

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

VERS DES SYSTÈMES D'APPRENTISSAGES SENSIBLES AU  
CONTEXTE : DÉVELOPPEMENT D'UN SOUS-SYSTÈME DE  
MODÉLISATION ET D'ESTIMATION D'ÉCART DE CONTEXTES POUR  
LA PRISE EN CHARGE DES EFFETS DE CONTEXTES

MÉMOIRE  
PRÉSENTÉ  
COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN INFORMATIQUE

PAR  
Wafa FENNANI

MAI 2020

UNIVERSITÉ DU QUÉBEC À MONTRÉAL  
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.10-2015). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»



## TABLE DES MATIÈRES

LISTE DES TABLEAUX . . . . .	vii
LISTE DES FIGURES . . . . .	ix
RÉSUMÉ . . . . .	xiii
CHAPITRE I INTRODUCTION . . . . .	1
1.1 Problématique . . . . .	2
1.2 Hypothèse de solution . . . . .	4
1.3 Objectif . . . . .	5
1.4 Méthodologie . . . . .	6
1.5 Démarche et organisation de la mémoire . . . . .	7
CHAPITRE II ÉTUDE THÉORIQUE DE CONTEXTE . . . . .	9
2.1 Contexte . . . . .	10
2.1.1 Évolution chronologique de définition de contexte . . . . .	10
2.1.2 Définition selon le domaine d'application . . . . .	12
2.2 Les systèmes sensibles au contexte . . . . .	14
2.3 Contexte interne et Contexte externe . . . . .	16
2.4 Les effets de contexte . . . . .	17
2.5 Les caractéristiques d'un contexte . . . . .	18
2.6 La qualité d'un contexte . . . . .	20
2.7 Le modèle de contexte . . . . .	22
2.8 Le raisonnement contextuel . . . . .	24
CHAPITRE III ÉTAT DE L'ART SUR L'OPÉRATIONNALISATION DU CONTEXTE . . . . .	27
3.1 Les catégories de contexte . . . . .	27

CHAPITRE IV SOUTENIR LE MODÈLE CLASH PAR DES OUTILS INFORMATIQUES (MAZCALC ET CAITS) . . . . .	35
4.1 Le modèle CLASH . . . . .	35
4.2 Méthodologie . . . . .	37
4.3 CAITS . . . . .	41
4.3.1 MazCalc . . . . .	42
CHAPITRE V ANALYSE CONCEPTUELLE DE MAZCALC . . . . .	45
5.1 Méthodologie AGILE . . . . .	45
5.2 MazCalc : vue Globale . . . . .	48
5.2.1 Composant Acteurs . . . . .	50
5.2.2 Diagramme de cas d'utilisation global . . . . .	52
5.3 Analyse des composants . . . . .	53
5.3.1 Le composant Modèles . . . . .	55
5.3.2 Composant Contextes . . . . .	57
5.3.3 Composant Règles de Calcul . . . . .	59
5.4 Besoins non fonctionnels . . . . .	61
5.5 étude conceptuelle dynamique . . . . .	63
5.5.1 diagrammes de séquences . . . . .	63
5.6 Diagramme de classes et Schéma de Base de données . . . . .	69
5.7 Architecture du systèmes MazCalc . . . . .	70
CHAPITRE VI CHOIX TECHNIQUES ET IMPLÉMENTATION . . . . .	75
6.1 Aspects Techniques . . . . .	75
6.1.1 Paradigme . . . . .	75
6.1.2 .NET, ASP.NET et ASP.NET Core . . . . .	77
6.1.3 Environnements logiciels : Visual Studio et MS SQLServer . . . . .	78
6.1.4 Entity Framework . . . . .	79
6.1.5 Gestion de version du code (GIT) . . . . .	79

6.1.6	DDD : Domain Driven Design . . . . .	81
6.1.7	TDD : Test Driven Development . . . . .	81
6.1.8	Patrons de conception . . . . .	83
6.1.9	SOLID . . . . .	83
6.1.10	Clean Architecture . . . . .	84
6.1.11	REST API . . . . .	87
6.1.12	OpenApi pour documentation API . . . . .	90
6.1.13	Sécurité . . . . .	90
6.2	Algorithmes de calcul d'écart . . . . .	92
6.3	Implémentions de MazCalc . . . . .	92
6.3.1	Les interfaces modèle . . . . .	92
6.4	Tests et Résultats . . . . .	98
6.4.1	La géothermie . . . . .	99
6.4.2	Création du modèle . . . . .	101
6.4.3	Instanciation des contextes . . . . .	105
	CONCLUSION . . . . .	115
	RÉFÉRENCES . . . . .	123



## LISTE DES TABLEAUX

Tableau	Page
6.1 Tableau des projets MazCalc et leurs équivalences dans l'architecture propre . . . . .	88



## LISTE DES FIGURES

Figure	Page
4.1 Le modèle CLASH : Interactions entre les contextes d'après (Forissier <i>et al.</i> , 2014b) . . . . .	36
4.2 déroulement d'un scénario d'apprentissage donné par (Forissier, 2019)	39
4.3 Déroulement de DBR de TEEC . . . . .	40
4.4 Architecture de CAITS donnée par (Forissier <i>et al.</i> , 2014b) . . .	42
5.1 le processus de la Méthodologie Scrum (Schwaber, 1997) . . . . .	47
5.2 Les composants du Système MazCalc . . . . .	49
5.3 Les rôles des Acteurs utilisant MazCalc . . . . .	51
5.4 Cas d'utilisation global . . . . .	53
5.5 Diagramme de cas d'utilisation du Concepteur du modèle . . . . .	56
5.6 Diagramme de cas d'utilisation de l'expert du modèle dans son contexte . . . . .	58
5.7 diagramme de cas d'utilisation de l'enseignant . . . . .	59
5.8 diagramme de séquence pour l'authentification des acteurs . . . . .	64
5.9 Fixation d'un objet d'étude par un enseignant . . . . .	65
5.10 Ajouter un paramètre à un objet d'étude . . . . .	66
5.11 Ajouter une valeur à un paramètre d'un objet d'étude . . . . .	68
5.12 Diagramme de séquence : définition du contexte d'un objet d'étude	69
5.13 Diagramme de classes . . . . .	70
5.14 Schéma de la Base de données MazCalc . . . . .	71
5.15 Architecture global du MazCalc . . . . .	72

6.1	Historique des changements du code MazCalc . . . . .	80
6.2	Résultats de test automatique dans MazCalc . . . . .	82
6.3	Diagramme de l'architecture propre donnée par (Martin, 2017) . .	85
6.4	MazCalc- Clean architecture . . . . .	87
6.5	Interface de définition d'un nouveau objet d'étude . . . . .	93
6.6	Interface de définition des paramètres de l'objet d'étude . . . . .	94
6.7	Interface de définition des valeurs d'un paramètre quantitatif . . .	95
6.8	Interface de définition des valeurs d'un paramètre composé . . . .	95
6.9	Interface de définition des valeurs énumérées d'un paramètre qualitatif	96
6.10	Interface de définition des valeurs ordonnées d'un paramètre qualitatif	97
6.11	Interface d'instanciation d'un objet d'étude dans un contexte . . .	97
6.12	Interface de choix de contextes de Géothermie à comparer . . . .	98
6.13	Liste des paramètres de l'objet d'étude Géothermie donnée par l'ex- pert . . . . .	100
6.14	Groupement par famille des paramètres de la Géothermie . . . . .	101
6.15	La géothermie telle que définit au Québec et à Guadeloupe . . . .	102
6.16	interface présentant l'objet d'étude fixé par un enseignant . . . . .	102
6.17	Interface d'attribution des rôles à un acteur . . . . .	103
6.18	Interface de la liste des acteurs et leurs rôles . . . . .	104
6.19	Interface de définition des familles de groupement des paramètres Géothermie . . . . .	105
6.20	Interface de liste des paramètres de l'objet d'étude Géothermie . .	106
6.21	Interface de la liste des valeurs du paramètre « Flux de la chaleur » de type quantitatif . . . . .	106
6.22	Interface de la liste des valeurs du paramètre « Type de formation » de type qualitatif énuméré . . . . .	107

6.23	Interface de la liste des valeurs du paramètre « Type de roche » de type Composé . . . . .	108
6.24	La liste des valeurs du paramètre « Âge de formation » de type qualitatif Ordonné . . . . .	109
6.25	Interface de Détails du modèle Géothermie et le groupement de ses paramètres par famille . . . . .	110
6.26	Interface de La liste des contextes distanciées de l'objet d'étude Géothermie . . . . .	111
6.27	Résultats numériques de calcul d'écart entre le contexte Québec et le contexte Guadeloupe . . . . .	112
6.28	Le Calcul d'écart tel que donné par l'expert de Géothermie Clair Anjou . . . . .	113
6.29	Illustration graphique des résultats de calcul d'écart entre le contexte Québec et le contexte Guadeloupe . . . . .	113



## RÉSUMÉ

Dans l'apprentissage des sciences, le contexte est une dimension importante de tout objet ou phénomène scientifique ainsi que ses variations. L'environnement (le contexte externe) prend part à la construction des conceptions mentales des apprenants. En effet, une collaboration entre des apprenants issus d'environnements différents permet une confrontation de leurs conceptions. La modélisation du contexte, est donc essentielle pour identifier les paramètres de contexte qui agissent sur ces conceptions. Dans nos études, nous présentons un projet de recherche en science de l'éducation qui se situe à l'intersection de l'informatique et d'enseignement en contexte. L'objectif ultime est d'améliorer le processus d'apprentissage en créant un système tutoriel intelligent (STI) qui permet d'une part, la réalisation et la conception des scénarios d'apprentissage en fonction des effets de contexte, et d'autre part, montrer qu'un apprentissage sensible au contexte est une méthode plus efficace et plus constructive pour le développement des conceptions mentales des apprenants. Pour atteindre cet objectif ultime, il est important de se doter d'une solution efficace pour la modélisation du contexte et l'estimation des effets de contextes : tels sont les deux objectifs principaux de ce travail de maîtrise.

Pour qu'un système soit sensible au contexte, il doit intégrer un système gestionnaire de contexte générique que nous avons appelé MazCalc. MazCalc permet de modéliser les contextes relativement à un objet d'étude donné, et de calculer l'écart entre ces contextes afin d'identifier les paramètres contextuels permettant de favoriser l'apparition des effets de contexte. L'implémentation d'un tel système nécessite la contribution de plusieurs experts des domaines tel que des experts didactiques, des experts de contexte, des experts des objet d'étude, des experts des système tutoriel intelligent... etc. MazCalc sera le cœur du STI qui lui fournira de toutes les informations contextuelles nécessaires pour élaborer les scénarios d'apprentissage.

Nous présentons dans ce mémoire la motivation de notre projet de recherche, ainsi qu'une étude théorique du domaine contextuel. Nous détaillons la démarche de d'implémentation de MazCalc qui débute par une analyse conceptuelle de nos besoins. Nous exposons par la suite les composants de notre système et leurs fonctionnements, ainsi que l'architecture globale de MazCalc. Cette démarche sera clôturée par une présentation de l'implémentation de ces composants. Une série

de tests sera effectuer sur les composants du système en utilisant un exemple réel d'objet d'étude et deux contextes bien étudiés et validés par un expert du domaine.

**Mots Clés** : modèle de contexte, système sensible au contexte, effets de contexte, système tutoriel intelligent, écart de contextes, Contexte de géolocalisation

## CHAPITRE I

### INTRODUCTION

L'objectif de ce mémoire est de présenter le travail réalisé dans le cadre d'un projet de recherche au sein du laboratoire de recherche en Gestion, Diffusion et Acquisition des Connaissances « GDAC » à l'UQAM avec la collaboration d'une équipe pluridisciplinaire du CRREF (Université des Antilles) et du LICEF (TELUQ). Le projet s'intitule TEEC « Technologies Éducatives pour l'Enseignement en Contexte », financé dans le cadre d'un ANR France-Québec impliquant l'UQAM, la TÉLUQ, l'Université des Antilles et l'Université de Paris VI. L'objectif est d'instrumenter et de tester une innovation didactique basée sur la confrontation à distance de contextes. Notre rôle dans ce projet visait essentiellement l'élaboration et l'implémentation d'un moteur de calcul d'écart entre des contextes visant un même objet d'étude de sorte à rendre explicite les différences importantes pouvant mener à un effet de contexte lors des confrontations entre les apprenants de différents contextes. Le but étant de disposer d'un outil capable d'aider les enseignants humains ou des systèmes tutoriels intelligents à tenir compte du contexte dans des activités pédagogiques qui l'exigeraient.

## 1.1 Problématique

Dans l'apprentissage des sciences, le contexte est une dimension importante pour tout objet ou phénomène scientifique. Les variations liées au contexte peuvent être aussi critiques pour une compréhension approfondie que les connaissances en jeu (concepts, règles, etc.) (Forissier *et al.*, 2014b). L'étude du contexte a suscité l'intérêt de la communauté scientifique, notamment en didactique des sciences, avec différentes perspectives et angles d'études.

En effet, certaines études se sont intéressées à l'analyse du phénomène de contextualisation dans l'apprentissage. Alors que d'autres se sont focalisées sur la sociologie de l'éducation et l'impact de ses facteurs telle que la classe sociale (Wilson, 1959) ou le sexe (Weinburgh, 1995) sur le choix de la discipline et sur les résultats scolaires des étudiants. D'autres recherches se sont axées sur la psychologie scolaire et sur le développement des représentations mentales des apprenants dans les processus d'apprentissage (Oers, 1998), c-à-d le contexte interne. C'est dans cette perspective que la notion de contexte environnemental a pris sa place : une vision du contexte reliée à l'impact écologique et géographique (Podschuweit et Bernholt, 2018) dans des situations authentiques d'apprentissage (Schwartz *et al.*, 2004), (King *et al.*, 2011), (van Eijck et Roth, 2010).

Le contexte environnemental est devenu une composante très importante dans la plupart des systèmes afin d'offrir un apprentissage plus adaptatif. Certains de ces systèmes utilisent les informations de localisation pour fournir des activités d'apprentissage omniprésent, permettant aux apprenants d'observer et de collecter des données (Rogers *et al.*, 2005). D'autres considèrent la proximité entre les apprenants pour soutenir l'apprentissage collaboratif (Lonsdale *et al.*, 2005). L'ensemble de ces systèmes repose sur une adaptation purement opérationnelle de la notion de contexte qui vise l'efficacité calculatoire du système.

Cependant, cette considération opérationnelle ne permet pas de résoudre la dimension sémantique dans laquelle l'objet d'étude est fondamentalement ancré au contexte. On parle alors de contexte immédiat.

En effet, les apprenants acquièrent des concepts (sous forme générale) en se basant seulement sur leurs propres contextes sans prendre en considération des contextes d'autrui. Le contexte immédiat est considéré comme allant de soi et généralisable. Ceci est très apparent surtout dans les dimensions géographiques et culturelles du contexte (Forissier *et al.*, 2014b).

Un parfait exemple en est l'étude de l'adaptation de l'espèce à son environnement, l'étude de la grenouille. Une des définitions les plus connues de la grenouille est comme suit (Forissier *et al.*, 2014b) : «C'est un animal de couleur verte, d'environ 10 cm de taille. Elle a les pieds palmés, et produit un son qui ressemble à "ribbit ribbit" vivant dans un étang où poussent les têtards au printemps». Cependant, cette description s'avère être différente pour les apprenants de deux contextes différents, entre autres au Québec (Canada) et en Guadeloupe. En Guadeloupe, la grenouille la plus commune est appelée siffleur ou *Euleutherodactylus*, elle est omniprésente dans les maisons et les jardins, de couleur brune ou jaune, c'est la plus petite grenouille au monde dont la taille est moins de 3 cm, ses pattes ne sont pas palmées, elle pond des œufs sur le sol où se forme de petites grenouilles. Le soir, tout au long de l'année, elle produit un chant strident similaire à "tui tui" (Bourdeau *et al.*, 2015), (Forissier *et al.*, 2014b) (Forissier *et al.*, 2014a). Au Québec, elle est connue sous le nom de "Wawaron" Bullfrog (*Lithobates catesbeianus*). C'est la plus grosse grenouille en Amérique du Nord, de taille 9 à 15 cm et de couleur Vert olive. Elle a les pattes palmées. Cet animal prend de nombreuses adaptations durant la saison hivernale. Ses têtards sont grands et vivent dans les eaux glacées et son chant ressemble à un cri de taureau (Bourdeau *et al.*, 2015), (Forissier *et al.*, 2014a), (Forissier *et al.*, 2014b).

Une collaboration entre des apprenants issus d'environnements différents passe par une confrontation de leurs conceptions (Forissier *et al.*, 2014b). Cette confrontation permet de produire un effet de contexte (Flanagan, 1954), (Merlo-Leurette et Forissier, 2009) qui représente un défi pour les apprenants, en particulier pour leurs modèles mentaux existant (Forissier *et al.*, 2014b). Cet évènement résultant du contexte doit être géré pour garantir l'efficacité de l'apprentissage en contexte (Merlo-Leurette et Forissier, 2009). C'est dans cette perspective que se situe notre problématique.

## 1.2 Hypothèse de solution

Le contexte fait partie intégrante des études scientifiques et devrait faire partie intégrante de l'apprentissage des sciences (Bourdeau *et al.*, 2015). Face à un enseignement très souvent détaché du contexte, l'idée de notre projet est de trouver une méthode ou méthodologie pour intégrer la notion de contexte géographique dans l'apprentissage des sciences. Il s'agit de permettre aux élèves, par échanges distants (parfois inter-contexte), de découvrir le rôle du contexte dans l'apprentissage de domaines d'étude variés, soit en sciences naturelles, en sciences humaines, en lettre et communication ou en économie (Forissier *et al.*, 2014b). Le modèle de contexte et son déploiement au sein d'un système auteur sensible au contexte se situe dans cette vision.

Dans le cadre de notre recherche, la notion d'effet de contexte est modélisée comme un écart dû à la confrontation de contextes internes entre deux apprenants lors d'une situation d'apprentissage (Forissier *et al.*, 2014b). Cette confrontation produit généralement un Clash (incident) qui fait ressortir la variabilité des deux contextes en lien avec l'objet d'étude en jeu. Lors des activités pédagogiques de confrontation (exemple : une classe virtuelle avec d'un côté des élèves du Québec

et de l'autre les élèves de la Gouadeloupe), le clash peut se manifester par des surprises observables sur les expressions faciales des participants, ou même des réactions fortes comme des cris, etc.

Il est possible de proposer une approche de modélisation de contextes qui permettent et facilitent l'estimation de l'écart entre deux ou plusieurs contextes, permettant ainsi de considérer l'écart ainsi estimation pour le développement d'un système hôte sensible au contexte. Nous souhaitons d'abord étudier le processus d'émergence de ces incidents (clash), et analyser les corrélations entre eux. Le modèle d'estimation de l'écart de contexte nécessite de concevoir un modèle computationnel du contexte et des instruments permettant à des élèves (dans un cadre d'apprentissage inter-contexte), de découvrir le rôle du contexte dans l'apprentissage d'un objet d'étude dans les différentes sciences (Forissier *et al.*, 2014b) (Mazabraud *et al.*, 2013). Nous prouverons que le modèle ainsi élaboré peut être intégré dans plusieurs variétés de systèmes hôtes afin de les rendre sensibles au contexte : éditeur de scénario, système tutoriel intelligent, LMS (ex. Moodle), etc.

### 1.3 Objectif

Ce travail de maîtrise vise à développer un système capable de modéliser un contexte et de calculer l'écart entre deux contextes afin d'aider au développement des connaissances des apprenants sur les effets de contexte. Pour ce faire, plusieurs sous-objectifs doivent être atteints :

a- Analyser l'approche développée par l'équipe du projet TEEC pour la modélisation du contexte physique d'observation proposée aux élèves en biologie, en géologie et en géothermie. Cette analyse aboutira à un modèle générique du contexte (Méta-Modèle du Contexte) fondé essentiellement sur des familles de paramètres pertinents par rapport au domaine d'étude (à l'objet d'étude).

b- Développer une méthode de collecte de données pour l'acquisition des valeurs des paramètres spécifiques à chaque contexte (Modèle d'un contexte)

c- Concevoir la base de données nécessaire pour l'implémentation du système de façon générique, c-à-d utilisable pour divers contextes d'apprentissage des sciences selon la nature (domaine de définition) des paramètres.

d- Concevoir et implémenter un outil visuel (application Web) permettant de définir facilement de nouveaux contextes, de faire des requêtes de calcul d'écart de contextes et de visualiser les résultats. Un tel outil (MazCalcWeb) sera destiné aux humain (concepteur de scénario pédagogique, enseignants, etc.) pour les aider à capturer et à générer les potentiels effets de contexte qui pourraient survenir lors des activités de confrontation. e- Implémenter un ensemble de services sous-forme d'une API (MazCalcAPI) pour permettre la création de méta-modèle de contexte, l'instanciation des modèles de contexte, et les requêtes de calcul d'écart formuler par des machines (applications). Une telle solution permettra ainsi d'intégrer MazCalc dans des environnements d'apprentissage hôte comme par exemple les systèmes tutoriels intelligents sensibles au contexte.

#### 1.4 Méthodologie

Le cadre méthodologique général du projet dans lequel se situe le sujet de maîtrise est de type DBR (designed-Based Research). Cette méthodologie se base sur la notion d'itération pour analyser un problème donné permettant d'expérimenter le modèle et la stratégie pédagogique avec des enseignants et des élèves, de valider les hypothèses grâce à une analyse des données éducatives (Forissier *et al.*, 2014b). Chaque itération identifie un objectif et des techniques de recherche, et la sortie de chaque itération est l'entrée de l'itération suivante (Bourdeau *et al.*, 2015). Les itérations des situations d'apprentissage planifiées pour notre projet

portent sur les thèmes de la géothermie, de l'environnement et des langues. Durant chaque itération, les apprenants sont guidés à comprendre les effets de contexte et à identifier les paramètres qui influencent ce phénomène ainsi que les paramètres indépendants du contexte qui sont des facteurs de généralisation (Forissier *et al.*, 2014b).

Dans ce cadre méthodologique général, nous intégrons nos résultats par raffinement successif à travers les itérations. Par exemple, le modèle de contexte issue de nos travaux et le prototype d'outil de modélisation que nous avons développé a été utilisé pour créer deux modèles de contexte en géothermies (un pour le Québec et l'autre pour la Guadeloupe), ce qui a été intégré dans une des itérations sur l'apprentissage de la géothermie. Nous maintiendrons la même démarche de validation pour la suite de nos travaux.

## 1.5 Démarche et organisation de la mémoire

Tout d'abord, suite à cette introduction, nous présentons au chapitre 2 une étude théorique du contexte où nous définissons toutes les notions nécessaires et liées au contexte. Ensuite, au chapitre 3, nous décrivons un état des lieux des travaux de recherche les plus intéressants pour notre travail. Cela nous aidera à comprendre la théorie, l'état des connaissances ainsi que les travaux actuels en la matière. Ensuite, en établissant le cadre théorique du projet au chapitre 4, nous situons notre travail dans le projet CAITS et nous établissons nos responsabilités et le plan pour intégrer notre solution dans un système hôte. Dans le chapitre 5, nous procédons à l'élaboration conceptuelle de la solution en expliquant notre démarche. Nous présentons ainsi notre méthodologie et nous élaborons les besoins du système que nous devons livrer. Nous présentons au chapitre 6 les différents détails de nos choix techniques ainsi qu'un survol des différentes techniques et

meilleures pratiques utilisées pour aboutir à un logiciel, performant flexible, facile à maintenir et évolutif . L'implémentation et l'évaluation de la solution sont décrites, fondées sur ces meilleurs pratiques. Nous terminons ce mémoire au chapitre 7 par une conclusion qui résume nos contributions et réalisations, ainsi que les limites et perspectives de notre travail.

## CHAPITRE II

### ÉTUDE THÉORIQUE DE CONTEXTE

L'objectif de ce travail est de développer un système générique qui peut être étendu au besoin. On vise particulièrement toutes formes de modèle de contexte basé sur localisation géographique. D'abord, le système doit être capable de modéliser un contexte et de définir ses paramètres et les marges de leurs valeurs. Il doit permettre aussi de comparer deux contextes distants et calculer l'écart entre eux tout en identifiant les paramètres de contextes dynamiques et statiques (caractériser un objet d'observation dans différents contextes). Ce système doit aussi offrir à chaque type d'acteur, tel qu'un expert du domaine ou un expert de contexte, d'utiliser les outils nécessaires pour effectuer son travail selon les règles et les tâches qui lui sont attribuées. Cet objectif ne peut être atteint sans avoir collecté les informations nécessaires du domaine de la contextualisation.

Le but de ce chapitre est donc d'exposer le cadre théorique de notre projet : Le contexte et la contextualisation. Nous devons comprendre c'est quoi un contexte ? Quels sont ses types ? Quelles sont les éléments clés à prendre en considération lorsque nous l'étudions ?

Ce chapitre contient les connaissances nécessaires pour la mise en place d'un système capable de bien modéliser un contexte, et qui sera par la suite intégré dans un système tutoriel intelligent efficace sensible au contexte afin de guider des ap-

prenants au cours de leur apprentissage.

## 2.1 Contexte

En examinant les définitions de contexte dans la littérature, nous nous rendons compte qu'il y a un écart qui est parfois assez important entre les différents travaux de recherche sur ce sujet, et comment les auteurs définissent le contexte. On peut renvoyer ces différences à trois facteurs majeurs :

- L'évolution chronologique de l'utilisation et de définition de contexte
- L'utilisation du contexte dans différents domaines d'applications
- Les différents points de vue.

### 2.1.1 Évolution chronologique de définition de contexte

L'étude de l'apprentissage en général, - et des études basées sur le contexte en particulier - a suscité de plus en plus d'intérêt des chercheurs surtout avec l'intégration de nouvelles technologies dans le domaine d'apprentissage. En effet, l'optimisation des processus d'apprentissage des étudiants, est devenu un but commun entre plusieurs disciplines incluant les sciences elles même - sujet d'apprentissage - (biologie, langues, science humaines ...), la psychologie-la sociologie (pour améliorer la qualité de vie des étudiants et leurs résultats scolaires) et la technologie de l'information où on s'intéresse à développer des systèmes intelligents que ce soit pour améliorer l'apprentissage des humains (systèmes tutoriels, LMS ...) ou pour permettre aux machines elles-mêmes d'apprendre (intelligence artificielle).

Les définitions du contexte sont ainsi assez diverses et font usage de différents termes pour qualifier le « contexte » comme par exemple « informations de contexte », « informations contextuelles » et « éléments de contexte » (Pradeep

et Krishnamoorthy, 2019). Bastien a qualifié le contexte comme étant l'ensemble d'éléments qui qualifie les situations ou les états dans lesquels l'objet d'étude est inclus. On parle alors d'ensemble éléments situationnels (Bastien, 1998). Un an plus tard, Dey et Abowd ont établie l'une des plus connues des définitions du contexte qui a été qualifié comme étant «toute information caractérisant la situation d'une entité. Cette entité peut être une personne, un objet, un lieu ou n'importe quel aspect pertinent de l'interaction entre un utilisateur et une application y compris l'utilisateur et l'application eux-mêmes» (Abowd *et al.*, 1999). En 2000, Chen et al. ont mis l'accent sur l'application, son environnement et ses paramètres car ceci reste le centre où se produit les événements auxquels s'intéresse l'utilisateur (Chen *et al.*, 2000). Bazire et Brézillon ont défini le contexte comme tout ce qui entoure l'objet d'étude. Il est constitué de toutes les circonstances qui encadrent un élément (objet ou un événement) et ne peut pas en être dissocié. L'élément en question perd son sens dès qu'on le sépare de son contexte. Les auteurs ont aussi mis l'accent sur la perception du contexte comme un espace de connaissances où l'interaction entre les éléments est plus descriptive que les éléments eux-mêmes. On a donc un espace partagé de connaissances qui inclut les différentes interactions entre les utilisateurs (leur objectif d'apprentissage par exemple), les objets du domaine (comme les états et contraintes), l'environnement (l'application par exemple) et les événements du système (historique des actions par exemple) (Bazire et Brézillon, 2005).

En 2006, Petersen et Cassens ont défini le contexte comme l'élément clé qui aide les entités intelligentes à comprendre comment les événements et les interactions dans le monde environnant affectent leurs comportements. On parle alors de conscience du contexte ou de l'environnement (Cassens et Kofod-Petersen, 2006). Schmidt a défini le contexte comme un «*sous-ensemble pertinent de l'état du monde à un moment donné (y compris les connaissances respectives de l'histoire*

*et les attentes pour l'avenir à ce moment*)» (Schmidt, 2006). Almazan a rejoint Bazire et Brézillon et a considéré le rôle principal de l'interaction et donc la relation entre les entités du système. Le contexte a été défini comme une ou plusieurs relations entre les objets d'information physique (une personne, une application, un objet, etc.), virtuel (service, message, etc.) ou concept (temps, lieu, etc.). Ces relations peuvent changer à n'importe quel moment pour n'importe quelle raison (Almazan, 2010).

En 2011, Tuzhilin a défini l'information contextuelle comme étant l'information qui a pour objet le contexte d'une entité. Cette information est décrite par les valeurs des facteurs contextuels qui peuvent influencer l'entité au quelle appartient le contexte comme par exemple l'environnement, l'heure, le lieu, les périphériques ... (Palmisano *et al.*, 2008). Adomavicius, en 2014, a mis l'accent sur la complexité de définition du contexte et l'a qualifié de "notion complexe" (Adomavicius et Jannach, 2014) tandis que Renata et al. (2014) a défini le contexte comme un acteur implicite ou explicite qui influence le contenu d'une activité d'apprentissage (Burbaite *et al.*, 2014). En 2019, Preeja et Shivsubramani ont généralisé le contexte en «*Toute information, lorsqu'elle est pertinente, devient le contexte de cette situation*» (Pradeep et Krishnamoorthy, 2019).

### 2.1.2 Définition selon le domaine d'application

Le contexte est une notion complexe à comprendre (Adomavicius et Jannach, 2014) et Bazire et Brézillon l'ont qualifié comme étant une notion très vaste et difficile à cerner du fait que le domaine d'utilisation influence la perception du contexte (Bazire et Brézillon, 2005). Donc, il est évident que le terme contexte a une multitude de définitions comme on l'a vu précédemment. Les définitions trouvées dans la littérature sont diverses et variées et spécifiques aux domaines

de recherche et d'application (Palmisano *et al.*, 2008). Bastien a indiqué que le contexte a pour rôle d'enlever l'ambiguïté et de faciliter l'interprétation de la situation dans le domaine Psychologique (Bastien, 1998), tandis que la définition donnée par Tuzhilin est propre au "systèmes de recommandation" (Palmisano *et al.*, 2008). Dans le domaine de l'informatique contextuelle, Chen s'est intéressé aux états des paramètres de l'environnement d'une application, où tout événement intéressant l'utilisateur prend place. Il a établi les caractéristiques du contexte dans cette perspective (Chen *et al.*, 2000). La définition la plus répandue du contexte est celle introduisant un ensemble de situations qui encadrent un objet ou un événement (Bazire et Brézillon, 2005). Dans le domaine de l'apprentissage à base des technologies (E-learning), la définition utilisée est celle de Schmidt (Schmidt, 2006) et Dey (Abowd *et al.*, 1999).

En plus de l'impact du domaine d'étude, le point de vue duquel un contexte est considéré semble aussi influencer la définition de ce dernier. Étymologiquement, le terme contexte a une origine latine "contextus" issu de la composition des termes "con" (ensemble) et "textere" (tisser), pour signifier "tisser ensemble". Le terme a été utilisé depuis le quinzième siècle pour dénoter "une composition, une chronique, tout le texte d'une écriture" (Bastien, 1998). Du point de vue ontologique (Bourdeau *et al.*, 2015), le contexte est «*un ensemble d'éléments liés structurellement ou fonctionnellement autour d'un élément central, tel que le contexte d'un mot, d'une action ou d'un événement, etc*». Dourich a considéré dans ses études trois points de vue (Dourish, 2004), (Verbert *et al.*, 2012) : En premier lieu, dans les sciences de phénoménologie, le contexte est un problème d'interactions entre deux activités ou objet ayant des caractéristiques définies dynamiquement. Seuls les phénomènes perçus d'un individu ou un objet sont considérés. En second lieu, du point de vue social, il affirme que «*le contexte ne peut pas être une description d'une situation mais plutôt une caractéristique de l'interaction de l'apprenant*»

avec son environnement. Enfin, quant à l'aspect technique, "le contexte doit avoir une définition pratique et plus spécifique en terme opérationnel». Pour opérationnaliser le contexte, les chercheurs ont tenté de définir le contexte en énumérant des catégories (Verbert *et al.*, 2012) que nous détaillerons dans le chapitre suivant. En science de communication, le contexte est défini comme «*la combinaison de l'historique de tout ce qui s'est passé au cours d'une période donnée, l'état général des connaissances des agents participants à un moment donné , le petit ensemble de choses qu'ils attendent à ce moment précis. Cependant, chaque entité impliquée dans une interaction à son propre contexte peut ou non être cohérent avec certaines parties du contexte des autres*» (Brézillon, 2003). En plus des points de vue mentionnés en sus, le contexte possède une dimension dynamique qui rend la modélisation problématique. Cette dimension dynamique est corollaire des interactions entre les agents. Autrement dit, sans agents en interaction, il n'y aurait pas de contexte (Pradeep et Krishnamoorthy, 2019).

## 2.2 Les systèmes sensibles au contexte

Plusieurs définitions relatives aux systèmes sensibles au contexte ont été établies. Le terme "sensibles aux contextes" ( "context-aware" en anglais) a été évoqué la première fois en 1994 par Schilit et Theimer (Abowd *et al.*, 1999) qui ont considéré que «*l'un des défis de l'informatique mobile est d'exploiter l'environnement en mutation avec une nouvelle classe d'applications tenant compte du contexte dans lequel elles sont exécutées* » . Ils ont énuméré les composants , tels que la localisation, l'identification des personnes proches et les périphériques accessibles, qui doivent être pris en compte dans ces applications (Schilit et Theimer, 1994) . Selon l'équipe de Hull, la sensibilité au contexte concerne l'informatique située dans un environnement « *ayant la capacité des dispositifs informatiques à détecter, interpréter et répondre aux aspects de l'environnement local et ceux de*

*l'utilisateur. Cette capacité promet à la fois d'ajouter de la valeur à l'utilisation actuelle des ordinateurs et de créer de nouveaux types d'applications.»* (Hull et al., 1997). Ryan et son équipe ont défini un système sensible au contexte comme une application ayant le pouvoir de collecter les informations de l'environnement au moyen des capteurs et permet les utilisateurs de choisir les contextes à étudier en fonction de leurs intérêts (Ryan, 1998) . Byun et Cheverst ont qualifié un système de "contextuel" s'il est capable d'extraire ,d'interpréter et d'exploiter les données provenant du contexte et adapter ses solutions en fonction du contexte d'usage actuel (Byun et Cheverst, 2004). Quant à Khedr et Karmouch, ils ont fixé trois éléments de base qui doivent exister pour qu'un système puisse être appelé "système sensible au contexte". Selon eux, un système sensible au contexte doit déduire de information du contexte en question , fournir ces informations contextuelles personnelles à l'utilisateur dès qu'il choisi son niveau intellectuel et de lui offrir des services de gestion de contexte, tel que la sauvegarde, la consultation et la modification de informations contextuelles si nécessaires (Khedr et Karmouch, 2004). Alors que pour Wrona et Gomez un système sensible au contexte doit avoir la capacité d'identifier les capteurs qui peuvent lui fournir des informations contextuelles, de tirer ces informations et les stocker dans un espace de sauvegarde et la phase la plus importante c'est qu'il doit raisonner sur ces informations pour développer d'autre informations plus pertinentes et plus utiles à partir de celles existantes (Wrona et Gomez, 2006).

Nous adoptons, pour notre étude, la définition proposée par Abowd et Dey comme une base de départ pour connaître les points clés de notre système sensible au contexte que nous souhaitons construire et qui disait : « *Un système est sensible au contexte s'il utilise le contexte pour fournir des informations et/ou des services pertinents à l'utilisateur, dont la pertinence dépend de la tâche de l'utilisateur.»* (Abowd et al., 1999) et de l'enrichir par des services de gestion

de contexte -comme indiqué dans la définition de Khedr et kamrouch- , et par un système de raisonnement sur les informations contextuelles comme Wrona l’a mentionné. Pour garantir un système robuste, générique et optimal sensible au contexte , il est important de cerner les paramètres clés du contexte et des informations contextuelles. Nous spécifions ces paramètres et ces caractéristiques dans les sections qui suivent.

### 2.3 Contexte interne et Contexte externe

Une des classifications les plus rencontrées dans la littérature est celle qui considère le niveau interne et externe du contexte. Selon Brézillon, la cognition est le critère déterminant le niveau du contexte, c-a-d interne ou externe, et qui repose essentiellement sur la nature de la relation entre le contexte lui-même et le contexte de la cognition de l’individu (Bazire et Brézillon, 2005).

D’une part , Bastien a défini le contexte externe comme “tout ce qui entoure et interagit avec une personne dans une situation donnée et a un temps donné” (Bastien, 1998), c’est-à-dire les éléments externes à l’individu comme sa localisation, et l’environnement spatial et écologique (Forissier *et al.*, 2014b) . D’autre part, le contexte est dit interne lorsqu’il appartient à l’apprenant et occupe une place centrale et indispensable dans la représentation de l’expérience perçue par l’individu lors de l’activité d’apprentissage (Bazire et Brézillon, 2005). Le contexte interne de l’apprenant est l’ensemble de ses modèles conceptuels acquis à travers ses connaissances acquises auparavant, de ses expériences passées, et de ses capacités cognitives. En d’autres termes, c’est sa représentation mentale sur un aspect étudié à un instant donné (van Wissen *et al.*, 2013). Les deux notions restent tout de même fortement liées car le contexte externe influence l’acquisition de connaissance et participe donc à l’évolution du contexte interne de l’individu (Anjou

*et al.*, 2017a).

## 2.4 Les effets de contexte

L'effet de contexte dans une situation pédagogique est défini par Leurette comme un événement produit par la tension ou la collision de deux contextes internes issus d'environnements éloignés, par rapport à un concept étudié (Merlo-Leurette et Forissier, 2009). En fait, dans une activité d'apprentissage, différents acteurs entrent en interaction, chacun avec ses propres expériences et ses propres conceptions cognitives. L'écart de contexte peut être entre l'enseignant et l'apprenant, deux apprenants différents ou même un apprenant et le manuel pédagogique. Cet écart peut prendre plusieurs formes telles que l'interrogation, le malentendu ou incompréhension, la contradiction, la négligence et peut aller même au rejet du contexte de l'autre (Forissier *et al.*, 2014b). Selon Anjou et son équipe, le terme « *effet de contextes* » est issu d'une analogie avec « *l'effet de contrat* » de Brousseau (1980) (Brousseau, 1980). Les effets de contextes « *se manifestent par un décalage entre un objectif d'enseignement ou d'apprentissage et leur réalisation* ». Ils sont qualifiés de cette manière lorsque le décalage est attribué « *aux différents contextes en présence dans le processus didactique* » (Anjou, 2018b). Ces effets sont l'origine de prise de conscience mutuelles du contexte de l'autre (apprenant). Ils permettent ainsi de générer des évolutions de conceptions situées (internes) vers d'autres évolutions plus élargies et plus riches. (Anjou *et al.*, 2017a).

Face au défi de ces effets contextuels, qui ont généralement des représentations conceptuelles différentes, les apprenants réagissent de façons différentes soit en changeant leurs propres conceptions ("changement conceptuel") ou tout simplement en acceptant plusieurs modèles pour la même connaissance ("représentations multiples") (Forissier *et al.*, 2014b). La mise en place d'une pédagogie basée sur

les effets de contexte peut permettre de mieux marquer l'esprit de l'apprenant en bousculant ces conceptions, ses connaissances acquises. Les effets de contextes provoquent alors un genre de choc cognitif permettant de heurter leur sérénité (Anjou, 2018b). En effet, la création de situations particulières dans lesquelles des apprenants sont amenés à confronter leurs représentations d'un objet au réel ou dans lesquelles «*des apprenants aux contextes internes différents, confrontent leurs représentations mutuelles en interagissant, permet de générer des effets de contextes prenant la forme de conflits cognitifs ou sociocognitifs*».

## 2.5 Les caractéristiques d'un contexte

Les informations de contexte sont générées à partir "des dispositifs de mesure". La manière dont ces informations sont acquises et leurs sources de provenance déterminent la qualité du contexte en question. Cela fait partie des caractéristiques du contexte comme l'a mentionné Akyildiz (Akyildiz *et al.*, 2002). Van Bunningen l'a introduit parmi les caractéristiques qu'il a étudiées à partir de la littérature (van Bunningen *et al.*, 2005). Il a ajouté que Les dispositifs de mesure peuvent être de taille très petite grâce au développement très rapide de la technologie des outils compacts. Les téléphones portables font partie de ces caractéristiques. Ils sont souvent utilisés aussi comme outils de collecte de données : il est possible de prendre des images, de faire des enregistrements audio, d'utiliser des applications installées pour mesurer la température, la fréquence, les battements de cœur...etc. Généralement, Les sources utilisées sont distribuées. Parfois il est nécessaire de tracer un lien de dépendance entre un certain nombre d'informations hétérogènes collectées à partir de différents types de source pour en déduire une nouvelle information.

Tous ces caractéristiques étudiées par Van Bunningen sont dues aux éléments

externes du contexte. Cependant, il existe d'autres caractéristiques qui résultent du contexte lui-même. Le contexte a trois caractères : dynamique, temporel et spatial. Le premier caractère est dû à la nature dynamique des situations en études en elles-mêmes. Les situations changent, par conséquent les informations contextuelles changent et bien évidemment le contexte change. Le second caractère est relatif aux informations du contexte qui peuvent varier ou rester constantes au fil du temps. Par exemple, la taille d'un animal ou d'une personne évolue dans le temps, la taille à l'âge de 3 mois est sûrement différente de celle de 20 ans. Le nombre des dents aussi change pour les premières années d'une personne, mais durant un certain temps, il peut être constant. Le dernier caractère tourne autour les informations contextuelles dépendantes de la localisation du contexte. Un même type d'information peut ne pas avoir la même valeur d'une place à une autre. Comme la température du sol change d'un pays à un autre, et même d'un coin à un autre. Le dernier caractère mentionné par Bunningen est l'imperfection et l'incertitude du contexte, qui résultent de son caractère dynamique et de la qualité des dispositifs utilisés : L'écart du temps entre la collecte de l'information et son utilisation donne un impact d'incertitude si la situation a changée au cours de ce temps, ou l'utilisation d'un dispositif avec une batterie mal chargée ou surchauffée peut ne pas donner une information exacte.

En 2006, Kim a mentionné quatre caractéristiques du contexte. Il a conservé le caractère "dynamique" et celui "distribué" du contexte, et il a ajouté un caractère "confidentiel" et un autre "transformable". Le premier est à cause de l'utilisation des informations personnelles et parfois privées. Le second est donné par le fait que l'agrégation des informations contextuelles produit de nouvelles informations de plus haut niveau, et si les informations de bas niveau sont erronées, celles de haut niveau le sont aussi (Kim et Lee, 2006).

## 2.6 La qualité d'un contexte

Contrairement aux humains, les infrastructures informatiques ne sont pas capables d'interpréter les informations de contexte et en tirer avantage lors de l'interaction. Ces informations doivent donc être fournies de manière explicite (Buchholz *et al.*, 2003).

Entre deux informations de même type et pointant sur la même propriété, il peut y avoir une nuance au niveau de qualité, ainsi il y aura certainement une meilleure que l'autre. Chaque information diffère d'une autre en matière de précision, de probabilité d'exactitude ou encore de confiance (Buchholz *et al.*, 2003). En raison de la nature dynamique du système étudié, des ressources multiples et de défauts des capteurs, les informations contextuelles sont en perpétuel changement, et peuvent être incorrectes, incohérentes ou incomplètes (Pradeep et Krishnamoorthy, 2019). Henricksen a ajouté que les informations contextuelles peuvent être aussi non disponibles, ambiguës à cause de différentes valeurs disponibles, imprécises ou erronées (Henricksen et Indulska, 2004). Dans ce cas, les informations de contexte peuvent ne pas être fiables ou utiles, donc il est impératif que la crédibilité d'information contextuelle soit assurée. (Kim et Lee, 2006). Ce qui donne à la qualité du contexte un rôle de premier degré dans les critères exigés pour les informations contextuelles (Pradeep et Krishnamoorthy, 2019).

La qualité d'information (Wang *et al.*, 1993), (Schilit *et al.*, 1994), (Kim et Lee, 2006), (Buchholz *et al.*, 2003) a toujours été considérée comme un domaine de recherche important. Plusieurs recherches ont été élaborées. Krause et Hochstatter ont défini la qualité d'information comme « *Toutes les informations inhérentes qui décrivent les informations de contexte et peuvent être utilisées pour déterminer la valeur des informations pour une application spécifique* » (Krause et Hochstatter, 2005) , (Pradeep et Krishnamoorthy, 2019). L'équipe de Man-

zoor a mentionné que « *La qualité du contexte indique le degré de conformité du contexte collecté par les capteurs à la situation actuelle dans l'environnement et aux exigences d'un consommateur de contexte particulier* » (Manzoor et al., 2014), (Pradeep et Krishnamoorthy, 2019). Alors que Buchholz a noté que la « *Qualité de contexte (QoC) désigne toute information décrivant la qualité des informations utilisées en tant qu'informations de contexte. Ainsi, la QoC fait référence à des informations et non au processus ni au composant matériel qui fournit éventuellement ces informations* » (Buchholz et al., 2003).

D'après la littérature nous constatons que la qualité de contexte regroupe un certain nombre de paramètres. Les chercheurs ont proposé différentes métriques afin de mesurer la qualité des informations. En 2003, Buchholz (Buchholz et al., 2003) et henricksen (Henricksen, 2003) ont étudié cinq métriques pour déterminer l'imperfection générée lors de la collecte de l'information : la précision, l'exactitude, la confiance, la résolution et la mise à jour de l'information.

La première métrique indique à quel degré l'information de contexte collectée est de l'information réelle. La seconde donne la probabilité qu'une information collectée soit correcte. La troisième est la probabilité que la source de l'information donne une information correcte. La quatrième reflète la granularité de l'information. Et la dernière montre depuis combien de temps cette information existe. En 2012, Krishnamoorthy a aussi considéré l'exactitude, la résolution et la confiance comme des métriques importantes pour qualifier une information. Il a ajouté la récurrence pour mesurer la répétabilité, la couverture pour quantifier la plage occupée et l'erreur moyenne d'une information (Krishnamoorthy, 2012). L'équipe de Manzoor a proposé d'autres mesures de qualité d'information totalement différentes de ce que nous avons mentionné en sus : la crédibilité, la validité, l'exhaustivité, la valeur critique, la convivialité, le droit d'accès à l'information, et la cohérence de l'information (Manzoor et al., 2014), (Pradeep et

Krishnamoorthy, 2019).

Dans notre recherche, les informations contextuelles seront introduites explicitement dans notre système de modélisation de contexte (appelé MazCalc). Pour ce faire, nous avons défini deux niveaux d'informations de contexte.

Un premier niveau "macro" où l'identification de tous les paramètres pertinents ainsi que la plage des valeurs que peut avoir chaque paramètre, indépendamment du contexte, est crucial pour construire le squelette de l'objet à étudier, c'est-à-dire le modèle de l'objet d'étude. De nombreux experts du domaine d'étude participeront donc à cette étape pour assurer une modélisation optimale, et par la suite une qualité d'information optimale.

Le second niveau "micro" auquel les experts de l'objet d'étude, dans un contexte donnée, spécifieront à partir de paramètres globaux (collectés précédemment dans un méta-modèle) ceux qui sont réellement présents dans le contexte en étude. A ce stade, une valeur unique sera attribuée à chaque paramètre. Cette valeur est aussi introduite par les experts ainsi que les détails relatifs à la source (cette source peut être un document de littérature, un capteur, un thermomètre, etc.), et le degré de confiance de cette valeur, afin de garantir une meilleure précision d'information. En raison de la nature dynamique des systèmes et des environnements (Pradeep et Krishnamoorthy, 2019), les dates de création des modèles et des objets d'étude dans leur contexte seront conservés aussi, pour garantir la traçabilité de l'information et donc sa mise à jour pour des fins corrections.

## 2.7 Le modèle de contexte

La modélisation de contexte a plusieurs raisons comme est établi par (Forisier *et al.*, 2014b). Tout d'abord, dans l'apprentissage des sciences, la modélisation permet d'avoir un outil scientifique qui est la représentation d'un ensemble par-

tagé de conceptions du contexte et de ses composantes. La modélisation signifie donc une spécification ou une structure avec un ensemble d'entités, catégories et relations pouvant être calculées, manipulées ou enrichies. De plus, le modèle contextuel permet d'utiliser le contexte comme un outil pédagogique pour les enseignants et les apprenants pour concevoir des activités d'apprentissage et ramasser les observations, mesurer et traiter les données. La troisième raison est de pouvoir intégrer le modèle du contexte lui-même dans une plate-forme où il peut être connecté aux capteurs. La modélisation du contexte permet aussi de bien décrire un phénomène. Donc avant d'entamer la phase de réalisation de notre modèle de contexte, il est important de comprendre c'est quoi tout d'abord un modèle de contexte.

*«Un modèle de contexte identifie un sous-ensemble concret du contexte qui peut être atteint de manière réaliste à partir de capteurs, d'applications et d'utilisateurs et qui peut être exploité dans l'exécution de la tâche. Le modèle de contexte utilisé par une application contextuelle donnée est généralement spécifié explicitement par le développeur de l'application, mais peut évoluer avec le temps »* comme l'a défini Henricksen (Henricksen, 2003) . En 2017, Gasparic a généralisé la définition à *«un ensemble de facteurs contextuels qui caractérise une situation»* (Gasparic *et al.*, 2017).

Pour bien modéliser un contexte, certaines exigences doivent être appliquées pour aboutir à un modèle de contexte efficace et robuste. Krishnamoorthy a détaillé treize exigences dans ses études (Krishnamoorthy, 2012). Il a indiqué qu'il est nécessaire d'utiliser une approche basée sur un modèle pour un développement facile des activités pédagogique. Un tel modèle doit permettre de grouper des informations contextuelles hétérogènes, distribuées et non repliables, de fusionner celles qui sont partitionnées et tracer les liens de dépendances entre les informations s'ils existent et le représenter de manière adéquate. Le modèle de contexte

doit être doté d'un service d'annotations de ces informations afin de garder la sémantique d'hétérogénéité des informations et l'exploiter de façon automatique dans des scénarios d'étude de contexte partagés. Un paramètre de contexte peut avoir une variété de valeurs qui dépendent du contexte en question, donc il est nécessaire que le modèle de contexte assure cette fonctionnalité tout en définissant la marge de ses valeurs permises et celles interdites. Il doit aussi utiliser un moyen de stockage des attributs garantissant la qualité de contexte que nous avons discuté dans la section 2.6 : entre autres garder une traçabilité sur la source qui a fourni l'information contextuelle. La sauvegarde des dates de collecte des données fait partie aussi de gestion de la qualité de l'information. Cet attribut prend en considération le caractère dynamique des systèmes d'études. Le modèle de contexte doit être extensible vu la variété des domaines d'application des informations contextuelles et leurs représentations. En garantissant ce qui est mentionnée, la facilité des formalismes de modélisation se dévoile et rend la manipulation des informations contextuelles plus facile et la construction des modèles conceptuels plus authentique au monde réel. A ce stade, le modèle de contexte englobe toutes les données utiles, donc comme une dernière exigence, ce modèle doit permettre, dans un système sensible au contexte, de raisonner sur ces données.

## 2.8 Le raisonnement contextuel

Une fois les informations contextuelles capturées et organisées dans un modèle, le raisonnement sur ces informations devient impératif pour en acquérir d'autres nouvelles plus riches et plus constructives. De nombreuses recherches ont couvert le sujet de raisonnement contextuel. Nurmi et Floréen (Nurmi et Floréen, 2004) l'ont défini comme *«l'activité de déduire des informations nouvelles et pertinentes à utiliser par les applications et les usagers à travers de différentes sources de données contextuelles existantes»*. Quant à Guan et son équipe, il s'agissait de

«formuler une déduction contextuelle de haut-niveau à partir d'un ensemble de contextes de bas-niveaux» (Guan *et al.*, 2007).

Diverses techniques de raisonnement sont mises en œuvre dans diverses recherches afin mettre en évidence l'utilité des informations contextuelles dans l'amélioration des connaissances. Se sont généralement des techniques d'apprentissage automatique (Pradeep et Krishnamoorthy, 2019). Parmi ces techniques on trouve la logique floue qui utilise une représentation naturelle et simple des informations (Kang *et al.*, 2014). Cette méthode offre un raisonnement approximatif à travers un continuum de possibilités qui soit valide une information, soit la rejette . Une autre technique utilisée est l'ontologie (Bikakis *et al.*, 2007). Elle supporte la représentations complexes des informations contextuelles et se caractérise par sa faible complexité informatique qui aide à gérer les situations à changements rapides. La logique probabiliste comme la Chaîne de Markov caché (HMM), est l'une des techniques de raisonnement que nous citons aussi. Cette technique gère l'incertitude par la combinaison des informations provenances de différentes sources . Elle a été utilisée en couche parallèles par Daehyun pour la reconnaissance de l'activité d'un utilisateur. (Sanchez *et al.*, 2007).

Après avoir établi les notions de bases de notre domaine de recherche, nous étudierons l'état de l'art sur l'opérationnalisation du contexte dans le chapitre suivant. En effet, nous ferons la revue des littératures qui se sont intéressées aux systèmes d'apprentissage basé sur le contexte et les différentes théories dans le domaine.



## CHAPITRE III

### ÉTAT DE L'ART SUR L'OPÉRATIONNALISATION DU CONTEXTE

Il a été admis, il y a longtemps, que le contexte joue un rôle critique dans la cognition humaine (Parker *et al.*, 2013), (Hollister *et al.*, 2017). Cette idée a été affirmée quand Sternberg a pris en compte l'intelligence contextuelles dans sa théorie (Sternberg *et al.*, 1995), (Hollister *et al.*, 2017). Depuis, plusieurs recherches se sont concentrées sur l'intégration des informations contextuelles dans le domaine de l'apprentissage et sur la conception des systèmes sensibles au contexte. D'un point de vue technique (Dourish, 2004), les chercheurs ont tenté de définir le contexte de manière plus spécifique dans une perspective opérationnelle (Winoograd, 2001). Plusieurs catégories de contexte sont mises en valeurs (Verbert *et al.*, 2012).

#### 3.1 Les catégories de contexte

Afin d'opérationnaliser le contexte, les chercheurs ont tenté de définir le contexte en énumérant des catégories.

Alors que la terminologie et la portée des éléments diffèrent considérablement, il existe également de nombreux chevauchements entre les définitions de contexte existantes. En 1994, Schilit et al. (Schilit *et al.*, 1994) a suggéré trois aspects principaux d'un contexte. le premier noté : "**où vous êtes**" : il prend

en considération le contexte physique comme l'éclairage, le niveau de bruit, les conditions de circulation et la température. Le second aspect noté : "**Avec qui vous êtes**" : cet aspect se concentre sur le contexte utilisateur : comme son profil, son emplacement, les personnes à proximité et sa situation sociale. Le dernier aspect est "**quelles ressources sont à proximité**". Il considère le contexte informatique, comme la connectivité, les coûts de communication, la largeur de bande de passante de communication, les ressources informatiques proches telles que les imprimantes, les écrans et les postes de travail..

Schmidt et al. (Schmidt *et al.*, 1999) a considéré sept catégories de contexte dans son étude qui sont l'utilisateur, l'environnement social de l'utilisateur, la tâche ou l'activité, la localisation, l'infrastructure, les conditions physiques et l'heure.

En 1999, Abowd et al. (Abowd *et al.*, 1999) ont classé le contexte en primaire et secondaire, ce qui est l'un des schémas de catégorisation de contexte populaires. Le contexte Primaire caractérise la situation d'une entité particulière, telle que l'activité, l'identité, le lieu et le moment, tandis que le contexte Secondaire dérive du contexte primaire, comme par exemple la liste d'amis d'une personne est le contexte secondaire du contexte primaire l'identité de la personne elle même.

En 2000, Abowd a donné une nouvelle catégorisation , où il énumère cinq notions minimales requises pour comprendre le contexte (Abowd et Mynatt, 2000) : Pour lui ,il faut identifier : premièrement, une personne en particulier : c'est le "**Qui**" de l'entité. Deuxièmement il faut identifier la tâche que fait cette personne : c'est le "**Quoi**". Troisièmement, il faut capter les informations de localisation de cette personne : c'est le "**Où**". Quatrièmement, il faut indiquer heure à laquelle cette personne se trouve à un endroit particulier : C'est le "**Quand**". Finalement il faut identifier pourquoi cette personne fait le "quoi" : c'est le "**pourquoi**".

Perera et ses collègues (Perera *et al.*, 2013) ont étudié le contexte dans le paradigme de l'Internet des objets et ils ont affirmé qu'un contexte donné ne peut pas être identifié à l'aide de la catégorisation fait par d'Abowd. Donc ils ont choisi de faire la classification selon le niveau conceptuel et le niveau opérationnel du contexte afin de fournir des schémas de catégorisation complets.

Cependant, Zimmermann a (Zimmermann *et al.*, 2007) a considéré dans sa catégorisation les quatre premières notions de Abowd qui sont : l'individu, l'activité, le lieu, l'heure, et il a rajouté "le relationnel" comme une cinquième catégorie de contexte fondamentale. L'individu chez Zimmermann ne se limite pas sur une entité humaine particulière uniquement, mais elle peut être aussi une entité naturelle, artificielle ou un groupe.

En se basant sur la classification de Zimmermann et al (Zimmermann *et al.*, 2007), Villegas and Müller (Villegas et Müller, 2010) ont proposé une taxonomie de regroupement pour chaque notion : L'individu est défini comme l'information relative à une entité indépendante naturelle, humaine, artificielle, ou groupe. La localisation est définie comme l'emplacement de l'entité et qui peut être soit physique si l'entité d'étude est physique, soit virtuelle si l'entité d'étude est virtuelle. L'heure est identifiée comme l'information associée au temps qui est soit définie en spécifiant l'heure de début et l'heure de fin, soit indéfinie, donc l'évènement s'est produit à cause d'une situation spécifique. L'activité est la tâche que fait l'entité. Le relationnel est défini soit par la relation sociale entre les individus, soit par la relation fonctionnelle entre les entités ou oppositionnelle défini par une agrégation ou association des entités.

Derntl et Hummel (Derntl et Hummel, 2005) ont catégorisé le contexte de façon plus détaillée : le contexte horaire (qui couvre le lieu, la date et l'heure), le contexte physique (qui peut être une personne), une revue ou un matériel péda-

gogique, le contexte numérique comme les documents électroniques, les services de formation en ligne, le contexte de dispositif comme le matériel informatique, les logiciels et la connectivité, et le contexte de l'apprenant qui couvre à la fois la dimension de l'individu et la dimension de l'activité.

Berri et al. (Berri *et al.*, 2006) ont fait la distinction entre deux grandes catégories de contexte. La première traite les aspects techniques des appareils mobiles et de leur environnement opérationnel, y compris la capacité de la bande passante du réseau et les périphériques d'entrée/sortie tel que l'écran, le clavier et la souris. La deuxième définit les éléments relatifs à l'apprenant, y compris ses buts, son niveau actuel de compréhension, le domaine de la matière d'étude et les conditions préalables.

Galleguillos et Belongie (Galleguillos et Belongie, 2010) ont présenté deux types de contexte : le contexte global et le contexte local. Ils ont appliqué cette catégorisation à la reconnaissance d'objet en utilisant les informations de contexte des images.

Les définitions citées de Schilit et al. (Schilit *et al.*, 1994), Schmidt et al. (Schmidt *et al.*, 1999), et Zimmermann et al. (Zimmermann *et al.*, 2007) sont des définitions génériques de contexte utilisées dans de nombreux domaines d'applications. D'autres catégorisations ont été proposées par des chercheurs du domaine de l'"apprentissage amélioré par la technologie" (Technology Enhanced Learning : TEL) (Verbert *et al.*, 2012). La plupart font référence à des catégories de contexte similaires à la catégorisation génériques, comme le lieu, l'heure, l'individu ou l'utilisateur, l'activité et les relations. Les conditions physiques, telles que l'éclairage et le niveau de bruit, sont définies moins souvent dans les systèmes d'apprentissage adaptatifs. En revanche, le contexte des ressources (y compris les ressources physiques et numériques pertinentes pour l'utilisateur) est utilisé plus souvent dans

ces applications.

Nous nous intéressons dans notre étude au contexte de localisation qui est devenu une composante très importante prise en compte par la plupart des systèmes technologiques afin d'offrir un apprentissage plus adaptatif. Les informations contextuelles de localisation sont établies soit explicitement, soit souvent implicitement via des capteurs de localisation comme le WIFI et/ou GPS (Zhou et Rechert, 2008). Parmi ces systèmes il y a ceux qui utilisent des étiquettes RFID ou des codes QR afin de localiser les objets de manière très précise comme TANGO (Ogata et Yano, 2004) , UoLmP (Gómez *et al.*, 2014). TenseITS (Cui et Bull, 2005), (Verbert *et al.*, 2012) repose sur une approche explicite où l'apprenant saisi manuellement le type d'emplacement, tel que domicile, université ou transport. Les systèmes d'apprentissage utilisent ces informations contextuelles, et plus précisément le contexte de localisation, de manière différente.

Dans leurs projet "Ambient Wood", Rogers et al. (Rogers *et al.*, 2005) se serve de l'emplacement pour fournir des activités d'apprentissage omniprésent , permettant aux apprenants d'observer et de collecter les données dans les bois.

MOBILearn (Lonsdale *et al.*, 2005) prend en compte la proximité entre apprenants pour soutenir l'apprentissage collaboratif. MOBILearn utilise un système de positionnement à ultrasons pour détecter la localisation. Il génère par la suite des contenus d'apprentissage pertinents et guide l'apprenant lors de ses activités. Il fournit aussi des suggestions de pairs apprenants ou enseignants se trouvant à proximité et qui travaillent sur des activités d'apprentissage similaires. MOBILearn fait la correspondance entre l'emplacement et le sujet d'intérêt actuel pour déterminer des ressources d'apprentissage.

UoLmP (Gómez *et al.*, 2014) est un système de type "adaptation générale"

qui est un sous-type de l'adaptation relative aux activités d'apprentissage. UoLmP vise l'adaptation des activités d'apprentissage de façon semi-automatique, en particulier, l'adaptation du flux de l'apprentissage d'un scénario pédagogique d'une part et des ressources, outils et services d'autre part. Il détecte les propriétés de la situation actuelle pour filtrer les contenus d'apprentissage et les capacités du périphérique utilisé par l'apprenant pour ajuster les contenu.

ALESS (Hsu *et al.*, 2016) est un système mobile sensible à l'environnement qui s'adapte aux mouvements des apprenants et leur fournit les informations nécessaires. Les informations contextuelles sont collectées en temps réels pour déterminer le meilleur parcours des activités d'apprentissage. ALESS implémente la technologie « RFID active de localisation » collé aux objets d'apprentissage. Ces étiquettes permettent d'offrir un guide instantané basé sur le statut de chaque apprenant.

CAMLearn (Soualah-Alila, 2015) utilisant les pratiques d'apprentissage déjà déployées dans les systèmes d'apprentissage électronique et les adoptant dans l'apprentissage mobile. Ce système est construit autour d'une ontologie qui définit le domaine d'apprentissage et prend en charge la prise de conscience du contexte. Le futur système adaptatif offrira un panel optimisé d'objets d'apprentissage correspondant au contexte actuel de l'apprenant.

L'ensemble des systèmes que nous avons présenté repose sur une adaptation selon le contexte de localisation. Le contexte de localisation est très important mais n'est pas suffisante pour traduire l'effet de contexte. Il s'agit d'une perspective purement opérationnelle de la notion de contexte qui vise l'efficacité computationnelle du système (en permettant par exemple de proposer des activités en tenant compte de la performance du réseau actuel). Toutefois, cette considération opérationnelle ne permet pas de résoudre la dimension sémantique dans laquelle

l'objet d'étude est fondamentalement ancré au contexte, faisant en sorte qu'un effet de contexte est possible lors d'une situation d'apprentissage collaboratif dans deux ou plusieurs contextes. C'est dans cette perspective que se situe notre problématique. Dans la section qui suit, nous présentons les premiers éléments de la solution que nous proposons pour développer un modèle de contexte qui permet non seulement de gérer l'écart de contexte , mais aussi aux systèmes hôtes d'en tenir compte pour améliorer leur efficacité pédagogique.



## CHAPITRE IV

### SOUTENIR LE MODÈLE CLASH PAR DES OUTILS INFORMATIQUES (MAZCALC ET CAITS)

Dans n'importe quelle situation donnée, il y aurait de nombreux facteurs qui influenceraient l'évolution de la situation. Pour reconnaître ces facteurs et représenter les informations de contexte produites par ces facteurs, un cadre est nécessaire pour gérer efficacement les informations de contexte. Ce cadre est appelé modélisation de contexte. Cette modélisation doit être générale car, comme la situation évolue de façon continue, nous ne pouvons pas prédire quelle information sera pertinente à l'avance. Le modèle de contexte est la base sous-jacente de notre système sensible au contexte (Forissier *et al.*, 2014b).

#### 4.1 Le modèle CLASH

Modéliser un contexte revient à définir sa spécificité et sa représentation. Chaque contexte est défini par un ensemble de paramètres associés à un phénomène ou objet d'étude. Ces paramètres ont tendance à changer de valeur d'un contexte à un autre pour un même objet d'étude, produisant un effet de contextes au niveau des connaissances des apprenants (Anjou *et al.*, 2017a). Dans le processus d'apprentissage, les apprenants sont guidés à comprendre cet effet de contextes et à identifier les paramètres qui influence ce phénomène, ainsi que les paramètres

indépendants du contexte qui sont des facteurs de généralisation. Du coup, leur ancienne représentation acquise sur l'objet d'étude en question n'est plus applicable dans le nouveau contexte. Les apprenants sont mis alors au défi de procéder à un changement conceptuel ou d'accommoder de multiples représentations. Ce processus permet aux apprenants de développer un modèle mental plus conscient et des connaissances plus précises (Forissier *et al.*, 2014b).

Nous souhaitons étudier le processus d'émergence de ces effets et de représenter le modèle de contexte interne des apprenants de façon indirect. L'idée est de représenter des objets scientifiques en termes de niveau de «contextualité» et de mesurer l'écart ces contextes externes à l'aide d'un calculateur de contexte afin prédire les thèmes et le temps d'apparition des effets de contexte. Cette façon de faire pourrait apporter des informations précieuses sur la compréhension et l'interprétation de l'objet d'étude(Forissier *et al.*, 2014b), que nous appelons "le modèle CLASH".

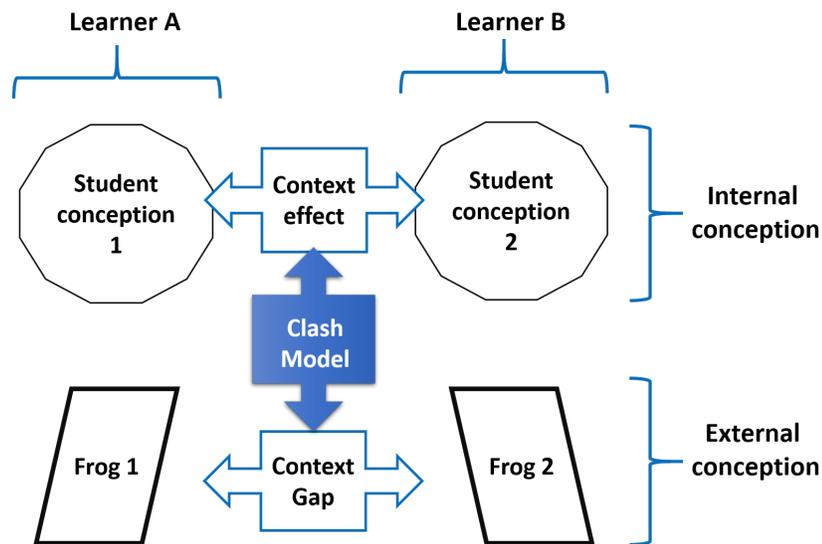


Figure 4.1: Le modèle CLASH : Interactions entre les contextes d'après (Forissier *et al.*, 2014b)

Afin de tester le modèle CLASH et plus spécifiquement nos hypothèses sur l'effet de contexte, nous avons adopté une méthodologie qui permet de produire, mettre en œuvre et tester de nouvelles idées de recherche, à travers une conception itérative.

## 4.2 Méthodologie

Notre hypothèse de recherche est comme suit. "L'apprentissage peut être plus efficace si on arrive à simuler les effets de contexte dans des scénarios d'apprentissage". Pour la valider, nous avons planifié d'étudier différents contextes par l'intermédiaire d'un certain nombre de scénarios pédagogiques afin de produire un scénario générique capable de reproduire des effets de contexte. La modélisation du contexte fait aussi partie des tâches à étudier. Cette modélisation servira comme un calculateur d'écart de contexte et de raisonneur sur ces résultats. Donc un outil informatique est nécessaire pour stocker les informations contextuelles et faire les calculs nécessaires. CAITS est un autre outil informatique qui intégrera l'outil de modélisation de contexte pour générer par la suite des scénarios pédagogiques simulant le contexte et ces effets dans le processus d'apprentissage. Ces deux outils informatiques nécessitent une étude approfondie pour savoir comment ils utiliseront les informations contextuelles, comment vont les représenter, comment sera leurs structures et comment leurs services doivent fonctionner ?

Nous avons remarqué qu'il existe différentes tâches à faire. Ces tâches comportent des chevauchements entre elles et doivent être étudiées en parallèle vu que les résultats trouvés dans l'une peuvent facilement changer les résultats d'une autre au niveau de la compréhension, de l'analyse ou de la conception. Dans le but d'atteindre ces objectifs, il est important d'utiliser une méthodologie de recherche qui nous permet d'élaborer différentes conceptions en même temps, de réunir

l'étude théorique et la mise en œuvre technique à la fois , de prendre en considération la complexité des situations d'apprentissage, de refléter l'état du monde réelle des tâches d'apprentissage, d'offrir la possibilité de produire des résultats à plusieurs fois du début jusqu'à la fin de notre projet et de tester nos hypothèses et les composantes créées au cours de notre recherche (Bourdeau, 2017).

Il s'est révélé que DBR (Designed-Based Research) est la méthodologie la plus appropriée à nos objectifs de recherche de notre projet TEEC. DBR a été annoncée pour la première fois dans la recherche de pionnier de Brown sous le nom d'expérience de conception (Bourdeau, 2017). Le défi était de réunir le processus théorique et le processus pratique de la recherche ce qui réduit l'écart entre la recherche en laboratoire et la recherche en situ. L'idée était d'introduire des processus en boucle dont les résultats de chaque boucle influence le comportement du système en étude (Brown, 1992) (Bourdeau, 2017). La méthodologie DBR est basée sur une approche d'itérations et de rétroaction (feedback). Elle permet d'analyser les résultats précoces ou intermédiaires, d'en tirer de nouvelles idées et de les tester dans les prochaines itérations : c'est de cette manière que permet d'évoluer les hypothèses de recherche (Anjou *et al.*, 2017a) (Anjou, 2018b). Cette approche est considérée comme une méthodologie des micro-systèmes. Son objectif est de décomposer un système complexe en sous-systèmes pour pouvoir l'étudier . Dans notre projet, une itération se résume la mise en place d'une nouvelle expérimentation d'enseignement en contexte à propos d'un objet d'étude donné. Au début de notre recherche, trois itérations de différents thèmes ont été planifiées au sein des institutions d'apprentissage et qui sont « la géothermie », « la grenouille » et « Dlo». (Bourdeau, 2017).

Chaque itération est structurée selon les principes de l'ingénierie pédagogique dont l'objectif est de simuler les effets de contexte chez les élèves. La planification des itérations est générique, dans le sens où elles peuvent être utilisées

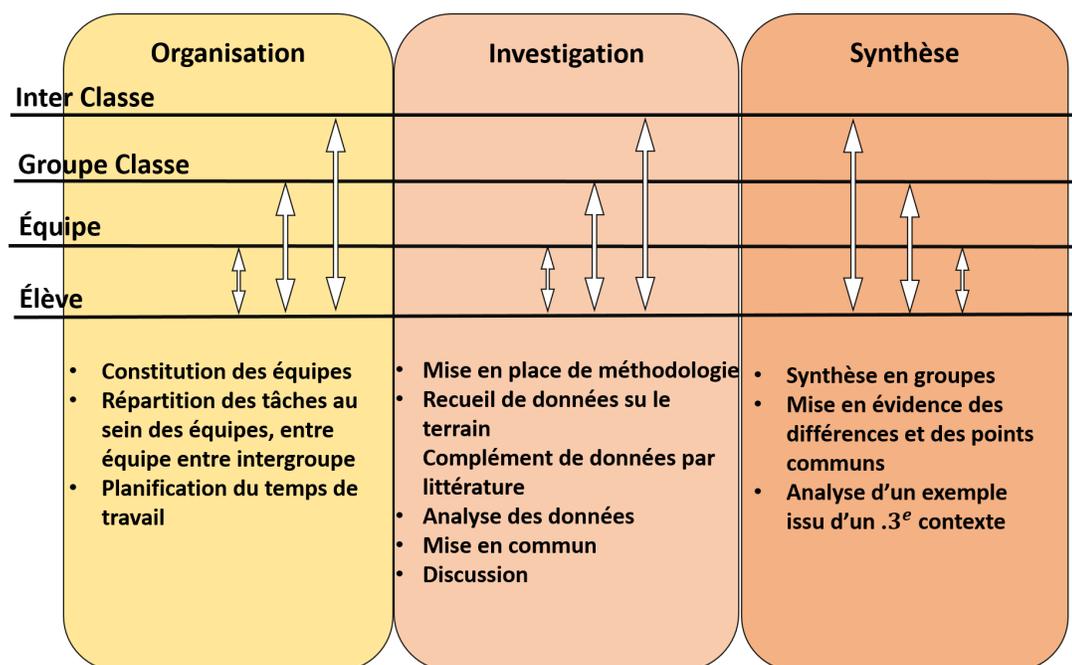


Figure 4.2: déroulement d'un scénario d'apprentissage donné par (Forissier, 2019)

dans de nombreuses situations d'apprentissage. Elle est inspirée du scénario de Jigsaw qui induit un double dynamique entre des groupes d'élèves de taille élevé et entre les équipes de chaque groupe (Aronson *et al.*, 1978) . Sur un sujet donné tel « la géothermie », « la grenouille » et « Dlo », c'est-à-dire durant une même itération, deux groupes de contexte différents se forment à partir des élèves et enseignants comme le montre la figure 4.2. Chaque groupe aborde les concepts en jeu à partir d'exemples inscrits dans son contexte. Un contexte dans notre étude est une localisation géographique, qui est le Québec ou la Guadeloupe. Ces groupes travaillent sur une thématique commune et forment des équipes au sein de chaque groupe. Ces équipes, qui étudient le même sujet, échangent les informations collectées et discutent les résultats obtenus de leur enquête entre eux, et puis avec leur groupe. La dernière phase de l'itérations les deux groupes de contexte différents (celui du Québec et celui de Guadeloupe) échangent et analyses leurs synthèses d'études, via une transfèrent de documents numériques et une vidéo-conférence.

La confrontation entre les différentes représentations des élèves sur un même sujet d'étude remet en question une construction mentale plus développée. Le scénario est implémenté dans un environnement numérique d'apprentissage Moodle qui comprend les fonctionnalités nécessaires pour la planification et l'accès au service de partage (Forissier *et al.*, 2014b) (Forissier, 2017), (Bourdeau, 2017), (Anjou *et al.*, 2017b).

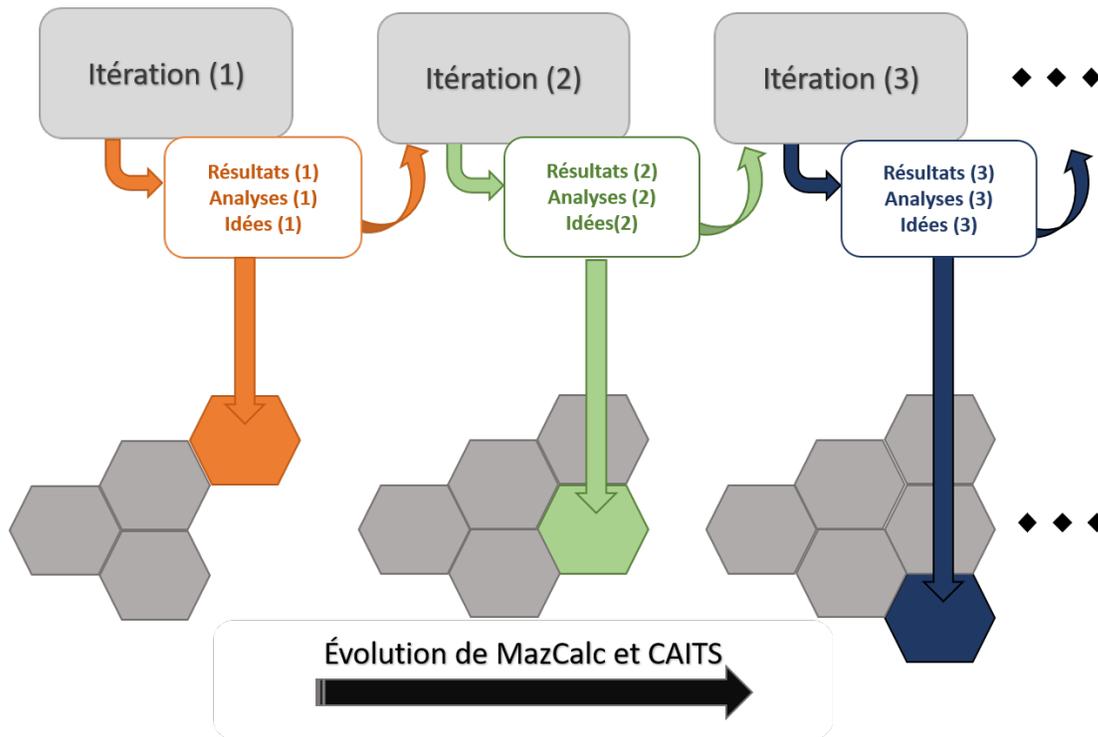


Figure 4.3: Déroulement de DBR de TEEC

À chaque itération, des nouvelles leçons sont apprises, de nouvelles idées sont développées et des propositions sont ajustées. Les données de chaque itération (celles qui sont en lien avec l'itération elle-même ou celle en lien avec l'objet d'étude) sont conservées. Elles serviront à tester les conceptions des logiciels en cours de développement (Bourdeau, 2017) : Les données en lien avec les objets d'études, c'est à dire les informations contextuelles, aideront à enrichir l'étude

analytique et conceptuelle de MazCalc (Figure 4.3. Tandis que les données en lien avec les itérations elles-mêmes, c'est-à-dire le scénario pédagogique et ses planifications, aideront à améliorer celles d'éditeur de scénarios et le système tutoriel intelligent CAITS.

### 4.3 CAITS

Le système tutoriel Intelligent Sensible au Contexte (CAITS- *Context Aware Intelligent Tutoring System*), comme tout autre système tutoriel Intelligent, a pour but de guider les apprenants durant leur apprentissage, mais en tenant compte du contexte. Il diagnostique les difficultés d'apprentissage rencontrées par les apprenants, et fournit des conseils adaptatifs pour soutenir l'apprentissage. Il prend en charge les scénarios d'apprentissage, les instruments à utiliser, ainsi que les activités d'apprentissage à faire par l'apprenant (Forissier *et al.*, 2014b) (Bourdeau *et al.*, 2015). CAITS inclut divers services et modules, où le calculateur de contexte MazCalc est son service clé (figure 4.4). Il fournit les données nécessaires sur les contextes à étudier. Les informations résultantes permettent aux auteurs de scénarios de faire des ajustements nécessaires via un système-auteur CEM. Ce système sera connecté à trois autres outils de gestion : CAS-Edit pour l'édition, Cas-Viz pour la visualisation et CAS-Sim pour la simulation. Le scénario résultat sera stocké par la suite dans la base de données CSLS. CAITS inclus aussi les trois modules CSDM (Context-Sensitive Domain Model), CSTM (Context-Sensitive tutor Model) et CSLM (Context-Sensitive Learner Model), qui sont des extensions des modules d'un STI classique (Bourdeau *et al.*, 2015) (Forissier *et al.*, 2014b).

En résumé, le système tutoriel intelligent adapté au contexte contient non seulement les modules de STI classiques (domaine, apprenant, tuteur) (Nkambou *et al.*, 2010), mais également deux composants liés au contexte, qui sont

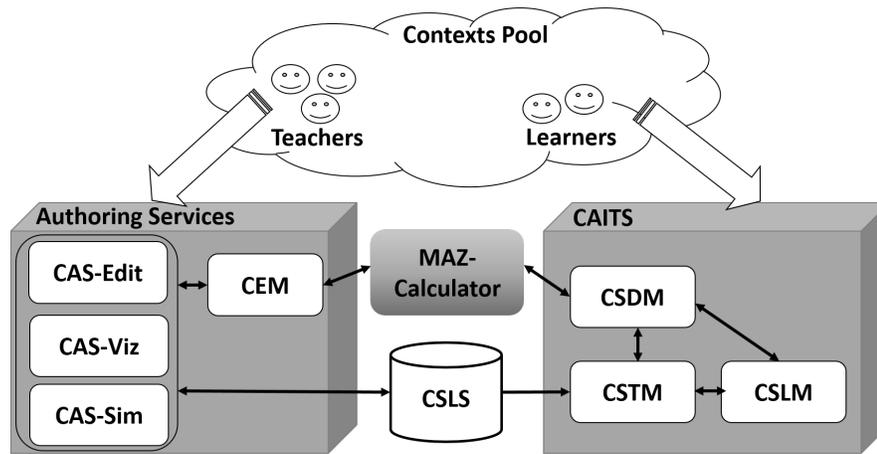


Figure 4.4: Architecture de CAITS donnée par (Forissier *et al.*, 2014b)

MazCalc , le modélisateur et calculateur de contexte, "Context Pool" qui est le gestionnaire de contexte, et un ensemble de services de création (CEM,CAS-Edit, Cas-Viz, CAS-Sim) pour le concepteur pédagogique ou l'enseignant afin d'adapter ou d'ajuster leur stratégie, et permettre aux chercheurs de tester leurs hypothèses (Forissier *et al.*, 2014b).

#### 4.3.1 MazCalc

Pour construit un système efficace sensible au contexte, une modélisation générique de contexte est nécessaire afin de couvrir tout son espace des cas et des états. Un modèle de contexte est un cadre qui permet de traiter chaque information et chaque paramètre du contexte disponible. Une fois que cette information est capturée, elle ne sera plus utile si elle n'est pas organisée et stockée sans perdre la sémantique. Nous avons donc besoin d'un mécanisme efficace d'organisation, de stockage et de récupération du contexte sous une version Web, appelé MazCalc. Cette version est indispensable pour visualiser les résultats d'avancement de notre calculateur de contexte. Une fois capturé et organisé, le contexte doit être utilisé dans des situations par des applications, entre autres CAITS. Par conséquent,

il doit exister un point d'intégration où les applications peuvent accéder à ce contexte et partager ses informations, et disposer de services pertinents disponibles qui aideront les apprenants à étudier les effets de contextes, d'où la nécessité de générer une version API de MazCalc.

MazCalc est un outil informatique qui a été proposé afin de modéliser un objet d'étude dans son contexte géographique et de calculer les écarts entre des contextes différents pour ce même objet d'étude et de prédire leurs effets. Modéliser un objet d'étude, revient à définir ces paramètres relatifs au contexte et ceux indépendants du contexte (absolus). Les paramètres contextuels ont tendance à changer de valeurs d'un contexte à un autre, tandis-que les paramètres absolus restent inchangés, ou presque, et gardent leurs valeurs quel que soit le contexte géographique (Psyché *et al.*, 2018).

L'outil MazCalc devrait être donc générique, dans le sens où il pourra être étendu selon le besoin à toutes formes de modèle de contexte basé sur la localisation, et qui permet, suite à une modélisation d'un objet étude, de calculer l'écart entre deux contextes mis en jeu. L'intervention des experts didactiques dans le processus de modélisation effective des différents contextes est donc nécessaire afin de bien identifier la structure générale du modèle de contexte, ainsi que les domaines ou les familles de ses paramètres associés.

Le calcul de l'écart de contexte est assuré par un modèle comportant des formules de calcul selon la nature du paramètre, ainsi que des algèbres de structures pour manipuler globalement des instances de contextes afin de mieux faire ressortir l'effet de contexte (ou le clash). Le résultat de ce calcul couplé aux contraintes pédagogiques techniques (durée de la formation, période, moyens disponibles. . .) permettent ensuite aux enseignants de créer un scénario pédagogique (éditeur de scénario) incluant des démarches d'investigation et des collaborations entre élèves

des deux contextes. Ce scénario est construit de manière à donner une importance aux écarts contextes afin de favoriser l'émergence des effets de contextes.

La finalité du MazCalc intégré est de faire en sorte qu'il soit facilement utilisé pour étendre un système hôte afin de lui conférer une sensibilité au contexte. C'est dans cette perspective que le projet poursuivra une démarche d'intégration de MazCalc dans CAITS, le système tutoriel intelligent pour l'apprentissage des sciences en contexte .

Après avoir expliqué la motivation pour concevoir l'application MazCalc, nous devons procéder à la conception et la réalisation du projet. Tous au long du prochain chapitre, nous décrirons notre démarche de travail ainsi que nos choix de méthodologie et de conception. Cette étape est très importante et sur celle qui repose l'échec ou le succès du projet.

## CHAPITRE V

### ANALYSE CONCEPTUELLE DE MAZCALC

Après avoir établi le cadre théorique du projet et présenté la problématique justifiant le développement de MazCalc, nous devons élaborer la conception détaillée de la solution en suivant une méthodologie appropriée pour pouvoir bien décrire les besoins du système, ses contraintes les différentes fonctionnalités de la solution souhaitée. L'objectif de ce chapitre est de présenter l'architecture et les éléments conceptuels du «MazCalc» tout en prouvant que cette conception permet de produire un système générique. «MazCalc» est le modèle de contexte qui sera intégré dans une prochaine étude dans le système tutoriel intelligent « CAITS », et qui sera le noyau de ce dernier.

#### 5.1 Méthodologie AGILE

La gestion de projet est une démarche assurant le bon déroulement du projet avec une planification et une gestion des différentes tâches. En rappel, notre problématique se situe à l'intersection de l'enseignement en contexte et de la technologie éducative.

Nos hypothèses scientifiques portent sur l'adaptabilité de cette innovation à différentes disciplines et plusieurs niveaux scolaires. Le rôle de la technologie informatique ici est de modéliser le contexte pour prédire les effets du contexte

et leur potentiel d'apprentissage productif et favoriser l'apprentissage des sciences dans un environnement d'apprentissage intelligent intégré mais mobile.

Deux branches d'études, donc deux méthodologies de travail sont à choisir pour le projet. Une première méthodologie est celle pour l'étude la contextualisation didactique, afin d'acquérir les connaissances nécessaires pour favoriser le succès de la recherche, et la deuxième est une méthodologie de génie logiciel pour l'analyse, la conception et le développement de l'outil de modélisation du contexte, "MazCalc". Ces deux méthodologies doivent coïncider avec les caractéristiques du projet TEEC. Ce dernier est un projet multidisciplinaire (une équipe multidisciplinaire) dans laquelle nous retrouvons un peu de tout comme, des experts du domaine de l'éducation, des experts du domaine du contexte, des cognitivistes, des experts du domaine STI, des ingénieurs... etc. l'implication active des acteurs du projet aux processus décisionnels (planification, mise en œuvre et évaluation) est indispensable. Durant des réunions d'interactions, les membres de l'équipe ont échangé leurs connaissances du domaine et ont discuté leurs idées. Toutes ces connaissances élicitées du domaine doivent être documentées de façon minutieuse et validées par toute l'équipe. Le choix des techniques à utiliser pour l'encodage du «MazCalc» doit être aussi étudié et validé. Il faudra aussi documenter les changements des besoins fonctionnels qui auront lieu sans doute vu le caractère dynamique des projets TEEC et MazCalc. En effet, comme ces projets sont purement basés sur la recherche, il est normal d'explorer différentes idées d'où le changement de besoins avec chaque nouvelle idée.

La méthode DBR est appropriée pour à la fois tester des hypothèses et améliorer l'innovation, en travaillant de concert avec les acteurs sur le terrain. Nous avons détaillé l'utilité et la mise en œuvre de DBR dans la section 4.2 du chapitre 4. Dans l'ingénierie de développement informatique, la méthodologie **Agile** adopte elle aussi une approche participative et met l'accent sur la collaboration

et privilégie les individus et leurs interactions par rapport aux processus et aux outils (Kent Beck, 2001). Elle favorise l'adaptation au changement plutôt que la conformité aux plans. Agile a été discutée pour la première fois en profondeur dans les années 1970 par William Royce qui a publié un article sur le développement de grands systèmes logiciels (Royce, 1970).

Plus tard en 2001, 17 développeurs de logiciels ont publié "agile manifesto" (Kent Beck, 2001). Agile manifesto est une proclamation formelle de quatre valeurs (les individus et les interactions sur les processus et les outils, les logiciels de travail sur une documentation complète, la collaboration avec le client sur la négociation des contrats et la réponse au changement au sujet d'un plan) et douze principes. Ces principes ont pour but de guider la gestion de projet agile en se reposant une approche itérative, incrémentale et adaptative. Nous avons opté pour la méthode Scrum d'agile (Schwaber, 1997) pour gérer le développement de l'application MazCalc. Cette méthode fournit aux développeurs et aux équipes un moyen pour produire un meilleur logiciel, plus rapidement grâce à un processus de sous-tâches itératifs et interactifs de courte durée comme l'indique la figure 5.1.

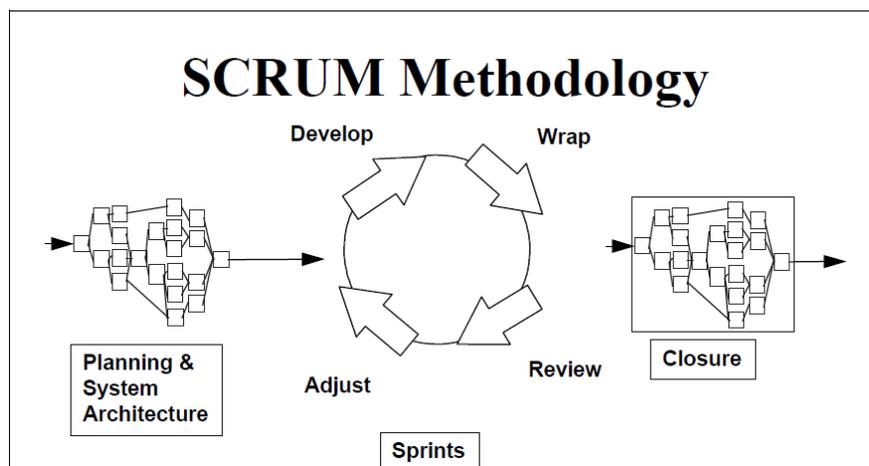


Figure 5.1: le processus de la Méthodologie Scrum (Schwaber, 1997)

Selon la méthode Scrum, tout projet de développement logiciel se déroule sur plusieurs phases. Au début, une phase de planification est nécessaire en suivant une communication en étoile avec les membres de l'équipe. La planification comporte l'identification des besoins du logiciel ainsi que leurs priorités sur une liste des tâches, appelées aussi "Backlogs". Un Backlog est une fonctionnalité globale du logiciel, comme "la gestion du modèle d'un objet d'étude", "la gestion des contextes", "la gestion des acteurs" ...etc. Chaque Backlog donc peut être devisé en un ensemble de sous-tâches, appelées "sprints" (Rising et Janoff, 2000).

Une fois un sprint est lancé, aucun changement n'est permis. Un sprint dure obligatoirement entre une et quatre semaines (Schwaber, 1997). Au cours d'un sprint, l'équipe de développement peut réduire la fonctionnalité à livrer pendant cette période, cependant la date de livraison reste inchangée (Rising et Janoff, 2000).

La finalité d'un sprint est un produit visible, construit à partir d'une ou de plusieurs implémentations rajoutées aux implémentations précédentes du produit global. Les sprints sont non-linéaires, flexibles et tendent à corriger le produit final. Après chaque sprint, une réunion se produit entre toutes les membres du projet. Toutes les informations du sprint clôturé sont rapportées. A ce stade tout peut changer : Les parties discutent ce qui peut être ajouté, supprimé ou rectifié et un prochain sprint est planifié. Le projet reste ouvert au changement jusqu'à la phase de clôture du projet (Rising et Janoff, 2000). Les sous-sections qui suivent, présentent les différentes composantes de notre solution.

## 5.2 MazCalc : vue Globale

En vue de bien concevoir notre application MazCalc, nous devons élaborer une étude de besoins pour cerner les différentes attentes du système qui consti-

tueront les objectifs à atteindre pendant la réalisation.

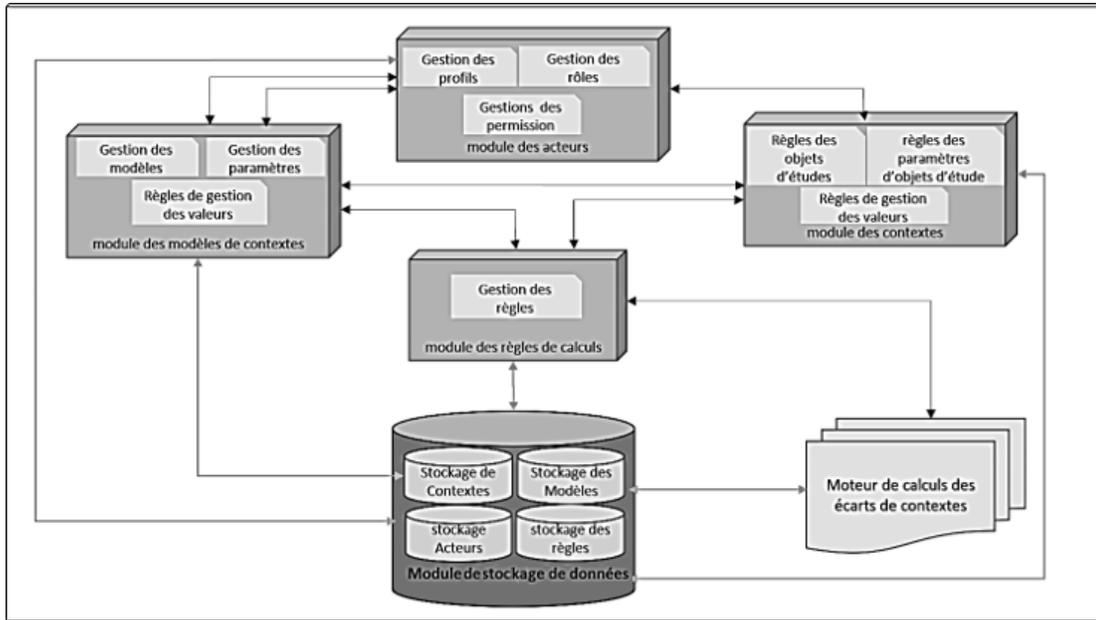


Figure 5.2: Les composants du Système MazCalc

Commençons tout d'abord par savoir de quoi MazCalc est composé. Notre Système comporte quatre composants : Composant "**modèles**", qui englobe un module gestionnaire des modèles d'objets d'étude et un module de stockage des informations correspondantes, Composant "**contextes**" qui contient un module gestionnaire des objets d'étude de contexte et son modules de stockage des informations contextuelles, Composant "**Acteurs**" composé d'un module gestionnaire des acteurs qui s'occupe des rôles et des permissions d'accès et un module de stockage correspondant et enfin, un Composant "**règles de calcul**" qui comporte un module de gestion des règles de calcul d'écart de contexte , son module de stockage et le moteur d'exécution de ces règles comme montré dans la figure 5.2.

Tous les modules interagissent entre eux selon des règles de communication et de permission : ces règles définissent "qui fait quoi?", "avec quelle permission",

"comment le faire?" et "Quand le faire". C'est ce qu'on appelle "les besoins fonctionnels" de notre système.

### 5.2.1 Composant Acteurs

Avant de répondre aux questions mentionnées en sus, la première tâche consiste à définir les limites du système (c.à.d. ce qui est inclus ou pas dans le système), puis à identifier les différentes entités en dehors de ce système mais qui interagissent avec lui. C'est ce que nous avons appelé les acteurs. Un acteur est donc la description d'un ensemble cohérent de rôles qu'un utilisateur joue lorsqu'il interagit avec le système. Chaque acteur doit avoir un nom qui le distingue des autres acteurs.

MazCalc est un modèle de contexte qui servira comme un outil supportant l'élaboration des scénarios d'apprentissage. Il doit impérativement permettre les interactions avec les enseignants, les concepteurs des modèles et les experts des objets d'études (experts du domaine) dans leur contexte. Chacun d'entre eux possède un rôle principal et bien spécifique dans le processus d'élaboration du contexte et de production des scénarios pédagogique. Nous identifions alors le rôle de chaque acteur comme suit :

- **acteur 1** : Concepteur expert de l'objet d'étude
- **acteur 2** : Spécialiste de l'objet d'étude dans son contexte.
- **acteur 3** : Enseignant.

Pour faciliter la communication et la manipulation des noms durant le processus de développement, nous appelons respectivement les acteurs comme suit : "Concepteur expert", "Spécialiste de contexte" et "Enseignant". Ces acteurs ont des tâches différentes (quoique complémentaires). Chacun apporte son savoir-faire à un ou plusieurs aspects depuis la conception du modèle contextuel jusqu'à l'étude

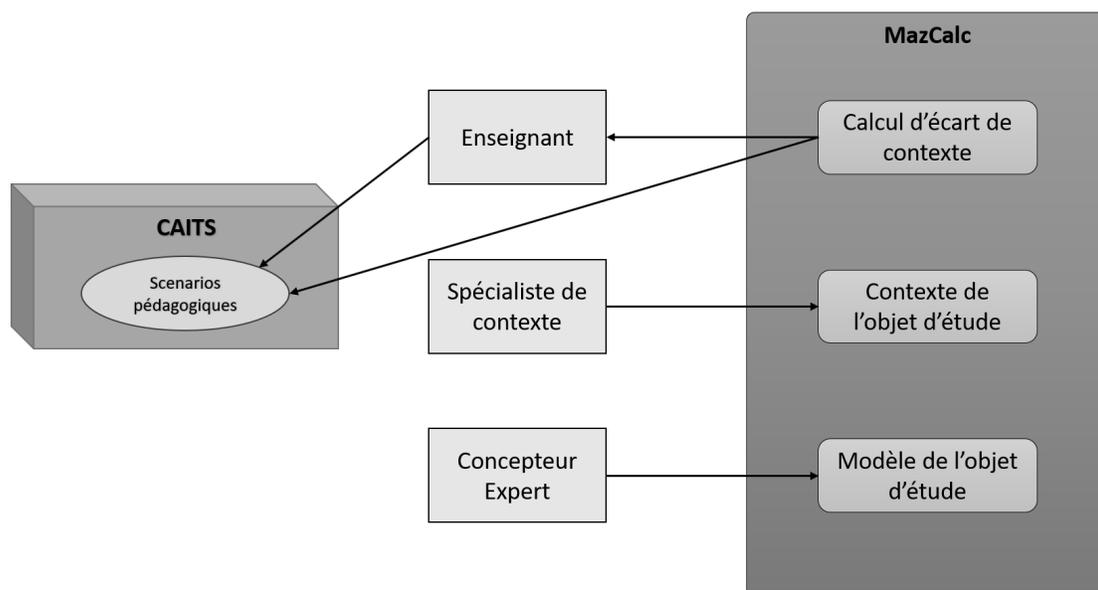


Figure 5.3: Les rôles des Acteurs utilisant MazCalc

analytique et comparative des attributs et des objets d'études (dans différents contextes).

Le rôle du Concepteur expert est la gestion d'un modèle d'un objet d'étude donnée indépendamment de son contexte. Cela porte essentiellement sur la gestion des paramètres du contextes, leurs familles et les différentes valeurs possibles que peuvent.. Le rôle du Spécialiste de contexte est de s'occupe de la création et la gestion d'un objet d'étude dans son contexte géographique. Il participe à l'instanciation d'un modèle de contexte (à partir du méta-modèle) à un contexte particulier. Alors que le rôle des Enseignants est d'accède aux résultats élaborés par le concepteur et l'expert , ainsi qu'aux résultats d'écart entre deux contextes donnés comme dans la figure 5.3. Un des besoins qui a été étudié est d'assurer le respect des responsabilités de chacun. Il faut ainsi répartir le système en différents modules puis donner accès aux utilisateurs seulement aux sections de l'application dont ils ont besoins. Un acteur ne devrait pas accéder aux fonctionnalités

d'un autre acteur sauf s'il possède lui aussi le rôle correspondant comme un rôle supplémentaire. Un acteur peut ainsi avoir plusieurs rôles à la fois et donc plus de permissions d'accès aux fonctionnalités. Par exemple, un acteur **A** peut être spécialiste du contexte relatif à un modèle **M1** mais peut aussi être concepteur pour un modèle différent **M2**. Le système devra donc autoriser cet acteur à gérer le contexte et les objets d'études pour le modèle **M1** seulement et d'ajouter les paramètres, leurs familles et les valeurs possibles pour le modèle **M2**.

### 5.2.2 Diagramme de cas d'utilisation global

Dans une première phase d'analyse, les grandes fonctionnalités de MazCalc sont identifiées implicitement en se basant sur les rôles des acteurs. Ces fonctionnalités ne sont pas indépendantes en réalité, comme nous l'avons vue. Il existe des interactions entre-elles en plus des interactions avec les acteurs. Pour apporter une vision plus claire et plus compréhensible de ces interactions, nous faisons recours au diagramme de Cas d'utilisation global de Langage de Modélisation Unifié (Unified Modeling Language : UML) sous forme d'une représentation graphique. Le diagramme de cas d'utilisation global a pour but de structurer nos attentes en modélisant les besoins de nos acteurs, spécifiant les grandes fonctionnalités et les limites du système et représentant les interactions entre eux.

Comme le montre le diagramme de cas d'utilisation global présenté à la figure 5.4, pour que MazCalc puisse garantir à chaque acteur ses besoins propres à lui, ce dernier doit s'identifier avant d'accéder à n'importe quel service ou ressource. Un autre point que montre le diagramme, c'est qu'il y a une hiérarchie entre les fonctionnalités du système. Il est impossible de faire la gestion des objets d'étude dans leur contexte, avant de faire la gestion du modèle de l'objet d'étude en question (c-à-d la création du méta-modèle du contexte de l'objet d'étude).

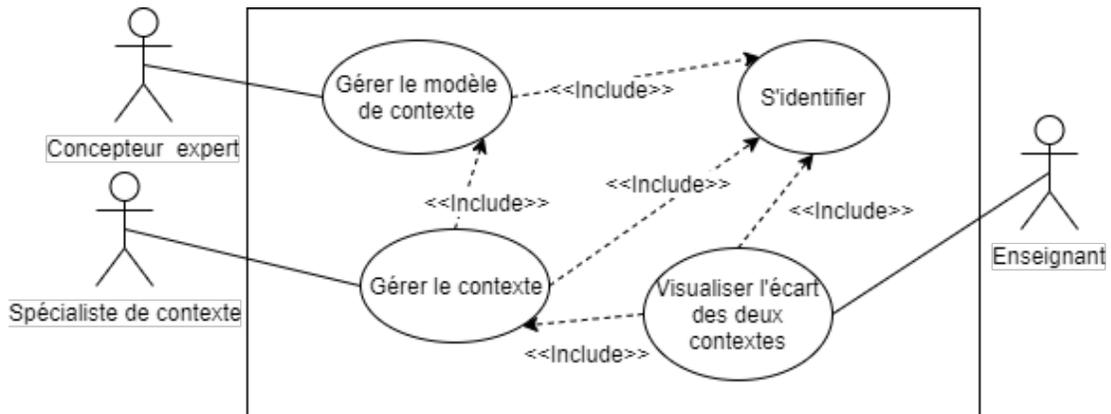


Figure 5.4: Cas d'utilisation global

En d'autres termes, le modèle de l'objet d'étude doit exister avant l'objet d'étude dans son contexte. De même pour l'accès aux résultats des calculs d'écart et les analyses. Les calculs se font dans la troisième position après que le modèle de l'objet dans son contexte a été créé (l'instanciation pour un contexte particulier). Nous avons choisi UML comme langage de modélisation pour profiter de la possibilité de traduire les diagrammes UML en classes et puisque nous avons opté pour une approche objet de l'implémentation (justifiée au prochain chapitre). Il sera donc utilisé tout au long de ce chapitre pour détailler notre étude conceptuelle de MazCalc.

### 5.3 Analyse des composants

Après une analyse approfondie de la littérature, étudiée tout au long du chapitre 2, nous avons tiré toutes les propriétés génériques caractérisant un contexte que nous avons jugées utiles et pertinentes pour notre étude. Ces caractéristiques ont été prises en considérations lors du codage de MazCalc, donnant lieu à un ensemble d'assertions :

- Un contexte est défini relativement à un objet d'étude, qui peut être une gre-

nouille, la géothermie...etc

- Un même objet d'étude peut être défini dans différents contextes (géographique)
- L'objet d'étude en question doit être généralisé dans un modèle indépendamment du contexte pour pouvoir l'étudier dans différents contextes, que nous appelons "un modèle".
- Un modèle d'un objet d'étude comporte un ensemble de paramètres fonctionnels que nous appelons des constantes contextuelles qui peuvent être d'ordre géographique, politique, social, historique, culturel ou économique et sont indépendantes du contexte.
- Un paramètre peut être qualifié ou quantifié donc le paramètre a un type soit "Qualitative" ou "Quantitative".
- Un paramètre a une valeur spécifique à un contexte qui peut changer d'un contexte à un autre car les informations contextuelles possèdent un caractère spatial. Un paramètre d'un modèle (généralisant) doit avoir une variété de valeurs, qui se fixera par la suite lorsque le contexte est fixé.
- La liste des valeurs peut être continue ou discrète, ordonnée ou non. Ces valeurs peuvent être calculées, mesurées ou données.
- Un paramètre peut dépendre d'un autre paramètre et peut donc être composé d'un paramètre parent. Pour un paramètre composé, chaque valeur de la liste du parent définit une liste de valeurs du paramètre fils. Ainsi, le paramètre fils (le composé) possède un dictionnaire de valeurs comme suit :

$$((Key_1, (a_1, a_2, a_3 \dots, a_n)), (Key_2, (b_1, b_2, b_3 \dots b_p))) \dots ((key_m, (v_1, v_2, v_3 \dots v_x)))$$

- Le type de paramètre et sa nature définissent sa règle de calcul d'écart.
- Un paramètre appartient à une ou plusieurs familles.
- Une famille groupe un ensemble de paramètre et peut être de type : "domaine d'observation", "échelle d'observation" (macro, micro..) ou "thématique d'obser-

vation".

- La source de la valeur d'un paramètre d'un objet d'étude dans un contexte donné doit être indiquée pour assurer la qualité de contexte et des informations contextuelles et donner une idée sur sa précision. Cette source peut être un dispositif de mesure, un capteur, un téléphone...etc
- Le degré de confiance que la valeur d'un paramètre d'un objet d'étude dans un contexte peut avoir doit être indiqué pour répondre aux critères d'un contexte qui est "la confiance" à donner à la source de l'information.
- La date de génération d'un objet d'étude dans son contexte doit être conservée, pour donner une estimation sur l'âge des informations contextuelles sauvegardées.
- Si les informations contextuelles d'un objet d'étude changent au fil du temps (caractère temporel ou dynamique du contexte), il est possible de définir de nouveau le même objet d'étude avec ses nouvelles informations contextuelles. Les anciennes informations contextuelles resteront sauvegardées pour garder une trace sur l'évolution contextuelle.

Ces assertions (ou propriétés) sont des éléments à qui doivent être vérifiés dans le système MazCalc. Ils serviront comme des éléments à encoder dans le système MazCalc. A partir de ces assertions, nous avons dégagé les trois composants principaux pour notre application pour la gestion des modèles, la gestion des contextes et la gestion des règles de calcul.

### 5.3.1 Le composant Modèles

Après que les concepteurs pédagogiques aient choisi le scénario d'apprentissage en dehors de MazCalc, l'objet d'étude ainsi que ses contextes en étude sont identifiés et devront être modélisés. A ce niveau plusieurs concepteurs d'un objet d'étude collaborent pour donner le squelette le plus général de cet objet d'étude

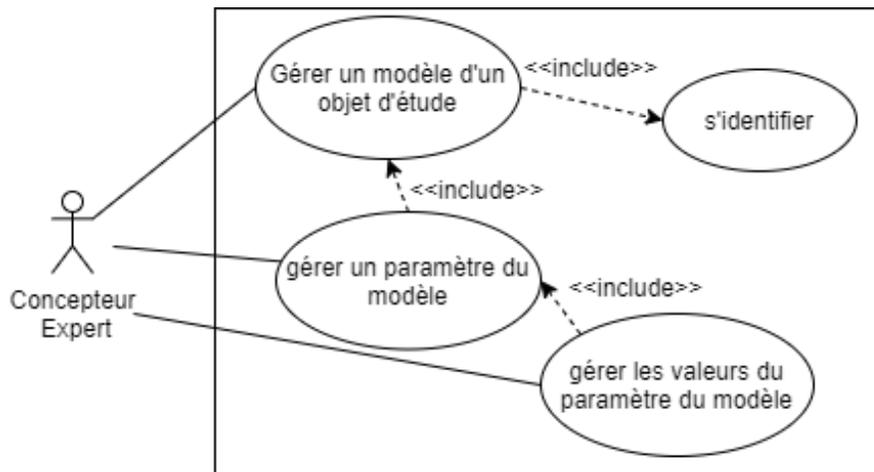


Figure 5.5: Diagramme de cas d'utilisation du Concepteur du modèle

à partir de son étude dans tous les contextes possibles. Ils doivent donc spécifier tous les paramètres possibles qui peuvent entrer en jeu. MazCalc doit donner la possibilité au concepteur de saisir les paramètres de l'objet d'étude autant de fois que son nombre (Figure 5.5).

Chaque paramètre, comme nous l'avons mentionné dans la section précédente, possède un type bien précis qui définit sa règle de calcul d'écart, et bien évidemment possède une valeur, mais cette valeur change d'un contexte à un autre, donc ce paramètre doit être défini avec une variété de valeurs. La marge des valeurs et la règle de calcul dépendent aussi du fait que le paramètre soit de type composite ou non. D'où, notre système doit permettre au concepteur d'introduire le nom du paramètre, sa règle de calcul, de choisir les familles auxquelles il appartient, et s'il est composite, quantitatif ou qualitatif. Dans le cas où le paramètre est composite, le système doit monter au concepteur la liste des paramètres déjà créés pour choisir le paramètre parent correspondant, et dans une prochaine étape il introduit les valeurs. Dans ce cas le paramètre parent doit être de type qualitatif, pour pouvoir offrir la possibilité d'introduire pour chaque valeur du paramètre

parent une liste de valeurs du paramètre fils. Le concepteur arrive à itérer sur la liste des valeurs du paramètre parents que MazCalc lui affiche, et pour chaque paramètre choisi, il lui introduit la liste des valeurs qui en dépend. Dans le cas où le paramètre n'est pas composite, nous trouvons deux cas de figure pour introduire ses valeurs qui sont :

- Si le paramètre est quantitatif, la valeur dans un contexte donné est une valeur numérique discrète. Si nous collectons toutes les valeurs possibles de tous les contextes, nous pouvons borner ces valeurs. Notre solution devrait alors permettre de donner les bornes de l'intervalle de ces valeurs (une seule valeur minimale, et une seule valeur maximale).
- Si le paramètre est qualitatif, ses valeurs sont de type « floue » c'est-à-dire non numériques, et l'ordre des valeurs peut impacter la règle de calcul :
  - o Dans le cas où l'ordre n'est pas important, ce type s'appelle « QualitatifÉnuméré ». MazCalc doit dans ce cas permettre d'introduire la liste des valeurs de ce paramètre un a un.
  - o Dans le cas où l'ordre est important, ce type s'appelle « QualitatifOrdonné ». Dans ce cas MazCalc doit permettre la liste des valeurs un a un et pour chaque valeur il doit permettre d'introduire l'ordre de cette valeur.

### 5.3.2 Composant Contextes

Ce composant est destiné, comme nous avons mentionné, aux spécialistes du contexte. Les spécialistes du contexte commencent par sélectionner à partir du modèle, construit par les concepteurs de l'objet d'étude dans un niveau précédent, le sous-ensemble des paramètres de l'objet d'étude qui entrent en jeu dans leur contexte et qu'ils jugent pertinents. Certains paramètres peuvent être pertinents dans un contexte donné mais non pertinent dans un autre comme par exemple la notion des Saisons climatiques qui n'existe pas dans les zones tropicales où la

température est de 18 degrés durant toute l'année, mais au Québec quatre saisons sont reconnues : "Été" , "Automne", "Hiver" et "Printemps" (Forissier *et al.*, 2014b). Ce paramètre fait partie des caractéristiques du climat, mais il n'est pas défini dans le contexte "région tropical". Si on essaye d'étudier l'objet d'étude "Climat", les concepteurs experts définiront le paramètre "saison" dans le modèle de l'objet d'étude. Au moment de l'instanciation de ce modèle dans des contextes donnés, le spécialiste du "Climat" au Québec choisit le paramètre "saison" comme un paramètre pertinent pour définir le climat du Québec. Cependant, pour le spécialiste du "Climat" d'une région tropicale, le paramètre "saison" sera ignoré lors de l'instanciation de l'objet d'étude "Climat" dans son contexte "région tropicale". Pour chaque paramètre choisi, le spécialiste doit spécifier sa valeur mesurée,

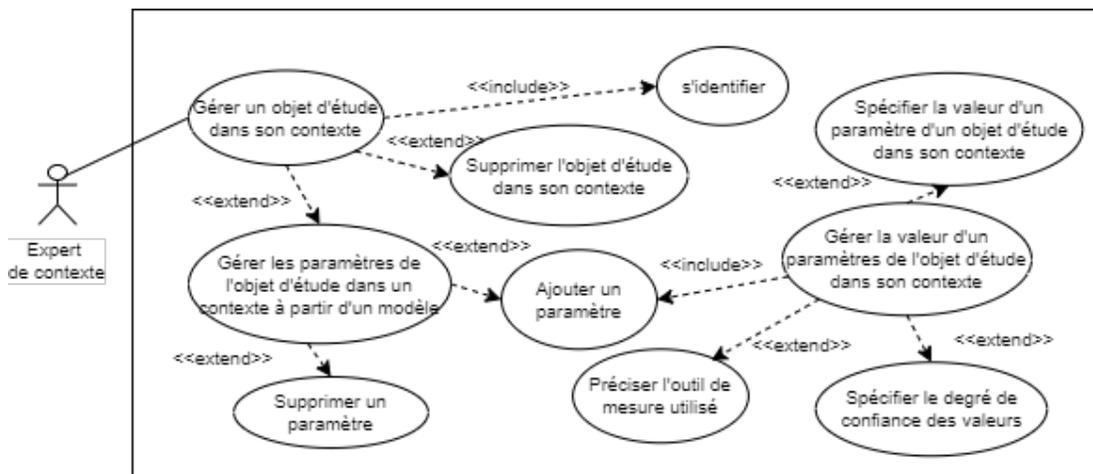


Figure 5.6: Diagramme de cas d'utilisation de l'expert du modèle dans son contexte

calculée ou estimée dans son contexte, et il doit indiquer la source de cette valeur et son degré de confiance. La valeur a spécifié peut-être donnée doit explicitement par un simple saisi dans un champ correspondant : Ce cas est pour les paramètres de type quantitatif, soit par une sélection d'une seule valeur parmi la liste des valeurs du paramètre en question : ce cas est pour les deux autres types. Le Maz-

Calc doit permettre alors aux spécialistes de faire tous ces tâches. La définition de l'objet d'étude dans son contexte est terminée à ce niveau comme indiquée dans le diagramme de cas d'utilisation de la figure 5.6. Il est possible d'étendre, si de nouvelles informations contextuelles se présentent, la liste des paramètres d'un contexte donné en ajoutant un ou plusieurs nouveaux paramètres existants dans le modèle de l'objet d'étude.

### 5.3.3 Composant Règles de Calcul

Après avoir instancier les contextes, il est possible à ce stade de choisir 2 ou plusieurs contextes d'un même objet d'étude pour enseigner leurs écarts de contexte (figure 5.7) . Uniquement deux contextes peuvent être comparés à la

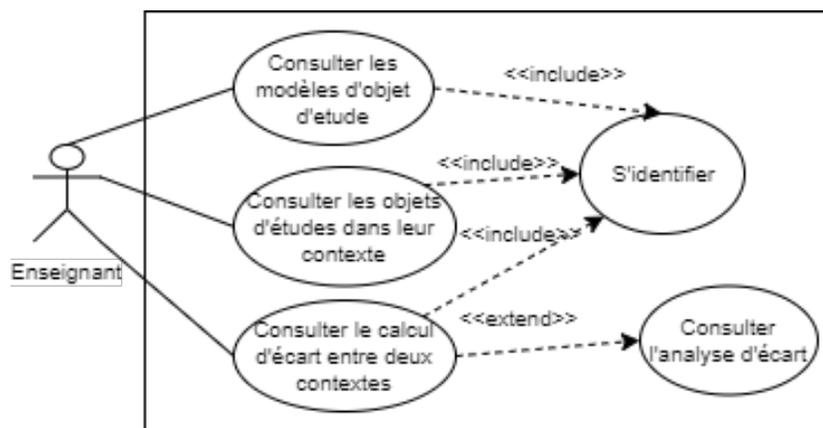


Figure 5.7: diagramme de cas d'utilisation de l'enseignant

fois. Si nous avons trois contextes à étudier **C1**, **C2** et **C3**, nous commençons par calculer l'écart entre **C1** et **C2**, puis **C1** et **C3** et enfin entre **C2** et **C3**. Comme nous avons mentionné dans les sections précédentes, à chaque paramètre contextuel est attribué une règle qui décide les lois de calcul d'écart entre les contextes. Le calcul d'écart se fait sur deux niveaux. Le premier niveau est le niveau des familles, qui est le groupement des paramètres contextuelles. Le second est au niveau des

paramètres. C'est-à-dire un même paramètre sera fixé dans les deux contextes, et nous appliquons la règle de calcul correspondante en introduisant ses valeurs dans la formule de calcul. Ce processus se fait de manière itérative.

En résumé, le calcul d'écart de contextes est un processus d'alignement entre deux contextes qui met en évidence les différences des valeurs des paramètres, et donc la différence entre les modèles. Nous tirons trois règles de calcul d'écart au niveau d'un paramètre. Ces règles ont été identifiées par des spécialistes de contexte (Anjou *et al.*, 2017a) qui sont :

- comparaison : le calcul d'écart est une comparaison entre les valeurs d'un paramètre. Si le paramètre a la même valeur dans les deux contextes, alors l'écart est égal à 0, sinon l'écart est égal à 1.

- éloignement : le calcul d'écart représente la marge de la différence entre les valeurs d'un paramètre. C'est ici que l'ordonnancement des valeurs prend lieu. L'ordre de chaque valeur par rapport dans la liste des valeurs est important comme nous avons indiqué dans la section précédente. Le résultat du calcul d'écart alors est l'éloignement des deux valeurs du paramètre.

- composite : le calcul d'écart est une pondération. Cette règle concerne les paramètres composites. Si l'écart du paramètre parent égale à 0 alors on applique la règle de calcul de comparaison sur les valeurs du paramètre fils. Le résultat de l'écart et la moitié du résultat retourné.

Afin de pouvoir appliquer ces trois règles de calcul convenablement, nous faisons recourt à une normalisation des informations contextuelles, c'est-à-dire les valeurs des paramètres sont transformées en valeurs numériques.

## 5.4 Besoins non fonctionnels

En plus de répondre aux besoins des utilisateurs de notre application, il faut assurer la manière avec laquelle ces buts sont atteints. On parle alors de qualité de logicielle.

En effet, comme la qualité nous est importante, nous avons essayé aux meilleures de nos compétences, avec le temps limité disponible, de produire un logiciel de qualité. A été notre objectif de conformité Pour ce faire, nous avons eu comme objectif de satisfaire les contraintes de la norme ISO-25010 (Estdale et Georgiadou, 2018). La norme ISO/IEC 25010-2011 liste 8 caractéristiques principales pour la qualité d'un logiciel, chacun ayant des sous caractéristiques : l'adéquation fonctionnelle, l'efficacité de performance, la compatibilité, l'utilisabilité, la fiabilité, la sécurité, la maintenabilité et la portabilité.

Tout d'abord, l'objectif principal est comme déjà mentionné d'assurer les tâches requises par les utilisateurs de MazCalc en répondant à tous les besoins fonctionnels de façon correcte, complète et appropriée. Nous assurons ainsi l'adéquation fonctionnelle. Ensuite, notre logiciel devra être efficace et performant et ainsi réagir dans un délai raisonnable, pour retourner les résultats requis aux utilisateurs tout en optimisant l'utilisation des ressources. MazCalc vérifie aussi la compatibilité vue que nous devons assurer la communication facile avec d'autres systèmes existants ou qui seront implémentés dans le futur. Nous devons accepter plusieurs formats standards de transfert de données (essentiellement JSON et XML). Un autre point important est d'exposer une documentation facile pour permettre aux développeurs d'autres systèmes d'interagir avec MazCalc sans poser de contraintes quant à la technologie utilisées. L'intégration devra être indépendante des langages de programmations. L'utilisabilité de la solution se manifeste dans la présentation d'interfaces homme machine intuitives, faciles à utiliser qui

permettent l'administration facile. Tous les formulaires d'entrée de données ont une capacité de validation pour améliorer l'expérience de l'utilisateur.

Comme nous avons utilisé une méthode agile, le système était disponible aux utilisateurs pour tester les fonctionnalités et avoir leurs remarques. MazCalc semble facile à utiliser et accessibles en tout temps car publié sur internet. Du côté de la fiabilité, le système doit être disponible en tout temps pour permettre à plusieurs équipes géographiquement éloignées (différents fuseaux horaires) d'accéder aux services. Le système continuera à fonctionner dans le cas de fautes et nous aurons des messages d'erreurs. Sauf dans le cas d'erreur fatale, le système sera contraint de s'arrêter. La sécurité du système est résultat de la combinaison de plusieurs techniques et standards (Https, OAuth, TLS, cryptographie...) et ce afin d'assurer les différents services de sécurité : la confidentialité, l'intégrité, la non-répudiation, la disponibilité et l'authentification.

L'utilisateur devra créer son propre compte. L'administrateur de MazCalc sera responsable ensuite d'activer le nouveau compte et attribuer les permissions nécessaires. Les permissions des utilisateurs doivent être explicites. On suit la bonne pratique de donner toujours le minimum nécessaire de privilèges. Mazcalc permet l'interaction avec des utilisateurs humains (Web) et logiciels (intégration à travers une API). Ceci a des implications au niveau de la sécurité en utilisant les techniques standards et recommandées pour chaque scénario. Ceci sera expliqué davantage dans le prochain chapitre. Les données du système doivent être cohérentes et intègres à tout moment. Ceci sera assuré principalement par l'utilisation d'un SGBD que nous expliquerons davantage dans le chapitre prochain. En ce qui concerne la portabilité, comme le code est modulaire avec un découplage maximal, il sera facile de remplacer les composants au besoin sans affecter le reste du système. .NET nous permet aussi d'avoir plusieurs moyens pour faciliter de déploiement des modifications (FTP, MsDeploy ou via un pipeline CI/CD

totalemment automatique). Le processus de déploiement prend quelques secondes.

Enfin, et ce n'est pas le moins important, pour garantir que la solution est maintenable, nous avons mis en place différents points de vérification, de "bilan de santé" (health check) en utilisant des techniques standards (fichier de log, redirection vers pages d'erreur, envoi d'email de notifications ..). L'architecture et le code doivent aussi suivre les patrons de conception pour assurer une maintenance facile et une évolution du code. Après avoir présenté une vue globale et détaillée sur des besoins fonctionnelles et non fonctionnelles de notre Système MazCalc, il est nécessaire de modéliser les détails des interactions Acteurs-Système pour trouver une bonne représentation des composants ainsi que le bon séquençement des scénarios de réalisation des "tâches acteurs".

## 5.5 étude conceptuelle dynamique

### 5.5.1 diagrammes de séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Nous avons mentionné dans notre étude que chaque acteur a des besoins spécifiques envers le MazCalc. Le système doit disposer d'une fonction d'"authentification" comme détaillée dans la figure 5.8.

L'acteur commence tout d'abord par saisir ses données de connexion sur l'interface d'authentification. Ces données ont été sauvegardées dans la base de données de MazCalc lors de la création d'un profile sur le système. L'interface commence par vérifier en local : si les données saisies sont manquantes elle affiche un message d'erreur a l'acteur. Sinon, elle envoie ces données à un contrôleur responsable de l'authentification des utilisateurs. Ce contrôleur essaye de récupérer

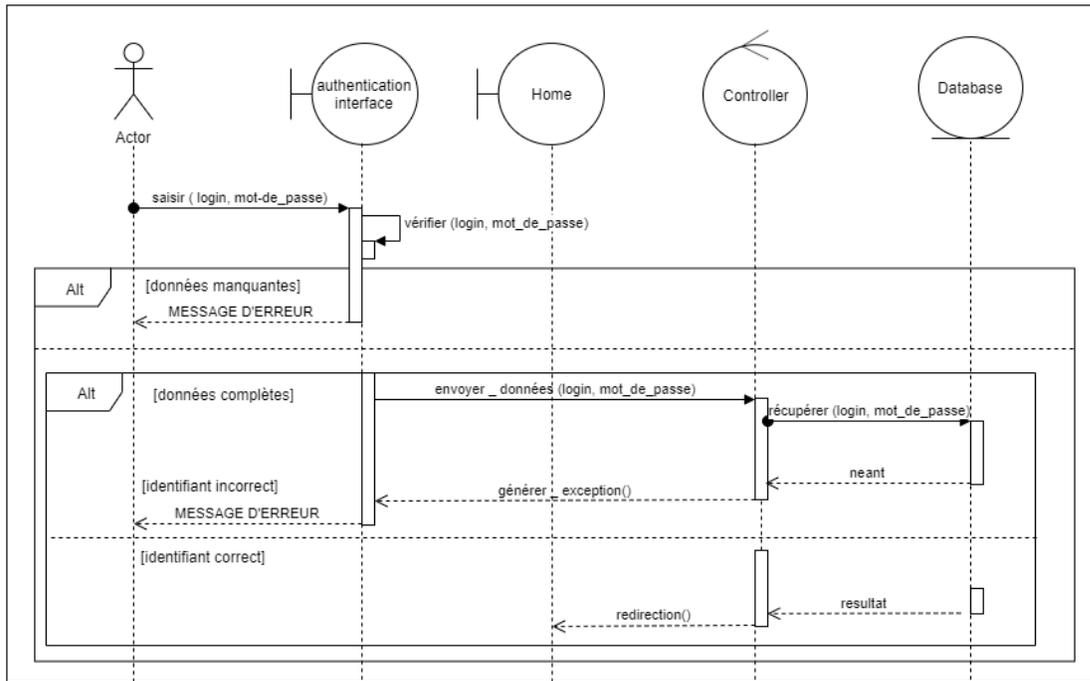


Figure 5.8: diagramme de séquence pour l'authentification des acteurs

toutes les informations du profil correspondant aux données saisies à partir de la base de données MazCalc. Si aucun profil correspond a n'est trouvé alors un message d'erreur indiquant « information incorrecte » est affiché. Si un profil est trouvé, un résultat contenant les informations de ce profile seront retourne au contrôleur. Ces informations incluent les rôles de l'acteur en question. Le contrôleur ouvre une connexion à cet acteur et fait une redirection vers la page d'accueil du MazCalc pour qu'il puisse accéder aux différentes fonctionnalités du système en respectant ces droits d'accès.

Si l'acteur connecté est un enseignant (5.9), il peut accéder à l'interface de gestion et la création des modèles. L'enseignant commence par saisir les données du modèle tel que le nom de l'objet d'étude et sa description. S'il a oublié de saisir le nom, un message d'erreur sera affiché. Sinon ces données seront envoyées à un contrôleur de modèle afin de vérifier la cohérence de ces données : si elles ne

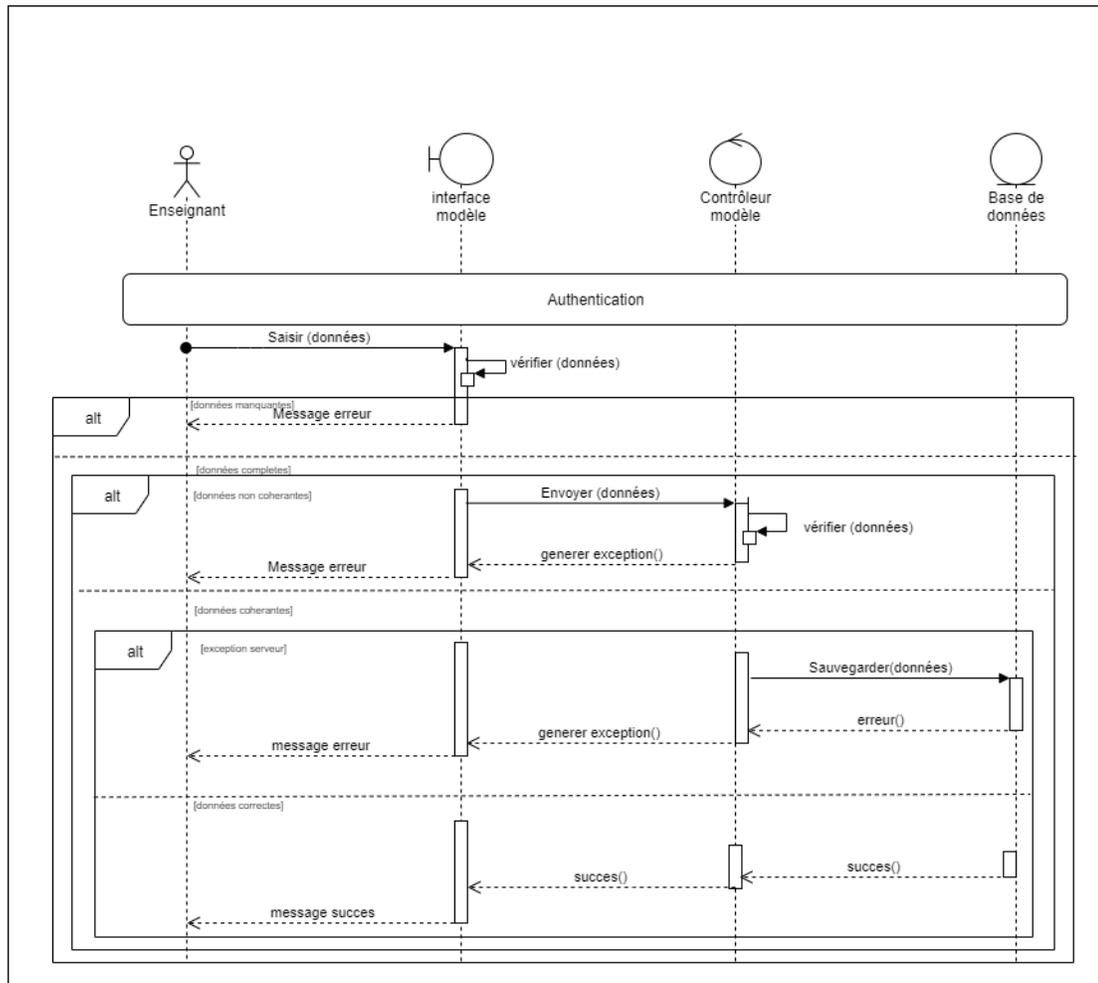


Figure 5.9: Fixation d'un objet d'étude par un enseignant

sont pas bonnes un message d'erreur est affiché au concepteur. Sinon le contrôleur l'envoi à la base de données de MazCalc. Si le contrôleur arrive à sauvegarder. Ces données dans la base de données, un message de succès sera envoyé. Sinon il envoie un message d'erreur indiquant un problème dans le serveur.

Par la suite, un concepteur peut continuer à spécifier les paramètres du modèle de l'objet d'étude, ou un autre concepteur peut exécuter la tâche. Le concepteur lance présentement l'interface d'ajout des paramètres pour saisir les

informations des paramètres un par un. Pour ce type de tâches, différents scénarios doivent être pris en considération (figure 5.10).

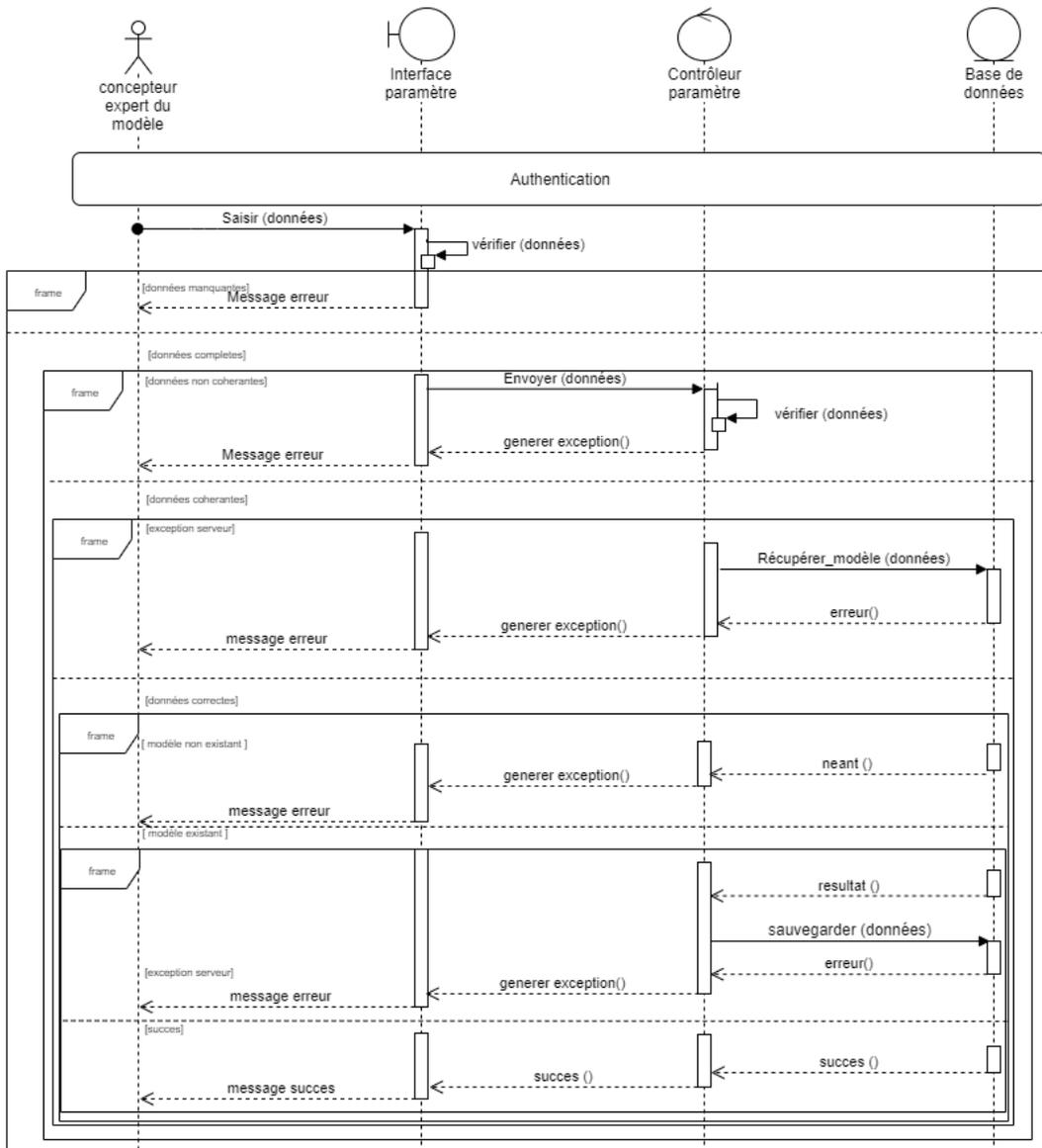


Figure 5.10: Ajouter un paramètre à un objet d'étude

Si les données sont manquantes, le concepteur reçoit un message d'erreur. Sinon le contrôleur des paramètres les récupère et essaye de les vérifier : si elles

sont incohérentes, le contrôleur génère une exception pour l'afficher par la suite au concepteur. Sinon, il récupère les données du modèle correspondant aux données saisies pour les attacher ensemble et faire la sauvegarde dans la base de données. S'il n'y a pas de problème à aucun niveau, un message de succès d'ajout est envoyé. Si le modèle n'a pas été trouvé à cause d'une suppression faite par un autre acteur connecté en parallèle que le concepteur présent, une exception est générée et envoyée au concepteur. Si au niveau de la récupération du modèle ou au niveau de l'attache du paramètre à son modèle correspondant, le serveur génère une exception. Un message d'erreur sera envoyé au concepteur indiquant la cause de cette exception.

Pour la tâche de saisi de toutes valeurs d'un paramètre d'un objet d'études donné, le même séquençement de réalisation que celui de l'ajout d'un paramètre est a considéré avec une légère modification Au lieu de récupérer le modèle en question seulement pour l'attacher à chaque valeur, il faut récupérer le modèle et le paramètre en question pour les attacher tous ensembles.

Au niveau où le modèle de l'objet d'étude , ses paramètres et leurs valeurs est définit, le Spécialiste de contexte peut accéder a l'interface de définition de l'objet d'études dans son contexte en s'identifiant. Il commence a choisir le modèle de l'objet d'étude en question (2, le contrôleur reçoit ce choix via l'interface contexte, et essaye de récupérer les données de ce modèle (ces paramètres et leurs liste de valeurs) pour en créer un formulaire de saisi. c'est le serveur génère une exception, elle sera affichée sur l'interface contexte. Si le modèle et ses données n'existent plus, le spécialiste reçoit un message d'erreur détaillée. Dans le cas où la base de données a retourné les données nécessaires, le contrôleur produit un formulaire contenant le nom du modèles, les paramètres a sélectionner et les valeurs à choisir et l'affiche sur l'interface contexte. a la réception, le spécialiste du contexte saisi les données correspondantes à son contexte et les envoie au contrôleur via l'interface.

Le contrôleur contexte sauvegarde le contexte et ses données dans la base de données et envoie un message de succès au spécialiste. En cas de problème, une exception est générée et le spécialiste reçoit un message d'erreur. 5.11.

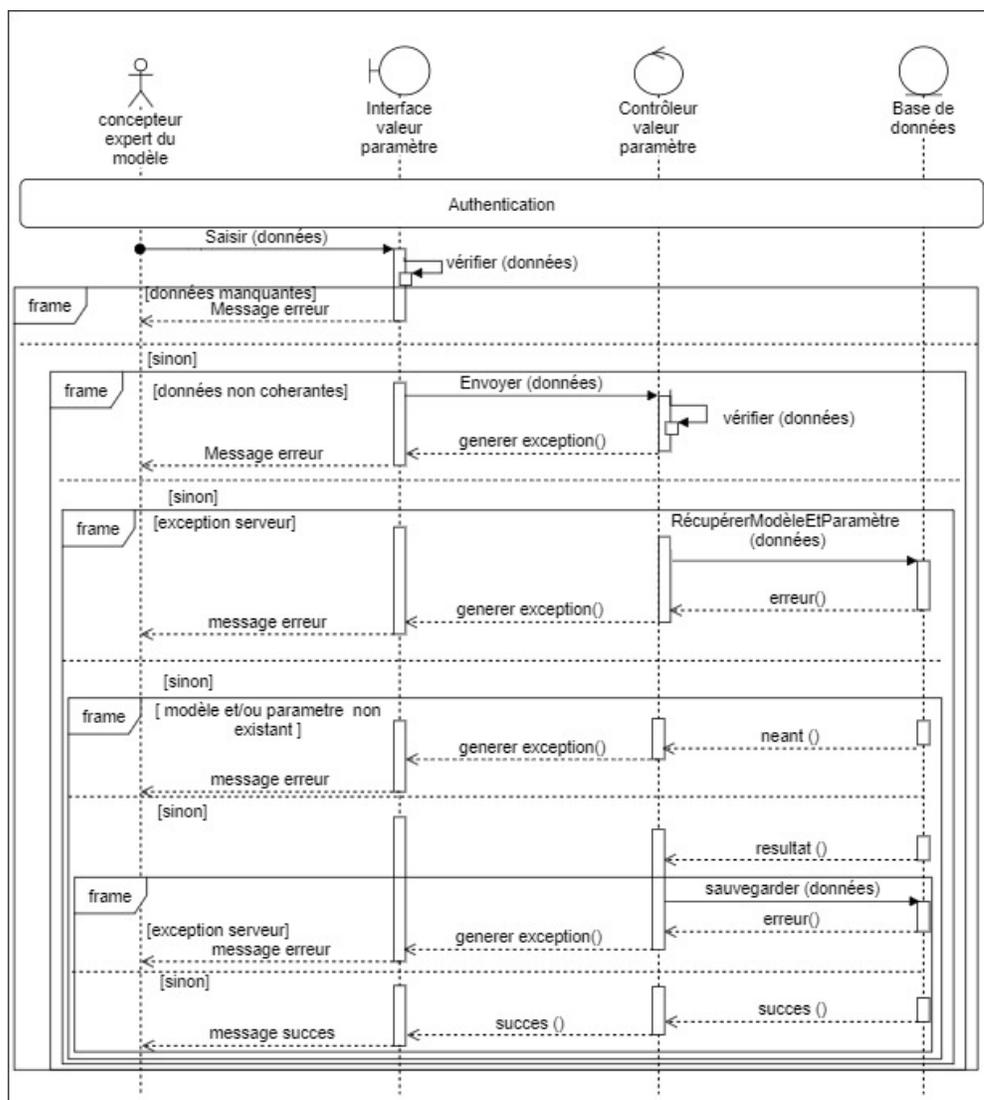


Figure 5.11: Ajouter une valeur à un paramètre d'un objet d'étude

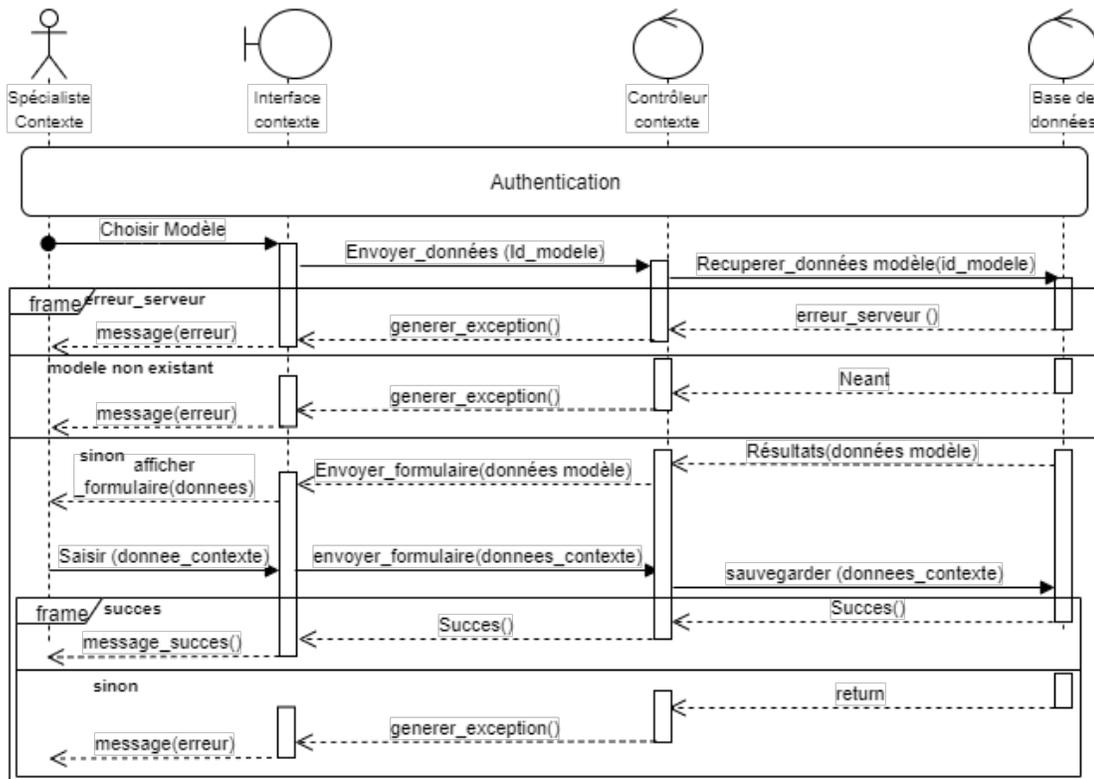


Figure 5.12: Diagramme de séquence : définition du contexte d'un objet d'étude

## 5.6 Diagramme de classes et Schéma de Base de données

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre elles. Ce diagramme fait abstraction des aspects temporels et dynamiques. La figure 5.13 représente Le diagramme de classes qui donne une idée globale sur le schéma futur de la base de données de notre système. Sur ce diagramme nous avons 11 premières classes du MazCalc. Ces classes seront les tables de notre base de données où les informations contextuelles, les données liées au modèle d'étude, ainsi que les règles de calcul d'écart de contextes seront enregistrées. Les liens entre les classes représentent la dépendance entre-elles, et peuvent générer d'autre

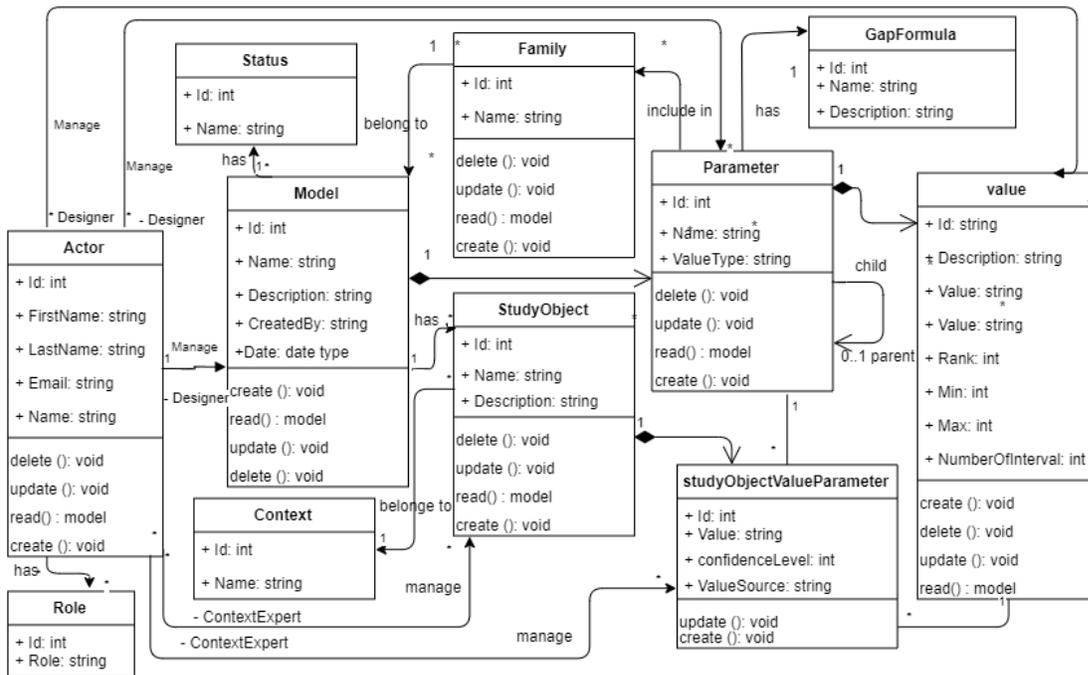


Figure 5.13: Diagramme de classes

tables s'il y a des dépendances multiples entre deux classes ou plus. C'est pour cette raison que le nombre de tables dans la base de données réelles sur la figure 5.14 est supérieur au nombre de classe figurant sur le diagramme de classe 5.13.

## 5.7 Architecture du systèmes MazCalc

Dans la figure 5.15 nous présentons les différent composants en interaction de notre solution. Le système offre deux interfaces de communication : Un site Web qui permet aux utilisateurs directs de MazCalc d'exécuter différentes opérations fonctionnelles telles que la gestion des modèles, paramètres, de contextes. Une API qui permet d'autres serveurs ou applications (PartnerApp1 et PartnerApp2 par exemple) de clients d'interagir avec notre système. Ces applications auront besoin d'avoir accès aux différentes fonctions de MazCalc autre que les modules d'ad-

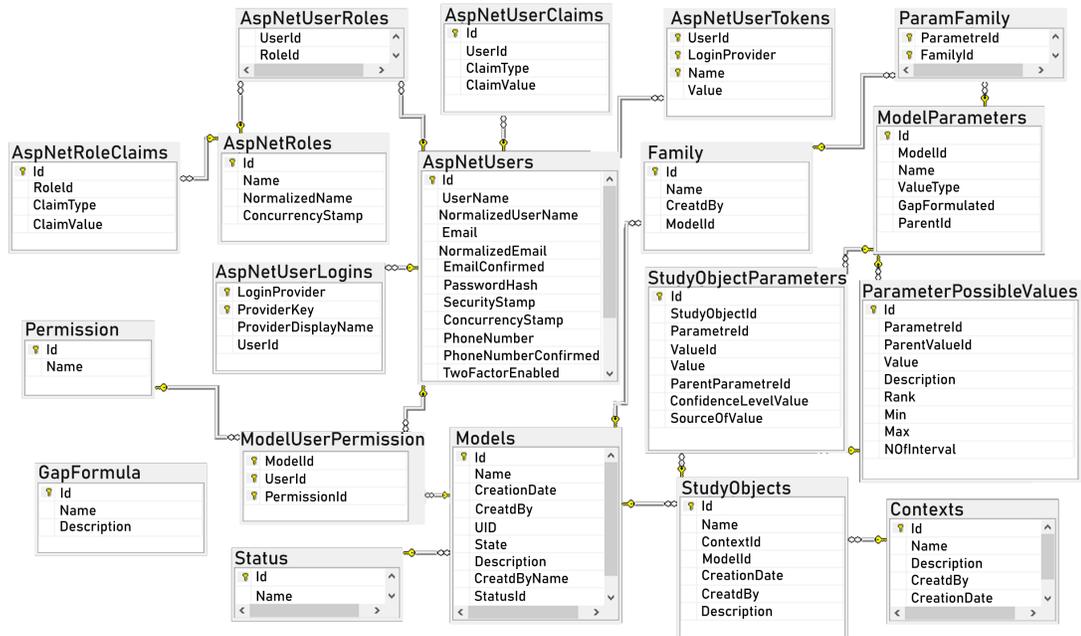


Figure 5.14: Schéma de la Base de données MazCalc

ministrations du système lui-même (telle que la sécurité et le contrôle d'accès). Ces applications appartiennent aux clients du projet qui ont besoins d'importer les fonctions de MazCalc dans leurs systèmes. Pour éviter de dupliquer le code, toutes les fonctionnalités communes sont implémentées au niveau de l'API. L'application web communique ainsi avec l'API pour assurer tous ces besoins fonctionnels. MazCalc web peut être considéré comme le premier client de l'API. Il offre une interface graphique qui permet d'afficher les résultats de cette dernière dans un format convenable pour les utilisateur humains. La sécurité est assurée par le serveur STS (Security Token Service) qui implémente les standards OAuth (Open Authorization) et OIDC (OpenId Connect). Toutes les informations de configuration ainsi que les données utilisées des utilisateurs sont persistées dans une base de données SQL qui sera déployée sur son propre serveur.

Il faudrait noter, que tous les composants MazCalc peuvent exister sur le

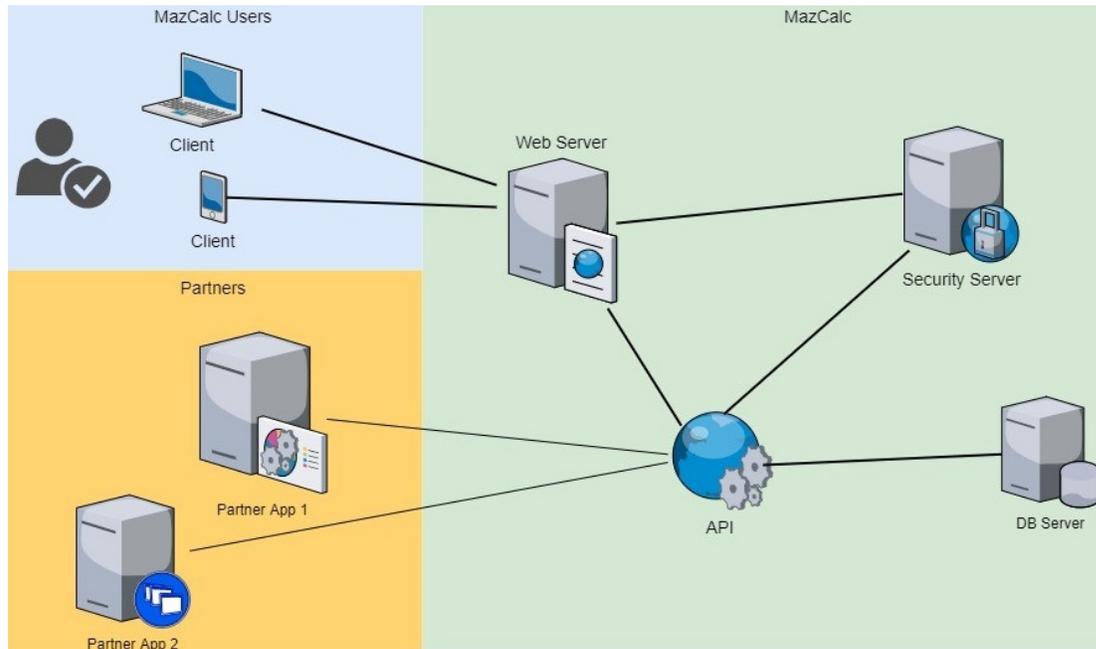


Figure 5.15: Architecture globale du MazCalc

même serveur physique et il est acceptable d'avoir un seul projet ASP.NET Core MVC qui combine le web, l'API et le serveur de sécurité. Cependant, nous avons opter pour développer chacun dans son propre projet web ce qui permet plus de flexibilité quant aux différents scénarios de déploiement. D'un cote, le code source de MazCalc sera rendu publique en tant que open-source. Si, pour raison ou une autre, nous avons besoin d'importer les fonctions de l'API dans un autre système directement dans le projet de partenaire, il suffira dans ce cas d'avoir le projet. D'un autre côté, si le nombre d'utilisateurs humains croît d'un facteur différent de celui des applications clients nous aurons un nombre d'intégrations (avec des applications clients) beaucoup plus grand que les utilisateurs. D'où pour répondre aux besoin de capacité, il suffit de faire évoluer l'API seulement (soit en utilisant un serveur plus puissant, soit en utilisant une redondance avec un load balancer pour aboutir à plusieurs copies de l'API). Aussi, la maintenance sera plus faciles car nous n'aurons pas un seul point de défaillance.

Après avoir présenté l'étude conceptuelle de notre système, nous entamons la phase de développement. Les choix techniques et les spécifications d'intégration de différentes technologies dans MazCalc seront détaillés dans le chapitre suivant.



## CHAPITRE VI

### CHOIX TECHNIQUES ET IMPLÉMENTATION

Après une présentation détaillée de l'architecture et des éléments de conception de MazCalc, la question qui se pose est comment implémenter la solution pour garantir un système qui répond aux exigences fonctionnelles tout en respectant les contraintes non fonctionnelles. Tout ce qui est exposé dans le présent chapitre est purement technique et vise à fournir les détails d'implémentation du système. En premier lieu, nous allons décrire les techniques et les outils que nous avons utilisés pour développer notre système de modélisation de contexte MazCalc. Nous expliquons ensuite comment tous les modules étudiés sont rassemblés. Les techniques utilisées seront brièvement expliquées ainsi que la motivation derrière l'utilisation de chacune. Ensuite, nous exposerons les principales règles de calcul que nous avons développées pour le fonctionnement du système et pour finir nous présenterons les résultats de notre implémentation.

#### 6.1 Aspects Techniques

##### 6.1.1 Paradigme

Tout au long du projet, nous avons essayé d'implémenter les meilleurs pratiques comme convenu dans la discipline de génie logiciel. La plupart des principes ou méthodes mentionnées ci-après sont répandues dans l'industrie et sont recon-

nue par la communauté de développeurs et d'architectes de solutions logicielles. Il existe différents paradigmes de programmation chacun ayant ses propres caractéristiques et ses utilisations. Nous pouvons assimiler le paradigme au style de programmation pour écrire le code et l'exécution des instruction (les états du système). Le choix de paradigme dépend généralement du type du problème, ses données et son domaine d'application. Nous citons quelques paradigmes de programmation, tel que la Programmation impérative ou déclarative, la programmation fonctionnelle (généralement utilisé dans les problèmes ayant une représentation mathématique), la programmation Orientée objet (système à ensemble d'objets, modules et interaction humaine) et la programmation logique (domaine de preuve automatique et intelligence artificielle). Les paradigmes impératif et fonctionnel ne s'appliquant pas à notre problème, nous avons opté pour le paradigme orienté objet car il est le seul qui permet de représenter les différents états et qui présente une interface homme-machine de façon simple. De plus, l'orienté objet et le langage de modélisation UML sont très complices. UML représente tout élément comme un objet (entité, évènement, module, graphe ...). Ainsi, il serait plus simple de traduire les diagrammes UML en classes qu'en programmes déclaratifs.

Vue la nature du projet qui nécessite le développement d'une application Web, et une composante de type "interface de programmation d'application" (Application Programming Interface : API), nous avons opté pour La plateforme ASP.NET Core pour le codage de notre Système MazCalc. C'est l'une des versions qui étend la multi-plateforme .NET. Ce choix est fondé principalement sur des contraintes techniques ainsi que des contraintes de maintenance et de sécurité. La simplicité d'utilisation de cet environnement fait partie des critères de notre choix. Nous discuterons nos choix dans les sections qui suivent.

### 6.1.2 .NET, ASP.NET et ASP.NET Core

**.NET** est une multi-plateforme de développement gratuite et open source et permet l'utilisation de multiples bibliothèques, plusieurs langages de programmation et éditeurs pour développer de nombreux types d'applications tel que des sites Web, des services, les applications bureau ou mobiles, de jeu et IoT. **.NET** contient plus de 90000 packages dont NuGet est son gestionnaire.

Pour le développement de notre application MazCalc, nous utilisons **ASP.NET** de la plateforme **.NET**. ASP.NET étend la plate-forme de développement **.NET** avec des outils et des bibliothèques spécifiques pour la création d'applications Web. Elle permet de créer différents types d'applications Web, notamment des pages Web, des API avec de "transfert d'état représentatif" (Representational State Transfer : REST), des micro-services. Elle offre aussi des librairies très utiles pour l'utilisation de différents patrons de conception Web comme le patron Modèle-Vue-Contrôleur (Model-View-Controller : MVC). Nous détaillerons ce patron de conception dans une prochaine section. ASP.NET comporte un système d'authentification qui inclut des bibliothèques d'authentifications, une base de données et modèles de pages pour gérer les connexions, entre-autres l'authentification externe (par Google, Twitter et facebook) et l'authentification multi-facteurs pour une meilleure sécurité d'accès. ASP.NET offre aussi une syntaxe de modèle de page, appelé Razor, pour créer des pages Web dynamiques en utilisant C #, HTML, CSS et JavaScript. Du côté serveur, où réside la logique métier et l'accès aux données, C # est utilisé comme le langage de programmation de l'application. C'est un langage de programmation simple et de type orienté objet nous allons l'utiliser pour le développement de notre application MazCalc.

Vue que le projet TEEC est encore en évolution et CAITS qui intégrera le système MazCalc est en phase d'étude, nous avons cherché à faciliter la tâche des

développeurs futurs de CAITS et à éliminer les contraintes techniques possibles relatives aux environnements de programmation. La version ASP.NET Core est le seul candidat choisi parmi toutes les versions ASP.NET de Microsoft, Vue qu'elle supporte de multiples plateformes, c'est-à-dire, les applications ASP.NET Core peuvent s'exécuter sur n'importe quel système opérationnel comme Windows, Linux et MacOS. ASP.NET élimine l'obligation de créer différentes applications pour différentes plates-formes à l'aide de différents cadres. Une application Web développée à l'aide de ASP.NET Core peut être hébergée sur plusieurs plates-formes avec n'importe quel serveur Web tel que Apache, IIS,etc. En d'autres termes IIS n'est plus l'unique option disponible. Il faut noter aussi que ASP.Core 3 est la version la plus récente existante à ce jour et c'est la version que nous utilisons pour le développement de notre application MazCalc.

### 6.1.3 Environnements logiciels : Visual Studio et MS SQLServer

Avec le choix de la plateforme .NET de Microsoft, l'éditeur de code VS (Visual Studio) est une décision facile. La version "Community" est totalement fonctionnelle et comporte toutes les fonctions de base nécessaire de l'éditeur, et d'autres fonctions avancées comme un assistant pour le code, un compilateur, une exécution des tests automatisée...etc. nous avons opté aussi pour un produit MS SQLServer de Microsoft pour son niveau d'intégration avec .NET et Visual studio. Le choix de SQLServer est basé aussi sur le type de besoin de manipulations de nos données : Notre système aura à persister des données tel que les données des utilisateurs, les informations contextuelles et paramètres des modèles...etc de façon permanente, Donc il est impératif d'utiliser une base de données de type SQL, Alors SQLServer est le bon choix pour notre application. La base de données de MazCalc présentée à la section 5.6 du chapitre 5 sera gérée par SQLServer via l'outil "SQL Server Management Studio". Il faudrait noter que nous avons utilisé

Entity Framework (EF) comme un outil de correspondance (Mapping) Objet-relationnel (ORM) qui nous permet d'abstraire la base de données et d'écrire nos requête SQL en C# .

#### 6.1.4 Entity Framework

ORM ou encore "correspondance objet-relationnel"(en anglais object-relational mapping) est une technique de programmation informatique qui crée l'image d'une base de données orientée objets avec un langage de programmation à partir d'une base de données relationnelle en définissant des correspondances entre la base de données originale et les objets générés. Microsoft offre l'ORM Entity Framework (EF) qui nous permet de créer une couche d'accès aux données (DAL) liée à notre base de données relationnelle et donne la possibilité de créer un schéma conceptuel composé d'entités (qui sont des modèles de données en entités EDM) codées en C# et qui permet de manipuler les tables de notre base de données SQL avec le langage C# grâce à l'outil "Linq To Entities". Le but étant de fournir une couche d'abstraction et d'accéder aux données sans avoir besoin accéder directement à la base de données ou utiliser des requêtes SQL. EF a une multitude de connecteurs pour des base de données autres que MSSQL. Tout autre choix de SGBD (MySQL, PostgreSQL, Sqlite, etc.) peut se faire sans changer le code du service d'accès aux données. La seule modification serait dans la configuration de EF au démarrage du site web et de l'API.

#### 6.1.5 Gestion de version du code (GIT)

Le code source de toute application subit un nombre élevé de changements tout au long de cycle de développement de la solution. Une solution pour garder l'historique des modifications est essentiel pour la gestion du cycle de développe-

ment. Aussi pour les équipes avec plusieurs développeurs sur le même projet, il est nécessaire de comparer et fusionner le code de tous les participants tout en leur permettant de travailler chacun en isolation. Un système de contrôle de version est la solution pour ces besoins. Pour notre projet, le besoin principal est de garder

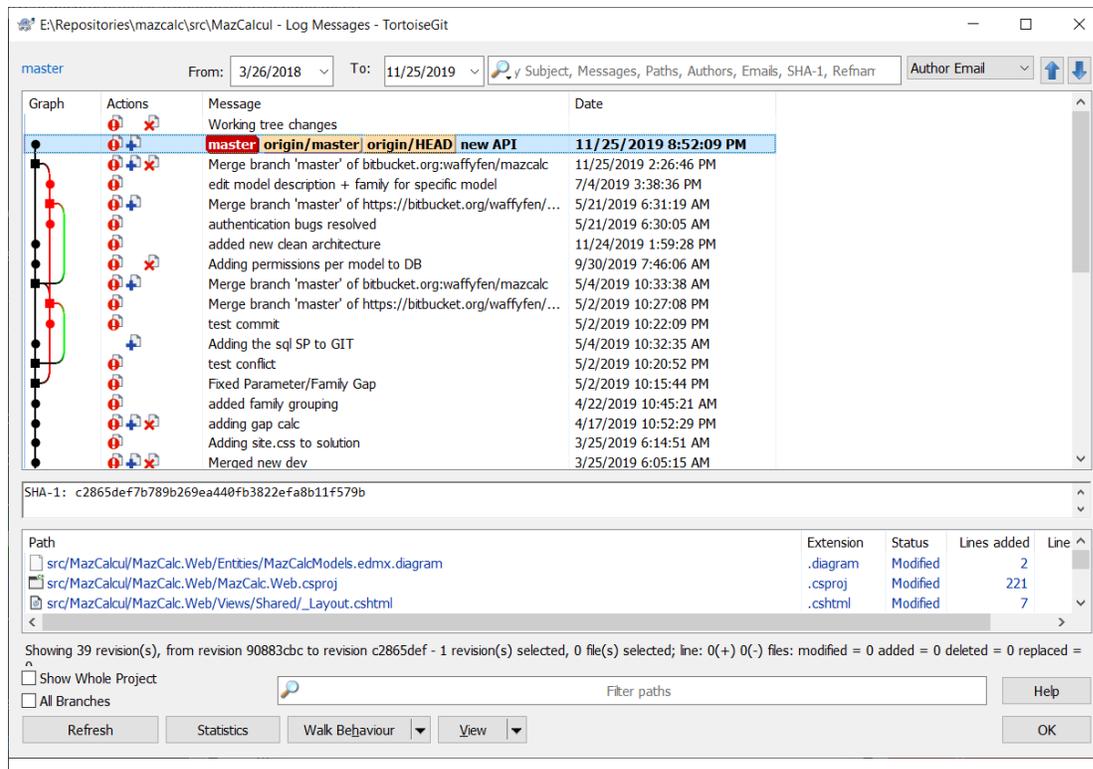


Figure 6.1: Historique des changements du code MazCalc

l'historique du code et avoir la possibilité de restaurer n'importe quelle version au besoin.

A cet effet, nous distinguons deux protocoles majeure SVN et GIT. Tous les deux peuvent couvrir notre besoin. GIT a été choisi car c'est le standard dans l'industrie ces temps-ci. Le dépôt du code peut être dans un réseau d'entreprise ou sur un serveur cloud sur internet comme Github, Gitlab ou Bitbucket. Le client GIT est TortoiseGIT qui permet d'avoir une interface graphique comme le montre

la figure 6.1 pour interagir avec le dépôt sans utiliser l'invite de commande. Le choix est purement personnel. Tout au long de développement nous avons utilisé Bitbucket. Nous envisageons à rendre le code disponible sur Github prochainement.

#### 6.1.6 DDD : Domain Driven Design

DDD est l'acronyme de Domain Driven Design. Ce n'est ni un Framework, ni une méthodologie, mais plutôt une approche décrite dans l'ouvrage du même nom d'Eric Evans (Evans et Design, 2003). Un de ses objectifs est de définir une vision et un langage partagés par toutes les personnes impliquées dans la construction d'une application, afin de mieux en appréhender la complexité. Nous avons appliqué le DDD au meilleur de nos connaissances pour rester fidèle au champ lexical du domaine du contexte afin de définir les entités, les états et les processus impliqués dans le système.

#### 6.1.7 TDD : Test Driven Development

TDD est une technique de conception agile qui devrait aboutir à la production d'un code propre fonctionnel. L'idée principale est d'écrire le minimum de code nécessaire sous forme de test isolé avant de le transformer (par le biais de refactoring) vers un code de production. Plus nous avançons dans le projet, plus nous écrivons de tests et nous obtenons comme un ensemble de tests comme résultat dont nous pouvons automatiser l'exécution. Ceci est important pour s'assurer qu'une nouvelle fonctionnalité ajoutée ne créera pas de bug. Une simple exécution de tous les tests permet de valider le tout d'un seul coup. Au niveau de MazCalc, nous avons développé des tests unitaires des fonctionnalités les plus importantes. Dans le cas où nous voulons tester un processus en entier, nous ne parlons plus

de test unitaire mais de test d'intégration. Généralement dans l'industrie de déve-

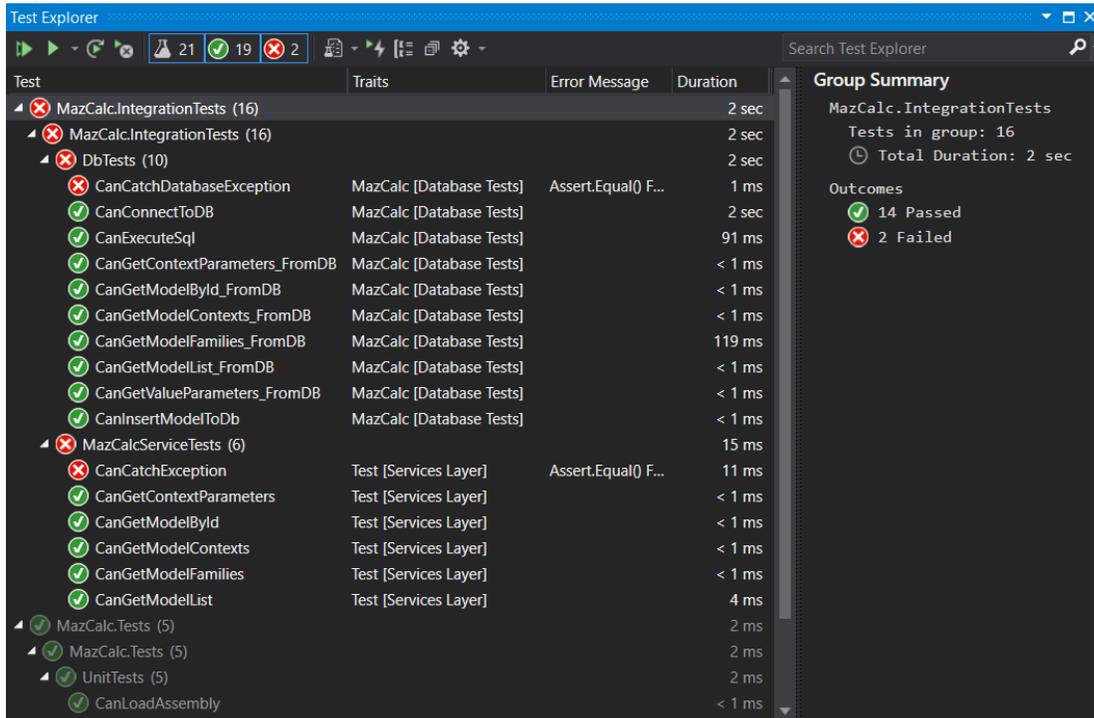


Figure 6.2: Résultats de test automatique dans MazCalc

loppement de logiciel, l'exécution des tests est faite par le serveur de compilation lorsqu'un processus d'intégration continue et de déploiement continue est appliquée (CI/CD pipeline) . Pour ce faire nous avons utilisé la suite d'outil de Visual studio ainsi que la bibliothèque xUNIT. Les tests sont exécutés manuellement après la compilation. La figure 6.2 montre un exemple d'exécution d'une série de tests automatiquement : tout le processus commence avec un simple bouton de menu ou une commande et en quelques secondes, nous avons les test et leurs résultats sur une interface graphique.

### 6.1.8 Patrons de conception

Un autre aspect du développement de logiciel est l'utilisation de patrons de conception. En effet, toute application vise à résoudre un problème donné. Plusieurs problèmes sont récurrents et assez connus et ils ont été déjà résolus. Les patrons de conceptions présentent des solutions prouvées et élégantes pour des problèmes connus. La liste de patron de conception est assez longue et continue à grandir de jour en jour face aux problèmes nouveaux résidents dans le cloud ou les systèmes distribués par exemple. Dans notre système, nous avons utilisé le patron de conception MVC. MVC est une plateforme de développement Web basé sur le modèle de conception "Modèle/vue/contrôleur". Ce patron prône le découpage des responsabilités entre les données métier (le Modèle), le traitement et le routage des éléments d'une application (le contrôleur), et la présentation de l'information (la vue). Ce qui rend c'est trois composants indépendants. Cette indépendance permet d'isoler les problèmes d'interface utilisateur dans View, les problèmes de comportement dans un contrôleur et les problèmes de stockage et de communication dans un modèle.

### 6.1.9 SOLID

C'est un ensemble de principes de développement de logiciel qui aide organiser le code dans des classes et méthodes afin de produire un code : lisible et facile à comprendre, Facile à maintenir, Réutilisable. SOLID est un acronyme pour les principes suivants :

- SRP (Single Responsibility Principle) : Une classe doit avoir une seule fonctionnalité. Cela ne signifie pas que chaque classe doit avoir une seule méthode mais peut avoir plusieurs méthodes qui sont directement liées et assurent une seule fonction générale de la classe.

- OCP (Open-Close Principle) : Les entités (classes, méthodes, modules ...) doivent permettre l'extension et interdire la modification. Ceci est généralement atteint en utilisant l'héritage et la redéfinition de la portion de code en question.
- LSP (Liskov Substitution Principle) : Les objets ne doivent pas altérer l'exactitude de l'application lorsqu'ils sont remplacés par leurs sous-types (classes héritées).
- ISP (Interface Segregation Principle) : Ce principe est exclusif aux interfaces et stipule que le client (implémentant l'interface) ne doit pas dépendre de méthode dont il n'en a pas besoin. En pratique, nous utilisons plusieurs petites interfaces au lieu d'une seule grande interface. Par exemple, si besoin est, nous ajoutons une fonctionnalité à une classe A -implémentant une interface B- non à l'interface mère B mais en créant une autre interface C et en utilisant l'héritage multiple. La classe implémentant les deux interfaces dans ce cas aura la combinaison des deux fonctionnalités. Tout autre classe implémentant seulement B, n'aura pas à définir la fonctionnalité de C .
- DIP (Dependency Inversion Principle) : “nous devons dépendre d'abstractions et non pas d'implémentations concrètes”. Cela se traduit en pratique par le fait qu'une classe dépend d'interface au lieu d'une implémentation de cette interface. C'est ce principe qui assure le découplage dans les applications et facilite aussi les tests unitaires.

#### 6.1.10 Clean Architecture

Ces dernières années, l'intérêt quant aux patrons d'architecture de solution a augmenté. Le résultat en est l'apparition de différentes architectures telles qu'Architecture Oignon, l'Hexagonale, Screaming Architecture, DCI .... etc. Quoique ces patrons sont variés dans leur détail, leur but ultime reste le même : la séparation des préoccupations. Cela signifie que le système en question va être composé de

différents modules qui assurent la testabilité, Indépendance de la base de données et Indépendance des plates-formes. Le premier but signifie que les règles d'affaires (noyau de l'architecture) peuvent être testées en isolation, c-à-d sans avoir besoin d'interface Utilisateur ou de base de données. Le second signifie que le changement de la couche de persistance se ferait sans affecter les règles d'affaires et devrait affecter des modifications mineures. Le dernier permet d'éviter les dépendances sur les bibliothèques ou des outils externes dans le noyau de l'application. Il est ainsi facile de changer ces bibliothèques au besoin sans changer le noyau. L'auteur dans (Martin, 2017) donne le diagramme de la figure 6.3 pour décrire "l'architecture propre". Les

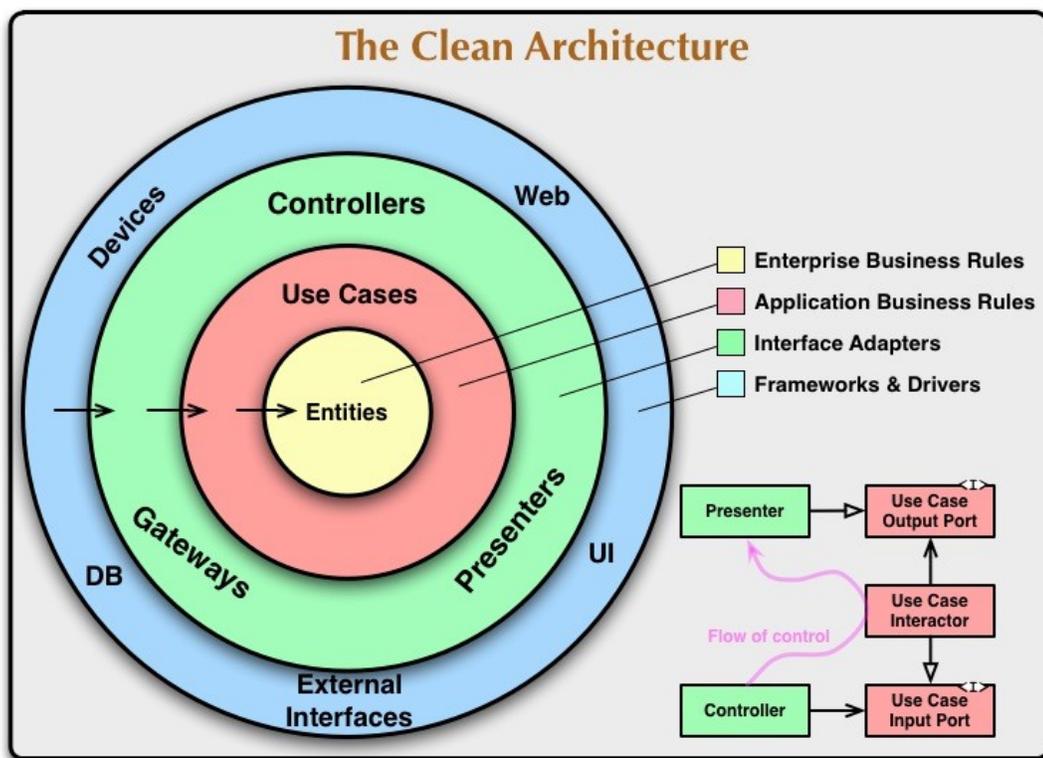


Figure 6.3: Diagramme de l'architecture propre donnée par (Martin, 2017)

cercles représentent différents modules ou couches logiques de l'application. Les flèches définissent la direction de dépendance. Donc une couche donnée dépend

seulement de ses couches internes et ne doit pas avoir aucune information sur les couches externes. La couche "Entités" englobe les entités du domaine et les règles d'affaires au niveau de l'entreprise (donc commun à toute les application). La couche "Cas d'utilisations" représentent les règles d'affaires de l'application. C'est assez similaire à la "couche affaire" dans les architectures n-tier. Cette couche dépend seulement des entités. Le changement apporte ici ne devraient pas impacter ni la couche internes ni celles externes. La couche "Adaptateurs d'interface" est le niveau responsable des transformations sur la structure de données des entités vers tout autre niveau ou service externe, comme par exemple convertir une entité du domaine vers la (ou les) structures équivalentes pour affichage à l'interface utilisateur. La couche Plate-formes et pilotes contient toutes les dépendances vers des service externe tels que la base de données, services ou API externes...etc.

Nous voyons maintenant comment nous appliquons l'architecture propre à notre solution MazCalc présenté a la figure 6.4 et son équivalence indiquée dans le tableau 6.1.

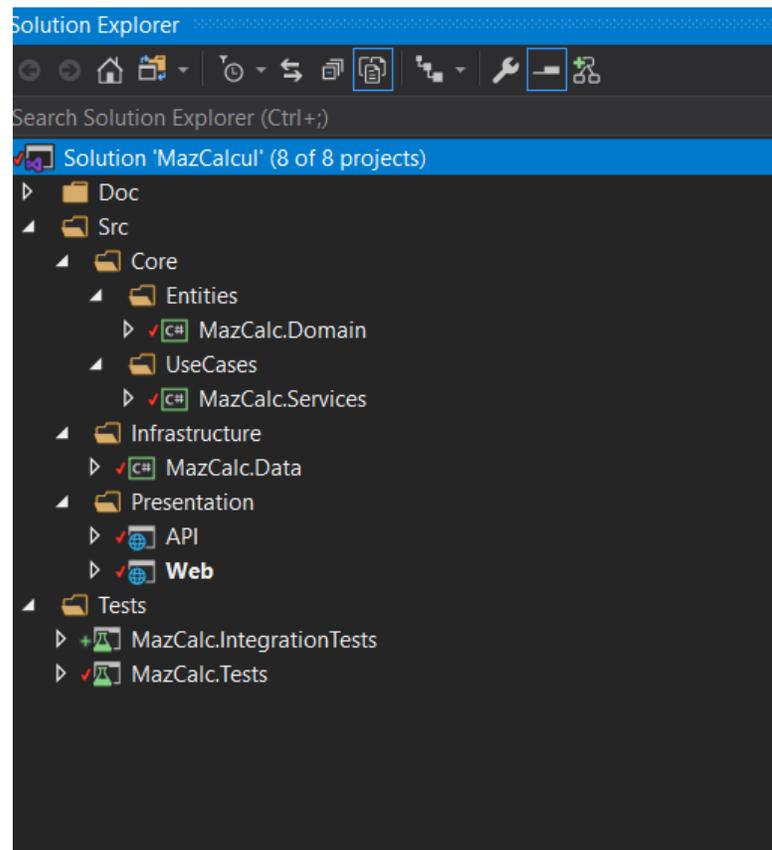


Figure 6.4: MazCalc- Clean architecture

#### 6.1.11 REST API

Comme établi dans l'étude de besoins de MazCalc, nous devons offrir la possibilité d'intégration avec d'autres applications partenaires sans changer notre code et sans connaître le code de ces systèmes que nous appelons "clients". Nous pouvons exposer les services de notre solution à travers un service Web (SOAP, RPC) ou une Web API (Application Programmable Interface). Nous avons choisi d'utiliser une API Web car nous utilisons le protocole HTTP seulement et cela rend la communication plus rapide vue que les services Web en général utilisent des phases d'établissement de communication, ou utilisent des entêtes dans les

Tableau 6.1: Tableau des projets MazCalc et leurs équivalences dans l'architecture propre

<b>Projet</b>	<b>Equivalent clean Description</b>	<b>Description</b>
MazCalc.Domain	Entités	Contient toutes les entités du domain
MazCalc.Services	Cas d'utilisation	Contient les règles d'affaire implémentées sous forme de services pour la gestion de Model, contexte, objets d'études ...etc.
MazCalc.Data	Base de données	Contient la couche persistance et l'implémentation de toute interaction avec SQL Server
API	Présentation	Contient l'API MazCalc
Web	Presentation / UI	Contient le site web MazCalc
Identity	STS	Le serveur responsable de la sécurité
MazCalc.IntegrationTests	—	Tests d'intégrations
MazCalc.Tests	—	Test Unitaire

messages (comme le protocole SOAP). Pour ce faire, nous avons suivi le style architectural de REST (REpresentational State Transfer) qui s'applique aux APIs

web (HTTP).

En gros REST, tire parti du vocabulaire de HTTP. C'est aussi basé sur la communication client-serveur. Un client envoi une demande d'information (ou d'opération) au serveur. Le serveur fournit le résultat de l'exécution sous forme de ressource (ensemble de données définissant l'état de la réponse du serveur). Les messages (requête et réponse) ont un format de l'un des types de média d'internet (Json ou XML en général). Les requêtes sont formées par une opération HTTP (GET, POST, PUT ...) avec l'adresse de la ressource demandée et le contenu des paramètres de la requêtes (soit dans l'adresse, soit dans le corps de la requête). Une API adhère à REST lorsqu'elle respecte les six contraintes suivantes :

- Orienté client-serveur (Client-server) : Le client demande une ressource par le biais d'une requête, le serveur envoie la ressource demandée dans sa réponse avec un code de statut HTTP (200 pour Ok, 4xx erreur par le client, 5xx erreur par le serveur).
- Sans état (Stateless ) : l'état du client est inconnu entre deux requêtes. Toutes les requêtes sont indépendantes les unes des autres.
- Cacheable (avec cache = mémoire) : lorsque la technique de cache est applicable, la ressource doit l'indiquer. Le cache peut se faire au niveau du client, sur le serveur ou sur les deux.
- un système de couche (Layered) : entre le client et le serveur répondant à la requête, il peut y avoir plusieurs serveurs au milieu (serveur de sécurité, cache, équilibreur de charge ...)
- Un code à la demande (optionnel) (Optional code on demand) : le client demander sous form de code qu'il va exécuter. (comme un code JS ou HTML).
- Une interface uniforme (Uniform interface) : Chaque ressource doit avoir une adresse (URI) unique avec un identifiant. Si besoin est, la ressource doit contenir

des liens (Hateoas) vers les informations reliées.

Pour implémenter notre API, nous utilisons un projet de type ASP.NET Core pour les API comme nous avons mentionné au début de ce chapitre.

#### 6.1.12 OpenApi pour documentation API

Comme notre l'API doit être consommée par d'autres applications. Nous nous devons donc de donner et une documentation détaillée pour toutes les opérations que nous offrons ainsi que la façon de l'utiliser, les options et les différents paramètres. Nous devons expliquer au développeur ou à l'architecte d'une autre application comment envoyer une requête pour une opération donnée et que doit être dans le format de la ressource retournée.

OpenApi est une spécification de description pour les APIs de type REST. Elle documente toutes les fonctions d'une API dans un fichier OpenApi sous format JSON ou YAML. Le fichier est utilisé pour générer une documentation destinée aux développeurs et généralement avec une possibilité de tester la fonctionnalité sur la page de documentation. Il peut même être utilisé pour générer le code du client et ainsi donner un exemple de code fonctionnel pour le développeur. Ceci est assuré avec des outils qui comprennent le format OpenApi. Dans le monde .NET, Swashbuckle est l'outil le plus populaire (Biehl, 2016). Il nous permet de générer une documentation sous forme de page web de notre API. Nous montrons cette documentation dans la partie publique de l'application web de MazCalc.

#### 6.1.13 Sécurité

MazCalc est composé essentiellement de deux applications qui nécessitent une attention particulière à la sécurité. Le deux requièrent la possibilité de d'au-

thentifier les utilisateurs, vérifier leurs permissions avant l'exécution de tout opération qui nécessite une autorisation particulière. Ces deux applications sont fondamentalement différentes (site web avec interface graphique , API sans interface mais un échange de données en JSON ou XML) et ont des clients différents (humains pour le web, ordinateurs ou code utilisant l'API). Pour cela, la solution classique de d'utilisation de nom d'utilisateur et de mot de passe n'est plus suffisante.

Deux standards ont émergé pour répondre aux besoins en sécurité pour les nouvelles applications : OAuth (pour l'autorisation) et OIDC (pour l'authentification). Chacun présente différents scénarios (grants ou flows). Chaque scénario vient pour résoudre le problème de sécurité pour certains types d'application et certains types de client. Il faut noter que ceci est devenu un besoin urgent car les applications modernes ne sont plus juste limitées par les frontières de l'entreprise mais nous voyons maintenant des API qui offre une intégration avec des clients ou autres. Ce type d'intégration signifie un défi additionnel pour la sécurité. Nous retrouvons maintenant des intégrations avec des systèmes de confiance totale (comme dans notre cas entre le site web et l'API), De confiance partielle ou sans confiance.

La centralisation de l'implémentation de la sécurité a son propre module (projet web "MazCalc.Identity") est un moyen qui nous permet d'avoir une seule localisation de la logique de sécurité, d'éviter la duplication de code et d'utiliser la sécurité comme service indépendant de l'application. Ceci offre la possibilité d'avoir une seule authentification entre plusieurs systèmes. Cette méthode d'implantation nous fournit des techniques standards utilisant des bibliothèques prouvées fidèles aux standards de sécurité. Nous avons basé notre serveur STS sur une bibliothèque open source "Identity Server" qui implémente "OAuth" et "OpenId Connect" à travers un ensemble de fonctionnalités autour le "jeton" (token) de

sécurité de type JWT (Json Web Token) utilisé dans les deux standards.

## 6.2 Algorithmes de calcul d'écart

L'objectif principal de MazCalc est de calculer l'écart entre deux contextes pour s'en servir, dans une phase ultérieure, pour élaborer des scénarios d'apprentissage. A cet effet, MazCalc doit doter d'un algorithme d'alignements au niveau des familles des paramètres dans les deux contextes, et des alignements au niveau des paramètres comme nous avons détaillé dans la section 5.3.3 du chapitre 5. Un autre algorithme pour coder les fonctions de calcul relativement aux règles que nous avons définies (règle de comparaison **R1**, règle d'éloignement **R2**, règle composite **R3**). Cet algorithme doit faire la reconnaissance de la règle applicable lorsqu'il reçoit un paramètre et les valeurs correspondantes des deux contextes. Le calcul de l'écart pour un paramètre composite est une tâche un peu plus complexe que les autres paramètres puis qu'il se fait sur trois étapes : la première étant de faire la condescendance entre la règle et le paramètre en question. La deuxième étant de faire le calcul de l'écart au niveau du paramètre parent. La dernière étant d'appliquer la fonction de calcul qui groupe le résultat de paramètre parent et les valeurs du paramètre fils, pour ressortir une valeur pondérée. MAzCalc, dans ce cas, doit définir un algorithme qui gère le calcul de ce paramètre différemment. Tous ces algorithmes seront codés explicitement en C#.

## 6.3 Implémentations de MazCalc

### 6.3.1 Les interfaces modèle

Dans cette partie, nous présentons les résultats d'implémentation de notre système MazCalc, dont l'analyse et la conception ont été détaillées dans le chapitre 5. La figure 6.5 représente l'interface de définition de l'objet d'étude.

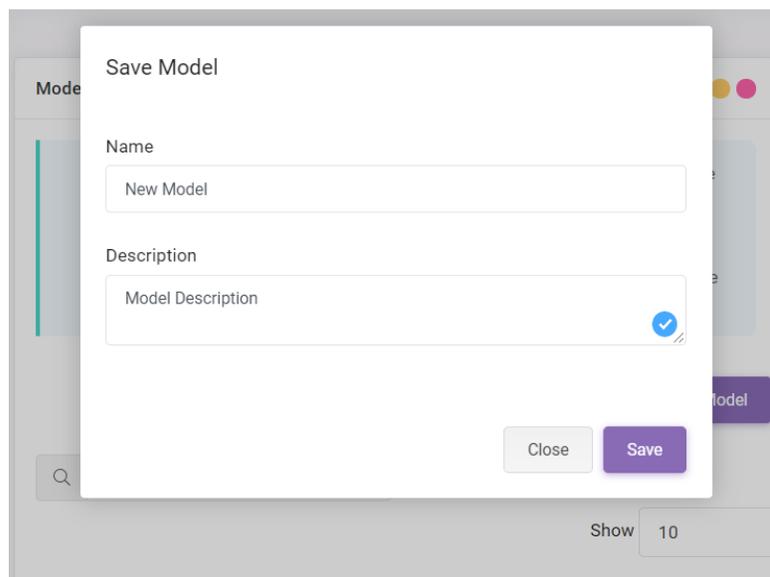


Figure 6.5: Interface de définition d'un nouveau objet d'étude

L'enseignant qui fixe le modèle, doit saisir le nom de l'objet à étudier ainsi qu'une petite description textuelle. Dans une deuxième étape, le concepteur Expert doit indiquer la liste des familles mise en jeu qui seront utilisées pour grouper les paramètres. Pour spécifier les paramètres du modèle, le concepteur doit se diriger vers l'interface de création des paramètres présentée dans la figure 6.6 : Cette interface permet d'indiquer le nom du paramètre, de préciser son type (s'il est QualitatifOrdonné, QualitatifÉnuméré ou Quantitatif) et de sélectionner la règle de calcul d'écart applicable parmi (soit comparaison, soit éloignement, soit composé) ainsi que la liste des familles auxquelles il appartient.

Selon le type du paramètre introduit, le concepteur peut se retrouver sur l'une des 4 interfaces :

- La première l'interface de la figure 6.7 est dédiée pour introduire l'intervalle

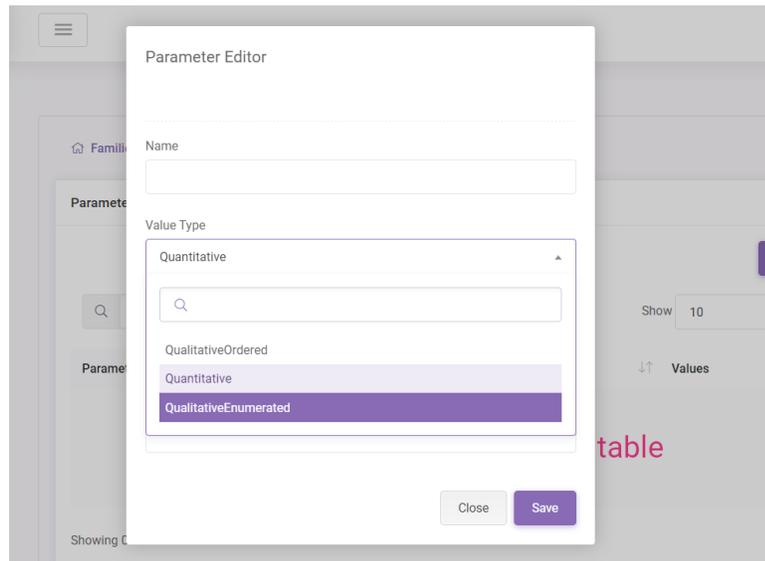


Figure 6.6: Interface de définition des paramètres de l'objet d'étude

des valeurs d'un paramètre quantitative. Il faut indiquer la valeur minimale, et la valeur maximale de cet intervalle, ainsi que le nombre de sous-intervalle possible (ceci est utile pour connaître la position de la valeur dans l'intervalle et qui servira après pour appliquer la formule de l'éloignement si elle est la règle de calcul correspondante à ce paramètre).

- La deuxième interface sur la figure 6.9 s'affiche pour un paramètre de type "Qualitatif Énuméré". Sur cette interface il faut juste énumérer toutes les valeurs possibles d'un paramètre.

- La troisième interface est celle de type du paramètre "Qualitatif Ordonné". Elle est présentée à la figure 6.10. Comme nous avons mentionné dans les chapitres précédents Pour ce type, l'ordre des valeurs est important, donc le concepteur doit introduire la valeur et son ordre dans la liste des valeurs pour bien appliquer la règle relative à un tel paramètre.

Save Model

Min

Max

Number of Interval

Close Save

Figure 6.7: Interface de définition des valeurs d'un paramètre quantitatif

Values for Parameter : Type of rock (Type = QualitativeEnumerated)

+ Add Value

Search

Show 10 entries

Value	Parent	Actions
Foliated	Metamorphic	
Nonfoliated	Metamorphic	
Volcanic	Magmatic	
Plutonic	Magmatic	
Filon	Magmatic	
Chemical	Sedimentary	

Figure 6.8: Interface de définition des valeurs d'un paramètre composé

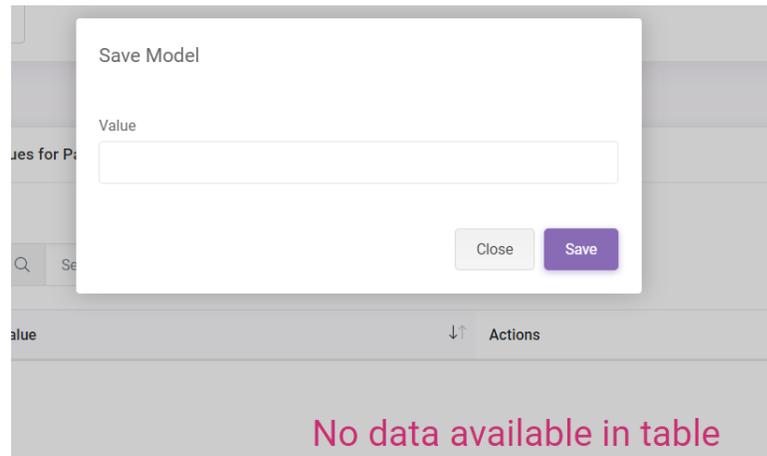
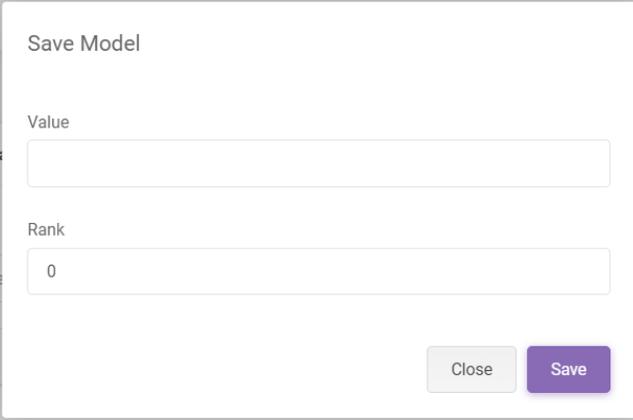


Figure 6.9: Interface de définition des valeurs énumérées d'un paramètre qualitatif

- La dernière interface indiquée sur la 6.8 est dédiée aux paramètres de type Composé, où il fallait itérer sur les valeurs du paramètre parent, et saisir à chaque fois la liste des valeurs correspondantes.

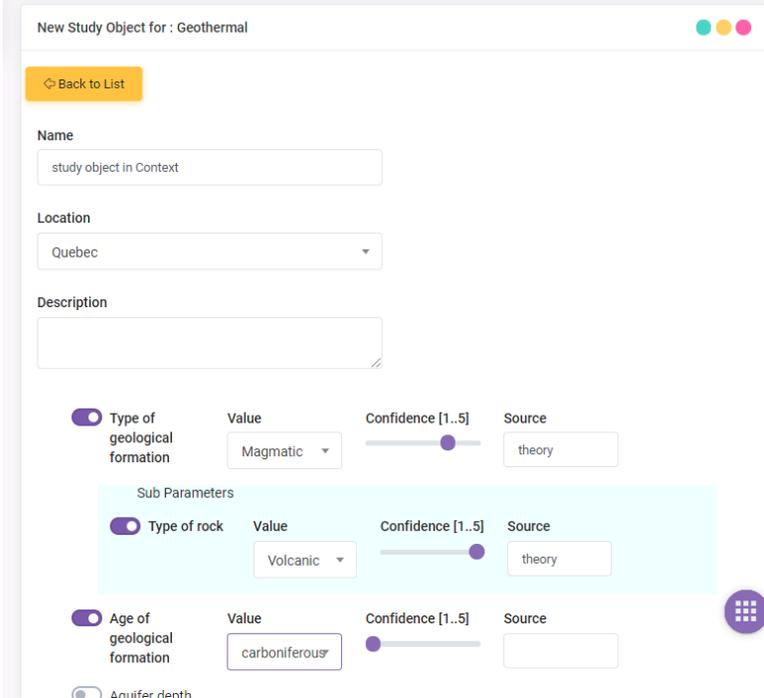
Après que le concepteur complète la modélisation de l'objet d'étude, l'expert de contexte peut alors accéder à l'interface montrée sur la figure 6.11 pour instancier la déclaration de l'objet d'étude dans son contexte. Sur cette interface la liste de tous les paramètres du modèle en question est affichée.

L'expert doit indiquer le nom de l'objet d'étude (par exemple s'il a un nom plus particulier dans un contexte), préciser le contexte d'étude et de sélectionner la liste des paramètres qu'il juge pertinents et qui entre en jeu dans ce contexte. Pour chaque paramètre sélectionné, le spécialiste doit spécifier sur cette même interface la valeur du paramètre, le degré de confiance à donner à cette valeur, ainsi que la source de la valeur. Sur cette interface la liste de tous les paramètres du modèle en question est affichée.



The image shows a 'Save Model' dialog box with two input fields. The first field is labeled 'Value' and is empty. The second field is labeled 'Rank' and contains the number '0'. At the bottom right of the dialog are two buttons: 'Close' and 'Save'. The dialog is overlaid on a background that includes a pink message: 'No data available in table'.

Figure 6.10: Interface de définition des valeurs ordonnées d'un paramètre qualitatif



The image shows a 'New Study Object for : Geothermal' form. It has a 'Back to List' button at the top left. The form contains several sections: 'Name' with a text input field containing 'study object in Context'; 'Location' with a dropdown menu set to 'Quebec'; 'Description' with a text area. Below these are three parameter sections, each with a toggle switch, a 'Value' dropdown, a 'Confidence [1..5]' slider, and a 'Source' text input. The first section is 'Type of geological formation' with 'Magmatic' selected. The second section is 'Type of rock' with 'Volcanic' selected. The third section is 'Age of geological formation' with 'carboniferous' selected. A 'Aquifer depth' section is partially visible at the bottom.

Figure 6.11: Interface d'instanciation d'un objet d'étude dans un contexte

Figure 6.12: Interface de choix de contextes de Géothermie à comparer

Si au moins deux contextes d'un même objet d'étude sont définis, il est possible d'en choisir deux pour les comparer comme indiquée sur la figure 6.12.

#### 6.4 Tests et Résultats

Nous avons présenté les différents services qu'offre MazCalc. Il nous reste à valider le bon fonctionnement du système. Est-il capable de bien modéliser un objet d'étude avec tous ces types de paramètres et leurs marges de valeurs possibles ? L'instanciation des contextes est-elle possible à partir des modèles créés préalablement ? MazCalc permet-il de calculer l'écart entre deux contextes relatifs à un même objet d'études ? Arrive-il à appliquer pour chaque paramètre la règle de calcul qui lui correspond correctement ? Les rôles pour chaque acteur sont-ils bien distribués et respectés ? C'est une liste de questions qui se posent et qui nécessitent un test d'évaluation du système afin de pouvoir en répondre.

L'évaluation du système à ce stade est une tâche importante pour pouvoir s'en servir dans l'avenir dans un environnement d'apprentissage réel. Il est impératif alors d'appliquer des exemples des objets d'études qui ont été bien modélisés

et validés par des experts du domaine d'étude. C'est objet d'étude doivent être aussi étudiés dans au moins deux contextes géographiques différents pour pouvoir calculer l'écart entre eux. Trouver plusieurs exemples avec ces critères est vraiment une tâche extrêmement difficile. Vu que les experts ne sont pas toujours à disposition. Pour cette raison nous utiliserons un seul exemple d'objet d'étude, qui est la Géothermie. La Géothermie un scénario d'apprentissage parmi les itérations DBR du projet TEEC et elle est le domaine d'expertise d'un membre de l'équipe, Claire Anjou. Nous avons Réussi à tester MazCalc grâce au modèle qu'elle a définie. Elle a nous a fourni deux contextes, malgré qu'ils ne soient pas complets mais valides, de la géothermie : le premier est la géothermie au Québec et le second est la géothermie à Guadeloupe.

#### 6.4.1 La géothermie

La géothermie Comme Claire l'a défini est « du grec « géo » qui signifie « Terre » et « thermós » se traduisant par « la chaleur », désigne à la fois l'étude des phénomènes thermiques internes au globe terrestre, et l'exploitation par l'homme de cette chaleur, à des fins énergétiques» (Anjou, 2018a).

Le modèle que nous utilisons est défini par 13 paramètres de différents types. Chacun ayant une liste des valeurs possible comme le montre la figure 6.13. Les titre des colonnes sur cette figure indique le nom du paramètre et dans chaque colonne la liste de ses valeurs possibles, suivis du nombre de ces possibilités. Nous distinguons 11 paramètres de type « Qualitatifs Ordonnés » qui sont : « Type de géothermie », « Âge de formation », « Potentiel géothermique », « Zone géodynamique », « Réseau de failles », « Énergie produite », « type de roche et type de relief », « Température de l'eau », « type de relief », « Profondeur de l'aquifère

Type_formation	Age_formation	type_géothermie	flux_chaleur	Profondeur de	Ph	Temp_eau	Type de relief
Sédimentaire	Quaternaire	pas de géothermie	0-40	pas d'aquifère	<2,0	< 10°C	Plain
Métamorphique	Tertiaire	très basse énergie	40-60	<200m	2,0-4,0	10-30°C	Plateau
Magmatique	Crétacé	basse énergie	60-85	200-400m	4,0-6,0	30-90°C	Mountain
	Jurassique	énergie profonde	85-120	400-600m	6,0-8,0	90-150°C	Hill
	Trias	haute énergie	120-180	600-1000m	8,0-10,0	> 150°C	Fjord
	Permien		180-240	1000-2000m	10,0-12,0		Canyon
	Carbonifère		240-360	2000-3000m	>12,0		
	Dévonien			3000-5000m	pas d'information		
	Silurien			>5000m			
	Ordovicien						
	Cambrien		min= 0				
	Précambrien		Max =500				
<b>3</b>	<b>12</b>	<b>5</b>	<b>9</b>	<b>9</b>	<b>8</b>	<b>5</b>	<b>6</b>
Potentiel_géoth	réseau de failles ?	Zone géodynamique	énergie produite	Type de roche			
faible	oui	Frontière de plaque	Chaleur	Sédimentaire	Magmatique	Métamorphique	
moyen	non	Intra plaque	Electricité	Chimique	Volcanique	Foliée	
important	pas d'importance			Biologique	Filonienne	Non Foliée	
<b>3</b>	<b>3</b>	<b>2</b>	<b>2</b>	Détritique	Plutonique		

Figure 6.13: Liste des paramètres de l'objet d'étude Géothermie donnée par l'expert

» et le « PH ». Le paramètre « Flux de la chaleur » est de type « Quantitatif », tandis que le paramètre « Type de formation » est de type « Qualitatif Énuméré ».

La règle de calcul de l'éloignement s'applique presque pour tous les paramètres, sauf pour « type de formation » et « Type de roche ». Pour le paramètre « Type de formation » nous appliquons la règle de comparaison alors que pour le paramètre « Type de roche », il est de type composite, alors la règle applicable est celle du « composite ». Son paramètre parent est « Type de formation ». Chaque valeur du paramètre « Type de formation » possède une liste de valeur possible pour le paramètre « Type de roche ». Par exemple, si le type de formation est « Magmatique », le type de roche peut être soit « Volcanique », soit « Filonienne » soit « Plutonique ». Ces paramètres sont groupés par domaine. Les 5 familles du domaine sont mentionnées sur figure 6.14.

Durant l'itération d'étude de la géothermie dans deux géolocalisations différentes, les valeurs des paramètres en été collectées par des étudiants de Guadeloupe et de

Paramètres du contexte pour le domaine de la Géothermie	
FAMILLE	PARAMETRES
Géologie générale	Type de formation géologique
	Âge de la formation géologique
	Type de roche
Géophysique	Type de géothermie exploitée
	Valeur du flux de chaleur
	Potentiel géothermique
	Température de l'eau
	Profondeur du toit de l'aquifère
Type d'énergie produite	
Hydrogéologie	Profondeur de l'aquifère
	fluid pH
Geomorphologique	Topographie
Structurale	Zone géodynamique
	Présence d'un réseau de failles

Figure 6.14: Groupement par famille des paramètres de la Géothermie

Québec et ont été validées par l'expert de la géothermie. Nous remarquons sur la figure 6.15 que les valeurs dans les deux contextes géographiques sont différentes. Par exemple le type de formation au Québec est « Métamorphique » alors qu'à Guadeloupe est « Magmatique ». L'électricité est le type d'énergie produite par la géothermie au Guadeloupe, alors qu'au Québec c'est la chaleur qui est produite.

Ce décalage présent entre les deux contextes les rend un bon exemple pour tester le système MazCalc et valider le calcul d'écart en résultat.

#### 6.4.2 Création du modèle

La première étape de test consiste à fixer l'objet d'étude en question par un enseignant : Donc un enseignant se connecte et crée le nom de l'objet d'étude, ainsi qu'une petite description comme il a été indiqué dans la figure 6.5 de la section 6.3 du chapitre 5.

Paramètres du contexte pour le domaine de la Géothermie			
FAMILLE	PARAMETRES	Guadeloupe	Québec
Géologie générale	Type de formation géologique	Magmatique	Métamorphique
	Âge de la formation géologique	Quaternaire	Précambrien
	Type de roche	Volcanique	Foliée
Géophysique	Type de géothermie exploitée	haute énergie	très basse énergie
	Valeur du flux de chaleur	250	50
	Potentiel géothermique	important	faible
	Température de l'eau	> 150°C	< 10°C
	Profondeur du toit de l'aquifère	1000-2000m	<200m
	Type d'énergie produite	Electricité	Chaleur
Hydrogéologie	Profondeur de l'aquifère	1000-2000m	<200m
	fluid pH	10,0-12,0	8,0-10,0
Geomorphologique	Topographie	Montagne	coline
Structurale	Zone géodynamique	Frontière de plaque	Intra plaque
	Présence d'un réseau de failles	oui	pas d'importance

Figure 6.15: La géothermie telle que définit au Québec et a Guadeloupe

Model List

- If you cannot see a model, please contact your administrator and make sure you have the proper permissions assigned to the model in question.
- To create a new model, you need to have one of the following roles : Administrator or Teacher
- To manage a model, you need to have : either the role Administrator or have access to the model with permission=ModelDesigner

+ Create New Model

Search

Show 10 entries

Model	Status	Created By	Creation Date	Action
Geothermal	Valid	Teacher Maz	2019-10-01	

Showing 1 to 1 of 1 entries

Figure 6.16: interface présentant l'objet d'étude fixé par un enseignant

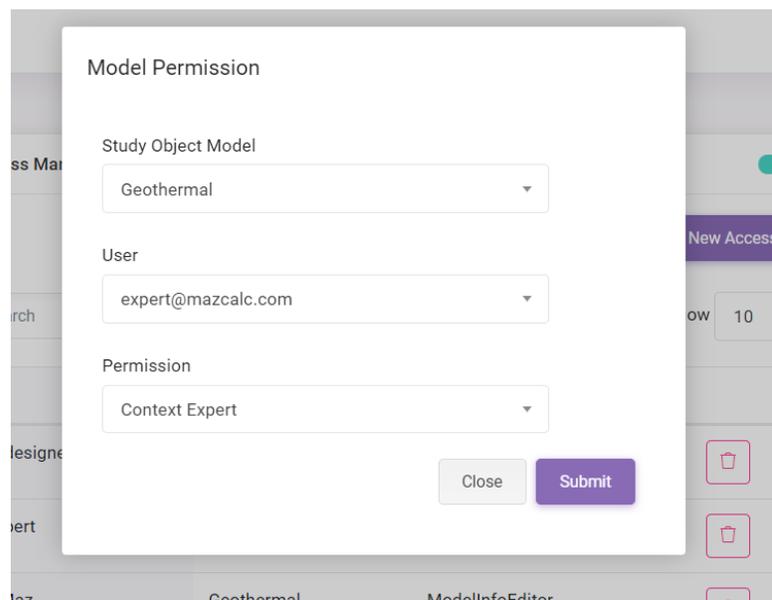


Figure 6.17: Interface d'attribution des rôles à un acteur

La figure 6.16 montre que le modèle a été bien créé. La date de création a été bien rajoutée au modèle géothermie ainsi que le nom de l'enseignant qui l'a créé. A partir de cette page, l'enseignant peut consulter les détails du modèle créé en cliquant sur le bouton « Details », mais aucun paramètre n'est encore spécifié.

Un acteur doit avoir le rôle d'un « concepteur Expert » pour cet objet d'étude pour qu'il puisse continuer à modéliser la Géothermie. Le gestionnaire des rôles, appelé Administrateur, doit se connecter pour attribuer les rôles spécifiques à chaque acteur.

L'administrateur choisi l'acteur en jeu et lui donne un rôle relatif à la géothermie via l'interface de la figure 6.17.

. L'acteur "designer Expert" a le rôle du concepteur, et l'acteur " Context

Model Access Management

+ Create New Access Rule

Search

Show 10 entries

User	Model	Permission	
designer designer	Geothermal	ModelDesigner	
expert expert	Geothermal	ContextExpert	
Teacher Maz	Geothermal	ModelInfoEditor	

Showing 1 to 3 of 3 entries

Figure 6.18: Interface de la liste des acteurs et leurs rôles

"Specialist" a le rôle d'un spécialiste du domaine dans un contexte comme indique sur la figure 6.18.

Le concepteur commence tout d'abord par définir les familles des paramètres via l'interface 6.19, par la suite il peut définir les paramètres de la géothermie, et spécifier les listes des valeurs correspondantes via l'interface de la figure 5.10 de la section 6.3.

Sur la figure 6.20 nous voyons que les 13 paramètres ont été bien rajoutés ainsi que leurs types respectifs. Le bouton "Edit" permet de consulter la liste des valeurs possibles pour un paramètre donné. La figure 6.21 affiche la liste des valeurs pour le paramètre « Flux de la chaleur », qui est de type « Quantitatif », où les bornes de son intervalle Min Max ont été spécifiées correctement.

Sur la figure 6.22 la liste des valeurs affichées est celle du paramètre « Type de formation » qui est de type « Qualitatif Énuméré ».

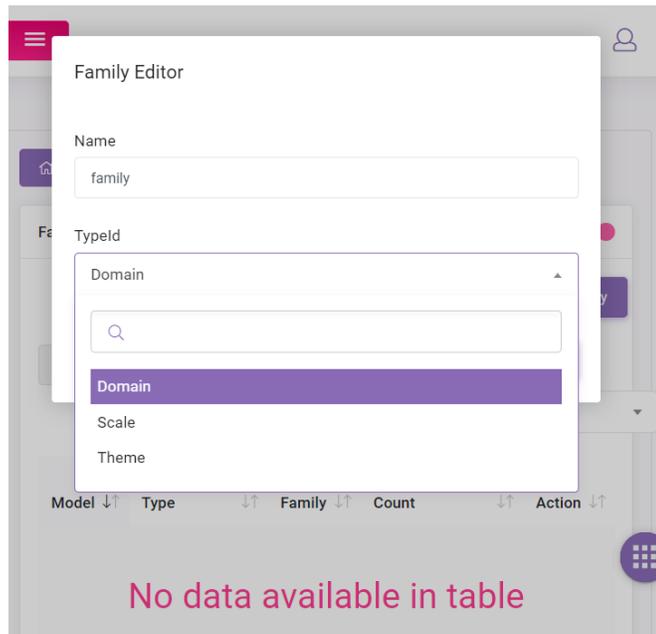


Figure 6.19: Interface de définition des familles de groupement des paramètres  
Géothermie

Le paramètre « Type de roche » est un paramètre composite comme nous avons mentionné, donc nous voyons que la valeur d'un parent correspond à une liste des valeurs pour le fils comme sur la figure 6.23 .

Pour le reste des paramètres, l'ordre est important et leur ajout se fait via l'interface indiquée la section 6.3 sur la figure 6.10. Le paramètre « Âge de formation » affiché sur la figure 6.24 est un exemple de ce type de paramètre.

#### 6.4.3 Instanciation des contextes

A la fin de la modélisation de la géothermie nous pouvons consulter les détails du modèle via l'interface de la figure 6.25.

[+ Create New Parameter](#)

Search

Show  entries

Parameter	Parent	Type	Gap Formula	Values	Action
Age of geological formation		QualitativeOrdered	Distance	12	%
Aquifer depth		QualitativeOrdered	Distance	9	%
Fluid PH		QualitativeOrdered	Distance	8	%
Geodynamics		QualitativeOrdered	Distance	2	%
Geothermal potential		QualitativeOrdered	Distance	3	%
Presence of a network of faults		QualitativeOrdered	Distance	3	%
Product of the geothermal		QualitativeOrdered	Distance	2	%
Thermal flow value		Quantitative	Distance	1	

Figure 6.20: Interface de liste des paramètres de l'objet d'étude Géothermie

Values for Parameter : Thermal flow value (Type = Quantitative)

Search

Show  entries

Value	Number of Interval	Actions
Min - Max : 0 - 500	7	

Showing 1 to 1 of 1 entries

< 1 >

Figure 6.21: Interface de la liste des valeurs du paramètre « Flux de la chaleur » de type quantitatif

Values for Parameter : Type of geological formation (Type = QualitativeEnumerated)

+ Add Value

Search

Show 10 entries

Value	Actions
Metamorphic	
Magmatic	
Sedimentary	

Showing 1 to 1 of 1 entries

< 1 >



Figure 6.22: Interface de la liste des valeurs du paramètre « Type de formation » de type qualitatif énuméré

Values for Parameter : Type of rock (Type = QualitativeEnumerated)

+ Add Value

Search

Show 10 entries

Value	Parent	Actions
Foliated	Metamorphic	
Nonfoliated	Metamorphic	
Volcanic	Magmatic	
Plutonic	Magmatic	
Filon	Magmatic	
Chemical	Sedimentary	



Figure 6.23: Interface de la liste des valeurs du paramètre « Type de roche » de type Composé

Values for Parameter : Age of geological formation (Type = QualitativeOrdered) ● ● ●

[+ Add Value](#)

Search

Show  entries

Value	Rank	Actions
carboniferous	7	
Quaternary	1	
Tertiary	2	
cretaceous	3	
Jurassic	4	
Triassic	5	

Figure 6.24: La liste des valeurs du paramètre « Âge de formation » de type qualitatif Ordonné

Parameters Details			
Grouped by : Domain			
Family	Parameter	Value Type	Gap Formula
General geology	Type of geological formation	QualitativeEnumerated	Difference
	Age of geological formation	QualitativeOrdered	Distance
	Type of rock	QualitativeEnumerated	Composed
Geophysics	Aquifer depth	QualitativeOrdered	Distance
	Type of geothermal	QualitativeOrdered	Distance
	Geothermal potential	QualitativeOrdered	Distance
	Thermal flow value	Quantitative	Distance
	Product of the geothermal	QualitativeOrdered	Distance
Hydrogeology	water temperature	QualitativeOrdered	Distance
	Aquifer depth	QualitativeOrdered	Distance

Figure 6.25: Interface de Détails du modèle Géothermie et le groupement de ses paramètres par famille

Si tout est correct, le Concepteur doit valider le modèle pour que ce dernier peut être instancié par la suite. A ce niveau, l'acteur ayant le rôle Spécialiste de la géothermie dans un contexte donné peut se connecter et instancier le modèle Géothermie dans son contexte via l'interface indiquée sur la figure 6.11 de la section 6.3. La figure 6.26 montre la liste des contextes de la géothermie instanciés qui sont au Québec et à Guadeloupe.

Les contextes de test sont maintenant prêts. Il nous reste que calculer l'écart entre eux via l'interface de la figure 6.11 mentionnée à la section 6.3. L'écart est affiché sur deux parties :

- La première partie est sous la forme d'un tableau présenté la la figure 6.27.

Study Objects for : Geothermal

+ Create New Study Object

Search

Show 10 entries

Context Location ↓↑	Study Object	Description ↓↑	Parameters ↓↑	Action ↓↑
Guadeloupe	Guadeloupe Géothermal energy		12	
Quebec	Quebec Geothermal energy		12	
Quebec	study obj Géothermal		4	
Quebec	styd pbject test		0	

Figure 6.26: Interface de La liste des contextes distancées de l'objet d'étude Géothermie

Il résume les contextes Québec et Guadeloupe, leurs paramètres pertinents, leurs valeurs, ainsi que le résultat du calcul au niveau de chaque paramètre et au niveau de chaque famille. En remarque sur ce tableau que les valeurs des contextes ont été introduite correctement comme il a été donné par l'expert de la géothermie Claire Anjou. Pour chaque résultat trouvé d'un paramètre, un commentaire indiquant le degré de l'écart entre les deux contextes au niveau de ce paramètre est prescrit. Les résultats d'écart trouvés par MazCalc sont identiques aux résultats donnés par l'expert de géothermie comme le montre la figure 6.28.

- La deuxième partie est sous une forme graphique présentée à la figure 6.29, qui illustre la surface de l'écart entre la géothermie en Guadeloupe et la géothermie au Québec. Ce graphique est donné selon les valeurs d'écart de chaque paramètre.

MazCalc est évalué par l'utilisation d'un exemple d'un objet d'étude réel.

Family	Parameter	Guadeloupe Géothermal energy	Quebec Geothermal energy	Gap	Description	Family Gap
General geology	Age of geological formation	Quaternary	Permian	1	Average Gap Between Parameters	1
	Type of geological formation	Magmatic	Metamorphic	1	Very Important Gap Between Parameters	
	Type of rock	Volcanic	Foliated	1	Very Important Gap Between Parameters	
Geomorphology	Topography	Mountain	Hill	0.2	Slight Gap Between Parameters	0.2
Geophysics	Aquifer depth	1000 - 2000 m	< 200m	0.5	Average Gap Between Parameters	0.84
	Geothermal potential	High Potential	weak Potential	1	Very Important Gap Between Parameters	
	Product of the geothermal	Electricity	Heat	1	Very Important Gap Between Parameters	
	Thermal flow value	250	50	0.83	No Gap Between Parameters	
	Type of geothermal	High energy	very low energy	0.75	Large Gap Between Parameters	
	water temperature	> 150° C	<10° C	1	Very Important Gap Between Parameters	
Hydrogeology	Aquifer depth	1000 - 2000 m	< 200m	0.5	Average Gap Between Parameters	0.32
	Fluid PH	10.0 - 12.0	8.0 - 10.0	0.14	Slight Gap Between Parameters	

Figure 6.27: Résultats numériques de calcul d'écart entre le contexte Québec et le contexte Guadeloupe

Geothermal energy				
Parameters	Context 1 : Guadeloupe	Context 2 : Quebec	Gap	Family gap
Type of geological formation	Magmatic	Metamorphic	1.00	1.00
Age for the geological formation	Quaternary	Precambrian	1.00	
Type of rock	Volcanic	Foliated	1.00	0.85
Type of geothermal	high temperature geothermal	very low temperature	0.75	
Thermal flow value	250	50	0.83	
Geothermal potential	high	weak	1.00	
Hydrothermal water temperature	> 150°C	< 10°C	1.00	
Top of the aquifer depth	1000-2000m	<200m	0.50	
Product of the geothermal	Electricity	Heat	1.00	0.32
Aquifer depth	1000-2000m	<200m	0.50	
fluid pH	10,0-12,0	8,0-10,0	0.14	0.20
Topography	Mountain	Hill	0.20	
Geodynamics	Border of a plate	Intra plate	1.00	1.00
Presence of faults network	yes	no importance	1.00	

Figure 6.28: Le Calcul d'écart tel que donné par l'expert de Géothermie Clair Anjou

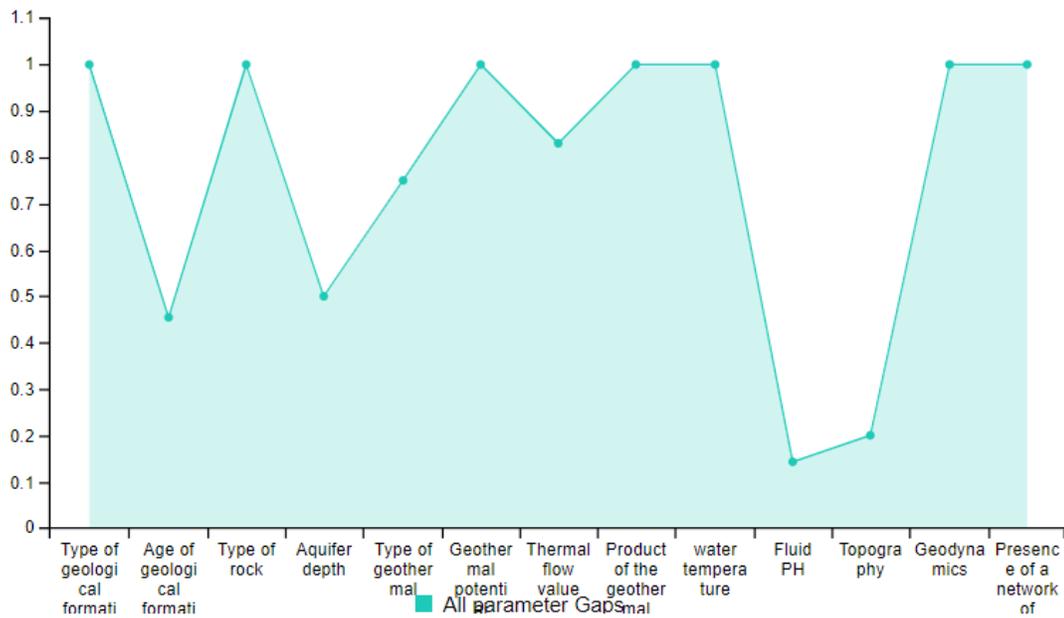


Figure 6.29: Illustration graphique des résultats de calcul d'écart entre le contexte Québec et le contexte Guadeloupe

Cette première évaluation était positive et elle nous a confirmé que toutes les fonctionnalités attendues par MazCalc sont présentes. La prochaine étape serait de faire une évaluation sur une base plus pratique après son intégration dans le système tutoriel intelligent CAITS. CAITS est en phase d'étude et il n'est pas encore implémenté. Il sera un projet futur de recherche.

## CONCLUSION

Ce mémoire présente la conception et le développement d'un système de modélisation de contexte MazCalc après une étude approfondie de la notion de contexte et de son usage en contexte éducatif. Ce Système permet de calculer l'écart entre deux contextes modélisés et fait ressortir les paramètres les plus pertinents qui permettent de créer des effets de contextes chez les apprenants durant une session d'apprentissage collaboratif inter-contexte.

MazCalc est constitué de 4 composants. Un module « modèles » où le squelette d'un objet d'étude est défini. Le module « contexte » où se fait l'instanciation des contextes à partir d'un modèle. Cette instanciation permet de déterminer les paramètres pertinents pour un contexte donnée est leurs valeurs associées. Un module « Calcul d'écart » qui s'occupe de calculer l'écart entre deux contextes relatifs à un même objet d'étude suivant des règles. Ces règles de calcul sont gérées aussi par ce module. Tous les services de ces composants sont accessibles via des interfaces Web, ou via une API.

Grâce à sa conception, qui est le résultat des études menées par des experts du contexte, MazCalc possède trois points forts : 1) C'est un outil générique qui permet non seulement une prise en charge de n'importe quel objet d'étude, mais également, prend en charge d'autre type de contexte. Dans notre étude nous sommes intéressés par l'étude du contexte géographique, mais MazCalc peut s'adapter pour d'autres catégories de contexte comme le contexte de temps. Dans ce cas le modèle peut être un objet d'étude dans un contexte donné, et son instanciation se fera selon la date de collecte de données par exemple. Le calcul d'écart

permettrait donc d'étudier l'évolution du contexte au cours du temps et identifier les paramètres statiques et les paramètres dynamiques pertinents du contexte. 2) MazCalc peut être utile dans d'autre domaine d'étude où l'étude de contexte, sa modélisation et son effet sont des éléments importants. Dans notre cas, MazCalc est un outil clé pour étudier l'effet de contexte dans l'apprentissage des sciences. 3) MacCalc peut fonctionner (grâce à sa version API) comme une entité autonome sur son propre site web, ou peut être intègre dans d'autres systèmes qui veulent se doter d'une sensibilité au contexte. CAITS est un exemple pertinent d'une telle utilisation.

L'évaluation préliminaire du systèmes MazCalc montre que les objectifs visés dans notre étude ont été atteints. Cependant, MazCalc est loin d'être parfait. D'autres perspectives de travail futur pourraient être envisagés. Durant notre étude, nous avons ignoré le caractère dynamique du contexte bien que nous gardons une trace sur la date de la création des contextes. Nous avons vu dans la littérature que le contexte change dans le temps, donc les informations contextuelles changent aussi. Ce qui fait que les valeurs des paramètres collectées pour les utiliser dans des scénarios pédagogiques peuvent changer même durant la phase d'apprentissage. Ceci rend les valeurs d'étude incorrectes, menant à de faux résultats et fausses conclusions. Pour éviter ce problème, il est nécessaire de mettre à jour ces valeurs en gardant une sauvegarde de toutes les anciennes valeurs de chaque paramètre d'un contexte. Nous envisageons aussi d'améliorer les commentaires sur les résultats de calcul d'écart qui sont donnés selon le résultat trouvé de l'écart. Les commentaires de calcul peuvent êtres donnés en fonction des paramètres du contexte et leurs valeurs pour expliquer davantage les raisons pour lesquelles ces écarts sont apparus. Un autre travail futur pouvant améliorer MazCalc consisterait à y intégrer un système de spécification automatique des paramètres de l'objet d'étude. L'étude authentique des objets d'étude nécessite toujours la

participation des experts du domaine, ce qui n'est pas toujours possible comme nous l'avons vu dans la phase de test de MazCalc : Nous n'avons pas pu trouver assez de modèles d'objets d'études et des contextes complets validés par des experts pour tester notre outil. Pour combler ce problème, le système de spécification automatique serait une bonne solution complémentaire. Ce système peut utiliser des techniques d'apprentissage machine pour extraire les paramètres d'un objet d'étude et leurs valeurs à partir des ouvrages, des articles, des images, etc.

Ce projet était une innovation dans l'apprentissage et a permis de réunir l'étude de contexte et ses effets avec la technologie pour donner naissance à MazCalc, le système de modélisation de contexte et calculateur d'écart de contexte. Ce système est générique grâce à sa conception robuste qui est le résultat des efforts de l'équipe multidisciplinaire du projet. Deux versions de MAzCalc ont été développées au sein de ma recherche. La première étant une version web capable de fonctionner indépendamment de tout environnement (<http://www.maz.farzilla.com>). La seconde est une version API qui a été testée dans la version web en lui fournissant ces services. La prochaine étape de ce projet sera de développer le système tutoriel intelligent CAITS qui intégrera l'API MazCalc qui lui fournira les informations contextuelles nécessaires pour construire des scénarios pédagogiques sensibles au contexte afin de mieux guider les apprenants dans leurs apprentissages.



## LISTE DES ABRÉVIATIONS, SIGLES ET ACROYMES

**ALESS** : Active LEarning Support System

**ANR** : Agence Nationale de la Recherche

**API** : Application Programmable Interface

**ASP** : Active Server Pages

**CAITS** : Context-Aware Tutoring System

**CAMLearn** : Context-Aware Mobile Learning

**CI/CD** : Continuous Integration/Continuous Deployment

**CRREF** : Centre de Recherches et de Ressources en Education et Formation

**CSS** : Cascading Style Sheets

**CSTM** : Context-Sensitive tutor Model

**CSLM** : Context-Sensitive Learner Model

**DAL** : Data Access Layer

**DBR** : designed-Based Research

**DCI** : Data Context Interaction

**DDD** : Domain Driven Design

**DIP** : Dependency Inversion Principle

**EF** : Entity Framework

**FTP** : File Transfer Protocol

**GDAC** : Gestion, Diffusion et Acquisition de Connaissances

**GPS** : Global Positioning System

**HMM** : Hidden Markov model

**HTML** : Hyper Text Markup Language

**HTTP** : HyperText Transfer Protocol

**HTTPS** : Hypertext Transfer Protocol Secure

**IIS** : Internet Information Services

**ISP** : Interface Segregation Principle

**JS** : JavaScript

**JSON** : JavaScript Object Notation

**JWT** : Json Web Token

**LICEF** : Laboratoire d'Informatique Cognitive et Environnements de Formation

**LMS** : Learning Management System

**LSP** : Liskov Substitution Principle

**MSSQL** : MicroSoft Structured Query Language

**MVC** : Model-View-Controller

**OAuth** : Open Authorization

**OCP** : Open-Close Principle

**OIDC** : OpenID Connect

**ORM** : Object-Relational Mapping

**QoC** : Quality of Context

**QR code** : Quick Response code

**REST** : Representational State Transfer

**RFID** : Radio Frequency Identification

**RPC** : Remote Procedure Call

**SGBD** : Système de Gestion de Bases de Données

**SOAP** : Simple Object Access Protocol

**SOLID** : SRP (Liskov Substitution Principle) OCP(Open-Close Principle) LSP (Liskov Substitution Principle) ISP (Interface Segregation Principle) DIP (Dependency Inversion Principle)

**SQL** : structured Query Language

**SRP** : Single Responsibility Principle

**STI** : système tutoriel intelligent

**STS** : Security Token Service

**SVN** : Subversion

**TANGO** : Tag Added learNinG Objects

**TDD** : Test Driven Development

**TEEC** : Technologies Éducatives pour l'Enseignement en Contexte

**TEL** : Technology Enhanced Learning

**TELUQ** : télé-université du Québec

**TenseITS** : Tense Intelligent Tutoring System

**TLS** : Transport Layer Security

**UI** : User Interface

**UML** : Unified Modeling Language

**UoLmP** : Units of Learning mobile Player

**UQAM** : Université du Québec À Montréal

**URI** : Uniform Resource Identifier

**WIFI** : WIreless FIdelity

**XML** : Unified Modeling Language

**YAML** : YAAML Ain't Markup Language



## RÉFÉRENCES

- Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M. E. et Steggles, P. (1999). Towards a better understanding of context and context-awareness. *Handheld and Ubiquitous Computing, Proceedings, 1707*, 304–307.
- Abowd, G. D. et Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(1), 29–58.
- Adomavicius, G. et Jannach, D. (2014). Preface to the special issue on context-aware recommender systems. *User Modeling and User-Adapted Interaction*, 24(1-2), 1–5.
- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. et Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications magazine*, 40(8), 102–114.
- Almazan, C. B. (2010). *Rover : architectural support for exposing and using context*. (Thèse de doctorat).
- Anjou, C. (2018a). *L'énergie géothermique : représentations, enseignements et contextes aux Antilles et dans la zone caraïbe*. (Thèse de doctorat).
- Anjou, C. (2018b). Modélisations de contextes pour l'apprentissage par effets de contextes : un exemple sur le sujet de la géothermie.
- Anjou, C., Forissier, T., Bourdeau, J., Mazabraud, Y., Nkambou, R. et Fournier, F. (2017a). Elaborating the context calculator : A design experiment in geothermy. Dans *International and Interdisciplinary Conference on Modeling and Using Context*, 513–526.
- Anjou, C., Forissier, T. et Delcroix, A. (2017b). Le comparateur de contexte : Un outil numérique pour la création de scénarios pédagogiques collaboratifs.
- Aronson, E. et al. (1978). *The jigsaw classroom*. Sage.
- Bastien, C. (1998). Contexte et situation. Houdé, O., Kayser, D., Koenig, O., Proust, J. & Rastier, F., *Dictionnaire des Sciences Cognitives*. Paris : PUF.

- Bazire, M. et Brézillon, P. (2005). Understanding context before using it. Dans *Modeling and Using Context : 5th International and Interdisciplinary Conference, CONTEXT 2005, Paris, France, July 5-8, 2005, Proceedings*, volume 5, p. 29. Springer Science & Business Media.
- Berri, J., Benlamri, R. et Atif, Y. (2006). Ontology-based framework for context-aware mobile learning.
- Biehl, M. (2016). *RESTful API Design*, volume 3. API-University Press.
- Bikakis, A., Patkos, T., Antoniou, G. et Plexousakis, D. (2007). A survey of semantics-based approaches for context reasoning in ambient intelligence. Dans *European Conference on Ambient Intelligence*, 14–23. Springer.
- Bourdeau, J. (2017). The dbr methodology for the study of context in learning. Dans *International and Interdisciplinary Conference on Modeling and Using Context*, 541–553. Springer.
- Bourdeau, J., Forissier, T., Mazabraud, Y. et Nkambou, R. (2015). Web-based context-aware science learning. 1415–1418.
- Brézillon, P. (2003). Representation of procedures and practices in contextual graphs. *The Knowledge Engineering Review*, 18(2), 147–174.
- Brousseau, G. (1980). L'échec et le contrat. *Recherches*, 41, 177–182.
- Brown, A. L. (1992). Design experiments : Theoretical and methodological challenges in creating complex interventions in classroom settings. *The journal of the learning sciences*, 2(2), 141–178.
- Buchholz, T., Küpper, A. et Schiffers, M. (2003). Quality of context : What it is and why we need it. Dans *Workshop of the HP OpenView University Association*.
- Burbaite, R., Bespalova, K., Damasevicius, R. et Stuiikys, V. (2014). Context aware generative learning objects for teaching computer science. *International Journal of Engineering Education*, 30(4), 929–936.
- Byun, H. E. et Cheverst, K. (2004). Utilizing context history to provide dynamic adaptations. *Applied Artificial Intelligence*, 18(6), 533–548.
- Cassens, J. et Kofod-Petersen, A. (2006). Using activity theory to model context awareness : A qualitative case study. Dans *FLAIRS Conference*, 619–624.
- Chen, G., Kotz, D. *et al.* (2000). *A survey of context-aware mobile computing research*. technical report, Technical Report TR2000-381, Dept. of Computer

Science, Dartmouth College.

Cui, Y. et Bull, S. (2005). Context and learner modelling for the mobile foreign language learner. *System*, 33(2), 353–367.

Derntl, M. et Hummel, K. A. (2005). Modeling context-aware e-learning scenarios.

Dourish, P. (2004). What we talk about when we talk about context. *Personal and ubiquitous computing*, 8(1), 19–30.

Estdale, J. et Georgiadou, E. (2018). Applying the iso/iec 25010 quality models to software product. Dans *European Conference on Software Process Improvement*, 492–503. Springer.

Evans, D.-D. D. et Design, D.-D. (2003). Tacking complexity in the heart of software.

Flanagan, J. (1954). The critical incident technique. *Psychological bulletin*, 51(4), 327—358.

Forissier, T. (2017). Apprendre les sciences en comparant les contextes, principes, technologies et observations. Dans *Ouvrir les murs de la classe avec le numérique, CIRTA 2017*.

Forissier, T. (2019). *Contextualisation et effets de contextes dans l'apprentissage des Sciences*. (Thèse de doctorat).

Forissier, T., Bourdeau, J. et Fécil, S. (2014a). Interfaces élève-machine pour apprendre à partir des contextes.

Forissier, T., Bourdeau, J., Mazabraud, Y. et Nkambou, R. (2014b). *Computing the Context Effect for Science Learning*, Dans P. Brézillon et A. J. Gonzalez (dir.). *Context in Computing : A Cross-Disciplinary Approach for Modeling the Real World*, (p. 255–269). Springer New York.

Galleguillos, C. et Belongie, S. (2010). Context based object categorization : A critical survey. *Computer vision and image understanding*, 114(6), 712–722.

Gasparic, M., Murphy, G. C. et Ricci, F. (2017). A context model for ide-based recommendation systems. *Journal of Systems and Software*, 128, 200–219.

Guan, D., Yuan, W., Lee, S. et Lee, Y.-K. (2007). Context selection and reasoning in ubiquitous computing. Dans *The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)*, 184–187. Citeseer.

- Gómez, S., Zervas, P., Sampson, D. G. et Fabregat, R. (2014). Context-aware adaptive and personalized mobile learning delivery supported by uolmp. *Journal of King Saud University - Computer and Information Sciences*, 26(1, Supplement), 47–61.
- Henricksen, K. (2003). *A framework for context-aware pervasive computing applications*. (Thèse de doctorat).
- Henricksen, K. et Indulska, J. (2004). Modelling and using imperfect context information. Dans *PerCom Workshops*, 33–37. Citeseer.
- Hollister, D. L., Gonzalez, A. et Hollister, J. (2017). Contextual reasoning in human cognition and the implications for artificial intelligence systems. Dans P. Brézillon, R. Turner, et C. Penco (dir.). *Modeling and Using Context*, 599–608., Cham. Springer International Publishing.
- Hsu, T.-Y., Chiou, C.-K., Tseng, J. C. et Hwang, G.-J. (2016). Development and evaluation of an active learning support system for context-aware ubiquitous learning. *IEEE Transactions on Learning Technologies*, 9(1), 37–45.
- Hull, R., Neaves, P., Bedford-Roberts, J. et al. (1997). *Towards situated computing*. Hewlett Packard Laboratories.
- Kang, D., Sohn, J., Kwon, K., Joo, B.-G. et Chung, I.-J. (2014). An intelligent dynamic context-aware system using fuzzy semantic language. In *Mobile, Ubiquitous, and Intelligent Computing* 143–149. Springer.
- Kent Beck, Mike Beedle, A. v. B. A. C. W. C. M. F. J. G. J. H. A. H. R. J. J. K. B. M. R. C. M. S. M. K. S. J. S. D. T. (2001). Manifesto for agile software development. *Software Development*, 9(8), 28–35.
- Khedr, M. et Karmouch, A. (2004). Negotiating context information in context-aware systems. *IEEE Intelligent Systems*, 19(6), 21–29.
- Kim, Y. et Lee, K. (2006). 2, 576–581.
- King, D. T., Winner, E. et Ginns, I. (2011). Outcomes and implications of one teacher’s approach to context-based science in the middle years. *Teaching Science*, 57(2), 26–30.
- Krause, M. et Hochstatter, I. (2005). Challenges in modelling and using quality of context (qoc). Dans *Mobility Aware Technologies and Applications : Second International Workshop, MATA 2005, Montreal, Canada, October 17–19, 2005, Proceedings*, volume 3744, p. 324. Springer Science & Business Media.

- Krishnamoorthy, S. (2012). *Rover-II : A Context-Aware Middleware for Pervasive Computing Environment*. (Thèse de doctorat).
- Lonsdale, P., Baber, C., Sharples, M., Byrne, W., Arvanitis, T. N., Brundell, P. et Beale, R. (2005). Context awareness for mobile learning : creating an engaging learning experience in an art museum. *Mobile learning anytime everywhere*, 115, 115–118.
- Manzoor, A., Truong, H.-L. et Dustdar, S. (2014). Quality of context : models and applications for context-aware systems in pervasive environments. *The Knowledge Engineering Review*, 29(2), 154–170.
- Martin, R. C. (2017). Clean architecture.
- Mazabraud, Y., Thomas, F., Jacqueline, B. et Nkambou, R. (2013). Forissier, t., bourdeau, j., mazabraud, y., nkambou r. . modeling context effects in science learning : The clash model. in context 2013, lecture notes in computer science 8175 [subseries lecture notes in artificial intelligence : Springer], 330-338. *Lecture Notes in Computer Science, subseries Lecture Notes in Artificial Intelligence*, 330–338.
- Merlo-Leurette, S. et Forissier, T. (2009). La contextualisation dans l'enseignement des sciences et techniques en guadeloupe. *Grand N*, 83, 19–26.
- Nkambou, R., Mizoguchi, R. et Bourdeau, J. (2010). *Advances in intelligent tutoring systems*, volume 308. Springer Science & Business Media.
- Nurmi, P. et Floréen, P. (2004). Reasoning in context-aware systems. *Helsinki Institute for Information Technology, Position paper*.
- Oers, B. V. (1998). From context to contextualizing. *Learning and Instruction*, 8(6), 473–488.
- Ogata, H. et Yano, Y. (2004). Context-aware support for computer-supported ubiquitous learning. Dans *The 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education, 2004. Proceedings.*, 27–34.
- Palmisano, C., Tuzhilin, A. et Gorgoglione, M. (2008). Using context to improve predictive modeling of customers in personalization applications. *IEEE transactions on knowledge and data engineering*, 20(11), 1535–1549.
- Parker, J. E., Hollister, D. L., Gonzalez, A. J., Brézillon, P. et Parker, S. T. (2013). Looking for a synergy between human and artificial cognition. Dans P. Brézillon, P. Blackburn, et R. Dapoigny (dir.). *Modeling and Using Context*, 45–58. Springer Berlin Heidelberg.

- Perera, C., Zaslavsky, A., Christen, P. et Georgakopoulos, D. (2013). Context aware computing for the internet of things : A survey. *IEEE communications surveys & tutorials*, 16(1), 414–454.
- Podschuweit, S. et Bernholt, S. (2018). Composition-effects of context-based learning opportunities on students' understanding of energy. *Research in Science Education*, 48(4), 717–752.
- Pradeep, P. et Krishnamoorthy, S. (2019). The mom of context-aware systems : A survey. *Computer Communications*.
- Psyché, V., Anjou, C., Fennani, W., Bourdeau, J., Forissier, T. et Nkambou, R. (2018). Ontology-based context modelling for designing a context-aware calculator.
- Rising, L. et Janoff, N. S. (2000). The scrum software development process for small teams. *IEEE software*, 17(4), 26–32.
- Rogers, Y., Price, S., Randell, C., Fraser, D. S., Weal, M. et Fitzpatrick, G. (2005). Ubi-learning integrates indoor and outdoor experiences. *Commun. ACM*, 48(1), 55–59.
- Royce, W. (1970). Managing the development of large software systems [electronic resource]/dr. winston w. royce. *University of Maryland*, 23–28.
- Ryan, N. (1998). Enhanced reality fieldwork : the context-aware archaeological assistant. *Computer Applications in Archaeology 1997, British Archaeological Reports*.
- Sanchez, D., Tentori, M. et Favela, J. (2007). Hidden markov models for activity recognition in ambient intelligence environments. Dans *Eighth Mexican International Conference on Current Trends in Computer Science (ENC 2007)*, 33–40. IEEE.
- Schilit, B., Adams, N. et Want, R. (1994). Context-aware computing applications. Dans *1994 First Workshop on Mobile Computing Systems and Applications*, 85–90.
- Schilit, B. N. et Theimer, M. M. (1994). Disseminating active mop infonncition to mobile hosts. *IEEE network*.
- Schmidt, A. (2006). Ontology-based user context management : The challenges of imperfection and time-dependence. In *On the Move to Meaningful Internet Systems 2006 : CoopIS, DOA, GADA, and ODBASE*.

- Schmidt, A., Beigl, M. et Gellersen, H.-W. (1999). There is more to context than location. *Computers & Graphics*, 23(6), 893–901.
- Schwaber, K. (1997). Scrum development process. In *Business object design and implementation* 117–134. Springer.
- Schwartz, R. S., Lederman, N. G. et Crawford, B. A. (2004). Developing views of nature of science in an authentic context : An explicit approach to bridging the gap between nature of science and scientific inquiry. *Science Education*, 88(4), 610–645.
- Soualah-Alila, F. (2015). *CAMLearn : A Context-Aware Mobile Learning Recommender System. Application to M-Learning Domain*. (Thèse de doctorat).
- Sternberg, R. J., Wagner, R. K., Williams, W. M. et Horvath, J. A. (1995). Testing common sense. *American psychologist*, 50(11), 912–927.
- van Bunningen, A., Feng, L. et Apers, P. (2005). Context for ubiquitous data management. Dans *International Workshop on Ubiquitous Data Management*, 17–24. IEEE.
- van Eijck, M. et Roth, W.-M. (2010). Towards a chronotopic theory of “place” in place-based education. *Cultural Studies of Science Education*, 5(4), 869–898.
- van Wissen, A., Kamphorst, B. et van Eijk, R. (2013). A constraint-based approach to context. Dans *International and Interdisciplinary Conference on Modeling and Using Context*, 171–184. Springer.
- Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I. et Duval, E. (2012). Context-aware recommender systems for learning : A survey and future challenges. *IEEE Transactions on Learning Technologies*, 5(4), 318–335.
- Villegas, N. M. et Müller, H. A. (2010). Managing dynamic context to optimize smart interactions and services. In *The smart internet* 289–318. Springer.
- Wang, R. Y., Kon, H. B. et Madnick, S. E. (1993). Data quality requirements analysis and modeling. Dans *Proceedings of IEEE 9th International Conference on Data Engineering*, 670–677. IEEE.
- Weinburgh, M. (1995). Gender differences in student attitudes toward science : A meta-analysis of the literature from 1970 to 1991. *Journal of Research in Science Teaching*, 32(4), 387–398.

- Wilson, A. B. (1959). Residential segregation of social classes and aspirations of high school boys. *American Sociological Review*, 24(6), 836–845.
- Winograd, T. (2001). Architectures for context. *Human–Computer Interaction*, 16(2-4), 401–419.
- Wrona, K. et Gomez, L. (2006). Context-aware security and secure context-awareness in ubiquitous computing environments. *Annales Universitatis Mariae Curie-Sklodowska, sectio AI–Informatica*, 4(1).
- Zhou, R. et Rechert, K. (2008). Personalization for location-based e-learning. Dans *2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies*, 247–253.
- Zimmermann, A., Lorenz, A. et Oppermann, R. (2007). An operational definition of context. *Modeling and Using Context*, 558–571.