

# Chapitre 3

## Paramétrer un modèle en situation d'interdisciplinarité

Le modèle présenté dans le chapitre précédent est un « état », c'est-à-dire que les mécanismes, paramètres et les valeurs de ceux-ci correspondent à un instantané, potentiellement amené à évoluer. Dans ce chapitre, nous nous attacherons à présenter le travail de paramétrage réalisé jusqu'ici, ayant abouti à la version du modèle tel que décrit auparavant. Par paramétrage, et avant d'en spécifier le sens, nous entendons ici le processus visant à doter le modèle de paramètres (empiriques, mesurables et techniques) lui permettant de mieux convenir aux objectifs fixés, soient-ils en terme de comportements attendus ou d'objectifs quantitatifs.

exaple !  
étape ?

trouver un terme

### 3.1 Paramétrer ? Quoi et quand ?

Le plus souvent (ref Seb, Clem...), une fois le modèle construit, le modélisateur s'attache à son paramétrage, en cherchant pour chaque paramètre la ou les valeurs qui permettront au modèle de s'approcher des données empiriques devant être reproduites. Cette étape, que l'on nomme généralement calibration, peut se faire de manière manuelle, par approximations successives (C. Cottineau et/ou S. Rey, d'après Hermann), par semi-automatisme, par exemple en effectuant des analyses de sensibilité (C. Schmitt, J. Hirtzel), ou encore de manière entièrement automatique (C. Schmitt, S. Rey, C. Cottineau avec PSE). L'approche souvent défendue, notamment dans les travaux les plus récents (Rimbault ?), voudrait que cette étape soit obligatoire pour toute modélisation, permettant par une exploration systématique de comprendre l'entièreté du comportement d'un modèle, et d'en faire dès lors un outil complètement maîtrisé rendant possible l'établissement d'une loi.

) = calibration.  
(et espèce de paramétrage)

) autres ex. ?

la 3e ?

Dans notre travail, nous souhaitons revenir sur cette vision de la modélisation, 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.

expliquer

lequel, le 3e ?

par exemple  
intégrer à fait de  
manière explicite et haché.

Pourquoi ne pas partir d'un  
et ultra simple dans  
un souci de pédagogie ?

### 3.1.1 Qu'est-ce que le(s) paramétrage(s) ?

#### Définition

Le terme recouvre deux sens différents, ont la distinction peut se faire selon qu'on l'utilise pour définir un processus ou pour caractériser une configuration. 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 préférons le terme de configuration de paramètres. 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.

On tend à distinguer le paramétrage presque hygiénique de la calibration, 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 du modèle, mais comme une composante inhérente à la construction d'un modèle, quelque 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.

#### Un modèle, des modèles

Cottineau, Reuillon et Rey ont montré qu'on construisait plus souvent des familles de modèles que des modèles uniques. Ces auteurs plaident pour une construction modulaire de modèles, chacun des mécanismes et complexifications devant être autonomes afin de pouvoir les assembler en autant de combinaisons que nécessaire à une exploration d'ensemble de leurs interactions.

Nous reprendrons de leur discours une vision moins ambitieuse, en considérant (ref Rey) simplement qu'en fait d'un modèle, la modélisation passe par plusieurs modèles, chacun se devant dès lors d'être adapté par un paramétrage dédié.

Pour illustrer ce propos, on peut prendre l'exemple d'une modification qui a eu lieu sur notre modèle au cours de son développement. 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 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

"paramétrage" ?

mieux valoir en le décrire de intro -

lequel ?

pas en précis

Entrez trop "endo", les nb. ont prouvé construction modulaire ex. JASSS

à chercher il en parlait à sa soutenance je crois

pas clair - Nouveau modèle = nouveau mécanisme avec " param ?

Ou param d'avant change aussi ? (oui, le + souvent !) OK, le le dir après ? Mais si mieux introduit



règles. 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. C'est un changement majeur, que l'on peut constater dans la figure 3.1 (1ère modification de la période B), ayant donc entraîné des adaptations de tous les plans du modèle (paramètres en B, mécanismes en C). La première version avait été paramétrée aussi correctement que possible, mais ce paramétrage était entièrement à refaire avec la nouvelle version (modifications de la période B, fig. 3.1). On pourrait dès lors différencier le modèle pré-existant et celui qui a suivi ce changement, et les considérer comme deux modèles puisque réagissant de manière extrêmement différente.

La on ne suit pas,  
mais les évol  
sont expliquées dans  
des modèles ?

à discuter  
+ profondément

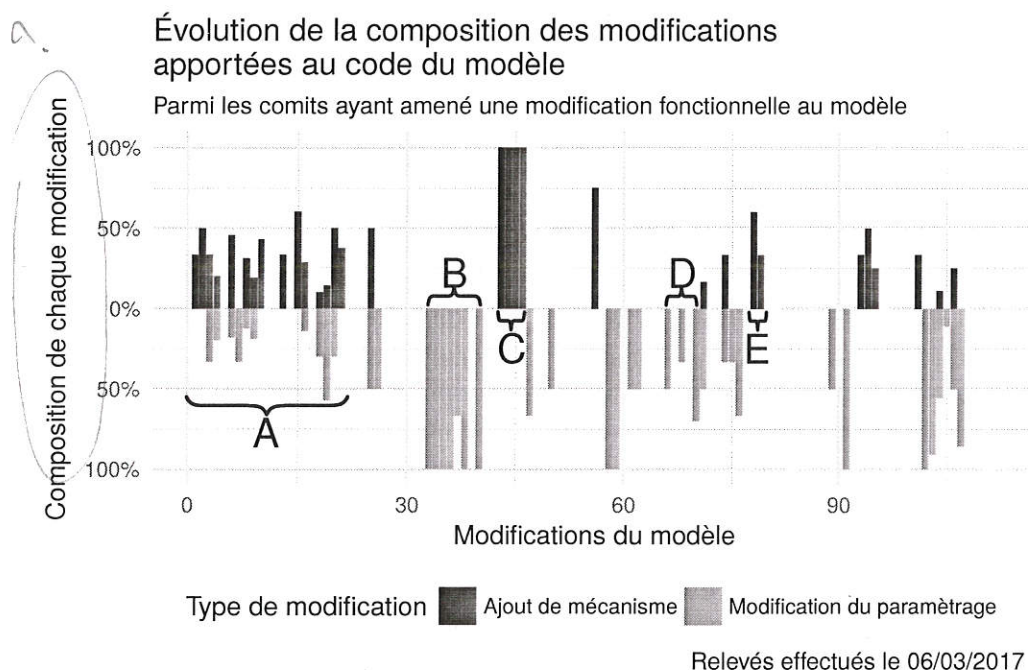


FIGURE 3.1 Temporalité du paramétrage du modèle.

Chaque « enregistrement » (*comit*) correspond à une sauvegarde du modèle, contenant un ou plusieurs changements. On a ici rapporté le nombre de changements de chaque type au nombre total de changements de chaque *comit* afin de démarquer des types de *comits*.

pas d'air !

Cet exemple renforce le caractère nécessaire du paramétrage, et qui plus est, de la continuité de cette étape, qui ne peut être pensée que comme une calibration finale d'un modèle abouti, auquel on ne peut en fait parvenir que par des paramétrages réguliers de modèles successifs moins aboutis.

?

### 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, la validation, l'évaluation ou encore la calibration. Nous définissons dans l'encadré 3.1 le sens donné à chacun de ces termes dans le cadre de ce manuscrit.

**Encadré 3.1 : Calibration, évaluation, validation...****Calibration**

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 à la 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. Cette étape, quelques soient les moyens employés, s'approche de la résolution utilisée dans le cadre de systèmes d'équations. Le contexte des systèmes complexes, et donc d'une non-linéarité des effets des paramètres, rend toutefois difficile l'obtention d'une unique configuration de paramètres optimale, et la calibration aboutit donc souvent à un ensemble de configurations possibles, constituant par exemple un optimum de Pareto (Trouver ref).

**Évaluation**

On emploie majoritairement ce mot pour décrire les méthodes permettant de comprendre le comportement du modèle et sa réaction aux différents paramètres. Là où la calibration cherche une configuration de paramètres optimale, l'évaluation tend surtout à caractériser la stabilité du modèle face à l'aléa ou à des configurations exceptionnelles. On vise ainsi à s'assurer que le modèle reproduise les faits stylisés voulus quelque soient son réglage, ou au moins à quantifier les intervalles de paramètres qui y satisfont.

**Validation**

La validation (refs Seb) tire son origine de sciences plus nomothétiques, et correspond donc à la démonstration qu'un modèle reproduit correctement ce qu'il représente. Au delà de l'évaluation, le terme amène une logique de preuve formelle que le modèle réagit bien ainsi et pas autrement quelque soient les conditions d'exécution. Cette démonstration formelle peut être effectuée sur des modèles à faible nombre de paramètres et mécanismes, mais le terme n'est jamais (à vérifier) employé dès lors que les modèles se complexifient.

**3.1.2 Quand paramètre-t-on un modèle ?**

Le paramétrage du modèle n'est pas une étape unique, qui se déroulerait une fois le modèle construit, d'une manière complètement automatisable, permettant l'aboutissement d'un modèle désormais finalisé. Au lieu de cela, le paramétrage se fait par étapes successives, chaque modification des hypothèses, des mécanismes, des objectifs ou encore des paramètres amenant à une nouvelle phase de paramétrage. Afin que les ??comportements du modèle soient cohérents avec les attendus, il convient donc de paramétrer le modèle aussi souvent que possible, à chaque modification de chacune de ses parties. Un premier paramétrage, souvent non mentionné, tient ainsi lieu lors de la construction en elle-même du modèle, et vise en particulier à doter les paramètres techniques

Il faut partir d'une def. + barrière à l'intro

pas en la d.?

q. JASSS  
Faire rappel du sens de chaque état ?

Indiquer que la pratique, dans le contexte de la 1<sup>re</sup> développée (SMA, intro d.?), elle n'est pas ainsi.

pas clair.  
à mettre avant le début de phase



de valeurs permettant aux autres paramètres de donner satisfaction avec des ordres de grandeur cohérents.

ref vers chap2 et encadré types de paramètres

### Tout au long de la construction

Comme le montre la fig. 3.1, les étapes de paramétrage sont omniprésentes sur la durée de construction du modèle. On peut en effet y noter que les *comits* du code portant sur des modifications de valeurs de paramètres sont très réguliers. Au delà de la valeur illustrative de ce modèle particulier, c'est un comportement que l'on retrouve régulièrement sur les modèles dont l'évolution est documentée, ce que le recours de plus en plus fréquent à des outils de versionnement (git en particulier) encourage.

La régularité et le systématisme du paramétrage sont bénéfiques à la création d'un modèle en ce qu'ils impliquent une utilisation fréquente du modèle, et une utilisation plus poussée : le paramétrage implique de tester plusieurs valeurs, et donc d'exécuter le modèle de nombreuses fois, d'autant plus en prenant en compte les réplifications nécessaires à la prise en compte de l'aléa d'un modèle.

Une analogie peut être faite avec un principe populaire dans le développement de logiciel libre, résumé par le mot d'ordre « *release early, release often* »<sup>1</sup>. Selon ce principe, une publication régulière et rapide du code source d'un logiciel permet d'avoir un retour rapide de la communauté des utilisateurs, et ainsi de faire remonter d'éventuels *bugs* et régressions bien plus rapidement que dans une voie plus classique.

Un modèle, dès qu'il devient suffisamment complexe pour que les modifications n'apportent pas que des changements prévisibles, est souvent sujet à des « effets de bord », c'est-à-dire des comportements irrationnels spécifiques qui n'arrivent que dans certaines conditions particulières. Comme dans n'importe quel logiciel informatique, les *bugs* les plus compliqués à résoudre sont ceux dont on ne comprend pourquoi ou quand ils se déclenchent. Les modèles réagissent de la même manière, provoquant de temps en temps des erreurs ou des comportements aberrants sans que l'on soit en mesure de les reproduire facilement. Un paramétrage régulier entraîne une plus forte probabilité de survenance de ces cas particuliers, et dès lors, une meilleure résistance des mécanismes.

### À chaque ajout ou changement de mécanisme

De manière plus spécifique, et pour les mêmes raisons qu'évoquées auparavant, il est indispensable de réaliser un paramétrage à chaque fois que les mécanismes de mobilisation d'un paramètre changent.

Le paramétrage permet de s'assurer de la cohérence des mécanismes et paramètres les uns avec les autres. Que le modèle soit construit par une seule personne ou par plusieurs, il est plus simple de corriger une erreur ou un comportement inattendu quand les modifications depuis le dernier état stable sont faibles plutôt que quand de nombreux changements ont été effectués. En effet, la résolution d'erreurs nouvelles dans un modèle se résume souvent à activer et désactiver les ajouts, un par un, afin de trouver la modification

1. Mot d'ordre en particulier popularisé par Eric Raymond, l'un des « évangélistes » du logiciel *open source*.

responsable. Dans un modèle complexe où les mécanismes et paramètres sont en partie interdépendants, quand plusieurs modifications sont effectuées, les petites erreurs et incompatibilités de chacune se combinent en des problèmes tout aussi complexes que les systèmes modélisés. On gagne donc énormément de temps de correction du modèle en identifiant régulièrement les problèmes, ce qu'un paramétrage à chaque changement permet.

Le recours régulier à un paramétrage remplit ainsi exactement le même rôle que les « bancs de test » de l'industrie, c'est-à-dire qu'il permet d'évaluer (cf. encadré 3.1) un modèle à chacune des modifications, s'assurant donc de sa qualité tout au long et minimisant le risque d'erreurs et donc la durée des phases de paramétrages plus importants.

### Avant toute modification majeure du modèle

Si le paramétrage a été régulier dans la construction du modèle, on note toutefois, dans la figure 3.1, que certaines phases (période B en particulier, mais aussi D) se caractérisent par une forte tâche de paramétrage avant une période de changement (C et E respectivement). Cela correspond à une phase d'exploration plus poussée du modèle, en vue de le faire évoluer par la suite, par exemple en complexifiant les mécanismes à l'oeuvre pour le rendre plus conforme aux attentes.

Quand le paramétrage n'est pas bien exécuté avant ces changements, les mécanismes à adapter sont moins clairs et moins bien compris, et un travail plus important d'essai-erreur est requis, comme les premiers temps de la construction du modèle (période A) l'illustrent : faute de bonne compréhension du modèle, l'évolution est plus poussive, dans un aller-retour constant entre paramétrage et ajout/modification de mécanismes.

Ce principe s'inscrit dans les méthodes de développement qualifiées d'« agiles », qui insistent en particulier sur l'incrémentalité de la construction : à chaque étape, aussi peu différente soit-elle de la précédente, le logiciel (ou le modèle ici) doit être fonctionnel, et on ne passe pas à une version suivante tant que ce n'est pas le cas.

Je reprendrais ça en chap 7 dans un encadré différenciant bien itératif/incrémental cf Agile Development Environment @ ravisr16

### 3.1.3 Comment paramétrer ?

#### Visual validation

(Trouver ref dans thèse Clémentine, sans doute Hermann encore)

Indicateurs

L'importance de la réplication

## 3.2 Premiers paramétrages

### 3.2.1 Au fil de la construction

Paramétrage technique

Ajout de paramètres

### 3.2.2 Paramétrer *in situ*

Paramétrage empirique

Changements majeurs

### 3.2.3 Première exploration systématique

## 3.3 Premiers résultats du paramétrage

### 3.3.1 Tendances

### 3.3.2 Variabilité

## 3.4 Comment traiter les sorties du/des modèle(s) ?

### 3.4.1 Nombre de sorties

### 3.4.2 Masse de sorties

