comportements individuels de mobilité résidentielle. Il montre en particulier qu'on peut parvenir à un état ségrégué même quand les comportements individuels sont majoritairement tolérants.

L'espace du modèle est défini comme une grille carrée composée de  $N^b$  cellules. Au début de la simulation, des agents de deux types, en proportions similaires, sont distribués aléatoirement dans l'espace du modèle. Chaque agent occupe une cellule, et le nombre total de cellules occupées (et donc d'agents) dépend d'un paramètre de densité  $d$  qui représente la part (entre 0% et 100%) du nombre de cellule qui sera occupée. Chaque agent est défini par une satisfaction. Celle-ci correspond à la part de cellules voisines occupées par des agents « étrangers », c'est-à-dire d'un autre type : plus le voisinage contient d'étrangers, moins l'agent est satisfait. Si cette satisfaction est inférieure à un paramètre de tolérance  $S$ , l'agent est considéré comme insatisfait et se déplace aléatoirement dans une cellule non occupée. A terme, Schelling montre que même en considérant des valeurs de  $S$  assez élevées, c'est-à-dire un comportement plutôt tolérant, la succession de choix individuels entraîne la mise en place d'une distribution spatiale très ségréguée. On considère que le modèle a convergé quand l'ensemble des agents sont satisfaits, ce que ne permettent pas toutes les combinaisons de valeurs de paramètres.

Dans ce modèle, on peut donc dénoter trois paramètres,  $N$ ,  $S$  et  $d$ .  $N$  peut être vu comme un paramètre d'*input*, en ce qu'il définit une partie de la configuration spatiale, mais aussi comme un paramètre technique. Il n'y a en effet aucune correspondance entre l'espace empirique modélisé et sa représentation théorique dans le modèle de simulation.  $S$  est un paramètre de contexte : il a un ancrage empirique, certes limité, mais dont l'ordre de grandeur peut être estimé par des sociologues ou psychologues. De plus, c'est le paramètre qui conditionne thématiquement la convergence ou non du modèle vers un état stable. C'est donc le paramètre qui sera mobilisé pour tester les hypothèses, empiriques

*corr. à une  
répartition aléatoire*

*identifier ?*

ou théoriques, des modélisateurs. Le dernier paramètre,  $d$ , peut être vu comme un paramètre technique puisqu'il ne s'appuie sur aucune connaissance empirique, mais aussi comme un paramètre de mécanisme : il n'a de sens que parce que le mécanisme de déplacement des agents dirige ceux-ci vers un « espace vide », et agit donc comme un paramètre de régulation de ce mécanisme. Avec un modèle conceptuel très proche, où les agents seraient capables « d'échanger » leur position plutôt que de rejoindre une cellule non occupée, ce paramètre n'aurait pas de sens.

**Objectif.** Le paramétrage de ce modèle consiste à fixer un  $N$  constant et à faire varier  $d$  et  $S$  afin ~~de~~ trouver des valeurs permettant la convergence vers une situation stable. Quand  $d$  est très faible (30% dans l'exemple de DAUDÉ et LANGLOIS (2006)), quelles que soient les valeurs de  $S$ , le modèle converge rapidement : quand l'espace du modèle dispose d'une faible densité d'agents, il est facile pour ceux-là de créer des agrégats homogènes distants d'agrégats de l'autre type d'agents. Quand  $d$  est plus important ( $\geq 66\%$ ), l'espace disponible étant limité, toutes les valeurs de  $S$  ne permettent pas la convergence du modèle. Le paramétrage du modèle aura donc pour objectif de trouver la valeur maximale possible – pour un  $N$  et un  $d$  donné – que peut prendre le paramètre de tolérance  $S$  tout en laissant le modèle converger.

**Paramétrage.** Contrairement au modèle gravitaire, le modèle de Schelling est stochastique : les agents se déplacent aléatoirement ~~quand~~ non satisfaits. Dès lors, deux exécutions du modèle avec le même jeu de paramètres n'entraîneront pas forcément la même configuration spatiale. Qui plus est, pour certains jeux de paramètres, seules certaines exécutions convergeront. Le paramétrage de ce modèle ressemble donc à celui qui est réalisé pour le modèle SimFeodal. La manière « traditionnelle », c'est-à-dire manuelle, consiste à fixer un  $d$ , puis à essayer d'augmenter le  $S$  tant que le modèle converge. Si le modèle converge pour tout  $S$ , on augmente la valeur de  $d$  et on recommence à chercher la valeur maximale possible pour  $S$ . De part la nature stochastique du modèle, chaque jeu de paramètres doit être simulé plusieurs fois, le nombre de ces répliques dépendant de la part d'aléa dans le comportement du modèle. Un modèle de Schelling calibré, c'est-à-dire dont le paramétrage est achevé, donnera pour un  $N$  fixé les valeurs maximales de  $d$  et de  $S$  atteignables.

a. Une description plus poussée accompagnée d'une description de l'exploration du modèle peut être lue dans (DAUDÉ et LANGLOIS 2006)

b. Nous reprenons ici la notation  $M(N, d, n, S)$  proposée par DAUDÉ et LANGLOIS (2006, p. 433) en n'explicitant toutefois pas le paramètre  $n$  décrivant le type de voisinage (4 ou 8) utilisé.

Le paramétrage du modèle de Schelling est donc très différent de celui du modèle gravitaire. Il nous semble plus représentatif de la démarche classique de paramétrage des modèles de simulation à base d'agents. Contrairement au modèle gravitaire, le paramétrage est ainsi executé de manière itérative et répétée : à chaque augmentation de la valeur de  $d$ , il faut ré-exécuter le calibrage du paramètre  $S$ , voire de  $N$  si on choisit de faire varier ce paramètre aussi.

il sont

OK  
= "essai et erreur" ?

on peut résoudre  
d'un côté on  
stochastic stat  
de l'autre essay et  
erreur

À travers ces deux exemples ayant traits à des méthodes différentes, on retrouve deux approches de paramétrage que l'on souhaite ici voire confondues : dans le premier cas, le paramétrage du modèle gravitaire tend à son calibrage, à la recherche d'une solution optimale, c'est-à-dire meilleure que toute autre. Pour un phénomène donné (un jeu de données précis par exemple) et un formalisme donné (l'expression la plus simple du modèle, ici sans contrainte), seule un couple de paramètres  $\alpha$  et  $k$  permet ainsi d'obtenir des flux modélisés proches des flux observés. Dans le cas du modèle de Schelling, il n'y a pas de recherche d'optimum : le paramétrage permet de comprendre le modèle en définissant les limites en matière de convergence. Une configuration de paramètres  $S$  et  $d$  ne sera pas meilleure qu'une autre, mais pour un  $S$  donné, on saura quelle est la valeur maximale de  $d$  permettant de parvenir, le plus souvent (en tenant compte de l'aléa) à cette convergence (et réciproquement).

### 3.3.2.3 Paramétrier SimFeodal

Un ou des modèles à paramétrier ? Ce dernier exemple montre l'importance des allers-retours entre évaluation et paramétrage, et illustre la démarche itérative et incrémentale du paramétrage. Parmi les étapes du paramétrage, le calibrage devrait ainsi être mené à chaque « version » du modèle. Ces versions, dans le modèle de Schelling, pourraient correspondre à un changement de valeur de  $S$  ou de  $N$ . Quand ces paramètres évoluent, il est ainsi nécessaire d'évaluer le modèle qui en résulte (y-a-t-il convergence ?), et en fonction du résultat, de calibrer les autres paramètres de manière à obtenir une configuration optimale.

Sans aller jusqu'à voir dans ces versions successives des modèles à part entière – ce que les tenant de la modélisation modulaire (GRIMM et al. 2005) ou multi-modélisation (PUMAIN et al. 2017) peuvent considérer –, on peut tout de même considérer que chacune de ces versions requiert des cycles de construction et d'évaluations indépendants les uns des autres (cf. figure 3.13). Chacune de ces versions devra dès lors être adaptée par une paramétrage dédié.

C'est, à notre sens, d'autant plus important que certaines versions de modèles peuvent donner lieu à des évolutions majeures, conceptuelles, méthodologiques ou techniques. Le paramétrage est dans ce cas à « re-faire » entièrement, c'est-à-dire que les étapes précédentes de paramétrages ne peuvent plus forcément aider à réduire l'amplitude des valeurs parmi lesquels fixer les paramètres par exemple.

Pour illustrer ce propos, on peut prendre l'exemple d'une modification qui a eu lieu sur SimFeodal au cours de son développement (dans la version « 0 », cf. tableau 3.1). On considérait jusque là que pour prendre en compte la population rurale (hors Tours donc), 1000 foyers paysans donnaient une bonne idée de la situation en 800 pour l'espace considéré. Lors d'une réunion, les thématiciens se sont rendus compte que ce nombre sous-estimait très largement la population réelle, et qu'il valait mieux passer ce paramètre à 4000 foyers paysans. Les autres paramètres, fixés en bonne partie empiriquement, n'avaient visiblement aucune raison d'évoluer du fait de ce changement. Les mécanismes et effets de

*term à la m  
on ne devrait pas c  
représenter une répartiti  
de distribution observé  
de 2 pop. -*

*N OK, S corr.  
à un autre registre !  
N et d déf. le  
contexte.*

*a priori*

seuil étaient en effet conçus de manière à être relatifs aux masses manipulées, et le changement attendu était une augmentation linéaire des indicateurs de sortie.

Dans les faits, cette légère modification a entraîné une obligation de repenser la quasi-totalité des autres paramètres et d'ajuster une bonne part des mécanismes, de la même manière qu'un changement de  $N$  dans le modèle de Schelling (encadré 3.6) demande de revoir  $d$  et  $S$ .

La concentration des foyers paysans était en effet bien trop rapide avec autant d'individus, et le modèle convergeait en quelques pas de temps vers une configuration presque statique et très concentrée. Tous les mécanismes de régulation de ce comportement étaient dès lors rendus ineptes, et il a fallu changer en profondeur la manière dont chaque paramètre et mécanisme interagissait avec les autres.

(effet mécaniq. du au nb +  
imp. d'ind)

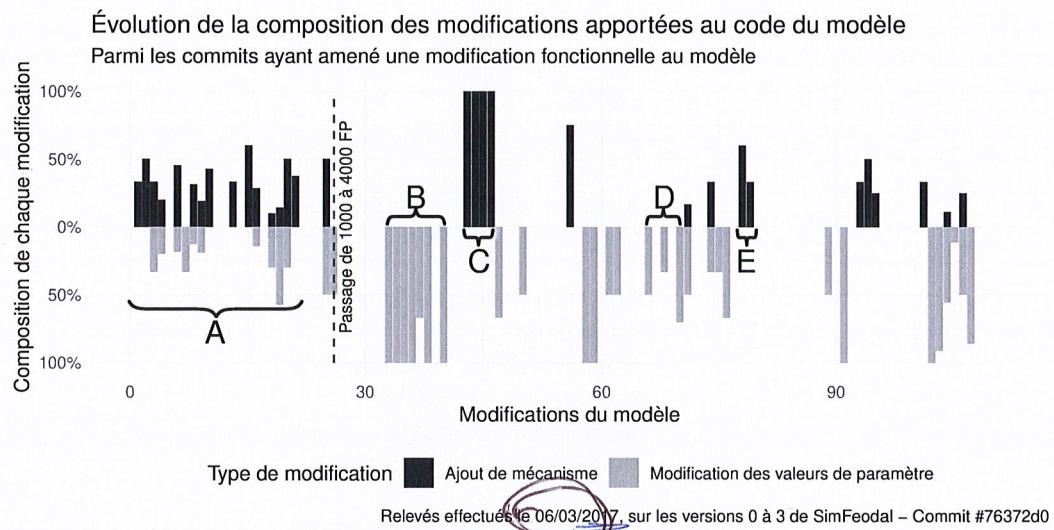


FIGURE 3.14 – Temporalité du paramétrage du modèle.

Chaque « enregistrement » (commit) correspond à une version modifiée et sauvegardée du modèle, contenant un ou plusieurs changements. On a ici rapporté le nombre de changements de chaque type (modification de mécanisme ou de valeur de paramètre) au nombre total de changements de chaque commit afin de figurer l'évolution des types de modification au cours de la construction du modèle.

C'est un changement majeur, que l'on peut constater dans la figure 3.14 entre les périodes « A » et « B » : le premier paramètre qui évolue est la population initiale, et on constate que de nombreuses modifications suivantes (B) portent sur l'ajustement de valeurs de paramètres. Malgré ce calibrage, on note que les résultats n'ont pas suffit et qu'il a donc fallu modifier aussi des mécanismes (période C), que ce soit au niveau du détail de leur implémentation ou au niveau conceptuel.

où je suis allé ?

La première version (fin de la période A) avait été paramétré aussi correctement que possible, mais ce paramétrage était entièrement à refaire avec la nouvelle version (modifications de la période B, figure 3.14).

Cet exemple renforce le caractère nécessaire du paramétrage, et qui plus est, montre que ce processus doit être mené de manière continue et répétée. Dans un modèle exploratoire comme SimFeodal (cf. chap1 ou chap2), on ne peut

1.?

à je suis allé ?  
= ajout de méca... ?

) mieux articuler au parag. précédent.  
+ déplacer figur.

ment que possible, mais ce paramétrage était entièrement à refaire avec la nouvelle version (modifications de la période B, figure 3.14).

Cet exemple renforce le caractère nécessaire du paramétrage, et qui plus est, montre que ce processus doit être mené de manière continue et répétée. Dans un modèle exploratoire comme SimFeodal (cf. chap1 ou chap2), on ne peut ainsi penser le paramétrage que sous la forme d'un calibrage qui serait l'étape ultime de construction du modèle avant validation. Si le paramétrage n'est pas régulier, alors chacune des versions et sous-versions successives ne pourra pas être évaluée de manière comparable, et on cours le risque de disqualifier une version du modèle qui aurait pu être plus plausible que les suivantes mais n'a pas été ajustée comme elle aurait pu l'être.

### Quels paramètres ?

Ajout de cette sous-sous-sous-partie ici.

S'insère entre la fin de « Un ou des modèles à paramétrier ? » et le début de « Historique de SimFeodal. », page 62 dans l'impression.



Dans la partie précédente, nous avons distingué plusieurs types de paramètres (voir l'encadré 3.3 dans la section 3.3.1.3), selon leur usage dans le modèle et leur niveau d'inscription dans les connaissances empiriques. Cette distinction nous semble particulièrement utile pour dresser une hiérarchie des paramètres sur lesquels jouer pour adapter le modèle. *pas ajuster donc ?*

Les **paramètres d'input**, pour commencer, n'ont pas vocation à changer lors de l'évolution du modèle : ils ont été définis de manière empirique, et ne peuvent donc revêtir d'autres valeurs uniquement pour améliorer l'ajustement du modèle. Toutefois, si les connaissances expertes évoluent, les phases de paramétrage du modèle sont l'occasion de répercuter ces nouvelles connaissances. Les changements de ces paramètres peuvent résulter d'un besoin de préciser la mise en œuvre des connaissances empiriques<sup>36</sup>, ou simplement de la correction d'approximations inexactes<sup>37</sup>. Dans tous les cas, le paramétrage de ces paramètres n'est pas guidé directement par les résultats d'une version du modèle, mais par les connaissances qui y sont injectées. Il en va de même pour les **paramètres de contexte** : ils peuvent être précisés ou corrigés, mais leur paramétrage ne doit pas être guidé par l'évaluation du modèle.

*défini où ?*

*# d'input ?*

Le cas des **paramètres de mécanisme** est plus emblématique du processus de paramétrage : ils reposent sur des ordres de grandeur plus incertains, et leur valeur peut donc assez librement être modifiée. Lors du paramétrage, il faut naturellement s'assurer de ne pas les faire varier en dehors des intervalles définis empiriquement, mais ils se prêtent très bien à des ajustements manuels. On peut ainsi sans crainte procéder à des modifications successives, par allers-retours entre l'évaluation des sorties du modèle et le changement de valeur

36. Un exemple d'un tel choix est présenté dans le chapitre 6 (partie 6.1.1.1), où l'on a choisi de faire passer les dimensions du monde simulé de  $100 \times 100$  km à  $80 \times 80$  km.

37. Comme c'est le cas dans l'exemple du passage de 1000 à 4000 foyers paysans présenté plus haut.

de ces paramètres. On pourra de cette manière figer, temporairement – en attendant de nouveaux ajustements dus au paramétrage d'autres paramètres –, des valeurs dont l'effet combiné sera plus satisfaisant sur le déroulement et l'aboutissement des simulations.

Les paramètres techniques, enfin, sont les candidats les plus évidents au paramétrage, et plus spécifiquement à sa composante de calibrage. Puisqu'ils ne reposent sur aucune connaissance empirique, ils constituent ainsi les « leviers » naturels sur lesquels on pourra mener le paramétrage, y compris de manière automatisée pour ces paramètres dont on peut faire varier les valeurs de plusieurs ordres de grandeur sans conséquence conceptuelle ou empirique sur le modèle. Comme dans l'exemple du paramétrage du modèle gravitaire (encadré 3.5), la modification de ces paramètres est extrêmement pratique pour compenser le paramétrage des autres types. Ainsi, si de nouvelles connaissances viennent pousser à modifier un paramètre de contexte par exemple (comme le paramètre de frein de la distance  $\alpha$ ), on peut espérer parvenir à compenser cette variation, sur les sorties, en modifiant un paramètre technique ( $k$  dans le modèle gravitaire) en conséquence.

Rappelons aussi, comme indiqué lors de la définition (section 3.3.2.1), qu'en dehors des valeurs des paramètres, le paramétrage est aussi l'occasion de modifier, d'une manière modérée, le détail de l'implémentation de certains mécanismes.

scénario  
?  
k ou l = ?  
" le + évident "  
" le + intéressant "?  
j'aurais plus  
terminé par les  
param.

Il ne défend pas l'inverse ?

Inserion sous-partie n° quels paramètres de la page suivante

ainsi penser le paramétrage que sous la forme d'un calibrage qui serait l'étape ultime de construction du modèle avant validation. Si le paramétrage n'est pas régulier, alors chacune des versions et sous-versions successives ne pourra pas être évaluée de manière comparable, et on cours le risque de disqualifier une version du modèle qui aurait pu être plus plausible que les suivantes mais n'a pas été ajustée comme elle aurait pu l'être.

**Historique de SimFeodal.** Dans le paragraphe précédent, de la même manière que dans le chapitre 2, nous avons plusieurs fois mentionné les versions et sous-versions de SimFeodal. Il nous paraît important ici de les exposer (tableau 3.1) ainsi que de décrire la logique de leur distinction.

Version	Date	Nom original	Nombre de sous-versions	Nombre de commits	Changements principaux
0	21/04/2014	Base	—	168	—
2	13/04/2016	Base2	3	31	<ul style="list-style-type: none"> <li>— Mise en place d'une hiérarchie des attracteurs</li> <li>— Ajout de la création de paroisses urbaines</li> <li>— Ajout de la différenciation entre châteaux et gros châteaux</li> </ul>
3	28/08/2016	Base3	4	39	<ul style="list-style-type: none"> <li>— Réduction des distances de déplacement local</li> <li>— Modification du mécanisme de construction de châteaux</li> <li>— Modification de la logique de calcul de la satisfaction matérielle</li> </ul>
4	25/04/2017	Base4	7	59	<ul style="list-style-type: none"> <li>— Changement de la répartition initiale des foyers paysans</li> <li>— Changement du mécanisme d'héritage/reconnaissance des agrégats</li> <li>— Changement du mécanisme de définition des pôles</li> </ul>
5	11/06/2018	v5	4	60	<ul style="list-style-type: none"> <li>— Modification générale de l'ordonnancement des mécanismes</li> <li>— Modification du mécanisme de définition des agrégats</li> <li>— Modification du calcul de satisfaction des foyers paysans</li> <li>— Mise en place de seuils évolutifs pour le déplacement local</li> </ul>
6	09/01/2019	v6 (22/09/2019)	6	63	<ul style="list-style-type: none"> <li>— Renommage global et refactorisation des paramètres</li> <li>— Suppression des mécanismes de lignage seigneurial</li> <li>— Changement du mécanisme de répartition des nouveaux châteaux</li> <li>— Changement des calculs de probabilité de construction de châteaux</li> </ul>

TABLEAU 3.1 – Historique des versions de SimFeodal.

(SemVer)

À partir de la version « 2 » de SimFeodal<sup>35</sup>, nous avons partiellement suivi les règles de « versionnement sémantique »<sup>36</sup>, c'est-à-dire la numérotation des versions selon une logique de « VersionMajeure.SousVersion.Patch ». ? ) pas très clair

On différencie ainsi des versions selon leur rétro-compatibilité, notamment au niveau de leurs sorties : entre la version 5 et la version 6 de SimFeodal, on retrouve de très nombreuses modifications du code (paramètres, mécanismes, sorties), mais le passage à la version 6 occure quand les paramètres sont renommés et que les données en sortie de la version 6 ne sont donc plus directement organisées de la même manière que celles de la version 5. désin. où ?

Au sein de chaque version, on retrouve plusieurs sous-versions (appelées « versions mineures » dans le modèle de versionnement SemVer), qui apportent toutes des changements suffisants pour justifier un changement de version (évolution de plusieurs paramètres et/ou mécanismes), mais dont la compatibilité avec les versions précédentes est totale.

Au sein de ces sous-versions, on peut encore distinguer des « sous-sous-versions » (« patches » en SemVer), qui concernent cette fois des modifications plus restreintes. Dans l'évolution de SimFeodal par exemple, les sous-sous-versions correspondent à différents essais de changement de paramètres que nous ne sommes pas certains de conserver par la suite. // essais et erreurs. ?

Au niveau de la granularité la plus fine enfin, chaque version, sous-version et sous-sous-version est composée de « commits », c'est-à-dire d'enregistrements historisés du code du modèle à un instant  $t$ . Ces *commits* contiennent un intitulé qui renseigne sur le contenu des modifications apportées, et surtout, ils intègrent une liste précise de ces modifications : les modifications de chaque ligne (de chaque fichier) modifiée sont ainsi conservées. Souvent, les *commits* de SimFeodal<sup>37</sup> sont « atomiques », c'est-à-dire qu'ils ne contiennent et ne documentent qu'un seul changement : paramètre modifié, modification d'un mécanisme, ajout d'une expérience... Cette atomicité n'est pourtant pas constante, et un *commit* peut contenir des centaines de modifications aussi bien qu'une seule. met en évidence

Cette forte hétérogénéité des *commits* renforce un dernier point qui nous paraît important. L'énumération menée dans cette thèse des versions, des sous-versions ou encore de tout changement du modèles, est exprimée selon un ordre chronologique, organisées au sein de la hiérarchie des versions, sous- ?

35. Dans le tableau 3.1, on peut remarquer que les sous-versions ne sont pas dénombrées pour la version 0. Si le code était bien historisé à l'époque, la logique de versionnement n'était pas encore véritablement à l'œuvre.

La version 0 est marquée par une forte instabilité et par de très fréquents changements. On notera d'ailleurs qu'il n'y a pas de version 1. En effet, si la version 0 est la première version « complète » du modèle, c'est-à-dire intégrant tous les agents et mécanismes identifiés pour le modèle conceptuel, c'est la version 2 qui est la première à avoir été un tant soit peu satisfaisante du point de vue de l'évaluation.

36. « Semantic versioning » ou « SemVer », voir <https://semver.org/>.

37. Il y en a plus de 400 à l'approche du rendu de ce travail, consultables dans le dépôt logiciel de SimFeodal :

⌚ <https://github.com/SimFeodal/SimFeodal/commits/master>

versions et patchs. Pour autant, comme indiqué dans le chapitre 1, la réalité de la construction d'un modèle, qui plus est dans un contexte exploratoire et collectif, ne peut s'inscrire nettement dans cette linéarité et dans cet emboîtement de versions. Le paramétrage et la construction d'un modèle, et du notre en particulier, représentent en effet un travail constant d'allers-retours entre l'identification et la résolution de problèmes. Par conséquent, la structure réelle de développement, que l'on peut retrouver dans l'historique des modifications du modèle (les « *commits* »), ne correspond donc pas exactement à la chronologie organisée présentée dans le tableau 3.1. Par exemple, dans les faits, une sous-version 5.3 peut avoir précédé la sous-version 5.1. Ce qui est ici présenté résulte d'une certaine ré-organisation ou re-construction narrative de l'évolution réelle de SimFeodal.

**Exploration de l'évolution d'un modèle exploratoire.** Afin d'appuyer notre discours relatif à la nécessité de mener l'évaluation et le paramétrage de manière continue et systématique à l'évolution d'un modèle de simulation, il nous a semblé approprié d'illustrer ce discours autour de l'analyse de l'évolution d'un modèle de simulation. Le cas de SimFeodal, modèle exploratoire, nous semble tout à fait convenir à une telle analyse exploratoire.

Le travail de qualification de chaque modification du modèle réalisé pour la figure 3.14 étant difficilement extensible à l'ensemble des *commits* de SimFeodal (la figure en présente une centaine, à relativiser vis-à-vis des plus de 400 existants pour lesquels il aurait été nécessaire, de manière systématique, de regarder le contenu et de le catégoriser manuellement entre modifications de paramètres, de mécanismes, d'expériences etc.), il nous a toutefois semblé intéressant d'en mener une analyse plus naïve et exploratoire. ~~Cette recherche~~

Pour mener cette analyse, nous avons analysé de manière automatique chacun des *commits* de SimFeodal, en observant, pour chacun, le nombre de lignes de code qui avaient été modifiées. Une changement d'une ligne de code, par exemple, correspond souvent à la modification de valeur d'un paramètre, alors que la modification d'un mécanisme entraîne le changement d'une dizaine à une cinquantaine de lignes. Pour ne pas biaiser l'analyse avec les changements de la documentation du modèle, ou encore avec les fichiers relatifs à l'exécution d'expériences, nous n'avons conservé que les changements de code effectués dans les fichiers relatifs au cœur du modèle.

Le *commit* est une unité « atomique » mais pas nécessairement homogène, comme dit plus haut, et à ce titre, il arrive souvent qu'un *commit* soit suivi rapidement après d'un second, voir d'un troisième *commit* venant en corriger un tout petit aspect (faute de syntaxe dans le code, oubli de répercussion d'une modification dans les sorties...). Nous avons donc aussi agrégé ces *commits* à l'échelle de la journée, en considérant que les commits d'une même journée de travail formaient un ensemble cohérent.

On a ensuite caractérisé le contenu de ces agrégations de *commits* journaliers en mesurant la taille médiane (en nombre de lignes de code modifiées) et la somme de ces modifications pour chaque jour. La somme des modifications

étant par nature d'un ordre de grandeur bien plus élevé que la médiane<sup>41</sup>, on a normalisé (réduction sans centrage) les valeurs afin qu'elles soient comparables graphiquement. La figure 3.15 est une représentation de cette analyse<sup>42</sup>, qui permet de visualiser le « rythme » des changements de SimFeodal et d'en tirer quelques remarques.

Les paragraphes suivants (+ conclusion) constituent la suite et fin du chapitre et n'étaient pas présents dans la version imprimée.

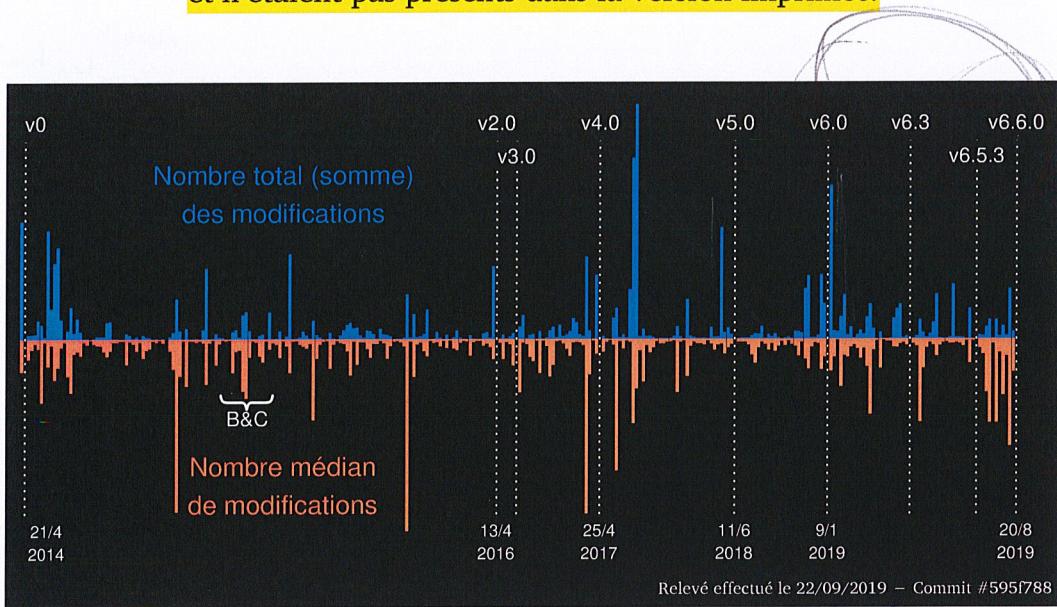


FIGURE 3.15 – Une exploration visuelle du rythme des changements de SimFeodal.

*N.B. : L'axe des abscisses n'est pas linéaire du point de vue temporel : l'ordre chronologique y est respecté, mais le temps n'y est pas régulier.*

On y remarque en premier lieu un certain motif, qui pourrait presque s'apparenter aux pulsations cardiaques, où l'on note des changements majeurs, espacés, entre lesquels de plus faibles phases de modifications surviennent. Ce « rythme » est avant tout une scorie de la manière dont le modèle a évolué, autour des versions représentées sur le graphique.

Les phases de plus intenses modifications, concrétisées ou précédées par des changements de version, sont ainsi intimement liées au rythme de ce travail de co-construction interdisciplinaire : les pics marquent en général la préparation, l'exécution, et les résultats immédiats des séances de travail collectives. Ces réunions étaient ainsi l'occasion d'évaluer le modèle, que ce soit en analysant ses sorties (évaluation externe) ou en ré-exposant les choix d'implémentation (évaluation interne), et en conséquence d'en mener le paramétrage, en faisant évoluer les paramètres et mécanismes. Certains pics correspondent aussi à des moments de production de « livrables », que ce soit à l'occasion de publications ou de communications, où il fallait stabiliser le modèle pour en communiquer les résultats.

41. Le changement médian maximal représente un changement de 300 lignes de codes, alors que la somme concerne, elle, plus de 1500 lignes...

42. Dans cette figure, les modifications regroupées dans l'accolade « B&C » correspondent aux phases B et C identifiées dans la figure 3.14.

À un niveau d'observation plus fin, on peut noter, aussi bien sur la somme des modifications que sur leur médiane, que les pics principaux sont le plus souvent précédés par une augmentation de la taille et de la quantité totale de modifications, puis suivis d'une diminution de ces deux indicateurs. Sur le plan « temporel », cela correspond aux phases de préparation des réunions, faite de petites explorations locales du comportement du modèle en vu de l'adapter, puis aux phases collectives résultant sur des choix de modifications plus importantes, modifications ensuite ajustées et finalement validées, jusqu'à la réunion suivante.

évoquer  
la construction ?

Sur le plan conceptuel, cela correspond à différentes phases de construction du modèle : les modifications qui précèdent les changements de version sont d'abord assez modérées, en médiane comme en somme. Cela correspond à du paramétrage assez modéré, portant sur la modification marginale de quelques valeurs de paramètres. À mesure que la taille des *commits* augmente, ces modifications sont de plus en plus importantes, et consistent ainsi à l'ajustement de plus de paramètres, à la modification plus fréquente de détails de mécanismes, etc.

du résultat de  
simulation plus ?

Les changements de version résultent le plus souvent d'une insatisfaction vis-à-vis de l'aboutissement de ces phases de paramétrage : même en poussant le paramétrage, même en menant un calibrage approfondi on ne parvient pas à obtenir un modèle assez satisfaisant. Il est alors nécessaire de revenir sur le modèle conceptuel et son implémentation (flèches grises de la figure 3.13), et donc de mener des modifications plus conséquentes, par exemple en transformant véritablement le fonctionnement de certains mécanismes. Les mécanismes étant modifiés, avec ajout ou suppression de paramètres par exemple, la rétro-compatibilité des sorties du modèle n'est plus assurée<sup>43</sup>, et l'on change alors de version.

à

Après ces changements de version, la taille des modifications diminue, jusqu'à atteindre des niveaux très faibles : la somme des modifications devient négligeable (à mi-chemin entre les versions 4 et 5 dans la figure 3.15 par exemple) et la médiane des *commits* est elle aussi très faible. Cela correspond exactement aux phases de paramétrages qui suivent les changements majeurs : le nombre de changements diminue, jusqu'à aboutir sur des modifications difficilement discernables qui correspondent au calibrage. Dans cette étape, seules une ou deux lignes de code changent en général, ce qui correspond aux modifications d'une valeur de paramètre.

On retrouve donc dans cette figure, et donc dans l'évolution effective de SimFeodal, l'ensemble des étapes décrites dans ce chapitre. Des phases de construction et d'évaluation se succèdent, avec à chaque fois un travail de paramétrage, parfois suivi de nouvelles phases d'implémentation voire de conception. On notera aussi que les versions sont assez visibles sur ce graphique, correspondant aux pics principaux, de même que les sous-versions, où les pics sont plus modé-

43. Sans même considérer d'ajout/suppression de paramètre, les modifications du sens et de l'utilisation effective de certains paramètres peuvent rendre leurs valeurs non comparables avec celles des versions précédentes.

très intérêtant  
c'est l'intro V il  
faut reviser et le  
def. de taille /  
médiane

rés : entre la version 6 et la fin du graphique (version 6.6), on arrive même sans difficulté à identifier chacune des sous-versions intermédiaires (6.1, 6.2...).

**Aboutissement du paramétrage.** Avec cette analyse exploratoire et l'exercice d'interprétation qui l'accompagne, nous avons voulu illustrer d'une part le processus de paramétrage mis en œuvre dans le modèle SimFeodal, mais surtout, d'autre part appuyer notre proposition de démarche de co-construction de modèles. Qui plus est dans un contexte interdisciplinaire, l'évolution du modèle ne peut ainsi passer que par des allers-retours fréquents entre son évaluation et sa modification.

Le paramétrage, et notamment sa composante de calibrage, ne peut dès lors être réduit à un exercice ultime et unique, qui visera à « finaliser » un modèle avant son emploi. Au contraire, l'application fréquente et régulière de phases d'évaluation et de paramétrage permet de garantir que les différentes pistes de modélisation explorées sont bien comparables, car justement comparées sur des versions aussi finement ajustées que possible à chaque fois.

## Conclusion

Dans ce chapitre, nous avons présenté et proposé une approche originale en matière d'évaluation de modèle de simulation à base d'agents, l'évaluation visuelle, qui nous paraît très adaptée à l'évaluation de modèles exploratoires construits dans un contexte interdisciplinaire. En proposant une évaluation basée sur la multiplication des points de vue, au moyen de nombreux indicateurs de sortie, nous pensons en effet que les différents acteurs de la modélisation peuvent chacun observer et analyser les points qui relèvent de leur expertise thématique propre. La mise en place d'une grille d'évaluation, a priori, et la hiérarchisation des critères permet de plus de garantir une vision globale de la capacité d'un modèle à produire et faire émerger les faits stylisés que l'on cherche à modéliser au sein d'une approche générative.

La forte profusion et diversité des indicateurs de sortie identifiés constitue de plus une aide précieuse au paramétrage d'un modèle : en multipliant les perspectives d'analyse des sorties d'un modèle, on facilite l'identification des mécanismes les moins satisfaisants. Avec des rapports détaillés sur le comportement d'un modèle, il est ainsi plus aisément de déterminer les composantes du modèle qui doivent être améliorées. C'est un avantage indéniable face à l'utilisation d'un indicateur quantifié unique, qui requerrait un passage systématique par une exploration globale de l'influence des paramètres et mécanismes. L'évaluation visuelle que nous proposons dans ce travail permet donc de faciliter, sur un plan conceptuel, le paramétrage, et avec lui les connaissances de chacun des membres impliqués dans la modélisation.

Nous avons toutefois aussi identifié le besoin impératif de mener cette étape de paramétrage aussi fréquemment que possible, à chaque modification du modèle. Cela implique la nécessité de mener tout aussi fréquemment une évaluation du modèle, car c'est par les allers-retours entre l'évaluation et le para-

métrage que le modèle progresse.

Alors que l'évaluation visuelle facilite l'identification des défauts du modèle, elle est en elle-même complexe à mener au regard de méthodes d'évaluation plus quantifiées et automatiques. Pour permettre une bonne progression du modèle, et donc l'augmentation de la fréquence des allers-retours entre évaluation et construction, il faut alors disposer d'outils permettant de faciliter et d'accélérer cette évaluation visuelle. La mise en place de solutions techniques et méthodologiques permettant de mener l'évaluation visuelle d'une manière collective et fluide revêt ainsi une portée cruciale dans la démarche de co-construction de modèle proposée dans ce travail de thèse.