



HAL
open science

Un métamodèle de calcul en temps continu pour les systèmes d'aide à la décision appliqués à la planification financière

Davy Héléard Héléard

► **To cite this version:**

Davy Héléard Héléard. Un métamodèle de calcul en temps continu pour les systèmes d'aide à la décision appliqués à la planification financière. Ingénierie, finance et science [cs.CE]. Université de Bretagne Sud, 2015. Français. NNT : 2015LORIS379 . tel-01316887

HAL Id: tel-01316887

<https://theses.hal.science/tel-01316887>

Submitted on 17 May 2016

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

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



THESE / UNIVERSITE DE BRETAGNE-SUD

sous le sceau de l'Université européenne de Bretagne

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITE DE BRETAGNE-SUD

Mention : STIC

Ecole doctorale SICMA

présentée par :

Davy Héléard

Préparée à l'Unité Mixte de Recherche CNRS 6074
IRISA - Institut de recherche en informatique et systèmes
aléatoires dans le cadre d'une convention CIFRE

**Un métamodèle de calcul en temps continu
pour les systèmes d'aide à la décision
appliqués à la planification financière**

Thèse soutenue le 1^{er} décembre 2015 devant le jury
composé de :

Pr Pierre-François MARTEAU, Université de Bretagne Sud –
UMR IRISA (président)

Pr Henri BASSON, Université du Littoral Côte d'Opale – EA
LISIC (rapporteur)

Pr Mourad OUSSALAH, Université de Nantes – UMR LINA
(rapporteur)

Pr Flavio OQUENDO, Université de Bretagne Sud – IRISA
(directeur de thèse)

M. Jean-Philippe GOUIGOUX, Directeur Technique MGDIS
(encadrant entreprise de la CIFRE)

Mme Dalila TAMZALIT, Université de Nantes – UMR LINA
(examineur)

M. Frank MOSSER, Directeur Général, MGDIS (examineur
invité)

Résumé

Dans le cadre de l'informatique décisionnelle, la programmation physico-financière doit permettre à des acteurs d'une collectivité provenant de divers domaines de faire converger leurs problématiques vers un objectif commun. Cette programmation permet ainsi à ces acteurs d'étudier les conditions de réalisation et de mettre en place un suivi de l'avancement de cet objectif.

L'une des principales difficultés de la modélisation d'une programmation physico-financière est que chaque acteur exprime ses problématiques dans des échelles de temps différentes. Leur mise en lien dans un modèle représentant une réalité commune pose donc des problèmes au mode de représentation discret traditionnellement utilisé par les outils d'analyse financière, basés sur la logique des tableurs.

Dans cette thèse CIFRE, un métamodèle de calcul en temps continu appliqué à la programmation physico-financière est proposé afin de permettre aux acteurs de regrouper leurs visions dans un modèle unique, tout en se plaçant sur des échelles de temps différentes. La modélisation continue développée est confrontée à la modélisation discrète (représentative de l'état de l'art) au travers d'un cas d'étude, montrant les avancées de la première vis-à-vis de la seconde.

Ce métamodèle innovant a été implémenté au sein de la société MGDIS, dans le cadre d'une convention CIFRE à l'aide d'une architecture orientée service. Cette architecture se base sur un style innovant conçu dans cette thèse afin de maximiser la capacité à paralléliser l'évaluation des modèles. Ce prototype est constitué de services web pour modéliser (service de modélisation) et évaluer les modèles (service de calcul).

La solution développée dans cette thèse a été conçue pour permettre la programmation physico-financière de gros volumes de données à l'échelle réelle. Elle a été validée sur un cas d'étude et répond aux exigences exprimées par les experts de la modélisation de programmation physico-financière de MGDIS qui ont émis un avis positif quant à l'applicabilité de la solution proposée.

Abstract

In the scope of Business Intelligence, planning aims to support multiple actors in their process of converging different views and problematics from different domains to get a shared business planning model. It is in particular the case of business planning in local government.

A major difficulty in business planning is that each actor states her/his views and problematics with a different time scale. Integrating them into a unique model that represents a common state of reality becomes very costly and awkward to manage when basing the construction of these models on discrete modeling techniques used by current tools of business planning.

This doctorate thesis proposes a novel solution, beyond the state-of-the-art, for addressing these issues: it conceives a novel metamodel based on a continuous time calculus. Through the developed approach, it allows multiple actors to integrate the different business logics of their planning domain in a shared model as well to observe it from different time scales. The advantages of our solution based on continuous time against solutions based on discrete time are presented through a case study.

The conceived metamodel was implemented within a real industrial set in MGDIS (a company specialized in business planning for local governments) following an innovative service oriented architecture: this architecture segregates the modeling from the evaluation to allow the parallelization of model evaluation for big volumes of data.

The overall solution conceived and implemented in this thesis was designed to be a real scale prototype to be applied to real scale problems. Besides the case study, it was validated by MGDIS experts on business planning against real requirements.

Remerciements

Je remercie :

- mon encadrant entreprise de ma thèse CIFRE Jean-Philippe Gouigoux (directeur technique de MGDIS) et mon directeur de thèse CIFRE Flavio Oquendo (professeur à l'UBS, directeur de recherche à l'UMR IRISA) pour leurs conseils et leur soutien tout au long de ces 3 années ;
- Emmanuel Frenod (professeur à l'UBS/LMBA) qui est à l'origine de l'intérêt de MGDIS pour l'application de la modélisation en temps continu à la programmation physico-financière, pour ses explications sur les facettes mathématiques de la modélisation en temps continu ;
- Frank Mosser (DG de MGDIS) et Jean-Marc Bienvenu (PDG de MGIS) pour avoir décidé de la tenue de cette thèse CIFRE à MGDIS, ainsi que pour leur soutien ;
- Tarik Chakkour (doctorant à l'UMR LMBA) qui a développé le calculateur numérique utilisé par mon prototype de preuve de concept ;
- Gildas Boichot (développeur à MGDIS) pour son exposition du produit SOFI au niveau du besoin et de l'implémentation ;
- Jean-Marc Bienvenu pour sa participation à l'élaboration du cas d'étude de programmation physico-financière ;
- Guénael Jamette (chef de projet à MGDIS), Jean-Marc Bienvenu et Frank Mosser pour leur relecture de ma thèse avec une vision applicative du point de vue de la programmation physico-financière ;
- les rapporteurs Henri Basson (professeur à l'EA LISIC) et Mourad Oussalah (professeur à l'UMR LINA) pour leur relecture avec une vision scientifique, leurs rapports ainsi que pour leurs suggestions ;
- les membres du jury Pierre-François Marteau, Henri Basson, Mourad Oussalah, Flavio Oquendo, Jean-Philippe Gouigoux, Dalila Tamzalit et Frank Mosser pour leur écoute et leurs remarques ;
- également toutes les personnes de MGDIS et de l'IRISA pour leur accueil.

Table des matières

1	Introduction	1
1.1	La problématique	1
1.1.1	L'informatique décisionnelle	1
1.1.2	Positionnement de la thèse.....	2
1.1.3	La programmation physico-financière	2
1.1.4	L'exemple d'une gestion de cantine.....	5
1.1.5	Les difficultés à lever	11
1.1.6	Les questions ouvertes.....	13
1.2	La genèse de la thèse	13
1.2.1	L'approche par la prévalence objet.....	13
1.2.2	L'approche par la parallélisation	15
1.2.3	La nécessité de penser continu	15
1.2.4	L'apport de l'urbanisation des SI.....	17
1.3	Le sujet de la thèse.....	18
1.3.1	Les objectifs métier	18
1.3.2	Les objectifs intellectuels	19
1.3.3	Les objectifs techniques	19
1.3.4	Le sujet scientifique.....	20
1.3.5	Le sujet industriel	20
1.4	Le contexte.....	21
1.4.1	Société MGDIS.....	21
1.4.2	Les travaux associés	22
1.5	L'organisation du manuscrit.....	22
2	L'état de l'art	23
2.1	Les approches historiques.....	23
2.1.1	Les approches par cubes de données	23
2.1.2	L'optimisation par contraintes	23
2.1.3	L'optimisation par algorithmes génétiques.....	24
2.2	La revue systématique	25
2.2.1	Formulation des questions de recherche.....	25
2.2.2	La réalisation	26
2.2.3	Recherche et sélection des études pertinentes	28
2.2.4	Evaluation de la qualité des études retenues	29

2.2.5	Extraction des données pertinentes et analyse de ces données.....	29
2.2.6	Interprétation des résultats vis-à-vis des questions de recherche.....	31
3	Un cas d'étude	33
3.1	Contexte	33
3.2	La planification de l'investissement	33
3.2.1	L'établissement de la fiche d'opérations	34
3.2.2	La recherche de subventions.....	34
3.2.3	La recherche d'emprunt.....	34
3.2.4	La planification physique.....	34
3.2.5	Le choix de l'investissement de construction de la nouvelle cantine	34
3.3	Le suivi du budget d'investissement	36
3.3.1	La construction du bâtiment	36
3.3.2	Les subventions	36
3.4	Le suivi du budget de fonctionnement	37
3.4.1	Le paiement par les familles.....	37
3.4.2	La gestion des repas	37
3.4.3	Les subventions de la CAF au fonctionnement	37
3.4.4	Le bilan financier	37
3.4.5	Ouverture de la nouvelle cantine.....	37
4	L'approche de solution et sa mise en œuvre	38
4.1	Le découpage en services.....	38
4.1.1	Le besoin	38
4.1.2	La solution avec SOFI discret	38
4.1.3	La solution avec SOFI continu.....	39
4.1.4	La comparaison des solutions	43
4.1.5	Les évolutions possibles	43
4.2	L'évaluation des modèles de planification financière	43
4.2.1	Le besoin	43
4.2.2	La solution avec SOFI discret	44
4.2.3	La solution avec SOFI continu.....	44
4.2.4	La comparaison des solutions	45
4.2.5	Les évolutions possibles	45
4.3	La parallélisation du moteur de calcul	45
4.3.1	Le besoin	45

4.3.2	La solution avec SOFI discret	45
4.3.3	La solution avec SOFI continu.....	46
4.3.4	La comparaison des solutions	47
4.3.5	Les évolutions possibles	48
4.4	La séparation de la logique du domaine et des données propre à un cas	48
4.4.1	Le besoin	48
4.4.2	La solution avec SOFI discret	48
4.4.3	La solution avec SOFI continu.....	49
4.4.4	La comparaison des solutions	54
4.4.5	Les évolutions possibles	54
4.5	L'instanciation d'éléments	54
4.5.1	Le besoin	54
4.5.2	La solution avec SOFI discret	54
4.5.3	La solution avec SOFI continu.....	55
4.5.4	La comparaison des solutions	55
4.5.5	Les évolutions possibles	55
4.6	La complexité des évaluations	55
4.6.1	Le besoin	55
4.6.2	La solution avec SOFI discret	55
4.6.3	La solution avec SOFI continu.....	56
4.6.4	La comparaison des solutions	56
4.6.5	Les évolutions possibles	56
4.7	La distinction entre les mesures et les champs.....	57
4.7.1	Le besoin	57
4.7.2	La solution avec SOFI discret	57
4.7.3	La solution avec SOFI continu.....	57
4.7.4	La comparaison des solutions	59
4.7.5	Les évolutions possibles	59
4.8	La modélisation des durées.....	59
4.8.1	Le besoin	59
4.8.2	La solution avec SOFI discret	59
4.8.3	La solution avec SOFI continu.....	59
4.8.4	La comparaison des solutions	60
4.8.5	Les évolutions possibles	60

4.9	La modélisation des équations.....	60
4.9.1	Le besoin	60
4.9.2	La solution avec SOFI discret	60
4.9.3	La solution avec SOFI continu.....	60
4.9.4	La comparaison des solutions	69
4.9.5	Les évolutions possibles	69
4.10	La délimitation du passé et du futur	69
4.10.1	Le besoin	69
4.10.2	La solution avec SOFI discret	69
4.10.3	La solution avec SOFI continu.....	71
4.10.4	La comparaison des solutions	72
4.10.5	Les évolutions possibles	72
4.11	L'évaluation des découpages temporels des calculs.....	72
4.11.1	Le besoin	72
4.11.2	La solution avec SOFI discret	73
4.11.3	La solution avec SOFI continu.....	73
4.11.4	La comparaison des solutions	75
4.11.5	Les évolutions possibles	76
4.12	Les valeurs saisies pour le passé et éventuellement le futur	76
4.12.1	Le besoin	76
4.12.2	La solution avec SOFI discret	76
4.12.3	La solution avec SOFI continu.....	76
4.12.4	L'exemple du gain de finesse de la saisie des ventes de tickets.....	78
4.12.5	La comparaison des solutions	80
4.12.6	Les évolutions possibles	80
4.13	L'estimation du futur à partir du passé.....	80
4.13.1	Le besoin	80
4.13.2	Les approches.....	81
4.13.3	L'exemple du remboursement de la TVA	82
4.13.4	L'exemple de l'estimation des ventes de tickets.....	83
4.13.5	L'exemple de l'estimation du coût des denrées alimentaires.....	89
4.13.6	La comparaison des solutions	92
4.13.7	Les évolutions possibles	92
4.14	Le calcul des emprunts et des amortissements	92

4.14.1	Le besoin	92
4.14.2	La solution avec SOFI discret	92
4.14.3	La solution avec SOFI continu.....	93
4.14.4	La comparaison des solutions	96
4.14.5	Les évolutions possibles	96
4.15	Conclusion	96
5	Expérimentation et validation.....	97
5.1	L'implémentation de la solution	97
5.1.1	La bibliothèque de calcul continu LEMF.....	97
5.1.2	Les technologies utilisées pour implémenter les services.....	98
5.2	La validation	98
5.2.1	La construction du modèle du cas d'étude	98
5.2.2	L'utilisation du modèle du cas d'étude	107
5.2.3	La mesure de complexité de l'évaluation du cas d'étude	112
5.2.4	La réponse aux questions de recherche.....	112
6	Conclusion et perspectives	114
6.1	Le bilan	114
6.1.1	Complétion de la thèse	114
6.1.2	Validation des résultats.....	114
6.1.3	Adéquation à la problématique	115
6.2	Le positionnement : contributions à l'état de l'art	115
6.3	Les perspectives	115
6.3.1	L'ergonomie de l'interface graphique pour modélisation continue native.....	115
6.3.2	La parallélisation et branchement sur optimisation par algorithmes génétiques	116
6.3.3	L'optimisation par inversion du modèle continu	116
6.3.4	La détermination en temps réel de la robustesse du modèle financier.....	117
	Références	119
	Annexe 1 : L'interface du service de modélisation	121
	Annexe 2 : Le modèle complet de la planification financière	123

Table des figures

Figure 1 : Les liens entre les domaines de programmation	5
Figure 2 : Les sources utilisées pour la recherche de publications	26
Figure 3 : La requête pour la recherche de publications.....	26
Figure 4 : La requête pour ScienceDirect	27
Figure 5 : La requête pour IEEE	27
Figure 6 : La requête pour SpringerLink	27
Figure 7 : La requête pour ACM	28
Figure 8 : Le nombre de résultats obtenus à partir des sources	28
Figure 9 : Liste des études primaires.....	29
Figure 10 : Le diagramme des études primaires par domaine d'application	30
Figure 11 : La liste des études primaires par domaine d'application.....	30
Figure 12 : Le diagramme des résultats obtenus par approche.....	31
Figure 13 : Les résultats obtenus par approche	31
Figure 14 : La vue d'ensemble du budget d'investissement et de fonctionnement	33
Figure 15 : La communication avec le service de modélisation et de calcul.....	39
Figure 16 : L'opération du contrat du service de calcul	40
Figure 17 : Le diagramme de classe de la partie des requêtes de calcul qui permet de décrire les modifications à apporter à un scénario	41
Figure 18 : Le diagramme de classe de la partie du contrat qui décrit les variables à évaluer et le résultat de l'évaluation	42
Figure 19 : Le diagramme d'instance d'une requête d'évaluation de l'équilibre du budget de fonctionnement après modification du montant de l'emprunt du budget d'investissement	42
Figure 20 : Le diagramme d'instance du résultat d'une requête d'évaluation de l'équilibre du budget de fonctionnement	43
Figure 21 : Le diagramme d'instance de la structure d'une variable traduite en calcul continu	45
Figure 22 : La parallélisation du service de calcul	46
Figure 23 : La vue d'ensemble du schéma de la base de données de la modélisation de SOFI discret	49
Figure 24 : La vue d'ensemble du métamodèle de la modélisation de SOFI continu	50
Figure 25 : La contrainte entre les variables d'instances et leur variable	52
Figure 26 : La contrainte entre les associations et leur lien.....	53
Figure 27 : Un diagramme d'instance illustrant le besoin d'agrèger les variables de lien	54
Figure 28 : La formule de la variable « Accumulation de la participation de la mairie » en modélisation discrète	56
Figure 29 : L'interprétation de la valeur d'une mesure	58
Figure 30 : L'interprétation de la valeur d'un champ.....	58
Figure 31 : Un exemple de formule de calcul de date	59
Figure 32 : La comparaison de formule de calcul d'un décalage de deux mois ou de deux fois un mois	59
Figure 33 : Un exemple de formule pour une variable « Equilibre » en modélisation discrète.....	60
Figure 34 : Un exemple de formule pour une variable « Equilibre » en modélisation continue	60
Figure 35 : Les contraintes sur le type des variables continues en fonction des opérateurs arithmétiques.....	61
Figure 36 : La signature de l'opération « If »	61
Figure 37 : Un exemple de formule pour une variable « Dépenses »	61

Figure 38 : La signature de l'opération « Delay ».....	62
Figure 39 : Un exemple d'application de l'opération « Delay »	62
Figure 40 : Une formule de décalage en modélisation continue	62
Figure 41 : La modélisation du décalage avec SOFI continu	63
Figure 42 : Evaluations d'un décalage avec SOFI continu	63
Figure 43 : La signature de l'opération « SumBy ».....	63
Figure 44 : Un exemple d'application de l'opération « SumBy »	63
Figure 45 : Une formule de concentration avec l'opération « SumBy »	64
Figure 46 : La signature de l'opération « SumOver »	64
Figure 47 : Un exemple d'application de l'opération « SumOver »	64
Figure 48 : Une formule de concentration avec l'opération « SumOver »	65
Figure 49 : La signature de l'opération « Accumulate ».....	65
Figure 50 : Un exemple d'application de l'opération « Accumulate »	65
Figure 51 : Une formule de concentration avec l'opération « Accumulate »	65
Figure 52 : La signature de l'opération « Repeat »	66
Figure 53 : Un exemple d'application de l'opération « Repeat »	66
Figure 54 : Une formule de concentration avec l'opération « Repeat »	66
Figure 55 : La formule de la variable « Remboursement de la TVA » en modélisation continue	67
Figure 56 : La formule du début de l'année du présent.....	67
Figure 57 : La grammaire des formules de calcul des variables continues	67
Figure 58 : Le diagramme de classes de l'arborescence syntaxique des formules de calcul	68
Figure 59 : La grammaire des formules de calcul des variables scalaires	68
Figure 60 : Le diagramme de classes de l'arborescence sémantique des formules de calcul.....	69
Figure 61 : La représentation du présent par la modélisation de SOFI discret	70
Figure 62 : Une date qui sépare les périodes mensuelles ne permet pas nécessairement de séparer les périodes annuelles.....	70
Figure 63 : Le calcul de l'année du présent avec une périodicité mensuelle	71
Figure 64 : La représentation du présent par la modélisation de SOFI continu.....	71
Figure 65 : Le découpage du temps par année ou par mois	71
Figure 66 : La formule du début de l'année du présent.....	71
Figure 67 : La formule du début de l'année du présent.....	72
Figure 68 : La représentation graphique de la formule d'un opérateur « If » avec une frontière	73
Figure 69 : La formule de la variable « Participation à l'équilibre » qui tient compte du déménagement.....	73
Figure 70 : Le tableau de calcul de la participation en tenant compte du déménagement.....	73
Figure 71 : La représentation graphique de la formule d'une fonction « If » avec une frontière.....	73
Figure 72 : Le graphe de calcul de la participation qui prend en compte le déménagement.....	74
Figure 73 : Intervalles d'évaluation des variables derrière une opération « Piecewise »	74
Figure 74 : Une formule d'une fonction « If » avec deux frontières	74
Figure 75 : La représentation graphique de la formule d'une fonction « If » avec deux frontières	75
Figure 76 : Une formule d'une imbrication de fonction « If ».....	75
Figure 77 : L'aplatissement d'une imbrication de fonction « If ».....	75
Figure 78 : Valeurs ponctuelles	76
Figure 79 : Valeurs avec une répartition constante	76
Figure 80 : Un exemple de patron par valeurs ponctuelles	77

Figure 81 : Un exemple de patron par assemblage d'une forme.....	77
Figure 82 : Le diagramme de classes des patrons de valeur.....	78
Figure 83 : Le patron de répartition estimée des ventes de tickets au cours d'une année scolaire.....	79
Figure 84 : La saisie semaine par semaine de la répartition réelle des ventes de tickets au cours d'une année scolaire.....	80
Figure 85 : Une formule de référence au passé en modélisation discrète.....	81
Figure 86 : La modélisation du décalage lié à la période d'évaluation.....	81
Figure 87 : La formule de la variable « Remboursement de la TVA » en modélisation discrète.....	82
Figure 88 : La modélisation du décalage lié à la période d'évaluation.....	82
Figure 89 : Les 2 étapes du calcul de la variable « Remboursement de la TVA » en modélisation continue.....	83
Figure 90 : La formule de la variable « Remboursement de la TVA » en modélisation continue.....	83
Figure 91 : La formule de la variable « Participation des familles » pour une périodicité annuelle en modélisation discrète.....	84
Figure 92 : Les 2 étapes du calcul de la variable « Participation des familles » en modélisation continue.....	85
Figure 93 : La formule de la variable « Quantité de tickets vendus » en modélisation continue.....	85
Figure 94 : La formule de la variable « Participation des familles » pour une périodicité mensuelle en modélisation discrète.....	86
Figure 95 : Les variables annuelles peuvent utiliser des variables mensuelles, mais l'inverse n'est pas possible.....	86
Figure 96 : La répartition réelle des ventes de tickets.....	87
Figure 97 : L'interpolation de la vente de tickets par année.....	87
Figure 98 : La moyenne des répartitions de ventes de tickets.....	87
Figure 99 : Les 2 étapes du calcul de la variable « Participation des familles » en modélisation continue avec une répartition plus précise.....	88
Figure 100 : La formule de la variable « Coût d'un repas » pour une périodicité annuelle en modélisation discrète.....	89
Figure 101 : La formule de la variable « Coût d'un repas » pour une périodicité mensuelle en modélisation discrète.....	89
Figure 102 : L'estimation du coût des denrées avec une méthode analogue à celle utilisée pour SOFI discret.....	89
Figure 103 : La décomposition des variations du coût des denrées en plusieurs facteurs.....	90
Figure 104 : La formule du coût indépendant de l'inflation ramené au 1 ^{er} Janvier de l'année du présent.....	90
Figure 105 : La courbe de répétition du coût d'un repas indépendant de l'inflation.....	90
Figure 106 : La formule de répétition du coût d'un repas indépendant de l'inflation.....	91
Figure 107 : La courbe de l'estimation du coût des denrées d'un repas.....	91
Figure 108 : La formule de la variable « Coût d'un repas » en modélisation continue.....	91
Figure 109 : Un diagramme d'instance du calcul de l'amortissement avec LEMF.....	93
Figure 110 : Les courbes d'un calcul d'amortissement.....	94
Figure 111 : Les courbes d'un calcul d'amortissement avec une incertitude.....	94
Figure 112 : Un diagramme d'instance du calcul de l'emprunt avec LEMF.....	95
Figure 113 : Les courbes d'un calcul d'emprunt à amortissement constant.....	96
Figure 114 : Le diagramme de classes de l'interface de la bibliothèque de calcul continu.....	97

Figure 115 : La vue d'ensemble du budget d'investissement et de fonctionnement	98
Figure 116 : Le diagramme d'instance montrant l'instanciation des budgets	99
Figure 117 : Le diagramme d'instance montrant le lien entre le budget d'investissement et de fonctionnement au niveau du total d'amortissement	100
Figure 118 : Le diagramme d'instance montrant la modélisation du basculement entre les deux budgets de fonctionnement lors déménagement	100
Figure 119 : Le diagramme d'instance montrant les dépendances entre les éléments du modèle ...	101
Figure 120 : Le diagramme d'instance montrant la modélisation du calcul de la participation des familles	102
Figure 121 : Le patron de répartition des ventes de tickets au cours d'une année scolaire.....	102
Figure 122 : Le diagramme de classes du patron de répartition des ventes de tickets au cours d'une année scolaire	103
Figure 123 : La courbe d'évolution des ventes de tickets de 2014 à 2016.....	103
Figure 124 : Le diagramme d'instance montrant la saisie de l'évolution des ventes de tickets de 2014 à 2016.....	103
Figure 125 : La formule de la variable « Quantité de tickets vendus » en modélisation continue	104
Figure 126 : Le diagramme de classes du patron de répartition des ventes de tickets au cours d'une année scolaire	104
Figure 127 : La courbe d'évolution du prix d'un ticket de Septembre 2013 à Septembre 2015.....	104
Figure 128 : Le diagramme d'instance montrant la saisie de l'évolution du prix d'un ticket de 2014 à 2016	105
Figure 129 : La formule de la variable « Achats d'aliments » en modélisation continue.....	105
Figure 130 : La formule de la variable « Coût d'un repas » en modélisation continue.....	105
Figure 131 : Le diagramme d'instance montrant la modélisation du calcul de la participation à l'équilibre de la mairie	106
Figure 132 : La formule du début de l'année du présent.....	106
Figure 133 : Le diagramme d'instance montrant le calcul de la somme des dépenses d'une opération	106
Figure 134 : Le diagramme d'instance montrant le calcul du montant à amortir	107
Figure 135 : La formule de la variable « Remboursement de la TVA » en modélisation continue	107
Figure 136 : Une requête de calcul évaluant la participation de la mairie	108
Figure 137 : Le cumul des économies réalisées par la commune avec la solution à haut rendement énergétique.....	108
Figure 138 : Une requête de calcul évaluant l'impact du prix des tickets sur la participation de la mairie	109
Figure 139 : Une requête d'importation du fichier XML du modèle.....	109
Figure 140 : Une opération de modélisation pour actualiser les valeurs affectées aux dépenses	109
Figure 141 : Une opération de modélisation pour actualiser la date de la fin d'une opération	109
Figure 142 : Une opération de modélisation pour actualiser la date de déménagement	109
Figure 143 : Une requête d'exportation du modèle en fichier XML	109
Figure 144 : Une opération de modélisation pour actualiser les valeurs affectées à la subvention d'investissement	110
Figure 145 : Une opération de modélisation pour instancier un emprunt	110
Figure 146 : Une opération de modélisation pour associer un emprunt au budget d'investissement	110

Figure 147 : Une opération de modélisation pour définir la valeur du capital emprunté	111
Figure 148 : Une requête de calcul évaluant l'impact des décisions sur l'équilibre du budget de fonctionnement	111
Figure 149 : Une opération de modélisation pour modifier la valeur des dépenses en achats d'aliments	112
Figure 150 : Une capture d'écran du prototype d'interface graphique	114
Figure 151 : L'interface de l'implémentation du service de modélisation avec WCF	122

1 Introduction

La gestion des finances d'une entité publique dans le temps (c'est-à-dire en rétrospective des montants passés, analyse de la situation présente et simulation prospective du futur), est une activité complexe et peu standardisée. Cette complexité se reflète dans la grande hétérogénéité des méthodes proposées par les cabinets de conseil en gestion des finances des collectivités publiques. Chaque consultant utilise ses propres classeurs et feuilles de calcul reflétant son expérience et l'état financier de ses clients, et il n'y a que très peu de standardisation. La comptabilité (c'est-à-dire la représentation écrite des flux financiers) est normalisée, mais la façon dont sont effectivement gérées les finances (et notamment du point de vue prospectif et programmatique) ne l'est pas.

Les causes racines de cette complexité sont une difficulté réelle à prévoir l'évolution d'une situation financière soumises à de nombreuses contraintes endogènes et exogènes, une sémantique complexe, l'intervention de différents acteurs (décideurs, financiers, opérationnels internes et externes...) avec des rôles et des objectifs très différents voire parfois opposés et enfin une difficulté de conceptualisation de l'axe temporel. La société MGDIS édite un logiciel de stratégie et programmation financière nommé SOFI, et qui permet de remplacer des approches ad hoc matérialisées dans des feuilles de calcul par un modèle générique de programmation physico-financière. Parfaitement adaptée aux approches « simples » de type élaboration d'une PPI (programmation pluriannuelle des investissements) portée par la direction des finances en réponse à une obligation réglementaire ou une volonté stratégique, l'outil trouve sa limite dès que l'on souhaite étendre son usage vers le pilotage financier au fil de l'eau en intégrant la complexité du dialogue de gestion entre financiers et opérationnels.

Après plusieurs essais infructueux réalisés par son équipe interne de recherche et développement, MGDIS a réalisé que l'atteinte de l'objectif précédemment cité ne se ferait qu'au prix d'une marche importante dans l'approche conceptuelle du problème et d'un effort de formalisme sortant de son cadre de R&D interne traditionnelle. Il a donc été décidé de lancer une thèse CIFRE sur le sujet en partenariat avec l'UMR IRISA.

1.1 La problématique

1.1.1 L'informatique décisionnelle

Les premières apparitions de l'informatique en entreprises ont été constatées principalement dans la gestion de la production (en particulier pour la gestion de stock) et dans la gestion financière (typiquement en comptabilité). Rapidement, le management s'est rendu compte du potentiel pour ses métiers propres. Ainsi, l'informatique a commencé à être utilisée pour la gestion des ressources humaines (paie, principalement), mais aussi pour le pilotage de l'entreprise dans son ensemble. Une bonne exploitation des données provenant des différents services permet un suivi plus efficace de l'activité de l'entreprise et d'améliorer son fonctionnement continuellement.

L'utilisation des données informatiques comme une aide à la décision des responsables d'entreprise est donc naturellement née quelques années après la première phase d'informatisation industrielle. Ce mouvement est constaté dans pratiquement toutes les branches de l'industrie. Bases de données, ETL (outils d'extraction, transformation et chargement de données), datamarts et autres systèmes spécialisés se sont rapidement développés comme outils d'accompagnement sur le sujet.

L'informatique décisionnelle, en particulier sous son vocable anglais de Business Intelligence, est très fortement associée aux outils traditionnels qui la servent, en particulier les cubes de données (cf. chapitre 2.1.1). Il est toutefois essentiel de bien faire la différence entre le but et les moyens.

Ces derniers sont aujourd'hui fortement remis en cause. Des études comme celle menées par Gartner en 2011 ont montré que sur un panel de grandes entreprises, 85% n'avaient pas réussi à exploiter leurs données de manière à créer un avantage concurrentiel significatif. En 2010, Le domaine de la BI est passé de la première à la cinquième place dans les priorités des décideurs informatiques (classement Gartner également).

Toutefois, il est clair pour tous les décideurs d'entreprise que le but n'a pas disparu et que l'informatique doit continuer à fournir des moyens de mieux piloter les entreprises. Simplement, il semble nécessaire que les outils et les pratiques soient fortement améliorés pour mieux y participer. Plusieurs approches sont actuellement développées. Un premier ensemble de méthodes est regroupé sous le terme générique de Big Data. Il se base sur des avancées techniques comme la meilleure maîtrise des clusters de données, les algorithmes prédictifs et la gestion des données sous forme de flux pour améliorer la connaissance métier apportée par les données. Une autre approche consiste à repartir des conditions d'échec de la Business Intelligence dans son appellation courante, et de tenter de les lever. C'est dans cette approche que se place cette thèse.

1.1.2 Positionnement de la thèse

L'expérience de MGDIS sur plusieurs dizaines d'années a montré que, dans le domaine de la programmation physico-financière, les principales difficultés concernant l'exploitation décisionnelle des données informatiques étaient la qualité des données entrantes et le faible niveau de modélisation métier des requêtes.

La qualité de la donnée est reconnue comme étant fondamentale (la citation « garbage in, garbage out » est désormais proverbiale parmi les personnes intervenant dans le domaine de la BI). Elle reste pourtant le parent pauvre de l'analyse décisionnelle. Ceci est certainement dû au fait que l'amélioration de la qualité de la donnée est une tâche rébarbative. En effet, peu d'outils performants sont disponibles et le travail nécessite de nombreuses interactions humaines. D'ailleurs, ce travail est généralement réalisé par des équipes techniques sans concerter les personnes du domaine concernées par ces données (ceci constitue d'ailleurs une autre raison de l'échec relatif de la BI), etc.

La mauvaise modélisation des questions vient également du fait que les équipes sollicitées pour la mise en œuvre de la BI en entreprise sont souvent purement techniques, les environnements matériels et logiciels étant complexes (le domaine est longtemps resté la vache à lait des éditeurs logiciels, des SSII et des fournisseurs de matériels haut de gamme). Les méthodes exposées pour interroger les cubes de données formés n'étaient donc pas construites en fonction des besoins des personnes censées exploiter la solution technique, ce qui conduisait à une inadéquation et au final à une non-appropriation des systèmes déployés.

Cette thèse se positionne sur les deux points d'échec cités. Elle propose une modélisation très proche des décideurs et gestionnaires financiers afin de réduire le fossé entre l'outil et ses utilisateurs, ainsi qu'un outil commun aux différents profils afin de garantir une donnée de qualité. Les techniques habituellement associées à la Business Intelligence sont décrites par la suite, bien qu'elles n'aient pas réussi à porter toute la valeur souhaitée, afin de mieux positionner le sujet de la thèse dans la progression intellectuelle de MGDIS.

1.1.3 La programmation physico-financière

1.1.3.1 Le domaine

Le domaine d'application de cette thèse est la gestion des finances des collectivités publiques en lien avec leur action de terrain et plus précisément, la programmation physico-financière. Bien que le

cadre soit restreint, de par l'activité de la société MGDIS, aux organismes publics, les notions et approches mises en œuvre sont utilisables telles quelles dans un cadre privé et de manière générale, dans n'importe quel domaine gérant des finances dans le temps.

Les clients finaux ciblés par le logiciel SOFI sont les organismes publics au sens large avec un positionnement important sur les communes, les EPCI, les centres hospitaliers et des entités plus importantes comme les conseils régionaux ou départementaux. Certaines de ces entités répondent à des obligations légales de projection de leurs finances dans le temps, avec des horizons différents selon leur taille, d'autres s'y attachent dans une volonté de bonne gestion ou en anticipation d'obligations à venir.

1.1.3.2 Une définition

La programmation physico-financière consiste à décrire un projet de développement et de le confronter à une prévision de financement. L'objectif est de pouvoir réaliser au mieux les missions de l'entité en maîtrisant les équilibres financiers entre charges et ressources, avec pour finalité un axe politiquement fort sur la gestion du risque financier (respect des contraintes prudentielles). La récente crise des financements des entités publiques a eu pour conséquence que l'Etat exige de plus en plus de ces dernières qu'elles se projettent dans un futur économiquement incertain en analysant les impacts des aléas financiers.

Au-delà de la gestion de risque, il s'agit bien sûr également d'assurer une bonne gestion de l'argent public. Les collectivités publiques, en plus de répondre à des réglementations prudentielles, sont aussi engagées dans des approches de rationalisation des dépenses. Cette rationalisation a pour objectif de servir au mieux les usagers tout en maintenant un endettement limité et des marges de manœuvre financières. L'actualité récente montre que ces problématiques sont de plus en plus connues par le grand public. Cette exposition, et son impact électoral potentiel, va de fait rendre encore plus importante la maîtrise des finances publiques.

La finalité des projets à réaliser est de produire des services aux usagers. En effet, l'actualité focalise souvent son analyse des finances des collectivités publiques sur le contrôle de celles-ci, mais le versant « positif » de la finance est la mise en œuvre de projets au service des citoyens. Du point de vue des élus, ceci est souvent nommé « programmes » ; du point de vue des gestionnaires en charge de leur réalisation, c'est le mot « opérations » qui est utilisé, et qui sera évoqué à de nombreuses reprises.

1.1.3.3 Une décomposition en trois couches

Dans le cas d'une collectivité locale, une programmation est réalisée par la concertation des acteurs de trois domaines : la programmation physique, financière et budgétaire. Ces trois couches correspondent à une répartition systématiquement observée par les entités soumises à l'obligation de programmation.

Cette répartition est la manifestation opérationnelle de la répartition des tâches à laquelle aboutit toute activité humaine suffisamment formalisée et rôdée. Elle n'est pas issue d'une quelconque loi, mais de la longue pratique de la prospective financière qui a peu à peu mené à cette organisation consacrée désormais comme une pratique standard. Ces trois couches sont définies plus précisément dans les paragraphes suivants.

1.1.3.3.1 Le niveau physique

Le niveau physique concerne la réalisation effective du service aux usagers et citoyens. La programmation physique consiste principalement à planifier le déroulement d'opérations physiques.

Le coût de ces opérations physiques est estimé afin d'établir le lien avec la finance, en se basant sur la facturation et la constatation de dépenses.

Afin de rendre difficiles les malversations, les gestionnaires au niveau physique n'ont en général pas accès à des moyens de paiement ou si, c'est le cas, sous des formes contrôlées leur interdisant de les détourner de leur usage approuvé par le niveau supérieur.

Au niveau physique, les concepts les plus utilisés sont ceux d'opérations, de devis, de travaux, de fournisseur, de bons et de factures. Ces activités sont limitées aux actions qui ne nécessitent pas de manipulation financière directe : les devis sont émis ou reçus, les travaux sont planifiés et suivis ou livrés, les fournisseurs et les factures sont gérés. En aucun cas un devis ou une facture n'est effectivement payée par les personnes intervenant au niveau physique.

Dans le cadre de la séparation entre l'ordonnateur et du payeur le paiement est réalisé sur justification au niveau du Trésor Public.

La responsabilité principale des gestionnaires au niveau physique est de veiller au bon déroulement des opérations financées par l'entité, aussi bien du point de vue de la planification, du niveau de réalisation ou du niveau de consommation des ressources financières mises à leur disposition dans le respect des contrats signés.

1.1.3.3.2 Le niveau financier

Le niveau financier est le plus proche de la stratégie, en l'occurrence celui sur lequel arbitrent les élus. La programmation financière consiste à planifier les dépenses et les recettes de façon à obtenir un équilibre qui respecte les ratios prudentiels et réglementaires. Le niveau financier est le plus abstrait, responsable des grands équilibres.

Dans le cadre de la séparation des tâches, les gestionnaires financiers ne sont pas responsables du bon déroulement des opérations physiques qu'ils financent, mais uniquement de la mise à disposition d'un budget tel que voté par les élus, tout en respectant les équilibres financiers. Le niveau financier est le plus abstrait dans la programmation physico-financière, mais en dehors de cette classification, il est lui-même piloté par le niveau stratégique, en l'occurrence dans les collectivités publiques par le niveau politique. Ce sont les élus qui donnent les orientations financières découlant des orientations politiques sur lesquelles ils ont été élus.

1.1.3.3.3 Le niveau budgétaire

Le niveau budgétaire est un niveau intermédiaire entre le niveau physique et le niveau financier. C'est à ce niveau que la stratégie financière se décline en une gestion opérationnelle des ressources financières afin de permettre la réalisation des opérations physiques. Il s'agit de décider comment répartir les ressources monétaires entre les différents services pour qu'ils puissent fonctionner correctement dans un cadre de budget annualisé. La programmation budgétaire consiste à planifier les crédits de paiement et autorisation de paiements votés.

Le débat budgétaire est d'une certaine manière le cœur de la politique, dans le sens où il transforme des orientations financières abstraites en une répartition effective des ressources monétaires à destination de telle ou telle ensemble d'opérations concrètes. Le niveau budgétaire est l'articulation de concrétisation entre le niveau financier qui est symbolique et le niveau physique qui est concret.

Les gestionnaires de budget sont avant tout responsables de la bonne retranscription des orientations financières dans les mouvements effectifs des ressources monétaires. Ils doivent mettre en lien les paiements des devis, travaux et autres opérations physiques avec les montants qui ont été

votés, en assurant en permanence la cohérence entre les deux. Le niveau budgétaire est le curseur glissant ajustant en permanence la stratégie avec l'opérationnel.

1.1.3.4 Les interactions

Les acteurs de la programmation physique, financière et budgétaire ont, comme ceci a été vu précédemment, des rôles différents. Il est donc essentiel que ces rôles travaillent en concertation pour que les trois niveaux soient correctement alignés sur un compromis partagé et réaliste. Une bonne prospective financière n'a aucun sens si le niveau physique réalise au-delà de ce qui est prévu. A l'inverse, une planification opérationnelle de qualité sera rapidement mise hors service si le financement ne suit pas. Il est donc essentiel d'assurer une cohérence entre les trois niveaux, et les rôles étant différents, ceci nécessite une concertation.

Au cours de cette concertation, dont le sommet est le débat d'orientation budgétaire, les acteurs sont amenés à prendre de nombreuses décisions concernant leur niveau de programmation, en particulier au niveau de la planification temporelle de ce niveau. Comme les trois planifications programmées sont liées entre elle, chaque décision prise concernant une planification nécessite des ajustements des deux autres, que ce soit sur des valeurs financières ou des critères temporels. Ces ajustements peuvent soulever des problèmes qui nécessitent de prendre d'autres décisions et ainsi de suite jusqu'à atteindre un compromis acceptable.

Ces différentes opérations d'ajustement sont décrites dans le schéma ci-dessous :

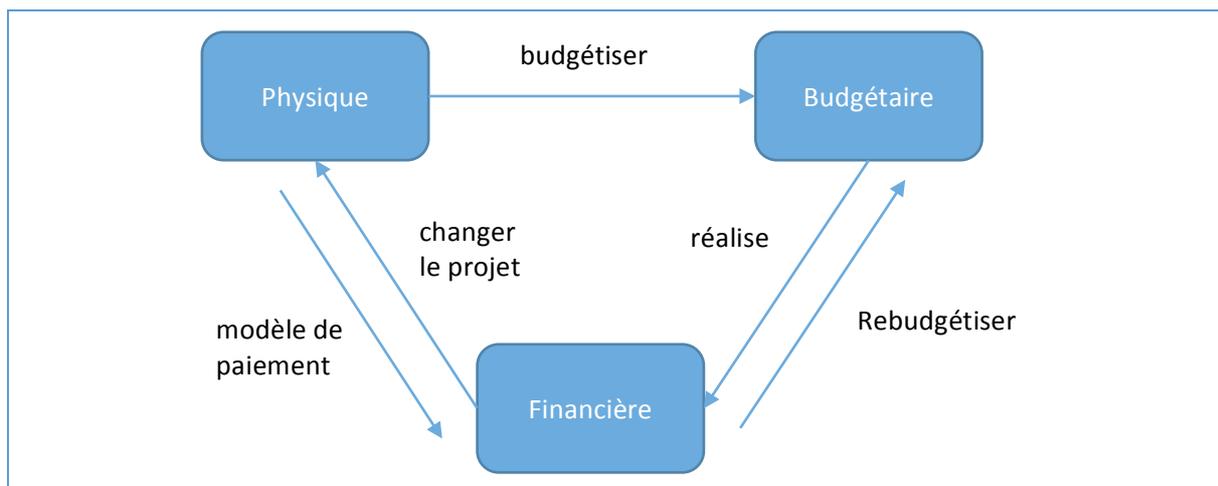


Figure 1 : Les liens entre les domaines de programmation

1.1.4 L'exemple d'une gestion de cantine

1.1.4.1 La programmation physico-financière d'une cantine

Afin de matérialiser les concepts sous-jacents à la présente étude, un exemple concret a été conçu pour accompagner au mieux les analyses. Cet exemple regroupe un ensemble varié de situations tout en restant le plus succinct possible. Il est présenté dans ce chapitre introductif, mais également détaillé par la suite, pour mettre en lien chacune des situations avec une fonctionnalité du modèle conceptuel créé pour répondre à la problématique. La modélisation de cet exemple selon l'approche proposée est présentée dans le chapitre 5.2. Cette modélisation d'un cas d'usage réaliste permet de tester la réalisation.

Après analyse de quelques situations ne permettant pas d'envisager de manière réaliste toutes les situations financières d'une collectivité publique, le cas de la gestion d'une cantine scolaire a été retenu.

Cet exemple possède de nombreux avantages. Tout d'abord, il procède d'une sémantique aisément compréhensible par une personne même étrangère à la programmation physico-financière. Nombreux sont en effet les individus qui ont bénéficié des services d'une cantine. Les concepts associés sont également facilement appréhendables (en tout cas au point de vue opérationnel). En effet, physiquement, une cantine ne fait que proposer à grande échelle les réalisations qu'un individu est capable d'opérer dans sa cuisine familiale. Cependant, au niveau du budget, la complexité est bien plus importante que celle du cadre familial. Quant au niveau financier, il est complètement différent : l'approche de stratégie et de vote est complètement étrangère au domaine domestique. Pourtant, le seul fait de rendre plus compréhensible les deux premiers niveaux rend l'exemple accessible et lui permet de faciliter la compréhension de la modélisation de la programmation physico-financière dans un logiciel comme SOFI.

L'exemple de la cantine est également pertinent car il permet d'introduire de manière réaliste des aléas sur les trois niveaux de la programmation financière. C'est en effet tout l'intérêt de l'utilisation d'outils informatiques que de prendre en compte rapidement des modifications du modèle. Si les aléas avaient été quasi-inexistants, la gestion de la programmation physico-financière serait certainement restée au niveau papier et n'aurait pas subi l'importante automatisation des dernières décennies. Or, la gestion d'une cantine dans la pratique expose à de nombreux aléas constatés au jour le jour par les clients de MGDIS dans leur gestion financière. Ces exemples d'aléas sont décrits dans les sections ci-dessous qui détaillent l'exemple sur les trois niveaux de programmation.

1.1.4.2 Les détails du niveau physique

La gestion d'une cantine scolaire au niveau physique de la programmation physico-financière fait intervenir les concepts physiques suivants :

- les repas, qui sont servis chaque jour à des élèves ;
- les personnels de gestion, en charge de la réalisation des repas, que cela soit de la manière la plus opérationnelle (cuisiniers) ou de la manière la plus abstraite (intendant), et dont le salaire représente une part importante du budget ;
- les fournisseurs, échangeant les denrées de base nécessaire à la création des repas contre une rémunération par l'entité en charge de la cantine ;
- les bâtiments, utilisés par les gestionnaires pour le service des repas, et qui sont également sources de dépenses de fonctionnement (électricité, eau, etc.) ;
- les matériels de cuisine (pour la préparation ainsi que pour le service), dont le financement et l'amortissement font partie des dépenses ;
- les élèves eux-mêmes, qui sont les consommateurs des repas, et par-delà leurs parents ou tuteurs, qui sont les fournisseurs d'une partie des recettes de la cantine, par l'achat des tickets ouvrant droit aux repas ;
- les services de mairie, qui fournissent souvent les bâtiments et participent financièrement aux coûts de la cantine, que ce soit en fonctionnement ou en investissement.

La mise en œuvre de ces concepts, et en particulier les évolutions qu'ils subissent dans le temps de manière prévisible (prospective) ou pas (aléas), va nous permettre de valider le fonctionnement correct de la modélisation informatique de la programmation physico-financière qui est le but de SOFI et, par-delà, de cette thèse. Nous citons ci-dessous quelques exemples de caractéristiques physiques qui influenceront sur le modèle financier :

- modulation du prix du repas en fonction des saisons et de la disponibilité des différents types de denrées (fruits abondants en été, légumes abondants en été et dans certains cas à l'automne, etc.) ;

- variation de l'activité en fonction des périodes scolaires : absence de repas lors des vacances, fermeture complète de la cantine sur l'été ;
- variation ponctuelle du coût, pour des repas de fêtes par exemple ;
- augmentation des salaires des personnels, recrutements et autres opérations sur les personnels ;
- déperdition de consommables, vieillissement du matériel, etc. ;
- opération de construction d'un nouveau bâtiment et mise en œuvre de nouveaux équipements, dans le cadre d'un programme politique des élus ;
- mise en place d'une politique de produits bio, influant sur les fournisseurs et les coûts, et nécessitant ainsi un arbitrage politique pour la temporalité de la mise en œuvre ;
- etc.

A nouveau, un des gros avantages de l'exemple de la cantine scolaire est que toutes les raisons de modification du modèle physique citées ci-dessus sont facilement compréhensibles par les lecteurs de ce document, quel que soit leur connaissance préalable du domaine de la programmation physico-financière des collectivités publiques. C'est un avantage important pour une présentation des concepts complexes associés à la réalisation pratique de la thèse aux clients de MGDIS.

1.1.4.3 Les détails du niveau financier

Au niveau le plus haut dans l'abstraction, à savoir le niveau financier, la gestion de la cantine scolaire consiste en un équilibre entre les dépenses et les recettes. Un déficit étant souvent considéré comme normal sur ce genre de service. L'équilibre est réalisé par des ressources en dehors du budget de la cantine lui-même. Typiquement, ce sont les impôts (à travers le financement d'une subvention d'équilibre) qui viennent compenser le déséquilibre. Au niveau du budget de la cantine, l'équilibre local est atteint soit par l'injection de ces crédits sous forme de subvention soit par une participation implicite (résultant du déficit constaté).

Le niveau financier est décomposé en un équilibre de fonctionnement et un équilibre d'investissement.

L'investissement correspond à toutes les dépenses et recettes non-ordinaires, c'est-à-dire ponctuelles dans la vie de la cantine, et qui ne vont pas se reproduire d'un budget (annuel dans le cas présent) à l'autre. Dans le cas d'étude, il peut s'agir de la construction de nouveaux bâtiments pour la cantine et de leur équipement. Cette dépense est exceptionnelle, et correspond à un investissement qui sera rentabilisé par une meilleure efficacité économique ou un niveau de qualité de service rendu supérieur. Des recettes sont également portées dans le budget d'investissement afin de financer les dépenses par exemple les subventions, les emprunts et l'autofinancement.

Le fonctionnement concerne toutes les dépenses et recettes courantes. Dans le cas de la cantine, il s'agit donc des salaires, des paiements des fournisseurs et autres dépenses récurrentes, ainsi que des ressources financières issues de la vente de tickets de cantine. Le budget de fonctionnement est équilibré en tenant compte des amortissements et des emprunts liés à la construction de la nouvelle cantine.

Les deux niveaux de budget doivent être équilibrés chacun de leur côté, même si les interactions sont bien sûr fortes entre les deux niveaux. Typiquement, la mise en œuvre d'un emprunt bancaire pour financer la construction ponctuelle d'un nouveau bâtiment va entraîner le remboursement récurrent pour plusieurs années de budget des mensualités ou autres traites.

Au niveau financier, il ne s'agit que de trouver les grands équilibres, et en aucun cas de détailler toutes les lignes des budgets. C'est le niveau budgétaire qui gère le lien entre ces grands équilibres

votés par les élus et la gestion au jour le jour de la cantine. Au niveau financier, les budgets servent à connaître les coûts globaux de fonctionnement d'une année de cantine (ainsi qu'éventuellement la saisonnalité) pour décider si un nouveau bâtiment se justifie (parmi bien sûr de nombreux autres critères de décision) et le meilleur moment pour mettre en œuvre l'opération. Des arbitrages budgétaires peuvent également avoir lieu au niveau financier sur des augmentations de salaire, car on touche alors aux grandes masses et les élus comme les préparateurs financiers doivent être consultés, pour décision et prise en compte financière, respectivement.

1.1.4.4 Les détails du niveau budgétaire

Dans le cas d'étude choisi, ce niveau est représenté par le budget de gestion de la cantine par la municipalité. La restauration scolaire est en effet placée sous la responsabilité de la commune, et c'est donc elle qui en gère le budget, même si une grande partie des entrants est fournie par l'intendant de la cantine. La commune porte la définition du budget et son entretien, mais participe également financièrement à ce budget par la dotation à la cantine, ainsi que par le salaire des employés qui sont généralement municipaux.

La description du niveau financier a un peu anticipé sur celle du niveau budgétaire, car les deux sont évidemment en lien, et ne peuvent bien être expliqués qu'en décrivant leurs différences constitutives. Pour revenir sur la description réalisée plus haut des interactions, il convient d'expliquer comment le niveau budgétaire géré par le gestionnaire doit produire une articulation acceptable entre le niveau financier (les grandes masses de dépenses autorisées votées par les financiers) et le niveau physique (les dépenses et recettes constatées au niveau opérationnel). Le budget a pour délicate mission de trouver une solution dans le temps qui concilie les deux exigences.

Le gestionnaire doit vérifier que les dépenses et les recettes progressent comme prévues, de façon à ne pas dépasser le montant autorisé une fois en fin d'exercice. En cas d'imprévu, le gestionnaire trouvera souvent une solution pour ne pas dépasser le déficit prévu. Les déséquilibres locaux sont pris en charge par le financier de la commune qui gère la trésorerie. Dans le cas contraire, des autorisations de paiements supplémentaires peuvent être votées par les élus.

Le gros du travail budgétaire, surtout sur une cantine où les dépenses récurrentes représentent la part majeure, est donc plutôt dans l'anticipation. La préparation budgétaire permet au financier de prévoir les emprunts et les lignes de trésoreries. Le niveau budgétaire doit donc justifier à la fois le niveau global des fonds demandés (en fonction de l'augmentation du coût de la vie, du niveau acceptable par les élus d'augmentation du coût du ticket de cantine, etc.) mais aussi les variations de celui-ci au cours de l'année (demande de budget supplémentaire en Décembre pour le repas de Noël, diminution drastique pendant les vacances d'été, etc.).

De manière générale, une fois le budget de la cantine scolaire préparé, le gestionnaire suit son budget et n'a pas besoin de solliciter le niveau financier avant l'année suivante. Il existe toutefois des cas d'aléas, typiquement liés aux investissements, qui peuvent toutefois nécessiter de remonter au niveau financier. Imaginons par exemple que pendant les travaux de terrassement nécessaire à la construction de la nouvelle cantine, un bloc de granit soit mis au jour. Il faudra peut-être faire venir un engin de travaux publics spécifiques pour le briser, et cela peut grever sérieusement le budget, ainsi que décaler les plannings, ce qui entraîne encore d'autres coûts (moindre recette, conséquences en chaîne, etc.). Ces aléas peuvent également être opérationnels, avec la création d'une classe supplémentaire à la rentrée, due à l'afflux de nouveaux élèves sur l'été, comme cela est souvent constaté dans les communes en fort développement (première couronne de villes majeures, typiquement). Au final, ces aléas sont finalement tellement fréquents que, bien que non prédictibles

dans leur nature, on peut s'attendre à deux ou trois décisions modificatives par an, surtout lorsque des opérations d'investissement sont engagées.

1.1.4.5 Les interactions

Après avoir détaillé le fonctionnement des trois niveaux de la programmation physico-financière pour notre cas d'étude, il convient de compléter cette description par l'analyse des interactions au sein de ce modèle. Nous avons anticipé en expliquant quelques liens entre les différents niveaux, mais les interactions ont également lieu au niveau des acteurs, de la planification, etc., ce qui apporte au final une complexité bien plus importante. Par contre, comme cela est systématiquement le cas dans les systèmes informatiques, c'est bien l'interaction entre les entités qui apporte le plus de richesse au système. Du point de vue de la modélisation, les relations entre les composants sont plus importantes que les composants eux-mêmes pour une représentation informatique rentable du flux de travail.

Ces interactions se manifestent de plusieurs manières, que nous allons détailler ci-dessous.

1.1.4.5.1 Les interactions de planification

L'approche temporelle est absolument fondamentale dans la modélisation des processus de décision financière. Il s'agit même du seul invariant de la modélisation fournie par cette thèse. Les techniques mises en œuvre peuvent tout à fait servir dans d'autres modèles que les finances publiques, et même que la programmation financière tout court. Par contre, la modélisation proposée est systématiquement associée à un axe principal représentant le temps. Ceci reflète l'importance fondamentale de l'approche temporelle dans la programmation physico-financière, qui se manifeste dans la pratique par le positionnement central de la planification dans les interactions entre acteurs du domaine.

Comme vu précédemment, la programmation financière possède deux sous domaines : la programmation du budget d'investissement et du budget de fonctionnement. Chacun de ces domaines est planifié sur une échelle de temps en rapport à leur objectif. La programmation budgétaire est réalisée sur une échelle annuelle puisque les crédits et autorisations de paiement sont votés chaque année. Le choix d'échelle de temps pour la programmation physique dépend de ce qui est planifié. Pour une construction de cantine, la planification est réalisée sur une échelle mensuelle. Concernant la planification financière, le budget d'investissement est fortement lié à la planification physique. Il utilise donc la même échelle. L'échelle choisie pour le budget de fonctionnement dépend du domaine d'application. Dans le cas d'une cantine, une échelle hebdomadaire est utilisée pour la planification du budget de fonctionnement car elle s'adapte bien au cycle alternant entre jours ouvrés et weekends.

Logiquement, les plus fréquentes des interactions autour de la programmation physico-financière sont liées à la programmation. Par exemple, une décision concernant l'efficacité énergétique du bâtiment de la nouvelle cantine appartient au domaine de la programmation physique. Cette décision impacte sur la durée et le coût des travaux. Les subventions et emprunts qui couvrent les dépenses du budget d'investissement sont planifiés en conséquence par la programmation financière. L'endettement affecte le budget de fonctionnement de la cantine à cause des intérêts. Cet endettement peut être réduit par une participation financière de la municipalité. Cette participation est votée et planifiée par la programmation budgétaire.

Les planifications à un niveau sont reprogrammées de nombreuses fois en fonction des deux autres niveaux de planifications. Afin de permettre une communication fluide entre les acteurs, la reprogrammation d'une planification doit être prise en compte pour les autres planifications le plus rapidement possible. C'est dans cette reprogrammation qu'un outil informatique est

particulièrement utile, car la gestion temporelle de budgets sur des échelles différentes de temps est un réel défi intellectuel. Certaines recompositions sont simples (mois vers trimestre, par exemple, qui consiste en un simple cumul), mais d'autres particulièrement épineuses (semaines vers mois, typiquement).

1.1.4.5.2 Les interactions de communication

Comme cela a été vu précédemment, les programmations physique, financière et budgétaire dépendent les unes des autres et évoluent de nombreuses fois au cours de la programmation d'une planification. Afin de permettre une communication fluide entre les acteurs, les modèles de ces trois domaines doivent évoluer ensemble de façon automatique. Cependant, comme chacun de ces domaines possède une logique et des échelles de temps différentes, leur coévolution n'est pas triviale. Par exemple, le gestionnaire de la cantine prévoit les repas sur une échelle hebdomadaire alors que la direction des finances planifie le budget sur une périodicité annuelle.

Se pose alors une seconde catégorie de problèmes, moins importante que celle regroupant la problématique de planification mais toutefois significative, à savoir la gestion des interactions de communication. En effet, avant de recouper les valeurs entre les différents budgets, il convient de gérer les signaux comme quoi un budget a évolué, de traiter ce signal dans les autres niveaux, de faire comprendre de manière non ambiguë les modifications effectivement réalisées, de donner les moyens de prendre les bonnes décisions de modification en conséquence et au final d'émettre en retour un signal éventuel de traitement. Ce signal rentrera alors dans la même boucle de traitement, et ce mouvement peut très bien être récursif.

La difficulté traditionnelle sur ce genre d'interactions est qu'elles se passent dans un média complètement décorrélé du support informatique de la modélisation. Par exemple, le gestionnaire constate une hausse des prix des denrées sur son dernier devis fournisseur, et va utiliser le téléphone pour signaler le changement au responsable du budget, qui va trouver une solution pour réaffecter temporairement des ressources. Dans le cas où le niveau supérieur doit être sollicité car le delta est trop grand et nécessite un vote de crédits supplémentaires, le gestionnaire du budget utilisera peut-être lui aussi un média différent, comme le courriel, et enverra des chiffres qui seront saisis au niveau financier dans un autre classeur de calcul ou un autre logiciel de gestion financière.

Toutes ces ruptures entre les médias de communication et les moyens de traitement de l'information potentiellement différents à chaque niveau posent des problèmes de communication : envoi à la mauvaise personne, personnel en congé et ne traitant donc pas la donnée, erreur de saisie sur les chiffres, tension due à des objectifs différents (le niveau physique cherche à rendre le meilleur service, le niveau financier à limiter les coûts), etc.

Ces interactions sont particulièrement complexes et au-delà des chiffres et de la planification « mathématique », ce sont elles qui sont la cause racine de la difficulté de gestion d'une cantine. Cette difficulté est à prendre au sens relatif : des milliers de cantines fonctionnent chaque jour correctement au niveau national et leurs finances sont saines. Mais cette gestion se fait toujours au prix d'une surveillance humaine et d'une consommation de temps significative. Après des décennies d'expérience, l'informatique n'est pas allée aussi loin que possible dans l'économie de traitements « accessoires ». Il existe une complexité incompressible du métier de la gestion de cantine scolaire, mais force est de constater que de nombreuses interactions qui pourraient être automatisées ne le sont pas encore.

1.1.4.5.3 L'activité de reprogrammation financière

Un cas particulier de mélange des interactions de planification et d'interactions de communication est l'exercice de reprogrammation budgétaire.

La programmation budgétaire permet de prévoir le budget nécessaire pour chaque exercice. Cependant, la réalité ne suit pas nécessairement les prévisions établies. En détectant les écarts par rapport aux prévisions, des décisions peuvent être prises pour reprogrammer le budget en cours d'exercice afin de respecter les objectifs prévus initialement ou de se rendre compte de la nécessité d'une extension de budget. Dans les deux cas, il est important d'identifier le problème au plus tôt afin d'avoir une marge de manœuvre assez importante pour pouvoir le résoudre.

Dans le cas d'une cantine, des dépenses supérieures aux prévisions peuvent être résolues en faisant des économies sur les repas, par exemple. Si cela n'est pas suffisant, l'adjoint à l'éducation peut accorder une extension du budget alloué à la cantine en resserrant celui de la bibliothèque ou d'une autre ligne budgétaire de la commune.

Afin de faire face aux imprévus, un suivi de l'évolution du budget est nécessaire. La planification établie lors de la programmation budgétaire sert de support au suivi. Le déroulement réel est comparé à la planification afin de détecter les écarts par rapport aux prévisions. La programmation budgétaire doit décrire de façon suffisamment claire l'intention derrière les valeurs choisies pour permettre d'expliquer l'écart constaté du budget.

1.1.5 Les difficultés à lever

Le détail des interactions entre les différents niveaux de programmation physico-financière sur l'exemple de la cantine a déjà permis de toucher du doigt les difficultés à lever dans cette thèse concernant la modélisation de la gestion financière. Ces difficultés doivent être organisées afin d'en extraire les questions de recherche.

1.1.5.1 La séparation des rôles

La séparation des rôles entre les différents acteurs de la programmation physico-financière est une des premières difficultés. La spécialisation dans le travail est une bonne chose en soi, mais la séparation des outils qui en découle souvent est quant à elle problématique, car elle a pour conséquence qu'il n'existe pas une vérité centrale partagée par les différents rôles, ce qui occasionne des incompréhensions. Si des questions de confidentialité ou de qualité de la communication s'y ajoutent, la gestion des finances peut devenir extrêmement complexe.

La séparation des rôles et acteurs intervenant dans les différents niveaux est de plus très prégnante : les études menant aux différents postes ne sont pas les mêmes, les acteurs sont souvent localisés dans des bureaux, voire des bâtiments, différents.

Les interactions de communication sont donc logiquement rendues complexes par cette séparation des rôles, ainsi que par la tendance à l'opposition qu'ils rencontrent souvent. Il est en effet courant que les services en charge de la gestion opérationnelle soient perçus comme dépensiers par les services financiers, ces derniers étant en retour vus comme des personnes en dehors des réalités et qui au nom des finances empêchent de réaliser pleinement la mission de service publique. Le fait de proposer un outil commun représentant une vérité financière compréhensible par les différents intervenants, et donc partageable, est donc un point fondamental dans la réalisation d'un outil informatique de la gestion physico-financière. Les rôles doivent bien sûr rester séparés, mais l'outil peut aider à ce que cette séparation reste bénéfique au final. Le simple partage d'une information cohérente, avec des chiffres similaires pour analyser la situation, est un énorme pas en avant par rapport à deux équipes ne se comprenant pas car chacun pense les chiffres de l'autre erronés et ne comprend pas les rapports entre sa propre hiérarchie et celle de son homologue.

Pour caricaturer, un financier a du mal à comprendre l'impact d'un défaut d'approvisionnement sur ses états financiers, et à l'inverse, un cuisinier a du mal à comprendre dans quelle mesure l'augmentation des taux d'intérêts va lui faire réduire ses ambitions pour le repas de Noël.

1.1.5.2 La sémantique

Il est révélateur que le mot « budget » ait été employé à plusieurs reprises dans la description du niveau financier. Il s'agit en partie de la manifestation de l'interdépendance des niveaux, mais aussi d'une illustration du fait que la sémantique n'est pas parfaitement définie entre les acteurs. Ainsi, le niveau financier parle souvent indifféremment d'équilibre du budget et d'équilibre des finances. Ces approximations sémantiques, constatées de manière systématique parmi les personnes en charge du domaine des finances publiques, montrent bien la complexité du domaine, mais surtout la difficulté de modélisation informatique d'un ensemble de concepts qui ne sont pas parfaitement définies intellectuellement par les participants à leur exploitation pratique.

Une des difficultés à lever par la thèse sera donc de permettre une bonne conceptualisation des entités en jeu, de façon à réduire ce biais sémantique. Dans un premier temps, le cas d'usage lui-même sera décortiqué pour faire remonter les noms exacts des différentes opérations afférentes à la gestion d'une cantine. Au-delà de ce premier exemple, la programmation physico-financière sera analysée et modélisée de manière stricte, de façon à faire ressortir un vocabulaire partagé, garant d'échanges de qualité entre les acteurs. Nous verrons plus loin qu'une partie de ces difficultés est levée par une urbanisation du système informatique dans lequel le présent travail vient se poser.

1.1.5.3 Les impacts des différences de temporalité

Un troisième type de séparation qui pose une difficulté à la compréhension partagée d'une programmation physico-financière est celle liée à la gestion du temps. L'axe temporel est en effet la dimension principale (et dans de nombreux cas qui nous concernent, la seule) du modèle théorique représentant informatiquement les métiers de la programmation financière dans les collectivités locales. Cette dimension est donc le principal point de liaison des différents niveaux entre eux.

Or, il se trouve que le maniement du temps est une des grandes difficultés de l'informatique, principalement car le concept est entouré de nombreuses incompréhensions, voire d'hypothèses fausses (par exemple, mais parmi tant d'autres, le fait qu'une semaine commence et finisse sur la même année), mais aussi parce que chaque personne est persuadée d'avoir une compréhension profonde de ces concepts, vu que nous les manions quotidiennement. Comme les différents niveaux de la programmation physico-financière sont sur des approches temporelles généralement différentes, il est nécessaire de bien prendre en compte les subtilités de composition et de transformation des données d'un axe temporel au suivant. Par exemple, les années fiscales sont composées de 12 mois de 30 jours. Un mois d'une année fiscale ne contient pas nécessairement autant de jours qu'un mois du calendrier grégorien.

L'exemple de l'agrégation des valeurs hebdomadaires sur des valeurs mensuelles a déjà été donné : il est alors impossible de simplement cumuler les valeurs, car le niveau financier traite de valeurs mensuelles et le niveau physique gère les factures et fournisseurs à la semaine.

Même en ayant compris ceci, il est nécessaire de poser une étape supplémentaire d'analyse. En effet, il peut paraître logique dans certains cas d'appliquer un prorata. Par exemple, si une personne payée à la semaine doit voir son salaire cumulé au mois, il est logique d'utiliser un prorata au jour. Mais cette même règle ne s'applique par exemple pas à un remboursement d'emprunt : si une telle valeur est portée au niveau du mois, elle ne s'étale en fait pas jour par jour, mais correspond à un paiement ponctuel, à une date donnée. Si cette information doit être transformée dans un axe temporel hebdomadaire, il conviendra de réaliser un calcul beaucoup plus complexe consistant à

retrouver la semaine associée à la date de paiement dans le mois pour lui affecter la totalité du paiement.

L'échelle de temps impose des difficultés conceptuelles fortes sur la modélisation de la programmation physico-financière, et la thèse a dû trouver un moyen de lever durablement cette difficulté, ce qui ne s'est fait qu'au prix d'un retour aux fondements mêmes de la notion de temps.

1.1.6 Les questions ouvertes

Comment formaliser une modélisation de planification financière applicable à des domaines variés ?

Comment évaluer un modèle d'application de la planification financière sur plusieurs échelles de temps ?

Comment permettre une intégration du suivi de la planification financière à un système d'information ?

1.2 La genèse de la thèse

En partant de tous les points de difficultés cités ci-dessus et des actions déjà engagées pour tenter d'y répondre, un premier sujet de thèse a été proposé, qui était fortement orienté sur la performance de calcul des modèles de programmation physico-financière, car il était admis que c'était un prérequis essentiel à des approches à forte valeur ajoutée comme l'optimisation budgétaire.

La thèse a, dans un premier temps, été confiée à une personne qui n'a pas donné satisfaction à MGDIS, et s'est donc arrêtée après quelques mois de travail. A ce moment précis, le sujet était orienté autour de la prévalence objet comme piste de recherche autour de l'amélioration forte des performances, avec comme sujet secondaire la parallélisation.

1.2.1 L'approche par la prévalence objet

1.2.1.1 Les principes

La prévalence objet est un vieux concept récemment remis au goût du jour dans les milieux informatiques (bien que sa connaissance soit restée à un niveau confidentiel). Elle part de la constatation que le matériel actuel n'impose plus les limites qui avaient menées à des systèmes compliqués servant à les contourner. Ces contournements restent utilisés par habitude, bien qu'ils ne soient pas adaptés.

Afin d'expliquer la prévalence objet, il faut revenir de quelques décennies en arrière, à une époque où 32 mégaoctets de mémoire vive sur un serveur représentaient un investissement coûteux. Les données à manipuler étant largement supérieures en taille à cet espace disponible (typiquement quelques centaines de mégaoctets, voire quelques gigaoctets), il a fallu recourir à des astuces de gestion de la donnée, qui était stockée sur des disques durs, proposant plus d'espace mais des temps d'accès beaucoup plus conséquent. Des travaux nombreux et longs de plusieurs années ont abouti à la gestion des bases de données, qui sont au final un moyen optimisé de remonter rapidement de la donnée pertinente en mémoire vive, depuis la mémoire de stockage. La gestion de base de données est un pan entier de l'industrie où des éditeurs se sont épanouis et ont porté à un haut niveau l'art de fournir rapidement la donnée attendue, par des approches variées comme les langages permettant de requêter les données de manière ciblée (SQL principalement), les méthodes d'amélioration du classement (index sophistiqués), les méthodes d'accélération de la fourniture de donnée (cache, découpage fonctionnel), etc.

Or, il se trouve que l'hypothèse de base ayant conduit au développement des logiciels de bases de données n'est désormais plus d'actualité. Certes, les masses de données ont augmenté, mais moins que les capacités maximales des serveurs. Il est aujourd'hui possible de se procurer un serveur avec 192 gigaoctets de mémoire vive, et ce en restant dans des budgets raisonnables même pour une PME. Bien sûr, certains ensembles de données dépassent largement cette taille. Mais dans l'informatique de gestion, une grande majorité est de taille largement inférieure. Pourquoi alors ne pas travailler directement dans la mémoire vive ? Les performances seraient alors bien plus élevées et surtout, le code manipulant les données pourrait se passer de toute la couche gérant l'aller-retour entre les deux types de mémoire. Cette couche est doublement complexe, car d'une part, elle nécessite de garder une cohérence systématique entre les deux mondes et, d'autre part, les deux types de stockage ne sont pas les mêmes imposant des opérations parfois complexes de modification des formats.

Il reste le problème de la gestion de l'arrêt du serveur. L'avantage du disque dur est que dans cette situation, la donnée n'est pas perdue. La mémoire physique étant volatile, ce n'est pas le cas (encore qu'il existe, pour des usages certes très restreints, de la mémoire vive capable de maintenir son état plusieurs minutes, voire heures, sans alimentation électrique). L'apport principal de la prévalence est justement de régler ce point, en stockant de manière pérenne les ordres aboutissant à la modification du modèle en mémoire. De fait, l'état est conservé en mémoire et toutes les opérations de lecture et de calcul sont réalisées dessus, mais les opérations de modification (bien plus rares dans des applications de gestion) sont stockées sur disque dur avant que leur effet soit appliqué sur le modèle en mémoire vive. Ainsi, en cas d'arrêt du serveur – qu'il soit maîtrisé ou inopiné, il est possible de revenir à l'état courant du modèle en mémoire, en appliquant les commandes de modification dans l'ordre original. Ceci nécessite quelques précautions (commandes idempotentes, gestion précise des dates de valeurs), mais rien de complexe. Pour éviter que des modèles lourds prennent trop de temps à remonter en mémoire, les frameworks de prévalence objet proposent des mécanismes de clichés mémoire asynchrones et diverses autres optimisations qui font que l'usage industriel est envisageable.

1.2.1.2 La promesse

L'utilisation des approches de prévalence objet dans le cadre de la thèse permettait d'envisager une forte accélération des traitements, la mémoire vive étant en pratique en moyenne 80 fois plus rapide que les disques durs, au moment d'une étude préalable réalisée par MGDIS.

La mise en œuvre sur les référentiels de SOFI pouvait donc aboutir à des améliorations de temps de calcul telles qu'elles auraient ouvert la voie à la possibilité de lancer des campagnes d'optimisation dans des temps très réduits. La fonctionnalité devenait alors beaucoup plus accessible aux personnes utilisant le logiciel. Cela permettait l'établissement rapide d'une vision commune optimale lors des modifications des budgets, comme décrit plus haut en objectif.

1.2.1.3 Les points d'achoppement

Deux points durs ont été détectés avant même le commencement des développements, à savoir que le logiciel SOFI étant monolithique, il convenait pour tirer parti de la prévalence objet de reprendre tout le code lié à la persistance et à l'accès au modèle. La modularité n'étant pas parfaite, ceci aurait amené à une revue quasi-complète du code source du logiciel.

Surtout, les calculs réalisés sur le modèle avaient été réalisés par rapport à une vision conceptuelle qui était finalement très dépendante du mode de stockage de l'information, et migrer vers une approche en prévalence objet obligeait alors – en tout cas pour tirer le maximum de bénéfice de l'évolution – à remettre en cause non seulement le stockage, mais également la façon dont le moteur

de calcul et les fonctions secondaires de SOFI tiraient de l'information de ce modèle. Le couplage avec le moteur de calcul proprement dit n'étant pas anodin, il aurait fallu, sinon le réécrire, du moins sérieusement recetter le fonctionnement du moteur de calcul par rapport au nouveau modèle de description des entités de SOFI.

1.2.2 L'approche par la parallélisation

1.2.2.1 Les principes

La parallélisation est le mode le plus logique d'accélération des traitements dans l'industrie logicielle, avec le mécanisme de cache. Autant l'approche du cache consiste à éviter de refaire un calcul, autant l'approche par parallélisation accepte de refaire plusieurs fois le même calcul. La parallélisation éclate l'ensemble des opérations demandées en autant de tâches indépendantes et les affecte sur des ressources multiples, de façon à diviser au mieux le temps de traitement total, avant de ré-agrégier les résultats.

1.2.2.2 La promesse

Si SOFI pouvait être modifié de façon à paralléliser les calculs des budgets, il serait alors plus simple de lancer de nombreux calculs dans différentes hypothèses et de retenir uniquement les meilleurs pour aboutir à une solution optimisée. Le coût en ressources devait bien sûr être pris en compte pour que le gain en fonctionnalité fasse plus que contrebalancer le coût en matériel informatique. Mais globalement, la parallélisation apparaissait comme la meilleure piste d'augmentation radicale de la performance. La thèse pouvait également creuser des pistes entièrement nouvelles comme les calculs sur GPU (Graphical Processing Unit : cartes graphiques, très adaptées au calcul rapide en virgule flottante), voire GPGPU (General Purpose GPU : cartes électroniques dédiées au calcul, basées sur des cartes graphiques mais sans lien obligatoire à l'affichage proprement dit).

1.2.2.3 Les points d'achoppement

Encore une fois, le point principal d'achoppement dans cette définition initiale du sujet de thèse était que le logiciel SOFI était relativement monolithique et que les liaisons entre le moteur de calcul proprement dit et les autres fonctionnalités (stockage, affichage, modélisation, etc.) étaient complexes. Le vocabulaire communément accepté dans le métier informatique est de parler de couplage fort. SOFI étant une application fortement couplée (les approches de couplage lâche étant apparues vers la fin de son développement), un gros effort était nécessaire sur la structuration du logiciel, voire sa réécriture, pour envisager de rendre parallèles les traitements de calculs financiers.

1.2.3 La nécessité de penser continu

1.2.3.1 Un tournant dans la définition du sujet de la thèse

C'est lors de l'arrêt de la thèse par la première personne qui en avait été chargée qu'un point essentiel a été atteint, à savoir que l'amélioration des performances seule ne permettrait pas de régler toutes les difficultés ciblées. En effet, des performances accrues rendent possible la fonctionnalité d'optimisation, mais la projection de cas d'usage client réalisée par MGDIS en parallèle montrait qu'un effort était nécessaire pour lever les problématiques d'interaction entre les acteurs. La résolution de ces points était d'ailleurs indispensable à la bonne utilisation des valeurs optimisées. Le risque était que la nouvelle version de SOFI réalise parfaitement les optimisations, mais que le logiciel ne soit toujours pas capable de réunir les acteurs autour d'une vision commune.

L'échange entre les experts métier, les clients et la filière développement a accouché d'un objectif chapeautant celui de la performance, et consistant à modéliser la programmation physico-financière à un niveau d'abstraction supplémentaire à celui existant.

1.2.3.2 *La problématique restante*

En effet, un modèle même parfaitement optimisé doit pouvoir être compris par des personnes qui ne travaillent pas au même niveau temporel. Or, c'est cette discrétisation du temps dans des traditionnels tableaux de valeurs qui a été identifiée comme posant le principal problème d'interaction. Et pendant une longue période, il n'a pas été trouvé de solutions à cette difficulté, tout simplement car l'utilisation de grilles de calcul est tellement ancrée dans le fonctionnement des analystes financiers depuis des dizaines d'années qu'il n'avait pas été imaginé de remettre ce principe en question.

1.2.3.3 *Une voie de résolution*

La remise en cause de ce qui était considéré comme un principe de base de l'analyse financière s'est produite par un changement de paradigme : changer l'approche de modélisation discrète par celle de modélisation continue. La modélisation continue consiste à prendre un dimensionnement dans sa version non-discrète et à poser les concepts et opérations du modèle dans ce cadre plutôt que de travailler sur des valeurs discrètes.

Pour prendre un exemple simplissime, au lieu de faire porter une valeur « fruits & légumes » avec 100 000 € dans la colonne pour l'année 2015, il s'agit désormais de raisonner autour d'une fonction mathématique décrivant la répartition de la densité de dépenses dans le temps, et dont l'intégration sur l'année 2015 retombera sur cette valeur de 100 000 €. Dans ce cas particulier, une fonction continue sur 10 mois avec une densité de 10 000 € par mois, et également continue sur les deux mois d'été avec une densité nulle peut être choisie. Bien sûr, des modélisations plus précises pourraient être envisagées, mais cela dépend du niveau de précision qu'on souhaite atteindre pour la prospective.

Evidemment, dans d'autres cas de dépenses, la fonction de répartition ne serait pas du tout la même. Par exemple, la représentation mathématique la plus logique pour les salaires est une fonction composée de pics (représentés par des fonctions de Dirac) portant la densité complète de salaire sur des instants précis de l'axe temporel qui sont typiquement le dernier jour ouvré du mois, à minuit.

1.2.3.4 *L'apport sur la modélisation conceptuelle*

Cette approche, bien que conceptuellement complexe, présente l'avantage de lever les problématiques de représentation incompatibles dans les différentes périodes temporelles. En effet, la valeur sur une périodicité donnée étant systématiquement retrouvée par l'intégration d'une seule fonction mathématique représentant la réalité, les nombres obtenus sont automatiquement cohérents.

Evidemment, du point de vue de l'utilisateur final, il convient de masquer cette complexité mathématique et de ne présenter que des états discrets. Mais en fond, c'est bien le moteur de calcul en temps continu qui constitue le nouveau modèle conceptuel de la programmation physico-financière.

Cette approche avait également pour avantage de forcer les modélisateurs à se poser les bonnes questions de la répartition effective des mouvements financiers. Pour reprendre les exemples précédemment cités, dans une approche discrète, rien ne force les utilisateurs du modèle à se poser la question de la répartition continue ou ponctuelle des dépenses. Or, ceci pose un réel problème métier lorsqu'il s'agit de gérer de la trésorerie au jour le jour. Un financier, à son niveau métier, peut se permettre de considérer les salaires et les dépenses pour les fruits et légumes au même niveau, à savoir une masse mensuelle, voire annuelle. Mais il n'empêche que pour le gestionnaire de la cantine, qui paie au jour le jour ses fournisseurs mais en fin de mois ses salaires, la différence est

fondamentale car c'est ce qui pilote sa trésorerie, sa relation à la banque et au final son budget propre.

Nous revenons donc aux difficultés d'interaction que nous décrivions plus haut : chacun s'occupant de son point de vue, si le modèle n'est pas commun, il est très difficile de réconcilier les valeurs et de faire comprendre à chacun les besoins de l'autre. Le fait de mettre en place un modèle mathématique commun, au-delà de l'apport technique, a également une portée très forte du point de vue conceptuel, car il modélise mieux une réalité partagée que des simples chiffres dans des cases. Surtout, chacun peut le regarder de son propre point de vue sans faire perdre de la richesse à un autre. Les états fonctionnent alors comme des plans de coupe ou des vues particulières d'un modèle central, plutôt que comme des représentations de portions de modèles incohérents et qu'il faut rassembler après coup, au prix de circonvolutions intellectuelles et de discussions compliquées entre acteurs.

1.2.3.5 La gestion de la précision

Une difficulté potentielle d'adoption de la modélisation continue est que la transcription d'un modèle mathématique continu en une valeur discrète (nécessaire pour la production d'états financiers et le paiement de valeurs effectives) s'accompagne nécessairement d'une perte de précision.

Dans la pratique, ceci n'est pas réellement une problématique, car cette perte de précision est consubstantielle aux métiers de la finance, les montants étant toujours arrondis aux montants légaux tolérés par la monnaie. Dans le cas de l'euro, l'arrondi est systématiquement appliqué deux chiffres après la virgule. Il y a donc déjà perte de précision connue et acceptée par le domaine.

De plus, dans les collectivités locales et en particulier dans l'usage de SOFI, l'approche prospective fait que les financiers ont plutôt tendance à gérer des masses en milliers d'euros, et l'arrondi est alors placé encore un niveau au-dessus. La perte de précision mathématique due à l'intégration d'une fonction pour donner un résultat discret est alors non significative. Au moment où le modèle est utilisé pour de la trésorerie, il convient bien sûr de maîtriser la précision du résultat, mais ceci ne se réalise qu'au prix de calculs supplémentaires tout à fait acceptables.

1.2.4 L'apport de l'urbanisation des SI

1.2.4.1 Le besoin

Il a été vu précédemment que les approches sur la prévalence objet et sur la parallélisation étaient toutes deux bloquées par le fait que SOFI était un logiciel monolithique. Le couplage des fonctionnalités rendait difficile de mettre en place une gestion différente de la persistance ou de migrer le moteur de calcul pour une version différente. Ces objectifs ont donc rapidement été retirés du sujet de thèse pour se consacrer à une approche plus profonde de modularisation de l'application pour supporter ce type d'évolution dans le futur, et qui avait en outre l'avantage d'être parfaitement alignée sur une évolution stratégique majeure de MGDIS pour ses logiciels, à savoir leur urbanisation.

1.2.4.2 La conceptualisation

L'urbanisation des Systèmes d'Information (SI) logiciels consiste à découper les applicatifs en ensemble de services fournis et qui sont implémentés par des composants complètement autonomes. Sans rentrer dans les concepts, il s'agit de séparer la complexité technique de la complexité du métier, en mettant en œuvre des interfaces normalisées.

Pour donner un exemple sur SOFI, au lieu qu'un seul logiciel stocke une modélisation financière et calcule ses résultats, il s'agit de découper les services entre la modélisation, le calcul et l'affichage. La relation entre la modélisation et le calcul est alors établie selon un contrat régissant la forme précise

des échanges entre les deux services, de façon que chaque fonctionnalité puisse évoluer sans impact sur l'autre. La recombinaison du scénario supporté par le logiciel dans son ensemble est alors réalisée par une orchestration des différents services sur lesquels le propriétaire du SI a la main, et peut donc modifier librement, ce qui n'est pas le cas dans les logiciels monolithiques.

1.2.4.3 *Autonomie*

L'objectif derrière l'urbanisation de SOFI est, comme cela a été présenté, de rendre autonome les différentes fonctionnalités du logiciel, afin de pouvoir le faire évoluer plus facilement. Si la persistance est ressortie, il devient plus simple de remplacer la base de données la supportant par une gestion de la prévalence objet des entités manipulées par SOFI. De même, si le moteur de calcul est modélisé informatiquement et contractuellement comme un service à part, il devient plus simple de le faire évoluer et le remplacer, voire même mettre en parallèle des versions différentes sans que l'appelant, en l'occurrence le mécanisme d'optimisation de SOFI, ne soit au courant que les services sont implémentés dans des technologies différentes.

1.2.4.4 *La sémantique*

L'expérience de MGDIS a également montré que l'activité de découplage associée à l'urbanisation d'une application logicielle force à atteindre un niveau de conceptualisation supérieur du métier, qui à son tour nécessite l'établissement d'une sémantique particulièrement stricte. L'urbanisation en services répond donc non seulement au besoin de découplage de SOFI du point de vue technique, mais également au besoin détecté de mise en œuvre d'un vocabulaire commun précis de façon que les acteurs s'alignent plus facilement sur une vision commune effectivement partagée.

1.2.4.5 *Le découpage et la simplification*

Enfin, un dernier effet bénéfique de la simplification est que, les services et responsabilités étant séparés, la taille des ensembles de données baisse naturellement par découpage, et la limite de la prévalence objet sur la taille maximale permise par la mémoire d'un serveur devient alors de moins en moins prégnante.

Au passage, il ne s'agit que d'un seul parmi les multiples avantages à découper une architecture logicielle et la simplifier. La simplification permet également de nouveaux usages et une maintenance facilitée, une maîtrise de la performance en particulier en montée en charge grâce à la meilleure granularité, une possibilité supplémentaire de partager des services, etc.

La réorientation du sujet de la thèse autour de la modélisation continue dans le temps et de l'urbanisation des services logiciels de modélisation physico-financière est bien une décision fondamentale, et riche de nombreuses promesses. En conséquence, elle oblige à un effort de réflexion et de conceptualisation nécessaire qui ne faisait que rendre plus prégnant le besoin d'une thèse sur le sujet.

1.3 *Le sujet de la thèse*

1.3.1 *Les objectifs métier*

Les objectifs de la thèse ont été partiellement exposés ci-dessus en expliquant les problématiques et les pistes ouvertes qui devaient être défrichées par la thèse. Une thèse CIFRE est réalisée pour les besoins d'un industriel, et il convient donc de partir des besoins des clients, qui sont au cœur de la démarche d'innovation. Les objectifs du métier sont définis de façon plus formelle ci-dessous. Leur définition est portée par trois questions de recherche :

- Comment formaliser une modélisation de planification financière applicable à des domaines variés ?

- Comment évaluer un modèle d'application de la planification financière sur plusieurs échelles de temps ?
- Comment permettre une intégration du suivi de la planification financière au système d'information ?

1.3.2 Les objectifs intellectuels

Il était clair dès le début de la thèse qu'une des grandes difficultés intellectuelles serait de produire une modélisation générique du domaine de la programmation physico-financière. Comme dit précédemment, la notion de temps discret est enracinée de manière tellement profonde dans le domaine que tous les exemples, documents, modèles font systématiquement appel à une vision discrète. La littérature, conformément à la façon de travailler des experts, propose systématiquement des tableaux de valeurs (l'axe temporel étant souvent matérialisé par les colonnes). Que le problème soit pris du point de vue de la comptabilité, de la prospective financière, de l'analyse rétrospective, de l'optimisation budgétaire ou de toute autre façon, les valeurs monétaires sont systématiquement représentées selon une vision discrète. La conception d'un modèle qui s'abstrait de cette présentation du temps est clairement un point intellectuel majeur dans le sujet de thèse.

Cette activité de conception est rendue encore plus complexe par le fait que le modèle intellectuel doit s'inscrire dans le second objectif d'urbanisation du logiciel final. En plus de concevoir un modèle générique à tous les usages de programmation physico-financière et qui ne se base pas sur une vision traditionnelle de l'axe temporel, il faut réussir à ce que les concepts trouvés soient parfaitement autonomes et puissent être traités par des systèmes informatiques découplés. Ceci ajoute clairement de la complexité, mais en même temps donne une méthode de simplification par découpage qui pourra être d'une grande utilité pour guider justement la conception.

Enfin, concevoir des systèmes intellectuellement complexes est une activité difficile, mais la vulgarisation qui doit nécessairement s'ensuivre l'est encore plus. Or, modéliser un système de programmation physico-financière n'est que la première étape. La seconde étape est la réalisation technique d'un logiciel correspondant, et elle est autant nécessaire à l'appropriation par les clients finaux des concepts, donc à la réussite de l'innovation portée par la thèse.

1.3.3 Les objectifs techniques

La mise en œuvre technique des concepts portés par la thèse fait intégralement partie de celle-ci, car une thèse CIFRE est portée par un besoin industriel et doit y répondre. La thèse doit donc répondre à des objectifs techniques, qui sont principalement les mêmes que ceux que MGDIS s'applique sur ses développements industriels.

Il s'agit principalement de contraintes de qualité du code fourni et du livrable du point de vue du client. Pour cette seconde partie, le sujet défini dans cette thèse comportait des défis en termes de maintien d'une bonne performance et d'une bonne ergonomie de fonctionnement.

Les problématiques de performance ont été abondamment détaillées car un calcul rapide du modèle financier est absolument indispensable à la mise en œuvre de fonctionnalités d'optimisation. En effet, les approches par inversion du modèle ayant pour l'instant été démontrées comme inopérantes. La modélisation continue pourrait remettre au goût du jour cette possibilité d'optimisation mathématique, mais ce sujet complexe a été volontairement mis de côté. L'objectif de cette thèse est de montrer la mise en œuvre d'un moteur de modélisation continu dans le cadre de la programmation physico-financière. Si cet objectif est atteint, la faisabilité d'une optimisation par des méthodes mathématiques pourra être envisagée dans le cadre d'autres travaux de recherche

ultérieurs. Dans un premier temps, les approches évolutives (algorithmes génétiques, principalement) ont été validées comme la voie de prédilection, pour les raisons précédemment évoquées.

Les problématiques d'ergonomie d'usage, bien que moins prédominantes que les problématiques de performance dans le cas présent, restent un objectif technique important dans la thèse. Le logiciel résultant doit faire passer de manière simple à l'utilisateur la puissance de la modélisation continue du temps. En première contrainte, l'utilisateur habitué depuis de longues années d'analyse financière à la modélisation discrète du temps ne doit pas se sentir dérouté par le nouveau modèle. Ensuite, les capacités étendues du nouveau modèle doivent lui être exposées de manière qu'il les utilise facilement.

1.3.4 Le sujet scientifique

1.3.4.1 *La modélisation en temps continu*

Du point de vue scientifique, la définition d'un modèle générique de programmation physico-financière nécessite d'appréhender aussi bien des notions de modélisation mathématiques que de programmation physico-financière. Ceci afin de cerner les ambiguïtés qui pourraient poser problème lors de la transformation en modèle mathématique ainsi que celles qui pourraient empêcher d'exprimer la logique de la programmation physico-financière de façon naturelle.

1.3.4.2 *Le découplage de responsabilités*

Le découplage de responsabilités fait l'objet de sujets de réflexion. Architectures orientées services, approches REST, Entreprise Service Bus et urbanisation des SI, tous ces sujets sont d'actualité dans le domaine de l'informatique industrielle. Il conviendra donc dans le cadre de la thèse d'avoir une approche formelle du sujet.

1.3.4.3 *L'équilibre performance / précision*

Le sujet de la gestion de la précision a été brièvement abordé, en expliquant que la précision variable due à la discrétisation des résultats continus ne posait pas un problème, dans le sens où non seulement une précision absolue n'était pas requise par le métier, mais en plus par le fait que, si c'était le cas, le modèle mathématique serait un bien meilleur candidat que le modèle discret qui réalise une approximation pour toutes les valeurs utilisées.

Toutefois, il reste important de connaître le niveau de précision atteint effectivement, ne serait-ce que pour pouvoir jouer dessus en fonction des besoins, et ce afin de gagner encore en performance. Nul besoin de réaliser un long calcul d'intégration pour atteindre une précision trois chiffres après la virgule lorsque la personne demandant un état financier est un gestionnaire de haut niveau qui travaille sur des masses monétaires exprimées en kiloeuros.

Le lien entre la performance et la précision pourra également être un sujet d'étude de la thèse, même s'il ne s'agit pas d'un objectif défini dans le sujet.

1.3.5 Le sujet industriel

1.3.5.1 *La description de l'existant*

Cette thèse s'inscrit dans un contexte industriel où MGDIS dispose déjà d'une version stable d'un logiciel éprouvé par les besoins de la programmation physico-financière, mais qui fonctionne en modélisation discrète. Ce logiciel nommé SOFI est en production chez plusieurs centaines de clients depuis presque une dizaine d'années. Chaque année environ, une version majeure de SOFI apporte de nouvelles fonctionnalités, et des modifications de plus faible ampleur sont réalisées de manière ad hoc.

Pour des raisons de commodité, la version existante sera nommée « SOFI discret » et la version mise en œuvre dans le cadre de cette thèse et embarquant un moteur de modélisation continue du temps sera nommée « SOFI continu ».

1.3.5.2 *Le contexte commercial*

Le logiciel SOFI a peu de concurrence en nombre, mais le concurrent principal est redoutable, car il s'agit d'un des logiciels les plus diffusés dans le monde, à savoir Excel. Nombreux sont les consultants financiers et les gestionnaires qui apprécient la liberté totale de modélisation que leur procure Excel. Le but de SOFI est de les aider à aller plus vite et de façon sécurisée, en gérant plus facilement le temps, les liens entre les modèles, l'instanciation de projets, etc. Bref, il cherche à se comporter comme un Excel spécialisé dans la programmation physico-financière en offrant des fonctions dédiées.

L'optimisation financière pourrait être une des fonctionnalités permettant à SOFI de se mettre en avant par rapport à Excel. En effet, des essais réalisés par MGDIS ont montré que les modèles financiers sont trop complexes pour que le Solveur, outil d'optimisation d'Excel, puisse trouver des solutions correctes dans un temps acceptable. Même des composants additionnels d'Excel spécialisés dans l'optimisation n'ont pas donné satisfaction dans les prototypes réalisés en amont de la thèse.

1.3.5.3 *Attendu commercial*

L'attendu commercial par rapport à la thèse CIFRE est de produire un modèle utilisable rapidement chez les clients. Bien sûr, la thèse ne peut pas supporter la programmation d'une nouvelle version complète de SOFI, car de nombreuses tâches n'ont rien d'innovant et doivent rester dans le giron de MGDIS. Par contre, il y a clairement une attente de collaboration avec les équipes de développement de MGDIS dans le but de réaliser une version livrable de SOFI, compatible en termes de modèles financiers avec la précédente ligne, et permettant dans une ergonomie raisonnable de tirer parti de la puissance d'un nouveau moteur de calcul basé sur la modélisation continue du temps.

Plusieurs expériences de thèses ou de travaux de doctorants ont montré par le passé l'importance capitale de faire participer le plus possible des clients finaux de la solution en amont. Dans le cadre de SOFI, il se trouve que l'exploitation des modèles est réalisée par les clients finaux de MGDIS, mais que la création des modèles eux-mêmes est presque toujours réalisée par les experts financiers de MGDIS. Ceux-ci seront donc à considérer pendant toute la thèse comme les clients de la solution à concevoir. Par conséquent, des revues régulières seront réalisées avec ces personnes.

1.4 Le contexte

1.4.1 Société MGDIS

Cette thèse est réalisée en cotutelle entre l'IRISA et la société MGDIS avec un contrat CIFRE (Conventions Industrielles de Formation par la Recherche).

L'IRISA (Institut de recherche en informatique et systèmes aléatoires) est une UMR (unité mixte de recherche). Le laboratoire est formé de l'association de 7 établissements tutelles (CNRS, ENS Rennes, Inria, INSA de Rennes, Institut Mines-Télécom, Université de Bretagne Sud (UBS), Université de Rennes 1) et un établissement partenaire (Supélec).

MGDIS est un éditeur de logiciels d'aide à la décision et de pilotage. Ses clients sont principalement des collectivités locales, des hôpitaux, des EHPAD, des ports de plaisance, des organismes d'Etat et des établissements publics et parapublics. MGDIS propose des logiciels pour analyser, simuler et piloter l'information financière ; gérer et piloter des dossiers administratifs pour instruire, gérer et maîtriser les aides, les subventions à percevoir et les opérations ; présenter des données publiques ;

suivre et évaluer des politiques et des programmes publics ; gérer et administrer des ports de plaisance.

1.4.2 Les travaux associés

Le calcul continu est la solution envisagée par MGDIS pour réaliser la modélisation financière et son calcul dans ses produits. Pour cela, un partenariat a été mis en place par MGDIS afin de développer une bibliothèque de calcul continu, nommée « LEMF (Library Embedded Finance) », dans le cadre d'une autre thèse.

Bien que le prototype réalisé dans cette thèse soit, dans un premier temps, le seul produit à utiliser la bibliothèque LEMF, une attention particulière sera portée à ce que les problèmes techniques ne débordent pas d'un projet sur l'autre. Pour cela, le couplage du prototype à la bibliothèque doit être réduit au minimum et évidemment être inexistant dans le sens opposé étant donné qu'une bibliothèque a vocation à être utilisée par de nombreux logiciels.

Plusieurs profils d'utilisateurs doivent être ciblés par la bibliothèque LEMF. L'interface de la bibliothèque LEMF doit proposer deux points d'entrée :

- l'un doit permettre une utilisation par des non mathématiciens grâce à une interface simple masquant la complexité des modèles mathématiques utilisés ;
- l'autre doit permettre aux chercheurs d'un laboratoire de mathématiques d'utiliser directement les concepts avancés de modélisation continue et de calculs numériques.

L'interface simplifiée est celle qui est utilisée par le prototype. Elle a fait l'objet de nombreuses analyses lors desquelles plusieurs pistes ont été envisagées. Ces pistes sont détaillées par la suite dans le chapitre 5.1.1.

1.5 L'organisation du manuscrit

Ce manuscrit est organisé de la façon suivante :

- le chapitre 2 présente un état de l'art des techniques d'aides à la décision dans le cadre de la planification financière,
- le chapitre 3 décrit un cas de planification physico-financière qui sera utilisé pour illustrer les propos des chapitres suivants,
- le chapitre 4 expose l'approche de solution proposée qui se base sur la modélisation en temps continu en la comparant point par point aux approches de modélisation en temps discret,
- le chapitre 5 montre une application de l'approche proposée au cas d'étude en mettant l'accent sur son expérimentation et sa validation,
- le chapitre 6 conclut ce manuscrit par le bilan des contributions de cette thèse au delà de l'état de l'art ainsi que par les perspectives ouvertes par ces travaux.

2 L'état de l'art

2.1 Les approches historiques

Les difficultés décrites ci-dessus ont déjà dans le passé fait l'objet d'approches de résolution de la part de la société MGDIS. Le logiciel SOFI dans sa première version existe depuis plus de dix ans, et des modules expérimentaux lui ont successivement été greffés, jusqu'à maintenant avec un succès limité, d'où l'existence de cette thèse.

2.1.1 Les approches par cubes de données

Les approches de Business Intelligence ont été expérimentées au début des années 2000. L'idée était que les données issues de la gestion financière soient exploitées dans des cubes pré-agrégés, et que cet accès rapide à des données composites permette à différents profils de se retrouver autour d'une donnée commune.

Le concept de « Single Version Of the Truth » (ou unique version des faits) était une solution partielle aux écarts des valeurs sur les différents niveaux qui amènent à des décisions différentes entre les acteurs. Toutefois, l'extraction de ces données vers un autre outil faisait que seule la consultation d'un énième rapport permettait de partager une vision commune. Au lieu d'avoir un outil commun de travail, les différents acteurs continuaient donc à utiliser des outils différents, et ne se retrouvaient qu'autour d'un outil tiers dont le maintien venait s'ajouter à leurs tâches opérationnelles. Logiquement, ce surcroît de manipulation n'a pas favorisé la pérennité de l'approche.

De plus, une des limitations fortes de l'approche Business Intelligence était que toutes les données devaient être centralisées pour que l'ensemble fonctionne, ce qui posait un gros problème technique de goulet d'étranglement (on parle de « Single Point Of Failure » ou point individuel de défaillance, en écho négatif au concept « Single Version Of the Truth » évoqué ci-dessus). Cette centralisation simple à réaliser avec des volumes de données limités voyait son coût augmenter de manière exponentielle avec les volumes à traiter.

La solution d'agrégation de données en mode Business Intelligence a donc eu du mal à trouver son public. Plus récemment, le théorème CAP (cohérence « Consistency », disponibilité « Availability », tolérance au partitionnement « Partition tolerance ») montrant qu'un partitionnement de la donnée est possible à condition d'accepter une consistance a posteriori a rendu possible la distribution des données sur des bases distantes, donné naissance au mouvement NoSQL et propose une alternative à l'approche par cubes de données pré-agrégés.

2.1.2 L'optimisation par contraintes

La première approche ayant échoué, la société MGDIS a changé son fusil d'épaule et est partie sur une autre approche, à savoir qu'une façon finalement idéale de trouver un terrain d'entente entre les différents intervenants était d'établir, en fonction de contraintes, un budget optimal qui les regrouperait sur une seule vision. L'idée était qu'en calculant une situation optimale à partir des contraintes physiques mais aussi financières, la solution mathématique sous forme de budget équilibré pourrait remplacer les valeurs de chacune des entités. Pour cela, il fallait que la détermination d'une solution optimale puisse être lancée lors d'une modification substantielle des valeurs du modèle, et que le logiciel rafraîchisse ensuite en retour les valeurs et la composition dans les modèles de travail des participants à la programmation financière.

Dans notre cas d'étude, la présence d'un aléa sur la construction d'une cantine était enregistrée par l'intendant, qui lançait un calcul d'optimisation avec les nouvelles valeurs. Celui-ci montrait la nouvelle valeur optimale pour la réservation de dépenses dans le budget de la cantine, et donc les

ressources financières nécessaires à provisionner. Le gestionnaire financier prenait acte de ces valeurs, les ajustait en fonction du vote éventuellement nécessaire, puis relançait l'optimisation qui avait pour conséquence d'augmenter le budget alloué aux dépenses de denrées, ce qui levait l'alerte opérationnelle.

Les premiers prototypes basés sur la solution de calcul sous contraintes de la société ILOG, nommée JConfigurator, étaient très prometteurs et des premiers modèles financiers ont effectivement été optimisés. La difficulté est venue du fait que cette solution ne passait pas du tout à l'échelle, et que les temps de traitement, excellents pour des modèles simples, devenaient tout simplement inacceptables lorsque les modèles devenaient plus complexes. L'objectif étant de traiter à terme des modèles extrêmement complexes, il était évident qu'une simple amélioration des ressources informatiques ne suffirait pas à compenser cette dégradation de performance et que, même avec les meilleurs serveurs, la solution serait inutilisable pour des vrais budgets d'hôpitaux, qui comptent parfois des centaines de projets avec les emprunts financiers afférents.

L'échec de cette seconde approche a été acté par le non-renouvellement de la licence de l'outil JConfigurator, au milieu des années 2000. L'abandon du produit par l'éditeur lors de son rachat par IBM tend par ailleurs à confirmer que ce type de composant logiciel n'était peut-être pas mûr.

2.1.3 L'optimisation par algorithmes génétiques

Le calcul sous contraintes porté par la solution basé sur ILOG JConfigurator était une solution dont la lenteur sous charge provenait principalement du fait qu'elle recherchait systématiquement l'optimum mathématique absolu. Or, il se trouve que les contraintes financières sur le modèle sont souvent relativement floues. Par exemple, lorsqu'un financier exige une valeur pour un indicateur appelé capacité de désendettement (CDD) inférieure à 15 ans, et qu'un consultant propose une solution lui permettant de réaliser toutes les opérations promises par les élus avec une CDD de 15 ans et un mois, cette solution sera bien sûr acceptée, en adaptant à la marge quelques dépenses si nécessaire pour retomber dans les ratios prudentiels.

Et c'est là que le coût de la précision mathématique des outils de calculs sous contraintes devient rédhibitoire : il est acceptable d'attendre longtemps une solution mathématiquement parfaite lorsqu'on a besoin qu'elle le soit. Mais dans le monde financier, les équilibres se font sur des grandes masses, et les gestionnaires apprécieront plus de se voir proposer plusieurs solutions imparfaites mais prometteuses qu'une solution qui colle parfaitement à des hypothèses qui vont de toute façon évoluer.

Après une période de veille sur le sujet de l'optimisation mathématique, MGDIS a lancé un stage sur l'optimisation par algorithmes génétiques, puis une thèse sur le sujet. L'avantage de ces algorithmes évolutifs est qu'ils ne cherchent pas à fournir une solution absolument parfaite au prix d'une utilisation énorme de ressources, mais évoluent progressivement vers des approches de plus en plus intéressantes. Il est donc possible de rapidement trouver des solutions correctes (et en tout cas largement assez précises pour des financiers raisonnant en dizaines, voire centaines, de milliers d'euros) et d'arrêter le calcul lorsque le degré de précision voulu est atteint. Surtout, ces algorithmes se comportent de manière que les temps de traitement suivent de manière exponentielle la précision voulue. Les solutions de précision faible sortent donc du moteur d'optimisation de manière extrêmement rapide.

La thèse correspondant à ce travail n'a pas abouti pour des raisons personnelles, mais le travail associé a donné lieu à une publication, et pose des bases stables pour un processus d'optimisation. Par contre, la modélisation utilisée pour la gestion financière était entièrement basée sur une

représentation standard du temps qui pose les problèmes que nous avons détaillés plus haut. Il était donc nécessaire de pousser plus loin la démarche, le problème de l'optimisation performante ayant été résolu.

2.2 La revue systématique

Afin d'établir l'état de l'art sur les techniques d'aide à la décision dans le domaine de la programmation physico-financière, nous avons utilisé une méthode de revue nommée cartographie systématique. A l'opposé de la méthode traditionnelle souvent retrouvée dans les articles, dite revue narrative (qui consiste en un rappel des connaissances portant sur un sujet précis, recueillies à partir de la littérature pertinente sans processus méthodologique systématique, explicite, d'obtention et d'analyse qualitative des articles inclus dans la revue), la revue systématique se distingue de la revue narrative par le fait qu'elle utilise une méthode rigoureuse afin de recueillir et d'analyser les articles portant sur un domaine particulier.

La revue systématique consiste à : (i) rassembler, évaluer et synthétiser toutes les études pertinentes qui abordent un problème donné, en l'occurrence la programmation physico-financière ; (ii) limiter l'introduction d'erreurs aléatoires ou biais.

L'élaboration d'une revue systématique est basée sur un protocole détaillé préalable et est composée des cinq étapes présentées : (1) Formulation des questions de recherche, comprenant la détermination des objectifs et des critères d'inclusion et d'exclusion des études ; (2) Recherche et sélection des études pertinentes ; (3) Evaluation de la qualité des études retenues ; (4) Extraction des données pertinentes et analyse de ces données ; (5) Interprétation des résultats vis-à-vis des questions de recherche.

Notre revue systématique propose ainsi une cartographie des articles de la littérature qui s'intéressent à l'aide à la décision dans le domaine de la planification financière, y compris la programmation physico-financière.

2.2.1 Formulation des questions de recherche

Afin de présenter l'état de l'art pour la problématique traitée dans cet article, notre revue systématique, afin d'élaborer une cartographie systématique, répondra aux questions de recherche suivantes :

- Quelles sont les approches utilisées concernant la modélisation de la planification financière pour les systèmes d'aide à la décision ?
- Comment la modélisation en temps continu a-t-elle été appliquée aux systèmes d'aide à la décision pour la planification financière ?
- Quels sont les domaines d'application de la modélisation de la planification financière pour les systèmes d'aide à la décision ?

Afin de répondre à ces questions, les bases documentaires utilisées pour mener cette cartographie systématique sont les bases anglophones majeures en informatique décisionnelle :

Source	URL
ScienceDirect	http://www.sciencedirect.com
IEEE Xplore	http://ieeexplore.ieee.org
SpringerLink	http://link.springer.com
ACM Digital Library	http://dl.acm.org

Figure 2 : Les sources utilisées pour la recherche de publications

Sur ces bases, les requêtes ont été définies à l'aide de mots clés :

- l'aide à la décision : « decision », « Business Intelligence », « BI » ;
- la planification financière : « budgetary planning », « business planning », « budget planning », « budgetary control », « business control », « budget control », « financial strategy ».

```
(  
  "budgetary planning"  
  OR "business planning"  
  OR "budget planning"  
  OR "budgetary control"  
  OR "business control"  
  OR "budget control"  
  OR "financial strategy"  
)  
AND  
(  
  decision  
  OR "Business Intelligence"  
  OR "BI"  
)
```

Figure 3 : La requête pour la recherche de publications

Suite aux résultats obtenus par l'exécution de ces requêtes, les critères d'inclusion et d'exclusion suivants ont été appliqués.

Les critères d'inclusion :

- La publication présente une approche pour réaliser un système d'aide à la décision appliqué à la planification financière d'un domaine.

Les critères d'exclusion :

- La publication a pour sujet la planification financière, mais ne présente pas d'approche pour réaliser un système d'aide à la décision.
- La publication présente une approche pour réaliser un système d'aide à la décision, mais n'a pas pour sujet la planification financière.
- La publication ne présente ni d'approche pour réaliser un système d'aide à la décision ni n'a comme domaine d'application la planification financière.

2.2.2 La réalisation

2.2.2.1 Les sources

2.2.2.1.1 ScienceDirect

La recherche est réalisée en incluant uniquement ces deux domaines : Computer Science, Decision Science.

La recherche a abouti à 57 Résultats.

```

tak
(
  (
    "budgetary planning"
    OR "business planning"
    OR "budget planning"
    OR "budgetary control"
    OR "business control"
    OR "budget control"
    OR "financial strategy"
  )
  AND
  (
    decision
    OR "Business Intelligence"
    OR "BI"
  )
)

```

Figure 4 : La requête pour ScienceDirect

2.2.2.1.2 IEEE Xplore

La recherche a abouti à 37 Résultats.

```

(
  "Abstract": "budgetary planning"
  OR "Abstract": "business planning"
  OR "Abstract": "budget planning"
  OR "Abstract": "budgetary control"
  OR "Abstract": "business control"
  OR "Abstract": "budget control"
  OR "Abstract": "financial strategy"
)
AND
(
  "Abstract": decision
  OR "Abstract": "Business Intelligence"
  OR "Abstract": "BI"
)

```

Figure 5 : La requête pour IEEE

2.2.2.1.3 SpringerLink

La recherche est réalisée avec comme filtre : « Computer Science », « Business Information Systems » et de langue anglaise.

La recherche a abouti à 9 Résultats.

```

(
  ab:"budgetary planning"
  OR ab:"business planning"
  OR ab:"budget planning"
  OR ab:"budgetary control"
  OR ab:"business control"
  OR ab:"budget control"
  OR ab:"financial strategy"
)
AND
(
  ab:decision
  OR ab:"Business Intelligence"
  OR ab:"BI"
)

```

Figure 6 : La requête pour SpringerLink

2.2.2.1.4 ACM

La recherche a abouti à 11 Résultats.

```
Abstract:
(
  (
    "budgetary planning"
    OR "business planning"
    OR "budget planning"
    OR "budgetary control"
    OR "business control"
    OR "budget control"
    OR "financial strategy"
  )
  AND
  (
    decision
    OR "Business Intelligence"
    OR "BI"
  )
)
```

Figure 7 : La requête pour ACM

2.2.3 Recherche et sélection des études pertinentes

La requête définie a été exécutée sur chacune des bases documentaires donnant les résultats suivants. Ces recherches ont été appliquées aux domaines de « Computer Science », « Decision Sciences » et « Business Information Systems ».

Source	Nombre de résultats
ScienceDirect (SD)	57
IEEE Xplore (I3E)	37
SpringerLink (SL)	9
ACM Digital Library (ACM)	11

Figure 8 : Le nombre de résultats obtenus à partir des sources

Suite à l'application des critères d'inclusion et d'exclusion, les études primaires retenues sont, par ordre chronologique décroissant, les suivants.

Type	Année	Auteurs	Source
JA	2014	Cha et al.	ACM
BS	2013	Baker et al.	SD
JA	2012	Sato et al.	SD
JA	2010	Lourdes Borrajo et al.	SD
CP	2009	Eisenreich et al.	ACM
JA	2008	Wang et al.	SD
CP	2007	Suggs et al.	I3E
JA	2007	Wong et al.	SD
JA	2006	Xie et al.	SD
JA	2006	Bermúdez et al.	SD
JA	2005	Wen et al.	SD
CP	2004	Wang et al.	I3E
CP	2003	Mayo et al.	I3E
JA	2001	Rees et al.	SD
JA	1998	Choi et al.	SD
JA	1997	Kim et al.	SD
JA	1997	Cho et al.	SD
JA	1994	Sethi et al.	SD
JA	1993	Baugh et al.	SD
JA	1988	Hruschka et al.	SD
JA	1983	Broeckx et al.	SD
JA	1980	Lin et al.	SD

Figure 9 : Liste des études primaires

2.2.4 Evaluation de la qualité des études retenues

Chaque étude primaire retenue présente l'une des deux qualités suivantes : (i) études basées sur des données probantes ; (ii) études basées sur des résultats novateurs.

2.2.5 Extraction des données pertinentes et analyse de ces données

Les études primaires ont été classées par domaines d'application. La quasi-totalité des articles s'intéresse au cas de la programmation financière en entreprise.

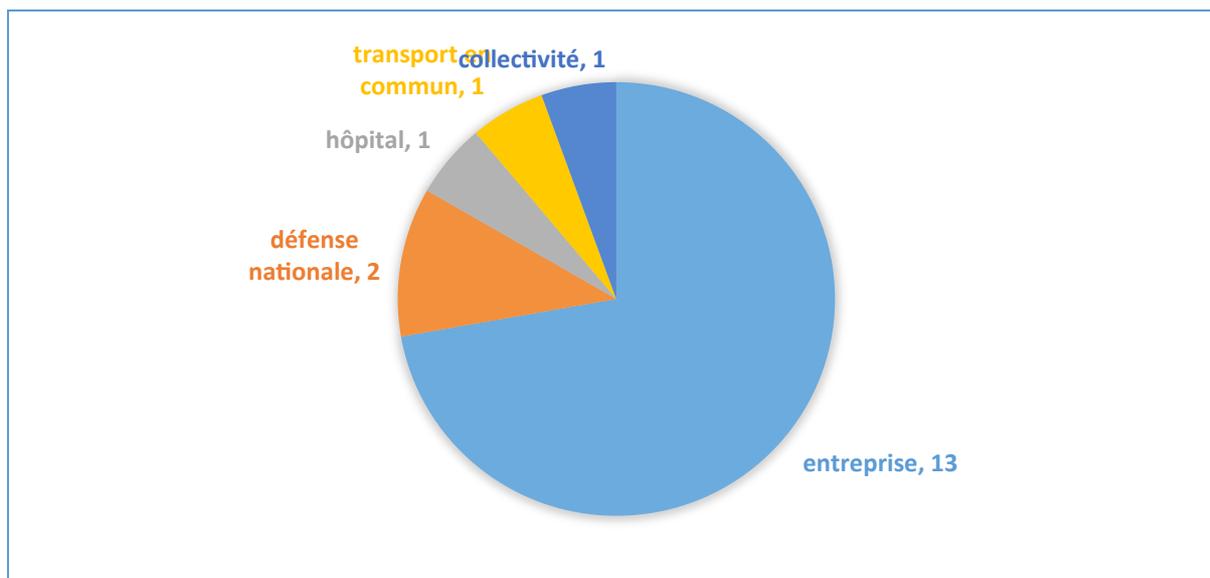


Figure 10 : Le diagramme des études primaires par domaine d'application

Plus précisément le tableau suivant liste les articles par domaine d'application.

Domaine d'application	Année	Auteurs	Source
Entreprise	2010	Lourdes Borrajo et al.	SD
	2007	Suggs et al.	I3E
	2005	Wen et al.	SD
	2004	Wang et al.	I3E
	1993	Baugh et al.	SD
	2006	Bermúdez et al.	SD
	1988	Hruschka et al.	SD
	2009	Eisenreich et al.	ACM
	2007	Wong et al.	SD
	2012	Sato et al.	SD
	1997	Kim et al.	SD
	1980	Lin et al.	SD
	2014	Cha et al.	ACM
Défense nationale	1998	Choi et al.	SD
	2006	Xie et al.	SD
Hôpital	2003	Mayo et al.	I3E
Transport en commun	2008	Wang et al.	SD
Collectivité	2008	Wang et al.	SD

Figure 11 : La liste des études primaires par domaine d'application

La plupart des articles se base sur des méthodes mathématiques afin d'optimiser des systèmes à contraintes multiples.

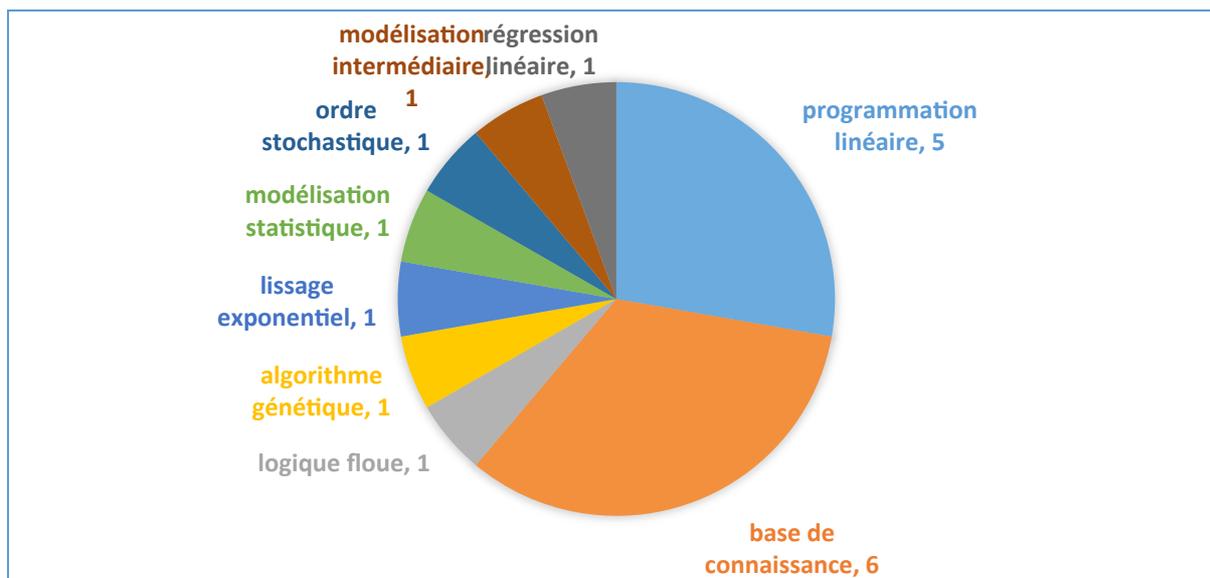


Figure 12 : Le diagramme des résultats obtenus par approche

Les approches de solution proposées sont identifiées dans le tableau suivant.

Objectif	Approche proposée	Année	Auteurs	Source
Optimisation	Base de connaissance	2010	Lourdes Borrajo et al.	SD
		2008	Wang et al.	SD
		2007	Suggs et al.	I3E
		2005	Wen et al.	SD
		2004	Wang et al.	I3E
		1993	Baugh et al.	SD
	Programmation linéaire	2012	Sato et al.	SD
		1998	Choi et al.	SD
		1997	Kim et al.	SD
		1997	Cho et al.	SD
		1980	Lin et al.	SD
	Lissage exponentiel	2006	Bermúdez et al.	SD
	Algorithme génétique	2001	Rees et al.	SD
Régression linéaire	2014	Cha et al.	ACM	
Dynamique des systèmes	2003	Mayo et al.	I3E	
Ordre stochastique	2007	Wong et al.	SD	
Modélisation statistique	2006	Xie et al.	SD	
Modélisation	Logique floue	1988	Hruschka et al.	SD
	Modélisation intermédiaire	2009	Eisenreich et al.	ACM

Figure 13 : Les résultats obtenus par approche

2.2.6 Interprétation des résultats vis-à-vis des questions de recherche

La première question de recherche concerne les approches utilisées concernant la modélisation de la planification financière pour les systèmes d'aide à la décision. Les principales approches proposées sont les bases de connaissance, la programmation linéaire, la logique floue et les algorithmes

génétiques. Elles ont pour objectif d'optimiser un système sous plusieurs contraintes. Cette problématique est orthogonale à la problématique traitée dans cet article qui s'intéresse davantage à la modélisation notamment lorsque plusieurs domaines sont concernés.

La deuxième question de recherche a pour objectif d'analyser comment la modélisation en temps continu a été appliquée aux systèmes d'aide à la décision pour la planification financière afin de positionner l'approche présentée dans cet article en fonction des travaux existants. Notre cartographie systématique n'a trouvé aucun article proposant une approche utilisant la modélisation en temps continu.

La troisième question de recherche concerne les domaines d'application de la modélisation de la planification financière pour les systèmes d'aide à la décision. Comme cela est visible sur la Figure 10, la plupart des articles s'intéressent aux entreprises. Toutefois, quelques articles traitent de défenses nationales, d'hôpitaux ou de transports en commun.

Cette étude corrobore la veille technologique réalisée par MGDIS avant de lancer une recherche coopérative à ce sujet. En effet, plusieurs des méthodes d'optimisation présentées par les articles identifiés par la cartographie systématique ont été expérimentées par l'entreprise. Ces expérimentations ont également montré la nécessité de développer une solution nouvelle, au-delà de l'état de l'art.

La modélisation discrète est le dénominateur commun dans les différentes approches de solution présentées dans la littérature ainsi que dans les approches expérimentées. Ainsi, en conclusion, « dépasser les limitations de la modélisation discrète » a été identifié comme étant le principal verrou technologique.

Notre approche s'attaque à ce verrou technologique en développant une solution novatrice fondée sur la modélisation en temps continu dédiée à la programmation physico-financière.

3 Un cas d'étude

Ce cas d'étude permet d'illustrer la problématique ainsi que l'approche de solution. Dans un premier temps, une programmation budgétaire est détaillée afin de montrer comment interagissent les acteurs qui participent à son élaboration. Ensuite, une chronologie d'événements permet de montrer comment sont réalisées les reprogrammations au cours du suivi.

3.1 Contexte

Ce chapitre relate l'histoire d'une municipalité fictive qui planifie le budget d'une restauration scolaire. De nouveaux quartiers résidentiels sont prévus dans la ville, les infrastructures doivent donc être adaptées à l'accroissement de la population. Une cantine pouvant accueillir plus d'élèves est nécessaire. La cantine actuelle étant éloignée des établissements scolaires, une nouvelle cantine doit être construite.

La planification de la restauration scolaire est réalisée par trois acteurs qui ont chacun une perspective différente. Les élus prennent des décisions politiques en tenant compte du budget. Le service scolaire planifie le budget de fonctionnement de la cantine. La direction des services techniques planifie le budget d'investissement de la construction de la nouvelle cantine.

La figure ci-dessous présente une vue d'ensemble du modèle utilisé au cours de ce scénario. Les flèches symbolisent l'impact du budget de construction sur le budget de fonctionnement.

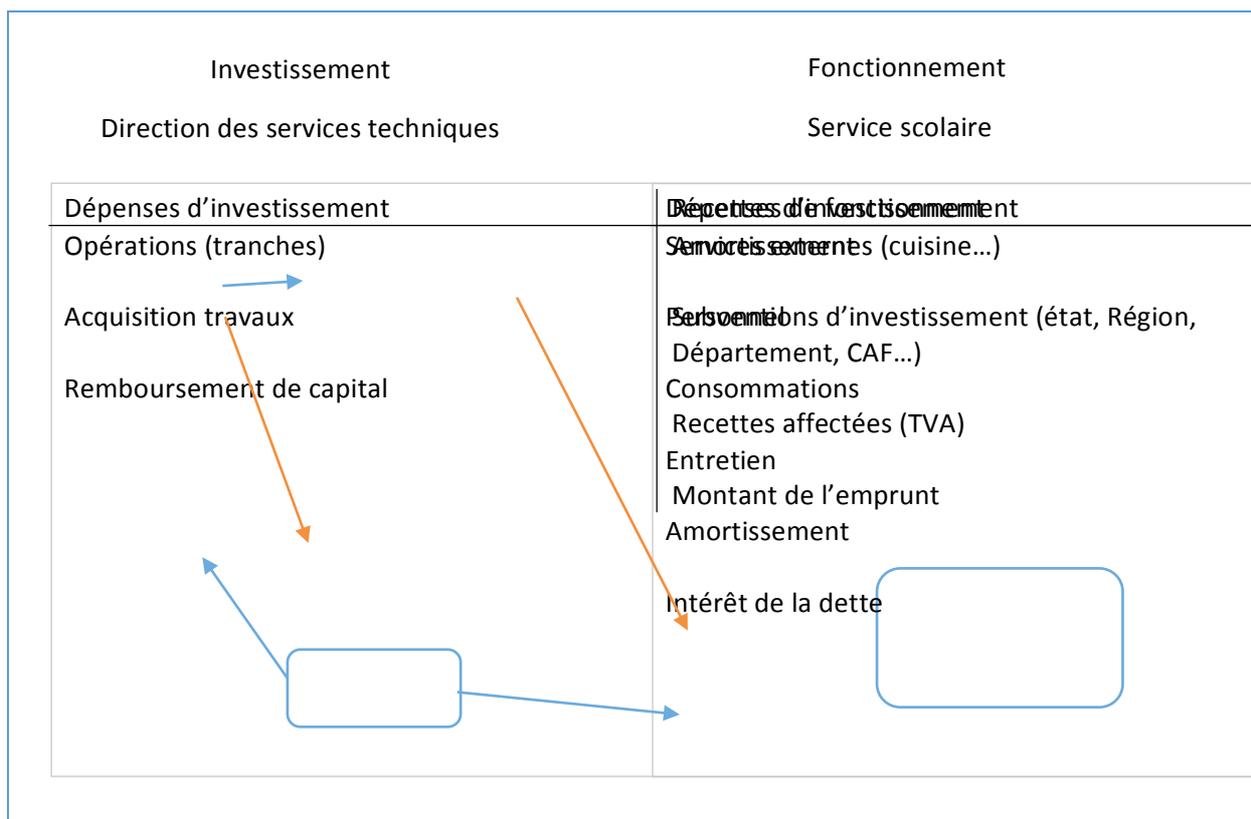


Figure 14 : La vue d'ensemble du budget d'investissement et de fonctionnement

3.2 La planification de l'investissement

La municipalité décide de la construction d'une nouvelle cantine pour remplacer l'existante. Le conseil municipal est réuni afin de décider de l'emplacement et de la capacité de la cantine. Les élus le composant demandent au service technique de leur proposer un budget d'investissement pour cette construction.

La planification du budget d'investissement de la construction d'une nouvelle cantine permet d'estimer les ressources nécessaires et l'impact sur le budget de fonctionnement du service de restauration communale.

3.2.1 L'établissement de la fiche d'opérations

La première étape de la planification de l'investissement consiste à établir la fiche des opérations en déterminant le coût de chaque opération de la construction. Les entreprises qui participent à la construction ne sont pas choisies directement. La collectivité passe par un appel d'offre afin de respecter la loi sur les marchés publics. La rédaction de l'appel d'offre est réalisée en collaboration avec un architecte et un maître d'œuvre. Les entreprises sont amenées à proposer plusieurs offres en fonction de l'efficacité énergétique visée.

Après la date limite de soumission des offres, les offres sont présentées aux élus qui décident lesquelles retenir en concertation avec la direction des services techniques.

3.2.2 La recherche de subventions

Une fois le coût total des opérations connu, le service des finances commence la recherche de subventions. Il envoie des demandes de subventions pour la construction de la cantine à la Caisse d'Allocations Familiales, au Conseil Régional et au Conseil Départemental. Après négociations entre le maire et le département, le Conseil Départemental promet de subventionner 30% du coût de construction avec un plafond fixé à 100 000€.

3.2.3 La recherche d'emprunt

Le conseil municipal délibère afin de déterminer le montant et la durée de l'emprunt. La durée de l'emprunt dépend de la durée d'amortissement de la construction de la cantine. Généralement, un bâtiment est amorti en 20 ou 30 ans. Le service des finances contacte plusieurs banques et retient la banque qui propose la meilleure offre.

3.2.4 La planification physique

La planification physique consiste à placer dans le temps les opérations de construction. La durée de chaque opération est définie dans les réponses aux appels d'offres. Le service technique et celui des finances se concertent pour planifier les dates des opérations.

3.2.5 Le choix de l'investissement de construction de la nouvelle cantine

3.2.5.1 Les deux solutions à départager

Le conseil municipal se réunit afin de décider de l'appel d'offre à retenir. Les réponses au lot de l'isolation ainsi que celui du chauffage ciblent plusieurs catégories d'efficacité énergétique. Pour chacun de ces lots, deux réponses sont plus intéressantes que les autres sans qu'il soit possible de les départager facilement car l'une nécessite un investissement financier moindre alors que l'autre possède une efficacité énergétique plus importante. L'électorat de la municipalité n'étant dans notre cas d'étude pas particulièrement préoccupé par l'écologie, seul l'aspect financier jouera sur la décision de retenir l'une des deux solutions (en fonction ou non d'un retour sur investissement démontré d'une meilleure efficacité énergétique).

Construire une cantine possédant une meilleure efficacité énergétique permettrait de réduire les dépenses de consommation d'énergie. Cependant, le coût de la construction serait plus élevé et ceci entraînerait une augmentation de l'amortissement et des intérêts. Les dépenses de consommation énergétique, l'amortissement ainsi que les intérêts font parties des dépenses de fonctionnement. La réduction d'une dépense entraîne l'augmentation d'une autre et inversement. C'est pourquoi il n'est pas évident d'estimer laquelle des deux solutions est au final la plus avantageuse financièrement.

3.2.5.2 *La recherche des équilibres*

Le service financier apporte son expertise du domaine de la planification financière afin de permettre au conseil municipal de faire un choix. L'objectif est d'établir un équilibre financier pour les deux réponses d'appel d'offre. Pour réaliser l'équilibre du budget de fonctionnement, le conseil municipal peut décider d'augmenter sa participation financière au fonctionnement de la cantine. Cependant, il existe d'autres leviers qui permettent aux élus de réaliser l'équilibre : le prix des tickets de cantine ainsi que la durée d'amortissement.

Le prix des tickets de cantine peut être augmenté dans l'objectif de générer davantage de recettes. Néanmoins, l'augmentation du prix des tickets peut réduire la fréquentation des élèves et atténuer l'effet bénéfique sur l'équilibre.

La durée d'amortissement est choisie en fonction de la durée de vie du bâtiment. Une estimation optimiste de la durée de vie du bâtiment permet de réduire la valeur annuelle de l'amortissement. Par contre, plus l'amortissement est long, plus l'emprunt sera long également et donc les intérêts augmentent. Une fois l'amortissement déclaré, la durée d'amortissement ne peut pas être reprogrammée.

Les dépenses liées aux opérations de construction du budget d'investissement sont amorties sur le budget de fonctionnement. De plus, l'emprunt contracté pour financer ces dépenses génère des intérêts qui font partie des dépenses de fonctionnement. Le budget de fonctionnement doit être équilibré en prenant en compte l'amortissement des opérations ainsi que le paiement des intérêts.

3.2.5.3 *La décision du conseil*

Deux planifications financières sont réalisées : l'une pour la solution à faible investissement et l'autre pour la solution à forte efficacité énergétique. Leurs budgets de fonctionnement sont ajustés au niveau des leviers présentés dans les paragraphes précédents jusqu'à ce qu'un compromis satisfaisant soit atteint avec un budget équilibré. Ces deux planifications financières montrent que la solution à forte efficacité énergétique est plus intéressante financièrement à partir de dix ans de fonctionnement.

Le conseil municipal décide d'augmenter sa participation financière les premières années afin de réduire la participation globale sur plusieurs dizaines d'années. Cette décision est en adéquation avec le plan de réduction des dépenses publiques promis lors des précédentes élections municipales.

3.2.5.4 *La nécessité d'une aide à la décision*

Comme cela a été vu dans les paragraphes précédents, les choix de construction concernant l'efficacité énergétique du bâtiment impactent la programmation physique et donc le budget d'investissement de la construction de la cantine au niveau des dépenses des opérations, de leur amortissement ainsi que de l'emprunt. L'amortissement et les intérêts de l'emprunt font partie des dépenses dans le budget de fonctionnement. L'augmentation de ces dépenses est nuancée par rapport aux économies réalisées sur les dépenses de consommation énergétique.

L'impact des choix concernant les opérations de construction doit être évalué sur le budget afin de pouvoir prendre une décision. Cette évaluation doit être rapide afin d'aboutir à une solution la plus avantageuse possible par rapport aux objectifs de la municipalité. En effet, plus nombreuses sont les options envisagées, moins il y a de risque de passer à côté d'une solution intéressante.

3.3 Le suivi du budget d'investissement

3.3.1 La construction du bâtiment

La construction du bâtiment de la nouvelle cantine est prévue pour se dérouler entre le 1^{er} octobre 2015 et le 1^{er} mars 2016. La construction du bâtiment est décomposée en opérations. Pour simplifier le cas d'étude, ces opérations sont limitées à la maçonnerie, à l'isolation, au chauffage. Chacune de ces 3 opérations est amortie indépendamment à la fin de sa réalisation.

L'opération de maçonnerie est prévue entre le 1^{er} octobre 2015 et le 1^{er} janvier 2016. L'échelonnage du paiement est prévu en trois paiements :

- paiement du premier tiers de la construction du bâtiment le 1^{er} novembre 2015
- paiement du deuxième tiers de la construction du bâtiment le 1^{er} décembre 2015
- paiement du dernier tiers de la construction du bâtiment le 1^{er} janvier 2016

L'opération d'isolation est prévue entre le 1^{er} janvier 2016 et le 15 février 2016. Elle est payée en deux fois :

- paiement de la première moitié de la construction du bâtiment le 8 février 2016
- paiement de la deuxième moitié de la construction du bâtiment le 15 février 2016

L'opération de chauffage est prévue entre le 15 février 2016 et le 1 mars 2016. Elle est payée à la fin des travaux.

3.3.1.1 Le retard dans la construction du bâtiment pour cause d'intempéries

Le 6 novembre 2015, le maître d'œuvre prévient le service financier que la construction du bâtiment va être retardée de 15 jours pour cause d'intempéries. Les tranches de la programmation physique doivent être décalées pour prendre en compte ce retard. Ceci impacte également la programmation financière, et ce aussi bien au niveau de l'investissement que du fonctionnement. En effet, les subventions, les paiements des travaux et les amortissements sont décalés en conséquence. Il est possible que cela entraîne un report de l'ouverture de la cantine. La direction du service technique, le service scolaire ainsi que les élus se concertent donc pour prendre en compte cet imprévu.

3.3.1.2 Le retard dans la construction du bâtiment ne respectant pas le contrat

La construction du bâtiment prend trois mois de retard pour des raisons injustifiées. Le contrat permet jusqu'à deux mois de retard avec des pénalités.

La municipalité hésite à engager des poursuites. Un procès coûte cher. Elle doit être sûre d'avoir assez de trésorerie pour le mener à terme. Le budget futur doit être connu avant de prendre une décision, or cet imprévu nécessite de revoir celui-ci. La municipalité a donc besoin d'obtenir rapidement une nouvelle prévision dans laquelle elle peut avoir confiance.

3.3.2 Les subventions

Le Conseil Départemental subventionne la construction de la cantine en trois versements :

- à 20% de l'avancement des travaux qui est prévu le 14 novembre 2015
- à 40% de l'avancement des travaux qui est prévu le 13 décembre 2015
- à 60% de l'avancement des travaux qui est prévu le 15 janvier 2016

Le deuxième versement de la subvention de la construction de la cantine par le Conseil Départemental qui devait avoir lieu en fin d'année n'est pas reçu à la date prévue. Le service financier estime que ce versement devrait avoir lieu le mois suivant. Le service des finances négocie une ligne de trésorerie avec la banque.

L'emprunt génère des intérêts à rembourser qui s'ajoutent aux dépenses de fonctionnement. Le budget de fonctionnement doit être équilibré afin de compenser cette nouvelle dépense. Afin de rétablir l'équilibre du budget de fonctionnement, des économies sont réalisées sur les repas servis au cours du mois.

3.4 Le suivi du budget de fonctionnement

3.4.1 Le paiement par les familles

Le paiement des repas est effectué par la vente de tickets. Les parents n'ont pas tous la même stratégie pour acheter les tickets de cantine. Certains achètent beaucoup de tickets à l'avance alors que d'autres achètent les billets plus fréquemment. La répartition des achats de tickets est à peu près similaire d'une année sur l'autre. Dans notre exemple, cette répartition possède un pic en début d'année scolaire puis est répartie en fonction de l'affluence. Les tickets achetés en début d'année représentent un surplus de trésorerie important. Ce surplus permet de réduire les emprunts contractés pour la construction.

3.4.2 La gestion des repas

Les dépenses liées à l'achat d'aliments varient au cours de l'année pour plusieurs raisons :

- le prix des fruits et des légumes varie en fonction des saisons
- les commandes sont interrompues pendant les vacances
- les commandes sont réduites les semaines contenant un ou plusieurs jours fériés.

De plus, une incertitude s'ajoute concernant le paiement des fournisseurs au niveau du délai d'encaissement des chèques.

Le Grenelle de l'environnement recommande aux restaurations collectives de composer leurs repas avec 20% d'aliments biologiques. Comme le respect des directives du Grenelle de l'environnement fait partie des promesses électorales du maire, le gestionnaire de la cantine tient compte de cette contrainte. Partant du constat que les aliments biologiques sont moins chers le printemps que l'hiver, il décide de servir des repas sans aliments biologiques l'hiver et de compenser en proposant davantage d'aliments biologiques au printemps.

Le gestionnaire de la cantine choisit cette répartition :

- de septembre à décembre, aucune composante bio ;
- de décembre à mars, deux composantes bio ;
- de mars à juin, une seule composante bio.

3.4.3 Les subventions de la CAF au fonctionnement

Chaque mois, la CAF subventionne la cantine d'un montant de 4 000 €.

3.4.4 Le bilan financier

Le bilan financier est établi à la fin de chaque année. Dans le cas où le suivi n'a pas été réalisé correctement, les événements imprévus ont entraîné des coûts supplémentaires qui n'ont pas été équilibrés par des mesures de réduction de dépenses. Le gestionnaire de la cantine doit demander une participation supplémentaire à la mairie pour équilibrer son budget.

3.4.5 Ouverture de la nouvelle cantine

Le 27 mars 2016, la nouvelle cantine est ouverte. Cette nouvelle cantine a une efficacité énergétique plus avantageuse que l'ancienne. Son ouverture permet de réduire les dépenses de fonctionnement. Ainsi, la participation de la municipalité peut être réduite.

4 L'approche de solution et sa mise en œuvre

Une solution permettant de réaliser des systèmes d'aide à la décision basée sur la modélisation en temps continu nommée « SOFI continu » est comparée à la solution précédemment mise en œuvre par MGDIS qui est représentative des solutions basées sur la modélisation en temps discret nommée « SOFI discret ».

La présentation de la solution est décomposée en plusieurs étapes. Chaque étape suit la même construction. Tout d'abord, un point précis du besoin du client ou du fournisseur est détaillé. Ensuite, l'exposition de l'approche de la solution existante et sa mise en œuvre pour SOFI discret permet de montrer les limites de celle-ci à répondre au besoin. De la même façon, l'exposition de l'approche de la solution proposée et sa mise en œuvre pour SOFI continu permet de voir comment celle-ci répond au besoin. Chacune de ces étapes est conclue en comparant les deux approches.

4.1 Le découpage en services

4.1.1 Le besoin

L'objectif du produit SOFI est de faciliter la planification financière. Les domaines d'application de la planification budgétaire sont variés. Ceci peut concerner une mairie, un centre hospitalier, ou tout autre organisme qui planifie un budget. Chacun de ces domaines possède une logique propre qui est parfois règlementée par des normes. Afin de s'adapter aux spécificités de chaque domaine, le produit SOFI utilise une modélisation de la logique du domaine. Les modèles sont réalisés dans un format commun à tous les domaines d'application de la planification financière.

Les modèles des domaines d'application sont au centre du processus d'utilisation du produit SOFI. Dans un premier temps, le modèle est créé par un expert du domaine. Dans un second temps, les clients saisissent les données connues propres à leur cas, selon le modèle précédemment défini. L'évaluation du modèle contenant cette saisie permet d'obtenir des estimations du futur en jouant sur plusieurs paramètres. Le produit SOFI fournit donc trois fonctionnalités principales : la modélisation de modèles de planification financière, la saisie des données selon un modèle et l'évaluation de celui-ci en fonction d'un jeu de données.

Les besoins concernant le processus dans lequel sont utilisées les fonctionnalités de SOFI varient d'une situation à l'autre. Par exemple, lors de la programmation financière, les participants vont souhaiter visualiser la programmation selon des points de vue différents et évaluer l'impact de choix sur la planification pour faire des propositions aux autres participants. Par la suite, lors du suivi, la mise à jour des données pourrait être réalisée automatiquement afin d'évaluer des indicateurs et éventuellement lever des alertes.

Les processus dans lesquels sont utilisées les fonctionnalités de SOFI permettent de répondre au besoin d'un client. Bien que les fonctionnalités principales intéressent tous les clients, il y a de fortes chances que chaque client ait un besoin spécifique. Les fonctionnalités de SOFI doivent pouvoir être utilisées par de nombreux processus différents.

4.1.2 La solution avec SOFI discret

Le produit SOFI discret est une application complète composée de web services, d'une base de données et d'une interface graphique. Le service contient toutes les fonctionnalités de l'application. Ceci permet un accès par réseau, mais il n'est pas possible d'utiliser ces services dans le cadre d'une architecture orientée service puisque tous les rôles sont confondus à cause d'un couplage entre les services. En effet, les fonctionnalités liées à la modélisation et celle liées à l'évaluation des modèles sont fortement dépendantes de la base de données qui sert à persister les modèles.

4.1.3 La solution avec SOFI continu

L'approche choisie pour SOFI continu est d'adopter une architecture orientée service afin d'intégrer les fonctionnalités de SOFI dans de nombreux processus. Les fonctionnalités de SOFI sont regroupées dans plusieurs services ayant des rôles précis.

La modélisation et l'évaluation des scénarios sont séparées dans deux services, à la manière du patron architectural CQRS (Command Query Responsibility Segregation). Ce patron consiste à séparer les commandes (de modification de scénario) et les requêtes (de calcul) dans deux services différents. Ceci permet d'optimiser le service de calcul indépendamment des problèmes liés à la modification du scénario. Par exemple, la parallélisation du service de calcul est grandement facilitée du fait que ce service traite uniquement des requêtes indépendantes les unes des autres (voir le chapitre 4.3.3.1).

Un scénario est modélisé en envoyant des commandes au service de modélisation. Le scénario est extrait sous forme d'un XML qui est passé au service de calcul pour réaliser des requêtes. Par exemple, une requête peut évaluer l'impact sur le bilan de la ville si la construction d'une cantine est décalée de 6 mois.

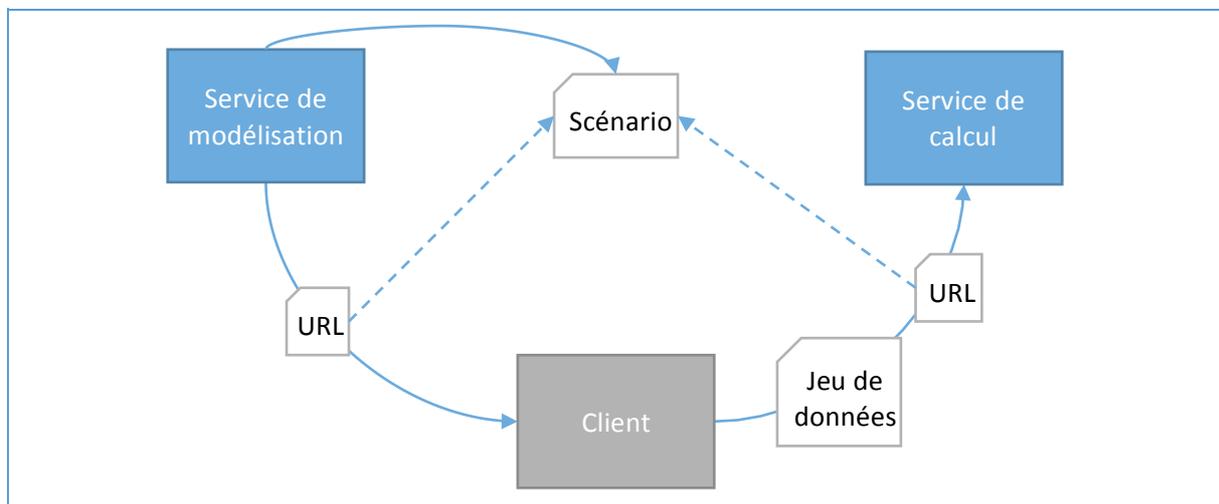


Figure 15 : La communication avec le service de modélisation et de calcul

4.1.3.1 L'évolution des scénarios

D'une requête à l'autre, le scénario reste inchangé. Après plusieurs requêtes, il est probable de vouloir qu'un jeu de données serve de base aux prochaines requêtes. Dans ce cas, les commandes sont envoyées au service de modélisation afin d'intégrer ce jeu de données au scénario. Ce scénario est extrait dans un fichier et son URL est fournie dans les requêtes suivantes.

4.1.3.2 Le contrat du service de calcul

Une requête au service de calcul consiste à appliquer des opérations à un scénario avant d'évaluer les variables spécifiées. Le service a été défini de façon à être « stateless » afin de faciliter la parallélisation des requêtes. Le chapitre 4.3 détaille les approches envisagées pour cette parallélisation. Les réponses d'un service « stateless » ne dépendent pas d'un contexte (ou état) : une requête à un service « stateless » donnera toujours la même réponse. Dans le cas du service de calcul, ceci implique que les modifications apportées par ces opérations ne sont pas conservées d'une requête à l'autre. De plus, le client doit fournir toutes les informations nécessaires à l'évaluation de sa requête car le serveur ne conserve pas de données concernant les requêtes précédentes.

Les opérations appliquées sur les scénarios par le service de calcul sont plus restreintes que celles permises par le service de modélisation. Le chapitre 4.4 explique que les modèles contiennent deux parties : la première concerne la logique du domaine et la seconde les données propres à un cas. Les opérations des requêtes du service de calcul portent uniquement sur la modélisation du cas. La modification de la logique du domaine appartient au rôle du service de modélisation.

```
ComputationResult ModifierRecuperer  
(  
    string scenarioURL,  
    IEnumerable<ScenarioModification> modifications,  
    IEnumerable<ContinuousVariableRead> continuousVariableReads,  
    IEnumerable<ScalarVariableRead> scalarVariableReads  
);
```

Figure 16 : L'opération du contrat du service de calcul

Le service de calcul permet de réaliser trois types d'opération sur le modèle de planification financière :

- une affectation d'une valeur sur une variable continue
- une affectation d'une valeur sur une variable scalaire
- une activation ou désactivation d'une partie du modèle pour exclure certaines valeurs des totaux

En général, l'ordre d'application des opérations de modifications peut influencer sur le résultat obtenu. Dans le cas présent, tous les types de modification semblent indépendants les uns des autres. Par exemple, si une instance est désactivée avant ou après l'affectation d'une variable, ceci n'a pas d'incidence. Les opérations de modification pourraient donc être regroupées par type comme cela a été fait pour les opérations de lecture. Cependant, le service peut être amené à évoluer pour prendre en charge de nouvelles opérations de modification. Comme il n'est pas possible de savoir si ces nouvelles opérations seront indépendantes de celles existantes, le contrat doit permettre de définir l'ordre des opérations sans restriction sur leur type. C'est pourquoi le contrat contient une liste unique pour toutes les opérations de modification.

Ces opérations sont également prises en charge par le service de modélisation. Le service de calcul peut être utilisé pour évaluer plusieurs solutions. Les opérations de la solution choisie sont ensuite appliquées au modèle par l'intermédiaire du service de modélisation. Le nouveau modèle obtenu peut être utilisé avec le service de calcul et servir de base aux futures décisions.

Les valeurs à affecter sont exprimées avec un patron de répartition de la même façon que pour le service de modélisation. Ceci permet de renseigner sur la répartition de la valeur affectée sur un intervalle de temps. Cette façon de représenter les valeurs est décrite dans le chapitre 4.12.

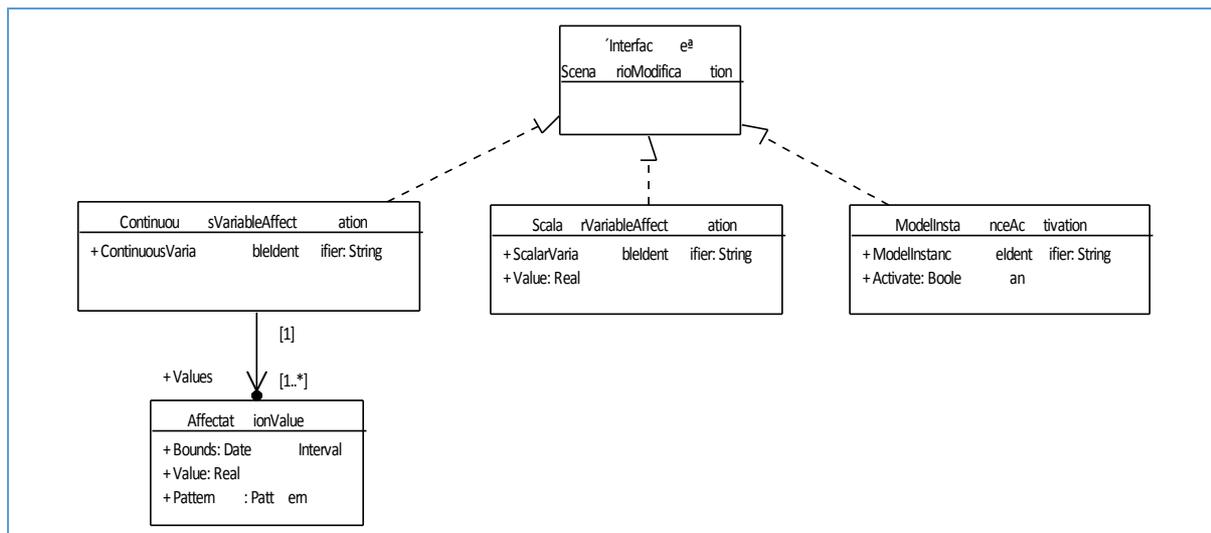


Figure 17 : Le diagramme de classe de la partie des requêtes de calcul qui permet de décrire les modifications à apporter à un scénario

La deuxième partie d'une requête de calcul permet de définir quelles variables doivent être évaluées après que les modifications aient été appliquées au modèle. Bien que les variables soient modélisées avec des fonctions le résultat de leur évaluation est une représentation discrète. Les valeurs de l'évaluation sont positionnées sur une échelle de temps par exemple pour chaque mois et délimitées par des bornes par exemple du 1^{er} janvier 2015 au 1^{er} Janvier 2025. Une requête de calcul peut contenir plusieurs demandes d'évaluation concernant une même variable. Ceci permet d'évaluer une variable sur plusieurs échelles de temps.

La classe « ContinuousVariableRead » représente une requête d'évaluation de variable continue. L'échelle de temps est représentée par la propriété « TimeStep ». Cette propriété est constituée d'un réel et d'une unité de temps. Le premier intervalle est placé à la date de début des bornes de temps représenté par la propriété « Bounds ». La fin du dernier intervalle se place avant la borne de fin de cette propriété ou la superpose.

Le résultat des évaluations est représenté par la classe « ContinuousVariableValue ». Ces propriétés « ContinuousVariableIdentifier », « Bounds » et « TimeStep » permettent de faire le lien avec la demande d'évaluation à laquelle elle répond.

Les variables scalaires peuvent aussi être évaluées afin de récupérer le résultat du calcul d'une date en fonction d'autres variables scalaires par exemple.

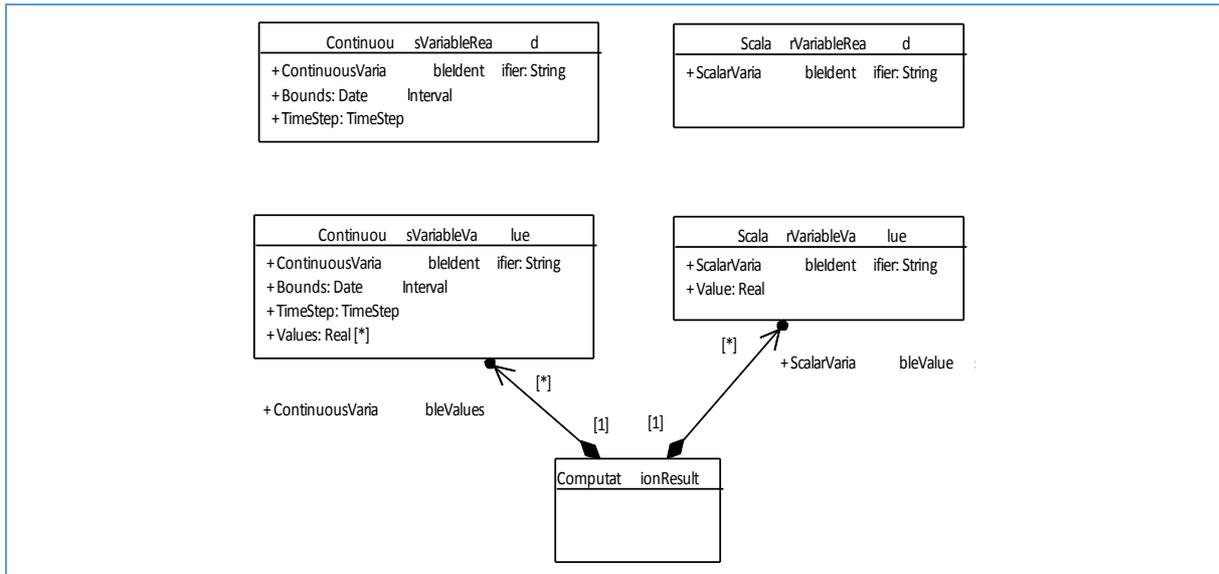


Figure 18 : Le diagramme de classe de la partie du contrat qui décrit les variables à évaluer et le résultat de l'évaluation

La construction d'une requête de calcul est décrite dans cette partie afin d'illustrer le contrat du service de calcul. Cette requête porte sur l'évaluation de l'équilibre de fonctionnement après avoir modifié le montant de l'emprunt du budget d'investissement. L'équilibre est évalué de 2015 à 2045 avec une échelle annuelle, de 2015 à 2025 avec une échelle mensuelle et de 2015 à 2016 avec une échelle hebdomadaire.

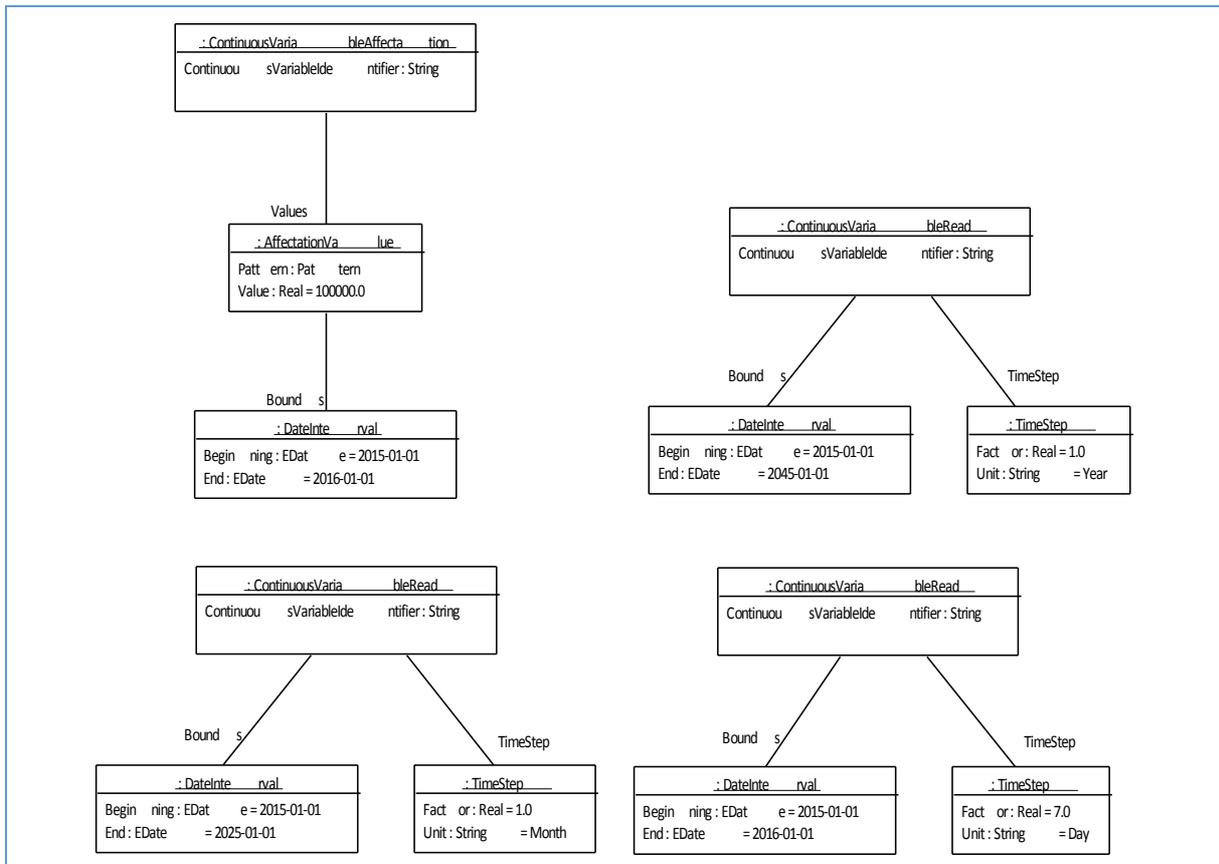


Figure 19 : Le diagramme d'instance d'une requête d'évaluation de l'équilibre du budget de fonctionnement après modification du montant de l'emprunt du budget d'investissement

Le diagramme d'instance ci-dessous présente la réponse du service de calcul à la requête détaillée précédemment. Chaque résultat de l'évaluation de variable continue correspond à une demande d'évaluation de la requête. Le diagramme ci-dessous ressemble au diagramme ci-dessus car les propriétés « ContinuousVariableIdentifier », « Bounds » et « TimeStep » du résultat d'évaluation représenté par une instance de la classe « ContinuousVariableValue » sont identiques à ceux de la demande d'évaluation représentée par une instance de la classe « ContinuousVariableRead ».

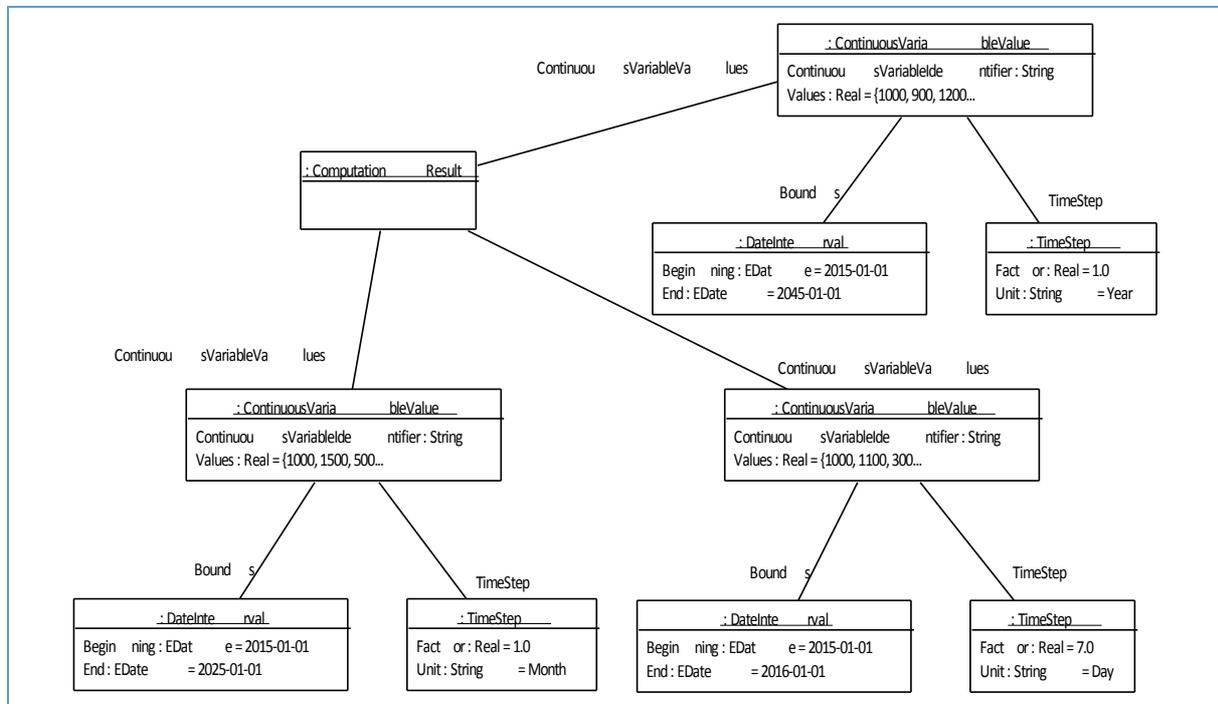


Figure 20 : Le diagramme d'instance du résultat d'une requête d'évaluation de l'équilibre du budget de fonctionnement

4.1.4 La comparaison des solutions

Le produit SOFI discret a été développé dans l'objectif d'obtenir un logiciel à part entière. Il en résulte que les fonctionnalités existantes ne sont pas utilisables dans un autre contexte que celui prévu pour le produit. De plus, l'ajout ou la modification des fonctionnalités peut amener à modifier l'ensemble du produit

Par exemple, le moteur de calcul est intéressant pour d'autres produits, mais l'ancienne implémentation de SOFI ne permet pas de l'utiliser sans installer le produit en entier et donc une base de données complète. De plus, si l'on souhaite paralléliser le moteur de calcul, on se trouvera contraint à modifier du code qui n'est pas lié à cette fonctionnalité.

Avec une approche orientée service, les rôles sont délimités par des contrats de services. Chaque service est défini de façon à pouvoir être utilisé indépendamment. Le processus spécifique à un client est réalisé avec des routes qui relient ces services. Ceci permet au client de construire un système d'information adapté à ses besoins.

4.1.5 Les évolutions possibles

Les besoins initialement décrits ont été remplis. Aucune évolution n'est envisagée actuellement.

4.2 L'évaluation des modèles de planification financière

4.2.1 Le besoin

Les variables des modèles de planification financière doivent être évaluées par le logiciel afin que celui-ci puisse fournir un support à la décision à ses utilisateurs.

4.2.2 La solution avec SOFI discret

Dans le cas de SOFI discret, le moteur de calcul repose sur un tableur. Une variable d'instance est représentée par une ligne où chaque colonne représente un intervalle de temps de durée identique (un an, un mois...).

4.2.3 La solution avec SOFI continu

Dans le cas de SOFI continu, les modèles de planification financière sont transformés en modèles mathématiques. Ces modèles mathématiques sont représentés sous forme de graphes où chaque nœud non-terminal représente une opération de fonctions et les nœuds terminaux des fonctions.

Voici une courte présentation des opérations utilisées :

- « Piecewise », pour définir par morceaux des fonctions
- « Shift », pour décaler dans le temps une fonction
- « Rescale », pour multiplier une fonction par une valeur constante
- « Identity », un opérateur sans effets pour faciliter le branchement du graph
- « Primitive », pour réaliser une somme continue
- « Convolution », expliquée dans le chapitre 4.14

Ainsi que les opérations arithmétiques : l'addition « Sum », la soustraction « Substraction », le produit « Product » et la division « Division ».

Et les fonctions utilisées :

- « Null », une fonction constante de valeur égale à zéro
- « Constant », une fonction constante
- « Affine », une fonction affine
- « Dirac », pour placer une valeur ponctuelle

Chaque variable d'un modèle est transformée en arbre d'opérateurs de fonction continue. Ces arbres sont reliés entre eux pour former le graphe de calcul du scénario. La base de chaque arbre est construite de manière identique. Une variable d'instance peut être vue comme une superposition de trois couches :

- des valeurs ajoutées temporairement lors d'une requête (`valuesMeasure`) ;
- des valeurs par défaut contenue dans le modèle (`defaultValuesMeasure`) ;
- le calcul défini pour la variable (`computationMeasure`).

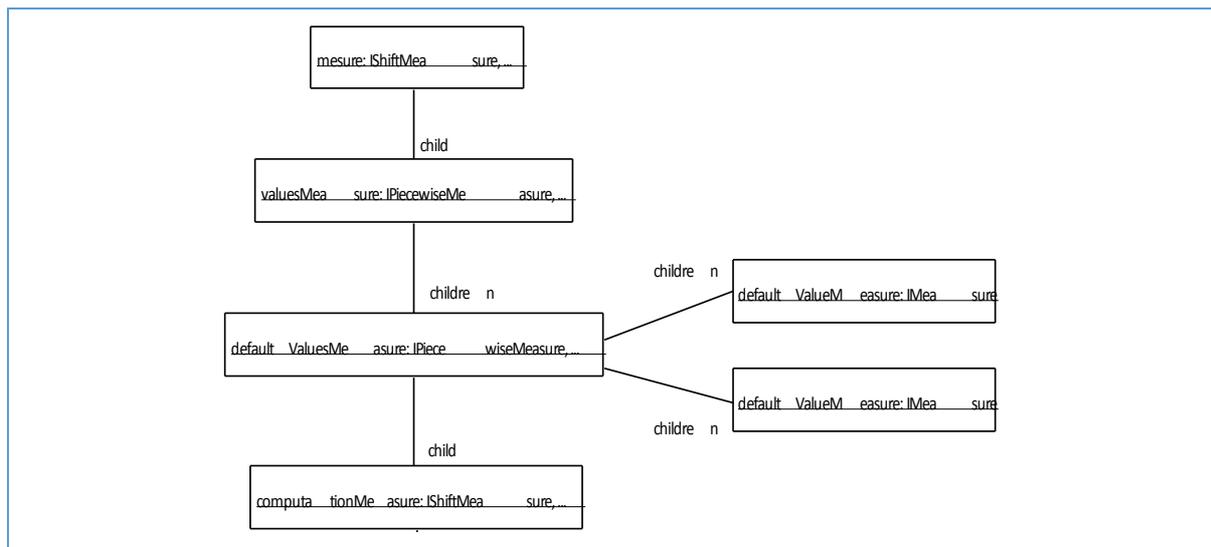


Figure 21 : Le diagramme d'instance de la structure d'une variable traduite en calcul continu

4.2.3.1 Changer une valeur

Les valeurs définies sur une variable d'instance sont modifiées en agissant sur les mesures enfant de « valuesMeasure ».

4.2.3.2 Activer/désactiver l'élément d'instance

Lorsqu'un élément d'instance est désactivé, toutes ses variables sont désactivées en remplaçant l'enfant de l'instance « mesure » par une mesure nulle.

4.2.3.3 Décaler l'élément d'instance

Lorsqu'un élément d'instance est décalé, toutes les valeurs définies sur les variables sont décalées en agissant sur les deux instances de « IShiftMeasure » : « mesure » et « computationMeasure ».

4.2.4 La comparaison des solutions

Dans les deux cas, les modèles de planification financière sont transformés vers des modèles adaptés à la réalisation de calculs. Ces modèles représentent des calculs de façon générale et sont utilisables dans d'autres contextes. Ceci permet de dissocier les problématiques de performance de calcul et d'interprétation des modèles de planification financière.

4.2.5 Les évolutions possibles

Les besoins initialement décrits ont été remplis. Aucune évolution n'est envisagée actuellement.

4.3 La parallélisation du moteur de calcul

4.3.1 Le besoin

Le chapitre de la problématique a exposé les problèmes de temps de réponse rencontrés lors des tentatives d'utilisation des techniques d'optimisation de modèles de programmation financière. Ces techniques d'optimisation, tel que les algorithmes génétiques, nécessitent de réaliser de nombreuses évaluations. Le fait d'exécuter plusieurs évaluations en parallèle permet d'utiliser toutes les ressources à disposition afin d'obtenir un compromis entre la précision et les temps de réponse acceptables.

4.3.2 La solution avec SOFI discret

Dans le cas de SOFI discret, la parallélisation de l'évaluation des modèles n'est pas envisageable car le moteur de calcul est très fortement couplé au modélisateur. Une évolution conséquente comme la

parallélisation du moteur de calcul entrainerait de nombreuses évolutions pour l'ensemble de SOFI qui seraient aussi coûteuses que de repartir de zéro.

4.3.3 La solution avec SOFI continu

4.3.3.1 Les niveaux de parallélisation du service de calcul

Comme cela est détaillé dans le chapitre 4.1, le service de calcul traite uniquement des requêtes. Ceci permet de définir le service de façon « stateless ». Comme les instances du service ne possèdent pas d'états, maintenir une cohérence entre les instances ne pose pas de problème. Les instances du service peuvent simplement être dupliquées et mises en parallèle avec un répartiteur de charge.

Bien que la parallélisation soit grandement simplifiée par la nature du service, l'optimisation de l'occupation de mémoire est importante à considérer car elle peut impacter les performances du service. En effet, pour traiter une requête, une instance du service a besoin du modèle concerné. Le chargement en mémoire vive de ce modèle peut être relativement long. Il y a de fortes chances que de nombreuses requêtes portent sur le même modèle. Pour des raisons de performances, les instances du service conservent dans un cache les derniers modèles utilisés. Plus une instance aura de mémoire disponible, moins il y aura de risque qu'elle se trouve en défaut de cache. C'est pourquoi une stratégie doit être mise en place afin de limiter les besoins en mémoire.

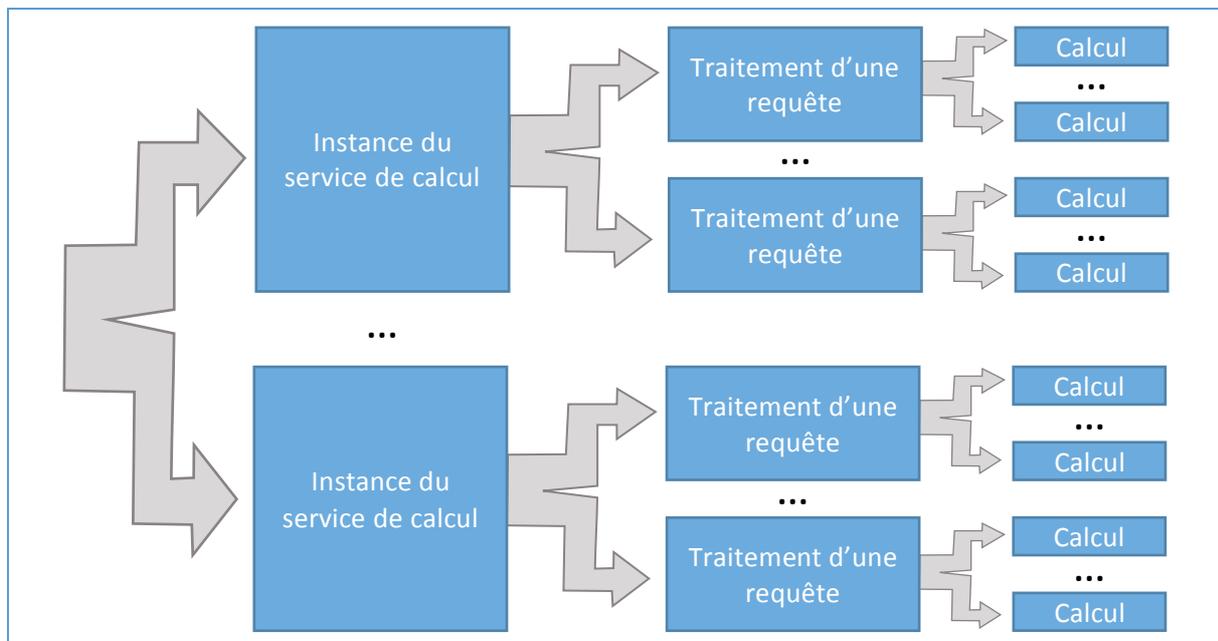


Figure 22 : La parallélisation du service de calcul

Dupliquer des instances du service sur une machine afin de paralléliser l'exécution des requêtes n'est pas optimal. En effet, dans ce cas, le code de l'instance du service est redondant en mémoire. De plus, le cache de plusieurs instances du service pourrait contenir des modèles identiques. Le fait de pouvoir exécuter plusieurs requêtes simultanément sur une même instance du service permet de se limiter à une instance par machine et ainsi réduire l'occupation mémoire en évitant de dupliquer le code et en mettant en commun les modèles mis en cache.

Une nouvelle fois, le fait que le service de calcul soit stateless facilite l'exécution de plusieurs requêtes en parallèle par une instance du service. En effet, les requêtes s'exécutent de façon complètement indépendante les unes des autres.

Lorsque plusieurs machines sont utilisées en parallèle, chaque machine possède une instance du service et donc un cache différent. Afin de limiter le nombre de modèles dans chacun de ces caches,

la répartition des requêtes sur les instances pourrait se faire en fonction du modèle utilisé par la requête.

La parallélisation du service de calcul peut se faire à plusieurs niveaux :

- à l'extérieur du service en répartissant la charge sur plusieurs services ;
- à l'intérieur du service en traitant plusieurs requêtes en asynchrone ;
- à l'intérieur du traitement d'une requête en utilisant une bibliothèque de calcul parallélisée.

Actuellement, aucun niveau de parallélisation n'est implémenté complètement, mais le service a été défini avec la parallélisation comme objectif. Ainsi, la parallélisation à ces trois niveaux peut être ajoutée sans difficultés :

- les instances de services de calcul peuvent être mises en parallèle sans se soucier de leur cohérence puisqu'ils sont stateless.
- un service de calcul pourrait traiter autant de requêtes en parallèle qu'il possède de modèle de calcul en mémoire.
- la bibliothèque LEMF propose une parallélisation des calculs, même si certaines portions de code de cette bibliothèque cassent encore la dynamique de parallélisation.

4.3.3.1.1 La parallélisation des requêtes

L'évaluation d'un modèle de planification financière est réalisée en transformant celui-ci en graphe d'opérateurs de fonction mathématique. Le chapitre 4.2 détaille l'approche prise pour réaliser cette transformation.

Bien qu'un scénario reste inchangé entre deux requêtes, le graphe de calcul correspondant est modifié selon le jeu de données fourni par la requête le temps de l'évaluation. Deux requêtes ne peuvent pas utiliser simultanément le même graphe. Les graphes de calcul sont donc dupliqués autant de fois que nécessaire à la parallélisation.

Lorsque de nombreux scénarios sont utilisés par une instance de service de calcul et que la mémoire vive devient insuffisante, des défauts de cache peuvent nécessiter de recalculer un graphe oublié ou même de recharger en mémoire un scénario. Cette situation augmente grandement le temps de réponse de la requête. Afin de réduire ses occurrences, les requêtes peuvent être réparties en limitant le nombre de scénarios utilisés par chaque service.

Une requête peut prendre moins de temps à s'exécuter si elle est similaire à celle qui a été calculée précédemment sur le graphe. En effet, plutôt que de rétablir le graphe dans son état d'origine, les différences entre les deux requêtes peuvent être appliquées. Ainsi, la modification du graphe est davantage localisée. Les résultats intermédiaires mis en cache par LEMF lors de l'exécution précédente ont plus de chance d'être conservés.

La probabilité d'enchaîner plusieurs requêtes similaires augmente avec le nombre de graphes disponibles. Il peut être intéressant d'augmenter le nombre de graphes au-delà du nombre nécessaire à la parallélisation afin d'exploiter cette optimisation.

4.3.4 La comparaison des solutions

L'approche prise pour SOFI continu a été pensée dès le départ pour permettre une parallélisation des évaluations des modèles de façon à éviter les problèmes rencontrés avec SOFI discret.

4.3.5 Les évolutions possibles

L'approche proposée permet de faciliter la mise en pratique de la parallélisation du service de calcul. La mise en pratique de cette parallélisation est un problème parallèle qui devra faire l'objet d'une analyse davantage approfondie des approches proposées dans ce document.

4.4 La séparation de la logique du domaine et des données propre à un cas

4.4.1 Le besoin

Une programmation financière sert à déterminer les décisions à prendre afin d'atteindre certains objectifs. Lors d'une planification financière, les décisions et les objectifs sont nombreux. Il est difficile, même pour un expert du domaine, de mesurer par une démarche d'analyse l'effet que peut avoir telle ou telle décision. C'est pourquoi l'évaluation du futur en fonction des décisions prises est confiée à un logiciel.

Chaque domaine d'application possède sa propre logique de planification budgétaire. En conséquence, afin d'évaluer le futur concernant un domaine particulier, le logiciel doit suivre la logique de sa planification financière. Les logiciels classiques de planification financière sont prévus pour un domaine d'application spécifique. La logique de ce domaine est directement implémentée dans le logiciel. Etant donné que SOFI doit pouvoir s'adapter à tout type de planification budgétaire, la logique d'un type de planification budgétaire n'est pas directement codée dans le logiciel, mais celui-ci se base sur un interpréteur de modèles. Le modèle spécifique décrivant la logique propre à la planification budgétaire d'un domaine particulier est fourni à SOFI lorsque des estimations doivent être réalisées.

Comme cela a été abordé dans le chapitre 4.1, le processus de création d'un modèle est effectué en deux temps. Dans un premier temps, un expert du domaine modélise la logique de planification budgétaire de son métier. Par exemple un expert de la gestion des centres hospitaliers crée un modèle qui sera ensuite utilisé comme support pour la planification de plusieurs centres hospitaliers. Dans un deuxième temps, le responsable de la planification d'un centre hospitalier rajoute les données propres à son centre en respectant le modèle commun.

Pour que les modèles soient compréhensibles par SOFI, ceux-ci respectent un métamodèle. Ainsi, le logiciel peut traduire un modèle de planification financière en modèle de calculs mathématiques. Le modèle mathématique créé permet ensuite de réaliser les évaluations voulues. Les modèles servent de pont entre la réalité à planifier et le modèle mathématique.

4.4.2 La solution avec SOFI discret

L'implémentation existante du modélisateur a été réalisée avec une base de données relationnelle. Ce choix n'est pas pertinent étant donné que l'intégralité du contenu de la base doit être chargée en mémoire avant de pouvoir réaliser le moindre calcul. Ceci a posé des problèmes de performances qui ont conduit à définir le modèle optimisé pour la lecture et l'écriture en laissant de côté la sémantique.

Ce modèle possède les notions principales exposées dans la partie suivante. D'ailleurs le vocabulaire utilisé est très proche. Cependant il n'est pas suffisamment formalisé. En effet, certaines notions permettant de modéliser la logique du domaine et celle permettant de modéliser le cas d'application de cette logique sont confondues dans la définition du modèle et nécessitent une interprétation de celui-ci afin de les dissocier.

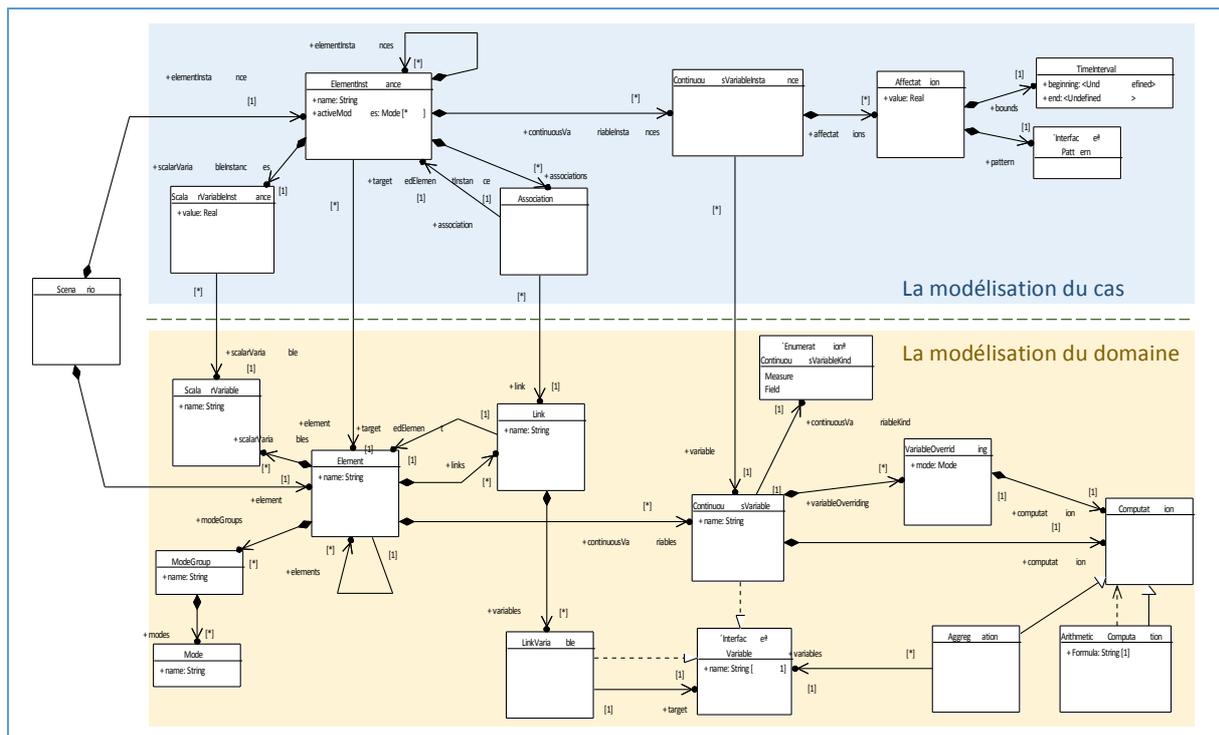


Figure 24 : La vue d'ensemble du métamodèle de la modélisation de SOFI continu

4.4.3.1 Les variables et les affectations

Les calculs modélisés par l'expert du domaine sont définis comme des systèmes d'équations. Les équations mettent en relation des variables continues (ContinuousVariable) et des variables scalaires (ScalarVariable). La valeur d'une variable continue varie en fonction du temps à la manière d'une fonction. L'évaluation d'une variable continue se fait sur un intervalle de temps. Par exemple, l'évaluation d'une variable continue qui représente les recettes de la vente de tickets de cantine du 1^{er} Janvier 2012 au 1^{er} Janvier 2015 correspond au total des recettes au cours de ces trois années. La valeur des recettes de la vente des tickets est liée au nombre de tickets vendus et au prix de vente. Lors de la modélisation, l'expert du domaine définit trois variables continues « recettes », « tickets » et « prix ». Il définit ensuite la relation entre ces trois variables correspondant au système ci-dessous :

$$\begin{cases} \text{recettes}(t) = \text{tickets}(t) \cdot \text{prix}(t) \\ \text{tickets}(t) = \dots \\ \text{prix}(t) = \dots \end{cases}$$

Chaque ligne de ce système est représentée par l'attribut « formula » de la classe « ArithmeticComputation ». Cet attribut est un texte qui respecte le langage décrit dans le chapitre 4.9.3.3.

A l'inverse, la valeur d'une variable scalaire ne dépend pas du temps. Typiquement une variable scalaire est une date. Par exemple, dans le cas d'étude, la date de déménagement de la cantine est une variable scalaire. Une variable scalaire peut aussi représenter une durée (par exemple la durée d'un emprunt). Une variable scalaire peut intervenir dans la définition du calcul d'une variable continue ; par contre une variable scalaire n'est calculée qu'à partir d'autres variables scalaires. La définition du calcul d'une variable scalaire est portée par l'attribut « formula » de la classe « ScalarVariable ». Cet attribut est un texte qui respecte un langage similaire à celui des variables continues, mais permet d'utiliser uniquement des variables scalaires. Ce langage est décrit dans le chapitre 4.9.3.3.

Les valeurs qui concernent un cas de planification budgétaire sont affectées à ces variables. Dans le cas d'une variable continue, l'affectation se place sur un intervalle de temps et la valeur est définie par un poids et une forme. L'affectation d'une variable continue est représentée par la classe « Affectation ». Le chapitre 4.12 détaille la modélisation des affectations de variables continues. Une variable continue est généralement affectée sur plusieurs intervalles. Pour reprendre l'exemple précédent, la variable qui représente le nombre de tickets vendus peut être affectée sur chaque année afin de définir combien de tickets sont vendus au cours de chaque année ainsi que leur répartition au cours de ces années.

Une variable scalaire est affectée par un réel, une date ou une durée. La valeur affectée est représentée par une expression qui permet de calculer une constante. Ceci permet de conserver la logique qui a amené à choisir la valeur à affecter. Sa grammaire est identique à celle des formules de calculs des variables scalaires qui est présentée dans le chapitre 4.9.3.3

4.4.3.2 Les éléments et leurs instances

La planification financière porte sur des entités physiques comme une cantine ou abstraites comme un amortissement. Une entité est modélisée par un système d'équations mettant en relation des variables. Cette partie du modèle décrit la logique qui permet d'évaluer les variables d'une entité en fonction des valeurs affectées à certaines variables.

Néanmoins, le nombre de ces entités diffère d'un cas d'utilisation du modèle à un autre. Par exemple, dans le cas d'étude décrit dans le chapitre 2.2, la planification financière appliquée au budget de fonctionnement d'une cantine est modélisée. Les calculs modélisés portent sur une entité physique qui est une cantine. Le cas d'utilisation du modèle concerne le déménagement d'une ancienne cantine vers une cantine neuve. Dans ce cas, deux occurrences de l'entité physique cantine existent. Le budget de fonctionnement de ces deux cantines suit la même logique, mais leurs valeurs qui caractérisent une cantine (comme l'efficacité énergétique, l'affluence...) sont différentes. En l'exprimant avec les termes du métamodèle : les variables et leur formule sont identiques, mais les affectations sont différentes.

Lors de la modélisation d'un domaine, l'expert du domaine définit les entités. Dans le métamodèle, les entités sont représentées par la classe « Element » qui contient les variables. Lors de la modélisation d'un cas d'utilisation, une entité est instanciée autant de fois que nécessaire pour ce cas d'utilisation. Dans le métamodèle, une instance d'entité est représentée par la classe « ElementInstance ». Son attribut « element » permet de connaître l'élément dont l'instance a été associée.

Les valeurs propres au cas d'utilisation sont définies pour chaque instance d'élément. Les classes « ContinuousVariableInstance » et « ScalarVariableInstance » représentent la relation ternaire entre une affectation, une variable et une instance d'élément. Les deux contraintes OCL de la figure ci-dessous permettent de vérifier que l'élément contenant la variable et l'élément dont provient l'instance soient identiques.

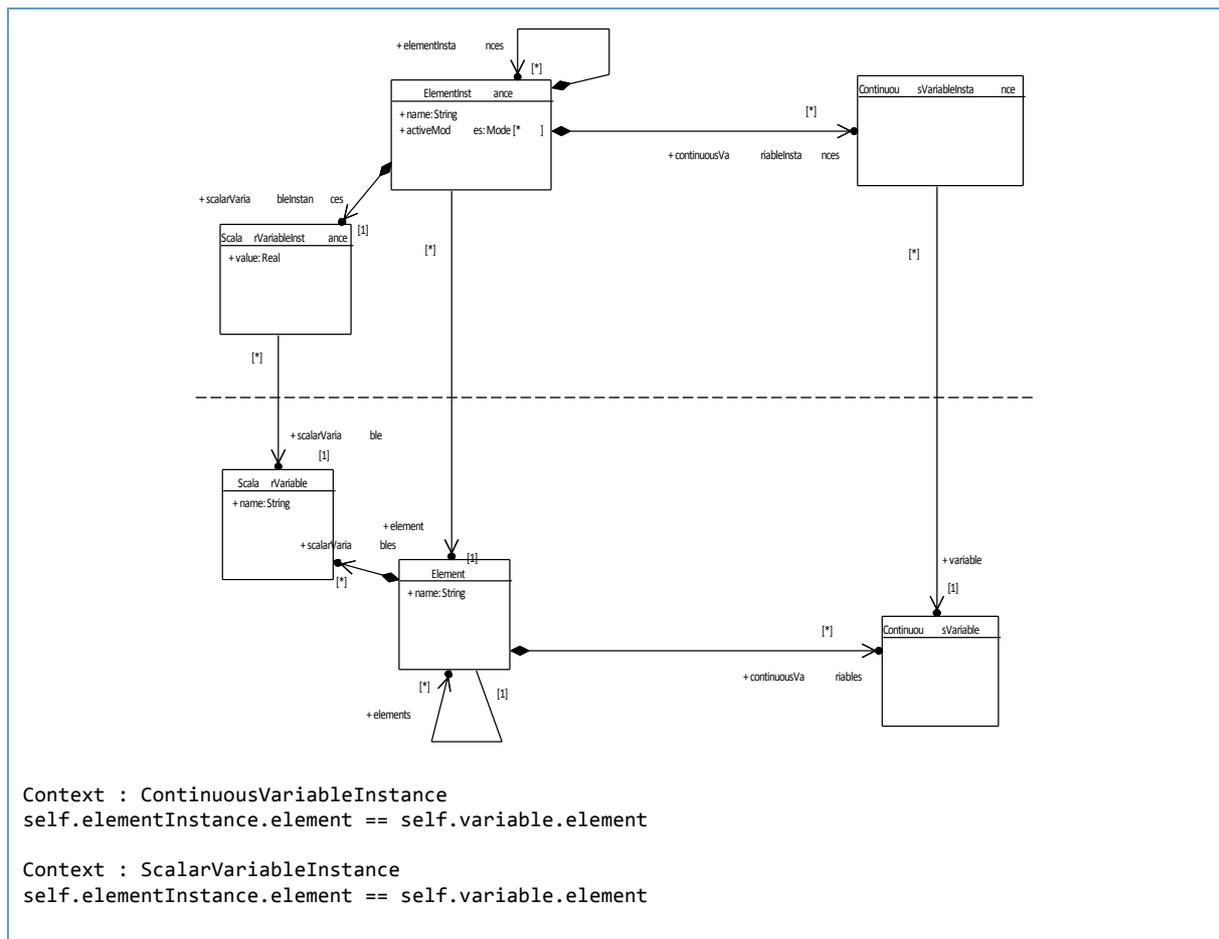
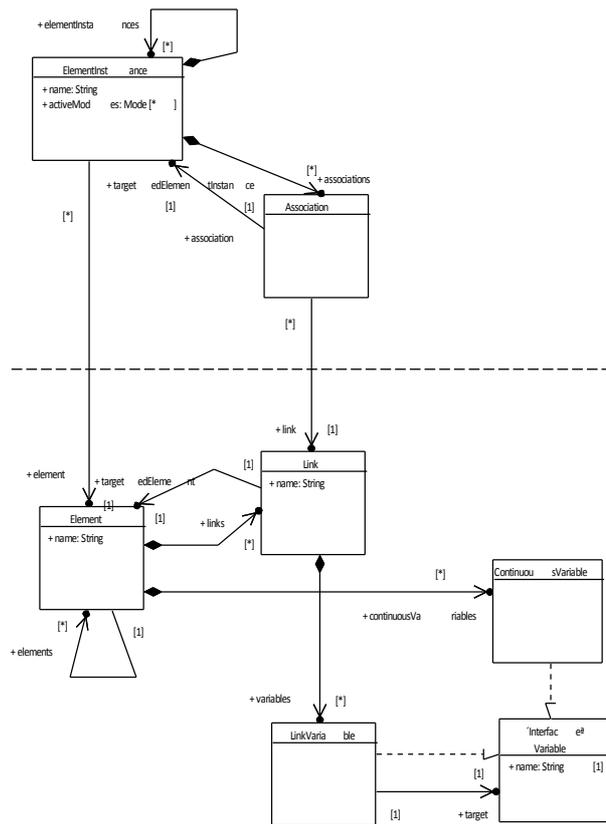


Figure 25 : La contrainte entre les variables d'instances et leur variable

4.4.3.3 Les liens, agrégations et associations

Certains calculs portent sur un ensemble d'entités. Par exemple, l'amortissement de la construction de cantine est calculé à partir de l'amortissement de chacune des opérations de construction. La variable de l'amortissement global réalise la somme des amortissements des opérations. L'expert du domaine définit un lien entre l'élément « Budget d'investissement » et l'élément « Opération » ainsi qu'une agrégation de type somme sur la variable « Amortissement ». Ce lien permet d'associer les instances de l'élément « Opération » à l'instance de l'élément « Budget d'investissement ». Les variables « amortissement » des instances associées sont sommées lors de l'évaluation afin de calculer l'amortissement global de la construction de la cantine.

Deux instances d'élément peuvent être associées ensemble si un lien existe entre deux variables de leurs modèles. Cette contrainte est formalisée par la figure ci-dessous.



Context : Association
 self.targetElementInstance.element == self.link.targetElement
 and self.elementInstance.model == self.link.originElement

Figure 26 : La contrainte entre les associations et leur lien

Un lien permet à un élément source d'utiliser une variable d'un autre élément ciblé. Dans le cas où plusieurs instances de l'élément ciblé sont associées à une instance de l'élément source, le lien fait alors référence à chaque variable de ces instances de l'élément ciblé. Ces variables d'instances doivent être agrégées avant de pouvoir être utilisées dans une formule de calcul. C'est pourquoi une variable de lien est utilisée dans les agrégations uniquement.

La figure ci-dessous présente deux instances d'élément « target 1 » et « target 2 » associées à une instance d'élément « origine 1 ». Dans ce cas, une agrégation de la variable de lien agrègera la variable « targetedVariable » des instances d'élément « target 1 » et « target 2 ».

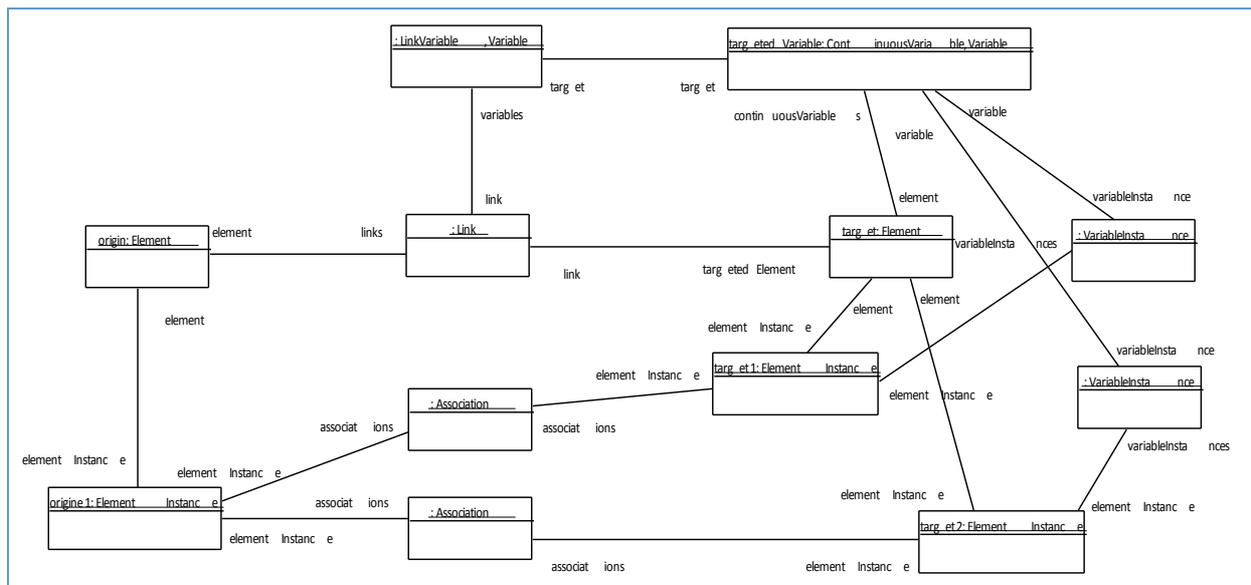


Figure 27 : Un diagramme d'instance illustrant le besoin d'agrégier les variables de lien

4.4.4 La comparaison des solutions

Le métamodèle de SOFI continu fait partie du contrat du service de modélisation et du service de calcul. Il est donc important que celui-ci soit explicite car les modèles pourraient être utilisés par d'autres services. De plus, il doit permettre une évolution de ses fonctionnalités en conservant une compatibilité descendante.

Le métamodèle de SOFI continu reprend le concept de la séparation de la modélisation du domaine et celle du cas déjà existant dans SOFI discret, mais formalise cette séparation de façon nette. Cette séparation rend les modèles davantage explicites et donc sont plus simples à interpréter. L'ajout de métadonnées est également facilité.

4.4.5 Les évolutions possibles

Il est envisagé de permettre de construire un modèle à partir d'une composition d'autres modèles de façon à rendre les modèles plus facilement modulables. Par exemple, ceci permettrait de réaliser un modèle de base d'un domaine et d'y ajouter une logique qui serait spécifique à un sous-domaine. L'approche de modélisation DEVS est une possibilité à envisager pour réaliser ceci [Baati and al. 2007].

4.5 L'instanciation d'éléments

4.5.1 Le besoin

Le besoin est d'instancier des éléments afin de les utiliser dans les jeux de données.

4.5.2 La solution avec SOFI discret

Dans le cas de SOFI discret, le service SOFI permet d'instancier des éléments directement sur le scénario sur lequel sont évalués les jeux de données. Ceci est possible puisque la modélisation et l'évaluation sont réalisées par le même service. Néanmoins, cette opération nécessite de reconstruire le moteur de calcul.

Le moteur de calcul aurait pu prendre en charge l'opération d'instanciation directement, mais ceci aurait demandé un développement qui a semblé trop conséquent au regard de l'optimisation qui en aurait découlé.

4.5.3 La solution avec SOFI continu

Comme cela est détaillé dans le chapitre 4.1, les requêtes envoyées au service de calcul contiennent des opérations à appliquer sur les instances d'éléments et leurs variables. Les identifiants servant de références à ces entités sont définies par le service de modélisation lors de leur création. Inclure l'opération d'instanciation parmi les opérations utilisables dans une requête de calcul obligerait le service de calcul à gérer également la définition des identifiants. Afin de bien délimiter les rôles entre le service de calcul et de modélisation, les opérations de créations d'entités du modèle sont exclusivement prises en charge par le service de modélisation.

Lorsque des requêtes de calcul nécessitent de nouvelles instances, le modèle doit être modifié en conséquence en utilisant le service de modélisation. Néanmoins, les opérations d'activation et de désactivation d'instance d'élément peuvent être utilisées afin d'éviter de passer par cette étape qui peut être coûteuse en temps d'exécution. Toutes les instances nécessaires aux jeux de données sont alors instanciées par le service de modélisation au préalable et sont activées ou non par le service de calcul en fonction des requêtes de calcul.

4.5.4 La comparaison des solutions

Ces deux solutions découlent du choix de séparer ou non la modélisation et l'évaluation en deux services. Ce choix est détaillé dans le 4.4.

4.5.5 Les évolutions possibles

Les besoin initialement décrits ont été remplis. Aucune évolution n'est envisagée actuellement.

4.6 La complexité des évaluations

4.6.1 Le besoin

L'évaluation des modèles de programmation financière est réalisée sur des échelles de temps différentes en fonction de l'intention de l'utilisateur. Par exemple, les élus choisiront une périodicité annuelle sur un intervalle de 30 ans afin d'avoir une vision à long terme alors que le gestionnaire de la cantine choisira une périodicité d'un jour sur un intervalle d'une année pour réaliser le suivi du budget de fonctionnement.

Il est expliqué dans la problématique que l'approche choisie pour permettre une communication efficace des acteurs participant à la programmation financière est d'utiliser un modèle de programmation financière unique. Un modèle calculable sur 30 ans possède plus de 10 000 jours. L'évaluation complète d'un modèle contenant plusieurs milliers de variables demande un temps conséquent. Ce délai n'est pas envisageable étant donné que les évaluations du modèle servent de support à la discussion. Lorsque le gestionnaire de la cantine évalue l'impact des propositions soumises lors d'une réunion sur la situation pour l'année courante jour par jour, il s'attend à obtenir le résultat en quelques secondes.

Afin de permettre des temps de réponses acceptables, Il faut éviter de rendre la complexité de l'évaluation d'un modèle (pour un intervalle et une périodicité donnée) dépendante du nombre de périodes totales contenues dans ce modèle.

4.6.2 La solution avec SOFI discret

SOFI discret se base sur le fonctionnement des tableurs. L'estimation du futur à partir du passé et les calculs d'emprunts sont réalisés avec des suites numériques. Ceci induit une dépendance horizontale entre les cellules qui s'ajoute à la dépendance verticale qui vient du fait que les variables sont calculées en fonction d'autres variables. Lors de l'évaluation d'une variable pour une période, une grande partie du modèle peut nécessiter une évaluation.

La dépendance horizontale est visible dans le cas du calcul de l'accumulation de la participation de la mairie. A cause de l'écriture récursive, pour calculer l'accumulation arrivée à une période donnée, toutes les périodes précédentes doivent être évaluées.

```
" Accumulation..."[n] := "Accumulation..."[n - 1] + "Participation de la mairie"[n - 1]
```

Figure 28 : La formule de la variable « Accumulation de la participation de la mairie » en modélisation discrète

La complexité de l'évaluation d'une variable d'un modèle sur un intervalle est linéairement dépendante au nombre de périodes contenues dans ce modèle. Ceci entraîne de très long temps de réponse lorsqu'un modèle de plusieurs dizaines d'années est découpé par périodes d'un jour. C'est pourquoi l'implémentation de SOFI discret se limite aux périodes annuelles et mensuelles.

4.6.3 La solution avec SOFI continu

L'approche prise pour SOFI continu afin d'évaluer un modèle de programmation financière est de le retranscrire en modèle mathématique à base de fonctions.

Certaines fonctions n'ont pas d'influence sur la taille de l'intervalle calculé. C'est le cas de la fonction exponentielle qui est utilisée pour modéliser l'inflation dans l'exemple présenté dans le chapitre 4.13.5.

D'autres fonctions nécessitent l'évaluation de leurs opérandes sur un intervalle plus large que l'intervalle d'évaluation. C'est le cas de l'opération d'accumulation dont les valeurs à un instant donné dépendent des valeurs précédentes. Cependant le modèle mathématique permet d'évaluer une variable pour un intervalle quelconque. La somme des valeurs précédentes peut hypothétiquement être réalisée en une seule évaluation. En réalité, certaines opérations peuvent nécessiter une discrétisation de cet intervalle, un produit de deux fonctions par exemple. Dans ce cas, la taille des intervalles de discrétisation est déterminée en fonction de la précision souhaitée. Ceci est important car la quantité de calculs réalisés n'est pas dépendante de la périodicité d'évaluation, mais est un compromis entre les performances et la précision.

4.6.4 La comparaison des solutions

Les problèmes de performance qui empêchent l'évaluation d'un modèle sur des périodicités fines avec SOFI discret sont levés avec l'approche définie pour SOFI continu. Ceci permet à un modèle unique d'être évalué avec de temps de réponse corrects sur des échelles de temps adaptées à chaque participant de la programmation financière.

4.6.5 Les évolutions possibles

Les besoin initialement décrits ont été remplis. Aucune évolution n'est envisagée actuellement.

4.7 La distinction entre les mesures et les champs

4.7.1 Le besoin

Dans les modèles de planification financière de SOFI, il existe deux types de variables continues : celles exprimées par unité de temps et celles dont l'unité est indépendante du temps. Par exemple, les recettes provenant de la vente des tickets de cantines sont calculées à partir de deux variables : la quantité de tickets vendus et le prix des tickets. Pour exprimer une moyenne, dans le premier cas, l'unité de temps est nécessaire pour que la phrase ait du sens : « La quantité moyenne de tickets vendus est de 10000 par an » alors que dans le deuxième cas la moyenne ne dépend pas d'une unité de temps : « Le prix moyen d'un ticket est de 2€ ».

La distinction entre les variables dont l'unité dépend du temps et celles dont l'unité en est indépendante est importante car elle nécessite une méthode d'évaluation différente. En effet, dans le premier cas, l'évaluation pour un intervalle de temps est égale à la somme des évaluations des partitions de cet intervalle. Par exemple, la quantité de tickets vendus au cours d'une année est égale à la somme des tickets vendus chaque mois de cette année. Dans le deuxième cas, une telle somme n'aurait pas de sens. En effet, le prix d'un ticket au cours d'une année n'est pas la somme des prix de chaque mois, mais pourrait être une moyenne pondérée par le nombre de tickets vendus chaque mois ou bien le prix à un certain instant, etc.

4.7.2 La solution avec SOFI discret

Dans le cas de SOFI discret, cette distinction n'est pas explicitée dans la modélisation. En s'appuyant sur le fait que la périodicité des variables est connue, il est possible de définir une formule de calcul qui donnera le résultat désiré. Cependant, cette formule devra être réécrite si la périodicité d'une des variables utilisée change.

4.7.3 La solution avec SOFI continu

4.7.3.1 La distinction au niveau du métamodèle

La modélisation de planification financière de SOFI continu, présentée au chapitre 4.4, fait la distinction entre les deux types de variable grâce à la propriété « kind » de la classe « ContinuousVariable ». Cette propriété est une énumération qui peut prendre deux valeurs : « Measure » dans le cas d'une variable dont l'unité dépend du temps et « Field » dans le cas d'une variable dont l'unité en est indépendante. Ce vocabulaire vient des mathématiques et est également utilisé par l'interface entre le moteur de calcul de SOFI et la bibliothèque de calcul numérique.

Les modèles de SOFI continu représentent les variables continues sous forme de fonctions. La nuance entre les deux types de fonctions se fait sur la façon de lire les valeurs sur ces fonctions. Dans le cas d'une variable de type « Measure », une valeur est évaluée pour un intervalle de temps, par exemple, le nombre de tickets de cantine vendus entre le 1^{er} Mai 2014 à minuit et le 1^{er} Juin 2014 à minuit. La valeur correspond à l'aire sous la courbe.

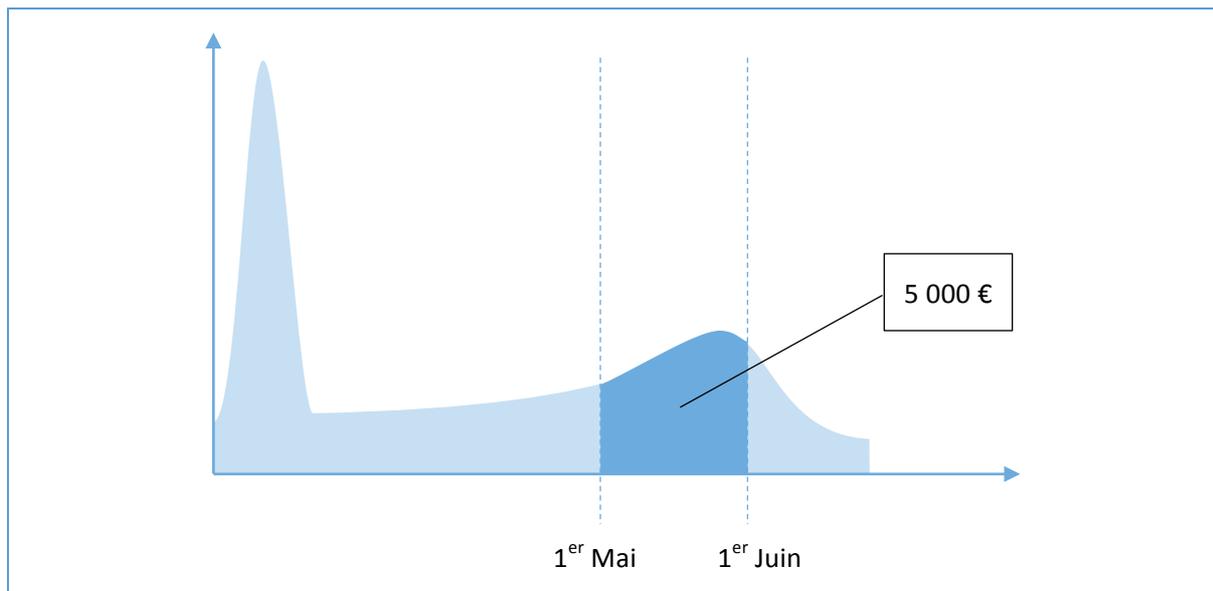


Figure 29 : L'interprétation de la valeur d'une mesure

Dans la partie précédente qui décrit le besoin, il a été montré que pour ce type de variable : « l'évaluation pour un intervalle de temps est égale à la somme des évaluations des partitions de cet intervalle ». C'est bien le cas ici puisque la partition de l'intervalle revient en quelque sorte à « découper » l'aire sous la courbe en lamelles.

Dans le cas d'un variable de type « Field », une valeur est évaluée pour un instant précis, par exemple, le prix des billets le 1^{er} Mai 2014 à minuit.

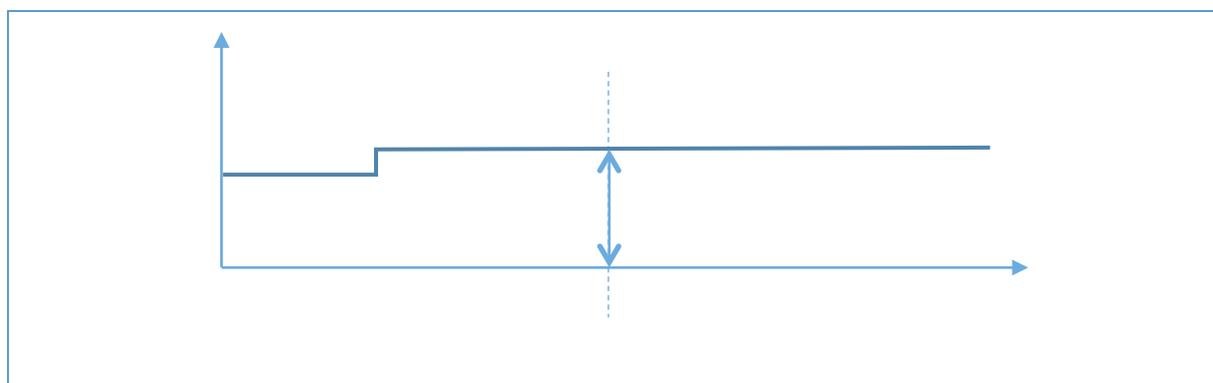


Figure 30 : L'interprétation de la valeur d'un champ

Par la suite, les variables de type « Measure » seront désignées par le terme « mesure » et les variables de type « Field » par le terme « champ ».

L'interface entre le moteur de calcul de SOFI et la bibliothèque de calcul numérique fait aussi la distinction entre mesures et champs.

Une mesure telle que définie par l'interface est une fonction $f_M(S)$ qui, à chaque intervalle S sous la forme $[a, b[$, associe un réel. Ce réel est l'intégrale de a à b d'une fonction f . Les mesures sont donc évaluées sur un intervalle de temps.

Un champ tel que défini par l'interface est une fonction dont l'image peut être évaluée pour un instant donné.

4.7.4 La comparaison des solutions

La distinction entre les champs et les mesures dans SOFI continu permet d'obtenir des modèles davantage explicites par rapport aux modèles de SOFI discret.

4.7.5 Les évolutions possibles

La vérification des modèles de SOFI continu ne prend pas en compte l'unité complète des variables mais uniquement le fait qu'elle dépend ou non du temps. Une addition de tickets par unité de temps avec des euros par unité de temps, bien qu'étant incohérent, ne serait pas détecté comme une erreur. Il pourrait être envisagé d'exprimer l'unité complète des variables de façon à détecter ce genre d'erreurs. Cependant, une grande majorité des variables du modèle du cas d'étude sont exprimées en euros par unité de temps. Donc, la probabilité de ce genre d'erreur de modélisation est faible pour le cas étudié. C'est pourquoi cette possibilité n'a pas été implémentée dans le prototype.

4.8 La modélisation des durées

4.8.1 Le besoin

Une durée de temps n'a pas la même valeur en fonction du contexte. Par exemple, dans le cas d'un calendrier grégorien, une durée d'un mois ne contient pas le même nombre de jours si l'on considère le saut d'un mois du 1^{er} janvier au 1^{er} février ou bien du 1^{er} février au 1^{er} mars.

En finance cette difficulté est parfois contournée par l'utilisation d'un calendrier spécifique. Dans ce calendrier les années sont composées de 12 mois de 30 jours. Certaines dates d'un calendrier n'existent pas dans l'autre et inversement.

4.8.2 La solution avec SOFI discret

Comme cela a été vu dans le chapitre 4.2, les modèles de SOFI discret ne permettent pas d'exprimer des durées inférieures à un mois. La modélisation de SOFI discret se rapproche de la logique du calendrier fiscale puisque le nombre de jours contenu dans un mois n'est pas pris en compte par les modèles.

4.8.3 La solution avec SOFI continu

La modélisation continue de SOFI représente les durées avec une unité : année, mois, semaine et jour. L'unité est interprétée en fonction du contexte. Par exemple, lors de l'ajout d'un mois à la date du présent, comme modélisé par la formule ci-dessous, le décalage équivalent en jour dépendra du calendrier utilisé et de la date du présent.

```
"Dans un mois"(t) := "Présent"() + Month(1)
```

Figure 31 : Un exemple de formule de calcul de date

Communément, lorsque l'on ajoute un mois on essaie de conserver le numéro du jour. Dans le cas du calendrier grégorien, si le présent est un 31 mars, le résultat sera le 30 avril. En ajoutant deux mois au 31 mars, le résultat est le 31 mai alors que l'ajout d'un mois au 30 avril donne le 30 mai. Ceci pose problème car l'ajout d'un mois consécutivement dans deux formules du modèle doit donner le même résultat que l'ajout de deux mois dans une seule formule. Le même problème se pose pour les années et le 29 Février.

```
"Dans un mois"(t) := "Présent"() + Month(1)  
"Dans deux mois"(t) := "Dans un mois"() + Month(1)
```

```
"Dans deux mois"(t) := "Présent"() + Month(2)
```

Figure 32 : La comparaison de formule de calcul d'un décalage de deux mois ou de deux fois un mois

Il pourrait être envisagé de réécrire les formules automatiquement afin d'éviter l'ajout de durées successives. Cependant, il est possible que l'on souhaite connaître le numéro du jour « imaginaire » d'une date calculée dans le cas où d'autres calculs devraient être effectués à l'extérieur du moteur de calcul.

4.8.4 La comparaison des solutions

L'approche pour manipuler les dates de SOFI continu prend en compte les spécificités des calendriers dans l'objectif de réaliser des estimations plus fines.

4.8.5 Les évolutions possibles

Les besoin initialement décrits ont été remplis. Aucune évolution n'est envisagée actuellement.

4.9 La modélisation des équations

4.9.1 Le besoin

SOFI permet de modéliser les domaines d'applications concernés par la planification financière. La logique des domaines est modélisée par des équations dont les variables évoluent en fonction du temps. Ces équations sont écrites par les experts pour modéliser la logique de leur domaine et interprétées par SOFI lors de l'évaluation des modèles. Il est important que ces équations soient exprimées dans un langage qui soit assez intuitif et précis afin que les experts aient une interprétation identique à celle du service de calcul. Dans le cas contraire, les calculs effectués par le service de calcul ne correspondraient pas à ceux prévus par les experts et seraient donc inexploitable.

4.9.2 La solution avec SOFI discret

La grammaire s'inspire de celle utilisée par les tableurs afin de faciliter l'adaptation des personnes habituées à ceux-ci. La principale différence avec les formules de tableur est que les références à d'autres cellules avec des coordonnées (ligne et colonne) sont remplacées par des références à d'autres variables avec comme coordonnées le nom de la variable et une période de temps.

```
"Equilibre"[n] := "Recettes"[n] - "Dépenses"[n]
```

Figure 33 : Un exemple de formule pour une variable « Equilibre » en modélisation discrète

Les chapitres suivants (4.10, 4.11, 4.12, 4.13) font le parallèle entre la modélisation des formules pour SOFI discret et SOFI continu pour des exemples choisis.

4.9.3 La solution avec SOFI continu

4.9.3.1 Les opérateurs de variable continues

Les formules de calculs permettent d'exprimer une variable en fonction d'autres variables de leur élément grâce à des opérateurs. Ces opérateurs contiennent certains opérateurs classiques de l'arithmétique comme l'addition, la soustraction, la multiplication et la division.

La formule ci-dessous présente l'utilisation de l'opérateur de soustraction entre les variables continues « Recettes » et « Dépenses ». Les variables continues sont suffixées de « (t) » afin de montrer que leur valeur dépend du temps.

```
"Equilibre"(t) := "Recettes"(t) - "Dépenses"(t)
```

Figure 34 : Un exemple de formule pour une variable « Equilibre » en modélisation continue

Les formules de calcul doivent respecter certaines contraintes concernant le type des variables utilisées afin d'être valides. En effet, la hauteur entre l'axe et la courbe des fonctions qui

représentent les variables n'ont pas la même unité si la variable est de type « Measure » ou « Field ». Ces contraintes portent sur les opérandes et le résultat des opérateurs. Le tableau ci-dessous répertorie les contraintes concernant les opérateurs arithmétiques. L'unité des variables de type « Measure » est notée « $u \cdot t^{-1}$ » et celle des variables de type « Field » est notée « u ».

Opérateur	Résultat	1 ^{ère} opérande	2 ^{ème} opérande
+	u	u	u
+	$u \cdot t^{-1}$	$u \cdot t^{-1}$	$u \cdot t^{-1}$
*	u	u_1	u_2
*	$u \cdot t^{-1}$	u_1	$u_2 \cdot t^{-1}$

« u » est une unité quelconque indépendante du temps (ex : €, ticket...)

Figure 35 : Les contraintes sur le type des variables continues en fonction des opérateurs arithmétiques

Lors de la vérification de la validité d'un modèle, une formule de calcul est transformée en arborescence sémantique qui respecte un modèle contenant les contraintes de type.

La somme de deux variables est possible uniquement si celles-ci sont toutes les deux des mesures ou toutes les deux des champs. La somme de deux mesures est une mesure et la somme de deux champs est un champ.

Le produit de deux variables est possible uniquement si l'une d'entre elles est un champ. Le produit de deux champs est un champ et le produit d'un champ et d'une mesure est une mesure.

D'autres opérateurs spécifiques aux formules de calcul de SOFI continu permettent d'exprimer des relations qui font intervenir des notions de temps. Ces opérateurs sont décrits dans les parties suivantes.

4.9.3.1.1 L'opération conditionnelle « If »

L'opérateur conditionnel « If » permet de définir un calcul différent en fonction du temps. Ces calculs peuvent être vus comme des fonctions définies par morceaux. Les frontières de ces morceaux sont définies par une expression temporelle.

Le premier paramètre de l'opération « If » est une condition qui dépend du temps. Le deuxième paramètre est la valeur donnée au résultat lorsque cette condition est vérifiée et inversement pour le troisième paramètre.

If(: condition, si vrai : valeur, si faux : valeur) : valeur

Figure 36 : La signature de l'opération « If »

Dans l'exemple suivant, la variable « Dépenses » est calculée différemment sur le passé, l'année du présent et l'avenir. Les expressions conditionnelles sont détaillées dans les chapitres 4.10 et 4.11.

"Dépenses"(t) := If(t < "déménagement"(), "Dépenses avant le déménagement"(t), "Dépenses après le déménagement"(t))

Figure 37 : Un exemple de formule pour une variable « Dépenses »

4.9.3.1.2 L'opération de décalage temporel « Delay »

La référence au passé est réalisée avec l'opération « Delay ». Cette opération permet de décaler sur l'axe du temps les valeurs d'une variable. Le choix de la durée de décalage est totalement libre. Ceci permet d'exprimer précisément le lien entre le passé et le futur.

L'opérateur « Delay » prend comme paramètre une durée de décalage et une valeur continue qui peut être aussi bien une mesure qu'un champ. Le chapitre 4.7 explique la nuance entre un champ et une mesure.

Delay(durée : valeur scalaire, : valeur continue) : valeur continue

Figure 38 : La signature de l'opération « Delay »

L'exemple ci-dessous est une des étapes de l'exemple de remboursement de TVA décrit dans le chapitre 4.13.3. Les valeurs de la variable calculée « Décalée » sont retardées de 3 mois et 4 jours par rapport à la variable « Source ». Le chapitre 4.8 détaille les problèmes liés à la modélisation des durées.

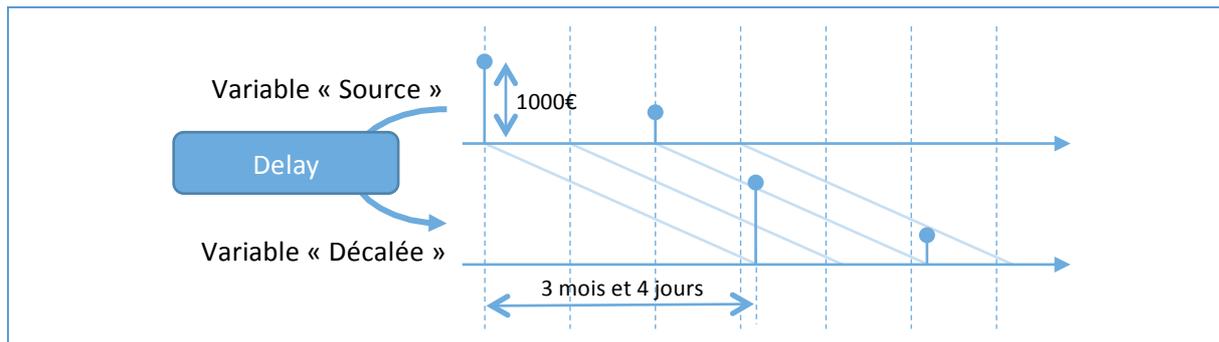


Figure 39 : Un exemple d'application de l'opération « Delay »

```
"Décalé"(t) :=
Delay
(
  Month(3) + Days(4),
  "Source"(t)
)
```

Figure 40 : Une formule de décalage en modélisation continue

Il n'y a pas de contrainte sur le choix de l'intervalle d'évaluation. Ainsi, la variable « Décalée » peut être évaluée pour tout intervalle souhaité. Comme le montre la figure ci-dessous, l'opération « Delay » conserve tout le détail de la variable « Source ».

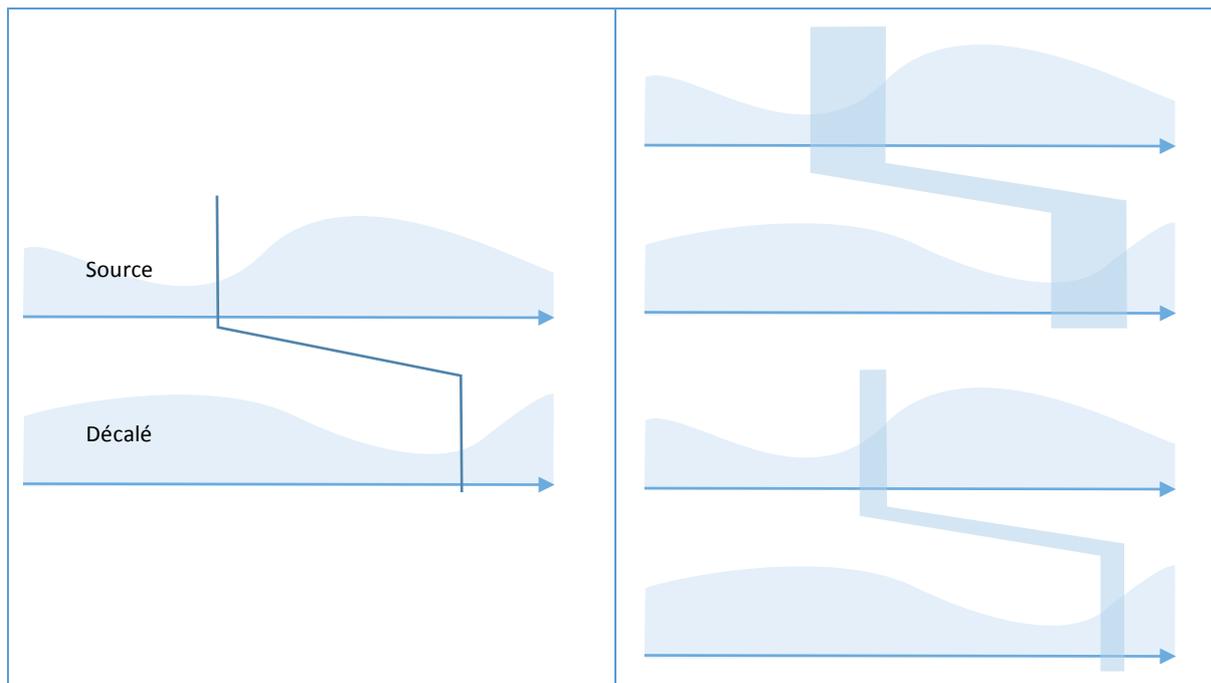


Figure 41 : La modélisation du décalage avec SOFI continu Figure 42 : Evaluations d'un décalage avec SOFI continu

Le chapitre 4.13.3 détaille un exemple d'utilisation de cette opération pour estimer le remboursement de TVA des dépenses des opérations.

4.9.3.1.3 L'opération de totaux périodiques « SumBy »

L'opération « SumBy » totalise les valeurs d'une variable par intervalle régulier. Par exemple, cette opération peut être utilisée afin de réaliser une somme de dépenses mois par mois. Cette opération est nécessaire pour la modélisation de SOFI continu, car les dépenses peuvent être placées au jour exact où elles ont lieu alors que dans l'exemple donné dans le chapitre 4.13.3 un remboursement de TVA calculé à partir de dépenses est effectué mois par mois.

L'opération « SumBy » prend deux paramètres. Le premier paramètre est une date qui fixe l'origine des sections à concentrer. Le deuxième paramètre correspond à la taille de ces sections. Le total est toujours placé au début de chaque section.

```
SumBy(date d'origine : valeur scalaire, durée : valeur scalaire, : valeur de type « Measure »)
: valeur de type « Measure »
```

Figure 43 : La signature de l'opération « SumBy »

Sur la figure ci-dessous, les trois dépenses qui ont lieu le premier mois sont additionnées et la somme est placée au début de ce mois.

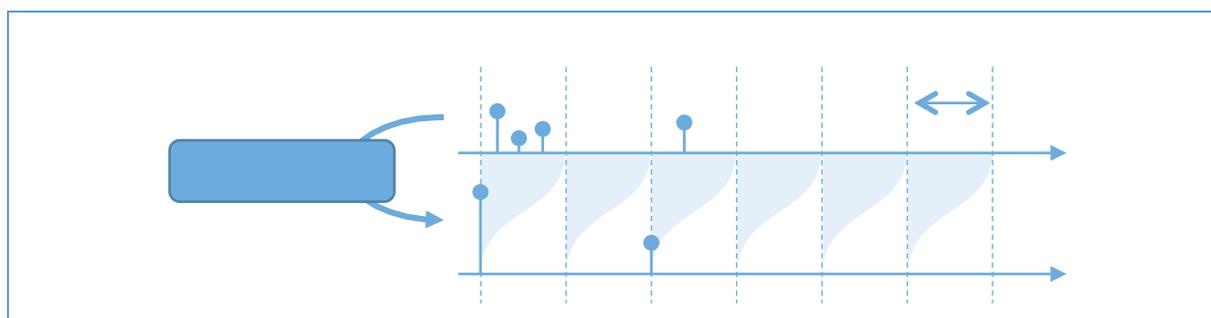


Figure 44 : Un exemple d'application de l'opération « SumBy »

Dans la formule ci-dessous, l'origine est fixée au 1 Janvier 2015 et la taille des sections est définie à 1 mois.

La formule réalise des sommes du 1^{er} du mois au 1^{er} du mois suivant à minuit. Le résultat de chaque somme est placé le 1^{er} de chaque mois à minuit. En fixant l'origine au 4 Janvier 2015, les sommes seraient réalisées du 4^{ème} du mois au 4^{ème} du mois suivant à minuit et le résultat serait placé le 4^{ème} du mois.

```
"Concentrée"(t) :=
SumBy
(
  '2015-01-01',
  Month(1),
  "Source"(t)
)
```

Figure 45 : Une formule de concentration avec l'opération « SumBy »

Le chapitre 4.13.3 détaille un exemple d'utilisation de cette opération pour estimer le remboursement de TVA des dépenses des opérations.

4.9.3.1.4 L'opération de total local « SumOver »

L'opérateur « SumOver » calcule la valeur d'une variable de type « Mesure » pour un intervalle de temps et place un Dirac dont le poids correspond au résultat. Ceci permet de réaliser un total local.

Les deux premiers paramètres de l'opération « SumOver » définissent les bornes de l'intervalle sur lequel est réalisée la somme. La troisième permet de choisir la position du Dirac dont le poids correspond au résultat de cette somme. Le dernier paramètre est la mesure à partir de laquelle la somme est réalisée.

```
SumOver(début : scalaire, fin : scalaire, position : scalaire, : valeur de type « Measure »,
: valeur de type « Measure »
```

Figure 46 : La signature de l'opération « SumOver »

Par exemple, dans le modèle du cas d'étude, l'ensemble des dépenses de construction de la nouvelle cantine est amorti à partir de la fin des travaux. Les dépenses doivent être rassemblées afin que leur amortissement commence au même instant. La variable « Montant à amortir » est calculée à partir de la somme des dépenses des opérations entre le début et la fin du scénario grâce à cette opération.

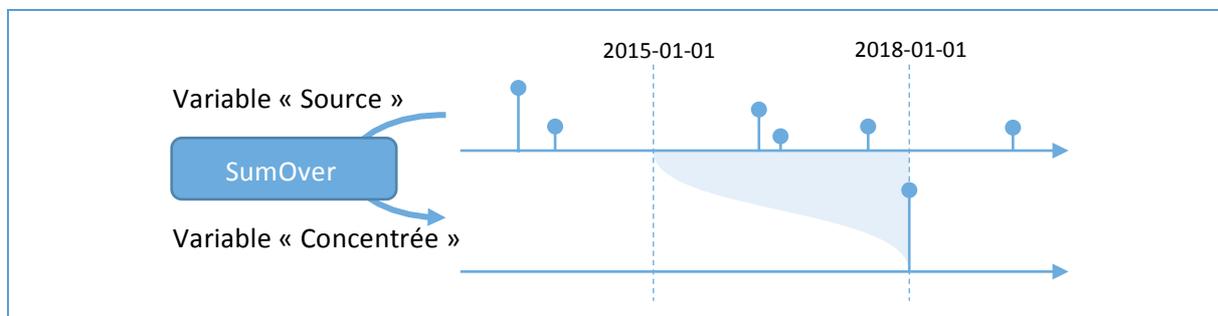


Figure 47 : Un exemple d'application de l'opération « SumOver »

La formule ci-dessous réalise la somme du 1^{er} Janvier 2015 au 1^{er} Janvier 2018 et place le résultat le 1^{er} Janvier 2018.

```
"Concentrée"(t) :=
```

```

SumBy
(
  '2015-01-01',
  '2018-01-01',
  '2018-01-01',
  "Source"(t)
)

```

Figure 48 : Une formule de concentration avec l'opération « SumOver »

Le chapitre 5.2.1.8 détaille un exemple d'utilisation de cette opération pour calculer le total des dépenses des opérations, qui est nécessaire au calcul de l'amortissement.

4.9.3.1.5 L'opération d'accumulation « Accumulate »

L'opération « Accumulate » accumule la valeur d'une variable continue de type « Mesure ». Ceci permet suivre l'évolution de la valeur d'une mesure évaluée entre un instant d'origine et un instant choisi.

Les deux premiers paramètres de l'opération « Accumulate » définissent une contrainte pour initialiser l'accumulation à une certaine valeur pour un instant donné. Le troisième paramètre est la mesure à accumuler.

```

Accumulate(date d'origine : valeur scalaire, valeur d'origine : valeur scalaire,
: valeur de type « Measure ») : valeur de type « Field »

```

Figure 49 : La signature de l'opération « Accumulate »

Par exemple, dans le modèle du cas d'étude, la variable « Participation de la mairie à l'équilibre » est accumulée afin d'observer l'évolution de la participation totale à partir de 2015.

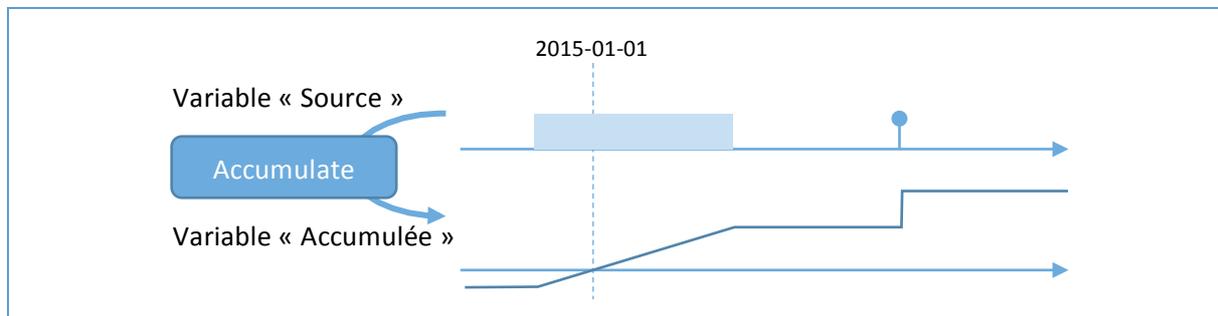


Figure 50 : Un exemple d'application de l'opération « Accumulate »

La formule ci-dessous réalise l'accumulation de la variable « Source » avec comme contrainte que la valeur du résultat soit égal à zéro le 1^{er} Janvier 2015.

```

"Accumulée"(t) :=
Accumulate
(
  '2015-01-01',
  0,
  "Source"(t)
)

```

Figure 51 : Une formule de concentration avec l'opération « Accumulate »

Le chapitre 5.2.1.6.1 détaille un exemple d'utilisation de cette opération pour calculer la participation totale de la mairie jusqu'à un instant donné.

4.9.3.1.6 L'opération de répétition « Repeat »

La fonction « Repeat » permet de répéter les valeurs d'un intervalle en boucle. Cette opération peut être utilisée afin de représenter une situation cyclique. C'est le cas dans l'exemple de la vente des tickets exposé dans le chapitre 4.13.4.

```
Repeat(date d'origine : valeur scalaire, taille de l'intervalle : valeur scalaire,  
: valeur continue) : valeur continue
```

Figure 52 : La signature de l'opération « Repeat »

La formule ci-dessous réplique les valeurs de la variable « Source » pour l'année 2015 sur toutes les autres années. Le premier paramètre est le début de l'intervalle à répéter (2015), le deuxième paramètre est la taille de cet intervalle (1 an) et le troisième paramètre est la variable contenant les valeurs à répéter (la variable source « Source »).

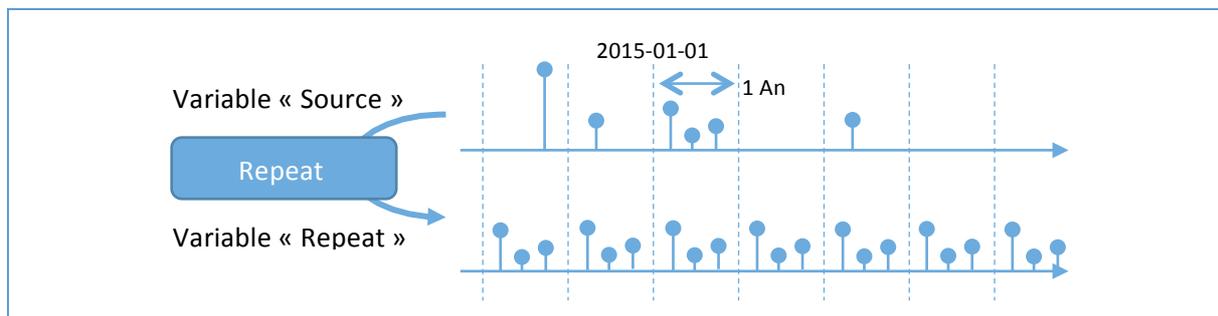


Figure 53 : Un exemple d'application de l'opération « Repeat »

```
"Répétée"(t) :=  
Repeat  
(  
  '2015-01-01',  
  Year(1),  
  "Source"(t)  
)
```

Figure 54 : Une formule de concentration avec l'opération « Repeat »

Le chapitre 4.13.4 détaille un exemple d'utilisation de cette opération pour estimer les futurs achats de tickets de cantine.

4.9.3.1.7 L'opération de valeur exponentielle « Exp »

Les fonctions exponentielles permettent de modéliser des concepts qui sont modélisés par les suites géométriques dans SOFI discret.

Afin de faciliter l'écriture, la fonction « Exp » prend trois paramètres qui permettent de construire une fonction exponentielle avec des contraintes de temps. Le premier paramètre permet de définir la date à laquelle la fonction est égale à 1. Un facteur permet éventuellement de définir une valeur initiale à cette date. Le deuxième et le troisième paramètre permet de définir l'équivalent de la raison pour une suite géométrique.

L'évaluation de la formule ci-dessous donne 1000 pour le 1^{er} Janvier à minuit de l'année du présent. Le rapport des valeurs de deux points espacés de un an vaut est de « 1.02 ».

```
"Exponentielle"(t) := 1000 * Exp(BeginningOf(Year(1), "now"()), 1.02, Year(1))
```

Figure 55 : La formule de la variable « Remboursement de la TVA » en modélisation continue

La fonction « Exp » est définie comme ceci :

$$Exp(t) = e^{\frac{\log(\text{raison})}{\text{période}}(t-\text{début})}$$

L'utilisation de cette fonction comme équivalent aux suites géométriques n'est pas évidente. Elle doit être expliquée avec un exemple détaillé. C'est ce que propose le chapitre 4.13.5 qui détaille la démarche sur la prise en compte de l'inflation pour l'estimation du coût des denrées d'un repas.

4.9.3.2 Les opérateurs de dates

Les dates et les durées utilisées comme paramètres des opérateurs présentés précédemment sont définies avec des expressions de calcul de dates. Ces expressions utilisent les variables scalaires qui peuvent représenter des dates ou des durées. Par exemple, la formule ci-dessous permet d'obtenir le début de l'année du présent. Le chapitre 4.10 explique l'utilité de ce genre de formules.

```
BeginningOf(Year(1), "now"())
```

Figure 56 : La formule du début de l'année du présent

Le calcul des dates peut faire intervenir les opérateurs, par exemple pour calculer la durée entre deux dates.

Les unités sont définies en utilisant des fonctions, Par exemple, la fonction « Year(n) » représente une année. L'utilisation des unités est présentée dans le chapitre 4.8.

4.9.3.3 La formalisation des formules

Les formules sont exprimées sous forme d'une expression alliant des calculs de fonctions continues et des calculs de réels. Ces formules de calcul sont interprétées en trois étapes. La première étape est une analyse syntaxique. L'arborescence de la formule est construite sans vérifier la cohérence de cette arborescence.

```
expression          = parenthesisExpression | temporalTermExpression
                    | continuousVariableExpression | scalarVariableExpression
                    | operationExpression | binaryExpression | number | date

parenthesisExpression = "(" expression ")"

continuousVariableExpression = variableName , "(t)"
scalarVariableExpression    = variableName , "()"

operationExpression = operatorName , "(" argumentListExpression ")"
argumentListExpression = expression "," (argumentListExpression | expression)

binaryExpression      = expression binaryOperation expression
binaryOperation       = "|" | "&" | "=" | "#" | "<" | ">" | "<=" | ">="
                    | "+" | "-" | "*" | "/" | "^"

temporalTermExpression = "t";
```

Figure 57 : La grammaire des formules de calcul des variables continues

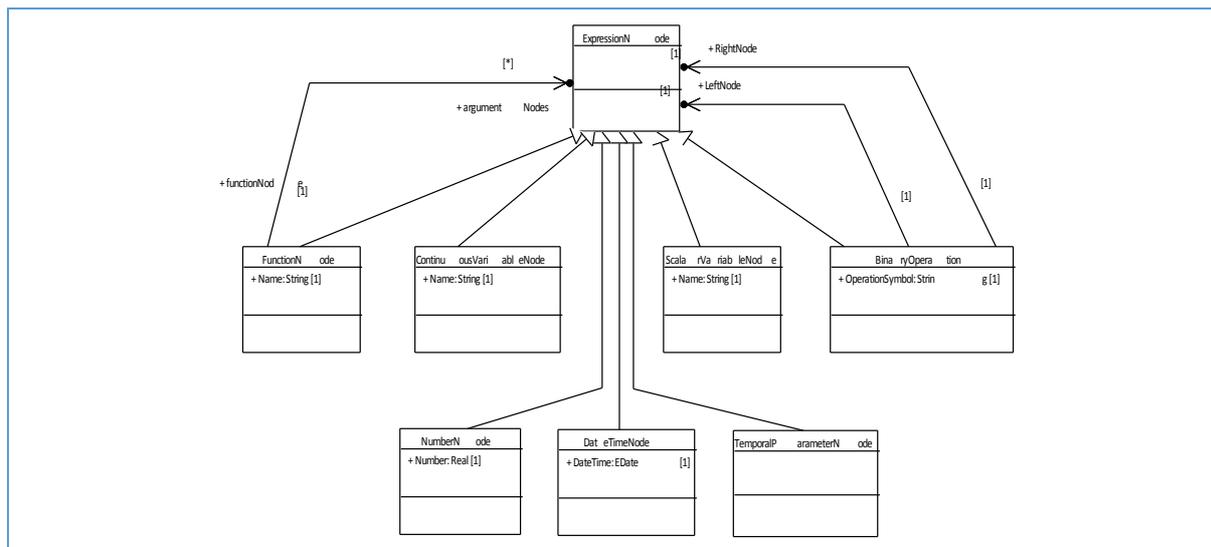


Figure 58 : Le diagramme de classes de l'arborescence syntaxique des formules de calcul

Le langage des formules de calcul des variables scalaires ne permet pas d'utiliser les variables continues. Sa grammaire est identique à celle des formules de calcul des variables continues, mis à part que l'expression « continuousVariableExpression » n'y est pas présente.

```

expression          = parenthesisExpression | temporalTermExpression
                    | scalarVariableExpression
                    | operationExpression | binaryExpression | number | date

parenthesisExpression = "(" expression ")"

scalarVariableExpression = variableName , "("

operationExpression = operatorName , "(" argumentListExpression ")"
argumentListExpression = expression "," (argumentListExpression | expression)

binaryExpression = expression binaryOperation expression
binaryOperation = "|" | "&" | "=" | "#" | "<" | ">" | "<=" | ">="
                | "+" | "-" | "*" | "/" | "^"

temporalTermExpression = "t";

```

Figure 59 : La grammaire des formules de calcul des variables scalaires

La deuxième étape est une analyse sémantique. L'arborescence créée lors de l'analyse syntaxique est parcourue afin de créer une nouvelle arborescence qui respecte un ensemble de règles : le type des paramètres des appels de fonctions et celui des opérandes des opérateurs sont vérifiés ; l'existence des variables utilisées dans la formule est vérifiée. Le même modèle d'arborescence est utilisé par les formules de calcul des variables continues et celles des variables scalaires. En effet, comme ce modèle prend en compte le type des nœuds, la formule d'une variable continue aura une racine de type « IContinuousFormulaNode » et celle d'une variable scalaire sera de type « IScalarFormulaNode ».

La troisième étape interprète l'arborescence sémantique pour créer un arbre de calcul.

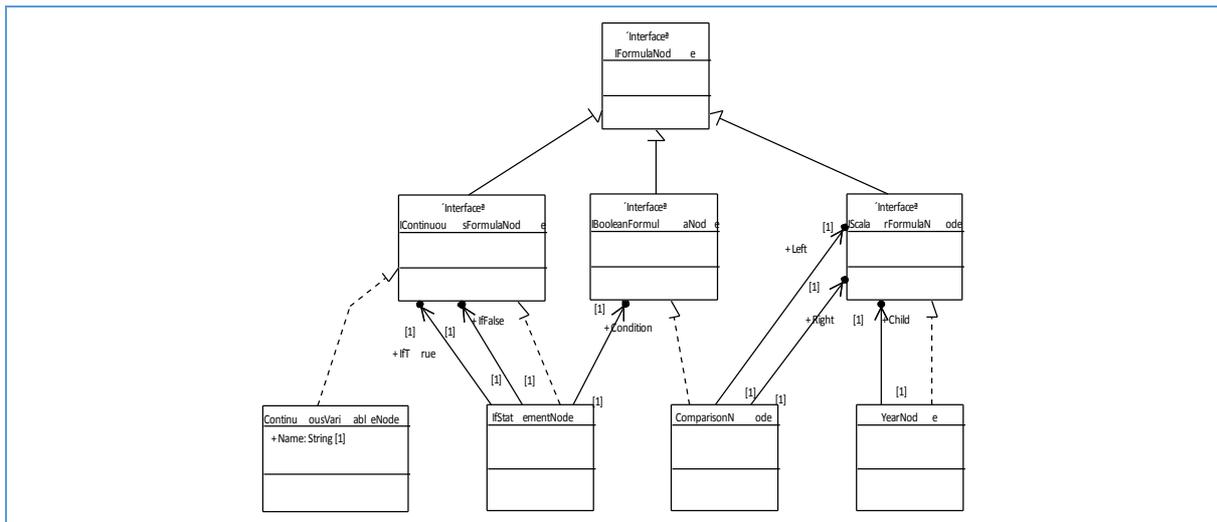


Figure 60 : Le diagramme de classes de l'arborescence sémantique des formules de calcul

L'étape d'analyse syntaxique et celle d'analyse sémantique peuvent générer des exceptions contrairement à l'étape d'interprétation qui ne peut pas échouer. En effet, l'arborescence créée par l'analyseur sémantique est toujours valide puisque c'est le rôle des deux premières étapes de vérifier les éventuelles erreurs. Ceci permet de vérifier la validité d'un modèle sans avoir à passer par l'interprétation de celui-ci.

4.9.4 La comparaison des solutions

Dans le cas de SOFI discret, lors de la modélisation des équations, l'expression du temps est forcée à une périodicité fixe ce qui oblige à déformer la logique à modéliser. Alors que, dans le cas de SOFI discret, les équations permettent de retranscrire les notions de temps en utilisant les unités adéquates.

4.9.5 Les évolutions possibles

La modélisation du temps de façon continue nécessite de prendre en compte de nombreuses nuances. Il est fort probable que d'autres opérateurs soient nécessaires dans certains cas qui n'ont pas encore été rencontrés. Cependant, les opérations définies dans ce chapitre permettent de montrer la logique appliquée afin de résoudre les besoins définis dans les chapitres suivants.

4.10 La délimitation du passé et du futur

4.10.1 Le besoin

Le passé est évalué en fonction de valeurs connues alors que le futur est estimé en fonction du passé et des décisions prises concernant le futur souhaité. Le passé et le futur sont deux cas qui nécessitent une modélisation particulière.

Un suivi est réalisé au fur et à mesure que le présent avance. Les estimations établies pour le futur sont remplacées par la saisie de ce qui est réellement arrivé. Le suivi est réalisé périodiquement sur des périodes assez courtes afin de gérer au plus tôt les imprévus.

A court terme, le futur est donc estimé sur des périodes fines alors que des périodes plus larges permettent d'avoir une vision plus globale à moyen terme.

4.10.2 La solution avec SOFI discret

La modélisation de SOFI discret représente le temps par une succession de périodes de longueur identique. SOFI discret permet d'utiliser deux échelles de temps : une échelle avec des périodes annuelles et une autre avec des périodes mensuelles. Les périodes plus fines ne sont pas exploitables

à causes d'un problème de performance détaillé dans le chapitre 4.6. Le présent est positionné sur une période de chacune de ces échelles. La figure ci-dessous montre les deux échelles de temps avec leur période du présent : année du présent et mois du présent. Ainsi, la notion de présent n'est pas un instant comme dans la réalité, mais une période de temps.

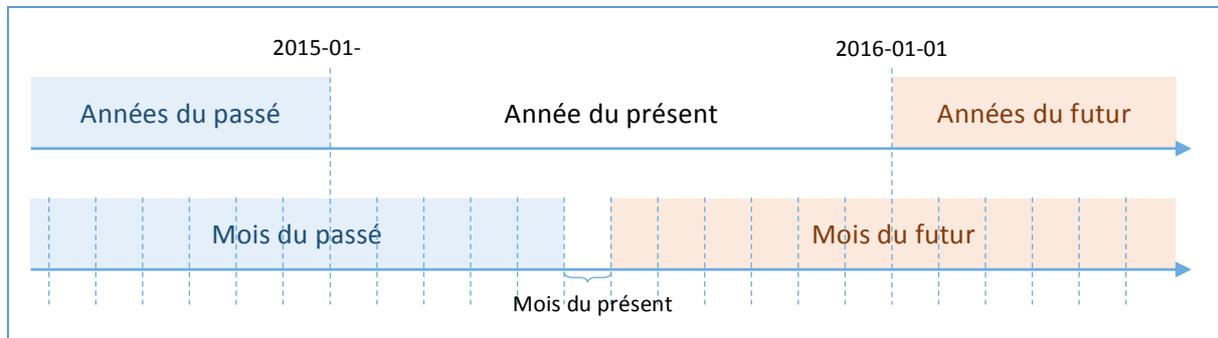


Figure 61 : La représentation du présent par la modélisation de SOFI discret

Ceci est dû à l'utilisation de deux échelles de temps de granularité différente. Sur une échelle mensuelle, le présent pourrait être positionné au début de chaque mois. Sur la figure ci-dessous, le présent est positionné au début du mois de Mai 2015. Les périodes mensuelles précédant la date du présent appartiennent au passé et celles qui suivent au futur.

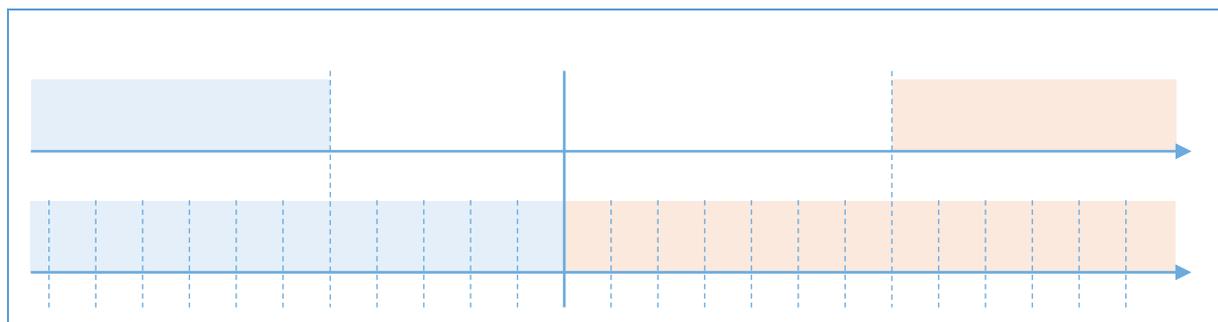


Figure 62 : Une date qui sépare les périodes mensuelles ne permet pas nécessairement de séparer les périodes annuelles

Le problème apparaît lorsque les deux échelles sont utilisées ensemble. En effet, la période de l'année 2015 de l'échelle annuelle est à cheval entre le passé et le futur. Comme le passé et le futur sont calculés différemment et qu'un seul calcul est possible par période, la période de l'année du présent est un cas particulier qui nécessite une formule de calcul différente de celle du passé et du futur. Afin de conserver une cohérence entre l'échelle annuelle et mensuelle, l'échelle mensuelle représente aussi le présent comme une période et non pas une date.

L'estimation du futur est réalisée sur de nombreuses années. L'estimation mois par mois pour les années suivantes n'est pas nécessairement intéressante. Afin de limiter les calculs inutiles, l'échelle annuelle est dans ce cas choisie pour estimer le futur. Néanmoins, afin d'assurer un suivi au cours de l'année actuelle, il est nécessaire de connaître les estimations mois par mois pour l'année du présent. Ainsi, les estimations de l'année présente seront comparées chaque mois au fur et à mesure de l'avancement du présent.

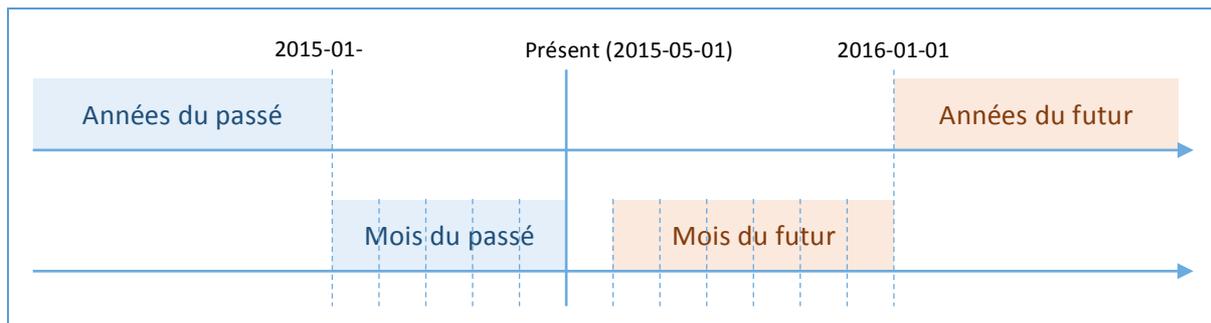


Figure 63 : Le calcul de l'année du présent avec une périodicité mensuelle

Dans cette situation, une estimation est réalisée pour les mois futurs de l'année du présent ainsi qu'une autre pour les années futures. Ces deux estimations nécessitent une formule de calcul particulière, car les formules de calculs sont écrites différemment pour une échelle mensuelle ou annuelle.

4.10.3 La solution avec SOFI continu

En modélisation continu, le présent est placé à un instant précis. Néanmoins, en fonction de ce qui est modélisé, il est possible de délimiter des zones sur des années, mois, semaines ou jour.

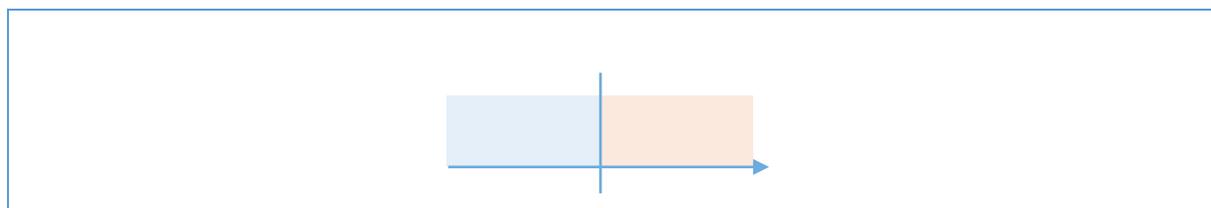


Figure 64 : La représentation du présent par la modélisation de SOFI continu

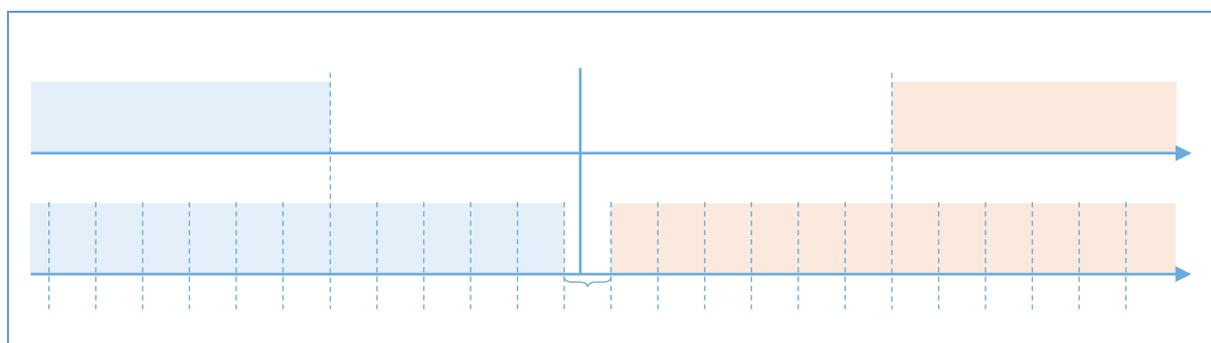


Figure 65 : Le découpage du temps par année ou par mois

L'opération « BeginningOf » permet de déterminer le début de la période contenant une date donnée. Ainsi, il est possible d'en déduire l'année, le mois, la semaine et le jour auquel appartient le présent. Par exemple, la formule ci-dessous permet d'obtenir le début de l'année du présent.

```
BeginningOf(Year(1), "now"())
```

Figure 66 : La formule du début de l'année du présent

Ces expressions sont utilisées pour définir des frontières entre deux portions ayant un calcul différent. Par exemple, dans la formule ci-dessous la variable « Variable A » est nulle jusqu'au début de l'année du présent puis est égale à la variable « Variable B ».

```

"Variable A"(t) :=
If
(
  t < BeginningOf(Year(1), "now"()),
  0,
  "Variable B"(t)
)

```

Figure 67 : La formule du début de l'année du présent

Ainsi, il est possible de décrire des dates relatives aussi bien au jour présent qu'au mois présent, à l'année présente ou à toute autre période. La période adéquate pour modéliser la réalité est choisie lors de l'écriture des formules de calcul. Par exemple, la périodicité naturelle de la vente de tickets est annuelle. En effet, chaque année les ventes de tickets sont similaires. Ainsi, pour se caler sur cette périodicité naturelle, les frontières sont placées en début d'année de façon à réaliser l'estimation année par année.

Le fait qu'une périodicité annuelle soit choisie pour modéliser les estimations n'empêche toutefois pas d'évaluer les ventes de tickets mois par mois ou semaine par semaine. En effet, le découpage réalisé pour définir les calculs n'a pas d'influence sur le choix de la périodicité d'évaluation.

4.10.4 La comparaison des solutions

Dans la modélisation de SOFI discret, le présent est un intervalle qui n'appartient ni totalement au passé, ni totalement au futur. Afin de réaliser un suivi au cours de l'année, les estimations pour l'année du présent sont réalisées par mois, même si l'estimation des années suivantes est réalisée par années pour éviter des calculs inutiles. Comme l'écriture des formules dépend de la périodicité dans laquelle elles sont exprimées, ceci implique d'écrire une formule différente pour les estimations mensuelles à court terme et les estimations annuelles à moyen terme.

Dans le cas de SOFI continu, ce problème ne se pose pas car le choix des périodes d'évaluations est indépendant de la modélisation. Ainsi, il est possible d'évaluer mois par mois pour l'année courante et année par année pour les années suivantes, sans que cela nécessite une gestion particulière dans le modèle. En effet, la même formule est évaluée dans les deux cas. Ceci réduit le risque d'erreur lors de l'écriture des formules. De plus, lors de la modification d'une formule, il n'y a pas de risque d'oublier de modifier son pendant pour l'autre périodicité.

4.10.5 Les évolutions possibles

Les besoins initialement décrits ont été remplis. Aucune évolution n'est envisagée actuellement.

4.11 L'évaluation des découpages temporels des calculs

4.11.1 Le besoin

Le chapitre précédent détaille la nécessité de découper le temps en plusieurs intervalles afin de définir des calculs différents pour le passé et le futur. D'autres raisons peuvent amener à définir des calculs différents sur certains intervalles. Par exemple, dans le scénario de la construction d'une nouvelle cantine, le déménagement est prévu à une certaine date. Ce chapitre s'intéresse à la façon dont les évaluations sont réalisées. Afin de s'abstraire des problèmes liés à la modélisation présentés dans le chapitre précédent, il est considéré que les deux solutions s'appuient sur la modélisation de SOFI continu.

Dans le cas d'étude, le déménagement de la cantine a lieu le ## du mois. Le calcul de la participation n'est pas le même du 1^{er} Janvier au ## et pour le restant de l'année. En effet, la nouvelle cantine a une meilleure efficacité énergétique. Les dépenses de fonctionnement ne sont donc pas les mêmes avant et après le déménagement.

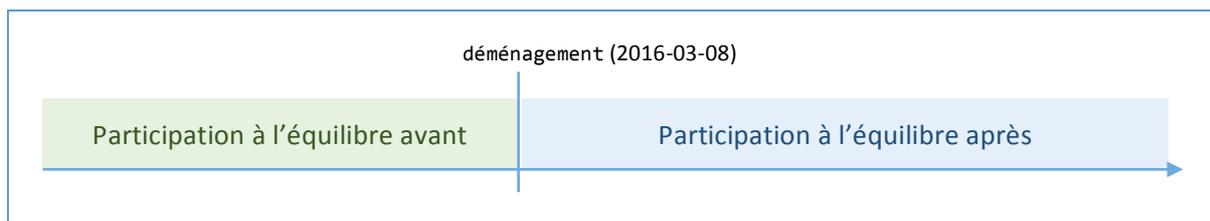


Figure 68 : La représentation graphique de la formule d'un opérateur « If » avec une frontière

Une opération « If » permet de placer un calcul différent sur deux intervalles complémentaires. Cependant les opérateurs « If » peuvent être imbriqués afin de dissocier davantage de cas. Un exemple d'imbrication est donné par la suite pour illustrer l'algorithme de traduction de cette opération en fonction définie par morceaux.

```
"Participation à l'équilibre"(t) := If(t < "déménagement"(), "Participation à l'équilibre avant"(t), "Participation à l'équilibre après"(t))
```

Figure 69 : La formule de la variable « Participation à l'équilibre » qui tient compte du déménagement

4.11.2 La solution avec SOFI discret

Comme cela a été vu dans le chapitre 4.2, le moteur de calcul de SOFI discret repose sur un tableau. Dans le tableau, les calculs sont définis période par période. Par exemple, pour le calcul de la variable « Participation à l'équilibre » la soustraction est répétée pour chaque période. Le calcul référence les deux cellules du dessus.

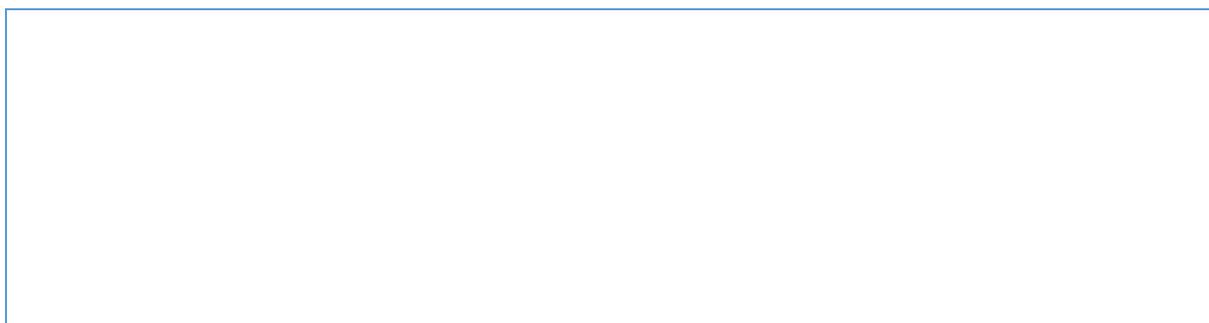


Figure 70 : Le tableau de calcul de la participation en tenant compte du déménagement

Le problème de cette représentation des calculs est qu'il force l'évaluation de toutes les variables du modèle selon la même périodicité. Dans l'exemple précédent, le déménagement de la cantine a lieu au cours d'une année. Comme une colonne du tableau représente une année, le déménagement est pris en compte seulement au début de l'année suivante.

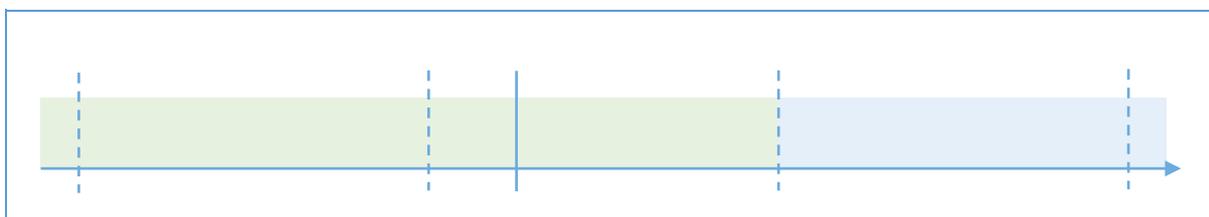


Figure 71 : La représentation graphique de la formule d'une fonction « If » avec une frontière

4.11.3 La solution avec SOFI continu

Les opérateurs « If » sont traduites en fonctions définies par morceaux (l'opérateur « Piecewise »). Les frontières de ces morceaux sont définies par une expression temporelle. Si l'expression contient des variables temporelles, son évaluation varie en fonction de ces variables. Les frontières peuvent

se déplacer et l'ordre des morceaux changer. La bibliothèque LEMF ne permet pas de définir des frontières variables. Les frontières sont donc réévaluées par un algorithme externe à LEMF lorsque celles-ci varient.

Comme cela a été vu dans le chapitre 4.2, le moteur de calcul repose sur une bibliothèque de calcul continu. Les calculs sont organisés sous forme de graphe. Chaque nœud est un opérateur de calcul continu.

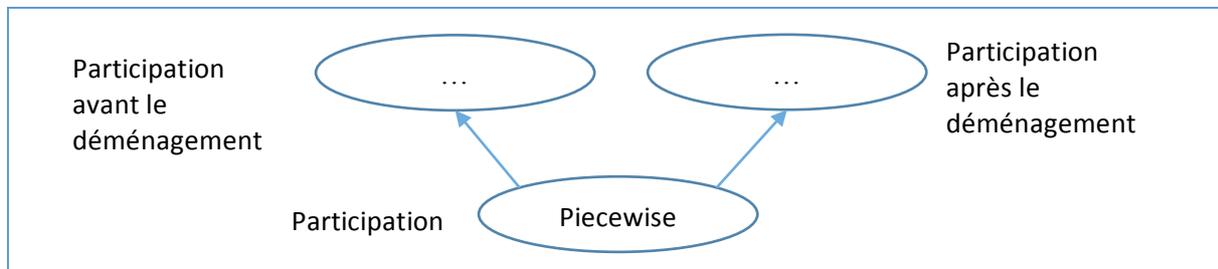


Figure 72 : Le graphe de calcul de la participation qui prend en compte le déménagement

Les variables d'instances correspondent à certains nœuds du graphe. Les nœuds peuvent être évalués pour n'importe quel intervalle de temps. Comme ceci est visible sur la figure ci-dessous, lors de l'évaluation de la participation pour l'année 2016, l'intervalle de temps est séparé en deux intervalles par l'opérateur « Piecewise ». La variable « Participation avant le déménagement » est évaluée jusqu'à l'instant du déménagement et la variable « Participation après déménagement » est évaluée à partir de cet instant. Ces deux évaluations sont ensuite additionnées afin d'obtenir l'évaluation de la participation pour l'année entière.

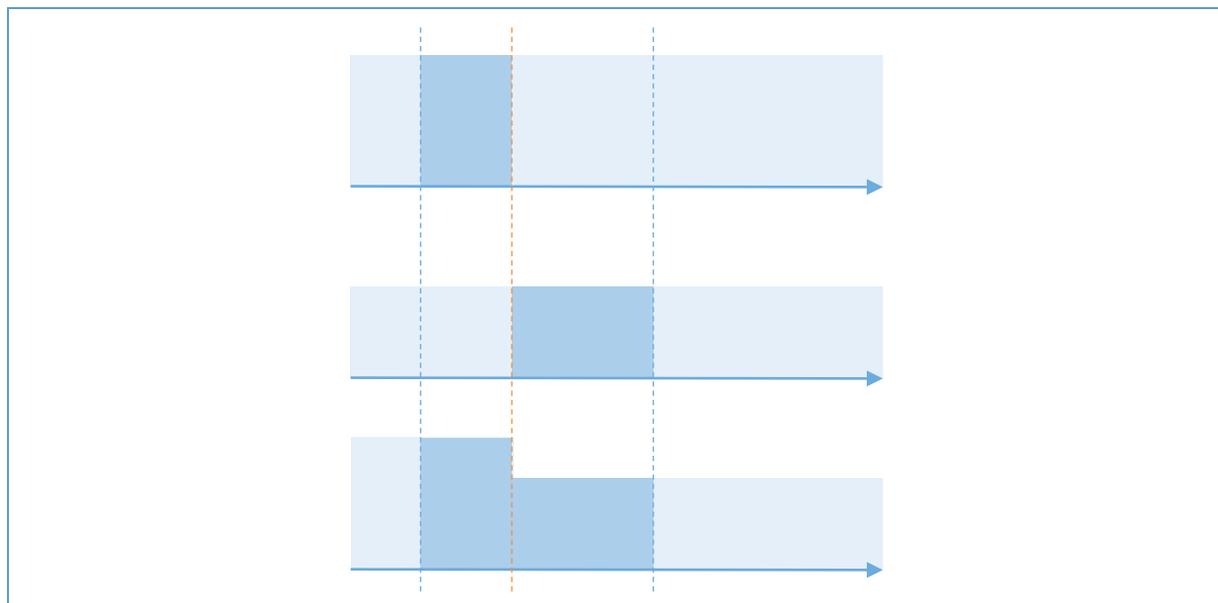


Figure 73 : Intervalles d'évaluation des variables derrière une opération « Piecewise »

L'opérateur peut servir à modéliser un calcul différent seulement sur une zone en définissant un encadrement de la variable « t ». Dans ce cas, bien qu'il n'y ait que deux calculs distincts, trois zones sont délimitées par deux bornes.

```
If(t > '2015-01-01' & t < '2016-01-01', "Variable A"(t), "Variable B"(t))
```

Figure 74 : Une formule d'une fonction « If » avec deux frontières

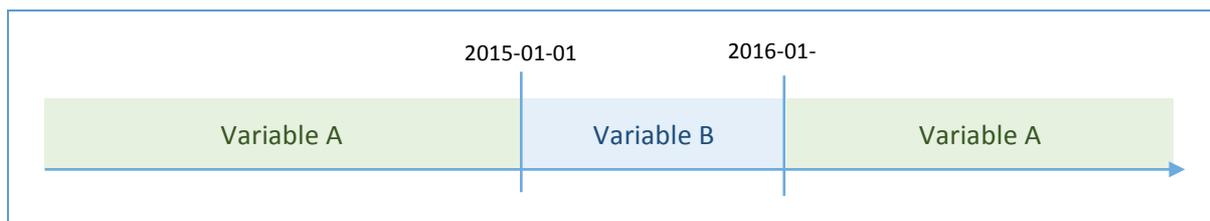


Figure 75 : La représentation graphique de la formule d'une fonction « If » avec deux frontières

Les opérateurs « If » peuvent être imbriqués afin de distinguer plus de deux cas. Les opérateurs « If » forment une arborescence où chaque nœud possède deux branches. Cet arbre représente une superposition de couches qui sont aplaties lorsque la définition par morceaux de LEMF (l'opérateur « Piecewise ») équivalente est traduite.

```
If(t < '2015-01-01', "Variable A"(t), If(t < '2016-01-01', "Variable B"(t), "Variable C"(t)))
```

Figure 76 : Une formule d'une imbrication de fonction « If »

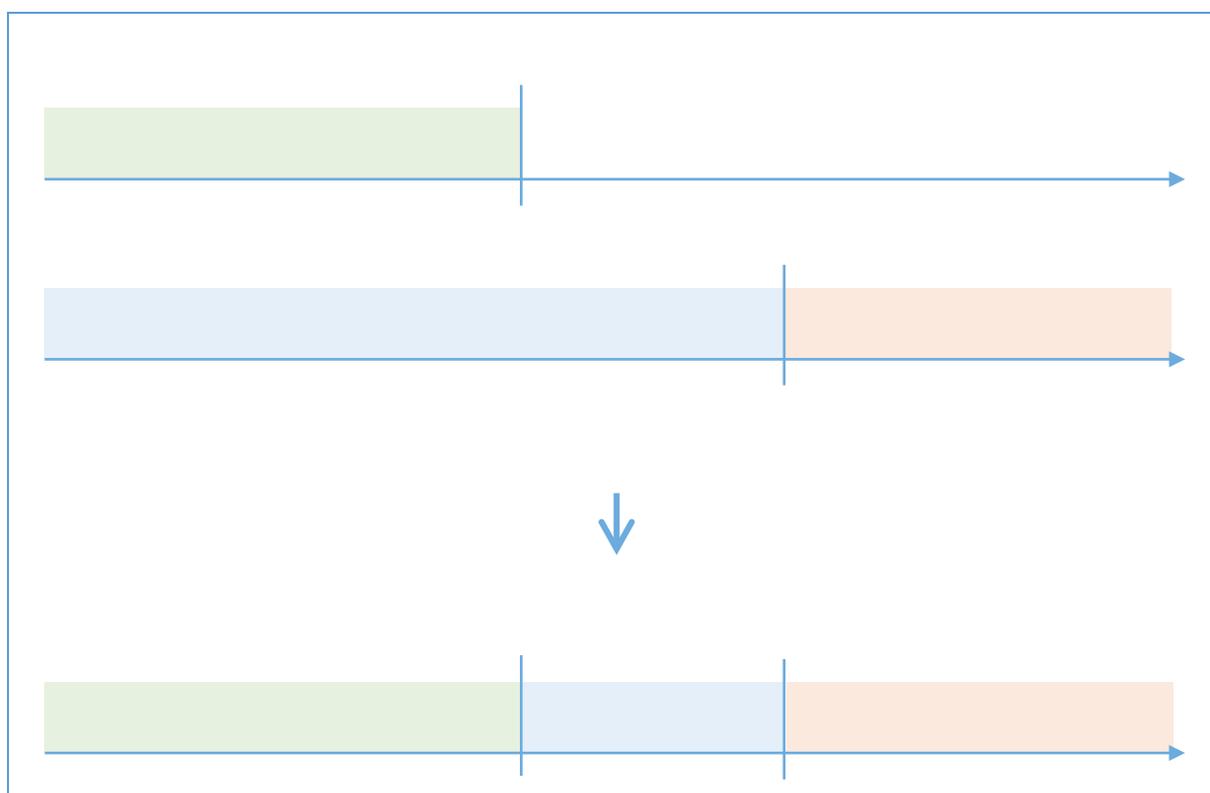


Figure 77 : L'aplatissement d'une imbrication de fonction « If »

4.11.4 La comparaison des solutions

La modélisation de SOFI continu fait abstraction de la périodicité d'évaluation. Ceci permet de choisir n'importe quelle périodicité lors de l'évaluation. En conservant la solution retenue pour le moteur de calcul de SOFI discret, il ne serait pas possible de bénéficier totalement de cet avantage. En effet, comme le moteur de calcul se base sur un tableur, le temps est découpé selon une périodicité. Le fait de réaliser les calculs période par période empêche tout changement au cours d'une période. Ainsi, la modélisation d'un déménagement au cours d'une année n'est pas possible à calculer avec une colonne par année. Une solution serait de passer à une colonne par jour, mais comme cela est décrit dans le chapitre 4.6 ceci impacterait beaucoup les performances.

Le moteur de calcul continu ne subit pas ces limitations grâce à l'utilisation d'un graphe pour représenter les calculs. En effet, comme chaque nœud peut être évalué pour n'importe quel intervalle, l'intervalle d'évaluation original peut se scinder en plusieurs morceaux ou subir d'autres opérations comme un décalage de temps ou une répétition, qui sont décrites dans le chapitre 4.9.

4.11.5 Les évolutions possibles

Les besoins initialement décrits ont été remplis. Aucune évolution n'est envisagée actuellement.

4.12 Les valeurs saisies pour le passé et éventuellement le futur

4.12.1 Le besoin

Comme détaillé dans le chapitre 4.10, le futur est estimé en fonction des valeurs saisies pour le passé. Ainsi, la qualité des estimations du futur dépend de l'efficacité de la modélisation à permettre de représenter la réalité.

Le besoin est de décrire les valeurs que prennent les variables en fonction du temps. Dans l'exemple de la construction de cantine, la vente des tickets est saisie pour les années passées.

4.12.2 La solution avec SOFI discret

Dans le cas de SOFI discret, les données sont saisies en renseignant une valeur pour chaque année ou chaque mois sans préciser comment cette valeur est répartie sur cette période.

4.12.3 La solution avec SOFI continu

Dans le cas de SOFI continu, une variable ne possède pas de périodicité. Une valeur est placée sur une variable en utilisant des dates.

4.12.3.1 Les valeurs ponctuelles

Les valeurs ponctuelles permettent de représenter un événement qui a lieu à un instant donné, comme un achat par exemple. Une valeur ponctuelle est définie avec une date et un montant.

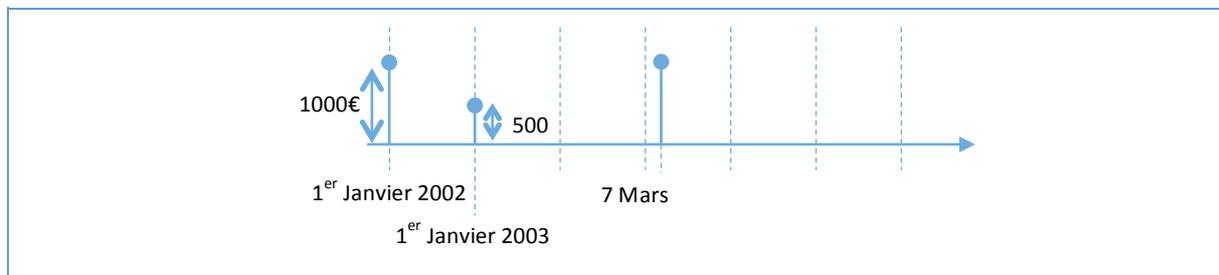


Figure 78 : Valeurs ponctuelles

4.12.3.2 Les valeurs réparties équitablement

Les valeurs réparties équitablement permettent de représenter un événement qui s'étale sur une durée, comme une vente de nombreux exemplaires par exemple. Une valeur répartie équitablement est définie avec un intervalle de temps et un montant à répartir équitablement sur cet intervalle.

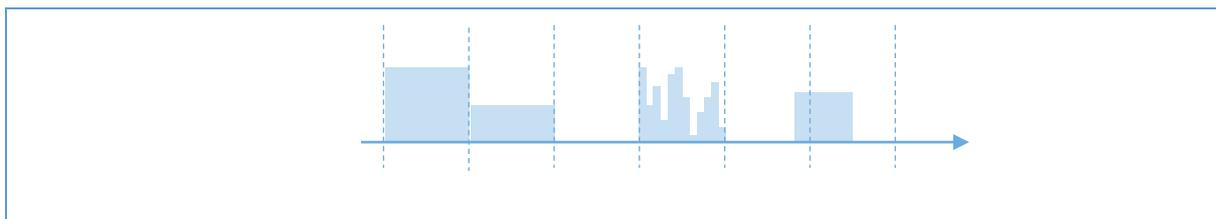


Figure 79 : Valeurs avec une répartition constante

4.12.3.3 Les allures de répartition

Lorsque les données ne sont pas connues avec certitude, saisir de nombreuses valeurs à tâtons peut rendre flou l'intention derrière la saisie. Afin de conserver cette intention, un intervalle est saisi en donnant un montant global et l'allure de la répartition sur cet intervalle. En effet, bien que les valeurs ne soient pas connues, le montant global ainsi que la répartition peuvent parfois être estimés.

La répartition de la valeur est définie par un patron. Deux types de patrons sont implémentés :

- les patrons par valeurs ponctuelles ;
- les patrons par assemblage d'une forme.

Les patrons par valeurs ponctuelles permettent de répartir la valeur saisie à plusieurs instants précis alors que celui par assemblage d'une forme permet de répartir la valeur saisie tout au long d'un intervalle de temps.

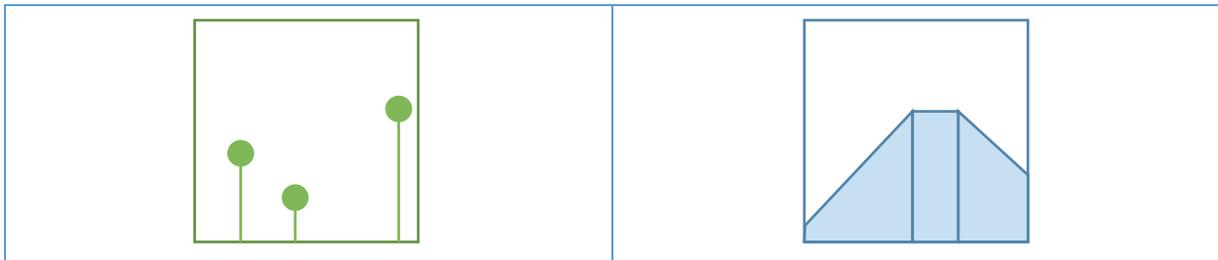


Figure 80 : Un exemple de patron par valeurs ponctuelles

Figure 81 : Un exemple de patron par assemblage d'une forme

Les valeurs saisies sont exprimées avec une valeur, un patron de forme ainsi qu'une contrainte décrivant comment utiliser ce patron. Il existe deux types de contraintes, à savoir « Surface » et « MeanHeight ». La contrainte « Surface » étire le patron de façon à ce que l'aire sous la courbe corresponde à la valeur donnée. La contrainte nommée « MeanHeight » étire le patron de façon à ce que la moyenne des valeurs de la fonction corresponde à la valeur donnée.

Ces deux contraintes sont proposées car chacune d'entre elles est davantage adaptée pour un type de variable continue. La contrainte « Surface » est adaptée aux variables de type « Measure » car c'est l'aire sous la courbe entre deux instants qui est calculée lors de l'évaluation de ce type de variable. La contrainte « MeanHeight » est adaptée aux variables de type « Field » car c'est la hauteur de la courbe à un instant donné qui est calculée lors de l'évaluation de ce type de variable. Le chapitre 4.7 explique la nuance qui existe entre ces deux types de variables continues.

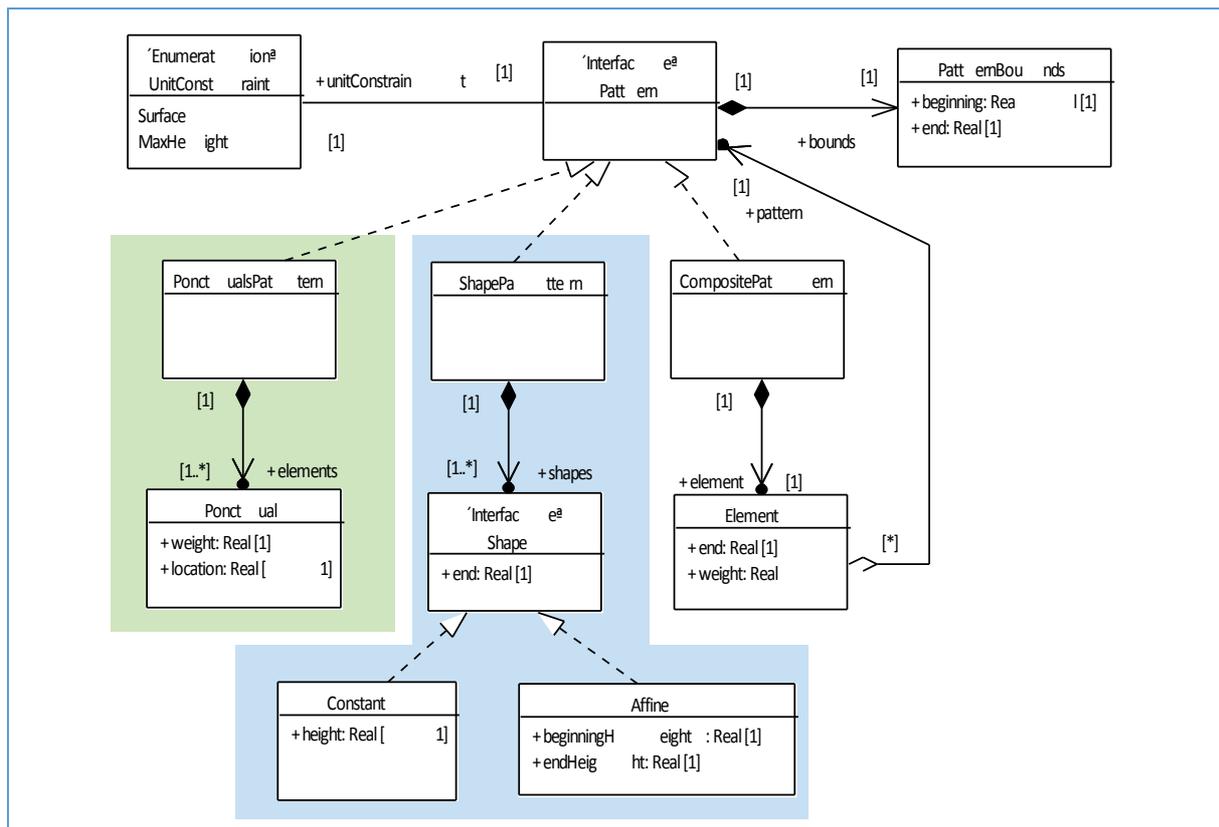


Figure 82 : Le diagramme de classes des patrons de valeur

4.12.4 L'exemple du gain de finesse de la saisie des ventes de tickets

4.12.4.1 La saisie annuelle

Dans un premier temps, les élus considèrent qu'une périodicité annuelle est suffisante afin de planifier le fonctionnement de la cantine. Le gestionnaire de la cantine estime que les tickets sont consommés à peu près équitablement au cours de l'année scolaire et qu'ils sont nécessairement vendus avant d'être consommés. Donc, dans un scénario pessimiste où les tickets sont achetés au plus tard, la vente de tickets peut être considérée comme constante sur l'année scolaire.

4.12.4.1.1 La solution avec SOFI discret

La variable correspondant à la quantité de tickets vendus est définie avec une périodicité annuelle. Une valeur représentant le nombre total de tickets vendus est saisie pour chaque année. Cette variable peut être évaluée uniquement pour des intervalles annuels. Ceci implique que les autres variables du modèle dont la valeur dépend de la quantité de tickets vendus doit adopter une périodicité annuelle également.

4.12.4.1.2 La solution avec SOFI continu

Les valeurs des ventes de tickets sont saisies avec une répartition constante pour chaque année. Bien que cette répartition ne soit pas totalement fidèle à la réalité, elle reste cohérente puisque les tickets sont vendus tout le long de l'année. Lors de l'évaluation du modèle pour un intervalle mensuel ou hebdomadaire, les variables qui dépendent de la quantité de tickets vendus seront imprécises, mais permettront de donner un ordre d'idée de ce que pourrait être la valeur réelle.

4.12.4.2 La saisie intra-annuelle

Les élus ont remarqué que la vente des tickets présente un pic en Septembre puis est plus ou moins constante au cours de l'année. En effet, certaines familles achètent beaucoup de tickets en une fois pour des raisons pratiques et d'autres familles plus régulièrement pour des raisons financières. Ceci

résulte en un surplus de trésorerie qu'ils souhaitent prendre en compte afin de financer certaines dépenses sans recourir aux emprunts.

Le gain en finesse de la saisie des ventes de tickets est réalisé en deux étapes. La première étape consiste à saisir une allure de la répartition des ventes qui semble correspondre à ce qui a été observé lors des années précédentes bien que les chiffres exacts ne soient pas connus. Après avoir tenu les statistiques de vente de tickets, la deuxième étape consiste à saisir la valeur exacte des ventes, idéalement semaine après semaine.

4.12.4.2.1 La solution avec SOFI discret

L'évaluation des modèles de SOFI continu est basée sur le principe des tableurs. Certaines semaines sont à cheval sur deux mois. Il n'est donc pas possible d'additionner les valeurs des semaines pour obtenir la valeur d'un mois comme cela est fait pour le calcul d'une année à partir des mois. C'est pourquoi SOFI discret ne permet pas de définir des variables avec une périodicité hebdomadaire.

La saisie de la quantité de tickets vendus est réalisée sur une périodicité mensuelle selon les comptes existants.

4.12.4.2.2 La solution avec SOFI continu

Lors de la première étape, le gestionnaire saisit une nouvelle répartition des ventes de tickets au cours de l'année davantage proche de la réalité qu'une répartition constante. Il estime qu'environ un quart des ventes sont réalisées en Septembre. Il construit un patron de répartition en trois morceaux constants. Le premier, long d'un mois, représente le pic de vente en Septembre. Le second, long de neuf mois, représente des ventes constantes pour le reste de l'année. Le dernier correspond aux grandes vacances, pendant lesquelles aucune vente n'est réalisée.

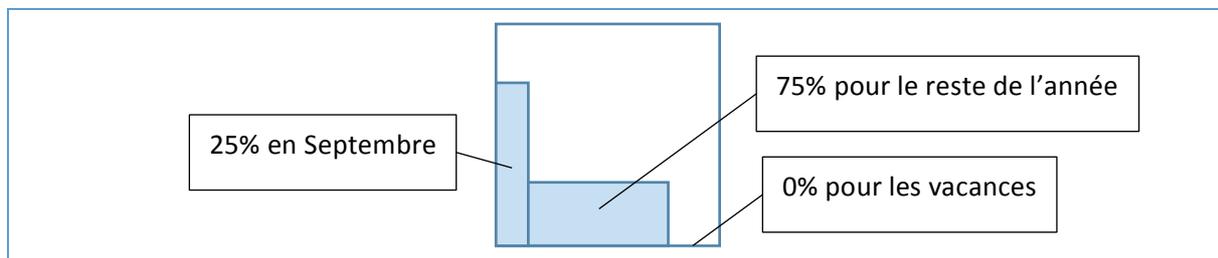


Figure 83 : Le patron de répartition estimée des ventes de tickets au cours d'une année scolaire

La valeur totale de chaque année est déjà saisie. La répartition constante sur la totalité de l'année est simplement remplacée par le patron de répartition qui présente le pic des ventes.

Après avoir tenu les statistiques de vente de tickets, la deuxième étape consiste à saisir la valeur exacte des ventes de chaque semaine. La répartition des ventes au cours d'une semaine est considérée comme négligeable donc une répartition constante est choisie.

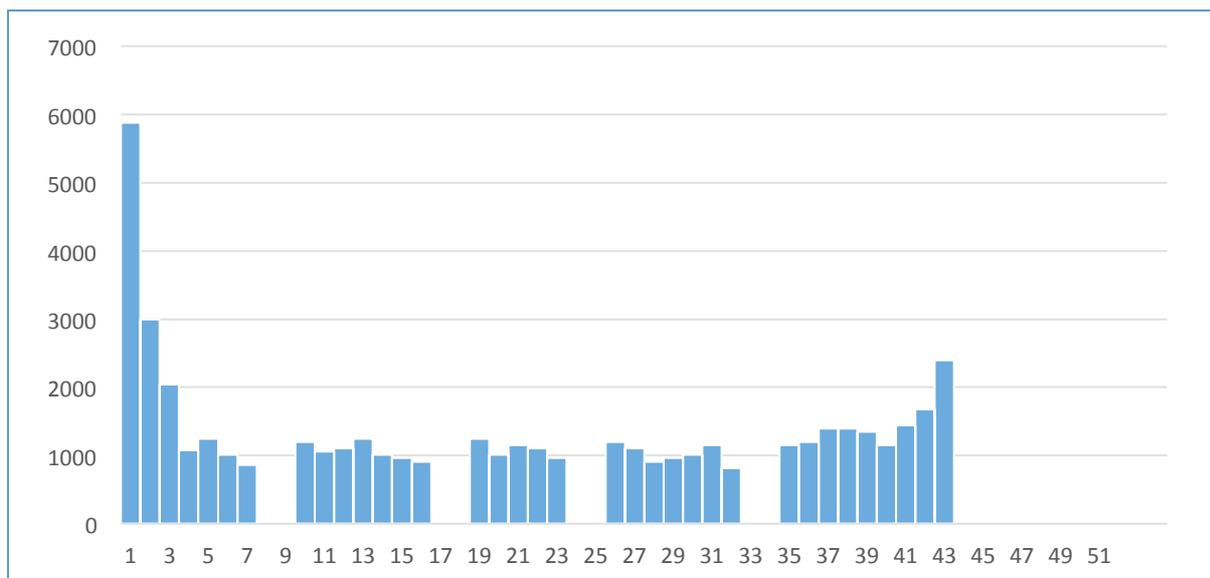


Figure 84 : La saisie semaine par semaine de la répartition réelle des ventes de tickets au cours d'une année scolaire

4.12.5 La comparaison des solutions

La solution retenue pour SOFI discret convient tant qu'une seule périodicité est utilisée pour tout le modèle. En effet, si toutes les valeurs sont définies en années et que les variables sont observées en années également, la solution est satisfaisante. Dans l'exemple de la construction de cantine, un suivi est réalisé semaine par semaine et une planification sur plusieurs années. Plusieurs périodicités cohabitent ; il est nécessaire de connaître la répartition entre chaque période pour réaliser les calculs. En effet, la trésorerie à disposition au cours de l'année ne sera pas la même si l'achat des tickets de cantine est concentré en début d'année ou réparti équitablement sur l'année.

4.12.6 Les évolutions possibles

Le prototype ne permet pas de définir des patrons avec des formes courbes. Les courbes de Bézier pourraient être une solution.

Le contrat des scénarios force à définir un nouveau patron pour chaque valeur. Afin d'alléger le poids des messages, la définition des patrons pourrait être factorisée.

4.13 L'estimation du futur à partir du passé

4.13.1 Le besoin

La modélisation offre aux élus une structure sur laquelle s'appuyer afin d'obtenir une compréhension fine du déroulement des années précédentes. Cette structure permet une exploitation par les logiciels des informations modélisant le passé. En utilisant cette approche, le logiciel SOFI est en mesure de réaliser des estimations du futur à partir de ces informations. Ce genre d'aptitude est commun à tous les logiciels de prospective financière, qu'ils travaillent par projection du passé ou par estimation en fonction du contexte mis à jour et de l'expertise transcrite dans le logiciel.

L'estimation du futur permet aux utilisateurs de juger de l'impact de leurs choix. Ainsi, ils peuvent affiner leurs choix afin d'obtenir la projection dans le futur souhaitée. Lors du suivi, ces estimations permettent de vérifier si la planification concorde avec la réalité afin de détecter les imprévus au plus tôt. Ces imprévus sont pris en compte en reprogrammant la planification.

SOFI continu est une évolution du produit SOFI discret qui a pour objectif d'utiliser le calcul continu afin de pallier les limitations qui seront exposées par la suite.

Deux exemples provenant du cas d'étude présenté au chapitre 2.2 serviront à illustrer une comparaison entre l'approche prise pour SOFI discret et celle pour SOFI continu afin d'estimer le futur. Ce scénario consiste à réaliser la construction d'une cantine en conservant un bilan financier équilibré. Le 1^{er} exemple concerne le calcul du remboursement de la TVA des dépenses des opérations de construction de la cantine. Le 2^{ème} exemple porte sur l'estimation des ventes de tickets de cantine en fonction des deux années passées.

4.13.2 Les approches

4.13.2.1 La solution avec SOFI discret

Une variable est évaluée uniquement sur les périodes prévues par les échelles de temps. Par exemple, avec une échelle mensuelle, une valeur peut être évaluée pour chaque mois.

Lorsqu'une variable est estimée en fonction des valeurs du passé, le calcul de cette variable fait référence à une variable pour une période passée en indiquant le nombre de périodes à décaler. Dans la formule ci-dessous, les valeurs de la variable « Source » sont décalées d'une période.

```
"Décalée"[n] := "Source"[n - 1]
```

Figure 85 : Une formule de référence au passé en modélisation discrète

Dans le cas où ces deux variables sont de périodicité mensuelle, la période de décalage temporel est d'un mois. Lorsque la variable « Décalée » est évaluée pour le mois de Janvier 2015, le résultat sera l'évaluation de la variable « Source » pour le mois de Décembre 2014.

	2014	2015			
	Dec.	Jan.	Fév.	Mar.	Avr.
Source	4	8	16	23	42
Décalée	2	4	8	16	23

Figure 86 : La modélisation du décalage lié à la période d'évaluation

4.13.2.2 La solution avec SOFI continu

Dans le cas de SOFI continu, les informations du passé sont manipulées grâce à une collection d'opérations. Cette collection contient des opérations simples comme l'addition et la multiplication, mais aussi des opérations plus complexes adaptées à la modélisation continue :

- « Delay » permet d'utiliser les valeurs du passé dans le calcul du futur ;
- « SumBy » permet de réaliser des sommes à intervalle régulier ;
- « Repeat » permet de répliquer cycliquement les valeurs d'un intervalle.

Ces trois opérations sont décrites plus en détails dans le chapitre 4.9.

Les opérations sont combinées pour définir comment les informations passées sont utilisées afin de calculer les estimations futures. La chapitre 4.13.3 présente un cas d'utilisation des opérations « Delay » et « SumBy » afin d'estimer un remboursement de TVA, alors que la chapitre 4.13.4

présente un cas d'utilisation des opérations « Delay » et « Repeat » afin d'estimer les ventes des tickets de cantine.

4.13.3 L'exemple du remboursement de la TVA

Lorsqu'un organisme vend un produit ou un service, la part de son chiffre d'affaire correspondant aux taux de la TVA est versée par l'organisme à l'Etat. Comme son nom l'indique, la TVA porte sur la valeur ajoutée ; les consommations intermédiaires doivent donc être déduites du chiffre d'affaire. C'est pourquoi, pour le cas des services assujettis, les organismes demandent à l'Etat un remboursement sur les consommations intermédiaires. Les demandes peuvent avoir lieu annuellement, trimestriellement ou mensuellement en fonction du montant à rembourser.

Le remboursement de TVA s'applique aux dépenses liées aux travaux de construction de la nouvelle cantine. Une demande de remboursement de la TVA est envoyée à l'Etat tous les mois par la municipalité. Le remboursement effectif a lieu le 4^{ème} du 3^{ème} mois suivant.

4.13.3.1 La solution avec SOFI discret

Le décalage entre la variable « Dépense » et la variable « Remboursement de la TVA » devrait être de 3 mois et 4 jours, mais ce décalage doit être arrondi à 3 mois. En effet, la modélisation de SOFI discret décrit les décalages en nombre de périodes. Or, un mois est la période la plus fine autorisée pour des raisons de performances comme justifié dans le chapitre 4.6.

Le calcul de la variable « Remboursement de la TVA » fait référence à la variable « Dépenses » décalée de trois périodes. Comme ces variables ont une périodicité mensuelle, le décalage correspond à trois mois. Ces deux variables doivent obligatoirement être de périodicité mensuelle car le décalage minimal avec une périodicité annuelle serait d'une année.

```
"Remboursement de la TVA"[n] := 0.2 * "Dépenses"[n - 3]
```

Figure 87 : La formule de la variable « Remboursement de la TVA » en modélisation discrète

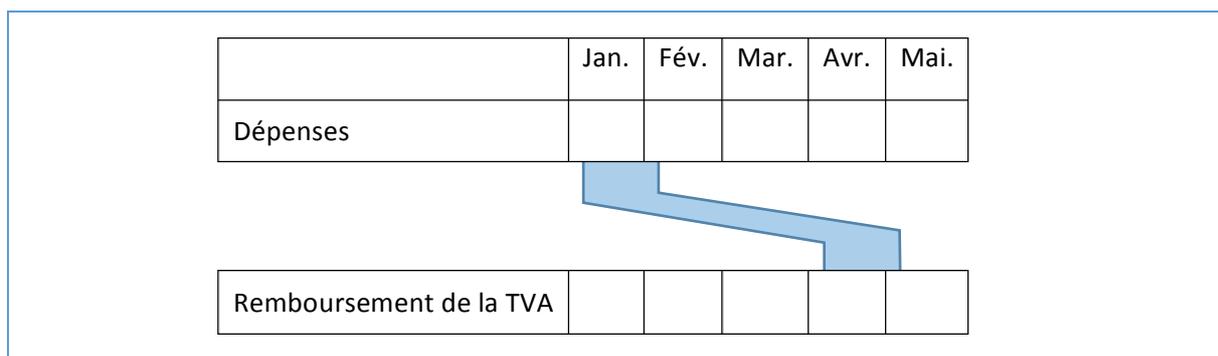


Figure 88 : La modélisation du décalage lié à la période d'évaluation

4.13.3.2 La solution avec SOFI continu

Plusieurs dépenses peuvent avoir lieu au cours d'un mois, mais elles sont remboursées à une date précise. Il est donc nécessaire de réaliser des totaux mois par mois. L'opération « SumBy » est utilisée à cette fin.

Le remboursement est effectué le 4^{ème} jour du 3^{ème} mois suivant. Les totaux sont donc décalés de 3 mois et 4 jours grâce à l'opération « Delay ».

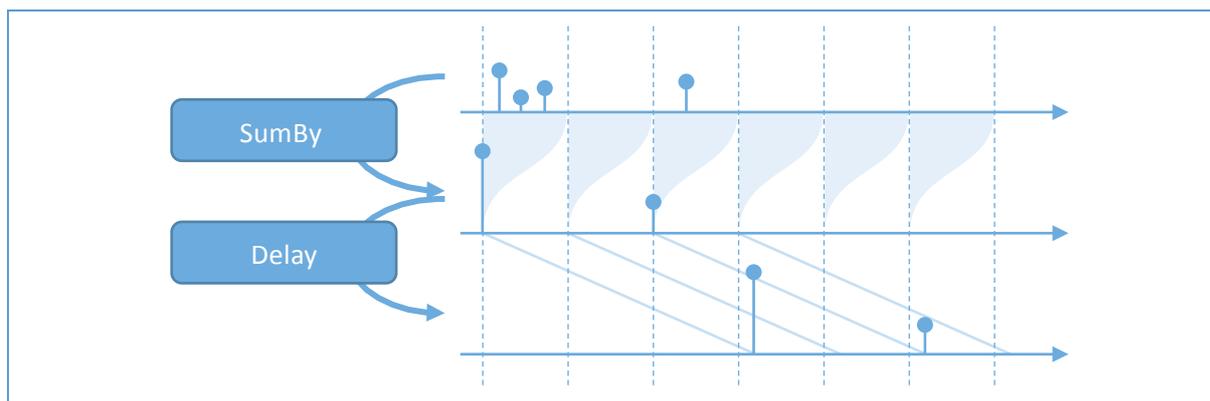


Figure 89 : Les 2 étapes du calcul de la variable « Remboursement de la TVA » en modélisation continue

L'ordre d'enchaînement de ces deux opérations a son importance. Si l'opération de décalage « Delay » était réalisée avant l'opération de concentration « SumBy », les sections de concentration des dépenses auraient lieu 4 jours avant la fin de chaque mois.

Les opérations englobées sont appliquées avant les opérations englobantes. Dans la formule ci-dessous, l'opération « SumBy » est exécutée avant l'opération « Delay ».

```
"Remboursement de la TVA"(t) :=
0.2 *
Delay
(
  Month(3) + Days(4),
  SumBy
  (
    '2015-01-01'
    Month(1),
    "Dépenses"(t)
  ),
)
```

Figure 90 : La formule de la variable « Remboursement de la TVA » en modélisation continue

4.13.3.3 La comparaison des solutions

Dans le cas de SOFI discret, l'estimation de la date de remboursement de TVA est encadrée au mois près. En effet, une seule valeur est estimée par mois pour la variable « Remboursement de TVA ». A la fin de chaque mois, l'estimation peut être comparée avec la réalité afin de détecter des imprévus. Un retard du remboursement prévu le 4^{ème} du mois apparaît donc au début du mois suivant, soit plus d'une vingtaine de jours après l'occurrence de l'imprévu. Ce sont vingt jours où les utilisateurs sont laissés dans le brouillard et voguent à la dérive.

Au contraire, la modélisation de SOFI continu permet de faire un suivi du remboursement jour par jour. En effet, comme la formule de calcul décrit la durée exacte de 3 mois et 4 jours, l'estimation du remboursement est connue au jour près. Chaque jour, le gestionnaire de la cantine peut vérifier si des remboursements sont en retard en comparant les estimations avec les dates effectives.

4.13.4 L'exemple de l'estimation des ventes de tickets

En l'année 2015, les élus souhaitent estimer les ventes de tickets pour les 20 années à venir en fonction des deux dernières années passées. L'estimation est réalisée en faisant une moyenne des ventes des deux dernières années et est considérée identique pour toutes les années suivantes.

4.13.4.1 L'estimation annuelle

Le cas précédemment détaillé dans le chapitre 4.12.4.1 est réutilisé ici.

4.13.4.1.1 La solution avec SOFI discret

Les modèles de SOFI sont découpés en trois périodes : le futur, le présent et le passé. Les raisons de ce découpage sont expliquées dans le chapitre 4.10. Les valeurs de la variable « Participation des familles » sont connues pour le passé. C'est pourquoi aucune formule n'est définie pour le passé.

La figure ci-dessous contient les formules du présent et du futur. Ces formules sont écrites pour une périodicité annuelle. L'indice « n » représente dans ce cas une année. La formule du présent est évaluée pour l'année 2015 et la formule du futur pour les années de 2016 à 2035.

La formule du présent réalise la moyenne entre la valeur connue pour 2014 et celle de 2013. En effet, l'expression « n – 1 » symbolise l'année précédant l'année calculée. Comme 2015 est l'année du présent, « n – 1 » fait référence à 2014 et « n – 2 » à 2013.

La formule du futur se contente de propager la valeur estimée pour l'année du présent. Pour « n » correspondant à l'année 2016, la variable prend la valeur calculée pour 2015. Pour 2017, la variable prend la valeur calculée pour 2016 qui est en fait celle de 2015 etc.

```
Présent :  
"Participation des familles"[n] := ("Participation des familles"[n - 1] + "Participation des  
familles"[n - 2]) / 2  
  
Futur :  
"Participation des familles"[n] := "Participation des familles"[n - 1]
```

Figure 91 : La formule de la variable « Participation des familles » pour une périodicité annuelle en modélisation discrète

4.13.4.1.2 La solution avec SOFI continu

Les années scolaires sont plus appropriées pour réaliser des estimations puisque le nombre d'élèves évolue entre chaque année scolaire. Pour se placer en années scolaires, un présent du 1er Septembre 2014 au 1er Septembre 2015 est défini avec les variables « Début du présent » et « Fin du présent ».

Le schéma ci-dessous montre le détail du calcul (les valeurs sont accentuées pour des questions de lisibilité). Les valeurs de la variable « Participation des familles » sont retardées d'un an et de deux ans de façon à ce que les valeurs de 2014 et 2013 se placent sur l'année 2015. Ainsi, en réalisant la moyenne de ces deux résultats intermédiaires, l'année 2015 équivaut à la moyenne des années 2014 et 2013. L'estimation des ventes pour l'année 2015 est ensuite dupliquée pour les années suivantes.

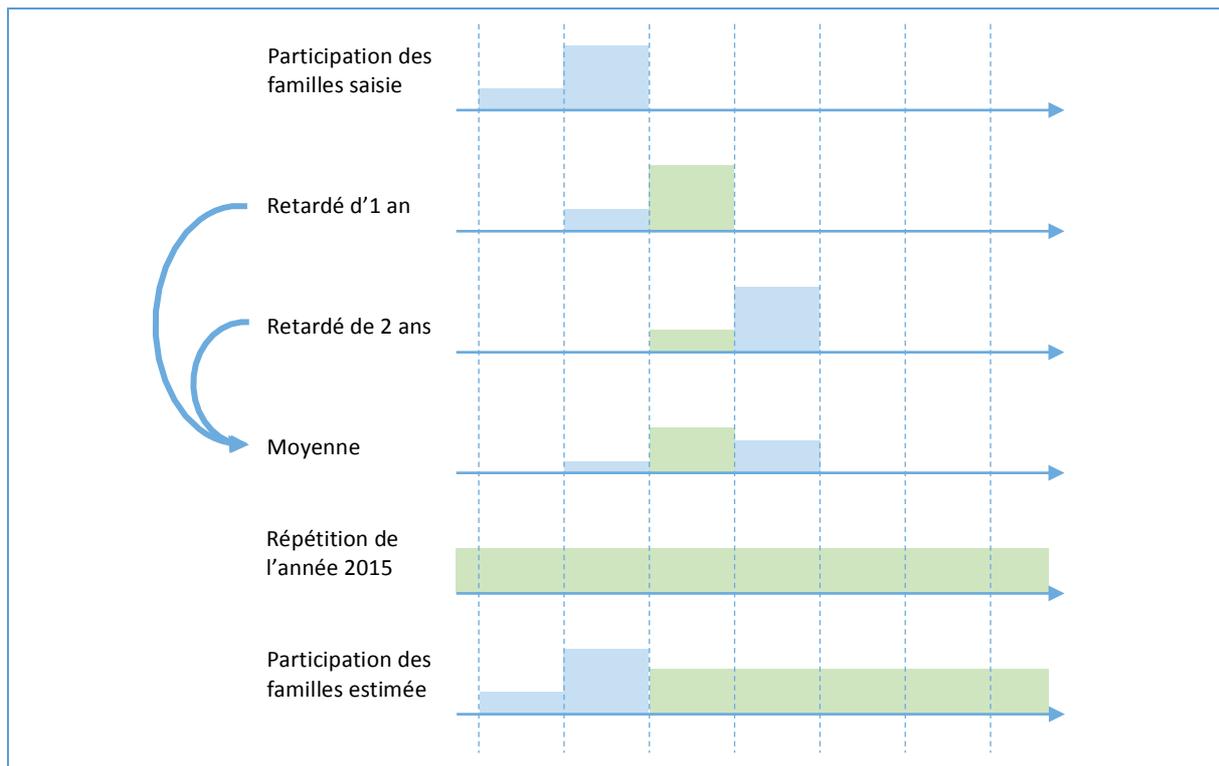


Figure 92 : Les 2 étapes du calcul de la variable « Participation des familles » en modélisation continue

La participation de la mairie retardée de 1 et 2 ans est obtenue avec l'opération « Delay ». Leur moyenne est réalisée avec les opérations arithmétiques d'addition et de division. La moyenne de l'année 2015 est reproduite en cycle sur les autres années avec l'opération « Repeat ». L'estimation réalisée est restreinte aux années du futur grâce à l'opération « If ». Comme les données du passé sont saisies, elles se placent par-dessus les valeurs du calcul.

```
"Quantité de tickets vendus"(t) :=
If(t < BeginningOf(Year(1), "now"(),
  0,
  Repeat
  (
    BeginningOf(Year(1), "now"(),
    Year(1),
    (Delay(Year(2), "Quantité de tickets vendus"(t))
    +Delay(Year(1), "Quantité de tickets vendus"(t))
    ) / 2
  )
)
```

Figure 93 : La formule de la variable « Quantité de tickets vendus » en modélisation continue

4.13.4.1.3 La comparaison des solutions

Lorsque l'observation des résultats se fait année par année, les deux solutions sont équivalentes. Ceci est normal puisque SOFI discret n'a aucun problème pour travailler en année. Il est bon de remarquer que SOFI continu permet de réaliser des modèles équivalents à ceux de SOFI discret puisqu'il est une amélioration de celui-ci.

La modélisation continue possède néanmoins un avantage : la répartition des ventes, bien qu'estimée grossièrement par l'utilisateur, est connue. La variable « Participation des familles » peut être évaluée sur des périodes plus fines que l'année. Les variables nécessitant des évaluations pour ces périodes peuvent utiliser cette variable.

La prochaine partie porte sur l'amélioration de la finesse de saisie de la répartition des ventes au cours de l'année. Il y est montré que l'avantage de la solution continue exposée précédemment facilite grandement cette tâche et la rend même naturelle.

4.13.4.2 L'estimation intra-annuelle

Le cas précédemment détaillé dans le chapitre 4.12.4.2 est réutilisé ici.

4.13.4.2.1 La solution avec SOFI discret

Comme cela a été dit dans le chapitre 4.6, SOFI permet de calculer année par année ou bien mois par mois. Dans le cas de la partie précédente, le décalage est de trois mois. Ceci empêche de calculer par années. Ici, au contraire, comme les décalages sont de une et deux années, le calcul peut aussi bien être exprimé pour une périodicité mensuelle ou annuelle. En effet, un décalage d'une année est équivalent à un décalage de 12 mois. Le choix entre ces deux périodicités dépend du niveau de détail souhaité.

Le choix de la périodicité annuelle est donc remis en cause afin de réaliser des estimations plus fines. Comme SOFI discret ne permet pas de période plus fine que le mois, c'est celle-ci qui est choisie.

Comme le présent est modélisé par un intervalle d'une année, 12 valeurs sont évaluées pour chaque « n » de Janvier 2015 à Décembre 2015. Afin d'estimer la vente de tickets pour le mois de Janvier 2015, la moyenne de la valeur de Janvier 2014 et Janvier 2013 est réalisée. La formule précédente est adaptée puisque l'unité de « n » passe d'une année à un mois. En effet, l'expression « n - 1 » fait référence au mois précédent et non plus à l'année précédente. Afin de faire référence au mois de l'année précédente, il faut remonter de 12 mois. C'est pourquoi le décalage a été multiplié par 12.

```
Présent :
"Participation des familles"[n] := ("Participation des familles"[n - (1 * 12)] + "Participation des
familles"[n - (2 * 12)]) / 2

Futur :
"Participation des familles"[n] := "Participation des familles"[n - (1 * 12)]
```

Figure 94 : La formule de la variable « Participation des familles » pour une périodicité mensuelle en modélisation discrète

Une variable annuelle peut utiliser la variable « Participation des familles » bien que celle-ci soit mensuelle. Dans ce cas, les valeurs de chaque mois d'une année sont additionnées pour obtenir la valeur de l'année. Par contre, l'inverse pose problème. En effet, pour qu'une variable mensuelle puisse utiliser une variable annuelle, il faudrait « deviner » les valeurs de chaque mois à partir des valeurs par année.

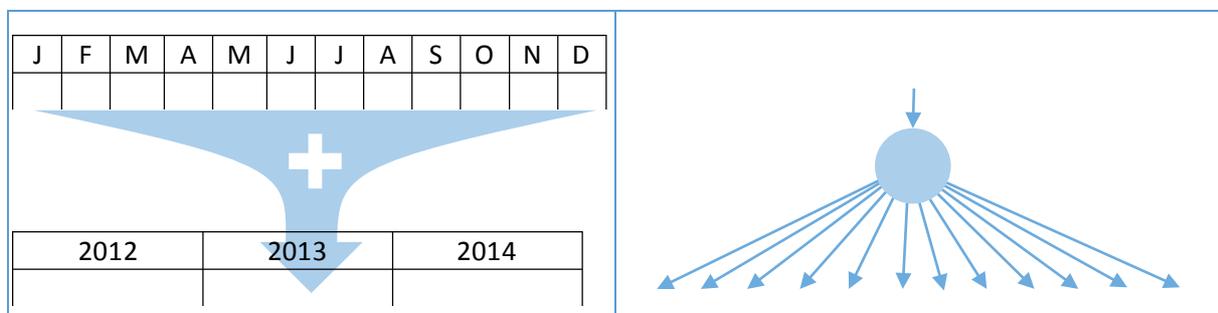


Figure 95 : Les variables annuelles peuvent utiliser des variables mensuelles, mais l'inverse n'est pas possible

Une interpolation linéaire semble être une solution pour estimer les valeurs de chaque mois à partir des valeurs par année, mais ceci est adapté uniquement pour des variations douces. Dans le cas des

ventes de billets, la valeur par mois ne suit pas la tendance des variations annuelles. Les valeurs dépendent du comportement d'achat des parents au cours de l'année. Une répartition similaire est retrouvée d'une année sur l'autre : de nombreuses ventes ont lieu en Septembre, puis de petites variations sont observées au cours de l'année. Les deux figures ci-dessous donnent un exemple sur trois ans avec la répartition réelle à gauche et celle évaluée avec une interpolation linéaire à droite. L'interpolation linéaire est dans ce cas complètement incohérente car elle donne une répartition à peu près constante au cours de l'année alors que c'est loin d'être la réalité. C'est pourquoi, SOFI discret ne permet pas de calculer des variables mensuelles à partir de variables annuelles.

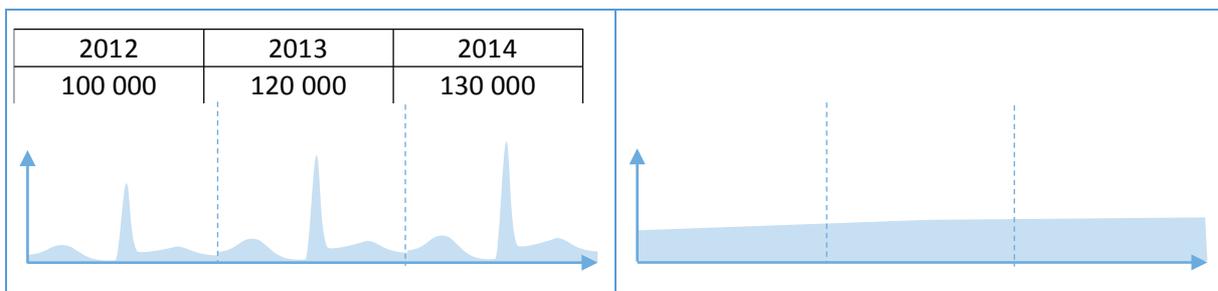


Figure 96 : La répartition réelle des ventes de tickets

Figure 97 : L'interpolation de la vente de tickets par année

Lorsque le besoin en finesse des estimations augmente, un modèle SOFI doit être revu presque intégralement. En effet, si une variable annuelle est transformée en variable mensuelle, toutes les variables dont elle dépend doivent être transformées à leur tour et ainsi de suite.

4.13.4.2.2 La solution avec SOFI continu

Le gestionnaire saisit une nouvelle répartition des ventes de tickets davantage proche de la réalité. Le modèle continu ne nécessite aucune autre modification pour prendre en compte cette répartition plus fine.

La précision apportée bénéficie à toutes les variables qui dépendent des ventes de tickets. En effet, la modélisation continue permet de conserver cette augmentation de finesse tout au long des calculs. Par exemple, comme illustrée sur la figure ci-dessous, la moyenne réalisée sur les ventes des deux dernières années conserve le détail de la répartition des ventes au cours de l'année.

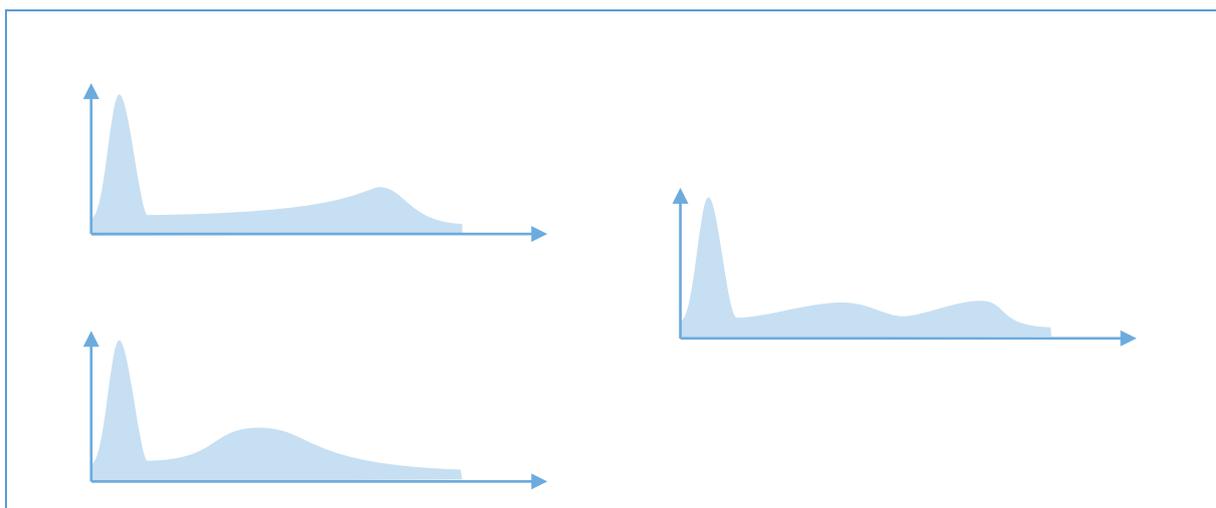


Figure 98 : La moyenne des répartitions de ventes de tickets

En mettant à jour la Figure 92 de la partie précédente avec la répartition plus fine des ventes de tickets, on observe que le calcul tient compte de cette répartition, alors que les formules de calcul sont identiques à celle utilisées précédemment. Ceci est dû au fait que la modélisation continue se

base sur des fonctions mathématiques. Ainsi, elle est prévue pour être évaluée à différentes échelles de temps en fonction du contexte. Par exemple, les élus se basent sur une échelle annuelle pour prendre des décisions dont l'impact s'étale sur plusieurs années alors que le gestionnaire de la cantine prévoit le budget d'une année semaine par semaine.

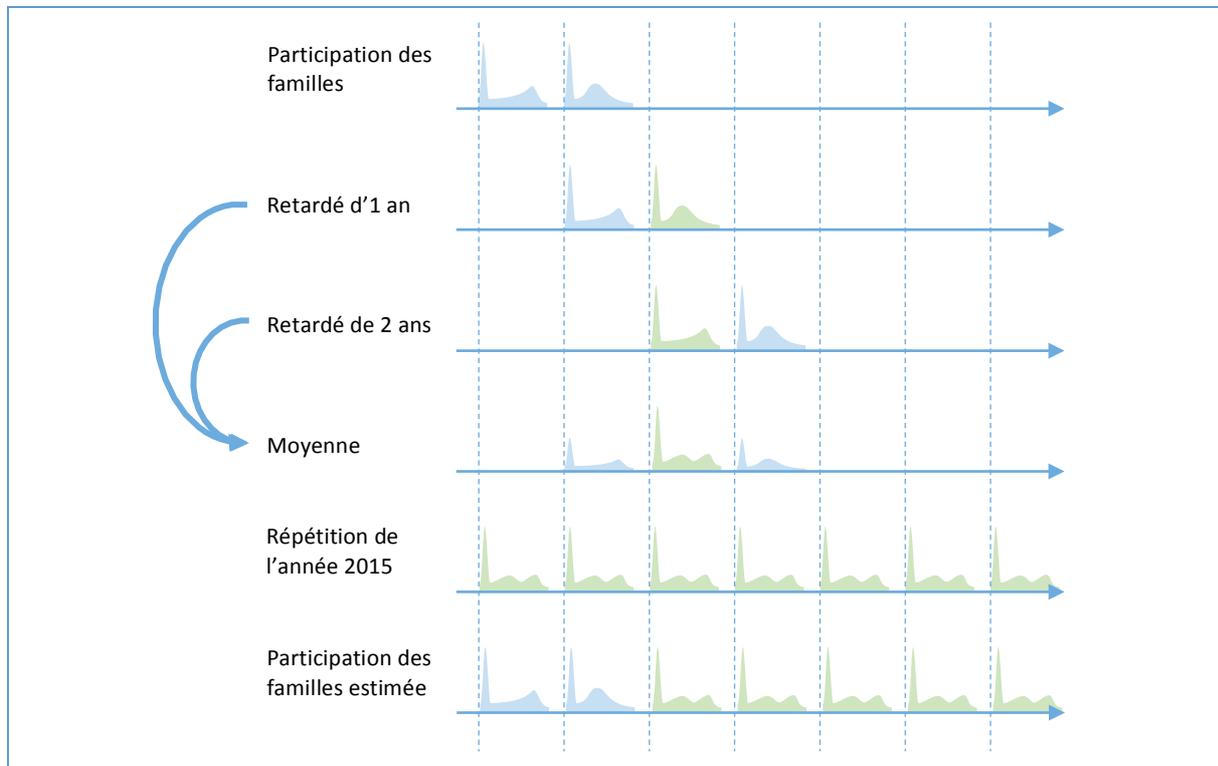


Figure 99 : Les 2 étapes du calcul de la variable « Participation des familles » en modélisation continue avec une répartition plus précise

4.13.4.2.3 La comparaison des solutions

La modélisation de SOFI discret est fortement liée à la périodicité choisie pour exprimer les modèles. Ceci a pour conséquence de rendre difficile le changement de périodicité pour ajouter de la finesse à la saisie et aux estimations des ventes de tickets. En effet, les formules d'estimation des ventes des années à venir ont dû être réécrites pour tenir compte du changement de périodicité. Ceci est d'autant plus problématique que l'utilisateur réalisant la saisie n'est pas nécessairement un modélisateur.

De plus, la modélisation de SOFI discret ne permet pas d'évaluer les modèles pour des périodes plus petites qu'un mois pour des raisons de performances.

La nouvelle saisie de la répartition des ventes au cours de l'année sur le modèle continu n'a pas été plus compliquée qu'un simple changement de valeurs. En fait, dans le contexte de la modélisation continue, le passage d'une saisie par année à une saisie plus fine correspond réellement à un simple changement des valeurs saisies. C'est une des conséquences du fait que la modélisation continue soit indépendante des périodes d'évaluation : les valeurs peuvent être placées à la position temporelle exacte souhaitée sans avoir à retravailler les formules de calcul.

4.13.5 L'exemple de l'estimation du coût des denrées alimentaires

4.13.5.1 Le besoin

L'estimation du futur fait souvent intervenir des notions de croissance exponentielle. Par exemple, l'inflation fait augmenter le coût des aliments d'une année sur l'autre. Les équations des modèles doivent permettre d'exprimer cette croissance.

Ce coefficient provient en général d'organismes de référence qui fournissent des valeurs tenant compte de paniers auxquels il est possible de se référer pour une meilleure évaluation. Dans notre cas, l'indice INSEE du coût à la consommation est une bonne référence.

4.13.5.2 La solution avec SOFI discret

Dans le cas d'une modélisation discrète, la moyenne sur une année du coût en denrées moyen d'un repas est calculée en multipliant simplement les dépenses de l'année précédente par le coefficient d'inflation.

```
Présent et Futur :  
"Coût d'un repas"[n] := 1.02 * "Coût d'un repas"[n - 1]
```

Figure 100 : La formule de la variable « Coût d'un repas » pour une périodicité annuelle en modélisation discrète

Afin de prendre en compte les variations des prix dues aux saisons, la variable doit passer à une périodicité mensuelle et la formule doit être adaptée. La formule est modifiée afin que chaque mois gagne 2% d'une année sur l'autre. Dans certains cas, une racine douzième aurait été appliquée au coefficient afin d'obtenir une pente progressive. Cependant, dans le cas présent, l'inflation au cours de l'année est déjà présente dans les données saisies sur les années passées.

```
Présent et Futur :  
"Coût d'un repas"[n] := 1.02 * "Coût d'un repas"[n - 12]
```

Figure 101 : La formule de la variable « Coût d'un repas » pour une périodicité mensuelle en modélisation discrète

4.13.5.3 La solution avec SOFI continu

La première solution qui vient à l'esprit pour la modélisation de l'augmentation du coût est de faire l'analogie à la modélisation discrète. En effet, en multipliant les données saisies pour une année par un facteur constant chaque année avec une fonction en plateaux, le coût évalué peut être considéré comme acceptable.

Cependant, la modélisation n'est pas décomposée de façon logique. En effet, cette fonction en plateaux ne se raccroche pas à un concept de la réalité. Se baser sur une telle logique nuirait aux capacités d'évolutions du modèle.

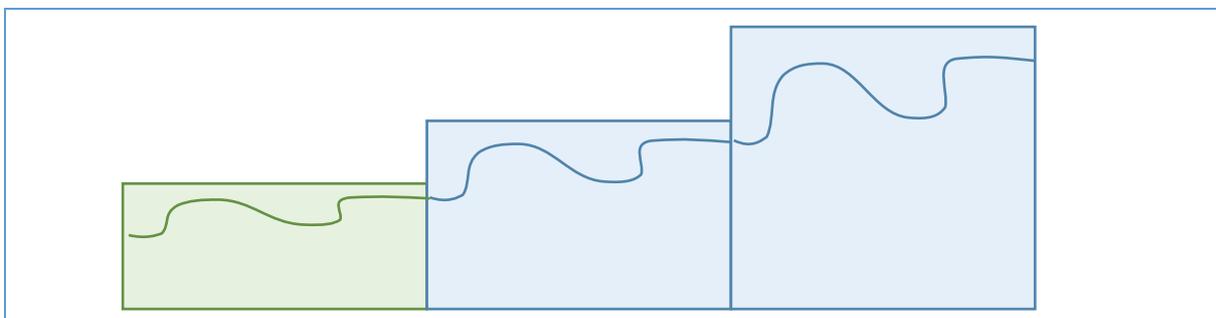


Figure 102 : L'estimation du coût des denrées avec une méthode analogue à celle utilisée pour SOFI discret

Afin d'estimer les coûts des années futures avec davantage de précision si nécessaire, il est important de comprendre ce qui détermine son évolution. En décomposant l'évolution du prix en plusieurs facteurs, il sera possible de jouer sur ceux-ci pour gagner en précision si besoin.

De nombreux facteurs sont responsables de l'évolution du coût des denrées. Les deux facteurs qui semblent avoir le plus importance ici sont l'inflation et les variations dues aux saisons.

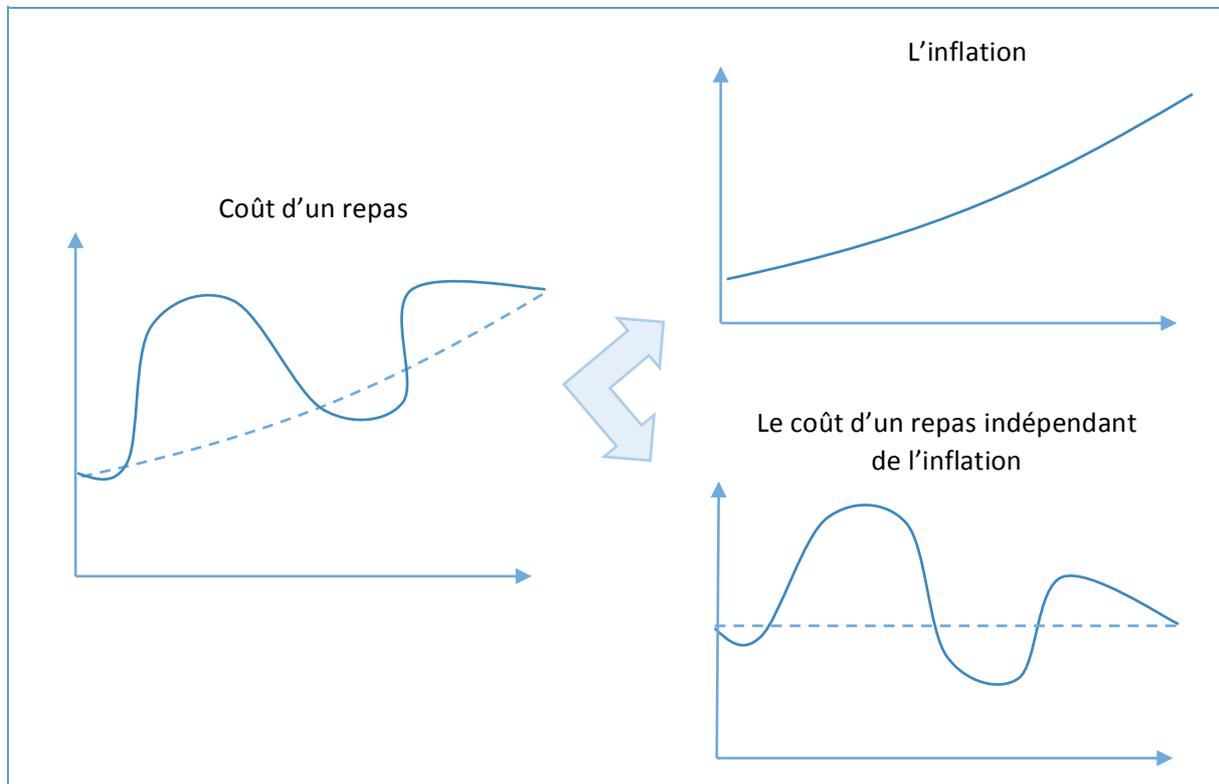


Figure 103 : La décomposition des variations du coût des denrées en plusieurs facteurs

La répartition du coût des denrées prend déjà en compte l'inflation. Afin d'en extraire une répartition indépendante de cette croissance, la variable est divisée par l'exponentielle qui représente cette croissance. La fonction « Exp » de la formule ci-dessous décrit une fonction exponentielle qui vaut « 1 » pour le début de l'année courante et croit de 2% par an.

```
"Coût d'un repas"(t) / Exp(BeginningOf(Year(1), "now"()), 1.02, Year(1))
```

Figure 104 : La formule du coût indépendant de l'inflation ramené au 1^{er} Janvier de l'année du présent

Une fois la répartition au cours de l'année déterminée, celle-ci est répliquée sur les années du futur. L'estimation réalisée avec la formule ci-dessous ne prend pas en compte l'inflation.

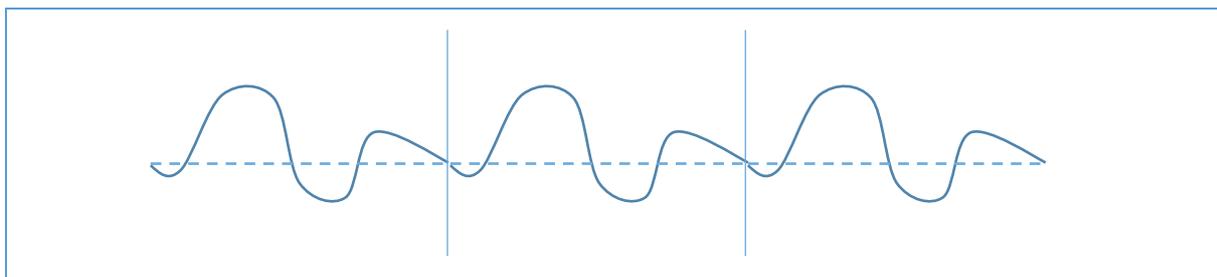


Figure 105 : La courbe de répétition du coût d'un repas indépendant de l'inflation

```

If(t < BeginningOf(Year(1), "now"()),
  0,
  Repeat
  (
    BeginningOf(Year(1), "now"() - Year(1)),
    Year(1),
    "Coût d'un repas"(t) / Exp(BeginningOf(Year(1), "now"()), 1.02, Year(1)))
  )
)

```

Figure 106 : La formule de répétition du coût d'un repas indépendant de l'inflation

Afin de prendre en compte l'inflation dans l'estimation du futur, les répartitions dupliquées sont multipliées par une fonction exponentielle qui représente l'inflation.

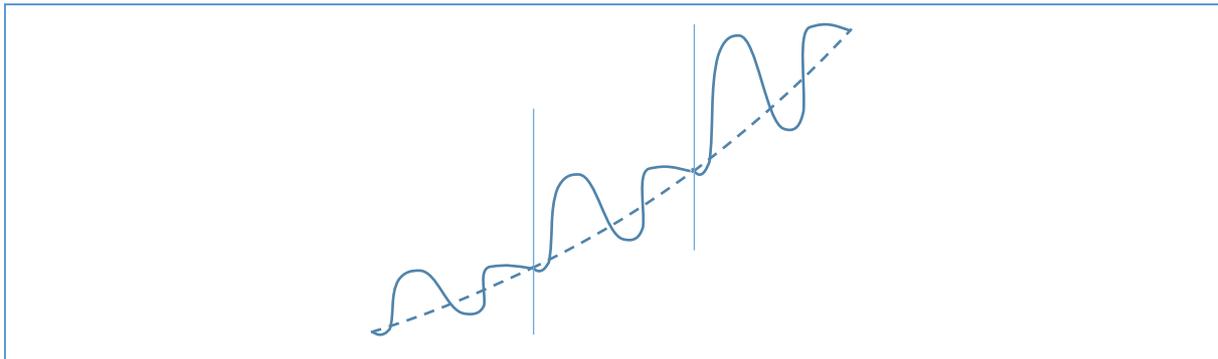


Figure 107 : La courbe de l'estimation du coût des denrées d'un repas

```

"Coût d'un repas"(t) :=
If(t < BeginningOf(Year(1), "now"()),
  0,
  Repeat
  (
    BeginningOf(Year(1), "now"() - Year(1)),
    Year(1),
    "Coût d'un repas"(t) / Exp(BeginningOf(Year(1), "now"()), 1.02, Year(1)))
  )
  * Exp(BeginningOf(Year(1), "now"()), 1.02, Year(1))
)

```

Figure 108 : La formule de la variable « Coût d'un repas » en modélisation continue

D'une année à l'autre, la variation du coût due aux saisons n'est pas tout à fait identique. Des événements non déterminés peuvent induire du bruit dans cette variation. Ce bruit peut être éliminé car il n'est pas pertinent de le conserver d'une année sur l'autre.

Une moyenne sur plusieurs années permet de réduire ce bruit, mais ce n'est pas la méthode la plus efficace. L'idéal serait de déterminer les facteurs nécessaires au calcul en utilisant la théorie du signal. La réduction du bruit n'est pas approfondie car elle ne changerait pas fondamentalement l'approche.

4.13.5.4 La comparaison des solutions

Les deux solutions proposées fournissent des résultats identiques, mais la modélisation continue semble mener intuitivement vers une logique plus proche de la réalité que celle discrète. Ceci est encourageant concernant la qualité sémantique des modèles continus.

4.13.5.5 Les évolutions possibles

L'approche prise pour SOFI continu est de décomposer les variations du coût des denrées. L'analyse du spectre des variations permettrait une analyse plus fine des différents facteurs participant à ces variations afin d'améliorer les estimations réalisées si nécessaire.

4.13.6 La comparaison des solutions

La modélisation discrète force à choisir la périodicité d'évaluation avant de modéliser. En effet, les durées sont exprimées en nombre de périodes les formules de calculs ; ces formules sont différentes si cette période est un mois ou une année. De plus, les dates et les durées sont arrondies à une période d'évaluation près. Les estimations laissent dans le flou concernant ce qui devrait se passer au cours d'une période d'évaluation. Sachant que la période d'évaluation la plus fine permise par SOFI discret est d'un mois, réaliser un suivi n'est pas envisageable.

La modélisation continue permet de réaliser des modélisations plus proches de la réalité en levant les contraintes de périodicités. En effet, les dates et les durées utilisées dans les formules de calcul permettent de décrire de façon exacte le lien entre le passé et le futur. Les estimations sont placées précisément dans le temps. Ainsi, lors d'un suivi jour par jour, le responsable de la cantine peut vérifier si la planification concorde avec la réalité et reprogrammer la planification en cas d'imprévu.

Comme les variables d'une modélisation continue peuvent être évaluées selon n'importe quelle périodicité, plusieurs acteurs peuvent se baser sur le même modèle bien qu'ils travaillent à des échelles de temps différentes. C'est le cas du modèle continu de la construction d'une cantine. Celui-ci est utilisé par les élus en années pour planifier les crédits de paiements, le service technique en mois pour planifier les opérations de construction et le service scolaire en jours pour planifier le fonctionnement de la cantine.

4.13.7 Les évolutions possibles

Grâce aux fonctions présentées précédemment (Month...), l'expression du décalage définie pour SOFI continu est assez riche sémantiquement pour permettre l'utilisation d'un autre calendrier que le calendrier fiscal. Par exemple, avec un calendrier grégorien, en fonction de la valeur de t le décalage d'un mois peut être de 28 à 31 jours. Il n'est donc pas possible d'exprimer le décalage d'un mois avec une durée au sens scientifique (en secondes). C'est pourquoi ces fonctions sont utilisées pour décrire le décalage temporel.

4.14 Le calcul des emprunts et des amortissements

4.14.1 Le besoin

Les amortissements et les emprunts sont des notions qui sont très couramment utilisées lors d'une planification financière. Bien qu'il existe plusieurs types d'emprunts et d'amortissements, par exemple les emprunts à remboursement constant, à annuité constante, avec des différés, etc., leur calcul est identique d'une planification financière à une autre. Permettre de modifier la modélisation de leur calcul ne présente pas d'intérêt. Ces calculs peuvent donc être définis directement dans l'implémentation du moteur de calcul afin d'éviter de reproduire leur calcul dans chaque modélisation.

4.14.2 La solution avec SOFI discret

Le moteur de calcul de SOFI discret transforme les modèles de planification financière en feuilles de calcul qui sont évaluées par un tableur. Le calcul des emprunts et des amortissements n'est pas défini directement sur ces feuilles de calcul mais une fonction personnalisée fait la liaison avec une méthode définie dans le code du moteur de calcul. Ceci permet de factoriser le calcul des emprunts

et des amortissements qui serait autrement dupliqué dans la feuille de calcul pour chacune de leur utilisation dans le modèle de planification financière source.

4.14.3 La solution avec SOFI continu

La modélisation mathématique du calcul des emprunts et des amortissements dans SOFI continu est fortement inspirée de celle présentée dans [1].

Les emprunts et les amortissements sont représentés par des éléments dont la propriété « OpaqueComutation » est définie « Emprunt » ou « Amortissement ». Les variables nécessaires au calcul et celles fournissant le résultat sont créées automatiquement lors de la création de l'élément. Cependant, ces éléments ne contiennent pas la modélisation des calculs d'emprunt ou d'amortissement. Le moteur de calcul se charge de rajouter les calculs lors de la traduction en modèle mathématique. Tous les calculs nécessaires à l'évaluation du modèle mathématique sont contenus dans celui-ci. Ceci est important pour pouvoir profiter des propriétés mathématiques de ce modèle, comme par exemple l'inversion des opérations.

Il est expliqué dans le chapitre 4.2 que lorsqu'un modèle de planification budgétaire est traduit en modèle mathématique, chaque variable est traduite en une arborescence d'opérateurs de fonctions mathématiques qui sont ensuite interconnectés pour former un graphe. Les arborescences correspondant aux variables d'un élément d'emprunt et d'amortissement sont définies en dur dans l'algorithme de traduction du modèle de planification financière mais sont insérées dans le graphe de la même manière que les autres variables du modèle.

Les deux diagrammes d'instances ci-dessous représentent les graphes d'opérateurs de fonctions qui permettent de calculer les amortissements et les emprunts. Pour des raisons de lisibilité, les nœuds permettant la définition de valeurs sur les variables ne sont pas présents sur ces schémas.

La mesure du capital amorti pendant la durée de l'amortissement est calculée à partir de la mesure du capital à amortir ainsi que de la mesure du patron d'amortissement. Le patron définit la durée ainsi que la répartition de l'amortissement. La mesure du capital à amortir définit le positionnement dans le temps du capital à amortir. L'opération mathématique de convolution « IConvolutionMeasure » est utilisée afin d'obtenir le capital amorti à partir de ces deux mesures.

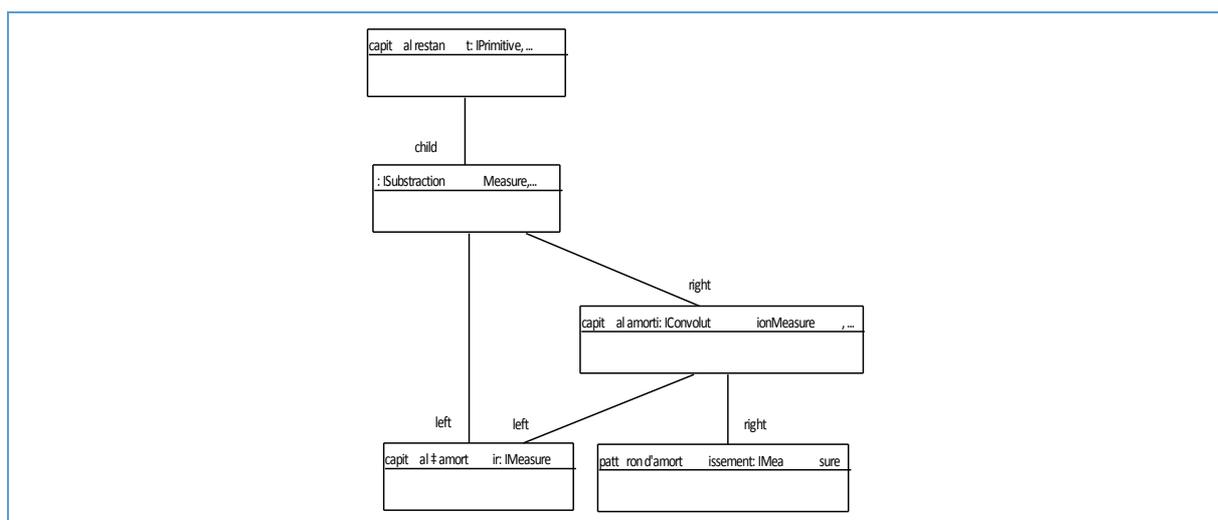


Figure 109 : Un diagramme d'instance du calcul de l'amortissement avec LEMF

Lorsque la date du début de l'amortissement est connue cette fonction est représentée par une répartition de Dirac. Ceci permet de concentrer le capital à amortir à l'instant précis où l'amortissement débute.

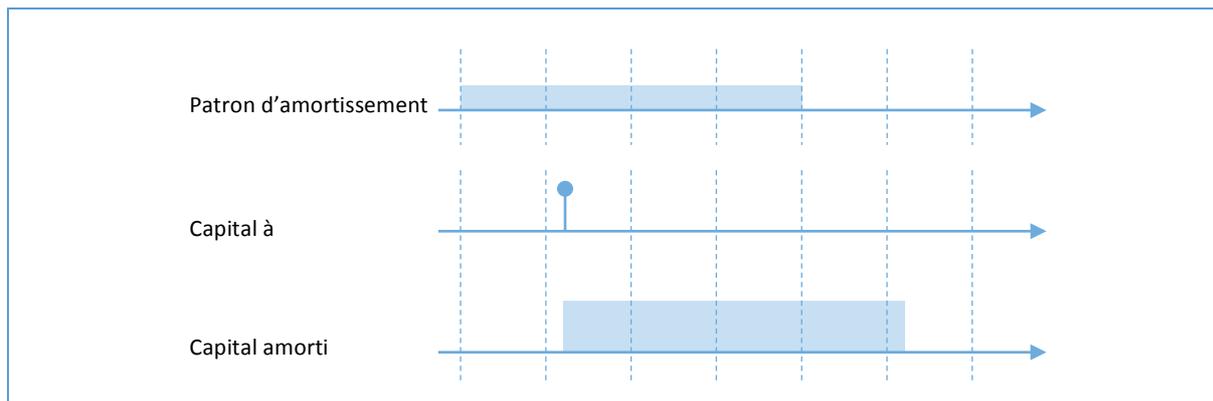


Figure 110 : Les courbes d'un calcul d'amortissement

Dans le cas d'une incertitude concernant le début de l'amortissement, le capital à amortir peut être davantage réparti avec par exemple une fonction gaussienne afin de tenir compte de cette incertitude dans l'évaluation du modèle.

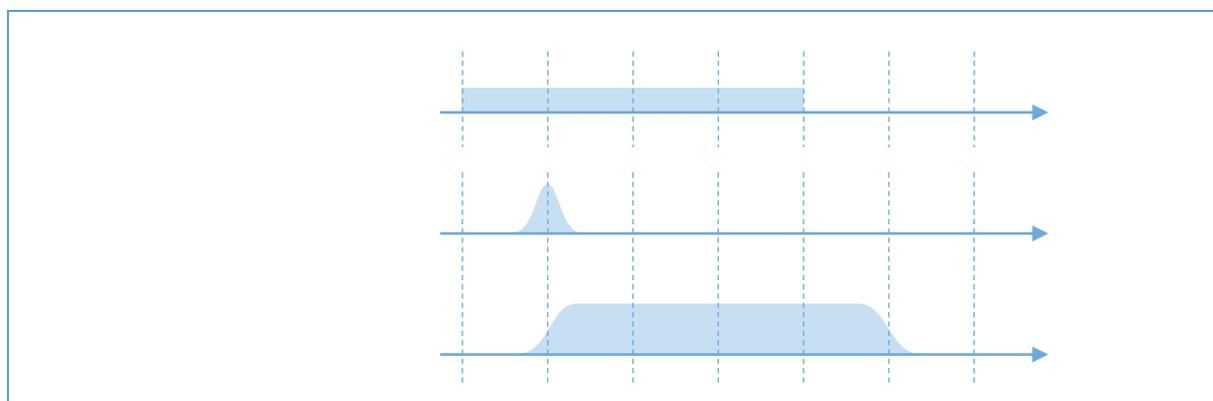


Figure 111 : Les courbes d'un calcul d'amortissement avec une incertitude

Le calcul de l'emprunt reprend le même principe que celui de l'amortissement et introduit le calcul des intérêts. Le modèle mathématique défini pour le calcul d'emprunt est différent des modèles utilisés habituellement par les banques. Au lieu d'un remboursement ponctuel à intervalle régulier, le modèle de l'emprunt utilisé par SOFI continu définit le remboursement réalisé de façon continue. Les résultats obtenus avec ce modèle diffèrent de moins de 10% par rapport au modèle discret utilisé par les banques. De plus, l'utilisation d'un modèle continue permet de conserver des temps de réponses acceptable pour les raisons exposées dans le chapitre 4.6.

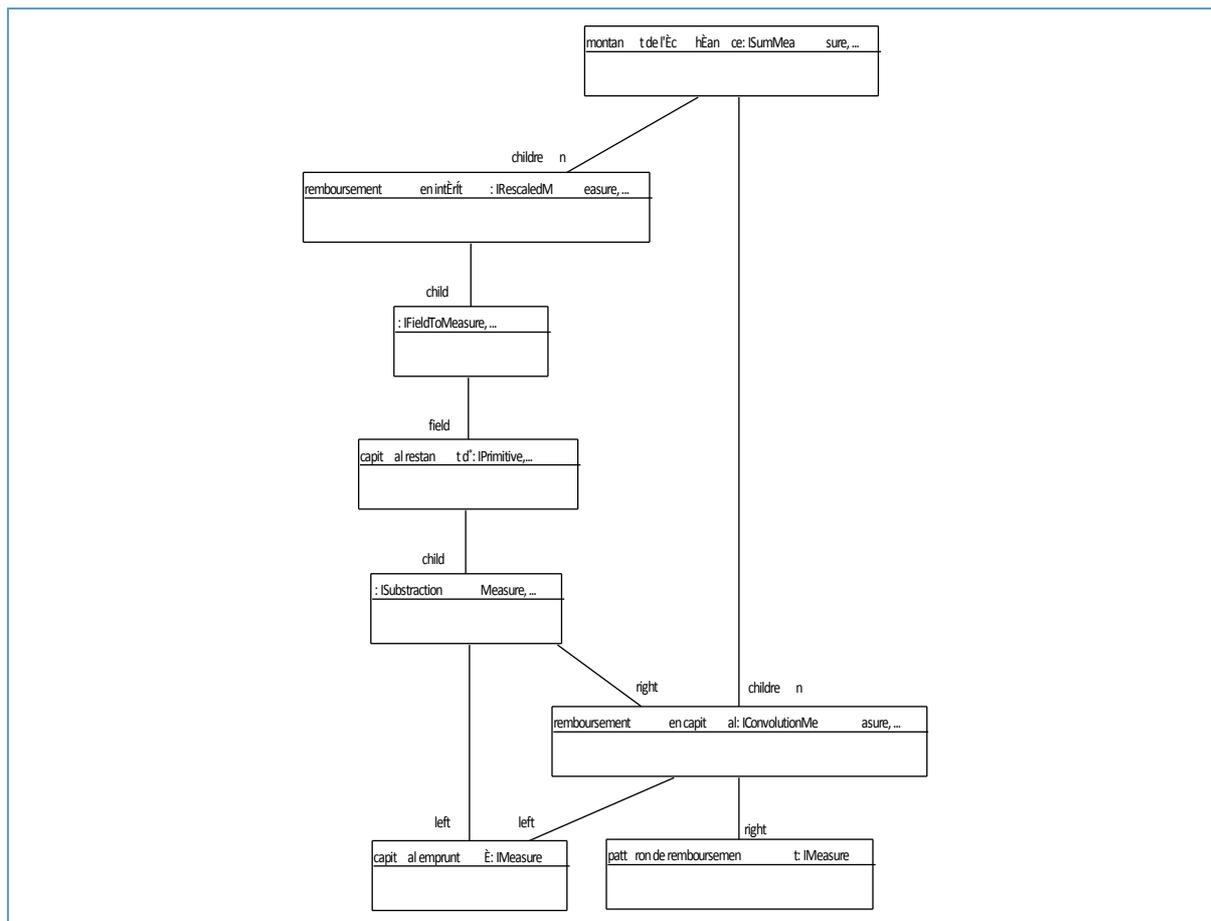


Figure 112 : Un diagramme d'instance du calcul de l'emprunt avec LEMF

La figure ci-dessous détaille le calcul d'un emprunt à remboursement de capital constant. Le même graphe d'opérations mathématiques permet de calculer d'autres types d'emprunt. Par exemple, pour calculer un emprunt à échéances constantes, le patron de remboursement doit être adapté. Au lieu d'être constant sur toute la durée de l'emprunt, il serait croissant.

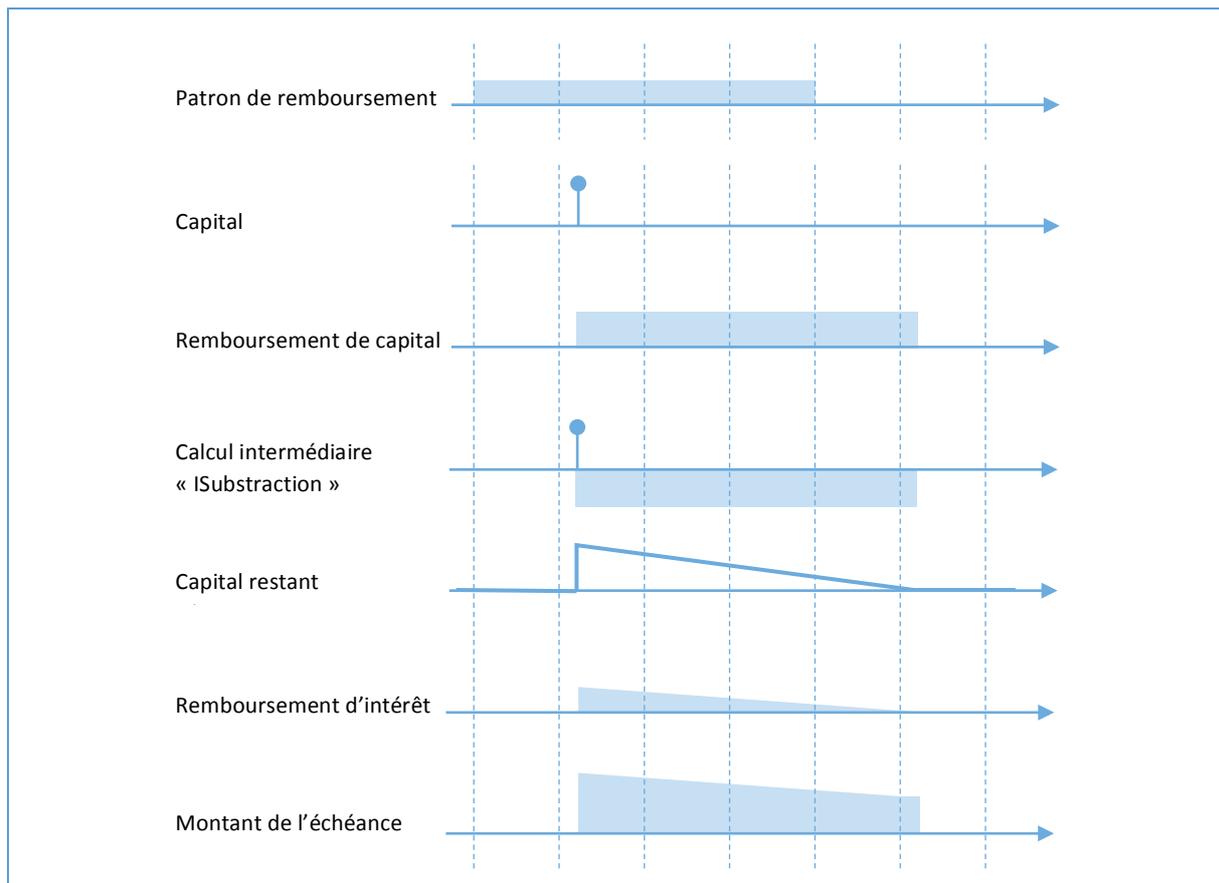


Figure 113 : Les courbes d'un calcul d'emprunt à amortissement constant

La variable du capital restant dû est un champ. Sa courbe s'interprète donc différemment des autres courbes. Chaque point de la courbe correspond au montant restant à rembourser. Comme détaillé dans le chapitre 4.7, c'est la hauteur de la courbe qui contient le sens de la variable.

4.14.4 La comparaison des solutions

Le calcul des emprunts dans SOFI continu est réalisé en s'appuyant sur les principes de la modélisation continue exposés dans les chapitres précédents afin de conserver leurs avantages au détriment d'une erreur d'évaluation acceptable.

4.14.5 Les évolutions possibles

Les besoin initialement décrits ont été remplis. Aucune évolution n'est envisagée actuellement.

4.15 Conclusion

L'approche propose d'utiliser la modélisation en temps continue afin de permettre la construction d'un modèle unique qui regroupe des concepts de périodicité hétérogène et son évaluation pour n'importe quelle période de temps. Pour ce faire, un métamodèle a été présenté en détaillant point par point comment celui-ci permet de représenter les notions liées au temps fidèlement.

Une architecture s'inspirant du patron architectural CQRS a été proposée afin de permettre une parallélisation massive des requêtes de calculs.

Le chapitre suivant détaille l'application de ces propositions au cas d'étude afin d'évaluer leurs faisabilités.

5 Expérimentation et validation

Dans un premier temps, les technologies ont servi pour l'implémentation du prototype seront détaillées, puis dans un second temps, la solution proposée est appliquée au cas d'étude.

5.1 L'implémentation de la solution

5.1.1 La bibliothèque de calcul continu LEMF

La bibliothèque LEMF est un calculateur numérique. Elle a été conçue dans l'optique de permettre à des non-mathématiciens d'utiliser des calculs numériques complexes dans leurs logiciels.

Contrairement à un logiciel de calcul numérique classique qui s'utilise avec un langage dédié, les calculs exprimés avec LEMF sont des compositions d'objets. Ceci permet de masquer certaines notions mathématiques qui pourraient être difficiles à appréhender par un développeur.

La définition d'un calcul peut être vue comme un graphe dont les nœuds terminaux sont des fonctions usuelles et les nœuds non-terminaux des opérations. LEMF permet à l'utilisateur de construire des graphes de calcul dynamiquement et de les faire évoluer.

5.1.1.1 L'interface de la bibliothèque

Les fonctionnalités de LEMF sont fournies par l'intermédiaire d'un ensemble d'interfaces dans le but de limiter le couplage, à la bibliothèque, des applications utilisatrices. Le couplage se limite au rôle de la bibliothèque : la manipulation du graphe de calcul et son évaluation. Les choix d'implémentation du calcul n'impactent pas l'utilisateur. Ceci est d'autant plus important qu'une bibliothèque de calcul peut être revue totalement dans le but d'optimiser les temps de calculs qui sont souvent critiques pour l'application utilisatrice.

La construction du graphe de calcul est réalisée par l'intermédiaire d'une fabrique abstraite (AbstractFactory). Ainsi, les opérations et les fonctions créées par cette fabrique sont utilisées par l'intermédiaire d'interfaces. Ceci évite de coupler les mécaniques liées à la réalisation du calcul et la construction du graphe de calcul. En effet, ceci permet de limiter ce qui est exposé aux utilisateurs de la bibliothèque.

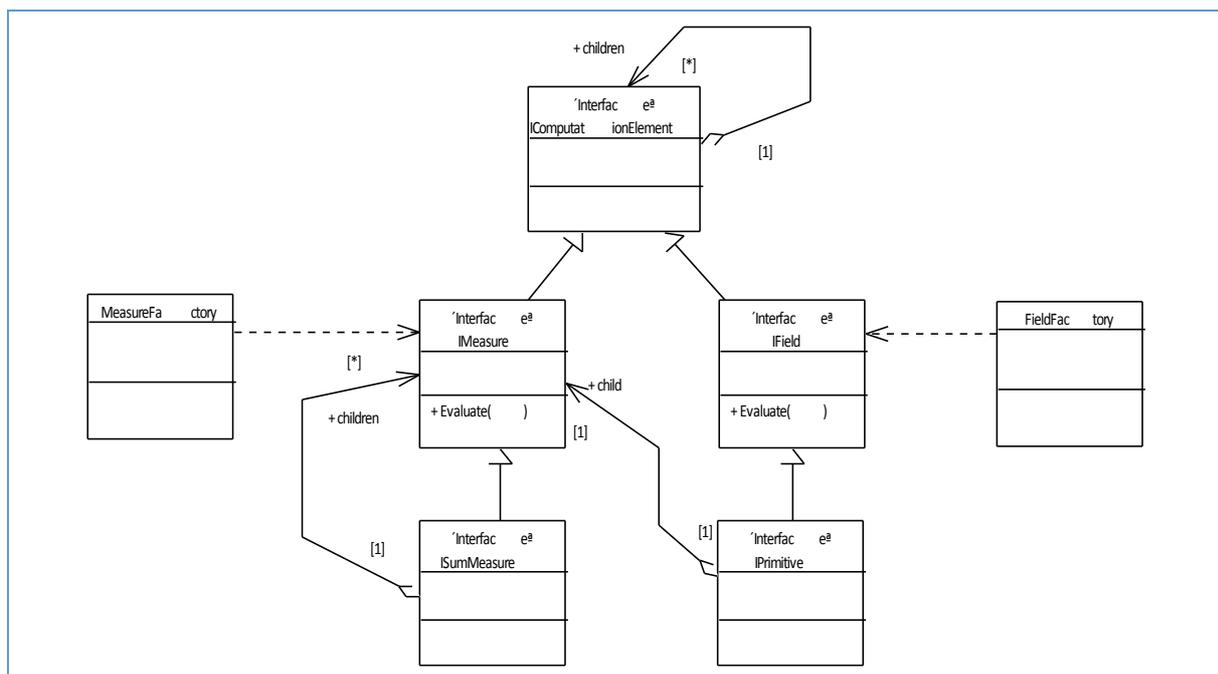


Figure 114 : Le diagramme de classes de l'interface de la bibliothèque de calcul continu

5.1.1.2 La parallélisation

La parallélisation de l'évaluation des mesures et des champs est réalisée avec une bibliothèque de Microsoft dédiée au calcul parallèle nommée « Task Parallel Library ». Le découpage est réalisé au niveau des opérations qui possèdent plusieurs opérandes. L'évaluation de chaque opérande peut potentiellement être exécutée en parallèle. Le choix de l'exécution effective est le rôle de la bibliothèque de Microsoft. Ce choix est réalisé en fonction de l'occupation des processeurs afin d'utiliser ceux-ci de façon optimale.

5.1.2 Les technologies utilisées pour implémenter les services

Les services ont été implémentés avec WCF car MGDIS souhaitait utiliser les technologies de Microsoft pour ce projet tout en étant compatible avec les standards de service web tel que WSDL et SOAP.

5.2 La validation

La validation de la solution proposée est réalisée en l'appliquant au cas d'étude en construisant un modèle dans une première partie puis en utilisant celui-ci afin de prendre des décisions dans une deuxième partie. Une troisième partie présente un test de performance réalisé sur l'évaluation du modèle du cas d'étude par le prototype du service de calcul. La validation est conclue par une réponse aux questions de recherches.

5.2.1 La construction du modèle du cas d'étude

Cette partie reprend le cas d'étude présenté dans le chapitre 2.2 afin de montrer comment il peut être modélisé conformément à la solution proposée dans le chapitre 4.

5.2.1.1 Les budgets d'investissements et de fonctionnement

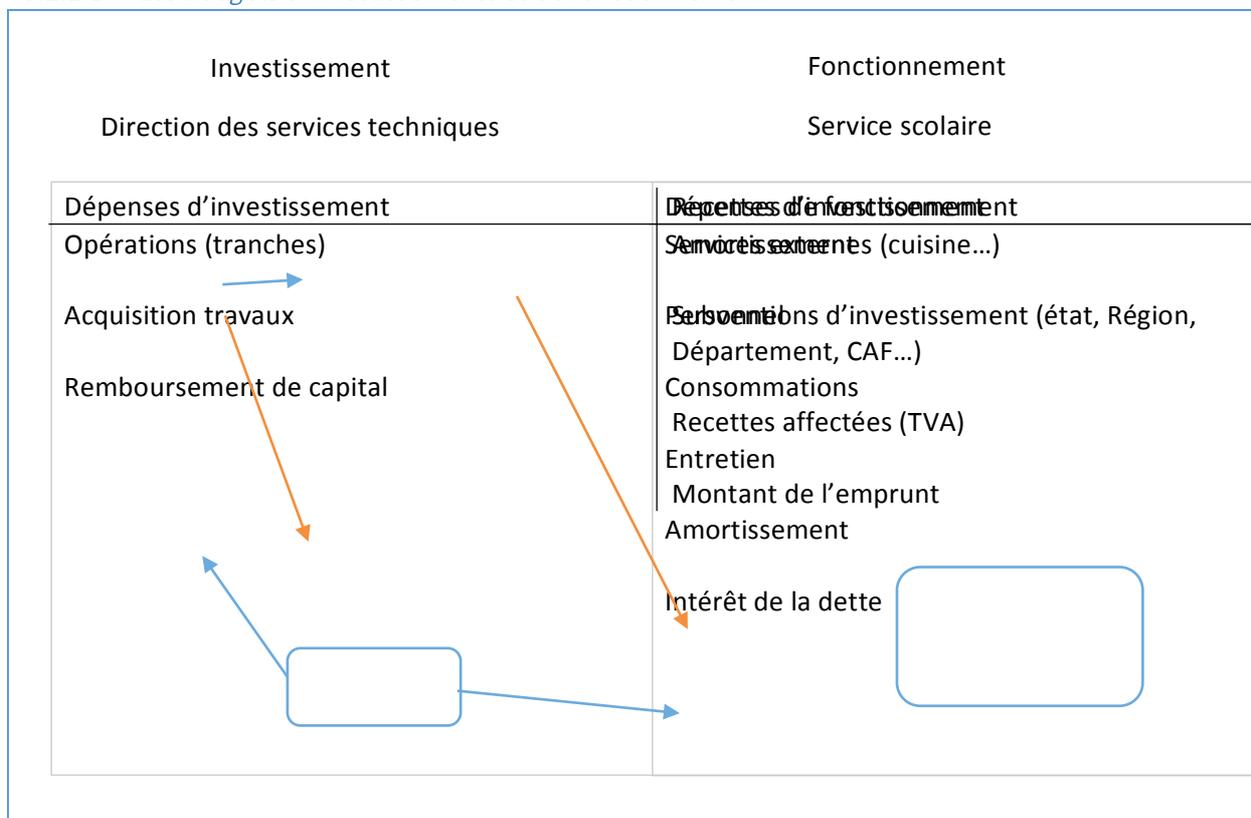


Figure 115 : La vue d'ensemble du budget d'investissement et de fonctionnement

Le modèle de planification budgétaire appliquée à la restauration scolaire décrit les calculs nécessaires à l'estimation d'un budget d'investissement et d'un budget de fonctionnement d'une

cantine. Chacun de ces budgets est modélisé par un élément (respectivement « Budget d'investissement » et « Budget de fonctionnement ») dont les variables sont présentées dans la figure ci-dessus.

Dans le cas d'étude, la modélisation du budget d'investissement est appliquée à la construction d'une cantine. Le budget d'investissement de la construction est modélisé par une instance de l'élément « Budget d'investissement ».

Dans le cas d'étude, la modélisation du budget de fonctionnement est appliquée au fonctionnement de l'ancienne cantine puis à la nouvelle cantine construite. Le basculement entre les deux budgets de fonctionnement est décrit dans le chapitre 5.2.1.3. Chacun des deux budgets de fonctionnements est modélisé par une instance de l'élément « Budget de fonctionnement ».

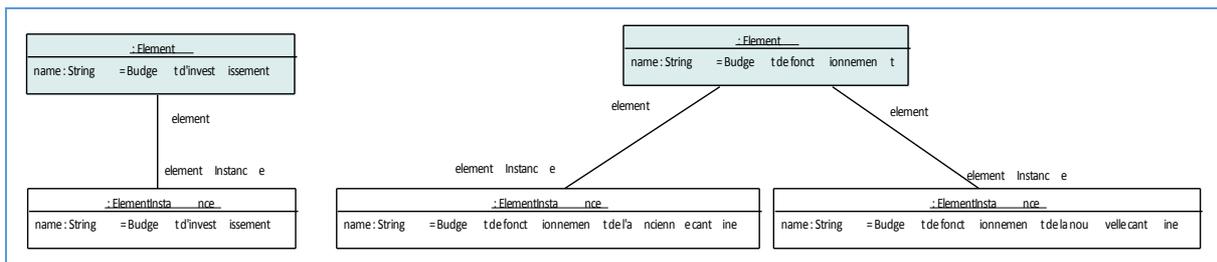


Figure 116 : Le diagramme d'instance montrant l'instanciation des budgets

5.2.1.2 L'association des budgets

Le budget de fonctionnement prend en compte les amortissements des opérations et les intérêts des emprunts des budgets d'investissement. La variable « Amortissement » d'une instance de l'élément « Budget de fonctionnement » est calculée en sommant les variables « Amortissement » des instances de l'élément « Budget d'investissement » associées. La variable « Intérêt de la dette » est calculée de la même façon. Cette variable n'apparaît pas dans la partie investissement de la Figure 115 car elle ne participe pas à l'équilibre du budget d'investissement. Elle présente dans l'élément « Budget d'investissement » uniquement car elle est calculée à partir des emprunts associés au budget d'investissement et que le budget de fonctionnement associé en a besoin.

Dans le cas d'étude, l'instance d'élément du budget d'investissement de la construction est associée aux instances du budget de fonctionnement de l'ancienne et la nouvelle cantine. En effet, l'impact de l'investissement de la construction sur le budget de fonctionnement commence avant le déménagement et continue ensuite.

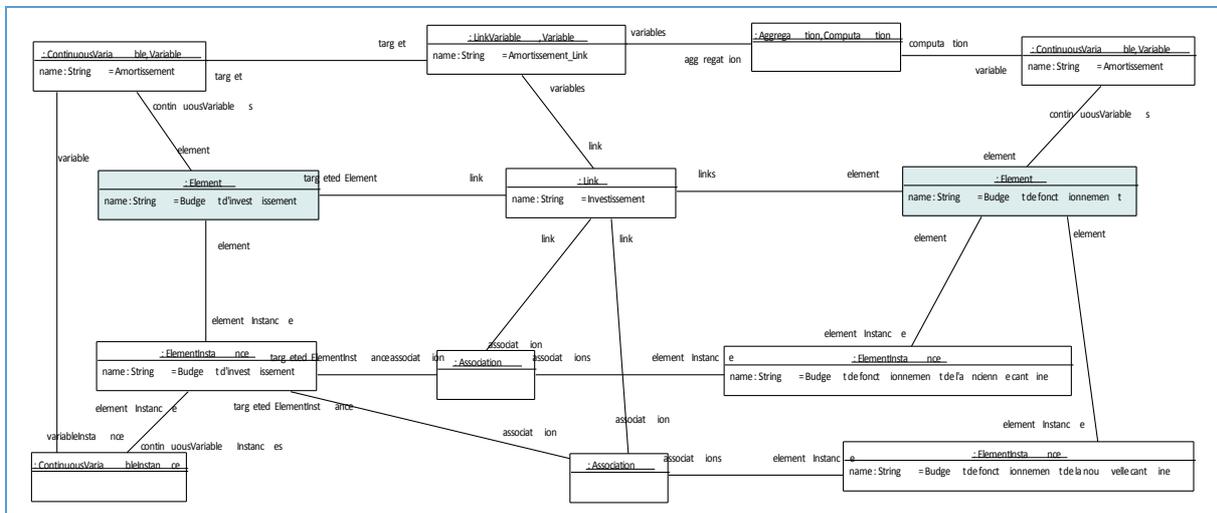


Figure 117 : Le diagramme d'instance montrant le lien entre le budget d'investissement et de fonctionnement au niveau du total d'amortissement

5.2.1.3 Le déménagement de la cantine

Le budget de fonctionnement de l'ancienne cantine et de la nouvelle sont différents. En effet, la nouvelle cantine possède une plus grande capacité et une meilleure capacité énergétique. Les valeurs données aux variables sont différentes ; par exemple la meilleure efficacité énergétique réduit les dépenses au niveau de la variable « Consommation ».

A la date du déménagement, le budget de fonctionnement bascule de l'ancienne cantine à la nouvelle. Un élément sert à aiguiller l'évaluation du budget de fonctionnement vers celui qui est effectif en fonction de la date. Cet élément possède les mêmes variables que l'élément du budget de fonctionnement. Toutefois, les calculs définis sont différents. À la place des calculs qui modélisent la logique d'un budget de fonctionnement, le calcul défini sur chaque variable redirige vers la variable correspondante de l'élément du budget de fonctionnement. Afin de différencier la variable de l'ancien et du nouveau budget, deux liens (« Avant le déménagement » et « Après le déménagement ») sont définis depuis l'élément « Budget de fonctionnement avec déménagement » vers l'élément « Budget de fonctionnement ».

Dans le cas d'étude, une seule instance est associée selon chacun des deux liens. Les agrégations sont réalisées sur une seule instance et pourraient donc sembler inutiles. Cependant, l'agrégation est obligatoire pour pouvoir utiliser une variable d'une autre instance puisque le nombre d'instances associées n'est pas contraint.

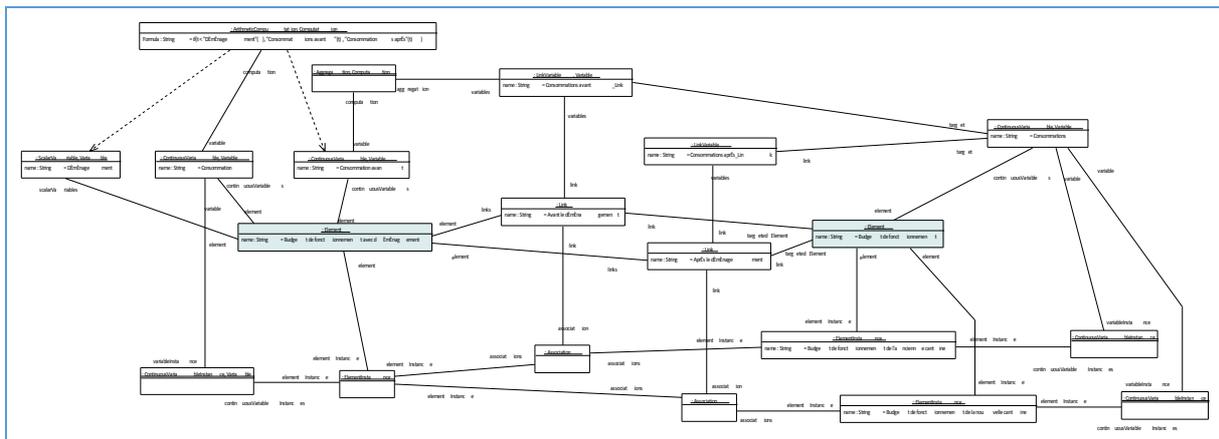


Figure 118 : Le diagramme d'instance montrant la modélisation du basculement entre les deux budgets de fonctionnement lors du déménagement

Dans le cas où plusieurs instances de l'élément « Budget de fonctionnement » sont associées selon le même lien, les variables de ces instances sont additionnées. Ceci permettrait de modéliser d'autres cas comme la construction d'une cantine qui s'ajouterait à celles existantes ou bien l'abandon de deux cantines pour une unique cantine plus grande, etc.

5.2.1.4 Le détail des variables

Les variables présentes dans la Figure 115 sont pour la plupart des totaux. C'est le cas de la variable correspondant aux dépenses liées aux opérations de construction. Comme la construction d'une cantine est un cas simple, aucune structure n'est imposée par le modèle concernant les opérations de constructions. Les opérations ajoutées par les membres du service des finances sont simplement additionnées.

Le calcul du total des dépenses des opérations est modélisé par l'agrégation d'un lien vers la variable des dépenses d'une opération. Le calcul d'une opération est défini dans le modèle « Opération » plutôt que dans le modèle « Investissement » afin de permettre d'associer plusieurs opérations à un budget d'investissement. L'ajout d'une opération se fait en instanciant le modèle « Opération ». Les variables « Dépenses » des instances de ce modèle sont additionnées par une agrégation pour obtenir la variable « Dépenses des opérations » de l'instance du modèle « Investissement ».

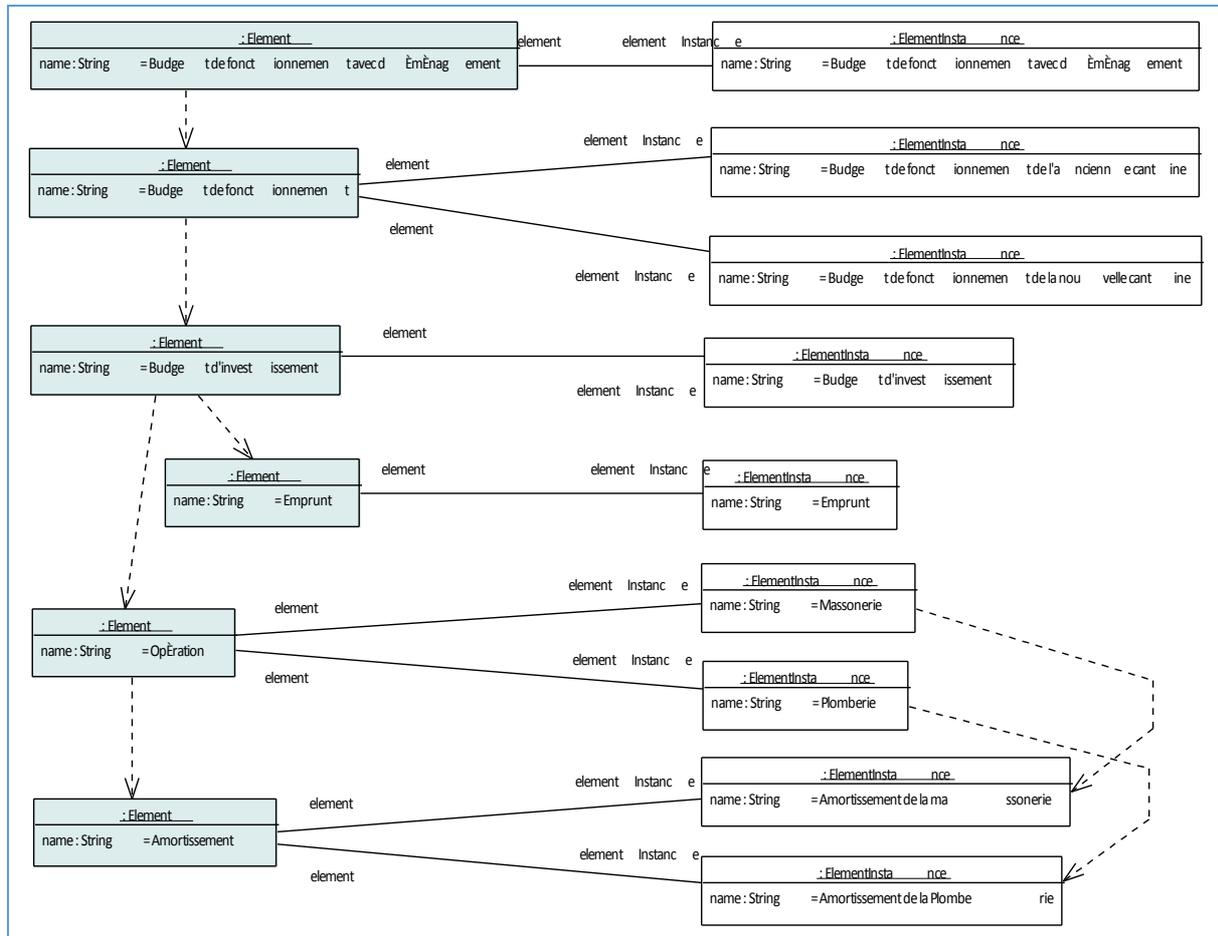


Figure 119 : Le diagramme d'instance montrant les dépendances entre les éléments du modèle

La modélisation est donc assez générale pour s'appliquer à d'autres types d'investissement comme une rénovation ou d'autres types de bâtiments. Cependant, pour des cas plus complexes, le modèle devrait permettre de réaliser des sous-totaux afin d'estimer le coût des opérations d'une certaine partie du bâtiment. Par exemple, pour un centre hospitalier les opérations de construction peuvent être séparées par département. Pour modéliser ceci, un enchaînement de deux agrégations est nécessaire : la première entre le modèle d'investissement et le modèle du sous-total par département et la seconde entre ce dernier modèle et le modèle de l'opération.

5.2.1.5 La participation des familles

La participation des familles est calculée en multipliant la quantité de tickets vendus par le prix des tickets. Pour simplifier le modèle, il est considéré que les tickets sont vendus à un tarif unique. La variable du prix des tickets est un champ. En effet, le prix est évalué pour un instant donné, par exemple le 1^{er} mai 2014 à minuit, alors que la variable de la quantité de tickets vendus est une mesure. En effet, la quantité de tickets vendus est évaluée sur un intervalle, par exemple la quantité

vendue au cours du mois de mai 2014. La multiplication de ces deux variables est également une mesure. En effet, les recettes générées par la vente des tickets doivent être évaluées sur une période.

La quantité et le prix des billets sont saisis sur le passé qui correspond aux années 2013/2014 et 2014/2015.

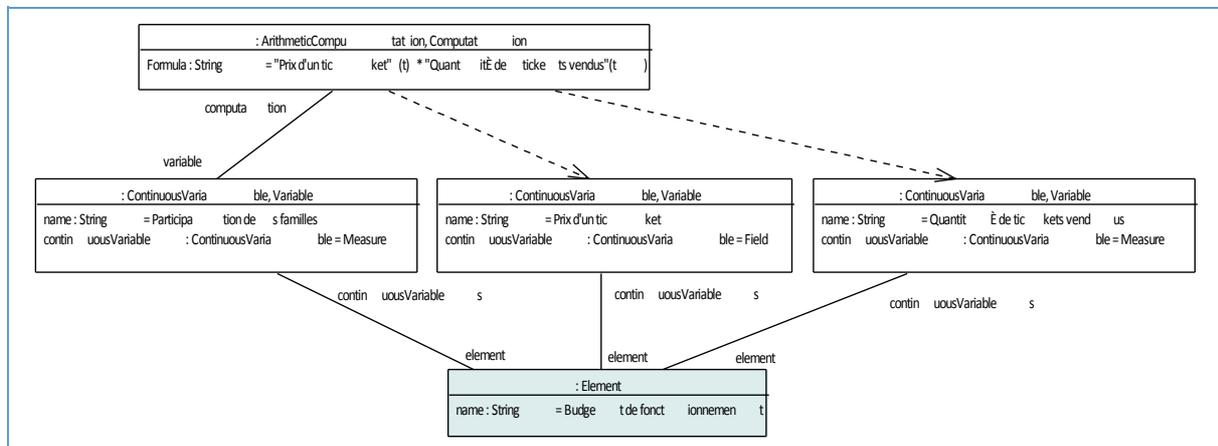


Figure 120 : Le diagramme d'instance montrant la modélisation du calcul de la participation des familles

5.2.1.5.1 La quantité et le prix des billets vendus

La quantité de billets vendus dans le futur est estimée en réalisant la moyenne de ces deux années. Cette moyenne permet d'atténuer le bruit auquel est sujet cette variable, la répartition des ventes étant légèrement aléatoire. Le prix pour le futur est un choix des élus. Par défaut, le prix est constant et conserve la valeur définie à la fin de la dernière année du passé. Les hausses de prix éventuellement prévus sont saisies pour remplacer cette valeur par défaut.

La variable de la quantité de tickets vendus est saisie par année avec un patron de répartition simple afin d'éviter de rendre l'exemple trop volumineux. Cependant, dans un cas réel, cette variable peut être saisie avec une périodicité plus fine afin de gagner en précision.

Il est considéré que certains parents achètent les tickets à l'avance. Ceci est représenté par une valeur plus élevée au début de l'année scolaire, comme ceci est visible sur la figure ci-dessous.

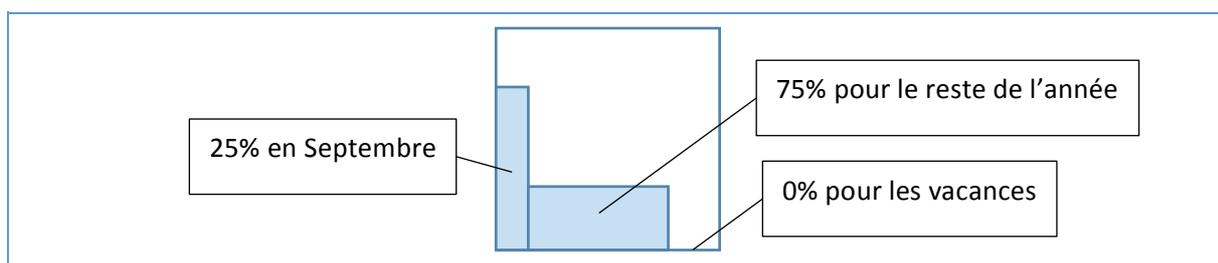


Figure 121 : Le patron de répartition des ventes de tickets au cours d'une année scolaire

Le patron contient trois valeurs constantes représentées par des instances de la classe « Constant ». Les bornes du patron correspondent à une année scolaire. La contrainte définie pour l'unité du patron est de type « Surface ». Ceci veut dire que lorsque ce patron est utilisé pour une saisie, sa forme est étirée de façon à ce que l'évaluation aux bornes de la saisie soit égale à la valeur saisie.

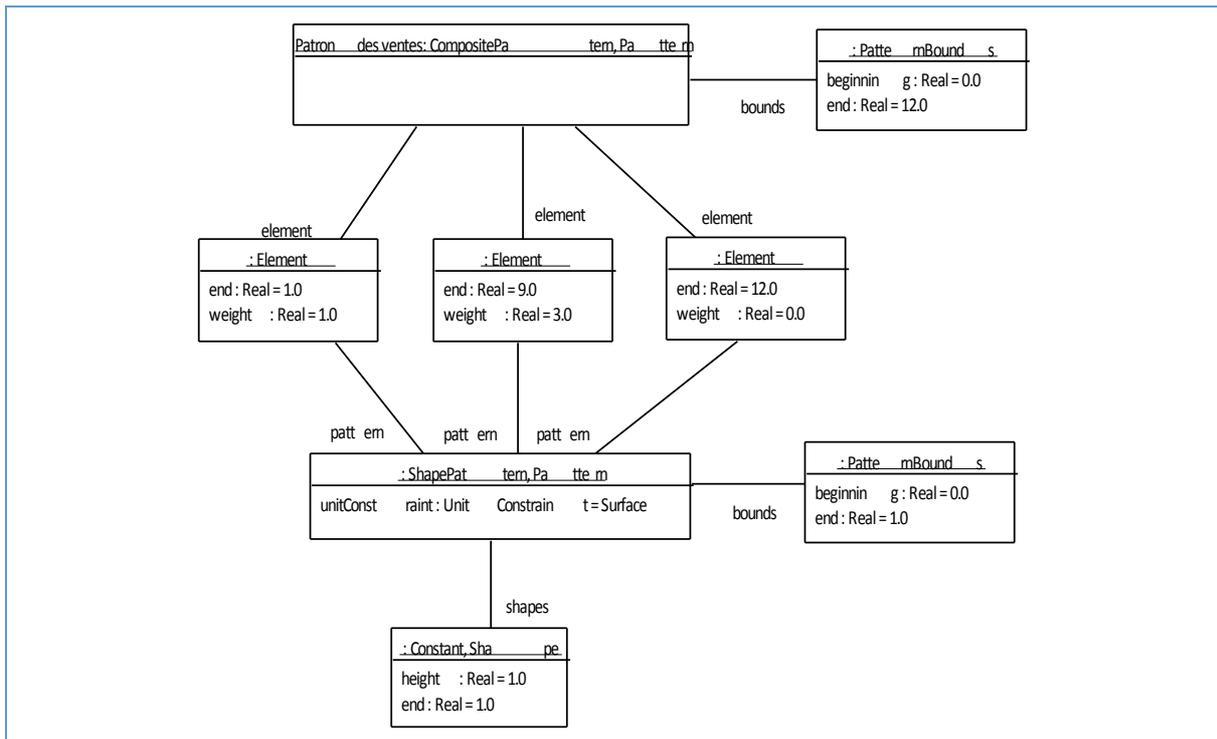


Figure 122 : Le diagramme de classes du patron de répartition des ventes de tickets au cours d'une année scolaire

50000 tickets ont été vendus au cours de l'année 2013/2014 et 50100 lors de l'année 2014/2015.

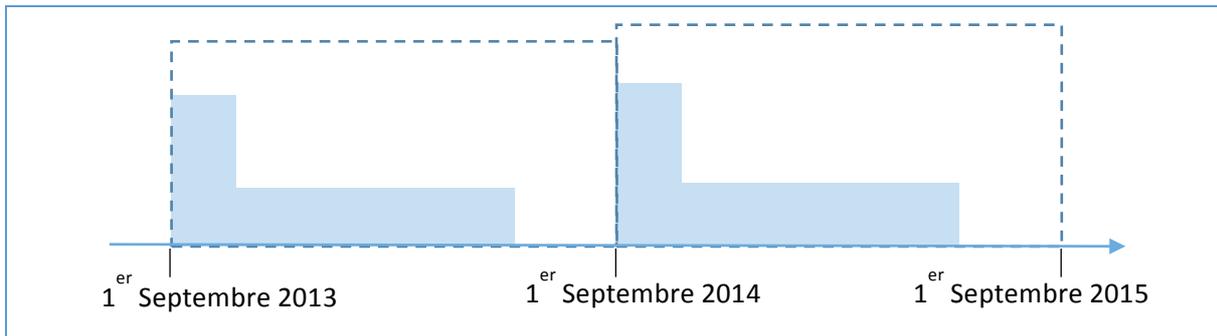


Figure 123 : La courbe d'évolution des ventes de tickets de 2014 à 2016

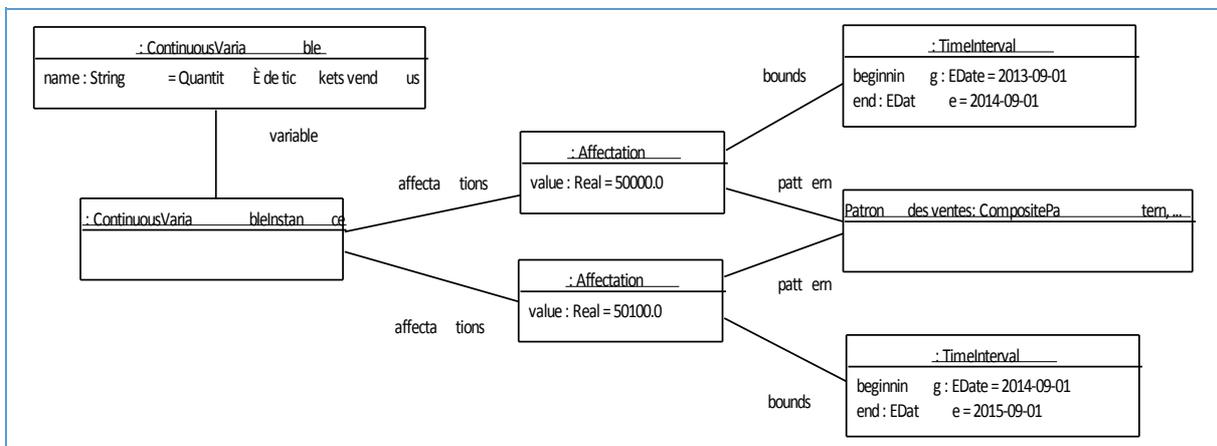


Figure 124 : Le diagramme d'instance montrant la saisie de l'évolution des ventes de tickets de 2014 à 2016

La quantité de tickets vendus pour les années futures est estimée en réalisant la moyenne des deux dernières années du passé.

```

"Quantité de tickets vendus"(t) :=
If(t < BeginningOf(Year(1), "now"()),
  0,
  Repeat
  (
    BeginningOf(Year(1), "now"()),
    Year(1),
    (Delay(Year(2), "Quantité de tickets vendus"(t))
    +Delay(Year(1), "Quantité de tickets vendus"(t))
    ) / 2
  )
)
  
```

Figure 125 : La formule de la variable « Quantité de tickets vendus » en modélisation continue

5.2.1.5.2 Le prix d'un billet

Le prix d'un billet est constant sur de larges périodes. Le patron est composé d'une seule valeur constante. La contrainte définie pour l'unité du patron est de type « MaxHeight ». Ceci veut dire que lorsque ce patron est utilisé pour une saisie, sa forme est étirée de façon à ce que l'évaluation au point culminant du patron soit égale à la valeur saisie.

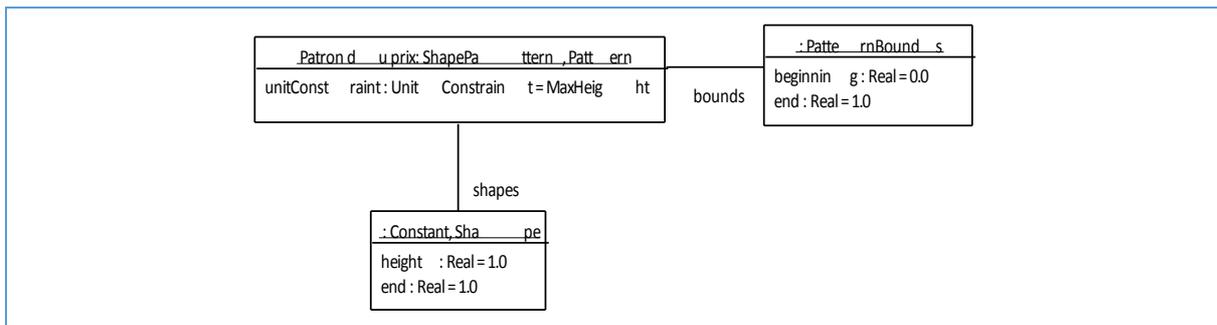


Figure 126 : Le diagramme de classes du patron de répartition des ventes de tickets au cours d'une année scolaire

Le prix des tickets présente une augmentation entre l'année 2013/2014 et 2014/2015. Il reste constant au cours de l'année.

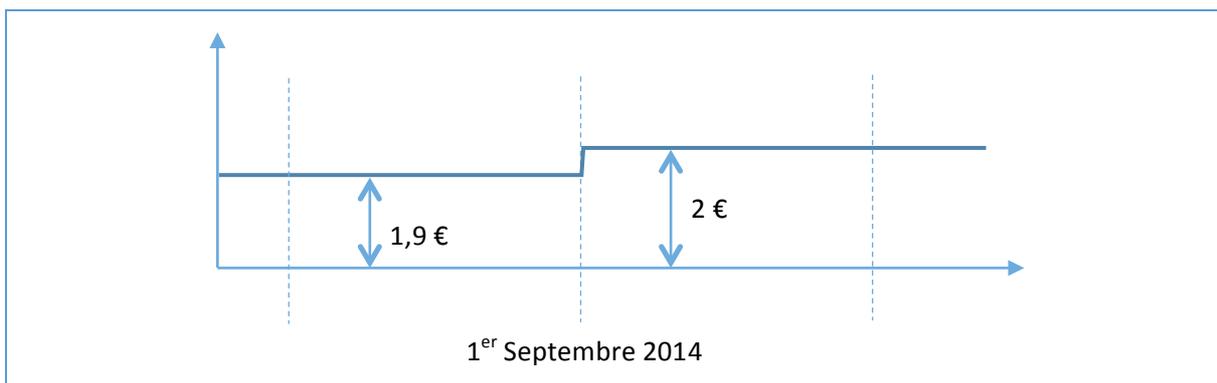


Figure 127 : La courbe d'évolution du prix d'un ticket de Septembre 2013 à Septembre 2015

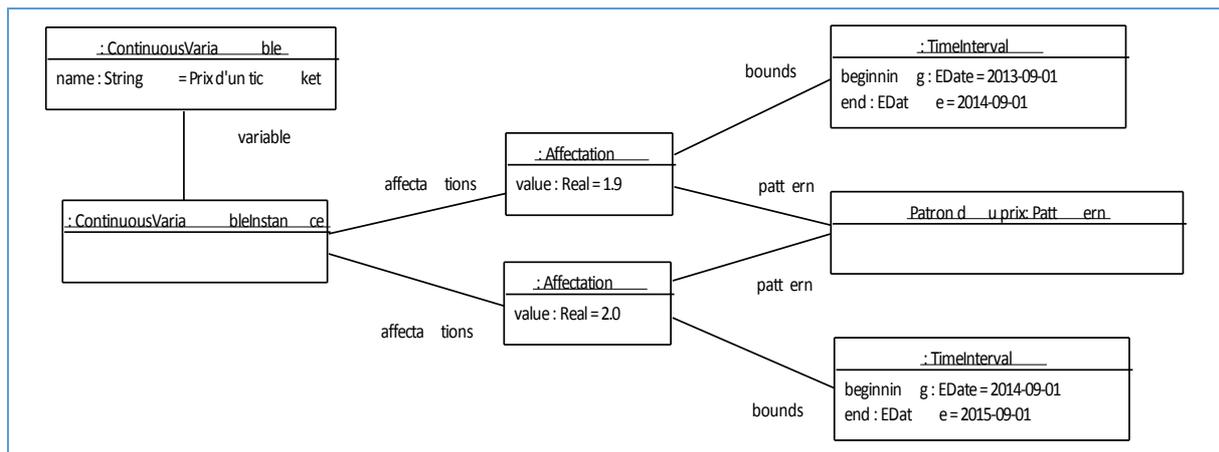


Figure 128 : Le diagramme d'instance montrant la saisie de l'évolution du prix d'un ticket de 2014 à 2016

5.2.1.6 Les dépenses en denrées

Les dépenses en denrées sont calculées par rapport au produit du coût d'un repas et du nombre de repas servis.

```
"Achats d'aliments"(t) := "Coût d'un repas"(t) * "Nombre de repas servis"(t)
```

Figure 129 : La formule de la variable « Achats d'aliments » en modélisation continue

5.2.1.6.1 Le coût en denrée d'un repas

La formule utilisée pour l'estimation des repas est celle dont la construction est détaillée dans le chapitre 894.13.5.

```
"Coût d'un repas"(t) :=
If(t < BeginningOf(Year(1), "now"()),
  0,
  Repeat
  (
    BeginningOf(Year(1), "now"() - Year(1)),
    Year(1),
    "Coût d'un repas"(t) / Exp(BeginningOf(Year(1), "now"()), 1.02, Year(1))
  )
  * Exp(BeginningOf(Year(1), "now"()), 1.02, Year(1))
)
```

Figure 130 : La formule de la variable « Coût d'un repas » en modélisation continue

5.2.1.6.2 Le nombre de repas servis

Le nombre de repas servis est saisi semaine par semaine. La moyenne des deux dernières années saisies sert d'estimation pour le futur de la même façon que dans le chapitre 5.2.1.5.1.

5.2.1.7 La participation à l'équilibre de la mairie

Le calcul de la participation de la mairie est défini afin d'équilibrer le budget de fonctionnement.

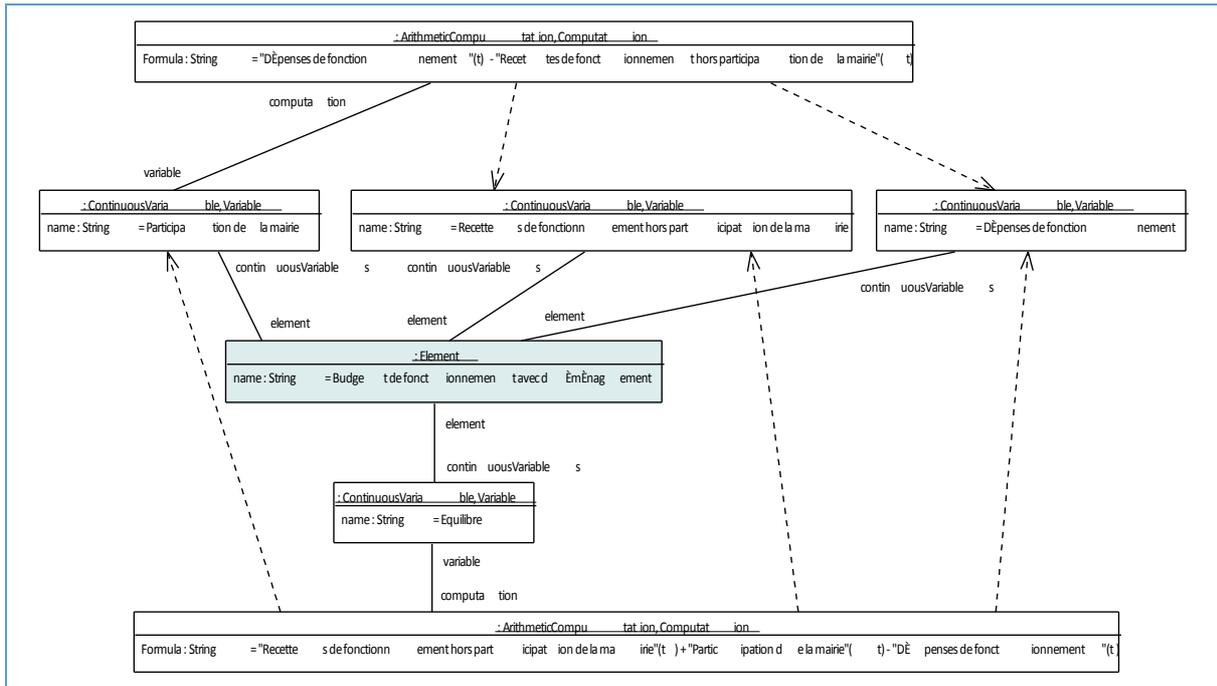


Figure 131 : Le diagramme d'instance montrant la modélisation du calcul de la participation à l'équilibre de la mairie

Une accumulation de la participation de la mairie est réalisée à partir du début du scénario afin de permettre la comparaison de la participation totale de la mairie jusqu'à une date donnée. Ce calcul est défini selon la formule ci-dessous.

"Participation de la mairie cumulée"(t) := Accumulate("Participation de la mairie", "Début du scénario"(t))

Figure 132 : La formule du début de l'année du présent

5.2.1.8 L'amortissement

Chaque opération de construction de la cantine est amortie individuellement à la fin de celle-ci. Le montant à amortir correspond aux dépenses totales de l'opération. La variable « Dépenses » est récupérée par l'intermédiaire d'un lien vers le modèle « Opération ».

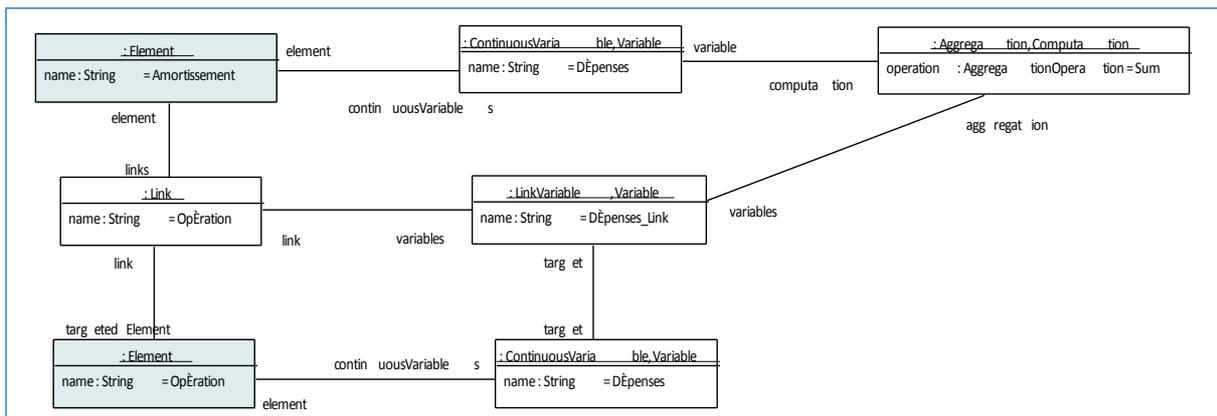


Figure 133 : Le diagramme d'instance montrant le calcul de la somme des dépenses d'une opération

Une deuxième étape est nécessaire afin d'obtenir le montant total qui est utilisé pour le calcul de l'amortissement. En effet, les dépenses sont réparties sur l'axe du temps en fonction du moment où elles ont eu lieu. L'opérateur « SumOver » est utilisé pour réaliser le total des dépenses entre le début et la fin du scénario et de le placer à la date du début de l'amortissement.

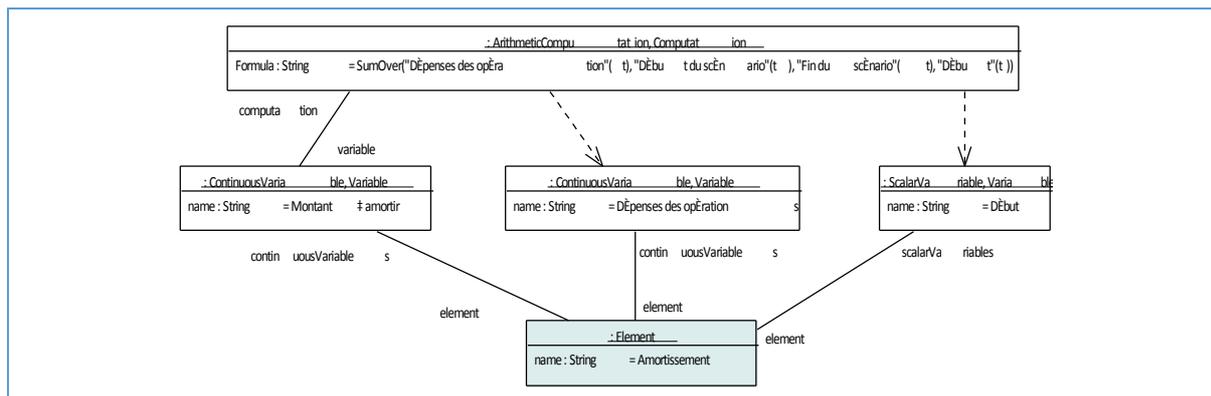


Figure 134 : Le diagramme d'instance montrant le calcul du montant à amortir

5.2.1.9 Le remboursement de la TVA

La formule utilisée pour le calcul du remboursement de la TVA est celle dont la construction est détaillée dans le chapitre 4.13.3

```
"Remboursement de la TVA"(t) :=
0.2 *
Delay
(
  Month(3) + Days(4),
  SumBy
  (
    '2015-01-01'
    Month(1),
    "Dépenses des opérations"(t)
  ),
)
```

Figure 135 : La formule de la variable « Remboursement de la TVA » en modélisation continue

5.2.2 L'utilisation du modèle du cas d'étude

5.2.2.1 Le choix de l'investissement de construction de la nouvelle cantine

Le conseil municipal souhaite comparer deux planifications financières afin de choisir entre deux possibilités de construction de la nouvelle cantine. Ces deux planifications ciblent objectif différent : l'une cible un faible investissement alors que l'autre cible une forte efficacité énergétique. L'objectif de la comparaison est de déterminer à partir de combien d'années l'investissement supplémentaire de la deuxième solution est compensé par les économies d'énergie.

Afin de réaliser une comparaison de ces deux solutions, deux modèles de planification budgétaire sont réalisés. La différence entre ces deux modèles se situe au niveau du coût et de la durée des opérations et des dépenses de consommation d'énergie.

Une recherche d'équilibre est réalisée sur chacun des deux modèles. L'emprunt doit être adapté en fonction des dépenses d'investissement afin d'équilibrer le budget d'investissement. Dans cet exemple, l'emprunt n'est pas déterminé automatiquement car ceci aurait complexifié le modèle. Les amortissements sont calculés automatiquement en fonction des dépenses d'opérations, ils ne nécessitent pas d'attention particulière.

La comparaison entre les deux planifications est réalisée au niveau de la variable de trésorerie du budget de fonctionnement qui exclut la participation de la mairie. Cette variable permet d'estimer la participation totale de la mairie jusqu'à une date donnée.

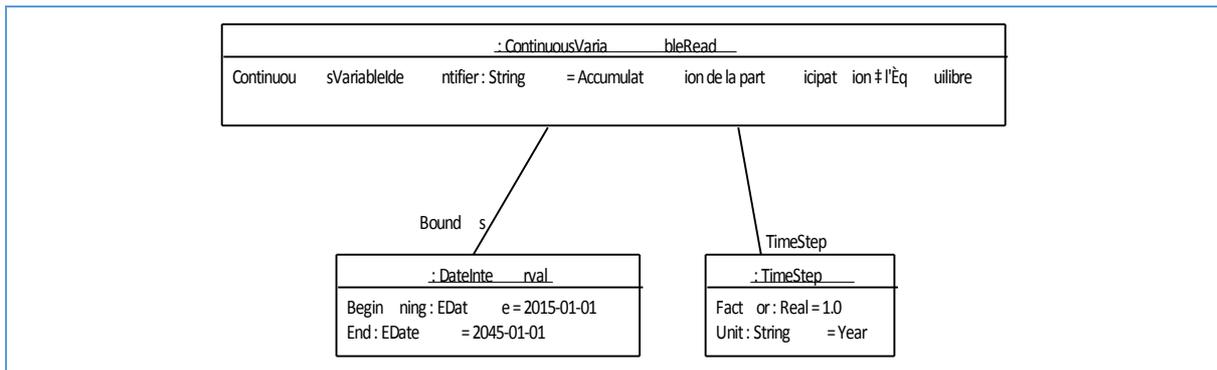


Figure 136 : Une requête de calcul évaluant la participation de la mairie

La figure ci-dessous présente la courbe de la différence de cette variable entre les deux modèles de planification. La courbe s'annule au cours de l'année 2022. La planification à forte efficacité énergétique devient donc plus intéressante environ 7 ans après le début de la construction. Le conseil municipal retient la solution avec une forte efficacité énergétique car elle est permise de réaliser des économies nettement avant la fin de vie du bâtiment.

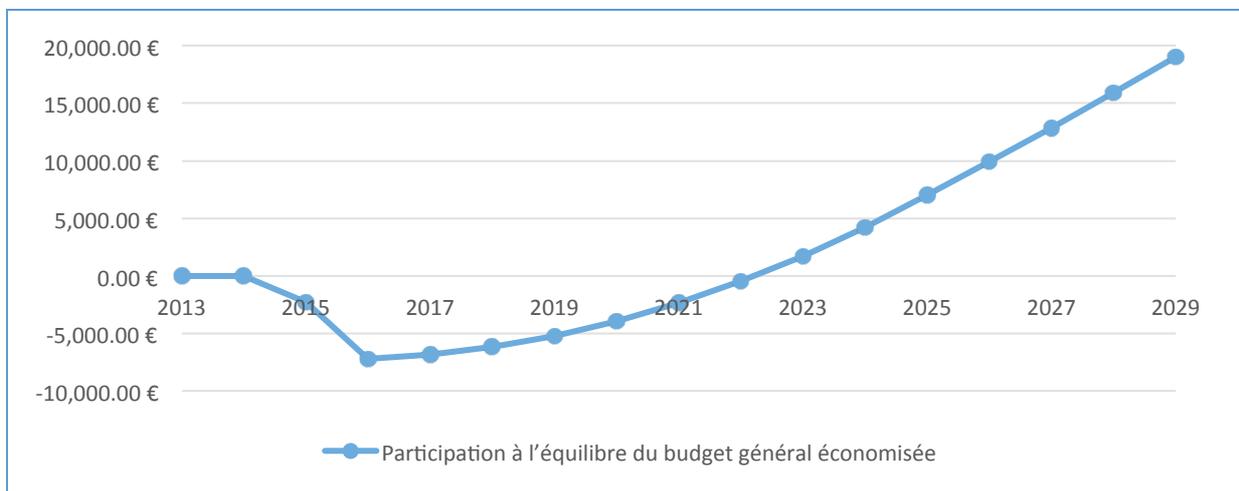


Figure 137 : Le cumul des économies réalisées par la commune avec la solution à haut rendement énergétique

Plusieurs requêtes sont réalisées sur la planification afin d'évaluer l'impact du prix des tickets sur la participation de la mairie.

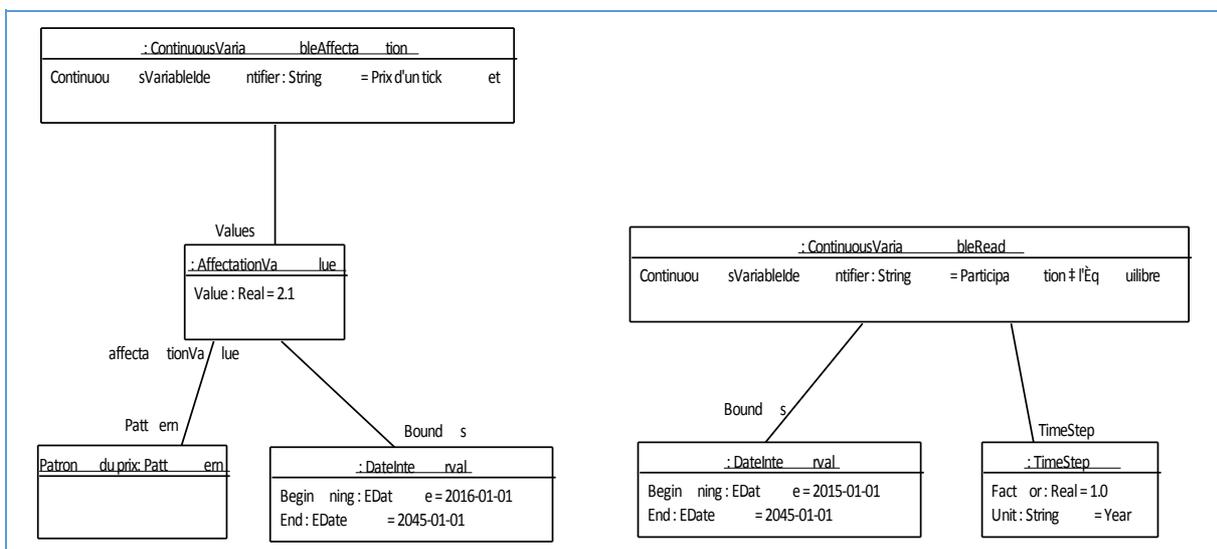


Figure 138 : Une requête de calcul évaluant l'impact du prix des tickets sur la participation de la mairie

Après plusieurs requêtes, le conseil municipal décide de conserver le prix des tickets à 2.00€.

5.2.2.2 La reprogrammation pour prendre en compte le retard dans la construction du bâtiment pour cause d'intempéries

L'opération de maçonnerie subit un retard de 15 jours pour cause d'intempéries. Le modèle est adapté pour prendre en compte ce retard en utilisant le service de modélisation.

```
void PostScenario
(
    string scenarioURL = "https://www.client.fr/Cantine_2015-01-14.xml"
    string scenarioIdentifiant = "Cantine 2015-01-14"
)
```

Figure 139 : Une requête d'importation du fichier XML du modèle

Le dernier paiement pour cette opération est décalé ainsi que ceux des opérations suivantes. Les valeurs saisies sur les variables « Dépenses » des instances « Opération d'isolation » et « Opération de chauffage » sont décalées.

```
void ShiftContinuousAffectation
(
    string scenarioIdentifiant = "Cantine 2015-02-23",
    string variableInstanceIdentifiant = "ModelInstances(Cantine)/ModelInstances(Opération d'isolation)
                                        /Variables(Dépenses)",
    string beginning = "2016-01-01",
    string end = "2016-04-01",
    string timeStepFormula = "Day(15)"
)
```

Figure 140 : Une opération de modélisation pour actualiser les valeurs affectées aux dépenses

Leurs variables « Début » et « Fin » sont saisies à nouveau pour correspondre à la nouvelle planification physique.

```
void PostScalarAffectation
(
    string scenarioIdentifiant = "Cantine 2015-01-14",
    string variableInstanceIdentifiant = "ModelInstances(Cantine)/ModelInstances(Opération de
maçonnerie)/Variables(Fin)",
    string valueFormula = "'2016-01-15'"
)
```

Figure 141 : Une opération de modélisation pour actualiser la date de la fin d'une opération

L'ouverture de la nouvelle cantine est également décalée de 15 jours en saisissant la variable « Déménagement ».

```
void PostScalarAffectation
(
    string scenarioIdentifiant = "Cantine 2015-01-14",
    string variableInstanceIdentifiant = "ModelInstances(Cantine)/Variables(Déménagement)",
    string valueFormula = "'2016-04-14'"
)
```

Figure 142 : Une opération de modélisation pour actualiser la date de déménagement

Le nouveau modèle est exporté en XML pour être utilisé avec le service de calcul.

```
string GetScenarioURL
(
    string scenarioIdentifiant = "Cantine 2015-01-14",
) = "https://www.modelisateur.fr/export-64d7a/Cantine_2015-01-14.xml"
```

Figure 143 : Une requête d'exportation du modèle en fichier XML

Le service des finances recherche un équilibre en s'appuyant sur le service de calcul pour explorer plusieurs possibilités. Le service des finances soumet une reprogrammation du budget qui rétablit un équilibre en tenant compte du retard de construction en procédant de la même façon que dans le chapitre 5.2.2.1.

Les opérations de modification contenues dans la requête de calcul retenue sont appliquées définitivement au modèle en utilisant le service de modélisation. Ce nouveau modèle est celui qui servira de base pour la suite du suivi.

5.2.2.3 La reprogrammation pour prendre en compte le retard du versement de la subvention

Le deuxième versement de la subvention de la construction de la cantine par le Conseil Départemental qui devait avoir lieu en fin d'année n'est pas reçu à la date prévue. Le service financier estime que ce versement devrait avoir lieu le mois suivant. Le service des finances négocie une ligne de trésorerie avec la banque. Le modèle est modifié pour tenir compte de cet imprévu.

La saisie de la variable « Subvention d'investissement » est actualisée pour prendre en compte le retard prévu d'un mois.

```
void ShiftContinuousAffectation
(
  string scenarioIdentifiant = "Cantine 2015-02-23",
  string variableInstanceIdentifiant = "ModelInstances(Cantine)/ModelInstances(Investissement)/Variables(
  Subventions d'investissement)",
  string beginning = "2015-12-13",
  string end = "2015-12-14",
  string timeStepFormula = "Month(1)"
)
```

Figure 144 : Une opération de modélisation pour actualiser les valeurs affectées à la subvention d'investissement

Une instance du modèle « Emprunt » est créée et associée avec le budget d'investissement.

```
string PostModelInstance
(
  string scenarioIdentifiant = "Cantine 2015-02-23",
  string parentModelInstanceIdentifiant = "ModelInstances(Cantine)"
  string modelIdentifiant = "Models(Cantine)/Models(Emprunt)",
  string modelInstanceName = "Emprunt pour le retard de la subvention"
) = "ModelInstances(Cantine)/ModelInstances(Emprunt pour le retard de la subvention)"
```

Figure 145 : Une opération de modélisation pour instancier un emprunt

```
void PostAssociation
(
  string scenarioIdentifiant = "Cantine 2015-02-23",
  string originModelInstanceIdentifiant = "ModelInstances(Cantine)/ModelInstances(Investissement)/Variables(
  Subventions d'investissement)",
  string linkIdentifiant = "Models(Cantine)/Models(Investissement)/Link(Emprunts)",
  string targetModelInstanceIdentifiant = "ModelInstances(Cantine)/ModelInstances(Emprunt pour le retard de la
  subvention)"
)
```

Figure 146 : Une opération de modélisation pour associer un emprunt au budget d'investissement

La variable du montant à emprunter est saisie de façon à compenser le versement qui n'a pas eu lieu.

```

void PostContinuousAffection
(
  string scenarioIdentifiant = "Cantine 2015-02-23",
  string variableInstanceIdentifiant = "ModelInstances(Cantine)/ModelInstances(Emprunt pour le retard de la
  subvention)/Variables(Capital emprunté)",
  IEnumerable<XmlAffectionValue> affectationValues
)

```

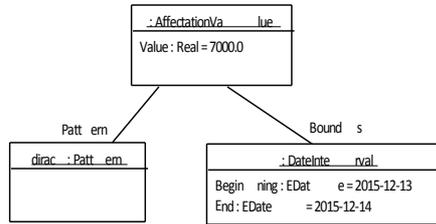


Figure 147 : Une opération de modélisation pour définir la valeur du capital emprunté

L'emprunt est remboursé le mois suivant à la nouvelle date prévue pour le versement de la subvention. Comme l'emprunt est remboursé d'un coup, le patron de remboursement correspondant est un Dirac.

L'équilibre du budget d'investissement est conservé du fait que les modifications apportées aux variables « Subvention », « Montant de l'emprunt » et « Remboursement de capital » se compensent. Cependant l'emprunt génère des intérêts à rembourser. Le budget de fonctionnement doit être équilibré afin de compenser cette nouvelle dépense. Le modèle est donc exporté du service de modélisation afin d'être utilisé avec le service de calcul pour servir de support aux décisions prises pour rétablir l'équilibre.

Afin de rétablir l'équilibre du budget de fonctionnement, le service des finances explore plusieurs possibilités en s'appuyant sur le service de calcul et finit par décider de réaliser des économies sur les repas servis au cours du mois et de compléter par une participation supplémentaire accordée par la mairie.

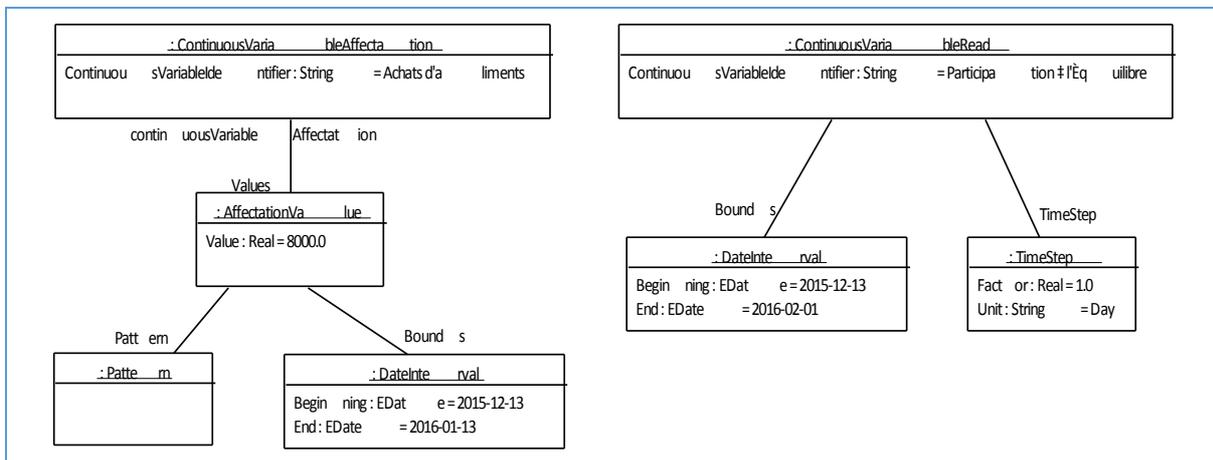


Figure 148 : Une requête de calcul évaluant l'impact des décisions sur l'équilibre du budget de fonctionnement

Les opérations de modification contenues dans la requête de calcul retenue sont appliquées définitivement au modèle en utilisant le service de modélisation. Ce nouveau modèle est celui qui servira de base pour la suite du suivi.

```

void PostContinuousAffection
(
  string scenarioIdentifiant = "Cantine 2015-02-23",
  string variableInstanceIdentifiant = "ModelInstances(Cantine)/ModelInstances(Fonctionnement avant)/Variables(
  Aliments)",
  IEnumerable<XmlAffectionValue> affectationValues
)

```

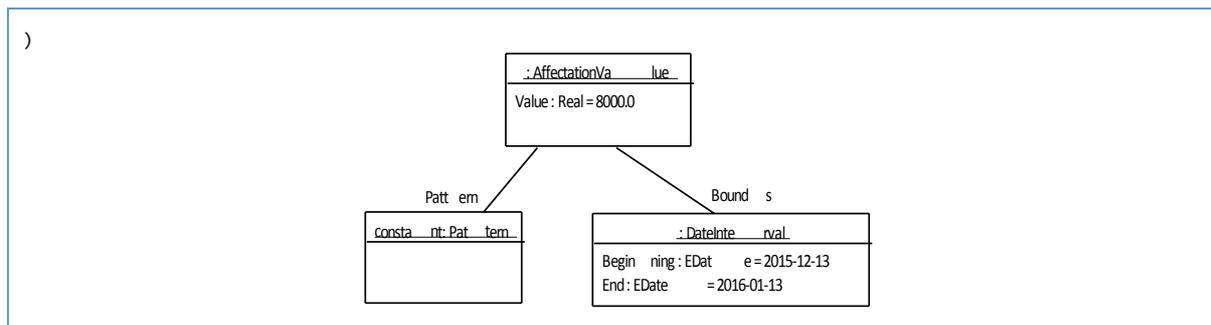


Figure 149 : Une opération de modélisation pour modifier la valeur des dépenses en achats d'aliments

5.2.3 La mesure de complexité de l'évaluation du cas d'étude

Le chapitre 4.6 explique comment le calcul continu permet d'évaluer un modèle unique sur des échelles différentes sans que le rapport entre les bornes du modèle et l'intervalle d'évaluation n'ait une influence néfaste sur les temps de réponse.

La mesure de la complexité est évaluée sur le modèle du cas d'étude. La variable « Accumulation de la participation à l'équilibre du budget général » est choisie pour réaliser les mesures car ceci permet de se placer dans une situation où l'évaluation dépend des valeurs précédentes. Comme ceci est exposé dans le chapitre 4.6, dans le cas de SOFI discret, c'est une situation dans laquelle le temps d'évaluation est proportionnel à ce rapport.

Les mesures du temps d'évaluation sont prises en faisant varier la taille de l'intervalle d'évaluation de la variable choisie. Les résultats sont identiques pour un intervalle d'un jour, d'une semaine, d'un mois et d'un an à 1% près.

Les mesures confirment l'hypothèse que l'évaluation d'une variable ne dépend pas de la taille de l'intervalle d'évaluation choisi. En effet, le temps de calcul est dépendant de la précision demandée à la bibliothèque de calcul. Il serait intéressant d'étudier l'évolution de la précision des calculs par rapport aux temps de réponse, mais ceci n'apporterait pas d'élément à cette validation.

5.2.4 La réponse aux questions de recherche

Comment formaliser une modélisation de planification financière applicable à des domaines variés ?

Comme cela a été vu dans le chapitre 4, le métamodèle défini reprend les fondements posés par un métamodèle de calcul discret existant en proposant pour chaque concept discret un raffinement en modélisation continue. Le métamodèle de calcul discret existant permettait déjà de définir la logique propre à chaque domaine d'application de la planification financière, le métamodèle proposé permet de le faire avec une modélisation continue.

Comment évaluer un modèle d'application de la planification financière sur plusieurs échelles de temps ?

La comparaison entre l'approche de modélisation continue et celle discrète a permis de montrer comment le métamodèle proposé permet de décrire les valeurs et les formules de calculs en s'abstrayant de la périodicité choisie pour l'évaluation de celui-ci. L'implémentation du prototype de service de calcul a permis de valider que les valeurs obtenues pour l'évaluation d'un unique modèle sur plusieurs échelles de temps sont correctes.

Comment permettre une intégration du suivi de la planification financière au système d'information ?

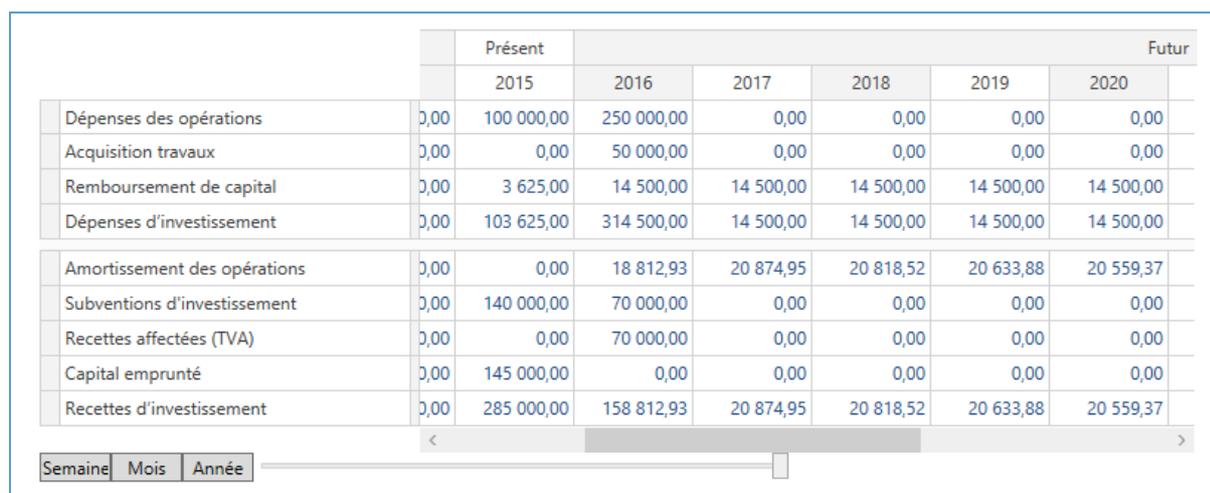
Pour permettre une intégration du suivi de la planification financière à un système d'information, le prototype a été développé avec une architecture orientée service. L'approche proposée pour cette architecture permet une parallélisation afin d'être en capacité de traiter une grande quantité de requêtes.

6 Conclusion et perspectives

6.1 Le bilan

6.1.1 Complétion de la thèse

Du point de vue technique, la thèse a abouti à la réalisation d'une nouvelle version du logiciel SOFI, basée sur un moteur de calcul en temps continu. L'objectif principal de validation de la technologie est donc atteint, dans le sens où l'interface graphique proposée reprend les fondamentaux de SOFI et que l'utilisateur ne sait donc pas, s'il reste sur des périodicités de rendu standards, que le moteur de calcul a changé.



		Présent					Futur	
		2015	2016	2017	2018	2019	2020	
Dépenses des opérations	0,00	100 000,00	250 000,00	0,00	0,00	0,00	0,00	
Acquisition travaux	0,00	0,00	50 000,00	0,00	0,00	0,00	0,00	
Remboursement de capital	0,00	3 625,00	14 500,00	14 500,00	14 500,00	14 500,00	14 500,00	
Dépenses d'investissement	0,00	103 625,00	314 500,00	14 500,00	14 500,00	14 500,00	14 500,00	
Amortissement des opérations	0,00	0,00	18 812,93	20 874,95	20 818,52	20 633,88	20 559,37	
Subventions d'investissement	0,00	140 000,00	70 000,00	0,00	0,00	0,00	0,00	
Recettes affectées (TVA)	0,00	0,00	70 000,00	0,00	0,00	0,00	0,00	
Capital emprunté	0,00	145 000,00	0,00	0,00	0,00	0,00	0,00	
Recettes d'investissement	0,00	285 000,00	158 812,93	20 874,95	20 818,52	20 633,88	20 559,37	

Figure 150 : Une capture d'écran du prototype d'interface graphique

A l'inverse, s'il souhaite se positionner en étude discrète d'un modèle sur des périodicités inconnues auparavant, les résultats sont disponibles et ont été validés par les experts financiers de MGDIS comme corrects. Ainsi, une analyse sur des demi-trimestres est désormais possible. De même, la vision du modèle du point de vue de la trésorerie, avec une valeur pour chaque jour est désormais disponible, et ce sur le même modèle mathématique que les gestionnaires financiers travaillant sur des années, ce qui est une richesse fonctionnelle considérable. Auparavant, la discrétisation au jour d'un modèle s'étalant sur vingt ans était impossible pour des raisons de performance. La nouvelle version de SOFI ne voit pas ses performances varier significativement en fonction du niveau de détail dans lequel on observe le modèle. Il s'agit sans conteste d'un point très positif sur le résultat de la présente thèse, car tel était le premier objectif fixé par MGDIS.

Un second objectif de thèse a été atteint dans le sens où une cartographie de services indépendants a été fournie, avec des communications standardisées entre les différents services. Ceci correspond à une urbanisation correcte de l'application SOFI, auparavant fournie sous forme d'un monolithe. Bien que les applications de la séparation des responsabilités pour améliorer les performances n'aient pas été testées, le découpage a pour sa part été validé par la réalisation d'une version applicative correcte du point de vue des résultats.

6.1.2 Validation des résultats

MGDIS a une grande expérience en matière de tests automatisés, et la validation de la thèse a dès le début été posée en termes liés à l'assurance qualité. En particulier, des tests automatisés de migration des modèles ont été mis en place de façon à vérifier que les montants calculés par le nouveau moteur étaient proches des montants simulés par l'ancien. Bien sûr, les montants ne pouvaient pas être exactement les mêmes, vu que le modèle discret est déjà lui-même une modélisation imparfaite de la réalité. Mais les erreurs étaient systématiquement inférieures à celles

acceptées pour la modélisation elle-même. Ces erreurs se sont d'ailleurs révélées provenir d'une modélisation plus fidèle de la réalité que celle du modèle discret.

6.1.3 Adéquation à la problématique

Au-delà de la validation des résultats eux-mêmes, il importe de vérifier que le logiciel sera adapté à son usage final. La version du logiciel fournie à l'issue de la thèse n'est pas passée dans le processus de validation complet de MGDIS, et ne peut donc pas être considérée comme une version officielle du logiciel. Il convient toutefois de noter que ceci n'était pas une demande de MGDIS. Le sujet de la thèse était de valider avec un prototype l'adéquation de la modélisation continue pour de la programmation physico-financière.

Ce sujet a été couvert, et ce document servira de base pour une mise en œuvre industrielle. La thèse a été validée par MGDIS comme suffisamment détaillée pour qu'une équipe de développement conçoive les nouvelles interfaces, les fonctionnalités de persistance et les expositions techniques de services nécessaires pour une mise en œuvre industrielle du prototype.

Entretemps, la validation de l'adéquation à la problématique a été réalisée par le responsable de thèse côté entreprise. Elle a consisté principalement en un audit du code réalisé ainsi qu'en une validation des APIs disponibles pour les différents services et de leur performance en utilisation réaliste, sur le modèle qui a servi d'exemple. Cette validation, menée par l'entreprise porteuse du projet, permet de valider du point de vue industriel que les objectifs initiaux fixés pour la thèse ont été atteints.

6.2 Le positionnement : contributions à l'état de l'art

L'état de l'art a montré qu'il existe des travaux concernant la modélisation mathématique avec des fonctions continues et d'autres concernant l'aide à la décision dans le domaine de la planification physico-financière, mais aucun article concernant les deux sujets simultanément n'a été trouvé. Ce document apporte une approche pour la réalisation d'un logiciel d'aide à la décision dans le domaine de la planification physico-financière qui s'appuie sur une modélisation mathématique avec des fonctions continues.

6.3 Les perspectives

La conclusion technique ayant abouti à la validation du travail réalisé, il devient possible d'explorer les différentes voies évolutives rendues possibles par l'aboutissement de la première étape portée par la présente thèse. Le remplacement du moteur de calcul discret de SOFI par un modèle basé sur des services et une modélisation continue du temps était en effet un moyen d'obtenir un premier résultat (à savoir la performance et la cohérence sur des périodicités d'étude hétérogènes) mais aussi un prérequis pour des objectifs plus lointains d'optimisation des modèles de programmation physico-financière et d'appropriation plus efficaces par les utilisateurs.

6.3.1 L'ergonomie de l'interface graphique pour modélisation continue native

Le sujet de la présente thèse était de changer le moteur de SOFI. Ceci étant réalisé, il conviendrait dans un second temps de changer les commandes du logiciel, c'est-à-dire son interface graphique. En effet, la modélisation de la programmation physico-financière implique désormais que la personne en charge de la conception des modèles puisse spécifier si une variable a une répartition continue dans le temps ou si elle suit des pics, voire des formes particulières.

Typiquement, une dépense de salaire doit être payée à un moment donné à la banque et il s'agit donc d'une valeur considérée comme un Dirac basé sur la fin du mois. A l'inverse, une modélisation des coûts des denrées sera plutôt sous la forme d'une fonction constante en première

approximation, car la cantine par exemple consomme tous les jours, voire même se fait livrer plusieurs fois par jour par des fournisseurs différents. Cette distinction évidente pour un mathématicien ne l'est aujourd'hui pas pour un financier habitué depuis des années à modéliser de manière discrète.

Un prochain travail de fond sur SOFI sera de se pencher sur le module de conception des modèles financiers et de donner les outils d'interface nécessaires pour que la personne en charge de la modélisation puisse faire les bons choix. On pourrait par exemple aussi imaginer des bibliothèques de répartitions des dépenses dans le temps pour des projets types donnés.

Ce travail pourrait éventuellement faire à terme l'objet d'une thèse dédiée, mais MGDIS a l'intention dans un premier temps de le faire réaliser par son équipe de développeurs.

6.3.2 La parallélisation et branchement sur optimisation par algorithmes génétiques

MGDIS dispose d'une expérience approfondie sur l'optimisation par algorithmes génétiques. Une première thèse sur le sujet n'a pas abouti, mais a tout de même donné naissance à un article publié dans une revue scientifique et établissant les conditions de faisabilité d'une optimisation d'un modèle de programmation physico-financière par ces types d'algorithmes. En particulier, l'idée d'accélération du processus par la restriction à un cube multidimensionnel a été écartée, ce qui permet de lever une hypothèse importante.

Le fait de disposer d'un moteur de calcul en modélisation continu utilisable comme un service autonome va permettre de relancer fortement cette approche d'optimisation. En effet, une des grandes difficultés sur la version existante de SOFI pour l'optimisation était qu'il était compliqué de distribuer les calculs. Les algorithmes génétiques avaient justement été adoptés parce qu'ils permettaient de découper les opérations d'optimisation en séquences complètement autonomes, chaque solution testée pouvant être évaluée indépendamment des autres. Or, c'est une difficulté de couplage technique de SOFI qui rendait ceci malaisé, car il fallait alors dupliquer la totalité du logiciel, y compris la base de données, pour paralléliser les traitements. Ceci a entraîné l'arrêt temporaire du projet.

La mise au point du moteur de calcul continu disponible comme un service permet de reposer la distribution du calcul d'optimisation sur un nombre quelconque de machines, chacune recevant simplement une instance du moteur et une description XML du modèle financier à calculer. Le parallélisme et l'utilisation des ressources sont alors excellents. De plus, l'utilisation de fonctions continues rend beaucoup plus simple et rapide l'estimation des résultats, qui n'ont même pas besoin d'être discrétisés pour que la fonction d'évaluation de la qualité de la solution simulée soit calculée.

Ce type de développement nécessitera toutefois une activité importante de recherche en amont. Il y aura donc au minimum un projet innovant pour le porter, et potentiellement une autre thèse CIFRE. MGDIS lancera l'un ou l'autre en fonction de ses priorités d'investissement et de l'intérêt de ses clients pour ce type d'approche, qui reste à valider.

6.3.3 L'optimisation par inversion du modèle continu

Nous avons rapidement cité dans le chapitre sur la mise en œuvre de la solution (section sur les emprunts et amortissements) la possibilité d'inverser le modèle mathématique. Bien que ce ne soit pas le premier but recherché dans la modélisation continu, il s'agit indubitablement d'un effet de bord extrêmement positif, car il ouvre à nouveau la voie à une optimisation mathématique du modèle.

Comme nous l'avons expliqué dans le chapitre introductif, des premières approches ont été tentées par MGDIS il y a une dizaine d'années avec un moteur de calcul sous contraintes pour optimiser un modèle de programmation physico-financière. Cette solution, bien que fonctionnelle sur des modèles très simples, s'effondrait en matière de temps de traitement sur des modèles un peu plus importants. Les modèles traités par certains clients de MGDIS étant parfois énormes, l'approche a été abandonnée et des méthodes d'optimisation par évolution contrôlée des solutions ont été explorées, en particulier les algorithmes génétiques. Dans toutes ces approches comme les traitements de type Monte Carlo, un des avantages est que le modèle est calculé dans le sens le plus court, avec des entrées qui, couplées aux règles mathématiques de définition financières, produisent des sorties. Il n'y a pas bijection car les calculs sont discrets, comportent des cumuls et au final plusieurs étapes non continues.

Le moteur de calcul de SOFI étant désormais complètement basé sur des fonctions continues, se pose désormais la question de revenir à une méthode d'optimisation basée sur l'inversion des calculs. En théorie, ceci est possible de deux manières. La première consiste en une inversion purement mathématique des fonctions. La complexité des modèles fait toutefois craindre une explosion exponentielle de la complexité des fonctions inversées. La seconde, basée sur une approximation des fonctions par des matrices, semble plus prometteuse. En effet, il devient possible de jouer sur le niveau de décomposition et donc de garder la main sur le rapport entre la précision et la performance, ce qui est un critère industriel important. De plus, de nombreux outils de calcul algébrique et d'inversion de matrices existent sur le marché, et en particulier certains qui permettent de tirer parti des performances exceptionnelles de calcul en virgule flottante qu'apportent les GPU et GPGPU. Une première expérience de MGDIS sur ce sujet fait que cette piste semble la plus intéressante à ce jour, et ouverte à une activité de recherche, potentiellement une autre thèse CIFRE. Elle est toutefois uniquement citée ici à titre de perspective, ce projet n'étant pas encore lancé.

6.3.4 La détermination en temps réel de la robustesse du modèle financier

Une dernière perspective d'utilisation du nouveau moteur de calcul de SOFI pourrait être dans la mise en œuvre d'un accompagnement du gestionnaire financier sur la robustesse de sa simulation. Lors de la recherche de valeurs correctes pour le modèle physico-financier, les gestionnaires recherchent avant tout des solutions qui permettent de réaliser le plus de projets possible tout en maintenant les ratios prudentiels d'endettement à un niveau souhaitable.

Cette activité est déjà complexe et nécessite un certain flair, mais elle pourrait être effectivement assistée par un moteur d'optimisation comme nous en avons parlé un peu plus haut. Toutefois, une fois un ensemble de valeurs correctes trouvé, le gestionnaire financier n'est pas au bout de son travail, car il doit encore évaluer la résistance de son modèle à des aléas. Que se passe-t-il par exemple en cas d'aléa sur les travaux de la cantine ? Le modèle choisi comme idéal vole-t-il en éclat, avec une dégradation brusque de tous les indicateurs ou rend-il possible de compenser les dépenses par un autre poste d'emprunt ou de recette sans mettre en difficulté l'entité financière ?

C'est tout le rôle du calcul de robustesse du modèle que d'assister le financier en lui préparant des calculs de simulation de tel ou tel aléa. Nous pourrions par exemple imaginer que chaque valeur simulée pour le modèle soit accompagnée d'un résultat en temps réel sur l'impact d'une baisse de 5% de la population ou d'une baisse drastique des dotations aux collectivités par l'Etat. Disposer de ce genre de mesures en temps réel aiderait fortement les gestionnaires à choisir une solution peut-être un peu moins prometteuse sur le nombre de projets financés, mais avec une bien plus grande résistance aux événements extérieurs. A nouveau, le moteur de calcul de SOFI en mode continu ouvre la porte à ce genre de réalisation, qui pourrait être portée sous la forme d'une thèse plutôt

orientée cette fois sur le métier de la finance. Une rapide veille montre en effet qu'aucun logiciel à ce jour n'offre d'indicateur de robustesse sur une solution financière proposée.

Enfin, l'interaction entre un mécanisme d'optimisation rapide et la détermination du niveau de robustesse d'une solution en temps réel serait extrêmement riche, car la robustesse pourrait participer à la détermination de la meilleure solution. La totalité des besoins imaginés à ce jour par MGDIS et ses clients serait alors couverte.

Références

- [Baati and al. 2007] Baati, Lassaad « Approche de modélisation DEVS à structure hiérarchique et dynamique » LSIS UMR-CNRS 6168, 2007.
- Baker, David, et Wendy Evans, éd. « 1 - Digital economics: introduction and overview. » In *A Handbook of Digital Library Economics*, 1^{er}21. Chandos Publishing, 2013.
- Baugh, P, A Gillies, et P Jastrzebski. « Combining knowledge-based and database technology in a tool for business planning. » *Information and Software Technology* 35, n° 3 (mars 1993): 131-137. doi:10.1016/0950-5849(93)90050-D.
- Bermúdez, J.D., J.V. Segura, et E. Vercher. « A decision support system methodology for forecasting of time series based on soft computing. » *The Fuzzy Approach to Statistical Analysis* 51, n° 1 (1 novembre 2006): 177-191. doi:10.1016/j.csda.2006.02.010.
- Broeckx, F. « Simulation in business planning and decision making: Thomas H. NAYLOR (Ed.) Volume 9, Number 1 in: Simulation Proceedings Series, Simulation Councils Inc., La Jolla, 1981, ix + 131 pages, \$30.00. » *European Journal of Operational Research* 12, n° 3 (mars 1983): 317. doi:10.1016/0377-2217(83)90206-0.
- Cha, Sang K., Kunsoo Park, Changbin Song, Kihong Kim, Cheol Ryu, et Sunho Lee. « Interval Disaggregate: A New Operator for Business Planning. » *Proc. VLDB Endow.* 7, n° 13 (août 2014): 1381-1392. doi:10.14778/2733004.2733011.
- Choi, Sang Hyun, Byung Seok Ahn, Chang Hee Han, Soung Hie Kim, et Jae Kyeong Kim. « Knowledge-based decision system for goal directed military resource planning. » *Computers & Industrial Engineering* 35, n° 1-2 (octobre 1998): 299-302. doi:10.1016/S0360-8352(98)00079-5.
- Cho, Kwun Ik, et Soung Hie Kim. « An improved interactive hybrid method for the linear multi-objective knapsack problem. » *Computers & Operations Research* 24, n° 11 (novembre 1997): 991-1003. doi:10.1016/S0305-0548(97)00021-X.
- Eisenreich, Katrin. « Towards an Algebraic Foundation for Business Planning. » In *Proceedings of the 2009 EDBT/ICDT Workshops*, 161-169. EDBT/ICDT '09. New York, NY, USA: ACM, 2009. doi:10.1145/1698790.1698817.
- Hruschka, Harald. « Use of fuzzy relations in rule-based decision support systems for business planning problems. » *European Journal of Operational Research* 34, n° 3 (mars 1988): 326-335. doi:10.1016/0377-2217(88)90153-1.
- Kim, Soung-Hie, Byeong-Seok Ahn, et Sang-Hyun Choi. « An efficient force planning system using multi-objective linear goal programming. » *Computers & Operations Research* 24, n° 6 (juin 1997): 569-580. doi:10.1016/S0305-0548(96)00040-8.
- Lin, W Thomas. « An accounting control system structured on multiple objective planning models. » *Omega* 8, n° 3 (1980): 375-382. doi:10.1016/0305-0483(80)90065-1.
- Lourdes Borrajo, M., Juan M. Corchado, Emilio S. Corchado, María A. Pellicer, et Javier Bajo. « Multi-agent neural business control system. » *Special Issue on Modelling Uncertainty* 180, n° 6 (15 mars 2010): 911-927. doi:10.1016/j.ins.2009.11.028.

- Mayo, D.D., W.J. Dalton, et M.J. Callaghan. « Steering strategic decisions at London underground: evaluating management options with system dynamics. » In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, 2:1578-1584 vol.2, 2003. doi:10.1109/WSC.2003.1261605.
- Rees, Jackie, et Reza Barkhi. « The problem of highly constrained tasks in group decision support systems. » *European Journal of Operational Research* 135, n° 1 (16 novembre 2001): 220-229. doi:10.1016/S0377-2217(00)00323-4.
- Sato, Yuji. « Optimal budget planning for investment in safety measures of a chemical company. » *Sixteenth international working seminar on production economics, Innsbruck, 2010* 140, n° 2 (décembre 2012): 579-585. doi:10.1016/j.ijpe.2012.05.030.
- Sethi, S.P., M. Taksar, et Q. Zhang. « Hierarchical decomposition of production and capacity investment decisions in stochastic manufacturing systems. » *International Transactions in Operational Research* 1, n° 4 (octobre 1994): 435-451. doi:10.1016/0969-6016(94)90006-X.
- Suggs, R., et B. Lewis. « Enterprise simulation - a practical application in business planning. » In *Simulation Conference, 2007 Winter*, 205-209, 2007. doi:10.1109/WSC.2007.4419602.
- Wang, Fen, G. Forgionne, et Lidan Ha. « Reestimation of e-business planning model in real business world. » In *e-Commerce Technology, 2004. CEC 2004. Proceedings. IEEE International Conference on*, 317-320, 2004. doi:10.1109/ICECT.2004.1319750.
- Wang, Huey-Jiun, Chien-Wei Chiou, et Yi-Kai Juan. « Decision support model based on case-based reasoning approach for estimating the restoration budget of historical buildings. » *Expert Systems with Applications* 35, n° 4 (novembre 2008): 1601-1610. doi:10.1016/j.eswa.2007.08.095.
- Wen, W., W.K. Wang, et C.H. Wang. « A knowledge-based intelligent decision support system for national defense budget planning. » *Expert Systems with Applications* 28, n° 1 (janvier 2005): 55-66. doi:10.1016/j.eswa.2004.08.010.
- Wong, Wing-Keung. « Stochastic dominance and mean-variance measures of profit and loss for business planning and investment. » *European Journal of Operational Research* 182, n° 2 (16 octobre 2007): 829-843. doi:10.1016/j.ejor.2006.09.032.
- Xie, Haifeng, Thierry Chausalet, Sam Toffa, et Peter Crowther. « A software tool to aid long-term care budget planning at local authority level. » *International Council on Medical and Care Computing (ICMCC)* 75, n° 9 (septembre 2006): 664-670. doi:10.1016/j.ijmedinf.2006.04.009.

Annexe 1 : L'interface du service de modélisation

```
[ServiceContract(Name = "DesignerService")]
public interface IDesignerService
{
    [OperationContract]
    XmlWorld GetScenarioXML(string scenarioIdentifier);

    [OperationContract]
    string GetScenarioURL(string scenarioIdentifier);

    [OperationContract]
    void PostScenario(string scenarioURL, string scenarioIdentifier);

    [OperationContract]
    XmlModelInstance PostScenario(string scenarioIdentifier, string modelName, string modelInstanceName);

    #region Accesseurs

    [OperationContract]
    XmlModelInstance GetModelInstance(string scenarioIdentifier);

    [OperationContract]
    XmlModelInstance GetModelInstance(string scenarioIdentifier, string modelInstanceIdentifier);

    [OperationContract]
    XmlVariableInstance GetContinuousVariableInstance(string scenarioIdentifier, string
continuousVariableInstanceIdentifier);

    [OperationContract]
    XmlScalarVariableInstance GetScalarVariableInstance(string scenarioIdentifier, string
scalarVariableInstanceIdentifier);

    [OperationContract]
    XmlModel GetModel(string scenarioIdentifier, string modelIdentifier);

    [OperationContract]
    XmlVariable GetContinuousVariable(string scenarioIdentifier, string continuousVariableIdentifier);

    [OperationContract]
    XmlScalarVariable GetScalarVariable(string scenarioIdentifier, string scalarVariableIdentifier);

    [OperationContract]
    XmlLink GetLink(string scenarioIdentifier, string linkIdentifier);

    #endregion

    #region Création

    [OperationContract]
    string PostModelInstance(string scenarioIdentifier, string parentModelInstanceIdentifier, string
modelIdentifier, string modelInstanceName);

    [OperationContract]
    void DeleteModelInstance(string scenarioIdentifier, string modelInstanceIdentifier);

    [OperationContract]
    string PostModel(string scenarioIdentifier, string parentModelIdentifier, string modelName);

    [OperationContract]
    void DeleteModel(string scenarioIdentifier, string modelIdentifier);

    [OperationContract]
    string PostContinuousVariable(string scenarioIdentifier, string modelIdentifier, string variableName);

    [OperationContract]
    void DeleteContinuousVariable(string scenarioIdentifier, string variableIdentifier);

    [OperationContract]
    string PostScalarVariable(string scenarioIdentifier, string modelIdentifier, string variableName);

    [OperationContract]
    void DeleteScalarVariable(string scenarioIdentifier, string variableIdentifier);

    [OperationContract]
    string PostLink(string scenarioIdentifier, string modelIdentifier, string linkName, string
targetModelIdentifier);

    [OperationContract]
    void DeleteLink(string scenarioIdentifier, string linkIdentifier);
}
```

```

    [OperationContract]
    string PostLinkVariable(string scenarioIdentifier, string linkIdentifier, string variableName, string
targetVariableIdentifier);

    [OperationContract]
    void DeleteLinkVariable(string scenarioIdentifier, string linkVariableIdentifier);

    #endregion

    #region Modification

    [OperationContract]
    void PostContinuousAffectation(string scenarioIdentifier, string variableInstanceIdentifier,
IEnumerable<XmlAffectationValue> affectationValues);

    [OperationContract]
    void ShiftContinuousAffectation(string scenarioIdentifier, string variableInstanceIdentifier, string
beginning, string end, string timeStepFormula);

    [OperationContract]
    void DeleteContinuousAffectation(string scenarioIdentifier, string variableInstanceIdentifier, string
beginning, string end);

    [OperationContract]
    void PostScalarAffectation(string scenarioIdentifier, string variableInstanceIdentifier, string valueFormula);

    [OperationContract]
    void DeleteScalarAffectation(string scenarioIdentifier, string variableInstanceIdentifier);

    [OperationContract]
    void PostModelInstanceActivation(string scenarioIdentifier, string modelInstanceIdentifier);

    [OperationContract]
    void DeleteModelInstanceActivation(string scenarioIdentifier, string modelInstanceIdentifier);

    [OperationContract]
    void PostAssociation(string scenarioIdentifier, string originModelInstanceIdentifier, string linkIdentifier,
string targetmodelInstanceIdentifier);

    [OperationContract]
    void DeleteAssociation(string scenarioIdentifier, string originModelInstanceIdentifier, string linkIdentifier,
string targetmodelInstanceIdentifier);

    [OperationContract]
    void PostDefaultAffectation(string scenarioIdentifier, string variableIdentifier,
IEnumerable<XmlAffectationValue> affectationValues);

    [OperationContract]
    void PostFormula(string scenarioIdentifier, string variableIdentifier, string formula);

    [OperationContract]
    void PostAgregation(string scenarioIdentifier, string variableIdentifier, string kind, IEnumerable<string>
agregatedVariableIdentifiers);

    [OperationContract]
    void DeleteComputation(string scenarioIdentifier, string variableIdentifier);

    #endregion
}

```

Figure 151 : L'interface de l'implémentation du service de modélisation avec WCF

Annexe 2 : Le modèle complet de la planification financière

Le document XML complet de la modélisation de la planification financière du cas d'étude est donné dans les pages suivantes.

```

<World xmlns="http://schemas.datacontract.org/2004/07/ScenarioXML" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Model>
    <Identifier>Models(Cantine)</Identifier>
    <Name>Cantine</Name>
    <Kind>Type</Kind>
    <OpaqueComputation i:nil="true"/>
    <ModeGroups/>
    <ScalarVariables/>
    <Links/>
    <Variables/>
    <Models>
      <Model>
        <Identifier>Models(Cantine)/Models(Investissement)</Identifier>
        <Name>Investissement</Name>
        <Kind>Type</Kind>
        <OpaqueComputation i:nil="true"/>
        <ModeGroups/>
        <ScalarVariables/>
        <Links>
          <Link>
            <Name>Emprunt</Name>
            <Identifier>Models(Cantine)/Models(Investissement)/Links(Emprunt)</Identifier>
            <TargetModelIdentifier>Models(Cantine)/Models(Emprunt)</TargetModelIdentifier>
            <Variables>
              <LinkVariable>
                <Identifier>Models(Cantine)/Models(Investissement)/Variables(Remboursement en capital_Link)</Identifier>
                <Name>Remboursement en capital_Link</Name>
                <TargetVariableIdentifier>Models(Cantine)/Models(Emprunt)/Variables(Remboursement en capital)</TargetVariableIdentifier>
              </LinkVariable>
              <LinkVariable>
                <Identifier>Models(Cantine)/Models(Investissement)/Variables(Capital emprunté_Link)</Identifier>
                <Name>Capital emprunté_Link</Name>
                <TargetVariableIdentifier>Models(Cantine)/Models(Emprunt)/Variables(Capital emprunté)</TargetVariableIdentifier>
              </LinkVariable>
              <LinkVariable>
                <Identifier>Models(Cantine)/Models(Investissement)/Variables(Remboursement en intérêt_Link)</Identifier>
                <Name>Remboursement en intérêt_Link</Name>
                <TargetVariableIdentifier>Models(Cantine)/Models(Emprunt)/Variables(Remboursement en intérêt)</TargetVariableIdentifier>
              </LinkVariable>
            </Variables>
          </Link>
          <Link>
            <Name>Opération</Name>
            <Identifier>Models(Cantine)/Models(Investissement)/Links(Opération)</Identifier>
            <TargetModelIdentifier>Models(Cantine)/Models(Opération)</TargetModelIdentifier>
            <Variables>
              <LinkVariable>
                <Identifier>Models(Cantine)/Models(Investissement)/Variables(Dépenses des opérations_Link)</Identifier>
                <Name>Dépenses des opérations_Link</Name>
                <TargetVariableIdentifier>Models(Cantine)/Models(Opération)/Variables(Dépenses)</TargetVariableIdentifier>
              </LinkVariable>
              <LinkVariable>
                <Identifier>Models(Cantine)/Models(Investissement)/Variables(Amortissement_Link)</Identifier>
                <Name>Amortissement_Link</Name>
                <TargetVariableIdentifier>Models(Cantine)/Models(Opération)/Variables(Amortissement)</TargetVariableIdentifier>
              </LinkVariable>
            </Variables>
          </Link>
        </Links>
        <Variables>
          <Variable>
            <Identifier>Models(Cantine)/Models(Investissement)/Variables(Dépenses d'investissement)</Identifier>
            <Name>Dépenses d'investissement</Name>
            <Nature>Measure</Nature>
            <Computation i:type="Agregation">
              <Kind>sum</Kind>
              <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                <a:string>Models(Cantine)/Models(Investissement)/Variables(Dépenses d'investissement hors emprunt)</a:string>
                <a:string>Models(Cantine)/Models(Investissement)/Variables(Remboursement de capital)</a:string>
              </VariableIdentifiers>
            </Computation>
          </Variable>
          <Variable>
            <Identifier>Models(Cantine)/Models(Investissement)/Variables(Dépenses d'investissement hors emprunt)</Identifier>
            <Name>Dépenses d'investissement hors emprunt</Name>
            <Nature>Measure</Nature>
            <Computation i:type="Agregation">
              <Kind>sum</Kind>
              <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                <a:string>Models(Cantine)/Models(Investissement)/Variables(Dépenses des opérations)</a:string>
                <a:string>Models(Cantine)/Models(Investissement)/Variables(Acquisition travaux)</a:string>
              </VariableIdentifiers>
            </Computation>
          </Variable>
          <Variable>
            <Identifier>Models(Cantine)/Models(Investissement)/Variables(Dépenses des opérations)</Identifier>
            <Name>Dépenses des opérations</Name>
            <Nature>Measure</Nature>
            <Computation i:type="Agregation">
              <Kind>sum</Kind>
              <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                <a:string>Models(Cantine)/Models(Investissement)/Variables(Dépenses des opérations_Link)</a:string>
              </VariableIdentifiers>
            </Computation>
          </Variable>
          <Variable>
            <Identifier>Models(Cantine)/Models(Investissement)/Variables(Acquisition travaux)</Identifier>
            <Name>Acquisition travaux</Name>
            <Nature>Measure</Nature>
            <Computation i:nil="true"/>
          </Variable>
          <Variable>
            <Identifier>Models(Cantine)/Models(Investissement)/Variables(Remboursement de capital)</Identifier>
            <Name>Remboursement de capital</Name>
            <Nature>Measure</Nature>
            <Computation i:type="Agregation">
              <Kind>sum</Kind>
              <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                <a:string>Models(Cantine)/Models(Investissement)/Variables(Remboursement en capital_Link)</a:string>
              </VariableIdentifiers>
            </Computation>
          </Variable>
          <Variable>
            <Identifier>Models(Cantine)/Models(Investissement)/Variables(Recettes d'investissement)</Identifier>
            <Name>Recettes d'investissement</Name>
            <Nature>Measure</Nature>
            <Computation i:type="Agregation">
              <Kind>sum</Kind>
              <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

```

```

    <a:string>Models(Cantine)/Models(Investissement)/Variables(Recettes d'investissement hors emprunt)</a:string>
    <a:string>Models(Cantine)/Models(Investissement)/Variables(Capital emprunté)</a:string>
  </VariableIdentifiers>
</Computation>
</Variable>
</Variable>
<Identifier>Models(Cantine)/Models(Investissement)/Variables(Recettes d'investissement hors emprunt)</Identifier>
<Name>Recettes d'investissement hors emprunt</Name>
<Nature>Measure</Nature>
<Computation i:type="Agregation">
  <Kind>sum</Kind>
  <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
    <a:string>Models(Cantine)/Models(Investissement)/Variables(Amortissement des opérations)</a:string>
    <a:string>Models(Cantine)/Models(Investissement)/Variables(Subventions d'investissement)</a:string>
    <a:string>Models(Cantine)/Models(Investissement)/Variables(Recettes affectées TVA)</a:string>
  </VariableIdentifiers>
</Computation>
</Variable>
</Variable>
<Identifier>Models(Cantine)/Models(Investissement)/Variables(Amortissement des opérations)</Identifier>
<Name>Amortissement des opérations</Name>
<Nature>Measure</Nature>
<Computation i:type="Agregation">
  <Kind>sum</Kind>
  <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
    <a:string>Models(Cantine)/Models(Investissement)/Variables(Amortissement_Link)</a:string>
  </VariableIdentifiers>
</Computation>
</Variable>
</Variable>
<Identifier>Models(Cantine)/Models(Investissement)/Variables(Subventions d'investissement)</Identifier>
<Name>Subventions d'investissement</Name>
<Nature>Measure</Nature>
<Computation i:nil="true"/>
</Variable>
</Variable>
<Identifier>Models(Cantine)/Models(Investissement)/Variables(Recettes affectées TVA)</Identifier>
<Name>Recettes affectées TVA</Name>
<Nature>Measure</Nature>
<Computation i:type="ArithmeticComputation">
  <Formula>"Dépenses des opérations"(t + 90) * 0.2</Formula>
</Computation>
</Variable>
</Variable>
<Identifier>Models(Cantine)/Models(Investissement)/Variables(Capital emprunté)</Identifier>
<Name>Capital emprunté</Name>
<Nature>Measure</Nature>
<Computation i:type="Agregation">
  <Kind>sum</Kind>
  <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
    <a:string>Models(Cantine)/Models(Investissement)/Variables(Capital emprunté_Link)</a:string>
  </VariableIdentifiers>
</Computation>
</Variable>
</Variable>
<Identifier>Models(Cantine)/Models(Investissement)/Variables(Remboursement en intérêt)</Identifier>
<Name>Remboursement en intérêt</Name>
<Nature>Measure</Nature>
<Computation i:type="Agregation">
  <Kind>sum</Kind>
  <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
    <a:string>Models(Cantine)/Models(Investissement)/Variables(Remboursement en intérêt_Link)</a:string>
  </VariableIdentifiers>
</Computation>
</Variable>
</Variables>
</Models/>
</Model>
<Model>
  <Identifier>Models(Cantine)/Models(Opération)</Identifier>
  <Name>Opération</Name>
  <Kind>Type</Kind>
  <OpaqueComputation i:nil="true"/>
  <ModeGroups/>
  <ScalarVariables/>
  <Links>
    <Link>
      <Name>Amortissement</Name>
      <Identifier>Models(Cantine)/Models(Opération)/Links(Amortissement)</Identifier>
      <TargetModelIdentifier>Models(Cantine)/Models(Amortissement)</TargetModelIdentifier>
      <Variables>
        <LinkVariable>
          <Identifier>Models(Cantine)/Models(Opération)/Variables(Capital amorti_Link)</Identifier>
          <Name>Capital amorti_Link</Name>
          <TargetVariableIdentifier>Models(Cantine)/Models(Amortissement)/Variables(Capital amorti)</TargetVariableIdentifier>
        </LinkVariable>
      </Variables>
    </Link>
  </Links>
  </Variables>
  <Variable>
    <Identifier>Models(Cantine)/Models(Opération)/Variables(Dépenses)</Identifier>
    <Name>Dépenses</Name>
    <Nature>Measure</Nature>
    <Computation i:nil="true"/>
  </Variable>
  <Variable>
    <Identifier>Models(Cantine)/Models(Opération)/Variables(Amortissement)</Identifier>
    <Name>Amortissement</Name>
    <Nature>Measure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>
      <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Opération)/Variables(Capital amorti_Link)</a:string>
      </VariableIdentifiers>
    </Computation>
  </Variable>
</Variables>
</Models/>
</Model>
<Model>
  <Identifier>Models(Cantine)/Models(Amortissement)</Identifier>
  <Name>Amortissement</Name>
  <Kind>Type</Kind>
  <OpaqueComputation>Amortissement</OpaqueComputation>
  <ModeGroups/>
  <ScalarVariables>
    <Variable>
      <Identifier>Models(Cantine)/Models(Amortissement)/ScalarVariables(Début)</Identifier>

```

```

    <Name>Début</Name>
    <Type>Date</Type>
    <Formula i:nil="true"/>
  </Variable>
  <Variable>
    <Identifier>Models(Cantine)/Models(Amortissement)/ScalarVariables(Durée)</Identifier>
    <Name>Durée</Name>
    <Type>Real</Type>
    <Formula i:nil="true"/>
  </Variable>
</ScalarVariables>
<Links/>
<Variables>
  <Variable>
    <Identifier>Models(Cantine)/Models(Amortissement)/Variables(Capital à amortir)</Identifier>
    <Name>Capital à amortir</Name>
    <Nature>Measure</Nature>
    <Computation i:nil="true"/>
  </Variable>
  <Variable>
    <Identifier>Models(Cantine)/Models(Amortissement)/Variables(Capital amorti)</Identifier>
    <Name>Capital amorti</Name>
    <Nature>Measure</Nature>
    <Computation i:type="OpaqueComputation">
      <UsedVariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Amortissement)/Variables(Capital à amortir)</a:string>
      </UsedVariableIdentifiers>
    </Computation>
  </Variable>
</Variables>
</Models>
</Model>
<Model>
  <Identifier>Models(Cantine)/Models(Emprunt)</Identifier>
  <Name>Emprunt</Name>
  <Kind>Type</Kind>
  <OpaqueComputation>Emprunt</OpaqueComputation>
  <ModeGroups>
    <ModeGroup>
      <Identifier>Models(Cantine)/Models(Emprunt)/ModeGroups(Type d'emprunt)</Identifier>
      <Modes>
        <Mode>
          <Identifier>Models(Cantine)/Models(Emprunt)/ModeGroups(Type d'emprunt)/Modes(RFTF)</Identifier>
          <Name>RFTF</Name>
        </Mode>
      </Modes>
      <Name>Type d'emprunt</Name>
    </ModeGroup>
    <ModeGroup>
      <Identifier>Models(Cantine)/Models(Emprunt)/ModeGroups(Modes de différé de remboursement)</Identifier>
      <Modes>
        <Mode>
          <Identifier>Models(Cantine)/Models(Emprunt)/ModeGroups(Modes de différé de remboursement)/Modes(Capital différé)</Identifier>
          <Name>Capital différé</Name>
        </Mode>
        <Mode>
          <Identifier>Models(Cantine)/Models(Emprunt)/ModeGroups(Modes de différé de remboursement)/Modes(2)</Identifier>
          <Name>2</Name>
        </Mode>
        <Mode>
          <Identifier>Models(Cantine)/Models(Emprunt)/ModeGroups(Modes de différé de remboursement)/Modes(In fine)</Identifier>
          <Name>In fine</Name>
        </Mode>
      </Modes>
      <Name>Modes de différé de remboursement</Name>
    </ModeGroup>
  </ModeGroups>
  <ScalarVariables>
    <Variable>
      <Identifier>Models(Cantine)/Models(Emprunt)/ScalarVariables(Taux d'intérêts)</Identifier>
      <Name>Taux d'intérêts</Name>
      <Type>Real</Type>
      <Formula i:nil="true"/>
    </Variable>
    <Variable>
      <Identifier>Models(Cantine)/Models(Emprunt)/ScalarVariables(Levée)</Identifier>
      <Name>Levée</Name>
      <Type>Date</Type>
      <Formula i:nil="true"/>
    </Variable>
    <Variable>
      <Identifier>Models(Cantine)/Models(Emprunt)/ScalarVariables(Durée de différé de remboursement)</Identifier>
      <Name>Durée de différé de remboursement</Name>
      <Type>Real</Type>
      <Formula i:nil="true"/>
    </Variable>
    <Variable>
      <Identifier>Models(Cantine)/Models(Emprunt)/ScalarVariables(Durée avant remboursement)</Identifier>
      <Name>Durée avant remboursement</Name>
      <Type>Real</Type>
      <Formula i:nil="true"/>
    </Variable>
    <Variable>
      <Identifier>Models(Cantine)/Models(Emprunt)/ScalarVariables(Durée de remboursement)</Identifier>
      <Name>Durée de remboursement</Name>
      <Type>Real</Type>
      <Formula i:nil="true"/>
    </Variable>
  </ScalarVariables>
  <Links/>
  <Variables>
    <Variable>
      <Identifier>Models(Cantine)/Models(Emprunt)/Variables(Capital emprunté)</Identifier>
      <Name>Capital emprunté</Name>
      <Nature>Measure</Nature>
      <Computation i:nil="true"/>
    </Variable>
    <Variable>
      <Identifier>Models(Cantine)/Models(Emprunt)/Variables(Montant de l'échéance)</Identifier>
      <Name>Montant de l'échéance</Name>
      <Nature>Measure</Nature>
      <Computation i:type="OpaqueComputation">
        <UsedVariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
          <a:string>Models(Cantine)/Models(Emprunt)/Variables(Périodicité)</a:string>
          <a:string>Models(Cantine)/Models(Emprunt)/Variables(Capital emprunté)</a:string>
          <a:string>Models(Cantine)/Models(Emprunt)/Variables(Jour de la première échéance)</a:string>
          <a:string>Models(Cantine)/Models(Emprunt)/Variables(Valeur du différé de remboursement)</a:string>
        </UsedVariableIdentifiers>
      </Computation>
    </Variable>
  </Variables>

```

```

</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Emprunt)/Variables(Indice Taux variables)</Identifiant>
  <Name>Indice Taux variables</Name>
  <Nature>Measure</Nature>
  <Computation i:nil="true"/>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Emprunt)/Variables(Périodicité)</Identifiant>
  <Name>Périodicité</Name>
  <Nature>Measure</Nature>
  <Computation i:nil="true"/>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Emprunt)/Variables(Intérêts courus non échus)</Identifiant>
  <Name>Intérêts courus non échus</Name>
  <Nature>Measure</Nature>
  <Computation i:type="OpaqueComputation">
    <UsedVariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Périodicité)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Capital emprunté)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Jour de la première échéance)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Valeur du différé de remboursement)</a:string>
    </UsedVariableIdentifiers>
  </Computation>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Emprunt)/Variables(Date de fin spécifiée)</Identifiant>
  <Name>Date de fin spécifiée</Name>
  <Nature>Measure</Nature>
  <Computation i:nil="true"/>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Emprunt)/Variables(Jour de la première échéance)</Identifiant>
  <Name>Jour de la première échéance</Name>
  <Nature>Measure</Nature>
  <Computation i:nil="true"/>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Emprunt)/Variables(Remboursement en intérêt)</Identifiant>
  <Name>Remboursement en intérêt</Name>
  <Nature>Measure</Nature>
  <Computation i:type="OpaqueComputation">
    <UsedVariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Périodicité)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Capital emprunté)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Jour de la première échéance)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Valeur du différé de remboursement)</a:string>
    </UsedVariableIdentifiers>
  </Computation>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Emprunt)/Variables(Valeur du différé de remboursement)</Identifiant>
  <Name>Valeur du différé de remboursement</Name>
  <Nature>Measure</Nature>
  <Computation i:nil="true"/>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Emprunt)/Variables(Remboursement en capital)</Identifiant>
  <Name>Remboursement en capital</Name>
  <Nature>Measure</Nature>
  <Computation i:type="OpaqueComputation">
    <UsedVariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Périodicité)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Capital emprunté)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Jour de la première échéance)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Valeur du différé de remboursement)</a:string>
    </UsedVariableIdentifiers>
  </Computation>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Emprunt)/Variables(Capital restant dû)</Identifiant>
  <Name>Capital restant dû</Name>
  <Nature>Field</Nature>
  <Computation i:type="OpaqueComputation">
    <UsedVariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Périodicité)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Capital emprunté)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Jour de la première échéance)</a:string>
      <a:string>Models(Cantine)/Models(Emprunt)/Variables(Valeur du différé de remboursement)</a:string>
    </UsedVariableIdentifiers>
  </Computation>
</Variable>
</Variables>
</Model>
<Model>
  <Identifiant>Models(Cantine)/Models(Fonctionnement)</Identifiant>
  <Name>Fonctionnement</Name>
  <Kind>Type</Kind>
  <OpaqueComputation i:nil="true"/>
  <ModeGroups/>
  <ScalarVariables/>
  <Links>
    <Link>
      <Name>Investissement</Name>
      <Identifiant>Models(Cantine)/Models(Fonctionnement)/Links(Investissement)</Identifiant>
      <TargetModelIdentifiant>Models(Cantine)/Models(Investissement)</TargetModelIdentifiant>
      <Variables>
        <LinkVariable>
          <Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Amortissement des investissements_Link)</Identifiant>
          <Name>Amortissement des investissements_Link</Name>
          <TargetVariableIdentifiant>Models(Cantine)/Models(Investissement)/Variables(Amortissement des opérations)</TargetVariableIdentifiant>
        </LinkVariable>
        <LinkVariable>
          <Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Remboursement en intérêt_Link)</Identifiant>
          <Name>Remboursement en intérêt_Link</Name>
          <TargetVariableIdentifiant>Models(Cantine)/Models(Investissement)/Variables(Remboursement en intérêt)</TargetVariableIdentifiant>
        </LinkVariable>
      </Variables>
    </Link>
  </Links>
</Variables>

```

```

<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Dépenses de fonctionnement)</Identifiant>
<Name>Dépenses de fonctionnement</Name>
<Nature>Mesure</Nature>
<Computation i:type="Agregation">
<Kind>sum</Kind>
<VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Dépense en denrées)</a:string>
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Charges salariales)</a:string>
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Consommations)</a:string>
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Entretien)</a:string>
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Amortissement des investissements)</a:string>
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Remboursement en intérêt)</a:string>
</VariableIdentifiers>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Dépense en denrées)</Identifiant>
<Name>Dépense en denrées</Name>
<Nature>Mesure</Nature>
<Computation i:type="ArithmeticComputation">
<Formule>"Coût en denrées d'un repas"(t) * "Nombre de repas servis"(t)</Formule>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Charges salariales)</Identifiant>
<Name>Charges salariales</Name>
<Nature>Mesure</Nature>
<Computation i:type="ArithmeticComputation">
<Formule>
</Computation>
</Variable>
<[[CDATA[If(t < BeginningOf(Year(1), "now"),
0,
Repeat
(
BeginningOf(Year(1), "now"),
Year(1),
"Charges salariales"(t)
)
)]]>
</Formule>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Consommations)</Identifiant>
<Name>Consommations</Name>
<Nature>Mesure</Nature>
<Computation i:type="ArithmeticComputation">
<Formule>
</Computation>
</Variable>
<[[CDATA[If(t < BeginningOf(Year(1), "now"),
0,
Repeat
(
BeginningOf(Year(1), "now"),
Year(1),
"Consommations"(t)
)
)]]>
</Formule>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Entretien)</Identifiant>
<Name>Entretien</Name>
<Nature>Mesure</Nature>
<Computation i:nil="true"/>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Amortissement des investissements)</Identifiant>
<Name>Amortissement des investissements</Name>
<Nature>Mesure</Nature>
<Computation i:type="Agregation">
<Kind>sum</Kind>
<VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Amortissement des investissements_Link)</a:string>
</VariableIdentifiers>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Remboursement en intérêt)</Identifiant>
<Name>Remboursement en intérêt</Name>
<Nature>Mesure</Nature>
<Computation i:type="Agregation">
<Kind>sum</Kind>
<VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Remboursement en intérêt_Link)</a:string>
</VariableIdentifiers>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Recettes de fonctionnement)</Identifiant>
<Name>Recettes de fonctionnement</Name>
<Nature>Mesure</Nature>
<Computation i:type="Agregation">
<Kind>sum</Kind>
<VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Recettes de fonctionnement hors participation de la mairie)</a:string>
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Participation à l'équilibre du budget général)</a:string>
</VariableIdentifiers>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Recettes de fonctionnement hors participation de la mairie)</Identifiant>
<Name>Recettes de fonctionnement hors participation de la mairie</Name>
<Nature>Mesure</Nature>
<Computation i:type="Agregation">
<Kind>sum</Kind>
<VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Participation des familles)</a:string>
<a:string>Models(Cantine)/Models(Fonctionnement)/Variables(Subventions de fonctionnement)</a:string>
</VariableIdentifiers>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Participation des familles)</Identifiant>
<Name>Participation des familles</Name>
<Nature>Mesure</Nature>
<Computation i:type="ArithmeticComputation">
<Formule>"Prix d'un ticket"(t) * "Quantité de tickets vendus"(t)</Formule>
</Computation>
</Variable>

```

```

    <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Subventions de fonctionnement)</Identifiant>
    <Name>Subventions de fonctionnement</Name>
    <Nature>Measure</Nature>
    <Computation i:type="ArithmeticComputation">
    <Formula>
<![CDATA[If(t < BeginningOf(Year(1), "now"()),
0,
Repeat
(
BeginningOf(Year(1), "now"()),
Year(1),
"Subventions de fonctionnement"(t)
)
)]>
</Formula>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Participation à l'équilibre du budget général)</Identifiant>
<Name>Participation à l'équilibre du budget général</Name>
<Nature>Measure</Nature>
<Computation i:type="ArithmeticComputation">
<Formula>"Dépenses de fonctionnement"(t) - "Recettes de fonctionnement hors participation de la mairie"(t)</Formula>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Prix d'un ticket)</Identifiant>
<Name>Prix d'un ticket</Name>
<Nature>Field</Nature>
<Computation i:type="ArithmeticComputation">
<Formula>
<![CDATA[If(t < BeginningOf(Year(1), "now"()),
0,
Repeat
(
BeginningOf(Year(1), "now"()),
Year(1),
"Prix d'un ticket"(t)
)
)]>
</Formula>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Coût en denrées d'un repas)</Identifiant>
<Name>Coût en denrées d'un repas</Name>
<Nature>Field</Nature>
<Computation i:type="ArithmeticComputation">
<Formula>
<![CDATA[If(t < BeginningOf(Year(1), "now"()),
0,
Repeat
(
BeginningOf(Year(1), "now"() - Year(1)),
Year(1),
"Coût en denrées d'un repas"(t) / Exp(BeginningOf(Year(1), "now"()), 1.02, Year(1)))
)
* Exp(BeginningOf(Year(1), "now"()), 1.02, Year(1))
)
]]>
</Formula>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Quantité de tickets vendus)</Identifiant>
<Name>Quantité de tickets vendus</Name>
<Nature>Measure</Nature>
<Computation i:type="ArithmeticComputation">
<Formula>
<![CDATA[If(t < BeginningOf(Year(1), "now"()),
0,
Repeat
(
BeginningOf(Year(1), "now"()),
Year(1),
(Delay(Year(2), "Quantité de tickets vendus"(t))
+Delay(Year(1), "Quantité de tickets vendus"(t))
) / 2
)
)]>
</Formula>
</Computation>
</Variable>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Nombre de repas servis)</Identifiant>
<Name>Nombre de repas servis</Name>
<Nature>Measure</Nature>
<Computation i:type="ArithmeticComputation">
<Formula>
<![CDATA[If(t < BeginningOf(Year(1), "now"()),
0,
Repeat
(
BeginningOf(Year(1), "now"()),
Year(1),
(Delay(Year(2), "Nombre de repas servis"(t))
+Delay(Year(1), "Nombre de repas servis"(t))
) / 2
)
)]>
</Formula>
</Computation>
</Variable>
</Variables>
</Models/>
</Model>
</Model>
<Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)</Identifiant>
<Name>Fonctionnement avec déménagement</Name>
<Kind>Type</Kind>
<OpaqueComputation i:nil="true"/>
<ModeGroups/>
<ScalarVariables>
<Variable>
<Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/ScalarVariables(Déménagement)</Identifiant>
<Name>Déménagement</Name>
<Type>Date</Type>
<Formula i:nil="true"/>
</Variable>
</ScalarVariables>

```



```

</LinkVariable>
<LinkVariable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Prix d'un ticket après_Link)</Identifiant>
  <Name>Prix d'un ticket après_Link</Name>
  <TargetVariableIdentifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Prix d'un ticket)</TargetVariableIdentifiant>
</LinkVariable>
<LinkVariable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Coût en denrées d'un repas après_Link)</Identifiant>
  <Name>Coût en denrées d'un repas après_Link</Name>
  <TargetVariableIdentifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Coût en denrées d'un repas)</TargetVariableIdentifiant>
</LinkVariable>
<LinkVariable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Quantité de tickets vendus après_Link)</Identifiant>
  <Name>Quantité de tickets vendus après_Link</Name>
  <TargetVariableIdentifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Quantité de tickets vendus)</TargetVariableIdentifiant>
</LinkVariable>
<LinkVariable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Nombre de repas servis après_Link)</Identifiant>
  <Name>Nombre de repas servis après_Link</Name>
  <TargetVariableIdentifiant>Models(Cantine)/Models(Fonctionnement)/Variables(Nombre de repas servis)</TargetVariableIdentifiant>
</LinkVariable>
</Variables>
</Link>
</Links>
<Variables>
  <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Dépense en denrées avant)</Identifiant>
    <Name>Dépense en denrées avant</Name>
    <Nature>Mesure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>
      <VariableIdentifiants xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Dépense en denrées avant_Link)</a:string>
      </VariableIdentifiants>
    </Computation>
  </Variable>
  <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Charges salariales avant)</Identifiant>
    <Name>Charges salariales avant</Name>
    <Nature>Mesure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>
      <VariableIdentifiants xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Charges salariales avant_Link)</a:string>
      </VariableIdentifiants>
    </Computation>
  </Variable>
  <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Consommations avant)</Identifiant>
    <Name>Consommations avant</Name>
    <Nature>Mesure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>
      <VariableIdentifiants xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Consommations avant_Link)</a:string>
      </VariableIdentifiants>
    </Computation>
  </Variable>
  <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Entretien avant)</Identifiant>
    <Name>Entretien avant</Name>
    <Nature>Mesure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>
      <VariableIdentifiants xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Entretien avant_Link)</a:string>
      </VariableIdentifiants>
    </Computation>
  </Variable>
  <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation des familles avant)</Identifiant>
    <Name>Participation des familles avant</Name>
    <Nature>Mesure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>
      <VariableIdentifiants xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation des familles avant_Link)</a:string>
      </VariableIdentifiants>
    </Computation>
  </Variable>
  <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Subventions de fonctionnement avant)</Identifiant>
    <Name>Subventions de fonctionnement avant</Name>
    <Nature>Mesure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>
      <VariableIdentifiants xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Subventions de fonctionnement avant_Link)</a:string>
      </VariableIdentifiants>
    </Computation>
  </Variable>
  <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation à l'équilibre du budget général avant)</Identifiant>
    <Name>Participation à l'équilibre du budget général avant</Name>
    <Nature>Mesure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>
      <VariableIdentifiants xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation à l'équilibre du budget général avant_Link)</a:string>
      </VariableIdentifiants>
    </Computation>
  </Variable>
  <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Dépense en denrées après)</Identifiant>
    <Name>Dépense en denrées après</Name>
    <Nature>Mesure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>
      <VariableIdentifiants xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Dépense en denrées après_Link)</a:string>
      </VariableIdentifiants>
    </Computation>
  </Variable>
  <Variable>
    <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Charges salariales après)</Identifiant>
    <Name>Charges salariales après</Name>
    <Nature>Mesure</Nature>
    <Computation i:type="Agregation">
      <Kind>sum</Kind>

```

```

    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Charges salariales après_Link)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Consommations après)</Identifier>
  <Name>Consommations après</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Charges salariales après_Link)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Entretien après)</Identifier>
  <Name>Entretien après</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Entretien après_Link)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation des familles après)</Identifier>
  <Name>Participation des familles après</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation des familles après_Link)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Subventions de fonctionnement après)</Identifier>
  <Name>Subventions de fonctionnement après</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Subventions de fonctionnement après_Link)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation à l'équilibre du budget général après)</Identifier>
  <Name>Participation à l'équilibre du budget général après</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation à l'équilibre du budget général après_Link)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Dépenses de fonctionnement)</Identifier>
  <Name>Dépenses de fonctionnement</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Dépense en denrées)</a:string>
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Charges salariales)</a:string>
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Consommations)</a:string>
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Entretien)</a:string>
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Amortissement des investissements)</a:string>
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Remboursement en intérêt)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Dépense en denrées)</Identifier>
  <Name>Dépense en denrées</Name>
  <Nature>Measure</Nature>
  <Computation i:type="ArithmeticComputation">
    <Formula>if(t &lt;t; "Déménagement"(), "Dépense en denrées avant"(t), "Dépense en denrées après"(t))</Formula>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Charges salariales)</Identifier>
  <Name>Charges salariales</Name>
  <Nature>Measure</Nature>
  <Computation i:type="ArithmeticComputation">
    <Formula>if(t &lt;t; "Déménagement"(), "Charges salariales avant"(t), "Charges salariales après"(t))</Formula>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Consommations)</Identifier>
  <Name>Consommations</Name>
  <Nature>Measure</Nature>
  <Computation i:type="ArithmeticComputation">
    <Formula>if(t &lt;t; "Déménagement"(), "Consommations avant"(t), "Consommations après"(t))</Formula>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Entretien)</Identifier>
  <Name>Entretien</Name>
  <Nature>Measure</Nature>
  <Computation i:type="ArithmeticComputation">
    <Formula>if(t &lt;t; "Déménagement"(), "Entretien avant"(t), "Entretien après"(t))</Formula>
  </Computation>
</Variable>
</Variable>
<Variable>
  <Identifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Amortissement des investissements)</Identifier>
  <Name>Amortissement des investissements</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Amortissement des investissements_Link)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>

```

```

<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Remboursement en intérêt)</Identifiant>
  <Name>Remboursement en intérêt</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Remboursement en intérêt_Link)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Recettes de fonctionnement)</Identifiant>
  <Name>Recettes de fonctionnement</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Recettes de fonctionnement hors participation de la mairie)</a:string>
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation à l'équilibre du budget général)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Recettes de fonctionnement hors participation de la mairie)</Identifiant>
  <Name>Recettes de fonctionnement hors participation de la mairie</Name>
  <Nature>Measure</Nature>
  <Computation i:type="Agregation">
    <Kind>sum</Kind>
    <VariableIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation des familles)</a:string>
      <a:string>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Subventions de fonctionnement)</a:string>
    </VariableIdentifiers>
  </Computation>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation des familles)</Identifiant>
  <Name>Participation des familles</Name>
  <Nature>Measure</Nature>
  <Computation i:type="ArithmeticComputation">
    <Formula>if(t &lt;t; "Déménagement"(), "Participation des familles avant"(t), "Participation des familles après"(t))</Formula>
  </Computation>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Subventions de fonctionnement)</Identifiant>
  <Name>Subventions de fonctionnement</Name>
  <Nature>Measure</Nature>
  <Computation i:type="ArithmeticComputation">
    <Formula>if(t &lt;t; "Déménagement"(), "Subventions de fonctionnement avant"(t), "Subventions de fonctionnement après"(t))</Formula>
  </Computation>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Participation à l'équilibre du budget général)</Identifiant>
  <Name>Participation à l'équilibre du budget général</Name>
  <Nature>Measure</Nature>
  <Computation i:type="ArithmeticComputation">
    <Formula>if(t &lt;t; "Déménagement"(), "Participation à l'équilibre du budget général avant"(t), "Participation à l'équilibre du budget général
après"(t))</Formula>
  </Computation>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Prix d'un ticket)</Identifiant>
  <Name>Prix d'un ticket</Name>
  <Nature>Field</Nature>
  <Computation i:nil="true"/>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Coût en denrées d'un repas)</Identifiant>
  <Name>Coût en denrées d'un repas</Name>
  <Nature>Field</Nature>
  <Computation i:nil="true"/>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Quantité de tickets vendus)</Identifiant>
  <Name>Quantité de tickets vendus</Name>
  <Nature>Measure</Nature>
  <Computation i:nil="true"/>
</Variable>
<Variable>
  <Identifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Nombre de repas servis)</Identifiant>
  <Name>Nombre de repas servis</Name>
  <Nature>Measure</Nature>
  <Computation i:nil="true"/>
</Variable>
</Variables>
</Model>
</Models>
</Model>
<ModelInstance>
  <Name>Budget cantine</Name>
  <Identifiant>ModelInstances(Budget cantine)</Identifiant>
  <ModelIdentifiant>Models(Cantine)</ModelIdentifiant>
  <IsActive>true</IsActive>
  <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <ScalarVariableInstances/>
  <VariableInstances/>
  <ChildrenModelInstances>
    <ModelInstance>
      <Name>Investissement</Name>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Investissement)</Identifiant>
      <ModelIdentifiant>Models(Cantine)/Models(Investissement)</ModelIdentifiant>
      <IsActive>true</IsActive>
      <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
      <ScalarVariableInstances/>
      <VariableInstances>
        <VariableInstance>
          <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Dépenses d'investissement)</Identifiant>
          <VariableIdentifiant>Models(Cantine)/Models(Investissement)/Variables(Dépenses d'investissement)</VariableIdentifiant>
          <DatedValues/>
        </VariableInstance>
        <VariableInstance>
          <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Dépenses d'investissement hors emprunt)</Identifiant>
          <VariableIdentifiant>Models(Cantine)/Models(Investissement)/Variables(Dépenses d'investissement hors emprunt)</VariableIdentifiant>
          <DatedValues/>
        </VariableInstance>
        <VariableInstance>
          <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Dépenses des opérations)</Identifiant>
          <VariableIdentifiant>Models(Cantine)/Models(Investissement)/Variables(Dépenses des opérations)</VariableIdentifiant>
        </VariableInstance>
      </VariableInstances>
    </ModelInstance>
  </ChildrenModelInstances>

```

```

</DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Acquisition travaux)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Investissement)/Variables(Acquisition travaux)</VariableIdentifier>
<DatedValues>
<AffectionValue>
<Bounds>
<Beginning>2016-01-01T00:00:00</Beginning>
<End>2017-01-01T00:00:00</End>
</Bounds>
<Value>50000</Value>
<Pattern i:type="PonctualPattern">
<Bounds>
<Beginning>0</Beginning>
<End>1</End>
</Bounds>
<Elements>
<Ponctual>
<Weight>1</Weight>
<Location>0</Location>
</Ponctual>
</Elements>
</Pattern>
</AffectionValue>
</DatedValues>
</VariableInstance>
<VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Remboursement de capital)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Investissement)/Variables(Remboursement de capital)</VariableIdentifier>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Recettes d'investissement)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Investissement)/Variables(Recettes d'investissement)</VariableIdentifier>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Recettes d'investissement hors emprunt)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Investissement)/Variables(Recettes d'investissement hors emprunt)</VariableIdentifier>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Amortissement des opérations)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Investissement)/Variables(Amortissement des opérations)</VariableIdentifier>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Subventions d'investissement)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Investissement)/Variables(Subventions d'investissement)</VariableIdentifier>
<DatedValues>
<AffectionValue>
<Bounds>
<Beginning>2015-11-14T00:00:00</Beginning>
<End>2015-11-15T00:00:00</End>
</Bounds>
<Value>70000</Value>
<Pattern i:type="PonctualPattern">
<Bounds>
<Beginning>0</Beginning>
<End>1</End>
</Bounds>
<Elements>
<Ponctual>
<Weight>1</Weight>
<Location>0</Location>
</Ponctual>
</Elements>
</Pattern>
</AffectionValue>
<AffectionValue>
<Bounds>
<Beginning>2015-12-13T00:00:00</Beginning>
<End>2015-12-14T00:00:00</End>
</Bounds>
<Value>70000</Value>
<Pattern i:type="PonctualPattern">
<Bounds>
<Beginning>0</Beginning>
<End>1</End>
</Bounds>
<Elements>
<Ponctual>
<Weight>1</Weight>
<Location>0</Location>
</Ponctual>
</Elements>
</Pattern>
</AffectionValue>
<AffectionValue>
<Bounds>
<Beginning>2016-01-15T00:00:00</Beginning>
<End>2016-01-16T00:00:00</End>
</Bounds>
<Value>70000</Value>
<Pattern i:type="PonctualPattern">
<Bounds>
<Beginning>0</Beginning>
<End>1</End>
</Bounds>
<Elements>
<Ponctual>
<Weight>1</Weight>
<Location>0</Location>
</Ponctual>
</Elements>
</Pattern>
</AffectionValue>
</DatedValues>
</VariableInstance>
<VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Recettes affectées TVA)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Investissement)/Variables(Recettes affectées TVA)</VariableIdentifier>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Capital emprunté)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Investissement)/Variables(Capital emprunté)</VariableIdentifier>
<DatedValues/>

```

```

</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)/Variables(Remboursement en intérêt)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Investissement)/Variables(Remboursement en intérêt)</VariableIdentifier>
  <DatedValues/>
</VariableInstance>
</VariableInstances>
<ChildrenModelInstances/>
<OutAssociations>
  <Association>
    <LinkIdentifier>Models(Cantine)/Models(Investissement)/Links(Emprunt)</LinkIdentifier>
    <ModelInstanceIdentifier>ModelInstances(Budget cantine)/ModelInstances(Emprunt)</ModelInstanceIdentifier>
  </Association>
  <Association>
    <LinkIdentifier>Models(Cantine)/Models(Investissement)/Links(Opération)</LinkIdentifier>
    <ModelInstanceIdentifier>ModelInstances(Budget cantine)/ModelInstances(Opération de maçonnerie)</ModelInstanceIdentifier>
  </Association>
  <Association>
    <LinkIdentifier>Models(Cantine)/Models(Investissement)/Links(Opération)</LinkIdentifier>
    <ModelInstanceIdentifier>ModelInstances(Budget cantine)/ModelInstances(Opération d'isolation)</ModelInstanceIdentifier>
  </Association>
  <Association>
    <LinkIdentifier>Models(Cantine)/Models(Investissement)/Links(Opération)</LinkIdentifier>
    <ModelInstanceIdentifier>ModelInstances(Budget cantine)/ModelInstances(Opération de chauffage)</ModelInstanceIdentifier>
  </Association>
</OutAssociations>
</ModelInstance>
<ModelInstance>
  <Name>Fonctionnement avant</Name>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)</Identifier>
  <ModelIdentifier>Models(Cantine)/Models(Fonctionnement)</ModelIdentifier>
  <IsActive>true</IsActive>
  <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <ScalarVariableInstances/>
  <VariableInstances>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Dépenses de fonctionnement)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Dépenses de fonctionnement)</VariableIdentifier>
      <DatedValues/>
    </VariableInstance>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Dépense en denrées)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Dépense en denrées)</VariableIdentifier>
      <DatedValues>
        <AffectationValue>
          <Bounds>
            <Beginning>2012-01-01T00:00:00</Beginning>
            <End>2013-01-01T00:00:00</End>
          </Bounds>
          <Value>115340.25374855824</Value>
          <Pattern i:type="ShapePattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>12</End>
            </Bounds>
            <Elements>
              <Shape i:type="Affine">
                <End>6</End>
                <BeginningHeight>1</BeginningHeight>
                <EndHeight>0.9</EndHeight>
              </Shape>
              <Shape i:type="Affine">
                <End>12</End>
                <BeginningHeight>0.9</BeginningHeight>
                <EndHeight>1.02</EndHeight>
              </Shape>
            </Elements>
          </Pattern>
        </AffectationValue>
        <AffectationValue>
          <Bounds>
            <Beginning>2013-01-01T00:00:00</Beginning>
            <End>2014-01-01T00:00:00</End>
          </Bounds>
          <Value>117647.05882352941</Value>
          <Pattern i:type="ShapePattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>12</End>
            </Bounds>
            <Elements>
              <Shape i:type="Affine">
                <End>6</End>
                <BeginningHeight>1</BeginningHeight>
                <EndHeight>0.9</EndHeight>
              </Shape>
              <Shape i:type="Affine">
                <End>12</End>
                <BeginningHeight>0.9</BeginningHeight>
                <EndHeight>1.02</EndHeight>
              </Shape>
            </Elements>
          </Pattern>
        </AffectationValue>
        <AffectationValue>
          <Bounds>
            <Beginning>2014-01-01T00:00:00</Beginning>
            <End>2015-01-01T00:00:00</End>
          </Bounds>
          <Value>120000</Value>
          <Pattern i:type="ShapePattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>12</End>
            </Bounds>
            <Elements>
              <Shape i:type="Affine">
                <End>6</End>
                <BeginningHeight>1</BeginningHeight>
                <EndHeight>0.9</EndHeight>
              </Shape>
              <Shape i:type="Affine">
                <End>12</End>
                <BeginningHeight>0.9</BeginningHeight>
                <EndHeight>1.02</EndHeight>
              </Shape>
            </Elements>
          </Pattern>
        </AffectationValue>
      </DatedValues>
    </VariableInstance>
  </VariableInstances>
</ModelInstance>

```



```

</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-07-01T00:00:00</Beginning>
    <End>2012-08-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-08-01T00:00:00</Beginning>
    <End>2012-09-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-09-01T00:00:00</Beginning>
    <End>2012-10-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-10-01T00:00:00</Beginning>
    <End>2012-11-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-11-01T00:00:00</Beginning>
    <End>2012-12-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-12-01T00:00:00</Beginning>
    <End>2013-01-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-01-01T00:00:00</Beginning>
    <End>2013-02-01T00:00:00</End>
  </Bounds>

```

```

</Bounds>
<Value>5000</Value>
<Pattern i:type="PonctualPattern">
  <Bounds>
    <Beginning>0</Beginning>
    <End>30</End>
  </Bounds>
  <Elements>
    <Ponctual>
      <Weight>1</Weight>
      <Location>25</Location>
    </Ponctual>
  </Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-02-01T00:00:00</Beginning>
    <End>2013-03-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-03-01T00:00:00</Beginning>
    <End>2013-04-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-04-01T00:00:00</Beginning>
    <End>2013-05-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-05-01T00:00:00</Beginning>
    <End>2013-06-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-06-01T00:00:00</Beginning>
    <End>2013-07-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-07-01T00:00:00</Beginning>
    <End>2013-08-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>

```

```

    <End>30</End>
  </Bounds>
  <Elements>
    <Ponctual>
      <Weight>1</Weight>
      <Location>25</Location>
    </Ponctual>
  </Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-08-01T00:00:00</Beginning>
    <End>2013-09-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-09-01T00:00:00</Beginning>
    <End>2013-10-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-10-01T00:00:00</Beginning>
    <End>2013-11-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-11-01T00:00:00</Beginning>
    <End>2013-12-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-12-01T00:00:00</Beginning>
    <End>2014-01-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-01-01T00:00:00</Beginning>
    <End>2014-02-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>

```

```

        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-02-01T00:00:00</Beginning>
    <End>2014-03-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-03-01T00:00:00</Beginning>
    <End>2014-04-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-04-01T00:00:00</Beginning>
    <End>2014-05-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-05-01T00:00:00</Beginning>
    <End>2014-06-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-06-01T00:00:00</Beginning>
    <End>2014-07-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-07-01T00:00:00</Beginning>
    <End>2014-08-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>

```

```

<AffectionValue>
  <Bounds>
    <Beginning>2014-08-01T00:00:00</Beginning>
    <End>2014-09-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectionValue>
<AffectionValue>
  <Bounds>
    <Beginning>2014-09-01T00:00:00</Beginning>
    <End>2014-10-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectionValue>
<AffectionValue>
  <Bounds>
    <Beginning>2014-10-01T00:00:00</Beginning>
    <End>2014-11-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectionValue>
<AffectionValue>
  <Bounds>
    <Beginning>2014-11-01T00:00:00</Beginning>
    <End>2014-12-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectionValue>
<AffectionValue>
  <Bounds>
    <Beginning>2014-12-01T00:00:00</Beginning>
    <End>2015-01-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectionValue>
</DatedValues>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Consommations)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Consommations)</VariableIdentifier>
</DatedValues>
<AffectionValue>
  <Bounds>
    <Beginning>2012-01-01T00:00:00</Beginning>
    <End>2013-01-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>1</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>0</Location>
      </Ponctual>
    </Elements>
  </Pattern>

```

```

</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-01-01T00:00:00</Beginning>
    <End>2014-01-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>1</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>0</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-01-01T00:00:00</Beginning>
    <End>2015-01-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>1</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>0</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
</DatedValues>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Entretien)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Entretien)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Amortissement des investissements)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Amortissement des investissements)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Remboursement en intérêt)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Remboursement en intérêt)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Recettes de fonctionnement)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Recettes de fonctionnement)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Recettes de fonctionnement hors participation de la
mairie)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Recettes de fonctionnement hors participation de la mairie)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Participation des familles)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Participation des familles)</VariableIdentifier>
</DatedValues>
<AffectationValue>
  <Bounds>
    <Beginning>2012-09-01T00:00:00</Beginning>
    <End>2013-09-01T00:00:00</End>
  </Bounds>
  <Value>60000</Value>
  <Pattern i:type="ShapePattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>12</End>
    </Bounds>
    <Elements>
      <Shape i:type="Constant">
        <End>1</End>
        <Height>0.25</Height>
      </Shape>
      <Shape i:type="Constant">
        <End>10</End>
        <Height>0.083333333333333329</Height>
      </Shape>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-09-01T00:00:00</Beginning>
    <End>2014-09-01T00:00:00</End>
  </Bounds>
  <Value>60000</Value>
  <Pattern i:type="ShapePattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>12</End>
    </Bounds>
    <Elements>
      <Shape i:type="Constant">
        <End>1</End>
        <Height>0.25</Height>
      </Shape>
      <Shape i:type="Constant">
        <End>10</End>
        <Height>0.083333333333333329</Height>
      </Shape>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>

```

```

    <Beginning>2014-09-01T00:00:00</Beginning>
    <End>2015-09-01T00:00:00</End>
  </Bounds>
  <Value>60000</Value>
  <Pattern i:type="ShapePattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>12</End>
    </Bounds>
    <Elements>
      <Shape i:type="Constant">
        <End>1</End>
        <Height>0.25</Height>
      </Shape>
      <Shape i:type="Constant">
        <End>10</End>
        <Height>0.083333333333333329</Height>
      </Shape>
    </Elements>
  </Pattern>
</AffectationValue>
</DatedValues>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Subventions de fonctionnement)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Subventions de fonctionnement)</VariableIdentifier>
  <DatedValues>
    <AffectationValue>
      <Bounds>
        <Beginning>2012-01-01T00:00:00</Beginning>
        <End>2013-01-01T00:00:00</End>
      </Bounds>
      <Value>48000</Value>
      <Pattern i:type="PonctualPattern">
        <Bounds>
          <Beginning>0</Beginning>
          <End>1</End>
        </Bounds>
        <Elements>
          <Ponctual>
            <Weight>1</Weight>
            <Location>0</Location>
          </Ponctual>
        </Elements>
      </Pattern>
    </AffectationValue>
  </AffectationValue>
  <Bounds>
    <Beginning>2013-01-01T00:00:00</Beginning>
    <End>2014-01-01T00:00:00</End>
  </Bounds>
  <Value>48000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>1</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>0</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
</AffectationValue>
  <Bounds>
    <Beginning>2014-01-01T00:00:00</Beginning>
    <End>2015-01-01T00:00:00</End>
  </Bounds>
  <Value>48000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>1</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>0</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
</DatedValues>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Participation à l'équilibre du budget général)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Participation à l'équilibre du budget général)</VariableIdentifier>
  <DatedValues/>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Prix d'un ticket)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Prix d'un ticket)</VariableIdentifier>
  <DatedValues/>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Coût en denrées d'un repas)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Coût en denrées d'un repas)</VariableIdentifier>
  <DatedValues/>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Quantité de tickets vendus)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Quantité de tickets vendus)</VariableIdentifier>
  <DatedValues/>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)/Variables(Nombre de repas servis)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Nombre de repas servis)</VariableIdentifier>
  <DatedValues/>
</VariableInstance>
</VariableInstances>
<ChildrenModelInstances/>
<OutAssociations>
  <Association>
    <LinkIdentifier>Models(Cantine)/Models(Fonctionnement)/Links(Investissement)</LinkIdentifier>
    <ModelInstanceIdentifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)</ModelInstanceIdentifier>
  </Association>
</OutAssociations>

```

```

</ModelInstance>
<ModelInstance>
  <Name>Fonctionnement après</Name>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)</Identifier>
  <ModelIdentifier>Models(Cantine)/Models(Fonctionnement)</ModelIdentifier>
  <IsActive>true</IsActive>
  <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <ScalarVariableInstances/>
  <VariableInstances>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Dépenses de fonctionnement)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Dépenses de fonctionnement)</VariableIdentifier>
      <DatedValues/>
    </VariableInstance>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Dépense en denrées)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Dépense en denrées)</VariableIdentifier>
      <DatedValues>
        <AffectationValue>
          <Bounds>
            <Beginning>2012-01-01T00:00:00</Beginning>
            <End>2013-01-01T00:00:00</End>
          </Bounds>
          <Value>115340.25374855824</Value>
          <Pattern i:type="ShapePattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>12</End>
            </Bounds>
            <Elements>
              <Shape i:type="Affine">
                <End>6</End>
                <BeginningHeight>1</BeginningHeight>
                <EndHeight>0.9</EndHeight>
              </Shape>
              <Shape i:type="Affine">
                <End>12</End>
                <BeginningHeight>0.9</BeginningHeight>
                <EndHeight>1.02</EndHeight>
              </Shape>
            </Elements>
          </Pattern>
        </AffectationValue>
        <AffectationValue>
          <Bounds>
            <Beginning>2013-01-01T00:00:00</Beginning>
            <End>2014-01-01T00:00:00</End>
          </Bounds>
          <Value>117647.05882352941</Value>
          <Pattern i:type="ShapePattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>12</End>
            </Bounds>
            <Elements>
              <Shape i:type="Affine">
                <End>6</End>
                <BeginningHeight>1</BeginningHeight>
                <EndHeight>0.9</EndHeight>
              </Shape>
              <Shape i:type="Affine">
                <End>12</End>
                <BeginningHeight>0.9</BeginningHeight>
                <EndHeight>1.02</EndHeight>
              </Shape>
            </Elements>
          </Pattern>
        </AffectationValue>
        <AffectationValue>
          <Bounds>
            <Beginning>2014-01-01T00:00:00</Beginning>
            <End>2015-01-01T00:00:00</End>
          </Bounds>
          <Value>120000</Value>
          <Pattern i:type="ShapePattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>12</End>
            </Bounds>
            <Elements>
              <Shape i:type="Affine">
                <End>6</End>
                <BeginningHeight>1</BeginningHeight>
                <EndHeight>0.9</EndHeight>
              </Shape>
              <Shape i:type="Affine">
                <End>12</End>
                <BeginningHeight>0.9</BeginningHeight>
                <EndHeight>1.02</EndHeight>
              </Shape>
            </Elements>
          </Pattern>
        </AffectationValue>
      </DatedValues>
    </VariableInstance>
  </VariableInstances>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Charges salariales)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Charges salariales)</VariableIdentifier>
  <DatedValues>
    <AffectationValue>
      <Bounds>
        <Beginning>2012-01-01T00:00:00</Beginning>
        <End>2012-02-01T00:00:00</End>
      </Bounds>
      <Value>5000</Value>
      <Pattern i:type="PonctualPattern">
        <Bounds>
          <Beginning>0</Beginning>
          <End>30</End>
        </Bounds>
        <Elements>
          <Ponctual>
            <Weight>1</Weight>
            <Location>25</Location>
          </Ponctual>
        </Elements>
      </Pattern>
    </AffectationValue>
  </DatedValues>
</ModelInstance>

```

```

<Bounds>
  <Beginning>2012-02-01T00:00:00</Beginning>
  <End>2012-03-01T00:00:00</End>
</Bounds>
<Value>5000</Value>
<Pattern i:type="PonctualPattern">
  <Bounds>
    <Beginning>0</Beginning>
    <End>30</End>
  </Bounds>
  <Elements>
    <Ponctual>
      <Weight>1</Weight>
      <Location>25</Location>
    </Ponctual>
  </Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-03-01T00:00:00</Beginning>
    <End>2012-04-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-04-01T00:00:00</Beginning>
    <End>2012-05-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-05-01T00:00:00</Beginning>
    <End>2012-06-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-06-01T00:00:00</Beginning>
    <End>2012-07-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-07-01T00:00:00</Beginning>
    <End>2012-08-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-08-01T00:00:00</Beginning>
    <End>2012-09-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>

```

```

<Pattern i:type="PonctualPattern">
  <Bounds>
    <Beginning>0</Beginning>
    <End>30</End>
  </Bounds>
  <Elements>
    <Ponctual>
      <Weight>1</Weight>
      <Location>25</Location>
    </Ponctual>
  </Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-09-01T00:00:00</Beginning>
    <End>2012-10-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-10-01T00:00:00</Beginning>
    <End>2012-11-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-11-01T00:00:00</Beginning>
    <End>2012-12-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2012-12-01T00:00:00</Beginning>
    <End>2013-01-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-01-01T00:00:00</Beginning>
    <End>2013-02-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-02-01T00:00:00</Beginning>
    <End>2013-03-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
  </Pattern>
</AffectationValue>

```

```

<Elements>
  <Ponctual>
    <Weight>1</Weight>
    <Location>25</Location>
  </Ponctual>
</Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-03-01T00:00:00</Beginning>
    <End>2013-04-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-04-01T00:00:00</Beginning>
    <End>2013-05-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-05-01T00:00:00</Beginning>
    <End>2013-06-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-06-01T00:00:00</Beginning>
    <End>2013-07-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-07-01T00:00:00</Beginning>
    <End>2013-08-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-08-01T00:00:00</Beginning>
    <End>2013-09-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
</AffectationValue>

```

```

</Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-09-01T00:00:00</Beginning>
    <End>2013-10-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-10-01T00:00:00</Beginning>
    <End>2013-11-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-11-01T00:00:00</Beginning>
    <End>2013-12-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-12-01T00:00:00</Beginning>
    <End>2014-01-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-01-01T00:00:00</Beginning>
    <End>2014-02-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-02-01T00:00:00</Beginning>
    <End>2014-03-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>

```

```

    <Beginning>2014-03-01T00:00:00</Beginning>
  <End>2014-04-01T00:00:00</End>
</Bounds>
<Value>5000</Value>
<Pattern i:type="PonctualPattern">
  <Bounds>
    <Beginning>0</Beginning>
    <End>30</End>
  </Bounds>
  <Elements>
    <Ponctual>
      <Weight>1</Weight>
      <Location>25</Location>
    </Ponctual>
  </Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-04-01T00:00:00</Beginning>
    <End>2014-05-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-05-01T00:00:00</Beginning>
    <End>2014-06-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-06-01T00:00:00</Beginning>
    <End>2014-07-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-07-01T00:00:00</Beginning>
    <End>2014-08-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-08-01T00:00:00</Beginning>
    <End>2014-09-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-09-01T00:00:00</Beginning>
    <End>2014-10-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">

```

```

<Bounds>
  <Beginning>0</Beginning>
  <End>30</End>
</Bounds>
<Elements>
  <Ponctual>
    <Weight>1</Weight>
    <Location>25</Location>
  </Ponctual>
</Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-10-01T00:00:00</Beginning>
    <End>2014-11-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-11-01T00:00:00</Beginning>
    <End>2014-12-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-12-01T00:00:00</Beginning>
    <End>2015-01-01T00:00:00</End>
  </Bounds>
  <Value>5000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>30</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>25</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
</VariableInstance>
<VariableInstance>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Consommations)</Identifier>
  <VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Consommations)</VariableIdentifier>
  <DatedValues>
    <AffectationValue>
      <Bounds>
        <Beginning>2012-01-01T00:00:00</Beginning>
        <End>2013-01-01T00:00:00</End>
      </Bounds>
      <Value>1000</Value>
      <Pattern i:type="PonctualPattern">
        <Bounds>
          <Beginning>0</Beginning>
          <End>1</End>
        </Bounds>
        <Elements>
          <Ponctual>
            <Weight>1</Weight>
            <Location>0</Location>
          </Ponctual>
        </Elements>
      </Pattern>
    </AffectationValue>
    <AffectationValue>
      <Bounds>
        <Beginning>2013-01-01T00:00:00</Beginning>
        <End>2014-01-01T00:00:00</End>
      </Bounds>
      <Value>1000</Value>
      <Pattern i:type="PonctualPattern">
        <Bounds>
          <Beginning>0</Beginning>
          <End>1</End>
        </Bounds>
        <Elements>
          <Ponctual>
            <Weight>1</Weight>
            <Location>0</Location>
          </Ponctual>
        </Elements>
      </Pattern>
    </AffectationValue>
    <AffectationValue>
      <Bounds>
        <Beginning>2014-01-01T00:00:00</Beginning>
        <End>2015-01-01T00:00:00</End>
      </Bounds>
      <Value>1000</Value>
      <Pattern i:type="PonctualPattern">

```

```

<Bounds>
  <Beginning>0</Beginning>
  <End>1</End>
</Bounds>
<Elements>
  <Ponctual>
    <Weight>1</Weight>
    <Location>0</Location>
  </Ponctual>
</Elements>
</Pattern>
</AffectionValue>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Entretien)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Entretien)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Amortissement des investissements)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Amortissement des investissements)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Remboursement en intérêt)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Remboursement en intérêt)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Recettes de fonctionnement)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Recettes de fonctionnement)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Recettes de fonctionnement hors participation de la
mairie)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Recettes de fonctionnement hors participation de la mairie)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Participation des familles)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Participation des familles)</VariableIdentifier>
</DatedValues>
<AffectionValue>
  <Bounds>
    <Beginning>2012-09-01T00:00:00</Beginning>
    <End>2013-09-01T00:00:00</End>
  </Bounds>
  <Value>60000</Value>
  <Pattern i:type="ShapePattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>12</End>
    </Bounds>
    <Elements>
      <Shape i:type="Constant">
        <End>1</End>
        <Height>0.25</Height>
      </Shape>
      <Shape i:type="Constant">
        <End>10</End>
        <Height>0.083333333333333329</Height>
      </Shape>
    </Elements>
  </Pattern>
</AffectionValue>
<AffectionValue>
  <Bounds>
    <Beginning>2013-09-01T00:00:00</Beginning>
    <End>2014-09-01T00:00:00</End>
  </Bounds>
  <Value>60000</Value>
  <Pattern i:type="ShapePattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>12</End>
    </Bounds>
    <Elements>
      <Shape i:type="Constant">
        <End>1</End>
        <Height>0.25</Height>
      </Shape>
      <Shape i:type="Constant">
        <End>10</End>
        <Height>0.083333333333333329</Height>
      </Shape>
    </Elements>
  </Pattern>
</AffectionValue>
<AffectionValue>
  <Bounds>
    <Beginning>2014-09-01T00:00:00</Beginning>
    <End>2015-09-01T00:00:00</End>
  </Bounds>
  <Value>60000</Value>
  <Pattern i:type="ShapePattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>12</End>
    </Bounds>
    <Elements>
      <Shape i:type="Constant">
        <End>1</End>
        <Height>0.25</Height>
      </Shape>
      <Shape i:type="Constant">
        <End>10</End>
        <Height>0.083333333333333329</Height>
      </Shape>
    </Elements>
  </Pattern>
</AffectionValue>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Subventions de fonctionnement)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Subventions de fonctionnement)</VariableIdentifier>
</DatedValues>

```

```

<AffectationValue>
  <Bounds>
    <Beginning>2012-01-01T00:00:00</Beginning>
    <End>2013-01-01T00:00:00</End>
  </Bounds>
  <Value>48000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>1</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>0</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2013-01-01T00:00:00</Beginning>
    <End>2014-01-01T00:00:00</End>
  </Bounds>
  <Value>48000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>1</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>0</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
<AffectationValue>
  <Bounds>
    <Beginning>2014-01-01T00:00:00</Beginning>
    <End>2015-01-01T00:00:00</End>
  </Bounds>
  <Value>48000</Value>
  <Pattern i:type="PonctualPattern">
    <Bounds>
      <Beginning>0</Beginning>
      <End>1</End>
    </Bounds>
    <Elements>
      <Ponctual>
        <Weight>1</Weight>
        <Location>0</Location>
      </Ponctual>
    </Elements>
  </Pattern>
</AffectationValue>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Participation à l'équilibre du budget général)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Participation à l'équilibre du budget général)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Prix d'un ticket)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Prix d'un ticket)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Coût en denrées d'un repas)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Coût en denrées d'un repas)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Quantité de tickets vendus)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Quantité de tickets vendus)</VariableIdentifier>
</DatedValues>
</VariableInstance>
<Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)/Variables(Nombre de repas servis)</Identifier>
<VariableIdentifier>Models(Cantine)/Models(Fonctionnement)/Variables(Nombre de repas servis)</VariableIdentifier>
</DatedValues>
</VariableInstance>
</VariableInstances>
<ChildrenModelInstances/>
<OutAssociations>
  <Association>
    <LinkIdentifier>Models(Cantine)/Models(Fonctionnement)/Links(Investissement)</LinkIdentifier>
    <ModelInstanceIdentifier>ModelInstances(Budget cantine)/ModelInstances(Investissement)</ModelInstanceIdentifier>
  </Association>
</OutAssociations>
</ModelInstance>
<ModelInstance>
  <Name>Fonctionnement avec déménagement</Name>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)</Identifier>
  <ModelIdentifier>Models(Cantine)/Models(Fonctionnement avec déménagement)</ModelIdentifier>
  <IsActive>true</IsActive>
  <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
</ActiveModelIdentifiers>
  <ScalarVariableInstances>
    <ScalarVariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/ScalarVariables(Déménagement)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/ScalarVariables(Déménagement)</VariableIdentifier>
      <Value>2016-09-01T00:00:00</Value>
    </ScalarVariableInstance>
  </ScalarVariableInstances>
</ScalarVariableInstances>
<VariableInstances>
  <VariableInstance>
    <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/Variables(Dépense en denrées avant)</Identifier>
    <VariableIdentifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Dépense en denrées avant)</VariableIdentifier>
    </DatedValues>
  </VariableInstance>
  <VariableInstance>
    <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/Variables(Charges salariales avant)</Identifier>
    <VariableIdentifier>Models(Cantine)/Models(Fonctionnement avec déménagement)/Variables(Charges salariales avant)</VariableIdentifier>
    </DatedValues>
  </VariableInstance>
  <VariableInstance>
    <Identifier>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/Variables(Consommations avant)</Identifier>

```



```

    <VariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/Variables(Participation à l'équilibre du budget
général)</Identifiant>
    </VariableInstance>
  </ModelInstances>
  <ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/Variables(Participation à l'équilibre du budget
général)</ModelInstances>
  <ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/Variables(Prix d'un ticket)</ModelInstances>
  <ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/Variables(Coût en denrées d'un repas)</ModelInstances>
  <ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/Variables(Quantité de tickets vendus)</ModelInstances>
  <ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avec déménagement)/Variables(Nombre de repas servis)</ModelInstances>
  <ChildrenModelInstances/>
  <OutAssociations>
    <Association>
      <LinkIdentifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Links(Investissement)</LinkIdentifiant>
      <ModelInstanceIdentifiant>ModelInstances(Budget cantine)/ModelInstances(Investissement)</ModelInstanceIdentifiant>
    </Association>
    <Association>
      <LinkIdentifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Links(Fonctionnement avant)</LinkIdentifiant>
      <ModelInstanceIdentifiant>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement avant)</ModelInstanceIdentifiant>
    </Association>
    <Association>
      <LinkIdentifiant>Models(Cantine)/Models(Fonctionnement avec déménagement)/Links(Fonctionnement après)</LinkIdentifiant>
      <ModelInstanceIdentifiant>ModelInstances(Budget cantine)/ModelInstances(Fonctionnement après)</ModelInstanceIdentifiant>
    </Association>
  </OutAssociations>
  </ModelInstance>
  <ModelInstance>
    <Name>Emprunt</Name>
    <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)</Identifiant>
    <ModelIdentifiant>Models(Cantine)/Models(Emprunt)</ModelIdentifiant>
    <IsActive>true</IsActive>
    <ActiveModelIdentifiants xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
    <ScalarVariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/ScalarVariables(Taux d'intérêts)</Identifiant>
      <VariableIdentifiant>Models(Cantine)/Models(Emprunt)/ScalarVariables(Taux d'intérêts)</VariableIdentifiant>
      <Value>0.05</Value>
    </ScalarVariableInstance>
    <ScalarVariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/ScalarVariables(Levée)</Identifiant>
      <VariableIdentifiant>Models(Cantine)/Models(Emprunt)/ScalarVariables(Levée)</VariableIdentifiant>
      <Value>2012-01-01T00:00:00</Value>
    </ScalarVariableInstance>
    <ScalarVariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/ScalarVariables(Durée de différé de remboursement)</Identifiant>
      <VariableIdentifiant>Models(Cantine)/Models(Emprunt)/ScalarVariables(Durée de différé de remboursement)</VariableIdentifiant>
      <Value>0</Value>
    </ScalarVariableInstance>
    <ScalarVariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/ScalarVariables(Durée avant remboursement)</Identifiant>
      <VariableIdentifiant>Models(Cantine)/Models(Emprunt)/ScalarVariables(Durée avant remboursement)</VariableIdentifiant>
      <Value>0</Value>
    </ScalarVariableInstance>
    <ScalarVariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/ScalarVariables(Durée de remboursement)</Identifiant>
      <VariableIdentifiant>Models(Cantine)/Models(Emprunt)/ScalarVariables(Durée de remboursement)</VariableIdentifiant>
      <Value>3600</Value>
    </ScalarVariableInstance>
  </ScalarVariableInstances>
  <VariableInstances>
    <VariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Capital emprunté)</Identifiant>
      <VariableIdentifiant>Models(Cantine)/Models(Emprunt)/Variables(Capital emprunté)</VariableIdentifiant>
      <DatedValues>
        <AffectationValue>
          <Bounds>
            <Beginning>2015-10-01T00:00:00</Beginning>
            <End>2015-11-01T00:00:00</End>
          </Bounds>
          <Value>145000</Value>
          <Pattern i:type="PonctualPattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>1</End>
            </Bounds>
            <Elements>
              <Ponctual>
                <Weight>1</Weight>
                <Location>0</Location>
              </Ponctual>
            </Elements>
          </Pattern>
        </AffectationValue>
      </DatedValues>
    </VariableInstance>
    <VariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Montant de l'échéance)</Identifiant>
      <VariableIdentifiant>Models(Cantine)/Models(Emprunt)/Variables(Montant de l'échéance)</VariableIdentifiant>
      <DatedValues/>
    </VariableInstance>
    <VariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Indice Taux variables)</Identifiant>
      <VariableIdentifiant>Models(Cantine)/Models(Emprunt)/Variables(Indice Taux variables)</VariableIdentifiant>
      <DatedValues/>
    </VariableInstance>
    <VariableInstance>
      <Identifiant>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Périodicité)</Identifiant>
      <VariableIdentifiant>Models(Cantine)/Models(Emprunt)/Variables(Périodicité)</VariableIdentifiant>
      <DatedValues/>
    </VariableInstance>
  </VariableInstances>

```

```

<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Intérêts courus non échus)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Emprunt)/Variables(Intérêts courus non échus)</VariableIdentifieur>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Date de fin spécifiée)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Emprunt)/Variables(Date de fin spécifiée)</VariableIdentifieur>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Jour de la première échéance)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Emprunt)/Variables(Jour de la première échéance)</VariableIdentifieur>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Remboursement en intérêt)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Emprunt)/Variables(Remboursement en intérêt)</VariableIdentifieur>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Valeur du différé de remboursement)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Emprunt)/Variables(Valeur du différé de remboursement)</VariableIdentifieur>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Remboursement en capital)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Emprunt)/Variables(Remboursement en capital)</VariableIdentifieur>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(Capital restant dû)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Emprunt)/Variables(Capital restant dû)</VariableIdentifieur>
<DatedValues/>
</VariableInstance>
<VariableInstance>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Emprunt)/Variables(LinkTo_Indice taux variables)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Emprunt)/Variables(LinkTo_Indice taux variables)</VariableIdentifieur>
<DatedValues/>
</VariableInstance>
</VariableInstances>
<ChildrenModelInstances/>
<OutAssociations/>
</ModelInstance>
<ModelInstance>
<Name>Opération de maçonnerie</Name>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Opération de maçonnerie)</Identifieur>
<ModelIdentifieur>Models(Cantine)/Models(Opération)</ModelIdentifieur>
<IsActive>true</IsActive>
<ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays/">
<ScalarVariableInstances/>
<VariableInstances>
<VariableInstance>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Opération de maçonnerie)/Variables(Dépenses)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Opération)/Variables(Dépenses)</VariableIdentifieur>
<DatedValues>
<AffectationValue>
<Bounds>
<Beginning>2015-11-01T00:00:00</Beginning>
<End>2015-11-02T00:00:00</End>
</Bounds>
<Value>50000</Value>
<Pattern i:type="PonctualPattern">
<Bounds>
<Beginning>0</Beginning>
<End>1</End>
</Bounds>
<Elements>
<Ponctual>
<Weight>1</Weight>
<Location>0</Location>
</Ponctual>
</Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
<Bounds>
<Beginning>2015-12-01T00:00:00</Beginning>
<End>2015-12-02T00:00:00</End>
</Bounds>
<Value>50000</Value>
<Pattern i:type="PonctualPattern">
<Bounds>
<Beginning>0</Beginning>
<End>1</End>
</Bounds>
<Elements>
<Ponctual>
<Weight>1</Weight>
<Location>0</Location>
</Ponctual>
</Elements>
</Pattern>
</AffectationValue>
<AffectationValue>
<Bounds>
<Beginning>2016-01-01T00:00:00</Beginning>
<End>2016-01-02T00:00:00</End>
</Bounds>
<Value>50000</Value>
<Pattern i:type="PonctualPattern">
<Bounds>
<Beginning>0</Beginning>
<End>1</End>
</Bounds>
<Elements>
<Ponctual>
<Weight>1</Weight>
<Location>0</Location>
</Ponctual>
</Elements>
</Pattern>
</AffectationValue>
</DatedValues>
</VariableInstance>
<VariableInstance>
<Identifieur>ModelInstances(Budget cantine)/ModelInstances(Opération de maçonnerie)/Variables(Amortissement)</Identifieur>
<VariableIdentifieur>Models(Cantine)/Models(Opération)/Variables(Amortissement)</VariableIdentifieur>
<DatedValues/>
</VariableInstance>

```

```

</VariableInstances>
<ChildrenModelInstances/>
<OutAssociations>
  <Association>
    <LinkIdentifier>Models(Cantine)/Models(Opération)/Links(Amortissement)</LinkIdentifier>
    <ModelInstanceIdentifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de la maçonnerie)</ModelInstanceIdentifier>
  </Association>
</OutAssociations>
</ModelInstance>
<ModelInstance>
  <Name>Amortissement de la maçonnerie</Name>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de la maçonnerie)</Identifier>
  <ModelIdentifier>Models(Cantine)/Models(Amortissement)</ModelIdentifier>
  <IsActive>true</IsActive>
  <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <ScalarVariableInstances>
    <ScalarVariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de la maçonnerie)/ScalarVariables(Début)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Amortissement)/ScalarVariables(Début)</VariableIdentifier>
      <Value>2016-01-01T00:00:00</Value>
    </ScalarVariableInstance>
    <ScalarVariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de la maçonnerie)/ScalarVariables(Durée)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Amortissement)/ScalarVariables(Durée)</VariableIdentifier>
      <Value>7200</Value>
    </ScalarVariableInstance>
  </ScalarVariableInstances>
  <VariableInstances>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de la maçonnerie)/Variables(Capital à amortir)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Amortissement)/Variables(Capital à amortir)</VariableIdentifier>
      <DatedValues>
        <AffectationValue>
          <Bounds>
            <Beginning>2016-01-01T00:00:00</Beginning>
            <End>2016-02-01T00:00:00</End>
          </Bounds>
          <Value>150000</Value>
          <Pattern i:type="PonctualPattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>1</End>
            </Bounds>
            <Elements>
              <Ponctual>
                <Weight>1</Weight>
                <Location>0</Location>
              </Ponctual>
            </Elements>
          </Pattern>
        </AffectationValue>
      </DatedValues>
    </VariableInstance>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de la maçonnerie)/Variables(Capital amorti)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Amortissement)/Variables(Capital amorti)</VariableIdentifier>
      <DatedValues/>
    </VariableInstance>
  </VariableInstances>
</ChildrenModelInstances/>
<OutAssociations/>
</ModelInstance>
<ModelInstance>
  <Name>Opération d'isolation</Name>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Opération d'isolation)</Identifier>
  <ModelIdentifier>Models(Cantine)/Models(Opération)</ModelIdentifier>
  <IsActive>true</IsActive>
  <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <ScalarVariableInstances/>
  <VariableInstances>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Opération d'isolation)/Variables(Dépenses)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Opération)/Variables(Dépenses)</VariableIdentifier>
      <DatedValues>
        <AffectationValue>
          <Bounds>
            <Beginning>2016-02-08T00:00:00</Beginning>
            <End>2016-02-09T00:00:00</End>
          </Bounds>
          <Value>60000</Value>
          <Pattern i:type="PonctualPattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>1</End>
            </Bounds>
            <Elements>
              <Ponctual>
                <Weight>1</Weight>
                <Location>0</Location>
              </Ponctual>
            </Elements>
          </Pattern>
        </AffectationValue>
        <AffectationValue>
          <Bounds>
            <Beginning>2016-02-15T00:00:00</Beginning>
            <End>2016-02-16T00:00:00</End>
          </Bounds>
          <Value>60000</Value>
          <Pattern i:type="PonctualPattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>1</End>
            </Bounds>
            <Elements>
              <Ponctual>
                <Weight>1</Weight>
                <Location>0</Location>
              </Ponctual>
            </Elements>
          </Pattern>
        </AffectationValue>
      </DatedValues>
    </VariableInstance>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Opération d'isolation)/Variables(Amortissement)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Opération)/Variables(Amortissement)</VariableIdentifier>
      <DatedValues/>
    </VariableInstance>
  </VariableInstances>

```

```

</VariableInstances>
<ChildrenModelInstances/>
<OutAssociations>
  <Association>
    <LinkIdentifier>Models(Cantine)/Models(Opération)/Links(Amortissement)</LinkIdentifier>
    <ModelInstanceIdentifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de l'isolation)</ModelInstanceIdentifier>
  </Association>
</OutAssociations>
</ModelInstance>
<ModelInstance>
  <Name>Amortissement de l'isolation</Name>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de l'isolation)</Identifier>
  <ModelIdentifier>Models(Cantine)/Models(Amortissement)</ModelIdentifier>
  <IsActive>true</IsActive>
  <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <ScalarVariableInstances>
    <ScalarVariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de l'isolation)/ScalarVariables(Début)</Identifier>
      <VariableIdentifier>Models(Amortissement)/ScalarVariables(Début)</VariableIdentifier>
      <Value>2016-01-01T00:00:00</Value>
    </ScalarVariableInstance>
    <ScalarVariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de l'isolation)/ScalarVariables(Durée)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Amortissement)/ScalarVariables(Durée)</VariableIdentifier>
      <Value>7200</Value>
    </ScalarVariableInstance>
  </ScalarVariableInstances>
  <VariableInstances>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement de l'isolation)/Variables(Capital à amortir)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Amortissement)/Variables(Capital à amortir)</VariableIdentifier>
      <DatedValues>
        <AffectationValue>
          <Bounds>
            <Beginning>2016-02-15T00:00:00</Beginning>
            <End>2016-02-16T00:00:00</End>
          </Bounds>
          <Value>120000</Value>
          <Pattern i:type="PonctualPattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>1</End>
            </Bounds>
            <Elements>
              <Ponctual>
                <Weight>1</Weight>
                <Location>0</Location>
              </Ponctual>
            </Elements>
          </Pattern>
        </AffectationValue>
      </DatedValues>
    </VariableInstance>
  </VariableInstances>
</ModelInstance>
<ModelInstance>
  <Name>Opération de chauffage</Name>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Opération de chauffage)</Identifier>
  <ModelIdentifier>Models(Cantine)/Models(Opération)</ModelIdentifier>
  <IsActive>true</IsActive>
  <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <ScalarVariableInstances/>
  <VariableInstances>
    <VariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Opération de chauffage)/Variables(Dépenses)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Opération)/Variables(Dépenses)</VariableIdentifier>
      <DatedValues>
        <AffectationValue>
          <Bounds>
            <Beginning>2016-03-01T00:00:00</Beginning>
            <End>2017-08-01T00:00:00</End>
          </Bounds>
          <Value>80000</Value>
          <Pattern i:type="PonctualPattern">
            <Bounds>
              <Beginning>0</Beginning>
              <End>1</End>
            </Bounds>
            <Elements>
              <Ponctual>
                <Weight>1</Weight>
                <Location>0</Location>
              </Ponctual>
            </Elements>
          </Pattern>
        </AffectationValue>
      </DatedValues>
    </VariableInstance>
  </VariableInstances>
  <ChildrenModelInstances/>
  <OutAssociations>
    <Association>
      <LinkIdentifier>Models(Cantine)/Models(Opération)/Links(Amortissement)</LinkIdentifier>
      <ModelInstanceIdentifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement du chauffage)</ModelInstanceIdentifier>
    </Association>
  </OutAssociations>
</ModelInstance>
<ModelInstance>
  <Name>Amortissement du chauffage</Name>
  <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement du chauffage)</Identifier>
  <ModelIdentifier>Models(Cantine)/Models(Amortissement)</ModelIdentifier>
  <IsActive>true</IsActive>
  <ActiveModelIdentifiers xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <ScalarVariableInstances>
    <ScalarVariableInstance>
      <Identifier>ModelInstances(Budget cantine)/ModelInstances(Amortissement du chauffage)/ScalarVariables(Début)</Identifier>
      <VariableIdentifier>Models(Cantine)/Models(Amortissement)/ScalarVariables(Début)</VariableIdentifier>

```

```

<Value>2016-01-01T00:00:00</Value>
</ScalarVariableInstance>
<ScalarVariableInstance>
<Identifiant>ModelInstances(Budget cantine)/ModelInstances(Amortissement du chauffage)/ScalarVariables(Durée)</Identifiant>
<VariableIdentifiant>Models(Cantine)/Models(Amortissement)/ScalarVariables(Durée)</VariableIdentifiant>
<Value>7200</Value>
</ScalarVariableInstance>
</ScalarVariableInstances>
<VariableInstances>
<VariableInstance>
<Identifiant>ModelInstances(Budget cantine)/ModelInstances(Amortissement du chauffage)/Variables(Capital à amortir)</Identifiant>
<VariableIdentifiant>Models(Cantine)/Models(Amortissement)/Variables(Capital à amortir)</VariableIdentifiant>
<DatedValues>
<AffectationValue>
<Bounds>
<Beginning>2016-03-01T00:00:00</Beginning>
<End>2016-04-01T00:00:00</End>
</Bounds>
<Value>80000</Value>
<Pattern i:type="PonctualPattern">
<Bounds>
<Beginning>0</Beginning>
<End>1</End>
</Bounds>
<Elements>
<Ponctual>
<Weight>1</Weight>
<Location>0</Location>
</Ponctual>
</Elements>
</Pattern>
</AffectationValue>
</DatedValues>
</VariableInstance>
<VariableInstance>
<Identifiant>ModelInstances(Budget cantine)/ModelInstances(Amortissement du chauffage)/Variables(Capital amorti)</Identifiant>
<VariableIdentifiant>Models(Cantine)/Models(Amortissement)/Variables(Capital amorti)</VariableIdentifiant>
<DatedValues/>
</VariableInstance>
</VariableInstances>
<ChildrenModelInstances/>
<OutAssociations/>
</ModelInstance>
</ChildrenModelInstances>
<OutAssociations/>
</ModelInstance>
</World>

```