



HAL
open science

DEMOS : une méthode de conception participative de systèmes d'information soutenant la démocratie des organisations

Raphaëlle Bour

► **To cite this version:**

Raphaëlle Bour. DEMOS : une méthode de conception participative de systèmes d'information soutenant la démocratie des organisations. Informatique [cs]. Université Toulouse 1 Capitole, 2019. Français. NNT : . tel-02493752

HAL Id: tel-02493752

<https://hal.science/tel-02493752>

Submitted on 28 Feb 2020

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

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



THÈSE

En vue de l'obtention du DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 1 Capitole

Présentée et soutenue par
Raphaëlle BOUR

Le 28 novembre 2019

**DEMOS : une méthode de conception participative de systèmes
d'information soutenant la démocratie des organisations**

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et
Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Chantal SOULE-DUPUY et Nathalie VALLES-PARLANGÉAU

Jury

Mme Jolita RALYTE, Rapporteur
Mme Agnès FRONT, Rapporteur
M. Max Chevalier, Examineur
M. Régis Meissonnier, Examineur
Mme Chantal SOULE-DUPUY, Directrice de thèse
Mme Nathalie VALLES-PARLANGÉAU, Co-directrice de thèse

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE
Délivré par l'Université Toulouse 1 Capitole

Présentée et soutenue par

Raphaëlle BOUR

Le 28 novembre 2019

**DEMOS : une méthode de conception participative pour des
systèmes d'information soutenant la démocratie des organisations**

Ecole doctorale :

EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité : **Informatique et Télécommunications**

Unité de recherche : **IRIT - Institut de Recherche en Informatique de Toulouse**

Thèse dirigée par

Chantal SOULE-DUPUY et Nathalie VALLES-PARLANGEAU

Jury

Mme Jolita RALYTE, Rapporteur

Mme Agnès FRONT, Rapporteur

M. Max CHEVALIER, Examineur

M. Régis MEISSONNIER, Examineur

Mme Chantal SOULE-DUPUY, Directrice de thèse

Mme Nathalie VALLES-PARLANGEAU, Co-directrice de thèse

Mme. Maryse SALLES, invitée

Remerciements

Mes premiers remerciements vont à Chantal Soulé-Dupuy et Nathalie Vallès-Parlangeau, mes deux formidables directrices de thèse. Travailler avec vous fut pour moi un vrai plaisir, et vous m'avez été d'un immense soutien. La bienveillance avec laquelle vous m'avez laissée trouver mon chemin, la constance avec laquelle vous m'avez poussée à approfondir, détailler, justifier chacun des aspects de cette thèse m'ont permis de mener sereinement ce projet à bien. Votre disponibilité, votre écoute et votre ouverture d'esprit m'ont fait me sentir libre et encadrée, m'offrant ainsi un environnement de travail idéal. Pour toutes ces raisons, et pour beaucoup d'autres encore : un grand merci.

Je remercie Jolita Ralyté et Agnès Front, mes deux rapportrices de thèse. Les rapports que vous avez rédigés donnent une autre dimension à mon travail : j'étais convaincue, d'un point de vue personnel, que ce sujet était important et que l'aborder sous cet angle avait du sens, vous êtes venues confirmer que tout cela avait un intérêt pour notre communauté des systèmes d'information, et cela me réjouit. Merci pour la lecture attentive que vous avez faite de mon manuscrit et pour la pertinence de vos remarques.

Je remercie également Max Chevalier, qui a accepté la présidence de ce jury. Nos échanges lors de chacune de nos rencontres m'ont permis de mieux appréhender le monde de la recherche. Ton sérieux et ta rigueur, mêlés à ta bonne humeur et ta convivialité font que c'est aujourd'hui un véritable honneur que tu sois le président de mon jury de thèse.

Je remercie Régis Meissonnier, examinateur de cette thèse, qui a bien voulu faire un pas de côté pour poser son regard de gestionnaire sur mon travail de recherche en informatique. La pluridisciplinarité pour traiter des sujets portant sur les systèmes d'information me semble nécessaire, et vous venez confirmer cette idée en acceptant de faire partie de mon jury, je vous en remercie.

Je remercie très chaleureusement Maryse Salles, qui m'a tant apporté durant ces années. Mon intérêt pour ton travail n'a cessé de grandir durant la thèse, et de m'enrichir. J'ai été passionnée par nos échanges et tous ces instants partagés m'ont véritablement donné l'envie de faire de la recherche avec cet indispensable regard critique que tu défends. De plus, la confiance que tu m'as accordée m'a encouragée et me portera pour la suite, j'en suis certaine. Merci pour tout cela.

Maryse, nous co-animons depuis trois ans l'atelier « Systèmes d'information et de Décision et Démocratie ». La petite équipe que nous formons maintenant au sein de cet atelier m'a permis d'entrevoir ce que la recherche universitaire permet : créer des liens, des passerelles, des échanges entre différentes disciplines pour explorer un sujet en profondeur et faire des propositions, c'est une vision qui me donne envie de continuer ! Gabriel Colletis, Anne Isla, Etienne Fieux, Adrien Peneranda, Claude Paraponaris, Laurent Cloarec et Jean-Sebastien Vayre : nos réunions sont toujours réjouissantes. Chacun, à travers son point de vue, sa singularité, son propre domaine de recherche m'a enrichie, pour cela je vous remercie tous.

Je remercie Dominique Rieu et Agnès Front une fois encore pour leur accueil au LIG à Grenoble. Ce fut court mais intense, et ces quelques jours passés avec vous m'ont permis d'ancrer mon travail dans le domaine de l'ingénierie des méthodes, effaçant ainsi mes doutes quant à la coloration trop peu technique de mes recherches. Nos échanges ont marqué un tournant dans ma thèse, je vous remercie d'avoir pris ce temps avec moi et de m'avoir encouragée.

Je remercie toutes les personnes avec qui j'ai collaboré et mené des expérimentations : le service de la Formation Continue de l'université (Fadilou, Dominique, Delphine, Véronique, Sébastien) et les

enseignants qui ont participé au projet (Fred, David, Alain). Vous avez mis à ma disposition un peu de votre temps, de votre intelligence collective, et nos échanges ont donné un sens profond à mon travail, je vous en remercie.

Je remercie Laurence Capus, de l'Université Laval au Québec, avec qui le dialogue fut fructueux et très agréable. Tu m'as ouvert de nouvelles perspectives, de nouvelles possibilités de champs d'application dans le domaine des plateformes d'enseignement et nos collaborations à venir seront riches, je l'espère.

Je remercie l'ensemble de l'équipe SIG, et plus généralement la communauté INFORSID. Je me suis sentie accueillie dans cette communauté des systèmes d'information, de Biarritz à Grenoble, en passant par Nantes, Toulouse et Paris lors des congrès. La simplicité des échanges, la bienveillance et l'ouverture d'esprit de cette communauté sont assurément porteurs pour les doctorants. Merci à vous tous pour cela. Je remercie tout particulièrement Olivier Teste, notre directeur d'équipe qui a toujours pris le temps de répondre à mes interrogations et de partager son expérience.

Je remercie les enseignants-chercheurs de l'UFR d'informatique de l'Université Toulouse 1 Capitole. Ici aussi j'ai découvert une équipe conviviale de maîtres de conférences, de professeurs et d'administratifs qui accueille les doctorants et les intègre pleinement. J'ai pu profiter de vos conseils, de votre expérience, des opportunités que vous m'avez offertes pour apprendre à enseigner, et pour progresser. Merci Fred, Julien, Sylvie, Umberto, Eric, Alain, Chihab, David S. et David N., Laurent, Stéphane, Jean-Marc, Ronan, et ceux que j'oublie. Merci à Florence et Michèle, qui font un travail considérable ici pour nous simplifier autant que possible la vie. Un merci tout particulier à Franck, qui a suivi mon parcours depuis la première année de Master puisqu'il y était mon enseignant. C'est maintenant au collègue que je m'adresse : je te remercie pour tous les conseils que tu m'as donnés, ainsi que pour la bonne humeur dont tu fais (souvent) preuve, et qui participe à me faire aimer l'atmosphère de cette fac.

Je remercie mes collègues doctorants (et pour certains docteurs maintenant) : Jiefu, Caro, Dennis, Arianna, Audren, Mehdi, David et tous les autres. Nos discussions au quotidien m'ont permis de sortir la tête du guidon pour parler de politique, de musique, de féminisme, de famille, de cours, de voyage et de beaucoup d'autres choses... Cela fait un bien fou, donc merci !

Je remercie mes amis, ils se reconnaîtront. Vous êtes des personnes attentionnées, chaleureuses, joyeuses, passionnantes et plus encore. En me rendant la vie plus douce, vous m'avez soutenue durant cette thèse comme vous ne pouvez pas l'imaginer. Un grand merci à vous.

Je remercie ma famille. Merci Mam, tu me facilites la vie dès que tu le peux et m'encourages toujours, c'est très réconfortant. Merci Papa et Coco, vous m'avez épaulée tout au long de la thèse et vos conseils attentionnés m'ont permis de tracer la bonne route, je crois. Merci Alice, Julie, Clara, Hugo d'avoir compris ce que je faisais, de vous y être intéressé et de m'avoir poussée. Merci aussi à ma très belle-famille Cabrol, et particulièrement à Mireille et Thierry. Depuis le début, et encore aujourd'hui, votre aide précieuse me permet de me consacrer à mon travail l'esprit tranquille.

Mes derniers remerciements vont bien sûr à Lucas. Avec toi la vie est belle. Tu as laissé de l'espace dans notre quotidien pour que je puisse me consacrer à ce projet. Avec ton soutien et ton amour, j'ai pu faire ce chemin paisiblement. Nos merveilleux enfants, Marius et Bonnie ont apporté de la joie à tout cela. Merci à vous trois !

Table des matières

INTRODUCTION GENERALE	1
Contexte et problématique	1
Objectifs et contributions de la thèse	3
Organisation du document.....	3

PARTIE 1 - CONTEXTE ET ETATS DE L'ART

CHAPITRE 1 - SYSTEMES D'INFORMATION ET DEMOCRATIE.....	9
1.1. De la démocratie dans la société à la démocratie des organisations	9
1.1.1. La démocratie dans nos sociétés.....	10
1.1.1.1. Des définitions de la démocratie.....	10
1.1.1.2. Les caractéristiques et enjeux de la démocratie	11
1.1.2. Le lien entre démocratie dans la société et démocratie dans les organisations	12
1.1.3. La démocratie dans les organisations : grands principes.....	12
1.2. Les systèmes d'information et la démocratie dans les organisations.....	14
1.2.1. Notre vision du système d'information.....	14
1.2.2. Genèse du système d'information numérique	15
1.2.2.1. Le passage de la réalité au système d'information	16
1.2.2.2. Le passage du système d'information au système d'information numérique.....	19
1.2.2.3. L'usage des systèmes d'information	21
1.2.3. Les caractéristiques d'un système d'information soutenant la démocratie.....	23
1.3. Les enjeux à créer des systèmes d'information qui soutiennent la démocratie	25
1.3.1. Gérer le shadow IT.....	25
1.3.2. Générer de l'innovation grâce aux travailleurs cognitifs	27
1.3.3. Partager la connaissance avec le principe des Communs	29
1.4. Conclusion du chapitre.....	32
CHAPITRE 2 - ETAT DE L'ART : LES UTILISATEURS ET LEUR SYSTEME D'INFORMATION	35
2.1. La place des utilisateurs dans le processus de construction des systèmes d'information ...	36
2.1.1. Quelles étaient et quelles sont les motivations ?	36
2.1.2. L'implication des utilisateurs dans le processus de recueil des exigences	37

2.1.3.	Les différentes démarches de conception impliquant les utilisateurs.....	39
2.1.3.1.	La conception centrée utilisateurs	39
2.1.3.2.	La conception participative	40
2.1.4.	Le développement agile et la participation des utilisateurs	41
2.1.4.1.	Présentation des valeurs et principes agiles	42
2.1.4.2.	Présentation des méthodes agiles	43
2.1.4.3.	Le glissement sémantique de l'agilité : de l'utilisateur au client	46
2.1.5.	Notre analyse et notre positionnement.....	47
2.2.	Le concept de point de vue en conception de systèmes d'informations.....	49
2.2.1.	La question du point de vue en ingénierie des exigences.....	49
2.2.2.	La modélisation des points de vue	50
2.2.3.	Les limites identifiées et notre positionnement	51
2.3.	Conclusion du chapitre.....	52

CHAPITRE 3 - ETAT DE L'ART : INGENIERIE DES METHODES 53

3.1.	Méthode d'ingénierie des systèmes d'information	53
3.1.1.	Qu'est-ce qu'une méthode ?.....	54
3.1.2.	Les modèles de produit	55
3.1.2.1.	Définition du modèle de produit.....	55
3.1.2.2.	Syntaxe abstraite et syntaxe concrète	56
3.1.2.3.	Synthèse sur les modèles de produit	58
3.1.3.	Les modèles de processus	58
3.1.3.1.	Définition des modèles de processus.....	58
3.1.3.2.	Orientations des modèles de processus.....	59
3.1.3.3.	Synthèse sur les modèles de processus	60
3.1.4.	Plateformes support aux méthodes.....	60
3.2.	Ingénierie des méthodes.....	62
3.2.1.	Définition de l'ingénierie des méthodes et bonnes pratiques.....	62
3.2.2.	Les principes de l'ingénierie des méthodes	63
3.2.2.1.	La métamodélisation	63
3.2.2.2.	La réutilisation et la modularité	64
3.2.3.	Le processus d'ingénierie des méthodes	65
3.2.3.1.	Cadre de référence	65
3.2.3.2.	Première intention : définir l'objectif de la construction d'une méthode	66
3.2.3.3.	Deuxième intention : Construire une méthode	66
3.3.	Conclusion du chapitre.....	67

PARTIE 2 - DEMOS : a DEsign Method for demOcratic information System

CHAPITRE 4 - POSITIONNEMENT DE NOTRE CONTRIBUTION	71
4.1. Les objectifs de DEMOS.....	71
4.2. Les principes de DEMOS.....	73
4.3. Composition de la méthode DEMOS.....	75
4.3.1. Les caractéristiques du modèle de produit de DEMOS.....	75
4.3.2. Les caractéristiques des modèles de processus de DEMOS.....	76
4.3.3. Méta-métamodélisation de la méthode DEMOS.....	77
4.4. Conclusion du chapitre.....	79
CHAPITRE 5 - PRESENTATION DE DEMOS.....	81
5.1. Le modèle de produit	81
5.1.1. Les aspects représentationnels du modèle de produit.....	82
5.1.1.1. L'aspect fonctionnel du modèle de produit	82
5.1.1.2. L'aspect dynamique du modèle de produit.....	83
5.1.1.3. L'aspect perspectif du modèle de produit	84
5.1.1.4. L'aspect statique du modèle de produit.....	84
5.1.2. Syntaxe abstraite et sémantique du modèle de produit	85
5.1.3. Syntaxe concrète du modèle de produit.....	87
5.2. Les modèles de processus	91
5.2.1. MAP générales.....	92
5.2.2. Séquencement de la méthode	93
5.2.3. Décomposition du processus de construction	95
5.2.3.1. Section 1 : Identifier les utilisateurs par analyse de documents.....	95
5.2.3.2. Section 2 : Identifier les utilisateurs par entretiens	96
5.2.3.3. Section 3 : Définir les points de vue par expressions des visions.....	98
5.2.3.4. Section 4 : Définir les points de vue par élicitation des missions.....	99
5.2.3.5. Section 5 : Concevoir un modèle par point de vue par représentation	101
5.2.3.6. Section 6 : Définir les points de vue par affinement du point de vue.....	103
5.2.3.7. Section 7 : Consolider les modèles par point de vue par similarités.....	104
5.2.3.8. Section 8 : Concevoir un modèle par point de vue par affinement du modèle ..	105
5.2.3.9. Section 9 : Fin par implémentation dans une BD graphe.....	106
5.2.4. Décomposition du processus d'amélioration continue	109

5.2.4.1.	Section 10 : Modifier un modèle par point de vue par mise à jour	109
5.2.4.2.	Section 11 : Redéfinir les points de vue par affinement du point de vue	110
5.2.4.3.	Section 12 : Modifier un modèle par point de vue par représentation	110
5.2.4.4.	Section 13 : Consolider les modèles par point de vue par similarités.....	111
5.2.4.5.	Section 14 : Modifier un modèle par point de vue par affinement du modèle ..	111
5.2.4.6.	Section 15 : Redéfinir les points de vue par consolidation de modèle	112
5.2.4.7.	Section 16 : Fin par mise à jour de la BD	112
5.2.5.	Les outils de la méthode.....	113
5.2.5.1.	Le photolangage	113
5.2.5.2.	Le tableau blanc ou Metaplan	114
5.3.	La plateforme dédiée	115
5.3.1.	Structure fonctionnelle de la plateforme DEMOS.....	115
5.3.2.	Usage de la plateforme à la suite de l’atelier de cadrage	116
5.3.3.	Usage de la plateforme à la suite des ateliers point de vue	117
5.3.4.	Usage de la plateforme à la suite de l’atelier de mise en commun	117
5.4.	Conclusion du chapitre.....	118

CHAPITRE 6 - EXPERIMENTATION ET EVALUATION DE LA METHODE 119

6.1.	Première expérimentation : outil pour un service de Formation Continue.....	119
6.1.1.	Contexte de l’expérimentation et besoins exprimés	119
6.1.2.	Déroulement de l’expérimentation.....	120
6.1.2.1.	Phase préparatoire	120
6.1.2.2.	Atelier de cadrage	122
6.1.2.3.	Ateliers point de vue	124
6.1.2.4.	Atelier de mise en commun	126
6.1.2.5.	Implémentation.....	128
6.1.3.	Evaluation et retours d’expérience	130
6.1.3.1.	Evaluation par les participants	130
6.1.3.2.	Notre analyse de l’expérimentation.....	131
6.2.	Deuxième expérimentation : Plateforme Tandem Canada.....	132
6.2.1.	Contexte de l’expérimentation et besoins exprimés	132
6.2.2.	Préparation de l’expérimentation.....	133
6.3.	Conclusion du chapitre.....	134

CONCLUSION ET PERSPECTIVES	135
Bilan des contributions.....	135
Perspectives.....	136
Bibliographie	141
Webographie.....	151

Liste des figures

Figure 1 : Plan de la partie 1.1.....	9
Figure 2 : Notre vision de la démocratie.....	11
Figure 3 : Les caractéristiques de la démocratie dans les organisations.....	13
Figure 4 : Plan de la partie 1.2.....	14
Figure 5 : La matrice structurelle du système général (J. L. Le Moigne 1994).....	15
Figure 6 : Genèse du système d'information numérique.....	16
Figure 7 : Passage de la réalité au système d'information.....	17
Figure 8 : Passage du système d'information au système d'information numérique.....	19
Figure 9 : Du système d'information numérique à la vie au travail dans l'organisation.....	21
Figure 10 : Caractéristiques du système d'information soutenant la démocratie dans l'organisation.....	23
Figure 11 : Plan de la partie 1.3.....	25
Figure 12 : La responsabilité cloisonnée des acteurs : une menace d'enclosure.....	31
Figure 13 : Les principales phases du processus de construction de système d'information.....	35
Figure 14 : Plan de la partie 2.1.....	36
Figure 15 : L'implication des utilisateurs dans le processus d'ingénierie des exigences.....	37
Figure 16 : L'implication des utilisateurs dans les démarches de conception.....	39
Figure 17 : Les degrés d'implication de l'utilisateur (Kushniruk et Nohr 2016).....	39
Figure 18 : L'implication des utilisateurs dans le développement des systèmes d'information.....	42
Figure 19 : Drawing Mona Lisa Iteratively and Incrementally (Patton 2008).....	45
Figure 20 : Plan de la partie 2.2.....	49
Figure 21 : Plan de la partie 3.1.....	54
Figure 22 : Niveaux d'abstraction relatifs au produit.....	55
Figure 23 : Décomposition du modèle de produit.....	58
Figure 24 : Niveaux d'abstraction relatifs au processus.....	59
Figure 25 : Extrait du métamodèle MAP.....	60
Figure 26 : Syntaxe concrète d'une MAP.....	60
Figure 27 : Définition d'une plateforme support à une méthode (Dupuy-Chessa et al. 2014).....	61
Figure 28 : Plan de la partie 3.2.....	62
Figure 29 : Les niveaux d'abstraction (Ralyte 2006).....	64
Figure 30 : Spectre des approches d'ingénierie des méthodes.....	64
Figure 31 : Cadre de référence du processus d'ingénierie des méthodes (Ralyte 2001).....	65
Figure 32 : Vision complète de la description d'une méthode.....	68
Figure 33 : Plan du chapitre 4.....	71
Figure 34 : Les objectifs de DEMOS.....	73
Figure 35 : Les principes de DEMOS.....	74
Figure 36 : Formalisation du modèle de produit de DEMOS.....	76
Figure 37 : Formalisation du modèle de processus de DEMOS.....	77
Figure 38 : Méta-métamodèle de produit pour la conception de la méthode DEMOS.....	78
Figure 39 : Méta-métamodèle de processus pour la conception de la méthode DEMOS.....	79
Figure 40 : Plan du chapitre 5.....	81
Figure 41 : Les aspects représentationnels du modèle de produit.....	82
Figure 42 : Syntaxe abstraite du modèle de produit de DEMOS.....	85
Figure 43 : Représentation des concepts et descripteurs dans le dictionnaire des données.....	89
Figure 44 : Enchaînement des processus de DEMOS.....	91
Figure 45 : MAP générale du processus de construction.....	92
Figure 46 : MAP générale du processus d'amélioration continue.....	92

Figure 47 : Sous-map de la section 1.....	96
Figure 48 : Sous-map de la section 2.....	97
Figure 49 : Sous-map de la section 3.....	98
Figure 50 : Sous-map de la section 4.....	100
Figure 51 : Sous-map de la section 5.....	101
Figure 52 : Sous-map de la section 6.....	103
Figure 53 : Sous-map de la section 7.....	104
Figure 54 : Sous-map de la section 8.....	106
Figure 55 : Sous-map de la section 9.....	107
Figure 56 : Instanciation multiple avec DEMOS.....	108
Figure 57 : Principe de spécialisation multiple.....	108
Figure 58 : Zoom sur la section 10.....	109
Figure 59 : Zoom sur la section 11.....	110
Figure 60 : Zoom sur la section 12.....	111
Figure 61 : Zoom sur la section 13.....	111
Figure 62 : Zoom sur la section 14.....	112
Figure 63 : Zoom sur la section 15.....	112
Figure 64 : Zoom sur la section 16.....	113
Figure 65 : Activité de photolangage lors d'un atelier DEMOS.....	114
Figure 66 : L'usage du tableau blanc dans les ateliers de DEMOS.....	115
Figure 67 : Structure fonctionnelle de la plateforme DEMOS.....	116
Figure 68 : Création des points de vue sur la plateforme.....	116
Figure 69 : Choix du point de vue sur la plateforme.....	117
Figure 70 : Création des concepts et descripteurs sur la plateforme.....	117
Figure 71 : Création des liens de similarité sur la plateforme.....	118
Figure 72 : Chemin dans la MAP pour l'expérimentation FCV2A.....	120
Figure 73 : Section 1 pour l'expérimentation FCV2A.....	121
Figure 74 : Section 2 pour l'expérimentation FCV2A.....	121
Figure 75 : Liste des utilisateurs pour l'expérimentation FCV2A.....	121
Figure 76 : Section 3 pour l'expérimentation FCV2A.....	122
Figure 77 : Illustration de la stratégie 3.3.....	122
Figure 78 : Mind map des visions.....	123
Figure 79 : Section 4 pour l'expérimentation FCV2A.....	123
Figure 80 : Extrait d'instanciation d'une première partie du modèle de produit.....	124
Figure 81 : Préparation de la salle pour l'atelier point de vue.....	124
Figure 82 : Section 5 pour l'expérimentation FCV2A.....	125
Figure 83 : Illustration de la stratégie 5.2.....	125
Figure 84 : Extrait d'instanciation d'une deuxième partie du modèle de produit.....	126
Figure 85 : Section 7 pour l'expérimentation FCV2A.....	126
Figure 86 : Illustration de l'activité 7.3.....	127
Figure 87 : Extrait d'instanciation d'une troisième partie du modèle de produit.....	127
Figure 88 : Section 8 pour l'expérimentation FCV2A.....	128
Figure 89 : Section 9 pour l'expérimentation FCV2A.....	128
Figure 90 : Extrait de la base de données Neo4J.....	129
Figure 91 : Création d'un stagiaire sur l'application.....	129
Figure 92 : Liste d'appel sur l'application.....	130

Liste des tableaux

Table 1 : Grille d'analyse de la participation des utilisateurs (Kautz 2011).....	47
Table 2 : Les neuf principes de Moody (Zehnder et al. 2017).....	57
Table 3 : Sémantique du modèle de produit de DEMOS.....	86
Table 4 : Syntaxe concrète du modèle de produit de DEMOS.....	87
Table 5 : Evaluation de la syntaxe concrète de DEMOS.....	90
Table 6 : Lien entre principes et intentions de DEMOS.....	93
Table 7 : Séquencement de la méthode DEMOS.....	94
Table 8 : Synthèse de la section 1 de DEMOS.....	96
Table 9 : Synthèse de la section 2 de DEMOS.....	97
Table 10 : Synthèse de la section 3 de DEMOS.....	98
Table 11 : Synthèse de la section 4 de DEMOS.....	100
Table 12 : Synthèse de la section 5 de DEMOS.....	102
Table 13 : Synthèse de la section 6 de DEMOS.....	103
Table 14 : Synthèse de la section 7 de DEMOS.....	105
Table 15 : Synthèse de la section 8 de DEMOS.....	106
Table 16 : Synthèse de la section 9 de DEMOS.....	107
Table 17 : Protocole d'évaluation de la méthode.....	131

INTRODUCTION GENERALE

Contexte et problématique

Pourquoi partir à la recherche de quelque chose dont tout le monde parle, que certains vendent avec succès et qui, par là même, semble se présenter comme un objet bien défini : le système d'information de l'organisation ? (Mélèse 1979) Mélèse a répondu à cette question en son temps, et ses réflexions résonnent plus que jamais 40 ans plus tard. Lorsque, à l'aube des années 80, Mélèse s'interrogeait sur le concept d'information, les organisations ont vu apparaître de nouvelles technologies informatiques, signes d'une « *montée des réseaux* » en leur sein (Mallard 2014). La décennie suivante a été marquée par l'introduction dans le langage des organisations des notions d'information et de TIC (Technologies de l'Information et des Communications) qui se sont répandues plus encore au début des années 2000, marquant l'ère de la transformation digitale des organisations. Aujourd'hui, les flux informationnels ont « *valeur de force motrice* » dans l'organisation (Raffenne 2007) et le mouvement d'informatisation du travail n'a cessé de se renforcer.

Alors que Mélèse nous demandait s'il fallait « *répudier le concept de système d'information* », (Mélèse 1979), il nous semble plus que jamais indispensable de continuer à étudier ce concept car il pose des questions déterminantes sur la vie dans les organisations. Dans les années 80 déjà, deux visions s'opposaient quant aux outils du système d'information. Certains, dans une perspective foucauldienne, voyaient en la bureautique une technologie de pouvoir tandis que d'autres nous engageaient à envisager les multiples possibilités d'innovation organisationnelle qui s'annonçaient. Dans les deux discours, la notion d'usage a rapidement trouvé une place centrale, qui s'est confirmée dans les années 90 lorsque la dichotomie flexibilité / rigidité a été au cœur des débats. L'usage a alors été considéré comme un objet de médiation, c'est-à-dire comme un filtre entre technologie et organisations via l'appropriation des outils par les acteurs.

Aujourd'hui, les questions sur l'impact des systèmes d'information dans l'organisation se durcissent. Dans le cadre d'une transformation du travail plus générale, le système d'information est vu comme un « *catalyseur des mutations organisationnelles déjà largement engagées* » (Boboc 2017). Ainsi, alors que « *l'information circule de plus en plus vite* », le travail s'intensifie grâce aux nouveaux outils de transmission de l'information (Gaborieau 2012). La capacité de contrôle du numérique sur le travail est décriée, et contraste avec les discours sur l'autonomie et la responsabilisation prônés par les managers (Tissier 2018). Des exemples tels que le travail sous commande vocale dans les entrepôts dévoilent le caractère « *destructif* » de certains outils (Gaborieau 2012) et interrogent dans certains cas sur la capacité du système d'information à permettre aux travailleurs de s'émanciper d'une conception taylorienne du travail (Colletis 2014).

Cette propagation des systèmes d'information dans l'univers productif s'est naturellement accompagnée d'une multiplication des projets informatiques. Et le peu de réussite de ces projets a de quoi alerter : Le Standish Group nous informait dans son rapport de 2015 que 52% des projets étaient en difficulté et 31% en échec (Standish Group 2015). Les deux facteurs principaux d'échec de ces projets sont, selon les cadres les mettant en place, le « *manque d'implication des utilisateurs* » et une « *expression incomplète du besoin* ». En ayant ces chiffres à l'esprit, il semble peu étonnant que certains outils de système d'information aient un impact aussi néfaste que nous l'avons évoqué sur la vie au travail. Cela signifie-t-il pour autant qu'il suffirait d'impliquer plus les utilisateurs et d'avoir une meilleure expression du besoin pour régler le problème ? Mais alors comment faire pour impliquer ces utilisateurs sachant que le Ministère du Travail nous indiquait récemment que les projets informatiques sont de véritables « *facteurs de risques psychosociaux dans l'entreprise* » (Ministère du travail 2018.) ? Il nous semble que la question du système d'information dans l'organisation doit s'inscrire dans un cadre plus général et pluridisciplinaire pour être envisagée de manière pertinente.

En repartant des critiques sur la rigidification du travail et sur le pouvoir exercé par les dispositifs numériques dans l'organisation, puis en nous appuyant sur la question de l'implication des utilisateurs dans les projets, il nous a semblé pertinent de nous ouvrir à la problématique plus générale de la démocratie. D'une part car la question des rapports de pouvoir est une question démocratique, qui peut se poser tant dans la société que dans l'organisation (Jardat 2012). D'autre part car la question de « *l'initiative citoyenne* », profondément d'actualité (Cazals 2010), interroge les possibilités laissées à chacun de discuter des cadres qui régissent sa vie : qu'il s'agisse de permettre au citoyen de faire bouger les « *murs de verre* » qui l'entourent (Gourgues et O'miel 2019) ou de permettre au travailleur de participer à la définition du système d'information qui normalise ses activités (Gori et Del Volgo 2009). Qui plus est, les possibilités de s'opposer, de marquer son désaccord (Ogien et Laugier 2011) sont des valeurs intrinsèques à la démocratie et elles se posent autant dans le champ de la vie citoyenne que dans l'organisation.

La démocratie dans nos sociétés constitue un véritable sujet de débat aujourd'hui (Fressoz 2019). Ses limites sont questionnées et nous sommes en pleine « *crise des démocraties contemporaines* » d'après nombre d'intellectuels (Piketty 2019). La recherche de solutions est encore une fois l'occasion de dresser un parallèle entre problématiques sociétales et problématiques organisationnelles. Entre autres : quelle place pour le citoyen dans la société, quelle place pour le travailleur dans l'organisation et quelle place pour l'utilisateur dans son système d'information ? Quels processus collectifs de délibération démocratique peut-on proposer quant aux normes dans la société, quant aux systèmes d'information qui incarnent les normes de l'organisation (Salais 2010) ?

Dans la communauté « Système d'information » en France, ces questions sont trop peu traitées car l'approche technique autour des systèmes d'information domine. Nous avons choisi, en nous inscrivant dans la perspective du constructivisme critique (Feenberg 2004), d'apporter une coloration pluridisciplinaire à notre recherche sur ce thème, tout en l'ancrant dans le domaine de l'informatique. C'est dans ce contexte qu'est né l'atelier « Système d'information et Démocratie » (<https://si2d.hypotheses.org/>), lors du colloque Inforsid 2017. En adressant cette problématique avec des chercheurs de spécialités diverses (informatique, droit, sociologie, mathématiques, économie, gestion), nous faisons dialoguer depuis trois ans ces disciplines pour questionner le lien entre système d'information et démocratie dans l'organisation.

L'un des axes de recherche permettant d'explorer ce sujet est l'éthique informatique. C'est un thème d'actualité dont les pouvoirs publics s'emparent aujourd'hui avec, par exemple, la proposition de création d'un conseil national d'éthique du numérique dans le Rapport Villani paru il y a peu (Villani 2018). Cependant, la question est souvent abordée sous l'angle des problématiques d'Intelligence

Artificielle. Ce n'est pas dans ce champ que nous nous inscrivons, mais bien dans celui des systèmes d'information et de leur conception, où la question est très peu traitée. Dernièrement, un guide du CIGREF à l'usage des acteurs du numérique proposait néanmoins d'aborder cette question d'un point de vue opérationnel, en proposant par exemple des catégories telles que « *l'ethic by design* » et « *l'éthique d'usage* » (CIGREF 2018). Avant cela, en 2015, Maryse Salles avait ouvert la voie, avec notamment une contribution majeure sur la responsabilité éthique des concepteurs de système d'information (Salles 2015). Elle y traitait déjà de la question des valeurs éthiques en informatique, parmi lesquelles la démocratie occupait la place de « *valeur éthique principale* ».

Il nous a semblé que l'éthique dans la conception de système d'information était un sillon à creuser pour que notre recherche en système d'information serve un objectif plus large de démocratie dans les organisations. Ainsi, nous avons choisi de proposer une méthode de conception de systèmes d'information soutenant la démocratie des organisations. Dès le départ nous avons décidé que notre contribution serait opérationnelle, qu'elle pourrait être expérimentée, adaptée à partir d'expériences de terrain. Il nous fallait pour cela proposer une méthode structurée, formalisée, précise et reproductible. Notre contribution : DEMOS (a DEsign Method for demOcratic information System), poursuit tous les objectifs cités plus haut. Ils sont présentés plus en détail dans la partie suivante de cette introduction.

Objectifs et contributions de la thèse

Notre projet de recherche, au travers de cette thèse, est de proposer une méthode de conception de systèmes d'information supportant la démocratie des organisations. Les résultats attendus de cette contribution sont les suivants :

- Une démarche de construction et d'amélioration continue de systèmes démocratiques, c'est-à-dire impliquant les futurs utilisateurs et leur permettant de débattre des normes et valeurs qui sous-tendent ces systèmes.
- La formalisation d'un métamodèle instanciable qui permette de construire des systèmes d'information « plus démocratiques ». Cela signifie pour nous des systèmes prenant en compte et reflétant une pluralité de points de vue, parfois divergents, mais toujours liés entre eux pour constituer un système partageable.
- L'outillage de notre méthode, sous la forme d'une plateforme dédiée. Celle-ci doit permettre de garantir le respect des points de vue exprimés lors de la conception, et assurer une traçabilité de tous les éléments validés durant la démarche.

Organisation du document

Notre thèse se divise en deux grandes parties : d'une part une présentation du contexte et les états de l'art des domaines dans lesquels nous nous inscrivons, d'autre part une présentation de la méthode DEMOS.

La première grande partie, consacrée à la présentation du contexte de notre travail et aux différents états de l'art sur lesquels nous nous sommes appuyés, se décompose en trois chapitres :

- Chapitre 1 - Systèmes d'information et démocratie
- Chapitre 2 - Etat de l'art sur l'implication des utilisateurs dans la construction des systèmes d'information
- Chapitre 3 - Etat de l'art sur l'ingénierie des méthodes

Dans le premier chapitre de cette partie, nous présentons tout d'abord le contexte dans lequel s'inscrit notre travail : les organisations, et plus précisément la démocratie dans les organisations. Nous avons choisi d'orienter notre recherche vers la démocratie car c'est un enjeu majeur, contemporain, et qui touche à toutes les sphères de nos vies. Nous souhaitons que notre travail de recherche dans le domaine de l'informatique permette d'adresser des problématiques humaines, concrètes et réelles. Nous espérons ainsi contribuer, avec la plus grande humilité, à l'amélioration des conditions de travail des personnes dans les organisations. Cette volonté a germé au fil de mon expérience de consultante en systèmes d'information, au plus près des utilisateurs, qui m'a montré à quel point les outils informatiques utilisés quotidiennement pouvaient jouer un rôle de catalyseur, aussi bien d'énergie et de bonne volonté durant les projets informatiques que d'inquiétude et de souffrance dans de nombreuses situations.

Nous nous penchons ensuite sur la question de la place des utilisateurs dans les projets informatiques. Cette question est centrale puisqu'elle est en lien direct avec la question démocratique : si les utilisateurs ne sont pas impliqués dans les projets informatiques, s'ils ne participent pas à la prise de décision sur les sujets les concernant directement, s'ils sont exclus du processus dès lors que les problématiques techniques sont abordées, alors comment imaginer que le système d'information puisse être construit démocratiquement ? A l'inverse, si les utilisateurs participent, coconçoivent, coconstruisent leur système d'information, dans quelle mesure cela peut-il contribuer à soutenir la démocratie dans l'organisation ? Ces questions, peu traitées dans la littérature, sont au cœur de nos réflexions et de notre contribution.

Enfin, notre travail s'inscrit dans le domaine de l'ingénierie des méthodes. En effet, notre contribution est une méthode de conception de systèmes d'information soutenant la démocratie des organisations, et il nous semblait pour cette raison indispensable de comprendre les tenants et aboutissants de la création d'une méthode. Ce troisième chapitre nous permet d'envisager ce sujet en nous appuyant sur les travaux existants, fondateurs, qui formalisent tous les aspects d'une méthode et de son processus de construction. Les choix que nous faisons parmi tous les éléments de la littérature du domaine sont une fois encore conduits par cet objectif de démocratie, afin de proposer une solution adaptable, opérationnelle et dont les utilisateurs peuvent s'emparer singulièrement en fonction de leurs besoins.

La deuxième grande partie de notre thèse est consacrée à la méthode que nous proposons : DEMOS. Elle est divisée en trois grands chapitres :

- Chapitre 4 - Positionnement de notre contribution
- Chapitre 5 - Présentation de DEMOS
- Chapitre 6 - Expérimentations de la méthode et évaluations.

Pour cette seconde grande partie, et en nous appuyant sur les trois premiers chapitres, nous articulons le positionnement de DEMOS autour de trois questions, permettant chacune de faire écho aux chapitres précédents. Ainsi, pour répondre à la question : *pourquoi DEMOS ?* nous reprenons les éléments de contexte que nous avons identifiés, et notamment les caractéristiques d'un système d'information soutenant la démocratie des organisations. Pour répondre à la question : *qu'est-ce que DEMOS ?* nous nous appuyons sur notre état de l'art sur la place de l'utilisateur dans la construction de systèmes d'information. Enfin, pour répondre à la question : *comment la méthode DEMOS est-elle construite ?* nous nous positionnons par rapport à notre état de l'art sur l'ingénierie des méthodes.

La méthode DEMOS est présentée dans un deuxième temps. Nous proposons le modèle de produit de la méthode, et en expliquons en détail chacun des éléments constitutifs. Nous présentons également les modèles de processus de la méthode, c'est-à-dire la démarche associée au modèle de produit. Pour

cela, nous formalisons chacune des stratégies mises en œuvre, et détaillons le séquençement de la méthode, ainsi que les outils supports que nous utilisons. Enfin, nous faisons un focus sur la plateforme support à la démarche que nous avons développée.

Dans la dernière partie nous présentons une expérimentation complète qui a été menée en 2018, avec le service de Formation Continue de l'Université Toulouse 1 Capitole pour un projet d'outil de gestion des présences. Cette expérimentation est largement illustrée par les résultats réels obtenus, et nous présentons l'évaluation de la méthode, ainsi que les pistes d'amélioration envisagées. Ces pistes sont ensuite creusées et font l'objet de nouvelles propositions dans la présentation d'une expérimentation à venir pour le projet de plateforme « Tandem Canada » dans le courant de l'année 2020.

Une conclusion vient ensuite clore cette thèse. Nous y présentons un bilan de nos contributions ainsi que des perspectives de court et de long terme.

Partie I

Contexte et états de l'art

Chapitre 1

SYSTEME D'INFORMATION ET DEMOCRATIE

Notre vision du système d'information est politique¹ au sens étymologique du terme. Nous nous positionnons dans un contexte organisationnel dans lequel la question de la démocratie nous semble absolument centrale et où le système d'information tient une place stratégique extrêmement impactante. C'est la raison pour laquelle nous avons choisi que cette thèse explore dans un premier temps le lien entre système d'information et démocratie dans les organisations. Les problématiques fondamentales que ce sujet comporte sont analysées dans ce chapitre à travers plusieurs questions : Comment définir la démocratie dans la société et comment définir la démocratie dans les organisations ? Quels sont les liens entre systèmes d'information et démocratie et comment ces systèmes d'information peuvent soutenir la démocratie des organisations ? A quels enjeux la proposition de systèmes d'information « démocratiques » répond-elle ?

1.1. De la démocratie dans la société à la démocratie des organisations

La démocratie semble être un sujet de débat entre philosophes depuis la Grèce antique. Elle fait l'objet de redéfinitions constantes, et les philosophes contemporains continuent de livrer de nouveaux éclairages sur ce concept. Dans un premier temps, et en nous basant sur diverses définitions de la littérature, nous proposons notre vision de la démocratie dans la société. Dans un second temps, notre travail s'inscrivant dans le cadre des organisations (entreprises, collectivités, institutions, associations), nous explicitons les liens qui existent entre démocratie dans la société et démocratie dans les organisations, afin notamment de mettre en exergue le fait qu'elles sont inséparables. L'exploration de ces liens nous permet dans un dernier temps de dresser une liste des grands principes qui soutiennent la démocratie dans les organisations (cf. Figure 1).

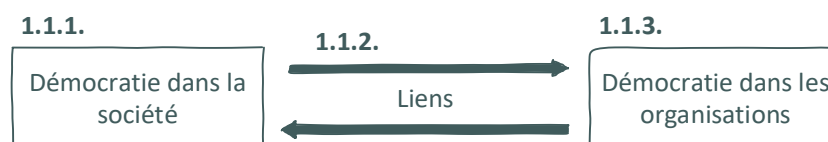


Figure 1 : Plan de la partie 1.1

¹ Le nom Politique est formé à partir de deux termes grecs : « polis », qui signifie « cité » et « -ikos », suffixe d'adjectif qui donne « -ique » en français. Ce mot est donc à l'origine un adjectif qui signifie « qui concerne le citoyen ».

1.1.1. La démocratie dans nos sociétés

La démocratie dans la société est définie par nombre de philosophes. Nous proposons dans un premier temps des définitions contemporaines qui adressent différentes facettes de ce concept. Dans un second temps, et en nous basant sur ces définitions, nous livrons notre vision de la démocratie, au travers des différentes caractéristiques que nous identifions. Dans un troisième et dernier temps nous partageons notre analyse des enjeux auxquels la démocratie doit selon nous répondre.

1.1.1.1. Des définitions de la démocratie

« Démocratie » vient de « Demos » : peuple et « Kratos » : pouvoir. Sa définition est mouvante et n'a cessé d'être repensée, rediscutée, remise en cause depuis la Grèce Antique. Sa réduction à la célèbre phrase de Lincoln « *le gouvernement du peuple, par le peuple, pour le peuple* », bien que nous ne la remettons pas en cause, nous semble insuffisante. Elle ne dit en effet rien des moyens que l'on peut mobiliser, des buts à atteindre, et n'apporte pas de précision sur ce que recouvre le terme de « peuple ». Des philosophes contemporains tels que Cornelius Castoriadis et Paul Ricoeur proposent des définitions plus précises, tout en conservant une vision large de ce qu'est la démocratie dans nos sociétés. Nous avons choisi de nous baser sur les réflexions de ces deux philosophes qui apportent un éclairage plus moderne sur ce concept.

« *La démocratie est le régime de l'autolimitation, autrement dit le régime de l'autonomie, ou de l'auto-institution* ». Au travers de cette définition, Castoriadis nous fait part de sa vision dite « radicale » de la démocratie. Selon lui, l'enjeu principal est de remettre en permanence en question les lois « *monopolisées par l'institué* », c'est-à-dire les pouvoirs qui sont entre les mains de l'exécutif, du législatif et du judiciaire (Castoriadis 1975). Il lie donc directement la définition de la démocratie à la question du pouvoir et de sa limitation.

Ricoeur pense la démocratie en termes éthiques (Ricoeur 1985). Il en donne deux définitions : l'une liée à la notion de conflit, et l'autre qui interroge la notion de pouvoir. « *Est démocratique un état qui ne se propose pas d'éliminer les conflits mais d'inventer les procédures leur permettant de s'exprimer et de rester négociables* ». De cette définition découle l'absolu besoin de respecter la pluralité des points de vue, incarnés par exemple par les partis politiques. Cette vision proposée par Ricoeur nous semble particulièrement intéressante car elle ne personnifie pas la question du pouvoir, mais pose des règles simples permettant la « vie en démocratie » dans nos sociétés. Il ajoute à cela la nécessité de la « *formation d'une opinion publique libre dans son expression* », sans quoi les procédures dont il est question seraient tellement biaisées qu'elles rendraient caduque la définition même de démocratie qu'il énonce. Cette deuxième proposition de Ricoeur, bien qu'absolument essentielle, est un sujet de débat inépuisable : qu'est-ce qu'une opinion publique libre ? Sans traiter de cette question dans le cadre de notre travail, cette proposition nous semble soulever un point à ne pas négliger : le droit à l'information, à l'accès à la connaissance, permettant à chacun d'être libre dans l'opinion qu'il se construit.

Ricoeur propose une seconde définition de la démocratie : « *la démocratie est le régime dans lequel la participation à la décision est assurée à un nombre toujours plus grand de citoyens*. » Cela ne peut se faire sans un rapprochement, une diminution de l'écart entre « *le citoyen et le souverain* ». Il ajoute en complément la condition sine qua none au respect de cette définition qu'est la séparation claire des trois pouvoirs (Ricoeur 1985). Outre cette question de la séparation des trois pouvoirs, il est à noter que Ricoeur utilise ici le terme de « participation à la décision », qui est selon nous une proposition très forte ne se limitant pas à la possibilité unique d'élire des représentants, mais qui oblige une société

dite démocratique à proposer à ses citoyens des façons de prendre part aux décisions le plus fréquemment possible.

Mais la définition de la démocratie est encore plus récemment remise en question car considérée par certains comme « *indécidable* ». En effet pour Caudron, sa dimension autoréférentielle, c'est-à-dire le fait qu'elle correspond à l'idéal auquel adhère celui qui tente de la définir, rend la notion même de démocratie indéfinissable. Il ajoute à cela que l'on analyse avant tout des pratiques démocratiques, ou au contraire entravant la démocratie, et que la démocratie revêt donc un caractère dynamique, et non statique (Caudron 2008). Nous partageons ce point de vue qui, d'une part, invite chacun à proposer sa propre définition de la démocratie et, d'autre part, nous ouvre à la possibilité de considérer la démocratie comme un processus, et par conséquent à la définir comme un ensemble de moyens en vue d'un but à atteindre.

1.1.1.2. Les caractéristiques et enjeux de la démocratie

Ces définitions contemporaines nous invitent à proposer notre vision personnelle de la démocratie. Nous avons choisi pour cela de dresser une liste de ce qui caractérise pour nous la démocratie, plutôt que d'en livrer une nouvelle définition formelle.

Parmi ces caractéristiques, la première pourrait être la séparation des trois pouvoirs : législatif (qui rédige et adopte les lois), exécutif (qui met en œuvre les lois) et judiciaire (qui veille au respect des lois). La seconde caractéristique serait la participation de chacun à la prise de décision. Cette participation, selon la « mouvance » à laquelle on se réfère peut être directe ou indirecte. Elle peut ainsi reposer sur une « *égalité de statuts* » entre tous (Jardat 2012), ou s'exercer par le biais de représentant élus. La troisième caractéristique serait le droit à la remise en cause constante des lois par le citoyen éclairé, c'est-à-dire libre de son opinion (cf. Figure 2). Il est à noter que cette troisième caractéristique comporte un aspect largement théorique et complexe portant sur la question de la libre information.



Figure 2 : Notre vision de la démocratie

Maintenant que nous avons défini ce qu'est pour nous la démocratie, nous nous intéressons aux enjeux auxquelles elle doit répondre. Pour commencer, la démocratie se définit avant tout en opposition à ce qui n'est pas démocratique (Caudron 2008). Le principal enjeu de la démocratie est donc tout naturellement de résister aux forces anti-démocratiques qui pourraient l'entraver. Dans nos sociétés cela se traduit par le fait de résister aux dérives totalitaires ou dictatoriales qui pourraient se présenter. De façon moins globalisante, il s'agit plus précisément de lutter contre les abus d'un pouvoir trop concentré. Pour cela, la démocratie proposant de mettre le pouvoir entre les mains de chacun, l'exercice de ce pouvoir par les citoyens doit pouvoir exister en dehors de la délivrance d'un mandat lors d'élections.

Le deuxième enjeu de la démocratie pourrait être résumé dans la notion d'intérêt du collectif. En effet, la sauvegarde de la liberté des membres du collectif, la stabilité et la résilience de ce collectif sont les

« *effets vertueux attendus de la démocratie* » (Jardat 2012). On peut rapprocher cette idée de « collectif » à celle de « peuple » évoquée plus haut. Il s'agit alors de comprendre que la démocratie a pour but de servir les intérêts du peuple, du plus grand nombre, et non de servir des intérêts individuels.

Enfin, et dans une perspective collective toujours, la démocratie est une façon de se « *fixer ensemble les fins qui [nous] semblent justes* » (Brière et Le Texier 2018). Cet enjeu nous permet d'aborder les résultats attendus de la démocratie, la finalité du processus. On peut alors considérer que la démocratie, en tant que processus, vise une finalité collective, sociétale, qui pourrait être résumée sous les expressions (certes galvaudées) de bien-être et de justice.

1.1.2. Le lien entre démocratie dans la société et démocratie dans les organisations

Pourquoi lier démocratie dans la société et démocratie dans les organisations ? Une première façon de répondre à cette question nous est apportée par Jardat. Il avance l'idée que l'on ne pourrait pas, en tant que citoyen, être baigné de façon continue dans « l'idée » de démocratie, et accepter de passer la moitié de sa vie dans des organisations qui ne sont pas démocratiques (Jardat 2012). La valorisation dans la société de la discussion, de l'émancipation, de l'équité, du refus des abus de pouvoir ne peut pas selon lui rester à la porte des entreprises, des associations et des collectivités. Le premier élément de réponse que l'on peut apporter est donc que, de fait, la démocratie dans la société a une influence sur notre perception de la démocratie dans les organisations. Les deux sont inséparables car elles concernent les mêmes acteurs, qui ne sauraient différencier totalement les grands principes auxquels ils se conforment selon leurs « moments de vie ».

D'autres auteurs, tels que Caudron, vont plus loin en affirmant que « *la réalisation de la démocratie au travail est une condition sine qua none de réalisation de la démocratie en général* » (Caudron 2008). Il tempère cela dit son propos en montrant, au travers de l'étude qu'il a menée, que l'évaluation de cette affirmation en situation réelle amène à des conclusions parfaitement contradictoires. L'on peut donc souhaiter, dans une visée humaniste, que se vérifie ce lien entre démocratie dans la société et démocratie dans l'organisation, mais on ne peut le considérer comme réellement validé.

Le troisième aspect nous permettant de traiter cette question du lien entre démocratie dans la société et dans les organisations attaque le sujet selon un autre angle. Cet angle est celui de « *l'entreprisisation du monde* » (Jardat 2019) qui peut être définie comme l'intrusion constante de l'organisationnel dans la vie des citoyens. Ce phénomène nouveau, qui vient effacer les anciennes structures communautaires pose une question simple : la démocratie pourra-t-elle survivre à une entreprisisation du monde si les organisations ne deviennent pas démocratiques ? Et la réponse nous semble évidemment « non ». Si nos sociétés sont de plus en plus organisationnelles, alors il semble que pour que la démocratie continue d'y être un principe fondateur, elle doit s'insérer dans tout ce qui compose les sociétés, y compris les organisations.

1.1.3. La démocratie dans les organisations : grands principes

Maintenant que nous avons défini ce que serait la démocratie dans la société, et quels sont les liens de cette dernière avec la démocratie dans les organisations, nous tâchons de décrire les grands principes de la démocratie dans les organisations. Pour cela, et de façon méthodique, nous reprenons

chacune des caractéristiques de la démocratie dans la société pour comprendre de quelle façon elle pourrait se traduire dans les organisations (cf. Figure 3).

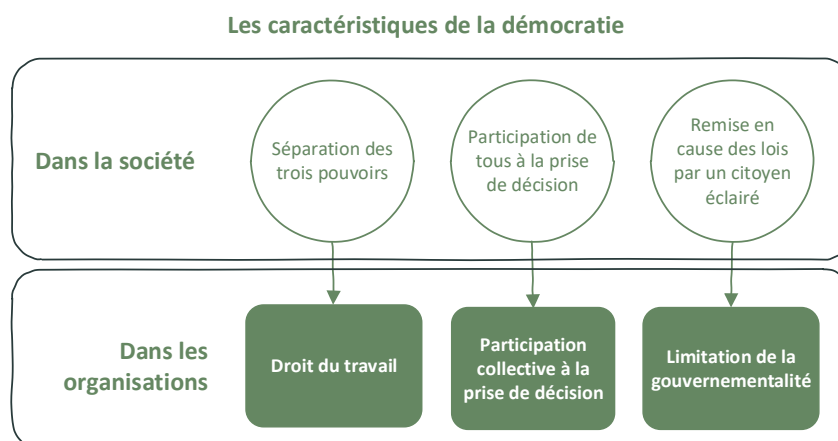


Figure 3 : Les caractéristiques de la démocratie dans les organisations

La séparation des trois pouvoirs est la première des caractéristiques de la démocratie, et ne saurait être appliquée de manière simpliste aux organisations. Elle nécessite d'avoir une vision méta-organisationnelle, qui considérerait l'Etat comme garant de cette séparation via le droit du travail. Ce sujet ne sera pas traité dans le cadre de cette thèse, il ne nous semble donc pas opportun de nous arrêter sur cette première caractéristique.

La deuxième caractéristique est la participation de tous à la prise de décision. Certaines formes d'organisation, notamment celles de l'Economie Sociale et Solidaire prônent une « *gouvernance collective* » qui encouragerait cette prise de décision collective (Boissin et al. 2015). Pour autant, il convient d'être prudent avec cette notion de participation qui, bien qu'étant un outil au service du processus de démocratisation, n'en est pas pour autant la garantie. En effet, les structures plus « classiques » tentent parfois de promouvoir un management démocratique en déléguant la prise de décision. Cette délégation de la prise de décision au collectif s'arrête souvent au niveau opérationnel ou tactique, et ne se porte que rarement sur les décisions stratégiques. On est limité ici à ce que l'on pourrait appeler une « *apparence de démocratie* » (Salais 2010) au sens où elle « *limite les pouvoirs du souverain* » (Brière et Le Texier 2018) mais ne permet pas l'élaboration de véritables propositions stratégiques collectives.

La troisième caractéristique qu'est le droit à la remise en cause des lois par le citoyen éclairé nous entraîne sur une autre question majeure : celle de la gouvernamentalité. Pour commencer, établissons une distinction claire entre gouvernance, gouvernement et gouvernamentalité. La gouvernance qui se définit comme l'exercice du pouvoir souverain du collectif est exercée dans les organisations privées par les actionnaires, qui constituent l'Assemblée générale de l'entreprise. Le gouvernement, qui exécute les décisions générales relève des compétences de la direction générale. La gouvernamentalité quant à elle désigne « *les modes par lesquels on entend au quotidien obtenir d'une population les comportements attendus* » (Jardat 2012). En d'autres termes, la gouvernamentalité influence et contraint sans pour autant s'afficher au travers de décisions élicitées. Cette gouvernamentalité peut donc s'exercer sans gouvernance et sans gouvernement. C'est ici que nous pouvons établir un lien entre la gouvernamentalité et la caractéristique que nous traitons dans ce paragraphe : le droit à la remise en cause des lois par le citoyen éclairé. Si cette gouvernamentalité « avance masquée », il devient alors impossible de la remettre en cause. La démocratie dans les organisations doit par conséquent se caractériser par une limitation de cette gouvernamentalité.

Nous avons proposé dans cette première partie notre vision de la démocratie, en nous appuyant sur les définitions de philosophes contemporains. Après avoir exploré le lien entre démocratie dans la société et démocratie dans les organisations, nous avons pu dresser une liste des caractéristiques de la démocratie des organisations : respect du droit du travail pour une séparation des trois pouvoirs, participation collective à la prise de décision et limitation de la gouvernementalité.

La démocratie des organisations ne saurait être garantie sans le soutien des outils supports à ces organisations, parmi lesquels le système d'information tient un rôle central. C'est cette relation tenue entre démocratie de l'organisation et système d'information que nous explorons dans la section suivante.

1.2. Les systèmes d'information et la démocratie dans les organisations

Le système d'information ne reflète pas la réalité, il la construit. Dans une perspective constructiviste, telle que celle de Simon (Simon 2004), nous considérons le système d'information comme le fruit de choix effectués par ceux qui le construisent, et par ceux qui décident de sa construction. Loin de l'apparente neutralité qu'on leur prête, les systèmes d'informations des organisations sont selon nous de véritables systèmes de représentations, qui impactent directement la façon dont l'entreprise se voit, et dont elle voit son environnement. Le système d'information peut être analysé d'un point de vue purement technique, ou d'un point de vue systémique, beaucoup plus large et influencé par l'épistémologie. Nous avons choisi d'adopter ce second point de vue pour présenter notre vision du système d'information et, afin de justifier notre approche constructiviste, nous présentons également une analyse de la genèse du système d'information dans l'organisation. Ces éléments nous permettent de conclure cette partie par les caractéristiques que devraient selon nous respecter un système d'information pour soutenir la démocratie des organisations telle que nous l'avons définie dans la section précédente (cf. Figure 4).

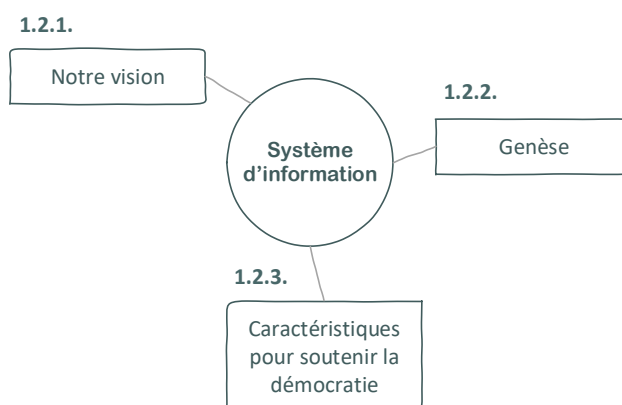


Figure 4 : Plan de la partie 1.2

1.2.1. Notre vision du système d'information

Jean-Louis Le Moigne définit le système d'information d'une organisation comme « l'ensemble de symboles signifiants, formels ou informels, circulant dans [celle-ci] » (J. Le Moigne 1973). Nous partageons cette vision du système d'information comme un système de représentations, de

« formalisations de visions du monde » comme l'écrit Maryse Salles (Salles et Colletis 2007). Cette définition porte avec elle une idée majeure selon nous : le système d'information est un construit. Et c'est un construit basé sur les représentations de celui qui construit. Ainsi, et contrairement à l'image souvent véhiculée, le système d'information n'est pas une photo, un reflet de la réalité mais bien un ensemble de représentations choisies.

La place du système d'information dans l'organisation est centrale. Il est à la croisée du système opérant et du système de pilotage (cf. Figure 5). Il occupe donc plusieurs fonctions, dont une première : « exprimer les représentations partagées indispensables à la réalisation de [...] la production au sens large » (J. L. Le Moigne 1994). En s'imposant comme le « langage » de l'organisation, le système d'information joue en effet un rôle irremplaçable et transverse sur tout le processus de production. Le second rôle du système d'information : « produire les représentations du système et de son environnement nécessaire au pilotage » (J. L. Le Moigne 1994) est également crucial, puisqu'il est en lien direct avec la prise de décision du haut management de l'organisation.

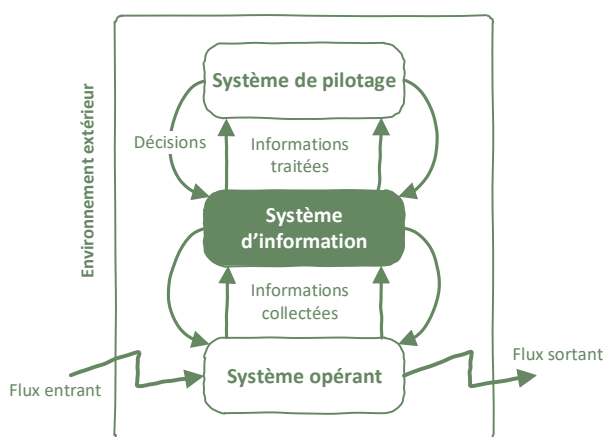


Figure 5 : La matrice structurelle du système général (J. L. Le Moigne 1994)

Il est maintenant pour nous nécessaire de faire une distinction claire entre système d'information et système d'information numérique (ou digital). Si l'on s'accorde à dire que le système d'information est le langage de l'organisation, alors le système d'information numérique en est la partie encodée, la partie informatisée. Tout ne pouvant être encodé, le système d'information numérique est une sous partie du système d'information. Comme Salles le décrit, la partie informatisée du système d'information a pour but de « rendre ces formalisations opérationnelles au travers de codifications particulières » (Salles et Colletis 2007).

1.2.2. Genèse du système d'information numérique

Le système d'information numérique est un construit. Il est le fruit d'un système de filtrages successifs, entre réalité et système d'information, puis entre système d'information et système d'information numérique (Bour, Salles, et Paraponaris 2019). C'est ce système de filtrage que nous présentons de manière macroscopique dans la figure suivante (cf. Figure 6). L'objectif de ce schéma et de sa décomposition est de montrer que les systèmes d'information numériques des organisations sont des dispositifs politiques (Rambaud 2018), au sens où ils ne codent pas de façon neutre une réalité qui leur

serait extérieure, mais au contraire sont porteurs de visions du monde, de croyances, de valeurs, et participent à construire la réalité dans laquelle fonctionnera l'organisation.

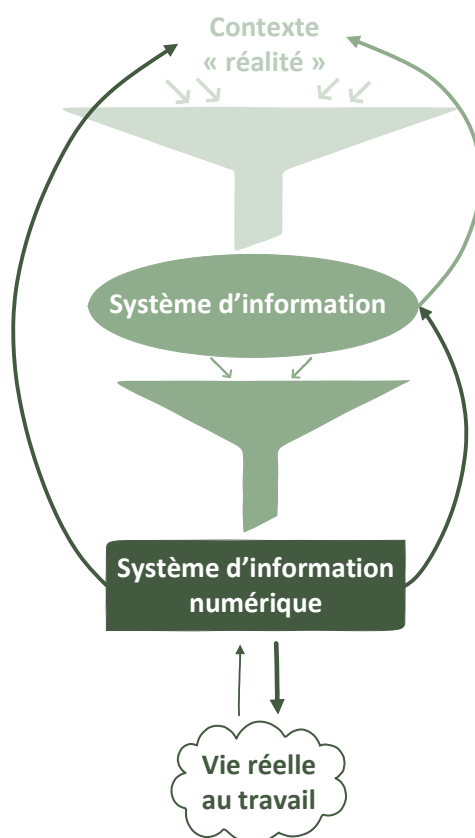


Figure 6 : Genèse du système d'information numérique

La couleur de plus en plus foncée des différents éléments du schéma correspond, à mesure que l'on va vers le système d'information numérique :

- À un accroissement de la normativité,
- À une visibilité et un caractère formel croissants (la partie la plus haute étant la plus implicite et la moins formelle).

Les flèches qui remontent depuis le système d'information et depuis le système d'information numérique expriment les effets de rétroactions qui sont évoqués par la suite. Dans les sections suivantes nous présentons les différents filtres qui permettent d'une part de simplifier (en ne retenant que certains éléments), d'autre part de façonner la réalité, afin de la représenter finalement dans le système d'information numérique. La première sous-section est consacrée au premier de ces filtres, qui permet de constituer le système d'information de l'organisation. Le passage du système d'information au système d'information numérique, au travers d'un second filtre, fait l'objet de la deuxième sous-section. Enfin, l'impact du système d'information numérique sur la vie au travail est évoqué dans la troisième et dernière sous-section.

1.2.2.1. Le passage de la réalité au système d'information

Le système d'information n'est pas un reflet de la réalité. D'une part car le système d'information ne prend en compte qu'une partie de la réalité (système d'entonnoir), d'autre part car les éléments provenant de la réalité passent par le filtre des visions du monde, des croyances, des idéologies ou encore des représentations (cf. Figure 7).

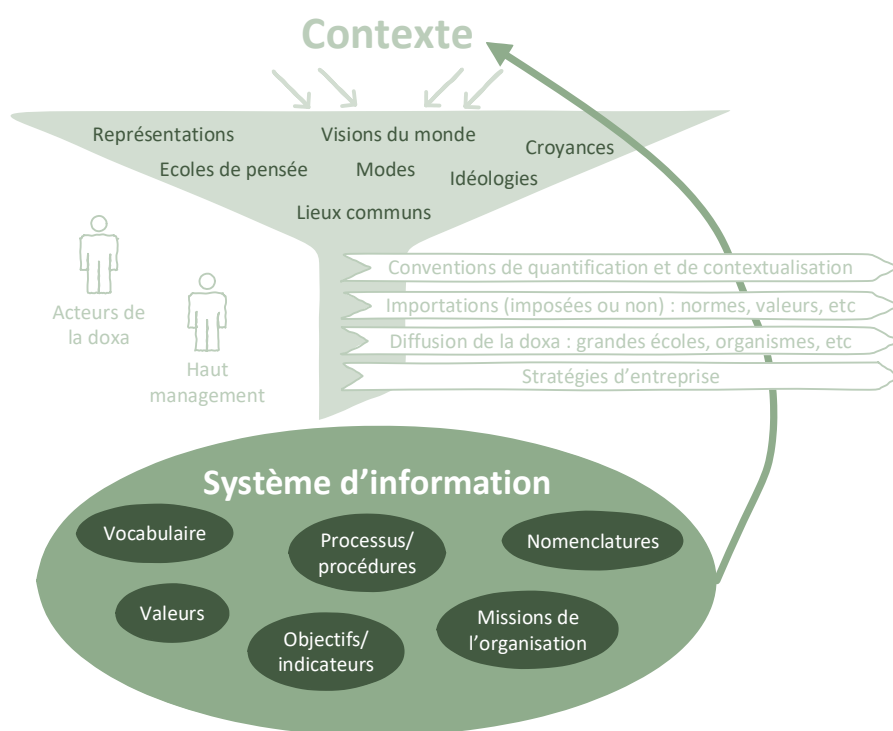


Figure 7 : Passage de la réalité au système d'information

Ces représentations, visions du monde, idéologies, lieux communs peuvent être catégorisés de deux façons. D'une part, les concepts qui sont portés par la société dans laquelle évolue l'organisation, et donc par les « *acteurs de la doxa* » (Salles et Colletis 2013). Ces derniers promeuvent des visions de ce que doit être la performance économique, la mission de l'entreprise ou encore et plus généralement la prospérité d'un pays par exemple. On pourrait nommer entre autres acteurs de la doxa² des institutions telles que le FMI, l'OMC, l'OCDE, etc. On compte également parmi eux des organismes à l'origine de la production de standards de qualité par exemple, ou encore de normes comptables auxquelles doit se conformer toute organisation. Tous ces standards et conventions ne sont donc pas seulement le fruit d'un choix de l'organisation elle-même mais bien d'une standardisation établie dans nos sociétés.

D'autre part on retrouve les visions et représentations portées par ce que l'on nomme le haut management de l'organisation. Il s'agit par exemple ici de la vision de la mission de l'organisation portée par son directeur. Ces visions du haut management de l'organisation se diffusent au travers de conventions, c'est-à-dire de règles, systèmes de mesure, que l'organisation choisit d'adopter et qui ont un impact sur la façon dont les données sont considérées. Pour citer Fieux, on peut dire que « *quantifier, c'est déjà décider* » (Fieux 2018). Pour illustrer cela, on peut évoquer le cas des nomenclatures de tâches dans les hôpitaux, qui portent une vision de ce que doit être le travail du personnel hospitalier. La nomenclature des tâches des infirmiers notamment ne mentionne pas la tâche « discuter avec la famille d'un patient ». Celle-ci n'est donc pas une tâche comptabilisée dans le

² Le terme de doxa renvoie aux « *visions du monde sous-jacentes aux principes et aux normes* » selon la définition de Maryse Salles à laquelle nous nous référons (Salles et Colletis 2013)

temps de travail de ces derniers, et par extension ne fait pas partie du travail « visible » des infirmiers. On voit ici que le choix d'une convention est bel et bien porteur d'une certaine vision de l'hôpital.

Le système d'information va ensuite s'exprimer concrètement par des entités reconnues comme ayant un sens pour l'organisation (clients, usagers, salariés, fournisseurs...), des catégories (types d'actes médicaux, fonctions, métiers...), des nomenclatures (de tâches, de produits...), des indicateurs (de performance, de coût, de qualité...), des processus de travail, un lexique (incluant les connotations des termes employés), etc. Tous ces éléments du système d'information font de celui-ci le langage de l'organisation, au sens où il est le prisme par lequel l'organisation se voit, et par lequel elle voit son environnement.

A partir de ce que nous avons décrit, on peut catégoriser de trois façons les liens qui unissent réalité de l'organisation et système d'information :

- Le système d'information est performatif. On entend par là qu'il fait exister dans la réalité ce qu'il prend en compte, ce qu'il représente. Ainsi, lorsqu'une nouvelle catégorie est créée dans le système d'information, cela la fait exister dans la réalité. Citons à titre d'exemple la catégorie des chercheurs-publiant qui a été créée récemment par l'Agence Nationale de la Recherche pour des besoins d'évaluation. Elle fait exister dans la réalité cette nouvelle catégorie, induisant des modifications dans le comportement des chercheurs comme le décrit Aggeri (Aggeri 2016).
- En miroir de cette performativité, on peut également affirmer que le système d'information construit un système d'invisibilité. En d'autres termes : ce qui n'existe pas dans le système d'information, n'existe pas dans la réalité de l'organisation. Pour citer l'exemple de la comptabilité sociale environnementale (Espana, Bik, et Overbeeke 2019), une organisation qui choisirait de ne pas intégrer ces représentations à son système d'information ne prendrait donc pas du tout en compte cette question dans sa réalité (Bachet 2019). En effet, comment mettre en place des actions, définir des objectifs, sur un sujet qui n'est pas mesurable, pas explicité ?
- Enfin, le système d'information et l'organisation sont consubstantiels. En paraphrasant Le Moigne, nous dirons que « le système d'information forme l'organisation qui le forme » (J.-L. Le Moigne 1994). Cette caractéristique est le résultat direct des deux caractéristiques définies plus haut.

Les impacts de ce passage de la réalité au système d'information sur la démocratie dans l'organisation sont multiples. D'une part, le caractère implicite des doxas, représentations, croyances, ainsi que les multiples renforcements et diffusions dont elles font l'objet les rendent très difficiles à identifier ou révéler, et, a fortiori, à remettre en cause. Or, nous avons vu que leur rôle était déterminant dans la formation du système d'information. D'autre part, au niveau du système d'information lui-même, les valeurs qu'il intègre ne sont pas vraiment discutées. Les définitions de la performance ou de la qualité par exemple, sont ainsi considérées comme « allant de soi », ne prêtant pas à discussion. Il en est de même pour la définition de la mission de l'entreprise (sa raison d'être). Enfin, le lexique employé dans l'organisation participe puissamment à renforcer les visions dominantes tout en les masquant : les mots sont en effet porteurs de visions du monde (Klinkenberg 2015). Pour donner un seul exemple, l'expression « résistance au changement », systématiquement connotée négativement, porte la vision d'un changement bon par essence, et donc non discutable.

Nous avons vu dans la section 1.1.3. que la gouvernementalité était un obstacle à la démocratie, et que le système d'information devait être un moyen de la limiter. Pour cela, ces visions,

représentations, et leurs traductions dans le lexique, les processus du système d'information de l'organisation doivent pouvoir être discutés, et remis en cause.

1.2.2.2. Le passage du système d'information au système d'information numérique

Si l'on s'accorde à dire que le système d'information est le langage de l'organisation, alors le système d'information numérique en est la partie informatisée. Tout ne pouvant être encodé, comme nous le montrons dans les paragraphes suivants, le système d'information numérique est une sous-partie du système d'information. Cette sous-partie a ses caractéristiques propres, dont une normativité et une performativité renforcées qui ont pour effet de réduire la complexité du système d'information, et de rigidifier ce qu'il encode. Entre autres exemples, on pourrait évoquer le fait que la complexité d'un processus de travail ne saurait être entièrement « programmée », ou bien encore que la structure d'une base de données ne saurait être facilement modifiée. La Figure 8 présente le passage du système d'information au système d'information numérique avec ses filtres, ses acteurs et ses processus de filtrage.

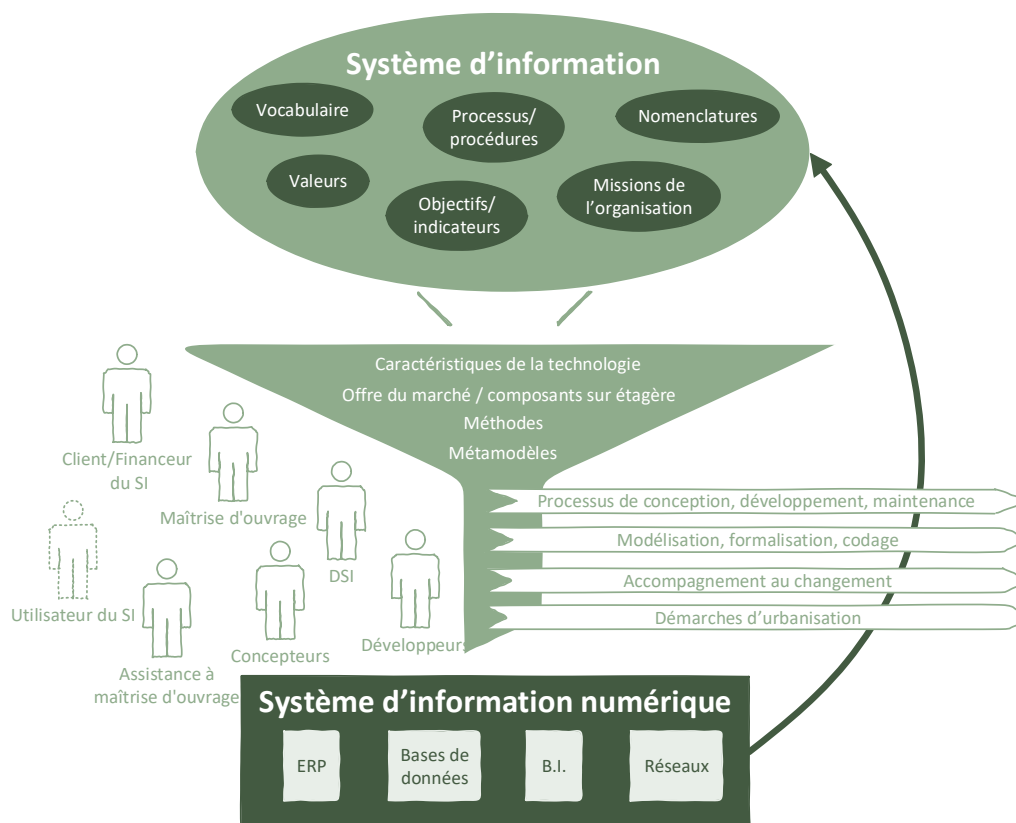


Figure 8 : Passage du système d'information au système d'information numérique

Le passage du système d'information à un système d'information numérique implique un nombre important d'acteurs, que l'on retrouve habituellement à la table des projets informatiques. Parmi eux, les donneurs d'ordres : clients et/ou financeurs du système d'information, qui décident de l'informatisation d'une partie du système d'information. Les motivations de ces derniers sont souvent exprimées au travers des termes de « modernisation, simplification, rationalisation, hausse de l'efficacité ». On retrouve également parmi les acteurs la maîtrise d'ouvrage, c'est-à-dire les porteurs de la vision métier, souvent soutenus par une assistance à maîtrise d'ouvrage (AMO). L'AMO est un service proposé en interne, ou une compétence importée via des consultants externes à l'organisation,

qui participe à l'expression des besoins, et au processus d'accompagnement au changement. Les concepteurs et développeurs constituent la maîtrise d'œuvre du projet informatique, et participent au processus de filtrage en appliquant des méthodes de conception/développement souvent très formalisées (nous évoquons ce point dans la section suivante). Tous ces acteurs, « experts » dans leur rôle sont, inconsciemment le plus souvent, impliqués dans des choix moraux au sujet des normes et des relations de pouvoir qu'ils implémentent dans le système d'information numérique (Walsham 2012).

Quant aux utilisateurs finaux, ils sont plus ou moins impliqués dans le passage du système d'information au système d'information numérique. Selon les choix des clients/financeurs, selon la relation des utilisateurs avec la maîtrise d'ouvrage et enfin selon les démarches de conception adoptées, ces utilisateurs finaux peuvent être totalement écartés du projet informatique, ou en être les co-constructeurs.

On peut distinguer quatre composantes du filtrage permettant de passer du système d'information à sa version numérisée. Dans un premier temps nous identifions les caractéristiques de la technologie. Ce filtre est incarné par les choix techniques qui sont faits : types de bases de données, langage informatique, architecture des composants, et ces choix sont loin d'être neutres. Une base de données relationnelle par exemple imposera une certaine catégorisation des données, qui ne sera pas la même qu'avec une base de données NoSQL ou sans schéma. De la même façon, le choix d'un langage de programmation objet viendra influencer la formalisation des objets de la réalité de façon prononcée. Le choix d'une architecture client-serveur ou d'une architecture web pour se conformer au schéma d'urbanisation du système d'information aura lui aussi un impact sur ce qu'il est possible d'attendre du système d'information numérique. Tous ces choix technologiques présentés le plus souvent comme neutres sont un véritable moyen de façonner le système d'information numérique.

La deuxième composante du filtrage que nous avons identifiée est l'offre du marché, dont les composants sur étagère. Le choix d'un ERP, ou d'un progiciel a lui aussi un impact très fort. En effet, ces logiciels sur étagères « *exogènes à l'organisation* » correspondent à des « *représentations du monde construites à l'extérieur* » de l'organisation (Salles et Colletis 2007). On comprend ici qu'il n'est pas anodin d'importer le progiciel depuis un grand compte vers un sous-traitant, ou bien encore d'importer un logiciel sur étagère adapté à des processus de gestion d'organisations privées dans une collectivité publique.

Dans un dernier temps, deux composantes du filtrage peuvent être liées : les méthodes de conception/développement et les métamodèles. En effet, ces métamodèles constituent une partie de l'outillage des méthodes. Lors d'un projet informatique, la mission des concepteurs consiste en une traduction des besoins exprimés dans un langage métier en modèles implémentables par des développeurs. Les métamodèles choisis sont alors autant de possibilités de codifier le système d'information : diagrammes de classe, schéma entité-association, processus représentés avec la notation Business Process Modeling (BPMn), etc. Cependant, tout ce qui est présent dans le système d'information ne saurait être modélisé. A cette étape, l'effet d'entonnoir exclut tout élément du système d'information qui ne saurait être représenté avec ces types de modélisation (Bour 2017). Le risque consiste alors, par exemple, à faire peser des décisions sur les seuls chiffres produits par le système d'information numérique, en ignorant que le système d'information numérique ne donne à réfléchir que sur ce qui est quantifiable.

Les méthodes de conception quant à elles sont très dépendantes des courants managériaux « en vogue ». C'est ainsi qu'après avoir exclu les utilisateurs du processus de conception, des formes telles que les méthodes participatives (ou centrées utilisateurs plus largement) et les méthodes agiles

proposent de réintégrer l'utilisateur final au processus. Les raisons qui motivent ces choix sont diverses. Elles vont de l'amélioration de l'utilisabilité du système, à une meilleure qualité de celui-ci en passant par une diminution de coûts liés aux phases de « correction » de systèmes d'information numériques qui ne correspondraient pas réellement aux besoins métier. Nous évoquons ces points en détail dans le chapitre 2 de cette thèse.

Le processus d'encodage aboutissant au système d'information numérique est une phase d'accentuation de la normativité et de la performativité du système d'information. En effet, la rigidification liée à l'inscription « dans le code » des objets, processus, indicateurs contribue à figer les choix amonts. La question de la dispute démocratique peut se poser ici : comment discuter les représentations, les normes, les valeurs lorsqu'elles sont informatisées, qui plus est lorsque l'on ne maîtrise pas la technique mise en œuvre ?

De plus, et alors que les méthodes de conception pourraient intégrer dans leur processus des possibilités de dispute démocratique, elles écartent l'utilisateur, ou l'intègrent en formalisant au préalable le cadre très restreint dans lequel il pourra s'exprimer. Ces utilisateurs, qui sont ceux qui sont directement impactés par la mise en place du système n'ont pas la possibilité de rediscuter les « lois », c'est-à-dire les normes qui leur sont imposées, et en cela le système d'information numérique peut constituer une entrave à la démocratie.

1.2.2.3. L'usage des systèmes d'information

Dans cette dernière partie, nous évoquons quelques aspects de l'influence des systèmes d'information numériques sur la vie de l'organisation et de ceux qui y travaillent. Comme illustré par les deux flèches de la Figure 9 ci-dessous, on notera que l'influence qu'exerce le système d'information numérique sur la vie au travail est beaucoup plus puissante que celle que les travailleurs³ peuvent en retour exercer sur lui.

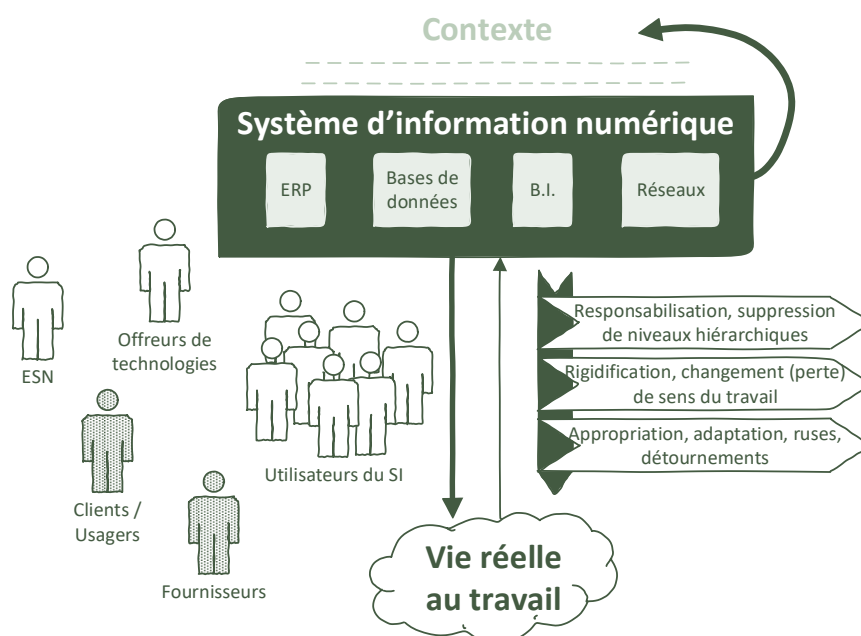


Figure 9 : Du système d'information numérique à la vie au travail dans l'organisation

³ Bien qu'il soit assez fortement connoté, nous utilisons ici le terme de travailleur à dessein. Il existe en effet aujourd'hui un grand nombre de personnes en situation de subordination au travail et qui ne sont pourtant pas

Le système d'information numérique d'une organisation concerne un grand nombre d'utilisateurs internes : exécutants, encadrement, haut management. Mais il s'impose aussi à des acteurs qui n'appartiennent pas à l'organisation au sens strict : fournisseurs (lorsque ce sont des grands comptes, ils imposent souvent l'utilisation de leur système d'information numérique aux fournisseurs et aux sous-traitants), mais aussi clients et usagers, qui sont « mis au travail » (Tiffon 2013) en particulier au travers du e-commerce.

Les systèmes d'information numériques ont entraîné des mutations dans le travail et dans le management. Nous ne citons ici que quelques-unes des conséquences de ces mutations. De par les caractéristiques de la technologie évoquées dans la partie précédente, et en particulier la rigidité de certains de leurs composants, les systèmes d'information numériques produisent un renforcement des normes. Une conséquence en est que les travailleurs ont de plus en plus de difficultés à s'éloigner de la tâche prescrite pour adapter cette tâche à leurs contraintes afin de parvenir à remplir les objectifs fixés. On pourrait avancer l'idée que la bureaucratie, cette rigidification à outrance des procédures, n'a pas attendu les systèmes d'information numériques pour exister. C'est vrai, mais ce serait ignorer le fait que la digitalisation vient renforcer ce phénomène en rendant tout ce qu'elle encode encore plus intangible, renforçant ainsi ce que l'on pourrait appeler une « *société de la norme* » (Gori et Del Volgo 2009).

Une autre spécificité forte des systèmes d'information numériques est qu'ils représentent dans le même temps, pour beaucoup de travailleurs, l'outil de travail et l'outil de contrôle. C'est au travers de leur activité supportée par le système d'information numérique qu'ils génèrent les données qui seront utilisées pour les encadrer et les évaluer. L'extension du domaine des systèmes d'information numériques dans l'évaluation des personnes a accompagné une domination du quantitatif (l'évaluation devenant d'ailleurs souvent synonyme de notation), mais aussi du principe d'une évaluation individualisée qui ne tient pas compte des collectifs. Ces derniers sont d'ailleurs mis à mal par beaucoup de dispositifs numériques qui isolent les travailleurs, un cas emblématique étant celui de la commande vocale dans les entrepôts de la grande distribution (Gaborieau 2012).

Si les systèmes d'information numériques produisent un durcissement des normes de travail, ils sont aussi l'objet d'adaptations diverses. Certains travailleurs vont ainsi chercher à « tromper » le système ou à le contourner (Leclercq-Vandelannoitte 2017). Dans ce dernier cas, ils mettent en place des processus de réappropriation, en réponse à la non-conformité des outils à leurs besoins réels, ou à leur caractère trop contraignant. Il s'agit de pratiques largement analysées par les ergonomes, et consistant pour l'utilisateur à s'éloigner de plus en plus de la tâche prescrite, pour adopter ses propres pratiques, regroupées sous le terme de tâche réelle (Vilela et al. 2015). Ils peuvent aussi se rapprocher plus du phénomène que les organisations nomment « Shadow IT ⁴».

Nous insistons ici sur un mode particulier d'adaptation, qui n'est pas propre aux systèmes d'information numériques, mais que ces derniers ont fait se multiplier : les effets de rétroaction. Ces effets sont l'une des expressions les plus visibles de la consubstantialité entre système d'information et réalité de l'organisation. La nomenclature des actes médicaux, correspondant à des tarifs différenciés de remboursement par l'assurance maladie, a pu entraîner, dans certains services de santé, une sur-prescription d'actes coûteux et un délaissement d'activités moins bien tarifées. De

salariées. Pour ne citer qu'eux, c'est le cas par exemple des travailleurs des plateformes (UBER notamment) (Palterini et Prévost-Carpentier 2019)

⁴ Le Shadow IT est défini comme l'ensemble des applications développées par les utilisateurs eux-mêmes, hors du périmètre du système d'information numérique défini par la Direction des Systèmes d'Information (DSI) de l'organisation (Rentrop et Zimmermann 2012).

même, la généralisation du management (et de l'évaluation) basé sur les indicateurs a produit, dans un grand nombre de cas, une recherche de l'amélioration de l'indicateur. Celui-ci prend alors la place de l'objectif, au risque parfois de compromettre l'atteinte de ce dernier.

L'impossibilité de discuter des valeurs et des normes qui sont intégrées dans les systèmes d'information numériques est une entrave manifeste à la démocratie dans les organisations. Cette entrave à toute dispute démocratique accentue les conflits de valeurs que vivent certaines personnes dans l'accomplissement de leurs tâches, et par la suite, aggrave la souffrance au travail qui en découle. Enfin, nous avons dit précédemment que la gouvernementalité représentait un mode a-démocratique de gouvernement et devait être très fortement limitée. Or les systèmes d'information, et singulièrement leur partie numérisée, apparaissent comme des instruments privilégiés d'une gouvernementalité sans contrôle, qui peut usurper la totalité du gouvernement des organisations.

1.2.3. Les caractéristiques d'un système d'information soutenant la démocratie

Notre description de la genèse des systèmes d'information numériques et de l'impact de ces derniers sur la démocratie dans les organisations nous engage maintenant à faire des propositions. Nous nous attachons donc dans cette partie à définir ce que seraient pour nous les caractéristiques d'un système d'information soutenant la démocratie des organisations.

Tout d'abord, notons que cette question de la démocratie dans les systèmes d'information est très peu abordée dans les articles traitant de l'éthique informatique, ou apparaît souvent comme secondaire. Parmi les auteurs qui s'intéressent à cette question, Floridi mentionne la démocratie comme l'un des sujets de l'éthique informatique, aux côtés du respect de la vie privée ou encore de la transparence (Floridi 2010). Maryse Salles a traité ce sujet de l'éthique informatique, et de la démocratie en particulier en la définissant comme « *la garantie de l'accès à des visions du monde plurielles* » (Salles 2015). Nous souhaitons aujourd'hui approfondir ce sujet en dressant une liste de caractéristiques que devrait selon nous avoir un système d'information pour pouvoir soutenir la démocratie dans les organisations. Nous abordons cette question selon deux angles : le premier est celui du processus de création du système d'information, le second est celui du système d'information lui-même (cf. Figure 10).

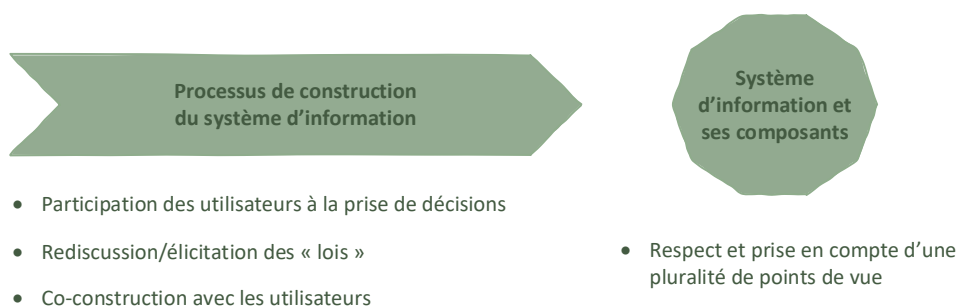


Figure 10 : Caractéristiques du système d'information soutenant la démocratie dans l'organisation

Si l'on se rapporte aux caractéristiques de la démocratie dans les organisations que nous avons traitées dans section 1.1.3., l'on peut s'intéresser dans un premier temps à la participation du collectif à la prise de décision. Du point de vue du processus de construction de système d'information, cela se rapporte au fait de permettre aux futurs utilisateurs de participer aux décisions ayant trait au système d'information qui les concerne (Clement 1994). Cela sous-entend qu'ils doivent pouvoir être présents durant le processus de construction, avec un rôle actif, et pas simplement informationnel. La problématique du nombre d'utilisateurs à impliquer dans un projet informatique de grande ampleur peut être résolue par la participation de représentants élus.

Dans un second temps, la construction du système d'information doit pouvoir faire l'objet d'une re-discussion des lois, ou à minima d'une élicitation de celles-ci. En effet, si l'on considère que le système d'information peut devenir l'un des instruments de la gouvernabilité des organisations, au travers des dispositifs de gestion implantés par des « *technologies invisibles* » (Maugeri 2007), on peut aussi envisager son contraire. Le système d'information pourrait être l'instrument permettant de discuter les normes, les règles, les standards imposés, et devenir ainsi un soutien à la démocratie dans les organisations. La démocratie ne pouvant vivre « *sans débat sur les représentations, les visions du monde* » (Salles et Colletis 2013), chaque nouveau projet informatique pourrait faire l'objet d'une remise à plat par le haut management, auprès de tous les futurs utilisateurs, de ces représentations et visions du monde qui s'imposeront à eux au travers du système d'information.

Dans un troisième temps, et toujours du point de vue du processus de construction du système d'information, la démocratie pourrait être soutenue par le renforcement d'un « collectif éclairé ». Pour cela, il s'agit ici de ne pas rendre opaques ou incompréhensibles les modèles réalisés par les concepteurs. En effet, si l'ensemble des futurs utilisateurs doit pouvoir comprendre comment le système d'information s'est construit, alors les choix des concepteurs de système d'information doivent se faire le plus possible en accord avec ceux-ci. Il faut pour cela limiter au maximum la prise de décision par les concepteurs, sans validation ou recours des utilisateurs, mais également s'assurer que l'on produit des modèles intelligibles.

Dans un dernier temps, et du point de vue du système d'information lui-même cette fois-ci, le respect d'une pluralité de points de vue semble être la dernière caractéristique que devrait avoir un système d'information pour soutenir la démocratie dans les organisations (Salles 2015). Pour cela, il convient d'empêcher que le système d'information ne soit le reflet d'un point de vue dominant seulement. Les différents points de vue exprimés, pris en compte lors du processus de conception doivent se traduire en points de vue dans les logiciels, bases de données développés. Cela impose que les outils soient capables d'être à la fois partagés entre plusieurs points de vue, et à la fois permettent à chaque utilisateur d'en avoir un usage adapté à son point de vue singulier.

Nous avons partagé dans cette deuxième partie notre vision du système d'information, dans une perspective constructiviste. Nous avons décrit notre point de vue sur la genèse des systèmes d'informations numériques, en nous appuyant sur un schéma présentant les filtrages successifs permettant de passer de la réalité d'une organisation à son système d'information numérique, et à l'usage de celui-ci par les travailleurs. Nous avons ensuite proposé des caractéristiques que devraient selon nous respecter un système d'information pour soutenir la démocratie dans les organisations : participation des utilisateurs à la prise de décision durant le processus de construction du système d'information, re-discussion et élicitation des « lois » régissant le système, co-construction avec les utilisateurs et enfin, respect et prise en compte d'une pluralité de points de vue dans le système d'information lui-même.

Nous nous intéressons à ce lien entre système d'information et démocratie car cette question est pour nous, philosophiquement et idéologiquement extrêmement importante. Cela étant, nos propositions sont aussi et surtout une réponse aux enjeux que doivent aujourd'hui adresser les organisations quant à leurs systèmes d'information. Ce sont ces enjeux que nous présentons dans cette troisième section.

1.3. Les enjeux à créer des systèmes d'information qui soutiennent la démocratie

Comme nous l'avons démontré dans les sections précédentes, il est envisageable de développer des systèmes d'information qui soutiennent la démocratie des organisations. Outre le fait de promouvoir cette démocratie dans les organisations, et dans la société en général, les enjeux à développer de tels systèmes sont nombreux. D'une part ils peuvent être un moyen de diminuer le recours au Shadow IT⁵ dans les organisations. Cette pratique, consistant pour les utilisateurs à développer leurs propres outils informatiques en contournant le circuit « officiel » de la Direction des Systèmes d'Information de l'organisation, ne saurait être totalement bannie. Cela dit, il est possible d'en diminuer les effets néfastes en proposant des systèmes d'information démocratiques, c'est ce que nous montrons dans une première section. D'autre part, l'un des enjeux majeurs des organisations est l'innovation, et nous pensons que les systèmes d'information conçus de manière démocratique peuvent être un levier de créativité et d'émergence d'idées nouvelles. Nous parlons de ce point dans la deuxième section. Enfin, la capitalisation de la connaissance a une véritable valeur pour les organisations aujourd'hui (Grundstein, Rosenthal-Sabroux et Arduin 2013), qui pratiquent d'ailleurs le *knowledge management* (management des connaissances). En nous basant sur la théorie des Communs, nous montrons comment un système d'information démocratique peut permettre de soutenir cette pratique (cf. Figure 11).

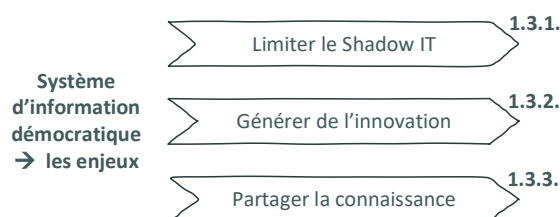


Figure 11 : Plan de la partie 1.3

1.3.1. Gérer le shadow IT

Le Shadow IT peut être défini de la façon suivante : ensemble des outils informatiques développés dans les unités métier de l'organisation, en parallèle des outils informatiques « officiels ». Ils sont en général non connus, et donc non maintenus par la DSI de l'organisation (Rentrop et Zimmermann 2012). Ce phénomène a émergé dans les années 80 mais s'est généralisé dans les années 2000, porté par l'accès de plus en plus facilité aux outils web pour chacun dans l'organisation : outils proposés par Google, tutoriels, Machine Learning, etc. Cette pratique est en partie adoptée par une génération nouvelle dans l'organisation, qui a pour habitude dans sa vie quotidienne de manipuler l'informatique. Les outils développés en Shadow IT sont de trois types : les solutions de communication, les applications faites maison type « Excel » ou « Access » et les solutions de Business Intelligence (BI) et de reporting.

La question du Shadow IT est un véritable sujet de préoccupations dans les DSI actuellement. En effet, cette pratique est très largement répandue : selon une étude menée en 2014, 32% de l'activité « logicielle » des employés se faisait hors du circuit de la DSI, sur des outils développés dans les départements (Chejfec 2012). Cette nouvelle, bien qu'inquiétant les DSI qui y voient là une prise

⁵ Nous avons fait le choix de ne pas traduire ce terme anglais, qui est utilisé sous cette forme dans les enquêtes effectuées au sein des organisations.

d'autonomie néfaste à l'organisation, peut aussi révéler des aspects positifs ; nous commençons par citer ceux-ci. D'une part cela montre une forte capacité des utilisateurs à conceptualiser leurs besoins. En effet, le développement d'outils informatiques, aussi simples soient-ils, demande un effort de conception. Bien que ce processus ne soit que rarement formalisé dans le cadre du Shadow IT, il montre une véritable capacité des utilisateurs à prendre du recul sur leur besoin, et à transformer cela en outil. D'autre part, cela montre la volonté des utilisateurs de s'investir pour créer leurs outils. Ces utilisateurs qui considèrent ne pas pouvoir assurer leur mission de façon optimale avec les outils qu'ils ont à disposition se mettent dans une posture pro-active pour rechercher leurs solutions. Cet attachement au travail « bien fait » montre leur capacité à se révéler dans une véritable posture de travailleur cognitif (Colletis 2014). De plus, les outils développés peuvent ensuite parfois se diffuser dans l'organisation, légitimant ainsi l'acteur dans sa pratique (Messaoudi, Meissonnier et Vitari 2019). Enfin, « *la déviance fait partie du modèle* » (Zarifian 1996), et il semble donc inévitable que des comportements de détournements aient lieu dans les organisations. L'usage du Shadow IT fait partie de ces comportements de « ruse » évoqués en partie 1.2.2.3. et montre ici encore la capacité d'adaptation des utilisateurs, face à des outils leur semblant inadéquats.

Cela dit, les effets néfastes évoqués plus haut sont bien réels. Tout d'abord la difficulté de partage et de mise en connexion de tous ces outils, tableurs Excel le plus souvent, développés à chaque étage de l'organisation. Les différentes composantes du Shadow IT se développent en effet en parallèle du système d'information officiel et sont rarement liés entre eux. Cela peut entraîner des risques de non alignement des processus, des données, qui peuvent générer des dysfonctionnements dans l'organisation. On peut également citer la problématique de perte de connaissances pour l'organisation, qui n'accède pas aux informations contenues dans ces outils de Shadow IT, et n'en bénéficie donc pas pour son système d'aide à la décision notamment. Or nombre de décisions sont prises sur la base de ces systèmes, qui s'ils ne tiennent pas compte de « toute » l'information de l'organisation, peuvent proposer des indicateurs et autres tableaux de bord éloignés de la réalité métier, et donc peu fiables. Enfin, et ce sont les problématiques les plus évoquées par les DSI, on peut ajouter la non-sécurisation de ces outils. Les outils en ligne notamment qui sont utilisés dans le cadre du Shadow IT présentent des failles de sécurité pouvant entraîner des risques importants pour l'organisation.

Selon nous, concevoir et développer des systèmes d'information soutenant la démocratie des organisations pourrait être une première réponse au Shadow IT. En effet, ce phénomène est avant tout une adaptation à un problème généralisé dans les organisations : l'inadéquation des outils aux véritables besoins métier. Cela est partiellement dû à une asymétrie d'information entre DSI et unités métier, et on peut supposer que la participation des utilisateurs telle que proposée dans un processus de construction démocratique des systèmes d'information serait une première réponse adéquate. En prolongeant cette idée, l'implication des utilisateurs dans la prise de décision au sujet des systèmes d'information qui les concernent permettrait de résoudre en partie cette question. Les utilisateurs seraient alors à même de faire des choix structurant concernant le système, qui serait alors plus adapté à leurs besoins réels. Enfin, prévoir les conditions dans lesquelles le système pourrait être adapté en fonction de l'évolution des points de vue servirait également à créer des systèmes d'information plus durables, et à limiter le Shadow IT.

Nous sommes conscients du fait que des systèmes d'information soutenant la démocratie ne sauraient être une réponse complète à la problématique du Shadow IT. Cela étant, nous avons montré dans cette première partie que cette proposition pourrait concrètement apporter des solutions aux DSI, qui, si elles adoptaient des démarches plus participatives et impliquant l'utilisateur dans le processus de décision, freineraient très certainement l'usage du Shadow IT qui est aujourd'hui en pleine expansion.

Cette proposition se base aussi sur le fait que la Shadow IT nous montre de réelles opportunités d'engagement et de créativité de la part des utilisateurs, qui pourraient bénéficier à l'ensemble de l'organisation.

1.3.2. Générer de l'innovation grâce aux travailleurs cognitifs

Nous pensons également que les systèmes d'information supportant la démocratie dans les organisations pourraient permettre de redonner de la maîtrise aux utilisateurs sur leurs procédures de travail, d'assouplir ces procédures et de les rendre plus adaptables. Il s'agit ici pour chacun de s'accomplir en tant que travailleur cognitif, qui se révèle « *dans des situations inédites qui, par définition, ne sont ni routinières, ni répétitives* » (Colletis 2014). Ce travailleur cognitif, comme Colletis le définit, est porteur d'innovations, force de propositions de solutions nouvelles lorsqu'il a à résoudre des problématiques inconnues jusqu'alors.

Commençons par décrire la situation de ces travailleurs lorsqu'ils doivent se conformer à des systèmes d'information non démocratiques : porteurs de normes non débattues, non élicitées, rigidifiant les procédures. Comment peut-on imaginer un travailleur proposant de l'innovation dans une telle situation (Dupuy 2015) ? Si le travail ne correspond qu'à une « *conformité à des normes* » (Zarifian 1996), portées par les systèmes d'information, alors le travailleur « *isolé derrière sa machine* » peut-il être force de proposition ? La réponse à ces questions nous semble bien évidemment négative, et cela pour trois raisons que nous allons détailler.

D'une part, lorsque le logiciel est en place dans les services de l'entreprise, il participe à la normalisation du travail des utilisateurs, au sens où il impose les procédures opérationnelles exécutées par ces derniers. Les activités sont cadrées, dans un but de rationalisation et d'efficacité, et il est difficile pour le salarié de s'affranchir de cette norme tout en restant dans le cadre de l'outil de travail qui lui est fourni. C'est ainsi que Bernard Morand décrit le logiciel comme « *sujet et objet de la norme* » (Morand 2007). Cette normalisation empêche les salariés de dévier des procédures, même si elles ne s'alignent pas totalement sur les objectifs des processus, et les désengage d'une certaine manière de leur mission : le logiciel venant leur imposer la manière dont il « faut » travailler, cette manière de travailler n'est pas remise en question. On revient à « *la dichotomie prescription-exécution, héritée de l'approche taylorienne du travail* », comme le dit Françoise Darses (Francoise Darses 2004) où le salarié exécute des tâches organisées non plus par le contremaître mais par le logiciel de gestion qui vient le remplacer. Si l'on s'accorde à dire que l'un des prérequis à l'innovation participative est un engagement fort des salariés pour l'entreprise, on constate ici que les logiciels de gestion qu'ils utilisent quotidiennement conduisent plutôt à leur désengagement, et cela constitue un premier frein à l'innovation.

D'autre part, si le système d'information a pour but d'optimiser les processus, il prévoit rarement la gestion d'éléments hors du cadre prévu. Nous connaissons tous le cas d'une demande à un guichet, qui ne peut être traitée parce qu'elle « ne rentre pas dans les cases ». Un agent face à cette demande peut imaginer une solution nouvelle mais soit il la traitera manuellement, soit il pourra « tordre » la réalité pour qu'elle s'intègre à la norme imposée par le logiciel. Dans tous les cas, d'une part ce cas extra-ordinaire sera absent du système d'information (ou sera présent uniquement dans des outils de Shadow IT), et d'autre part le système freine la créativité qui pourrait être déployée par les agents pour gérer le cas. Un autre prérequis à l'innovation participative que nous avons identifié est la créativité. Or on s'aperçoit ici qu'elle est freinée au quotidien pour les utilisateurs des logiciels de gestion qui sont soumis à un cadre inflexible, les obligeant à se conformer à une procédure routinière.

Enfin, si les logiciels de gestion sont les outils de travail quotidiens de certains salariés, ils ont aussi une autre fonction : produire de l'information support à la décision. Dans un premier temps, les logiciels produisent les données utiles aux tableaux de bord avec des statistiques très précieuses aux entreprises. On parle alors d'outils de reporting. On y retrouve par exemple pour le service commercial des informations sur les produits vendus, pour le service marketing des informations sur la cible, etc. Ces données, centralisées dans le logiciel par les salariés, constituent le patrimoine informationnel de l'entreprise, et donc une partie de sa richesse. Dans un second temps, l'information produite par les logiciels de gestion peut également porter sur les processus de travail eux-mêmes. En effet, ils ont la capacité d'informer les managers sur les temps de traitement des dossiers, les cadences de production, en produisant des statistiques sur le personnel. C'est ce que Salvatore Maugeri appelle « *dispositifs de gestion* » (Maugeri 2007). Or ces technologies que l'on appelle aussi les technologies invisibles ne le sont pas tant que ça. Les salariés sont conscients de ces contrôles, et le fait qu'ils ne soient pas explicites suscite souvent leur défiance. Nous considérons pourtant que la dernière condition à l'émergence d'innovation participative est un climat de confiance pour les salariés. On constate ici que ces dispositifs de gestion viennent « polluer » ce climat dans l'entreprise et y amènent une forme d'hostilité.

Une organisation qui souhaiterait être innovante doit selon nous se positionner sur deux points. Dans un premier temps, mettre en place des principes et des normes qui correspondent à des représentations innovantes, et les partager (Salles 2007). Il nous semble que l'innovation ne peut se faire dans le cadre contraint et restreint de représentations et de visions qui ne sont pas elles-mêmes innovantes. Pour revenir à l'exemple de la Comptabilité Sociale Environnementale évoquée plus haut, elle porte avec elle de nouvelles visions de la comptabilité de l'organisation, qui peuvent être un véritable levier d'innovation. En restant dans le cadre des normes comptables classiques, il semble plus difficile de parvenir à proposer des innovations ayant trait au développement durable dans l'organisation par exemple. Dans un second temps, l'organisation doit laisser la possibilité aux travailleurs d'innover, en adaptant de nouvelles procédures à des « *situations événementielles complexes* ». Pour cela le système d'information doit conserver suffisamment de souplesse pour ne pas contraindre le travailleur à une procédure unique ou trop difficilement adaptable. Sans nier les gains en termes d'efficacité que permettent des systèmes d'information dits « rationalisés », libérer un tant soit peu leur structure permettrait sans doute de gagner en opportunités d'innovations.

La conception d'un système d'information soutenant la démocratie pourrait permettre de répondre à ces enjeux. Dans un premier lieu en permettant de discuter des représentations sous-jacentes aux systèmes d'information, et en y intégrant dès lors des représentations innovantes. En effet, la participation d'utilisateurs au plus proche du métier dans les débats autour des représentations portées par le système contribuerait certainement à l'émergence d'idées créatives et innovantes. En second lieu en impliquant les travailleurs dans le processus de conception, et en les laissant décider de la possibilité de garder une certaine souplesse dans les procédures, pour laisser place à l'apparition d'innovations. Cet aspect est à mettre en lien direct avec la question précédemment traitée du Shadow IT. Si les utilisateurs ont aujourd'hui largement recours au Shadow IT pour traiter les situations ou procédures exceptionnelles – et donc parfois innovantes – c'est notamment parce que les outils qu'ils ont à leur disposition sont trop rigides. Les laisser participer aux choix de conception pour laisser la possibilité d'une « marge de manœuvre » quant aux outils informatiques pourrait représenter une possibilité nouvelle pour les organisations de capter les processus innovants.

Nous sommes conscients du fait que des systèmes d'information démocratiques ne sauraient être le seul et unique levier d'innovation dans l'organisation. Cependant, ils peuvent concourir à l'émergence de nouvelles idées, et surtout ils peuvent être vecteurs de discussions sur de nouvelles représentations

innovantes au sein de l'organisation. De plus, nous avons vu que les logiciels (de gestion notamment) pouvaient être de véritables freins à l'innovation, et qu'il était possible de renverser la tendance en proposant des systèmes d'information démocratiques.

1.3.3. Partager la connaissance avec le principe des Communs⁶

De manière générale, le bien commun est défini comme « *toute ressource partagée par un groupe de gens* » (Ostrom 2007). Cette définition s'est longtemps appliquée aux ressources naturelles et en opposition aux biens privés ou publics. Elle met en lumière le fait que le bien commun n'a pas toujours une vocation universelle, au sens de « partagé par tous », mais peut être réservé à une communauté. Ostrom ajoute à cette définition que « *chaque commun est un cas particulier* » (Hess et Ostrom 2007), avec ses propres problématiques, et qu'il doit donc être analysé, observé, gouverné de manière adaptée. On peut aujourd'hui considérer une acception plus large du Commun, incluant la ressource mise en commun, mais également les « *formes et méthodes de son entretien* » (Le Crosnier 2013). C'est cette définition qui nous propose de passer du bien commun au Commun, en ouvrant le champ des communs aux biens intangibles, et aux processus de création et de gestion de ces ressources. Ainsi, on peut considérer comme un Commun : la connaissance, la culture, le savoir, mais également les moyens de gestion que l'on y associe. Le Commun est donc à la fois l'action et le résultat de l'action, c'est « *le processus et les informations/données résultant du processus* » (Deneulin et Townsend 2007). Ce Commun s'expose à de nouvelles menaces, qui ne pèsent pas uniquement sur la ressource elle-même, mais aussi sur la communauté, « *dans sa capacité à créer ou adapter* » le Commun (Le Crosnier 2013). On parle de menaces d'enclosure, et James Boyle a proposé une redéfinition de ces menaces, un « *second mouvement des enclosures* », qui apparaît sous la forme d'une « *privatisation des savoirs* » (Boyle 2003). Enfin, l'une des autres facettes du Commun est qu'il est vivant, qu'il est une sorte de « *chantier permanent* » et que c'est cette dynamique du Commun qui nécessite l'existence d'une communauté apte à s'organiser pour le gouverner.

La définition que nous avons élaborée du Commun à partir de ces points de vue complémentaires est la suivante : il est composé d'une ressource et des moyens de gestion de cette ressource. Il est partagé et pris en charge par une communauté. Le Commun est en constante évolution et la communauté doit, pour le faire perdurer, se doter de moyens de gouvernance. Desreumaux et Brechet se sont intéressés à la notion de Commun dans l'entreprise, en partant d'un constat : les acteurs de l'organisation agissent dans un intérêt supérieur à l'intérêt particulier pour la pérennité de l'entreprise. Pour les auteurs, la poursuite de cet intérêt supérieur « *conditionne la satisfaction des intérêts spécifiques* » (Desreumaux et Brechet 2013), car chacun recevra de l'entreprise ce qu'il attend si cette dernière reste « *une affaire prospère* ». Et dans la poursuite de cet intérêt supérieur, les acteurs s'organisent, collaborent, et mettent en œuvre leurs compétences et leurs connaissances. D'après les auteurs, le Commun équivalent de la ressource naturelle serait donc, dans le cas de l'entreprise, la connaissance collective. On entend ici la connaissance collective comme les savoirs et savoir-faire des acteurs des processus, mais également comme les moyens de gestion de ces savoirs. Au sein de l'organisation, les savoirs et savoir-faire peuvent s'exprimer au travers des procédures de travail et des données qui circulent entre les acteurs.

En reprenant la définition du Commun que nous avons établie précédemment, on peut donc considérer que la connaissance collective et les moyens de création et de gestion de cette connaissance représentent un Commun. Ils sont un Commun pour la communauté d'acteurs qui

⁶ Les recherches autour des biens communs ont fait apparaître ce terme de Commun, toujours noté avec une majuscule afin de le distinguer clairement du concept de bien commun.

composent l'organisation, et qui mettent en œuvre des modes de gouvernance, en adaptant les procédures notamment, pour faire évoluer cette connaissance.

Aujourd'hui, les organisations ont besoin de rendre tangible ce Commun, d'une part pour capitaliser et partager cette connaissance collective, d'autre part pour l'informatiser. Pour cela, ils établissent des modèles, qui permettent de représenter une partie du Commun. Pour capitaliser et partager, les organisations produisent des modèles de type BPM⁷, qui sont une représentation des processus dans l'organisation. Pour informatiser, les modèles les plus couramment utilisés sont l'entité/association et les différents diagrammes objet. Les données manipulées, ainsi que les flux et traitements associés sont ainsi représentés dans un formalisme permettant, après transformations successives, la programmation de logiciels informatiques. Les modèles que nous avons évoqués ont tous pour but de capter une partie de la connaissance collective, à un instant T, pour la rendre tangible et l'utiliser. C'est donc un moyen, pour les organisations de représenter tout ou partie du Commun.

Lorsque les organisations choisissent d'informatiser leurs processus, elles mettent en œuvre des projets informatiques. Ces projets ont pour but le développement ou l'achat d'un logiciel métier qui sera le support des processus de production ou de gestion dans l'organisation. Depuis les années 80, les démarches projet ont évolué pour impliquer de plus en plus l'utilisateur final dans la conception de son futur logiciel. Cependant, un principe reste inchangé : les futurs utilisateurs doivent en début de projet exprimer leur connaissance collective afin d'expliquer leur métier, et la façon dont ils mènent à bien leur mission. A partir de cette matière, les concepteurs réalisent des modèles (ou les coconçoivent avec les futurs utilisateurs) qui serviront de base de travail aux développeurs qui produisent le logiciel. Ces modèles, dont nous avons parlé précédemment, permettent de formaliser les données manipulées, les flux, les traitements, les processus, etc. Ils permettent en d'autres termes de capter le Commun, pour que ce dernier puisse être opérationnalisé au travers du logiciel. Comme le dit Charlotte Hess : « *Si les formats numériques facilitent l'accès aux contenus, ils permettent également de s'approprier ce qui fut longtemps intangible* » (Peugeot 2011).

Il y a selon nous deux raisons pour lesquelles on peut considérer que ces logiciels font peser une menace d'enclosure sur le Commun. Dans un premier temps, le processus de production des logiciels dépossède les acteurs de l'organisation de leur Commun. En effet, les logiciels sont construits par transformations successives de modèles : on passe des modèles conceptuels aux modèles logiques puis aux modèles physiques. Alors que les modèles conceptuels sont appréhendables par les acteurs de l'organisation, la technicité requise pour les transformer en modèles logiques et physiques est réservée aux experts. Ce cloisonnement fort entre les acteurs du projet informatique représente une menace d'enclosure car il tend à déposséder la communauté détentrice de la connaissance collective de cette dernière (cf. Figure 12). En effet, le Commun, ou plutôt la représentation de celui-ci implantée dans le système d'information n'est plus accessible aux acteurs de l'organisation qui sont les futurs utilisateurs des logiciels. Or, comme nous l'avons dit précédemment, le Commun est défini par le fait qu'il est aux mains d'une communauté dont les acteurs partagent un intérêt commun.

⁷ Le BPM peut signifier Business Process Modeling ou Business Process Management. Dans l'utilisation que nous faisons de ce terme, nous nous concentrons sur le Modeling, c'est-à-dire la modélisation des processus métier.

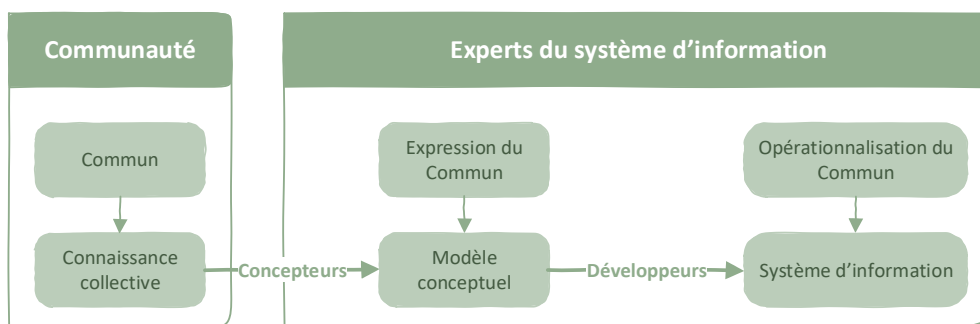


Figure 12 : La responsabilité cloisonnée des acteurs : une menace d'enclosure

Dans un second temps, lorsque la connaissance collective est captée dans les modèles, et implantée dans le système d'information numérique, il est très complexe de venir la modifier car les logiciels reposent sur des structures rigides dont la réingénierie est coûteuse. Parmi les modèles conceptuels utilisés depuis les années 80, le modèle entité association (et son équivalent logique : le modèle relationnel) est celui qui permet de structurer les bases de données relationnelles. Ces bases de données sont aujourd'hui les plus utilisées, elles sont efficaces, et permettent d'optimiser la gestion des données, mais elles imposent également une rigidité aux logiciels qu'elles supportent (Bruchez 2015). Ainsi, une fois le logiciel développé à partir des modèles conceptuels, il est très complexe de venir modifier ces mêmes modèles, car ces derniers structurent le logiciel. Les évolutions étant complexes, elles sont coûteuses pour les organisations, qui doivent faire appel à des experts pour répondre à leurs demandes. Or ces modèles sont une représentation du Commun, que nous avons défini comme étant vivant, en évolution constante. Le fait que les utilisateurs du logiciel ne puissent pas faire évoluer eux-mêmes ce dernier lorsque leur connaissance collective évolue est donc une véritable menace d'enclosure. Non seulement on ôte son caractère dynamique au Commun, qui est comme « emprisonné » dans le logiciel, mais on empêche également tout moyen de gouvernance de celui-ci par la communauté.

Les menaces d'enclosure que font porter les systèmes d'information numériques sur la connaissance collective des organisations peuvent être une source de difficulté pour ces dernières. Dans un premier temps, et dans la continuité de la partie précédente concernant l'innovation, le logiciel impose des normes de travail (Morand 2007), qui sont en adéquation avec la connaissance collective captée à un instant T (lors des premières phases du projet informatique). Si ces normes pouvaient se modifier, au rythme de l'évolution de la connaissance collective, elles seraient peu contraignantes. Mais le fait que le logiciel ne puisse évoluer simplement entraîne des difficultés à faire correspondre la norme qu'il impose avec la connaissance collective « réelle ». Ce décalage dessert l'organisation, et diminue sa capacité à s'adapter. Dans un second temps, les acteurs des processus deviennent des exécutants, et n'ont plus la possibilité d'être des travailleurs cognitifs, tels que définis par Colletis (Colletis 2014). En effet, les utilisateurs de logiciel se conformant aux automatismes que produit ce dernier, ils n'ont plus la possibilité de faire évoluer leurs connaissances lorsqu'ils sont face à des situations particulières puisque ces nouvelles connaissances ne peuvent s'inscrire pas dans le cadre du logiciel qu'ils doivent utiliser. Dit autrement, cette menace d'enclosure conduit à un appauvrissement des connaissances et des savoirs. Stiegler parle à ce propos d'une forme de « *prolétarianisation des acteurs* » des processus de gestion (Stiegler et Ricard 2014).

Les menaces d'enclosure que nous avons mises en évidence sont propres au Commun que nous avons défini : la connaissance collective et les moyens de gestion de celle-ci. Ce sont les systèmes d'information, et qui plus est leur partie informatisée, que nous avons identifiés comme porteurs de ces menaces, et pourtant, nous considérons que des systèmes d'information démocratiques peuvent

être une véritable chance pour le développement des Communs. Ils pourraient être un vecteur de création et de gestion du Commun et permettre de résister aux menaces d'enclosure en mettant en œuvre les principes suivants :

- Dans un premier temps les acteurs ne devant pas être dépossédés de leur Commun, ils doivent pouvoir participer à la construction du système d'information support à leur processus métier. Ils ne doivent pas être écartés au profit d'experts techniques, et doivent pouvoir maîtriser la construction du système en comprenant les tenants et les aboutissants.
- Dans un deuxième temps, le Commun - qui est d'une certaine manière opérationnalisé dans le système d'information - doit pouvoir vivre, évoluer et s'enrichir. Pour cela, les utilisateurs des logiciels doivent pouvoir contribuer à l'amélioration du système, au rythme de l'évolution de leurs connaissances. Le système doit alors pouvoir conserver une certaine forme de souplesse et s'adapter aux points de vue des utilisateurs.
- Dans un troisième et dernier temps, le Commun ne pouvant exister sans une gouvernance efficace, les utilisateurs doivent pouvoir participer au processus de décision quant au système. Pour cela, leur rôle ne doit pas être juste consultatif, mais réellement participatif et leurs choix doivent être pris en compte.

Les caractéristiques d'un système d'information soutenant la démocratie dans les organisations nous semblent être en lien direct avec les principes édictés ci-dessus. En effet, un système d'information démocratique doit être construit par les utilisateurs pour les utilisateurs au travers de démarches participatives. De plus, les points de vue doivent pouvoir y être représentés, prenant ainsi en compte une pluralité de visions, de connaissances dans l'organisation. L'un des enjeux du système d'information soutenant la démocratie pourrait donc être la diffusion, le partage de la connaissance, autrement dit le développement d'un Commun alimenté par le collectif constituant l'organisation.

Nous avons décrit dans cette troisième partie quels étaient pour nous les enjeux auxquels un système d'information démocratique pouvait être une réponse. Outre le soutien à la démocratie dans les organisations, qui représente un véritable enjeu à lui seul, ce type de système d'information permettrait de mieux maîtriser les recours au Shadow IT, mais serait aussi un véritable levier d'innovation et un vecteur de soutien du Commun d'une organisation : sa connaissance collective.

Sans être des réponses complètes à ces enjeux, les systèmes d'information démocratiques sont pour nous une partie de la réponse à ces problématiques actuelles des organisations.

1.4. Conclusion du chapitre

En partant d'une réflexion philosophique sur la démocratie dans nos sociétés et dans les organisations, nous avons dans ce chapitre formalisé une vision de ce que serait un système d'information soutenant la démocratie. En dressant une liste des grands principes de la démocratie dans la société, et en déclinant ces principes pour les adapter aux organisations, il nous a semblé évident que les systèmes d'information avaient un rôle tout à fait central à jouer quant à ces questions.

Comme nous l'avons dit, notre vision du système d'information est politique et notre définition s'appuie sur des théories constructivistes. Partant de cela, il nous apparaît que des alternatives aux

choix de construction du système d'information couramment mis en œuvre peuvent être proposés. Il est ainsi possible d'envisager que le système d'information devienne un soutien à la démocratie dans l'organisation, et non une entrave, s'il respecte un certain nombre de critères que nous avons énoncés. Cet objectif, s'il est déjà très ambitieux, peut également servir d'autres enjeux auxquels sont aujourd'hui confrontées les organisations. Parmi eux, la lutte contre le Shadow IT, l'accroissement de l'innovation et le partage de la connaissance nous ont semblé particulièrement importants.

Parmi les caractéristiques principales du système d'information démocratique, la participation des utilisateurs au processus de construction et la prise en compte des points de vue dans le système d'information lui-même sont absolument nécessaires. La proposition que nous faisons dans cette thèse s'inscrit dans ce cadre. Mais avant de décrire en détail notre proposition, il nous semble indispensable de regarder comment ces questions sont aujourd'hui traitées dans la littérature. C'est ce que nous nous proposons d'explorer dans le deuxième chapitre de cette thèse, au travers des démarches, méthodes et modèles qui adressent ces problématiques.

Dans une première perspective, la garantie de démocratie passe avant tout par une implication des utilisateurs au processus de construction de leur système d'information. Cette implication active, c'est-à-dire entraînant une participation directe à la prise de décision, a pour but de leur permettre de débattre des normes et valeurs encapsulées dans le système d'information, mais aussi d'en faire des « citoyens éclairés ». On entend par cette dernière expression que leur implication est aussi une façon de comprendre pourquoi et comment les outils du système d'information ont été construits, et de pouvoir ainsi limiter l'usage de ces outils comme dispositifs de la gouvernementalité. La première partie de notre état de l'art traite donc de cette question de l'implication des utilisateurs dans la construction de *leur* système d'information.

Dans une seconde perspective, un système d'information dit démocratique est un système d'information qui garantit le respect d'une pluralité de points de vue et donc son corrélat : une non-conformité de tous à un point de vue dominant. Cette question des points de vue des utilisateurs touche au système d'information lui-même, à sa structure et à son architecture qui peuvent venir ancrer dans le réel une vision dominante comme nous l'avons montré dans la section 1.2. Pour ces raisons, nous consacrons la deuxième partie de notre état de l'art à cette question du point de vue des utilisateurs sur *leur* système d'information.

Chapitre 2

ETAT DE L'ART - LES UTILISATEURS ET LEUR SYSTEME D'INFORMATION

L'état de l'art que nous proposons s'articule autour de trois grandes phases de la construction de système d'information : le recueil des exigences, la phase de conception et la phase de développement du système d'information (cf. Figure 13). Ces trois étapes qui s'enchaînent le plus souvent de manière chronologique questionnent chacune un aspect de la construction du système d'information : *Pourquoi est-il construit et quels sont les besoins auxquels il répond ?* pour la phase de recueil des exigences. *Quelles sont les services attendus, de quelle information ai-je besoin et comment le structurer ?* pour la phase de conception. Enfin *comment développer et architecturer le système d'information afin qu'il réponde aux besoins ?* pour la phase de développement.



Figure 13 : Les principales phases du processus de construction de système d'information⁸

Notre état de l'art est divisé en deux parties. D'un côté nous nous penchons sur l'implication des utilisateurs dans le processus de construction des systèmes d'information. Cette partie correspond à notre première problématique : l'apport de démocratie dans le processus de construction du système d'information. Nous analysons l'implication des utilisateurs depuis le recueil des exigences jusqu'au développement en passant par la conception du système d'information. De l'autre côté, nous nous intéressons à la prise en compte des points de vue utilisateurs depuis la phase de recueil des exigences jusqu'aux modèles conceptuels produits lors de la phase de conception, pour comprendre les enjeux liés à notre seconde problématique : le caractère démocratique du système d'information lui-même, autrement dit sa capacité à prendre en compte une pluralité de points de vue.

⁸ Le terme de conception utilisé dans ce schéma est une traduction du *design* anglais ne comprenant pas la phase d'analyse. La phase d'analyse du besoin se fait en amont de la conception, au niveau du recueil des exigences.

2.1. La place des utilisateurs dans le processus de construction des systèmes d'information

L'implication des utilisateurs dans le processus de construction de leur système d'information est l'une des caractéristiques permettant de le qualifier de démocratique. Cette implication est traitée dans la littérature depuis le recueil des exigences jusqu'au développement de système d'information, en passant par leur conception. Dans cette première partie nous nous questionnons tout d'abord sur les motivations qui ont poussé les concepteurs à accentuer l'implication des utilisateurs dans les projets informatiques depuis une vingtaine d'années notamment. Nous nous intéressons ensuite à l'implication des utilisateurs lors de la phase de recueil des exigences, puis nous dressons un panorama des diverses méthodes de conception garantissant une place à l'utilisateur, à des degrés divers. Enfin, et pour aborder le champ du développement, après avoir décrit dans les grandes lignes l'agilité, nous faisons un lien entre les méthodes de développement agiles et la conception participative. Nous concluons cette première partie par les limites que nous avons pu identifier dans cet état de l'art, et par notre positionnement (cf. Figure 14).

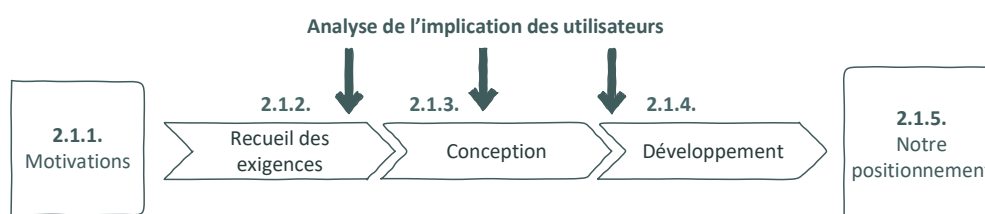


Figure 14 : Plan de la partie 2.1

2.1.1. Quelles étaient et quelles sont les motivations ?

« 66% des logiciels développés ne répondent pas aux attentes des utilisateurs que ce soit au niveau de leur fonctionnalité ou de leur comportement » (Wanderley et al. 2014). Ce chiffre avancé par Wanderley en 2014 amène les concepteurs de systèmes d'information à un constat amère : malgré les coûts et la durée importante des projets informatiques, les logiciels ne donnent majoritairement pas satisfaction aux principaux intéressés que sont les utilisateurs finaux. En 2006 déjà, McConnell avait pu identifier dans la littérature que le manque d'implication des utilisateurs était le facteur d'échec le plus important des projets informatiques (McConnell 2006). La participation grandissante des utilisateurs finaux à la conception de leur logiciel depuis une vingtaine d'années a donc été avant tout un argument rationnel pour améliorer les fonctionnalités et la qualité dudit logiciel (Kushniruk et Nohr 2016).

Ajoutons à cela que la cocréation, la co-conception ou tout simplement la collaboration des concepteurs avec les utilisateurs est également considérée comme la garantie d'une plus grande durabilité des systèmes. Au contraire de systèmes imposés (Mahaux et Castiaux 2015) à leurs utilisateurs, les systèmes dans lesquels ces derniers sont de véritables acteurs de la conception (Caelen 2004; Françoise Darses 2004) sont en général mieux acceptés (Lyons, Smuts, et Stephens 2001). Ils sont en effet plus adaptés en termes de fonctionnalités et plus conformes aux attentes d'ergonomie des utilisateurs. L'acceptation de ces logiciels par les utilisateurs est donc plus grande et plus rapide, nécessitant une phase d'accompagnement au changement moindre. De plus les systèmes sont plus durables au sens où, correspondant mieux aux attentes des utilisateurs, les modifications qu'ils nécessitent n'interviendront que plus tard dans le cycle de vie du logiciel.

Un dernier argument en faveur de la participation des utilisateurs aux projets de conception nous semble particulièrement pertinent : l'empowerment de ceux-ci. Les résultats attendus de cet

empowerment, bien que difficilement quantifiables, relèvent d'un véritable choix politique. L'empowerment peut être défini comme la possibilité de donner aux Hommes un sentiment de pouvoir plus important (Mahaux et Castiaux 2015), une plus grande motivation à agir, mais aussi un meilleur contrôle sur ce qui les concerne directement à long terme (Lyons, Smuts, et Stephens 2001). Simonsen va même jusqu'à considérer cette possibilité de participation comme un droit fondamental. Selon lui, ceux qui sont affectés par un changement résultant de la conception ou de la mise en œuvre d'un système informatique doivent être en capacité d'influencer cette conception et ce processus d'implémentation (Simonsen 2013). Cela représente, toujours selon lui, un enjeu d'émancipation pour les utilisateurs dont ils « *conçoivent l'avenir* ».

Les motivations actuelles qui poussent à impliquer davantage les utilisateurs dans la construction de leur système d'information sont donc de trois ordres : amélioration de la qualité des logiciels, production de systèmes plus durables et empowerment des utilisateurs. Bien que les enjeux d'accroissement de la qualité et d'amélioration de la durabilité des systèmes représentent des motivations majeures, nous nous intéressons plus particulièrement à la question de l'empowerment dans la suite de cet état de l'art. Nous cherchons plus précisément à distinguer les méthodes de recueil des exigences, de conception et de développement qui permettent de soutenir un empowerment démocratique des utilisateurs de systèmes d'information.

2.1.2. L'implication des utilisateurs dans le processus de recueil des exigences

Dans cette partie nous nous intéressons à l'implication des utilisateurs dans le processus de recueil des exigences de la construction des systèmes d'information (cf. Figure 15). Cette première grande phase des projets informatiques est absolument cruciale puisqu'elle est celle où se discutent les raisons d'être d'un système, les besoins auxquels il doit répondre, et donc les normes et valeurs qu'il encodera.

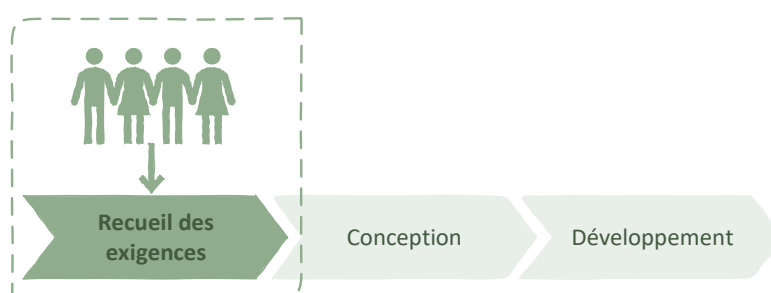


Figure 15 : L'implication des utilisateurs dans le processus d'ingénierie des exigences

La définition de l'ingénierie des exigences telle que livrée par Ross et Schoman dans les années 70 inclut l'idée que cette discipline ne s'intéresse pas seulement à la question de ce que doit faire le système, mais aussi à celle du *pourquoi* il doit le faire (Ross et Schoman 1979). Il s'agit ici du point de départ de l'ingénierie des exigences orientée par les buts (Lamsweerde 2009; Lamsweerde 2000). Milne ajoute à cela que les notions de pouvoir et de politique ne doivent pas être écartées de la question des exigences, car elles sont vitales à une bonne élicitation de ces dernières, et n'interfèrent pas de manière négative (Milne et Maiden 2012). Sans évoquer pour le moment la question de l'implication des utilisateurs dans le processus d'ingénierie des exigences, ces premières définitions nous montrent quel lien il est possible d'établir entre exigences et démocratie. Les exigences représentant un enjeu politique, au sens où elles adressent les questions stratégiques liées au projet informatique : elles sont au cœur de la question de la démocratie dans les organisations.

Aujourd'hui, l'importance de l'implication des utilisateurs en tant que parties prenantes dans le processus d'ingénierie des exigences, et notamment durant la phase d'élicitation, est relativement établie. Cependant, les façons dont sont considérés les utilisateurs dans ce type de démarches

diffèrent fortement. Pour certains auteurs, les utilisateurs sont vus comme un input clé dans le processus, les considérant comme participant à la chaîne de valeurs. C'est le cas de Fricker par exemple (Fricker 2010), qui apporte une vision que l'on pourrait qualifier de « productive » au sens où l'implication des utilisateurs est vue comme un moyen d'augmenter la valeur du produit d'un processus. Pour d'autres, les utilisateurs acquièrent un véritable statut de co-créateur des exigences, tout comme les autres parties prenantes de cette phase (Mahaux et al. 2013). Dans ce cas-ci, l'objectif poursuivi est plutôt celui de l'innovation, par le biais d'un processus créatif, que celui de l'empowerment démocratique. Dans les deux propositions évoquées, le choix d'impliquer les utilisateurs est avant tout une garantie d'accroissement de la qualité du processus.

En ingénierie agile des exigences, l'approche centrée humains est un facteur de succès des projets, et des systèmes qu'ils permettent de développer. Cette notion de succès, difficilement quantifiable, n'est pas précisée outre mesure (Schön, Thomaschewski, et Escalona 2017). Ici la participation des utilisateurs se fait lors des phases de test du système (au regard des exigences), et par un système de feedback concernant l'expérience utilisateurs quant aux scénarios d'usage établis. Dans ce cas, la participation des utilisateurs à la phase d'élicitation des exigences est un moyen de comprendre mieux le contexte d'utilisation, mais les utilisateurs n'ont pas un véritable rôle participatif. Ils interviennent par le biais de consultations, et l'impact de leurs propositions se limite aux questions d'usage du système d'information.

Depuis les années 90, la question des risques est au cœur des préoccupations, et la participation des utilisateurs à la définition des exigences du projet est une façon de diminuer les risques d'incompréhension qui peuvent se jouer. Ces risques, qui pourraient se répercuter sur le projet dans son ensemble, s'ils ne sont pas pris en compte dès la phase d'élicitation des exigences, seraient amoindris par l'instauration d'une compréhension mutuelle, des termes utilisés notamment, entre experts et utilisateurs (Kotonya et Sommerville 1996). Les utilisateurs sont alors d'une certaine façon considérés comme un potentiel risque, au regard de leurs résistances au changement notamment. Leur implication permet de diminuer ce risque, mais ne leur garantit en rien un plus fort empowerment démocratique, ou une participation plus accrue au processus de décision.

Depuis les années 2000 cependant, de nouvelles visions de l'ingénierie des exigences, plus proches de celle que nous partageons, apparaissent. Lukes par exemple propose en 2004 d'intégrer la question des normes, des valeurs et de l'idéologie au processus d'ingénierie des exigences (Lukes 2004). La question du pouvoir en tant que problématique organisationnelle est pointée comme un élément crucial de la phase de recueil des exigences (Wieggers 2006) afin notamment d'éclaircir les intérêts et motivations de chacun, tout en prenant en compte les différentes visions du monde qui coexistent dans l'organisation (Milne et Maiden 2012). Mais si l'apport des utilisateurs, en tant que parties prenantes au processus, est présenté comme indispensable pour une meilleure compréhension de ces sujets (Lin et Silva 2005), ces différentes propositions n'intègrent pas directement de méthode pour impliquer l'utilisateur et s'assurer que ses apports soient ensuite respectés et traduits dans la phase suivante de conception du système d'information.

La méthode Value Sensitive Design (VSD) constitue sans doute la méthode la plus aboutie en termes de prise en compte des valeurs des utilisateurs dans la construction de systèmes d'information (Salles 2015). Au niveau du recueil des exigences, elle propose trois types d'analyse combinées : une analyse conceptuelle, une analyse empirique et une analyse technique de la question des valeurs (Friedman et al. 2013). La méthode permet ainsi de connaître les valeurs des différentes parties prenantes du projet, le contexte dans lequel s'inscrit le projet, ainsi également les solutions techniques qui peuvent être retenues compte tenu des valeurs identifiées. Cette méthode nous semble absolument nécessaire, mais insuffisante du fait qu'elle ne propose aucune formalisation sur la traduction de ces valeurs dans

la conception propre du système (en termes de modèles), puis dans l'implémentation des modèles conçus.

Le recueil des exigences est la première phase de construction d'un système d'information. Elle tient un rôle crucial au sens où elle est à l'origine des normes, valeurs et principes qui seront ensuite encodés dans le système d'information, tel que nous l'avons décrit dans la section 1.2.2. Un système d'information démocratique nécessite que sa construction débute par une phase d'élicitation des exigences participative, et nous avons montré que différentes propositions en ce sens étaient faites en ingénierie des exigences. Cela étant, si les propositions sont diverses, nous nous appuyons avant tout sur celles qui suggèrent de faire participer l'utilisateur de manière active, et de façon à éliciter les représentations et visions sur lesquelles le système reposera. On retient aussi qu'il est nécessaire de proposer une véritable démarche d'accompagnement de cette participation, afin que les éléments ayant fait l'objet de choix à cette étape du projet soient bel et bien transcrits dans la conception, puis le développement du système d'information.

2.1.3. Les différentes démarches de conception impliquant les utilisateurs

Dans cette partie, nous nous intéressons à l'implication des utilisateurs dans le processus de conception des systèmes d'information (cf. Figure 16). Cette phase est centrale puisque c'est durant cette étape que les choix structurants concernant le système d'information sont faits : organisation des données, des processus, etc.

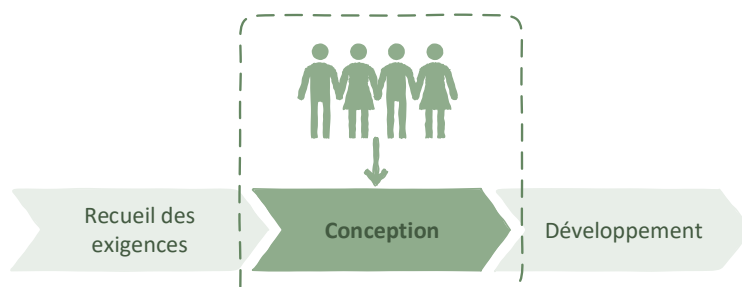


Figure 16 : L'implication des utilisateurs dans les démarches de conception

On peut identifier deux grandes familles de démarches proposant de prendre en compte l'utilisateur : la conception centrée utilisateurs et la conception participative. Ces démarches proposent un degré d'implication différent, dessinant ainsi un continuum allant de l'utilisateur comme objet d'étude à l'utilisateur acteur en passant par l'utilisateur coopératif (cf. Figure 17 : l'utilisateur est représenté avec un chapeau, le concepteur avec des lunettes).

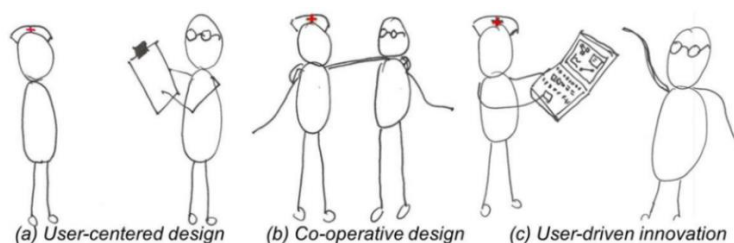


Figure 17 : Les degrés d'implication de l'utilisateur (Kushniruk et Nohr 2016)

2.1.3.1. La conception centrée utilisateurs

La conception centrée utilisateurs fait par définition un focus important sur ceux pour qui le logiciel est conçu. Les utilisateurs sont interrogés sur leurs pratiques, sont parfois observés en situation de travail, en vue de concevoir un système plus utilisable (Kushniruk et Nohr 2016). La pratique commune

à toutes les méthodes centrées utilisateur est le recueil des besoins auprès des utilisateurs eux-mêmes, et la validation par ces derniers du système développé, au travers de démarches itératives le plus souvent. Cela étant, les pratiques se révèlent très variables d'une démarche à une autre.

L'une des pratiques courantes est celle du prototypage. Les utilisateurs interagissent régulièrement avec un prototype du système en développement afin de valider ou d'invalidier la solution. Le plus souvent il s'agit de considérations ergonomiques et d'Interface Homme Machine (IHM) plus que de considérations d'ordre fonctionnel (Dupuy-Chessa et al. 2011). Les démarches dite « d'ergonomie constructive » par exemple proposent des simulations de travail basées sur des scénarios, qui sont enrichis au fil du temps afin de laisser place aux idées nouvelles des futurs utilisateurs (Barcellini, Van Belleghem, et Daniellou 2013).

La conception collaborative, qui est elle aussi une démarche centrée utilisateurs propose d'impliquer encore plus l'utilisateur en le faisant participer durant les réunions de conception, afin qu'il y apporte sa contribution continue. Il peut ainsi venir modifier les résultats observés par les concepteurs lors de l'utilisation des prototypes, et venir influencer les choix de conception (Greenbaum 1991). Les compétences de chacun sont clairement séparées, mais un apport mutuel permet d'enrichir la phase de conception.

Ces démarches, bien qu'impliquant les utilisateurs, présentent des limites évidentes. L'utilisateur est considéré comme un objet d'étude, il a un rôle informationnel et éventuellement consultatif, mais sa participation reste tout de même limitée. En effet, son rôle dans la prise de décision réelle n'est pas clairement défini. Les retours qu'il fait, les évaluations qu'il apporte sont des éléments qui viennent nourrir la conception, mais qui n'ont pas un caractère impératif. Ce sont les concepteurs et les développeurs qui mènent le processus (Ferrario et al. 2014). En ce sens, ils sont les détenteurs du pouvoir de décision, et sont donc ceux qui font les arbitrages quant à la conception du système. De plus, l'utilisateur n'est impliqué qu'à partir de la phase d'expression des besoins, et pas en amont. Cela signifie donc que la participation proposée aux utilisateurs se limite à un cadre fonctionnel déjà délimité et donc contraint. En d'autres termes, les utilisateurs travaillent sur la solution, mais ne sont pas impliqués dans la phase amont de définition du problème (Dell'Era et Landoni 2014).

Si ces démarches constituent les prémices de ce que nous considérons comme une participation de l'utilisateur au processus de construction de son système d'information, elles sont selon nous trop limitées. En termes de démocratie, elles ne respectent pas certaines des caractéristiques que nous avons établies telles que la participation des utilisateurs à la prise de décision. Elles ont cependant le mérite d'ouvrir la voie à des démarches bien plus participatives. Ces démarches, plus fidèles pour nous à ce qui est attendu d'une démarche de construction de système d'information démocratique sont présentées dans la section suivante.

2.1.3.2. La conception participative

Avec la conception participative, l'utilisateur acquiert un véritable statut de coconcepteur. Il s'agit d'un changement radical, car cela suppose d'estomper le « *traditionnel fossé issu de l'héritage taylorien, qui distingue prescription de l'activité (par les concepteurs) et exécution de l'activité (par les utilisateurs)* » (Darses 2004). Les utilisateurs sont la source de connaissance du projet (Kushniruk et Nohr 2016) et leur participation en tant que coconcepteur est donc absolument centrale (Kensing et Blomberg 1998).

Kujala a dressé en 2003 une liste des apports attendus de la conception participative (Kujala, 2003), qui peuvent être regroupés de la façon suivante :

- Une meilleure qualité du système développé. En effet, les utilisateurs participent dès la phase de recueil des exigences, qui devient alors plus précise. Cela permet dans un premier temps une compréhension globale du projet plus fine, mais aussi une expression des besoins bordée par un périmètre connu de tous.
- Une meilleure définition des fonctionnalités. Ces dernières étant choisies, décrites et validées par les utilisateurs, deux écueils majeurs des projets informatiques peuvent être évités : d'une part des fonctionnalités manquantes, d'autre part et à l'inverse des fonctionnalités inutiles. On peut lier directement à cela une meilleure gestion des coûts, car habituellement la modification ou l'ajout de fonctionnalités peut très sérieusement grever le budget alloué à un projet.
- Une plus grande acceptation du système par les utilisateurs. Cet aspect, parfois négligé par les autres démarches de conception est un véritable apport de la conception participative. Les utilisateurs étant au cœur du projet, les « résistances au changement » sont largement amoindries.
- Une plus grande participation des utilisateurs au processus de décision dans l'organisation. Cet aspect, sans doute le plus différenciant des démarches centrées utilisateurs est absolument central, nous y revenons dans l'un des paragraphes suivants.

Parmi les démarches de conception participative, le « user driven innovation » fait figure d'exemple. Cette démarche permettant de concevoir des systèmes innovants laisse une très large place aux utilisateurs, en les plaçant au centre de l'équipe de conception (Kushniruk et Nohr 2016). Ici le rôle des développeurs, sans passer au second plan, est dévolu aux tâches de support et de facilitation du processus. Les utilisateurs sont donc ceux à qui revient la prise de décision, afin de baser les innovations sur leurs besoins réels et sur leurs connaissances, auxquelles on reconnaît une grande valeur (Von Hippel 1995).

De manière plus générale, on peut observer que la conception participative se distingue nettement de la conception centrée utilisateurs en reconnaissant aux utilisateurs un véritable droit à la prise de décision, du fait de leur statut de coconcepteur (Françoise Darses 2009). Elle est en ce sens qualifiée de démocratique par certains car, en admettant la puissance de l'impact des systèmes d'informations sur le travail des utilisateurs et leur dimension sociale (Sanders et Stappers 2008), elle propose d'en faire un levier d'empowerment. Les utilisateurs peuvent pour cela participer à la définition du projet dans son ensemble lors des phases initiales (Dearden et Rizvi 2008), améliorant ainsi la connaissance de tous sur les systèmes d'information à construire (Kautz 2011) et participer activement au processus de décision concernant les choix de conception les impactant directement.

Ces démarches de conception participatives sont celles sur lesquelles nous nous appuyons dans la suite de notre travail. Elles permettent d'impliquer l'utilisateur tout au long du processus de conception, de les faire participer à la prise de décision. Nous présentons en conclusion de cette section les critères indispensables à retenir pour que ce type de démarche de conception puisse être un véritable levier d'empowerment démocratique des utilisateurs.

2.1.4. Le développement agile et la participation des utilisateurs

Dans cette partie de l'état de l'art, nous nous intéressons à l'implication des utilisateurs dans le processus de développement de systèmes d'information (cf. Figure 18). Nous nous centrons ici sur les méthodes agiles, puisqu'elles sont principalement celles qui proposent une implication des utilisateurs et/ou des clients dans leurs démarches. Cette dernière étape de la construction des systèmes d'information est une véritable possibilité de poursuivre l'implication enclenchée lors des phases de

recueil des exigences et de conception, afin d'assurer une participation globale de l'utilisateur dans la construction de son système d'information. Il est donc primordial pour nous de traiter de ce point.

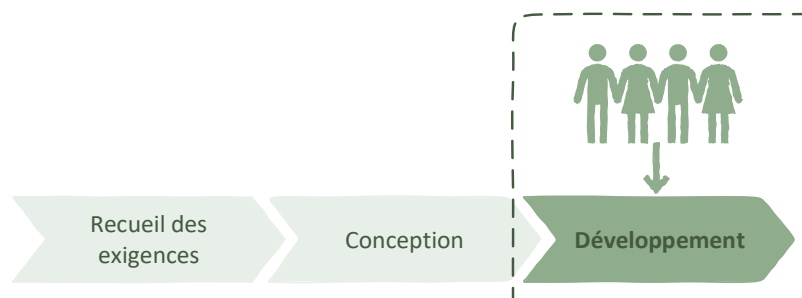


Figure 18 : L'implication des utilisateurs dans le développement des systèmes d'information

2.1.4.1. Présentation des valeurs et principes agiles

L'agilité et les pratiques agiles existent depuis les années 80, sans avoir pour autant été identifiées comme telles. L'introduction de cycles de développement courts et de réunions quotidiennes par exemple sont des pratiques agiles qui se sont largement répandues avant l'apparition du Manifeste Agile. Des méthodes telles que Scrum et Xtreme Programming (XP) existaient elles aussi avant 2001, et ont été « requalifiées » d'agiles suite à la parution du Manifeste (Aubry 2018). Des modes de management dits « anglo-saxons » privilégiant l'horizontalité et le partage des connaissances ont également été mis en œuvre avant que ne paraisse le Manifeste Agile (Vickoff 2003). L'agilité est donc un mouvement qui est né et s'est construit autour de pratiques innovantes et disruptives.

Le Manifeste Agile, co-signé en 2001 par 17 experts du développement logiciel a eu pour effet de permettre une identification plus claire des valeurs et principes agiles sous lesquels se sont par la suite rangés des méthodes de développement, des modes de management ou plus généralement des pratiques dans les organisations (Highsmith et Cockburn 2001). De façon méthodique, quatre valeurs et leur déclinaison en douze principes ont été proposées (Agile manifesto 2001) :

- Valeurs agiles :
 - Les individus et leurs interactions plus que les processus et les outils
 - Des logiciels opérationnels plus qu'une documentation exhaustive
 - La collaboration avec les clients plus que la négociation contractuelle
 - L'adaptation au changement plus que le suivi d'un plan
- Principes agiles :
 - Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
 - Accueillez positivement les changements de besoins, même tard dans le projet. Les processus agiles exploitent le changement pour donner un avantage compétitif au client.
 - Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
 - Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
 - Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
 - La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
 - Un logiciel opérationnel est la principale mesure d'avancement.

- Les processus agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
- Une attention continue à l'excellence technique et à une bonne conception renforce l'agilité.
- La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
- Les meilleures architectures, spécifications et conceptions émergent d'équipes autoorganisées.
- À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

Il est parfaitement visible que la conception centrée utilisateurs évoquée dans le chapitre précédent et l'agilité partagent un socle de valeurs communes (Chamberlain, Sharp, et Maiden 2006). Notamment, le fait de privilégier la collaboration ou encore l'adaptation au changement sont des valeurs agiles (Aubry et Beauregard 2013) qui peuvent s'appliquer à la conception centrée utilisateurs. Cependant, les méthodes agiles sont aujourd'hui très répandues dans les organisations, et leur mise en œuvre n'est pas toujours conforme aux valeurs et principes édictés (Bour 2018). Pour cette raison notamment, certaines méthodes agiles sont actuellement décriées (Jeffries 2018) par ceux qui en ont pourtant été les instigateurs. La section suivante nous permet d'évoquer trois des principales méthodes agiles : Scrum, XP et RAD, et d'en détailler les grands principes.

2.1.4.2. Présentation des méthodes agiles

Pour comparer les différentes méthodes agiles que nous avons souhaité présenter, nous avons choisi de faire une description organisée en grands thèmes plutôt que par méthode. Ainsi, nous discutons dans un premier temps de la constitution des équipes de chacune des méthodes. Nous présentons dans un deuxième temps la façon dont ces méthodes de développement sont outillées en termes de recueil des besoins et de conception puis nous décrivons les grands principes de développement de chacune.

Equipes agiles

La particularité des méthodes agiles est que, conformément aux valeurs et principes qui sont les leurs, elles impliquent toujours des représentants du besoin métier dans leurs équipes tout au long du processus de développement. SCRUM propose quatre rôles principaux au sein de l'équipe de développement (Schwaber et Beedle 2002) :

- Le Product Owner (PO), qui peut être traduit par « propriétaire du produit ». Ce PO est celui qui exprime le besoin fonctionnel, et qui est le garant de la conformité des développements avec les attentes du client. Le PO doit être disponible durant toute la phase de développement pour répondre aux questions et demandes, mais il n'est pas toujours présent.
- Le Scrum Master, garant du respect de la méthode, et facilitateur du projet. Il est présent tout au long du projet au sein de l'équipe.
- Les développeurs.
- Les « stakeholders », parmi lesquels on peut trouver des utilisateurs finaux ou leurs représentants.

XP propose une composition d'équipe similaire avec une particularité : tous les membres de l'équipe sont développeurs (sauf le client) mais peuvent assumer un rôle particulier durant le projet. L'équipe XP se compose de la façon suivante (Erickson, Lyytinen, et Siau 2005) :

- Le client, qui peut être un utilisateur final ou un utilisateur « clé ».
- Le coach, garant du respect de la méthode.
- Les développeurs.
- Le tracker, qui assure le suivi du projet au jour le jour.

RAD se distingue des deux méthodes précédentes en proposant une organisation d'équipe « en miroir » (Vickoff 1996) :

- D'un côté l'équipe utilisateurs, avec un chef de projet utilisateur et des utilisateurs.
- De l'autre l'équipe technique, avec un chef de projet technique et des développeurs.
- Un expert de la méthode peut également être présent afin, comme pour le Scrum Master et le coach, de s'assurer du bon déroulement de la méthode, et d'assurer un rôle de facilitateur.

On voit au travers de ces décompositions d'équipe que RAD est la seule méthode qui garantit la présence d'utilisateurs durant le processus. Scrum et XP restent plus vagues sur ce point, nous y reviendrons notamment dans la section 2.1.4.3.

Recueil des besoins et conception pour le développement agile

Le développement agile se base, selon les méthodes, sur une phase de recueil des besoins plus ou moins formalisée. SCRUM et XP s'appuient sur un backlog de produit, autrement dit un carnet de produit. Le backlog de produit n'est pas figé : il a son propre cycle de vie, et continue d'évoluer tout au long du projet, de façon totalement indépendante du développement. C'est dans ce backlog que sont recensés tous les besoins métier, et il n'existe aucune préconisation dans ces méthodes sur la marche à suivre pour le constituer. Ce backlog contient un ensemble de User Stories (Cohn 2004). Une User Story est décrite comme apportant « *de la valeur à une ou plusieurs parties prenantes qui utilisent le produit* » (Aubry et Beauregard 2013). La User Story est décrite en ces termes : *en tant que ... je veux ... afin de* C'est le Product Owner avec SCRUM et le client avec XP qui est en charge de la rédaction de ces User Stories, et qui décrit les tests fonctionnels qui permettront de les valider. La façon dont ces deux acteurs rédigent les User Stories n'est pas précisée : avec ou sans les utilisateurs ? Selon quelle organisation ? De plus, cette phase de recueil des besoins avec SCRUM et XP ne prévoit pas de débat autour de la raison d'être du projet ou de ses objectifs (Schön, Thomaschewski, et Escalona 2017).

RAD formalise nettement plus le recueil des besoins avec ses JRP (Joint Requirement Planning) (Wood et Silver 1995). Il s'agit en général de deux journées passées « à l'écart du monde » par l'ensemble de l'équipe projet, et dédiées à la rédaction des besoins utilisateurs. L'équipe technique est présente afin de s'imprégner et d'envisager d'ores et déjà la faisabilité du projet, mais c'est bien l'équipe d'utilisateurs qui décrit les besoins. Les JRP permettent d'avoir un cadrage du projet clair, et de discuter de ce qui doit être laissé de côté : les objectifs du projet sont exprimés et partagés par tous. Ceci étant, le déroulement des JRP n'est pas outillé ou décrit au travers d'une méthode formelle.

Les méthodes agiles sont des méthodes itératives et incrémentales. Ainsi, les phases de conception et développement portent sur des parties de projet, et se répètent jusqu'à l'obtention du produit final. Jeff Patton a illustré cela au travers de la création du célèbre tableau de La Joconde (Patton 2008) de manière itérative et incrémentale comme le montre la Figure 19 ci-dessous.

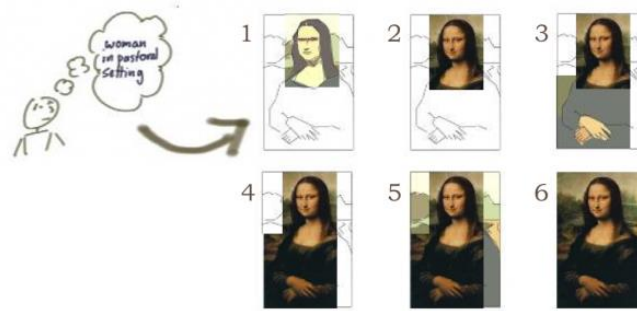


Figure 19 : Drawing Mona Lisa Iteratively and Incrementally (Patton 2008)

Avec SCRUM et XP, le projet est découpé en sprints, c'est-à-dire en itérations de durée fixe. Au début de chaque sprint, une partie du backlog de produit général est sélectionnée afin qu'elle fasse l'objet de développements durant le sprint (Sutherland et Schwaber 2011). Ce choix se fait en accord avec le représentant du métier. La conception des fonctionnalités à développer durant le sprint se fait en début de sprint, lors d'un « sprint planning meeting » durant lequel les User Stories choisies sont détaillées et spécifiées. Durant cette étape, le PO (pour Scrum) et le client (pour XP) sont présents, mais il y a une grande variabilité sur l'implication des utilisateurs à cette étape, qui dans tous les cas n'est pas obligatoire.

Avec RAD, la conception est faite par l'équipe technique lors des JAD (Joint Application Design). A l'inverse des JRP, c'est l'équipe technique qui a la main, mais l'équipe utilisateurs est présente pour ajouter des éléments lors de cette phase de conception si besoin. Tous les développeurs, regroupés dans l'équipe technique, sont présents lors de cette étape, qui se déroule à la manière des JRP (toute l'équipe est réunie, à l'écart du monde, etc.), mais sur un temps parfois moins long. L'intérêt de cette méthode est qu'une fois encore, elle ancre comme un principe le fait qu'équipe utilisateur et équipe technique participent à tout le processus (Vickoff 1996).

Développement agile

Le développement agile est au cœur du processus itératif et incrémental de ces méthodes. Avec SCRUM, l'organisation des sprints est très formalisée, avec notamment des préconisations organisationnelles. Parmi les rituels de SCRUM (Aubry 2018), on retrouve le « Daily Scrum », qui permet à l'équipe de se regrouper chaque matin pour organiser le travail de la journée, le « sprint review », qui correspond à une réunion de livraison des développements au PO et durant laquelle les fonctionnalités développées sont validées (ou non), et enfin la rétrospective, qui est une pratique durant laquelle l'équipe de développement cherche à s'améliorer. La visibilité offerte par ce type d'organisation sur la conception et le développement est très intéressante, car elle permet de garantir une intégration du représentant du métier au travers de rituels formalisés.

Avec XP, il y a moins de préconisations organisationnelles, mais plus de préconisations techniques pour le développement puisqu'il s'agit d'une méthode de développement dirigée par les tests (Beck 2003). L'intégration continue est un principe au cœur de cette méthode qui a une orientation très « technologique ». Le client est inclus durant les phases de test, c'est la raison pour laquelle il doit être sans cesse présent au sein de l'équipe.

Avec RAD, le développement est basé sur un principe de prototypage. Des points réguliers sont faits avec l'équipe utilisateurs, qui grâce à ses retours permet un affinage des fonctionnalités. Cela peut donner lieu à de nouvelles sessions de JAD, afin de retravailler la conception, mais on ne remonte jamais jusqu'au niveau des JRP, c'est-à-dire du cadrage du projet.

Certains, afin d'accentuer l'inclusion des utilisateurs dans le processus de développement agile, proposent notamment d'intégrer aux projets agiles des éléments d'ergonomie constructive (inclusion d'un ergonomiste et des utilisateurs) afin de faire évoluer le système (Schwartz et al. 2009). On ajoute alors à l'amélioration de l'utilité proposée par l'agilité (au travers d'une couverture fonctionnelle plus précise) l'amélioration de l'utilisabilité du système (Silva da Silva et al. 2011). D'autres poursuivent le même but en argumentant de manière plus générale en faveur d'une meilleure collaboration entre utilisateurs et praticiens dans le développement agile (Ferrario et al. 2014).

2.1.4.3. Le glissement sémantique de l'agilité : de l'utilisateur au client

Un dernier élément d'analyse nous permet d'explorer l'aspect participatif de l'agilité : la sémantique des termes client et utilisateur. Alors que la méthode agile considérée comme étant la plus ancienne – RAD – préconise une participation des utilisateurs auprès des équipes de conception tout au long du processus, les méthodes plus récentes ont opéré un glissement sémantique en substituant le terme d'utilisateur pour celui de client.

Dans le Manifeste Agile dans un premier temps, on peut observer que la version française traduit fidèlement les deux termes anglais de *customer* : client et de *user* : utilisateur. Le premier terme est réservé aux aspects stratégiques du projet avec la valeur « *La collaboration avec les clients plus que la négociation contractuelle* » et le principe « *Notre plus haute priorité est de satisfaire le client [...]* ». Le second est réservé lui aux aspects plus opérationnels avec les principes « *Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet* » et « *[...] ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant* ». Le Manifeste Agile commence dès lors à s'éloigner d'une vision totalement participative telle que proposée avec RAD en n'incluant pas l'utilisateur dans les aspects stratégiques du projet. Or nous avons vu dans la description des caractéristiques d'un système d'information démocratique que l'aspect stratégique portant sur les représentations, les valeurs du système et sur la participation à la prise de décision devait être traité avec le concours des utilisateurs eux-mêmes.

Le glissement sémantique s'est poursuivi plus récemment avec la méthode aujourd'hui la plus populaire : SCRUM (Aubry et Beauregard 2013). Alors que les termes d'utilisateur et de client sont parfois utilisés, il leur est préféré celui de Product Owner. Et la sémantique associée à ce terme reste des plus vagues. En effet, si tous s'accordent à considérer que le PO est le responsable de ce qui doit être fait (Sverrisdottir, Ingason, et Jonasson 2014), son rôle de représentant n'est pas toujours clairement défini. Pour certains, il est le client, ou le représentant du client (Cervone 2011), pour d'autres il endosse le rôle de représentant de toutes les parties prenantes (Pichler 2010). Il apparaît clairement ici que cette distinction n'est pas sans conséquence. Etant donné le rôle central du Product Owner dans les projets agiles SCRUM, le fait qu'il ne soit « que » le représentant du client peut l'éloigner d'un objectif d'inclusion des utilisateurs au processus de développement. A contrario, si ce Product Owner est un représentant de toutes les parties prenantes, et donc notamment des utilisateurs, la possibilité de continuité entre SCRUM et la conception participative par exemple apparaît plus naturelle.

L'agilité, dans ses principes et valeurs, nous semble être une approche participative satisfaisante. Elle permet d'impliquer les utilisateurs jusque dans le développement de leur système d'information, ne les désengageant pas ainsi de cette partie plus « technique ». Ceci étant, toutes les méthodes agiles ne se valent pas quant à l'implication des utilisateurs eux-mêmes, et il convient de rester vigilant sur la participation des utilisateurs aux projets, et sur leur non substitution par les clients ou autres PO lorsqu'ils ne sont pas clairement désignés comme des représentants des utilisateurs.

2.1.5. Notre analyse et notre positionnement

Le recueil des besoins est une phase majeure de la construction des systèmes d'information. C'est en effet à cette étape que les valeurs et visions du monde des utilisateurs s'expriment. Parmi les méthodes existantes, certaines telles que Value Sensitive Design portent une attention particulière à l'expression de ces valeurs via une implication des utilisateurs, mais la plupart manquent d'un outillage suffisant pour assurer une traçabilité de ces valeurs dans tout le processus de conception et de développement qui suit. On peut ajouter à cela que seules les méthodes de développement agiles nous permettant d'envisager des possibilités de remise en question des besoins exprimés lors de la phase de recueil des exigences, mais ici encore avec un manque d'outillage et de formalisme.

Parmi les démarches de conception que nous avons citées, différents degrés d'implication des utilisateurs apparaissent. Les démarches centrées utilisateurs, dont l'ergonomie constructive ou la conception collaborative par exemple, attribuent à l'utilisateur un rôle informationnel et/ou consultatif. La qualité du système développé s'en trouve améliorée, en termes d'utilisabilité notamment, mais on ne peut pas parler d'empowerment démocratique de ces derniers. Les approches participatives accordent quant à elles aux utilisateurs un rôle central, avec un véritable statut de coconcepteur, et leur empowerment passe notamment par une participation claire au processus de décision et à la définition initiale des objectifs du projet (Dearden et Rizvi 2008).

Pour approfondir la question des démarches participatives, nous avons fait un focus sur la démarche nommée « user driven innovation » (Kanstrup et Bertelsen 2016) et nous pouvons à présent identifier une limite majeure de ce type de démarche : le focus qui est fait sur des utilisateurs « leaders » (Von Hippel 1995). Bien que nous nous positionnions dans le domaine des démarches participatives, cette notion de participants « leader » doit être absolument précisée afin de respecter notre vision d'une démarche participative. Pour cela, nous avons fixé un certain nombre de critères, inspirés de Kautz (cf. Table 1) :

- La participation doit inclure des utilisateurs à titre individuel, et pas des utilisateurs « types » qui représenteraient une vision fictive et peu précise.
- La participation des utilisateurs doit être directe. Si elle ne peut l'être en raison de la taille du projet notamment, la désignation de représentant doit se faire par une voie démocratique (élection ou tirage au sort). Nous y reviendrons en détail dans le chapitre 4.
- Les utilisateurs doivent avoir un rôle véritablement actif et participatif. Leur présence n'est pas informationnelle ou consultative : ils doivent participer au processus de décision.
- L'objectif de la méthode ne doit pas être réduit à une amélioration de l'utilité et de l'utilisabilité du système, mais correspond à un véritable choix politique : celui de proposer aux utilisateurs un empowerment démocratique.

Table 1 : Grille d'analyse de la participation des utilisateurs (Kautz 2011)

Types de participants	Utilisateur en personne Utilisateur représentatif Utilisateur fictif
Formes de participations	Participation directe Participation indirecte
Rôles des participants	Rôle informatif Rôle consultatif Rôle participatif
Objet de la participation	Empowerment fonctionnel Empowerment démocratique

La méthode que nous proposons n'est pas une méthode agile au sens stricte du terme, mais est compatible avec les valeurs et les principes de l'agilité tels que décrits dans le Manifeste (Agile Manifesto 2001). Nous insistons particulièrement sur le fait que ce sont bien les utilisateurs finaux que nous considérons comme indispensables, et que ce sont donc eux qui doivent faire partie de l'équipe durant les cycles de développement itératifs et incrémentaux. Ici encore, afin de prolonger le travail de conception, les utilisateurs doivent avoir une véritable voix dans le processus de décision concernant le développement de l'outil qui les concerne. Pour cela, concepteurs/développeurs et utilisateurs doivent parvenir à une « intelligibilité mutuelle » (Francoise Darses 2004) assurant aux utilisateurs une compréhension globale du projet pour être en capacité de prendre des décisions.

L'état de l'art sur l'agilité nous a permis de pointer les manques des méthodes, et nous permet de positionner notre travail quant à la phase de développement. A la manière de RAD, il nous semble intéressant de lier équipe technique et équipe utilisateur dès le recueil des exigences, mais en formalisant davantage la méthode, et en l'outillant avec les User Stories de SCRUM et XP par exemple. De plus, le développement itératif et incrémental tel que proposé par Scrum avec ses rituels nous semble être une véritable opportunité d'impliquer l'utilisateur jusqu'à cette étape technique de développement, assurant ainsi une continuité avec les phases de recueil des besoins et de conception participatives.

Nous avons décrit pourquoi et comment les utilisateurs pouvaient être impliqués dans la construction de leur système d'information, conformément aux préconisations que nous avons faites sur l'élaboration de systèmes d'information démocratiques. Bien que les motivations soient diverses, celles à laquelle nous nous intéressons particulièrement est la question de l'empowerment démocratique des utilisateurs.

Sur toutes les phases du processus de construction de systèmes d'information, nous avons pointé les différentes propositions existantes quant à l'implication des utilisateurs. Pour la première phase de recueil des exigences, nous avons pu voir que la participation des utilisateurs était cruciale, et qu'elle était un sujet peu abordé dans la littérature. Nous avons quand même identifié des méthodes participatives de recueil des exigences, qui nous semblent devoir être outillées méthodiquement. Pour la seconde phase qu'est la conception de systèmes d'information, des méthodes centrées utilisateur ont ouvert la voie à des méthodes plus participatives dans lesquelles notre proposition nous semble pouvoir s'inscrire. Enfin, pour ce qui est de la phase de développement de système d'information, les valeurs et principes agiles nous semblent pouvoir guider des méthodes de développement participatives dans lesquelles les utilisateurs conservent la place privilégiée qui leur aura été proposée durant le recueil des exigences et la conception.

Ces éléments de la littérature sont pour nous un éclairage sur la première question que nous abordons : le processus démocratique de création de système d'information. La seconde question que nous adressons maintenant est celle de la prise en compte d'une pluralité de points de vue dans un système d'information soutenant la démocratie dans l'organisation.

2.2. Le concept de point de vue en conception de systèmes d'informations

La notion de point de vue est au centre de nos réflexions dans cette thèse. La démocratie en termes d'éthique informatique étant considérée comme la garantie de l'accès à une pluralité de points de vue, il nous semble indispensable de proposer de construire des systèmes d'information en faisant émerger ces points de vue, en les prenant en compte et les intégrant au système lui-même. La notion de point de vue s'entend pour nous comme « point de vue utilisateur ». En effet, la pluralité des points de vue est représentée par les différents points de vue dans l'organisation, incarnés par les utilisateurs. Pour reprendre l'exemple de l'hôpital auquel nous avons fait référence dans la genèse du système d'information (section 1.2.2), l'on peut aisément comprendre que le point de vue d'un gestionnaire se distingue très nettement de celui du personnel soignant par exemple. Ils sont pourtant des utilisateurs du même système d'information, mais les représentations, valeurs et visions du monde qu'ils ont peuvent être différentes.

Cette deuxième partie d'état de l'art est donc consacrée à la question des points de vue en conception de système d'information. Nous nous penchons dans un premier temps sur cette notion dans la phase de recueil des exigences, afin de comprendre comment sont considérés les points de vue à cette étape stratégique, et de quelle façon il est possible de les faire émerger. Nous regardons dans un second temps comment ces points de vue, s'ils ont été identifiés, peuvent être pris en compte dans la conception du système d'information lui-même. Nous identifions alors quelques propositions de modélisation conceptuelle qui adressent cette question. Enfin et dans un dernier temps, nous étudions les limites de ces propositions, et nous positionnons par rapport à celles-ci (cf. Figure 20).

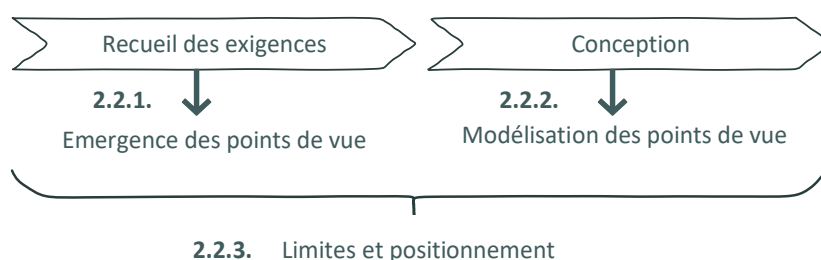


Figure 20 : Plan de la partie 2.2

2.2.1. La question du point de vue en ingénierie des exigences

Nous avons évoqué plus haut les processus de recueil des exigences participatifs. L'implication des utilisateurs qu'ils proposent est certaine, mais un point commun unit toutes ces démarches que nous avons citées : les auteurs ne font pas de distinction au sein du groupe des utilisateurs. En effet, l'importance de l'identification des différentes parties prenantes est fréquemment mise en avant (Sharp, Finkelstein, et Galal 1999), mais relève d'une distinction entre groupes : clients, opérateurs, architectes, développeurs et utilisateurs, comme proposée dans le standard de l'IEEE (IEEE Computer Society 1996). Les utilisateurs sont un groupe parmi les autres, qui semble être considéré de façon homogène.

Une exception cependant est à noter dans le travail de Sharp, qui opère un premier niveau de distinction en séparant les utilisateurs directs (qui interagiront avec le logiciel) et les utilisateurs indirects (qui auront besoin des produits du système) (Sharp, Finkelstein, et Galal 1999). Mais cette distinction reste à modérer car bien que la notion d'utilisateurs directs se rapproche effectivement de la définition des utilisateurs finaux, il n'en reste pas moins qu'ici encore ceux-ci sont considérés comme un groupe homogène.

L'utilisation du terme « point de vue » recouvre lui aussi des définitions diverses lorsqu'il est utilisé dans le cadre de l'ingénierie des exigences. Pour Kotonya par exemple, cette notion renvoie à celle de client du système, utilisant la métaphore de l'architecture client-serveur des systèmes (Kotonya et Sommerville 1996). Sommerville quant à lui propose des arguments en faveur d'une approche par point de vue en ingénierie des exigences, et adresse cette question dans une perspective de séparation des problèmes entre les parties prenantes. Il considère lui aussi les utilisateurs comme un groupe homogène, et les distingue des autres parties prenantes que sont les managers de l'organisation, les systèmes informatiques, les entités externes et les ingénieurs impliqués dans la conception, le développement et la maintenance du système (Sommerville et Sawyer 1997).

Alors que la phase de recueil des exigences devrait être selon nous une phase d'émergence de points de vue au démarrage du projet, les méthodes participatives d'ingénierie des exigences ne proposent pas de mettre en relief les points de vue au sein même du groupe utilisateur. Il nous semble donc que sur ce point de nouvelles propositions doivent être faites, afin de faire de cette étape une véritable opportunité d'émergence des points de vue.

2.2.2. La modélisation des points de vue

La conception logicielle passe avant tout par une transcription des besoins exprimés, de façon plus ou moins participative, en modèles conceptuels. L'activité de modélisation consiste en une cartographie générale de tous les aspects du système d'information : données, processus, flux, etc. Nous nous intéressons particulièrement dans cette partie à la modélisation des données, activité pour laquelle la notion de point de vue est pour nous centrale. Le point de vue, regroupant des visions, des représentations et des valeurs communes à un groupe d'utilisateurs se traduit en effet de manière directe dans le vocabulaire employé. Pour continuer de filer la métaphore d'un système d'information hospitalier, on peut imaginer que le terme de qualité par exemple ne renvoie pas aux mêmes valeurs et visions selon que l'on se place d'un point de vue gestionnaire ou d'un point de vue soignant. Ce vocabulaire : les mots employés, sont ceux que l'on retrouve dans les dictionnaires de données qui constituent une partie du système d'information, et qui sont modélisés au travers de modèles de données.

Les concepteurs se conforment à un certain nombre de standards dans leur activité de modélisation, parmi lesquels on retrouve les métamodèles. Une définition formelle de ce que sont les métamodèles nous est donnée par l'Object Management Group (OMG) : les logiciels sont spécifiés par des modèles, qui sont eux-mêmes des instances de métamodèles. En général, la notion de point de vue dans ces métamodèles est abordée selon deux angles :

- D'une part le point de vue peut désigner le métamodèle utile à chaque étape du développement logiciel. Dans le cas d'une modélisation des données dite relationnelle par exemple, les points de vue seront compris au sens de niveaux de modélisation : modélisation conceptuelle, puis logique, puis physique.
- D'autre part la notion de point de vue peut renvoyer à celle de rôle dans le projet informatique. Dans ce cas-ci, alors que certains modèles seront orientés vers un point de vue de concepteur, d'autres seront adaptés aux points de vue des architectes ou des développeurs (Françoise Darses, Détienne, et Visser 2001).

Ici encore, la distinction des points de vue divergents au sein du groupe d'utilisateurs n'est pas évoquée. L'usage d'un modèle unique pour tous les utilisateurs est habituel, c'est l'une des caractéristiques du diagramme de classe par exemple (Berardi, Calvanese, et De Giacomo 2005). Le modèle obtenu par les concepteurs est valable pour tous les utilisateurs du futur système, mais cette

apparente « simplicité » cache en fait des cas de figure qui peuvent s'avérer plus complexes. En effet, si la phase d'élicitation des exigences évoquée dans la partie précédente n'a pas fait émerger de points de vue distincts au sein du groupe utilisateurs, cette vision peut être validée. En revanche, si des points de vue différents sur les objets ont été révélés au cours du processus, l'usage de la notion d'héritage s'avère alors indispensable pour les représenter, complexifiant ainsi grandement le modèle final. Comme le décrit Dupuy-Chessa (Dupuy-Chessa et al. 2014), outre une complexité hiérarchique certaine, le modèle peut alors devenir très hétérogène et être d'un volume extrêmement important. Son appropriation par les utilisateurs dans une perspective participative est alors largement compromise.

Pour ces raisons, quelques propositions alternatives de modélisation des points de vue ont émergé. Parmi elles, les systèmes d'information orientés composants gèrent la modélisation des points de vue par une décomposition en composants de chacune des « vues » (Caron, Muller, et Vanwormhoudt 2003). Cette proposition n'a pas pour but de faciliter la prise en compte de points de vue utilisateurs, suite à une participation accrue de ces derniers au processus de conception, mais bien de résoudre une problématique technique. Les décompositions et la complexité hiérarchique que représentent ces modèles les rendent absolument incompréhensibles, et très rigides durant le cycle de vie du logiciel.

Nassar quant à lui propose une adaptation des standards UML par l'ajout de la notion d'extension de vues (Nassar 2003). Ici encore, cette solution est à envisager comme une réponse à un problème d'ordre technique concernant les droits d'accès et n'est en ce sens pas orientée utilisateurs. L'objectif est de séparer les rôles dès cette phase de modélisation des données, mais ne propose pas de prendre en compte réellement des points de vue différents sur un même objet en termes de sémantique.

Cet état de l'art des types de modélisation dites « par point de vue » nous montre que les propositions de modélisation conceptuelle actuelles ne répondent pas à la question du point de vue telle que nous l'entendons. Lorsqu'elles concernent les points de vue utilisateurs, elles sont une réponse à des problématiques techniques de droits d'accès avant tout. Le reste du temps, elles interprètent la notion de point de vue comme une séparation des rôles des acteurs d'un projet informatique, ou dans une perspective de décomposition chronologique. Notre vision des points de vue utilisateurs comme autant de façon d'envisager les représentations et valeurs qui sous-tendent un système d'information n'est pas contenue dans les propositions actuelles. Les systèmes d'information démocratiques, qui seraient une garantie de respect d'une pluralité de points de vue doivent donc faire l'objet de nouvelles propositions.

2.2.3. Les limites identifiées et notre positionnement

Dans ce second état de l'art nous avons proposé d'adresser deux problématiques distinctes : celle de la notion même de point de vue utilisateur en ingénierie des exigences et celle de la modélisation de ces points de vue durant la phase de conception de système d'information. Nous avons observé dans un premier temps que l'ingénierie des exigences, bien qu'ayant pris le tournant de la participation des utilisateurs, notamment dans l'ingénierie orientée par les buts, n'établit pas de distinction claire entre les utilisateurs, et les considère comme un groupe homogène. Cette vision n'est pas la nôtre, puisque l'acceptation de la prise en compte d'une pluralité de points de vue est une condition sine qua none de la démocratie dans le système d'information, comme présenté dans le chapitre 1. Les utilisateurs, c'est-à-dire ceux impliqués par la mise en œuvre du système doivent alors être à même d'exprimer des voix discordantes dans le processus d'ingénierie des exigences.

Dans un second temps, nous avons pu observer que la modélisation des points de vue utilisateurs était un problème insuffisamment traité en conception de systèmes d'information. Si les rares solutions

proposées peuvent s'avérer convenables d'un point de vue technique, elles ne sont pas du tout orientées utilisateur et ne peuvent donc pas nous donner satisfaction. En effet, notre parti pris étant celui de la démocratie, l'émergence et la modélisation de points de vue ne saurait dans notre cas donner lieu à des modèles incompréhensibles pour les utilisateurs eux-mêmes.

Nous avons proposé dans cette seconde partie d'état de l'art d'identifier de quelle façon la question du point de vue était prise en compte dans la conception de systèmes d'information : de l'élicitation des exigences à la modélisation conceptuelle des données. Dans l'ensemble, notre vision du point de vue n'est pas celle portée par les méthodes analysées, et il nous semble donc indispensable de proposer d'autres façons de considérer ces points de vue.

D'une part, l'élicitation des exigences dans un processus de construction de système d'information démocratique doit pour nous impérativement considérer les utilisateurs comme un groupe hétérogène avec des points de vue divergents. D'autre part, la modélisation conceptuelle doit être à même de prendre en compte des points de vue utilisateurs, tout en garantissant des modèles compréhensibles par les utilisateurs eux-mêmes.

2.3. Conclusion du chapitre

Notre vision du système d'information soutenant la démocratie des organisations est portée par plusieurs principes. Ceux-ci peuvent être regroupés en deux grandes catégories : d'une part le système d'information se construit sur la base de la participation des utilisateurs, depuis la phase d'élicitation des exigences jusqu'au développement, en passant par la conception. D'autre part, le système d'information prend en compte une pluralité de points de vue utilisateurs.

Nous avons organisé cet état de l'art autour de ces deux grands principes. La première partie était donc consacrée à l'implication des utilisateurs au processus de construction de son système d'information. Si nous avons pu identifier des méthodes de conception participatives répondant à nos besoins, il nous semble que les démarches de recueil des exigences et de développement agile devaient être largement précisées pour correspondre aux caractéristiques des systèmes d'information démocratiques.

La seconde partie était consacrée à un état de l'art sur les points de vue, depuis leur émergence lors de la phase de recueil des exigences jusqu'à leur prise en compte dans les modèles de données de la phase de conception. Sur ces deux points il nous a semblé que les propositions actuelles étaient largement insuffisantes pour nous donner satisfaction quant au respect d'une pluralité de point de vue dans un système d'information démocratique.

Les propositions que nous ferons dans les chapitres consacrés à DEMOS seront basées sur notre analyse du présent état de l'art, et sur les caractéristiques d'un système d'information démocratique que nous avons listées dans le chapitre précédent.

Chapitre 3

ETAT DE L'ART – INGENIERIE DES METHODES

Notre contribution, au travers de cette thèse, est une proposition de méthode de conception de systèmes d'information démocratique nommée DEMOS. Les principaux objectifs de cette méthode sont ceux que nous avons identifiés comme soutien de la démocratie dans les organisations dans le chapitre 1 : une démarche de construction participative et démocratique du système d'information, et la prise en compte d'une pluralité de points de vue dans le système d'information lui-même.

Avant de présenter en détail cette méthode dans la partie II, nous avons choisi de nous arrêter sur deux points structurant selon nous : qu'est-ce qu'une méthode ? et comment construit-on une méthode ? Ces questions sont traitées dans le domaine de l'ingénierie des méthodes, et largement détaillées au travers de questions beaucoup plus précises sur les modèles sous-jacents aux méthodes, les techniques et outils et le processus d'ingénierie des méthodes d'ingénierie des systèmes d'information. Nous proposons dans ce chapitre un état de l'art de la discipline de l'ingénierie des méthodes qui, sans être exhaustif, nous permet de nous positionner quant aux choix méthodologiques que nous avons faits. Nous pourrions ainsi dans le chapitre suivant donner nos cadres de référence pour la construction de notre méthode.

3.1. Méthode d'ingénierie des systèmes d'information

Avant de présenter le processus d'ingénierie d'une méthode d'ingénierie des systèmes d'information, il nous a semblé primordial de définir en détail ce qu'était une méthode. Mais nous ne saurions commencer cette partie sans nous placer à nouveau dans le cadre de la démocratie. Cela nous impose de ne pas considérer la méthode sous un angle normalisant ou restrictif, qui empêcherait tout débat ou toute remise en cause, mais bien de voir en une méthode la possibilité de guider la construction du système d'information. En ce sens, nous nous accordons avec les propos de Salles pour qui « *une méthode est avant tout un soutien et non un cadre normatif* » (Salles 2013).

Nous définissons le terme de méthode avec ce présupposé-là dans la première section. Nous analysons ensuite de quoi est composée une méthode, en proposant une distinction claire entre deux éléments : le ou les modèles de produit et le ou les modèles de processus, dans les deux sections suivantes. Enfin nous faisons un focus particulier sur les plateformes support aux méthodes (cf. Figure 21).

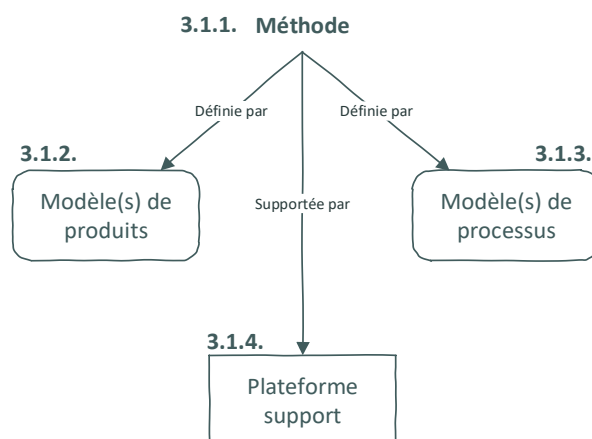


Figure 21 : Plan de la partie 3.1

3.1.1. Qu'est-ce qu'une méthode ?

Brinkkemper définit une méthode comme une démarche pour réaliser un projet de développement de système. Selon lui, la méthode est basée sur un cheminement de pensée spécifique composé de directives et de règles, et structuré de manière systématique dans des activités. (Brinkkemper et Harmsen 1998). Retenons de cette définition qu'une méthode est dynamique, qu'elle formalise une démarche, un processus, et qu'elle est structurée de façon à la rendre reproductible.

Les buts d'une méthode d'ingénierie des systèmes d'information peuvent être regroupés sous quatre principaux items, proposés par Jolita Ralyte (Ralyte 2006) :

- Dans un premier temps, une méthode doit aider les ingénieurs à comprendre le domaine d'application du projet.
- Dans un deuxième temps, la méthode permet de spécifier les besoins que le nouveau système d'information devra satisfaire.
- Dans un troisième temps, la méthode doit aider à concevoir et réaliser une solution informatique en adéquation avec les besoins en proposant des modèles. Ces modèles sont une façon de représenter les différentes vues et perspectives du système d'information en construction.
- Dans un dernier temps, la méthode est un vecteur de communication entre les différentes parties prenantes du projet de construction d'un système d'information : ingénieurs, futurs utilisateurs, clients, équipes de développement (dont les concepteurs et les développeurs).

Ces objectifs sont ceux que l'on peut attribuer aux méthodes en général. Des objectifs spécifiques à chaque méthode viennent bien entendu préciser chacun de ces éléments, et compléter les objectifs généraux assignés à la méthode proposée.

De manière formelle une méthode se décompose en deux aspects ayant trait à l'ingénierie : le produit et le processus. Ces deux composantes, que Brinkkemper appelle des techniques, recouvrent aussi bien les aspects représentationnels que les aspects procéduraux d'une méthode (Brinkkemper et Harmsen 1998). Une méthode comporte donc « *un ou plusieurs modèles de produits et un ou plusieurs modèles de processus* » (Ralyte 2006), ce sont ces aspects que nous détaillons dans les deux parties suivantes.

3.1.2. Les modèles de produit

Les modèles de produit sont la première composante d'une méthode. Ils permettent de formaliser les aspects représentationnels de la méthode. Dans cette partie nous définissons dans un premier temps ce que l'on appelle un produit et un modèle de produit, puis nous complétons cette définition en la précisant grâce aux concepts de syntaxe abstraite, syntaxe concrète et sémantique.

3.1.2.1. Définition du modèle de produit

Jolita Ralyte propose la définition du produit suivante : « *Le produit est le résultat d'application d'une méthode, c'est la cible désirée d'un développement de SI. [...] Un produit est exprimé dans les termes d'un modèle de produit* » (Ralyte 2006). Le produit obtenu à chaque application de la méthode pour un projet de construction de système d'information est donc l'instance d'un modèle de produit (cf. Figure 22). C'est au travers du modèle de produit que l'on formalisera les caractéristiques attendues du produit.

La littérature à ce sujet nous indique qu'une méthode peut faire l'objet d'un ou plusieurs modèles de produit. Dans les deux cas, plusieurs décompositions du modèle de produit sont proposées, afin de modéliser l'ensemble du système d'information :

- Décomposition selon les facettes du système d'information : statique, dynamique et fonctionnel.
- Décomposition selon le niveau de granularité, par exemple : classe, paquetage et sous-systèmes.
- Décomposition par niveau d'abstraction, par exemple : objet, classe, méta-classe.
- Décomposition par type de besoins : fonctionnels, non fonctionnels, intentionnels.

Le ou les modèles de produit doivent être représentés en s'appuyant sur un métamodèle. Ce métamodèle est donc naturellement appelé métamodèle de produit (cf. Figure 22). Il devra être adapté à la décomposition choisie et aux éléments qu'il doit permettre de représenter.

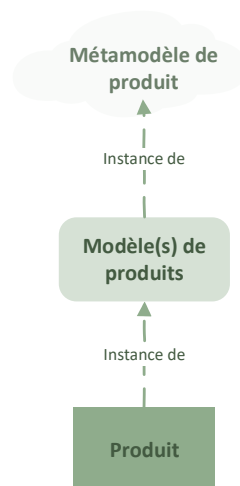


Figure 22 : Niveaux d'abstraction relatifs au produit

La notion de modèle de produit peut être rapprochée de celle du langage. En effet, le modèle de produit formalise le vocabulaire utilisé dans la mise en œuvre de la méthode et les concepts manipulés. Pour cette raison, nous avons choisi de rapprocher des travaux sur les modèles de produit d'autres

travaux portant sur les syntaxes des langages de modélisation, c'est ce que nous traitons dans la partie suivante.

3.1.2.2. **Syntaxe abstraite et syntaxe concrète**

Le modèle de produit est en quelque sorte le langage de modélisation de la méthode. En règle générale, les langages de modélisation sont définis par trois éléments : la syntaxe abstraite, la syntaxe concrète et la sémantique. Le fait de séparer ces trois éléments est une façon de « *gérer la complexité* » de la définition d'un langage de modélisation, car il s'agit ici de traiter d'une part les éléments du langage, et d'autre part leur représentation (Dupuy-Chessa et al. 2014). Nous avons donc fait le choix de décrire le modèle de produit en reprenant les éléments de syntaxe proposés par Dupuy Chessa dans ses travaux sur les langages de modélisation des systèmes d'information.

La syntaxe abstraite et la sémantique

« *La syntaxe abstraite capture le vocabulaire et la taxonomie (i.e. les concepts) du langage* ». (Dupuy-Chessa et al. 2014). En suivant le cheminement que nous avons adopté jusqu'alors, la syntaxe abstraite est donc l'un des éléments permettant de décrire le langage de modélisation du produit que nous cherchons à obtenir en appliquant une méthode d'ingénierie des systèmes d'information (Fondement et Baar 2005).

La syntaxe abstraite est complétée par la sémantique, qui permet aux utilisateurs du langage une compréhension précise et complète de celui-ci (Fondement et Baar 2005). Le choix de description de cette sémantique est libre mais doit respecter un objectif de clarté. Un dictionnaire des données décrivant chaque élément du langage peut, par exemple, être la solution retenue.

Krogstie a dressé une liste des caractéristiques que doivent respecter les langages de modélisation des méthodes de conception de systèmes d'information, et que nous avons synthétisée de la manière suivante (Krogstie 2003) :

- Le langage doit être en adéquation avec le domaine métier. Lorsque l'on développe un système d'information support aux processus métier, il convient par exemple d'utiliser les termes de l'organisation tels que processus, procédures, activités, etc. Cette précision, bien qu'elle puisse sembler simpliste, n'est pas inutile compte tenu du fait que les langages de modélisation sont le plus souvent utilisés par des concepteurs ayant une connaissance plus ou moins précise du domaine métier.
- Le langage doit être en adéquation avec les connaissances des concepteurs du domaine. Dans le cas où la démarche de conception est participative, c'est-à-dire où les utilisateurs sont considérés comme des coconcepteurs, il s'agit de proposer un langage manipulable par les concepteurs-utilisateurs. Le langage doit aussi être compris de toutes les parties prenantes. Il s'agit d'un véritable enjeu car les parties prenantes d'une démarche, notamment participative, ont un bagage fonctionnel et/ou technique qui peut être très différent. Un véritable effort dans la conception du langage de modélisation doit être fait afin de satisfaire ce critère.
- Le langage doit être en mesure de capter « toutes » les connaissances des participants (de les « *externaliser* » (Krogstie 2003)), y compris celles qui sont implicites ou peu formelles.

La syntaxe concrète

« *La syntaxe concrète décrit la notation, c'est à dire la représentation des éléments du langage* » (Dupuy-Chessa et al. 2014). Cette syntaxe peut être graphique ou textuelle, et vient compléter la syntaxe abstraite ; il s'agit de rendre les concepts du langage manipulables dans le cadre de la méthode. La syntaxe concrète permet en quelque sorte de « capter » la syntaxe abstraite pour les

différentes parties prenantes, et permet éventuellement de spécifier les représentations sur une plateforme support à la méthode, comme nous le verrons dans la partie suivante.

La syntaxe concrète peut être définie au travers d'un schéma d'affichage, directement relié à la syntaxe abstraite. Ainsi, chaque élément de la syntaxe concrète apporte une précision sur un élément de la syntaxe abstraite (Fondement et Baar 2005). La pertinence de la syntaxe concrète peut faire l'objet d'une évaluation, c'est ce que propose Moody au travers des neuf critères qu'il a établis (Moody 2009), et que nous détaillons dans la Table 2 :

Table 2 : Les neuf principes de Moody (Zehnder et al. 2017)

Principe	Description du principe
Clarté sémiotique	Il doit y avoir autant d'éléments dans la syntaxe abstraite que dans la syntaxe concrète. La correspondance entre chacun des éléments de syntaxe 1:1.
Discriminabilité perceptuelle	Les différentes représentations graphiques, autrement dit les éléments de notation doivent se distinguer les uns des autres.
Transparence sémantique	Les éléments de notation doivent suggérer le sens réel de l'élément de syntaxe abstraite auquel ils sont reliés.
Management de la complexité	Les notations doivent permettre de gérer la complexité grâce à des mécanismes tels que la hiérarchisation ou la modularisation.
Intégration cognitive	L'usage des éléments de notation visuelle doit être guidé par des mécanismes tels que les annotations, les cartes de navigation, etc.
Expressivité visuelle	Les notations doivent utiliser les possibilités liées aux variables graphiques pour être expressives : taille, couleur, position, etc.
Double codage	Du texte peut venir compléter la notation visuelle.
Economie graphique	Le nombre d'éléments de la notation doit être appréhendable (gérable cognitivement) par les participants de la méthode.
Ajustement cognitif	La notation graphique doit pouvoir s'adapter à différents publics.

La syntaxe abstraite, la syntaxe concrète et la sémantique d'un langage de modélisation sont les éléments constitutifs d'un modèle de produit (cf. Figure 23). Chacune de ces parties respecte le formalisme choisi par le concepteur de la méthode, et peut être évaluée séparément. Une évaluation globale permet ensuite de vérifier que chacun des éléments est conforme aux autres, afin de s'assurer de la plus grande cohérence lors de la conception du modèle de produit.

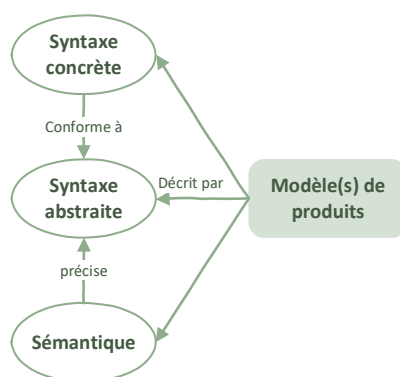


Figure 23 : Décomposition du modèle de produit

3.1.2.3. Synthèse sur les modèles de produit

Les modèles de produit sont au cœur d'une méthode de conception. Ils permettent de préciser le formalisme permettant d'obtenir le produit de la méthode, autrement dit les modèles implémentables pour développer le système d'information. Ces modèles sont conformes à un métamodèle dont ils sont l'instance, et le produit est lui-même l'instance du modèle de produit proposé par la méthode.

Le modèle de produit, véritable langage de modélisation de la méthode doit être développé en séparant les problématiques liées aux concepts de celles liées à la représentation de ceux-ci. Pour cette raison, le modèle de produit est décomposé en syntaxe abstraite, syntaxe concrète et sémantique. Chacun de ces éléments peut être évalué au regard de critères proposés dans la littérature portant sur l'ingénierie des méthodes.

3.1.3. Les modèles de processus

Les modèles de processus sont la deuxième composante d'une méthode. Ils permettent de formaliser les aspects procéduraux de la méthode, autrement dit la démarche qui s'applique. Dans cette partie nous définissons dans un premier temps ce que l'on appelle un modèle de processus, puis nous analysons quelles sont les différentes orientations que peuvent avoir les modèles de processus.

3.1.3.1. Définition des modèles de processus

Le processus décrit « *un ensemble d'activités interreliées et menées dans le but de définir un produit* » (Ralyte 2006). Cette première définition succincte nous permet de faire un lien immédiat entre la notion de produit évoquée dans la partie précédente, et celle de processus. Le produit est considéré comme une cible à atteindre de la méthode, et le processus décrit alors les moyens permettant d'atteindre cette cible.

Le processus est exprimé au travers d'un modèle de processus, qui comme le modèle de produit est lui-même l'instance d'un métamodèle appelé : métamodèle de processus. Le modèle de processus « *décrit à un niveau abstrait et idéalisé la façon d'organiser la production du produit : les étapes, les activités qu'elles comprennent, leur ordonnancement, et parfois les critères pour passer d'une étape à une autre* » (Ralyte 2006). La qualité du produit obtenu est extrêmement dépendante du processus mis en œuvre. En effet, chaque activité proposée dans le cadre du processus doit permettre d'obtenir une partie du produit, conformément au modèle de produit proposé (cf. Figure 24).

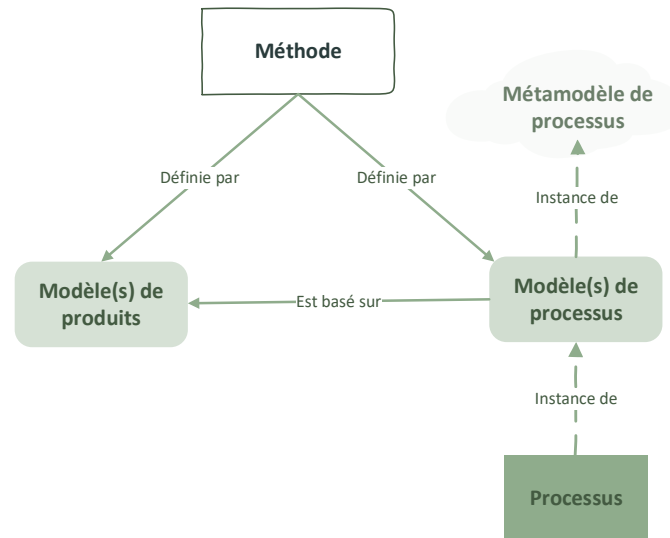


Figure 24 : Niveaux d'abstraction relatifs au processus

Le modèle de processus, tout à fait structurant dans la construction d'une méthode, se définit par le choix de métamodèle qui est fait. Ce métamodèle correspond à une orientation particulière, sélectionnée et voulue par le concepteur de la méthode. C'est ce point que nous adressons dans la partie suivante.

3.1.3.2. Orientations des modèles de processus

La littérature relative à l'ingénierie des méthodes nous permet de distinguer cinq types de modèles de processus (Hug 2009; Colette Rolland 2005; C. Rolland, Prakash, et Benjamin 1999) :

- Les modèles orientés activité, qui représentent les activités et leur ordonnancement pour la réalisation d'un produit. On peut citer parmi ces modèles la cascade ou le cycle en V.
- Les modèles orientés produit, qui couplent l'état du produit à l'activité que génère ce produit. C'est le cas des diagrammes d'état transition par exemple.
- Les modèles orientés décision, qui présentent les transformations ou les élicitations successives du produit dues à des décisions.
- Les modèles orientés contexte, qui prennent en compte l'intention et la situation d'un acteur à un instant donné du projet.
- Les modèles orientés stratégie, qui permettent de représenter les processus multi-démarches et prévoient plusieurs chemins possibles pour élaborer le produit en se basant sur les notions d'intentions et de stratégies. Ce type de modèle est une extension des modèles précédents.

Dans le cadre de ce travail, nous nous intéressons particulièrement aux modèles orientés stratégie. Le métamodèle de processus permettant de créer ces modèles de processus est le formalisme MAP, proposé par Colette Rolland (C. Rolland, Prakash, et Benjamin 1999). L'avantage de ce type de métamodèle de processus est qu'il permet de laisser aux utilisateurs de la méthode différentes possibilités d'atteindre un but donné. C'est aussi un moyen d'envisager la méthode comme étant non linéaire et non chronologique, la rendant ainsi fortement adaptative. Les objectifs à atteindre (les intentions) sont clairement fixés, mais les chemins à emprunter pour y parvenir (les stratégies) sont à sélectionner librement. Les stratégies peuvent être mises en œuvre dans leur totalité ou partiellement, tant que l'objectif est atteint. Ce métamodèle se décompose en sections (cf. Figure 25), constituées :

- D'une intention, c'est-à-dire un but à atteindre
- D'une ou plusieurs stratégies, qui sont autant de façons d'atteindre l'intention.

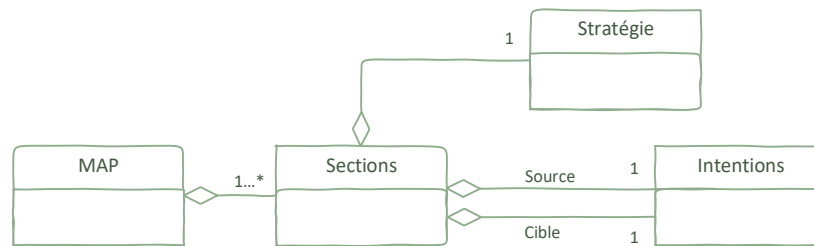


Figure 25 : Extrait du métamodèle MAP

Les intentions sont représentées sous la forme d'ovales, dans lesquels le nom de l'intention à atteindre est écrit. Les stratégies, représentées par des flèches orientées, peuvent permettre de passer d'une intention à une autre, ou encore d'une intention à elle-même. Cette possibilité caractérise le fait qu'une stratégie peut permettre de venir raffiner une intention, autrement dit peut venir enrichir la partie de modèle de produit qui est construite au travers de l'intention. La Figure 26 présente la syntaxe concrète d'une MAP. Par souci de clarté, toutes les sections ne sont pas représentées sur le schéma : seules deux d'entre elles sont représentées à titre d'exemple.

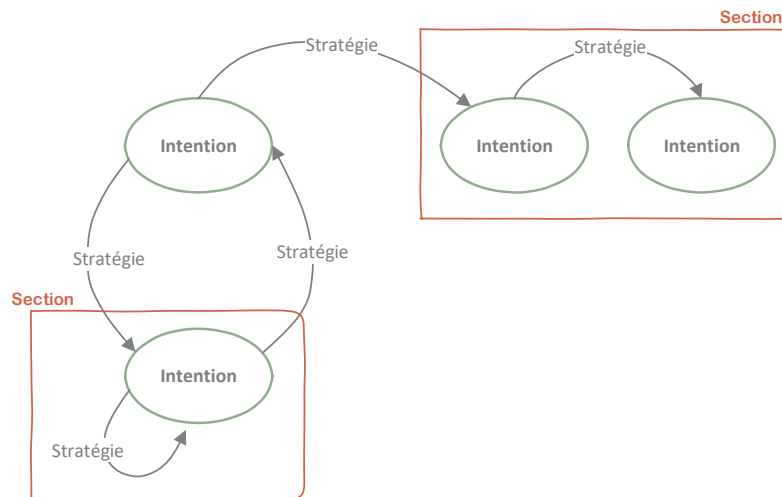


Figure 26 : Syntaxe concrète d'une MAP

3.1.3.3. Synthèse sur les modèles de processus

Les modèles de processus permettent de définir la marche à suivre pour atteindre le produit souhaité de la méthode. Ces modèles décrivent donc les activités, mais aussi les outils et techniques nécessaires, et sont directement reliés aux modèles de produit.

Les modèles de processus sont des instances de métamodèles, sélectionnés en fonction de l'orientation qu'ils proposent. Nous avons choisi dans cette partie de faire un focus particulier sur les modèles de processus orientés stratégie, et soutenus par le métamodèle MAP. La conception du modèle de processus autour des concepts d'intentions et de stratégies permet de proposer une démarche flexible, correspondant au contexte de démocratie dans lequel nous nous inscrivons.

3.1.4. Plateformes support aux méthodes

Une méthode d'ingénierie des systèmes d'information peut être supportée par une plateforme dédiée. L'objectif de cette plateforme qui vient outiller la méthode, est de proposer à minima une traçabilité

de l'évolution du modèle de produit, et de façon plus poussée, de permettre d'automatiser une partie du processus de développement (Brinkkemper et Harmsen 1998).

Dupuy-Chessa propose de construire une plateforme support à une méthode en s'appuyant sur les éléments suivants (Dupuy-Chessa et al. 2014) :

- La syntaxe abstraite, c'est-à-dire les concepts du modèle de produit.
- La syntaxe concrète, c'est-à-dire les éléments de représentation textuelle et/ou graphique de la syntaxe abstraite
- Un modèle de mapping qui permet de configurer les liens entre les deux syntaxes.

L'articulation entre les différents éléments à prendre en compte est schématisée dans la Figure 27.

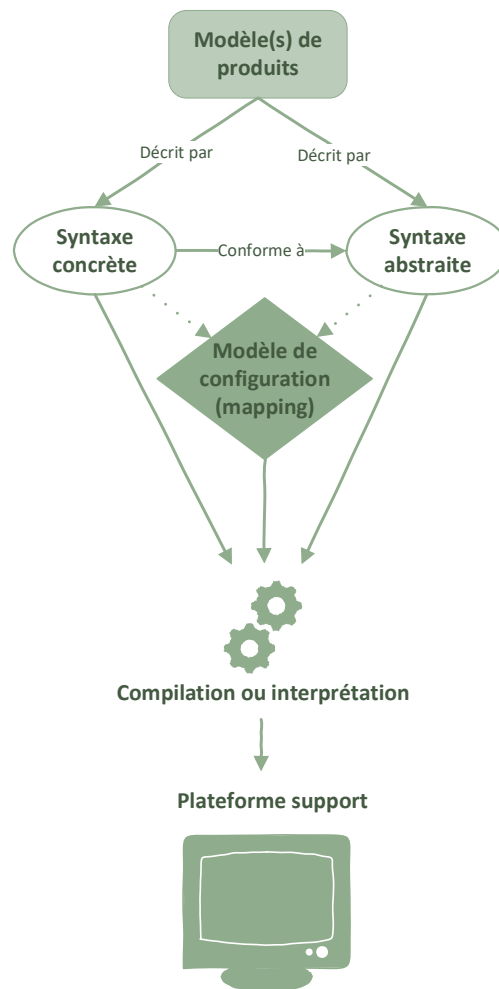


Figure 27 : Définition d'une plateforme support à une méthode (Dupuy-Chessa et al. 2014)

La méthodologie traditionnellement utilisée dans la communauté de l'ingénierie des méthodes pour produire une plateforme support est la suivante (Ráth, Ökrös, et Varró 2010) :

- Dans un premier temps construire la syntaxe abstraite, en conformité avec le métamodèle de produit choisi. Il s'agit d'abord d'identifier les concepts, puis d'établir les liens entre ceux-ci.
- Dans un second temps construire la syntaxe concrète du modèle de produit, en associant chacun des éléments de notation visuelle aux éléments composant la syntaxe abstraite.

Cette première partie d'état de l'art nous a permis de définir ce qu'était une méthode, et d'en décrire les éléments constitutifs. Les notions de modèle de produit et de modèle de processus ont été présentées, et nous permettront par la suite de nous positionner quant à la méthode que nous proposons : DEMOS.

Les méthodes, qui ont pour nous vocation à soutenir la construction de systèmes d'information sont envisagées non comme un cadre normatif, mais comme un véritable outil. Pour cette raison, nous avons choisi d'aborder le modèle de produit comme un modèle appréhendable par tous les acteurs du processus de construction, utilisateurs finaux compris. Cette simplicité ne doit pendant pas venir entacher la richesse des modèles, qui viendront représenter les aspects statique, dynamique, fonctionnel et éventuellement d'autres aspects du système d'information à construire. En poursuivant toujours cet objectif de démocratie, nous avons choisi de nous arrêter sur le modèle de processus orienté stratégie, qui laisse les participants libres de définir le meilleur chemin à adopter pour atteindre l'objectif d'un modèle de produit complet. Les méthodes peuvent en outre être soutenues par une plateforme dédiée, qui permet de répertorier tous les éléments du modèle de produit obtenu, et d'assurer la plus grande traçabilité possible.

Les éléments de méthode étant maintenant formalisés, nous nous intéressons dans la section suivante à l'ingénierie des méthodes en tant que telle, c'est-à-dire au processus de conception d'une méthode.

3.2. Ingénierie des méthodes

L'ingénierie des méthodes constitue une discipline à part entière dans le domaine des systèmes d'information. Notre contribution dans ce travail de thèse étant une méthode de construction de systèmes d'information soutenant la démocratie des organisations, nous nous positionnons clairement dans le champ de cette discipline.

Dans cette partie, nous définissons ce qu'est l'ingénierie des méthodes, et en donnons les contours en dressant les grands principes qui la structurent. Nous expliquons ensuite le processus de construction d'une méthode (cf. Figure 28). Comme dans la partie précédente, tous ces éléments nous permettent par la suite de positionner le processus d'ingénierie que nous avons choisi pour la création de la méthode DEMOS.

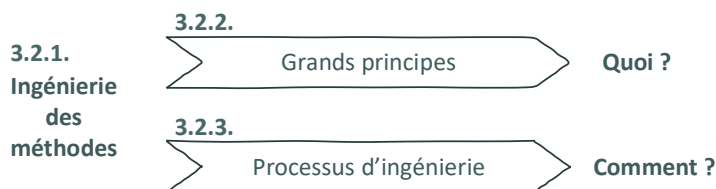


Figure 28 : Plan de la partie 3.2

3.2.1. Définition de l'ingénierie des méthodes et bonnes pratiques

Brinkkemper définit l'ingénierie des méthodes comme la discipline de conceptualisation, de construction et d'adaptation de méthodes, techniques et outils pour le développement des systèmes

d'information (Brinkkemper 1996). Ralyte partage cette vision orientée « cycle de vie » en utilisant elle aussi les termes de construction et d'adaptation, mais elle précise la notion de système d'information en ajoutant que ceux-ci répondent aux « *exigences particulières d'une situation d'entreprise* » (Ralyte 2006). En d'autres termes, l'ingénierie des méthodes existe justement car il n'y a pas une méthode universelle de construction de systèmes d'information, mais bien une pluralité de méthodes répondant aux besoins d'un contexte particulier dans le projet d'une organisation.

Le risque naturel à adapter des méthodes à chaque contexte particulier est bien sûr celui de la multiplication de méthodes. Le fait de proposer une nouvelle méthode doit par conséquent être justifié par un besoin nouveau et clairement différenciant. Dans tous les cas, une méthode émerge rarement ex nihilo, et est souvent basée sur la réutilisabilité de tout ou partie d'une méthode existante, nous détaillons cela dans la section suivante.

L'ingénierie des méthodes présente un objectif clair : proposer des démarches qui garantissent de parvenir à des méthodes d'ingénierie des systèmes d'information formelles et clairement définies. C'est une réponse aux problèmes fréquemment rencontrés (Ralyte 2006), et qui peuvent être listés ainsi :

- Méthodes trop générales et donc mal adaptées aux problèmes particuliers rencontrés.
- Méthodes difficiles à faire évoluer pour prendre en compte l'expérience résultant de leur utilisation.
- Méthodes trop imprécises ne proposant pas un guidage fin des activités à mettre en œuvre.
- Méthodes ne prenant pas en compte les connaissances heuristiques acquises par leurs utilisateurs lorsqu'ils les appliquent.

Ralyte identifie une cause principale à ces difficultés : la faiblesse des modèles de processus proposés, qui ne sont que de simples prescriptions d'activités organisées en séquences (Ralyte 2006). Les méthodes sont alors rigides, imprécises et peu claires quant aux objectifs qu'elles poursuivent.

L'ingénierie des méthodes est donc une discipline indispensable et riche, qui traite de tous les aspects relatifs aux méthodes. Elle est aussi un réel support pour la création de nouvelles méthodes, au travers notamment des principes qui ont été fixés, et que nous détaillons dans la section suivante.

3.2.2. Les principes de l'ingénierie des méthodes

Ralyte a proposé en 2006 une série de principes relatifs à l'ingénierie des méthodes. Nous avons choisi de sélectionner et de présenter les principes qui nous semblent structurants : la métamodélisation, la réutilisation et la modularité.

3.2.2.1. La métamodélisation

La métamodélisation peut être définie comme « *la modélisation des modèles qui composent une méthode* » (Ralyte 2006). Comme nous l'avons vu dans la partie précédente, il existe deux modèles permettant de constituer une méthode : le modèle de produit et le modèle de processus. Le principe de métamodélisation consiste à proposer des modèles de produit et de processus interconnectés de manière à ce que la méthode soit cohérente.

Ainsi, « *toute activité du processus doit agir sur un élément du produit et inversement il n'y a pas d'élément du produit qui ne soit manipulé par le processus* » (Ralyte 2006). En d'autres termes, les éléments du modèle de processus produisent des éléments du modèle de produit et en miroir, les éléments du modèle de produit consomment les éléments du modèle de processus. Cette vision du principe de métamodélisation est synthétisée au travers de la Figure 29.

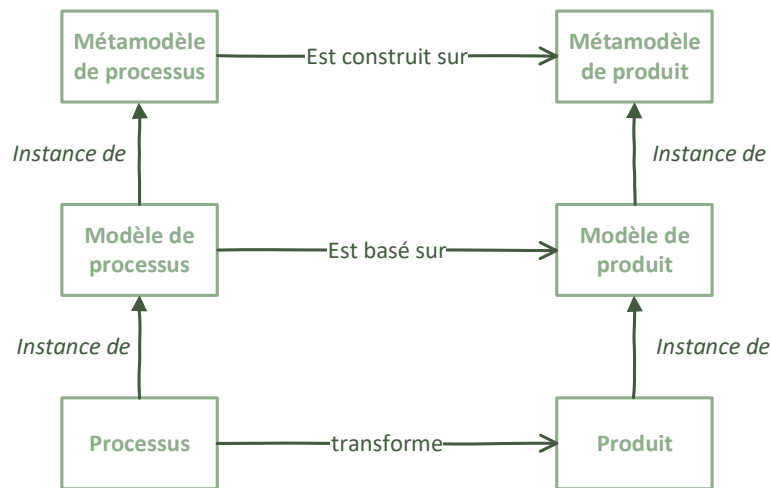


Figure 29 : Les niveaux d'abstraction (Ralyte 2006)

3.2.2.2. La réutilisation et la modularité

Comme nous l'avons évoqué précédemment, la multiplication des méthodes est une véritable problématique dans le domaine. L'ingénierie des méthodes propose donc de mettre en œuvre les principes de modularité et de réutilisabilité afin notamment de palier à cela.

Le principe de réutilisation peut être défini comme l'assemblage de différents composants de méthodes qui ont « déjà fait leur preuve ». Cela suppose une étude approfondie des méthodes existantes qui répondent à la problématique que l'on s'est fixée. Pour que ce principe puisse exister, il faut que les méthodes auxquelles on « emprunte » des éléments aient elles-mêmes été conçues de façon modulaire.

Le principe de modularité peut être défini comme le découpage « en blocs réutilisables » de la méthode (Ralyte 2006). Cela permet, dans un premier temps de choisir d'appliquer tout ou partie d'une méthode afin de l'adapter à chaque projet. Dans un second temps, c'est aussi un moyen de rendre les « blocs de méthode » réutilisables par d'autres, afin qu'ils mettent en œuvre le principe de réutilisation. Cela suppose une description très complète de chaque élément de méthode, afin de la rendre appréhendable par d'autres concepteurs.

Ces principes de modularité et de réutilisation peuvent donner lieu à un positionnement de chaque méthode sur le spectre proposé par Harmsen (Harmsen, Brinkkemper, et L. Han Oei 1994) et présenté en Figure 30.

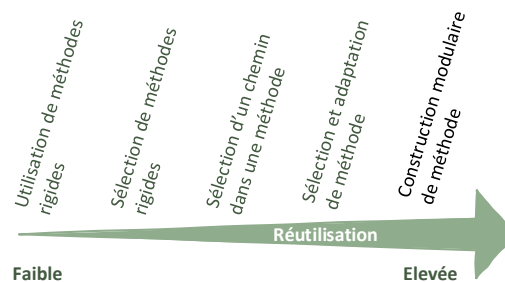


Figure 30 : Spectre des approches d'ingénierie des méthodes (Harmsen, Brinkkemper, et L. Han Oei 1994)

Les principes d'ingénierie des méthodes que nous avons présentés sont absolument nécessaires pour que la méthode développée réponde aux exigences que se fixe la discipline : précision et formalisation

avant tout. Ces principes sont ensuite déclinés tout au long du processus d'ingénierie des méthodes que nous détaillons maintenant.

3.2.3. Le processus d'ingénierie des méthodes

Le processus d'ingénierie des méthodes a été formalisé par Ralyte dans un article de 2001 (Ralyte 2001). Cette formalisation nous semble encore aujourd'hui très claire et très complète. Nous allons donc présenter dans une première section le cadre de référence qu'elle propose. Nous irons ensuite dans le détail de chacune des sections proposées pour positionner notre expérience de l'ingénierie des méthodes pour la conception de DEMOS.

3.2.3.1. Cadre de référence

Le processus d'ingénierie des méthodes est basé sur trois principes majeurs : la métamodélisation, la réutilisation et la modularité. Le cadre de référence proposé par Ralyte pour le processus d'ingénierie des méthodes reprend ces principes en les déclinant dans des stratégies. Le cadre de référence qu'elle propose est formalisé dans une MAP, avec deux intentions principales : définir l'objectif de la construction d'une méthode et construire une méthode. Chaque intention peut être atteinte en passant par différentes stratégies présentées dans la Figure 31.

Chacune des stratégies proposées fait l'objet d'une description et d'un positionnement dans la partie suivante. Nous nous intéressons dans un premier temps aux stratégies permettant d'atteindre la première intention : définir l'objectif de la construction d'une méthode. Dans un second temps nous détaillons les stratégies permettant l'atteinte de la seconde intention : construire une méthode.

Chacune des sections respecte le formalisme suivant : *< Intention de départ, Intention d'arrivée, Stratégie >*.

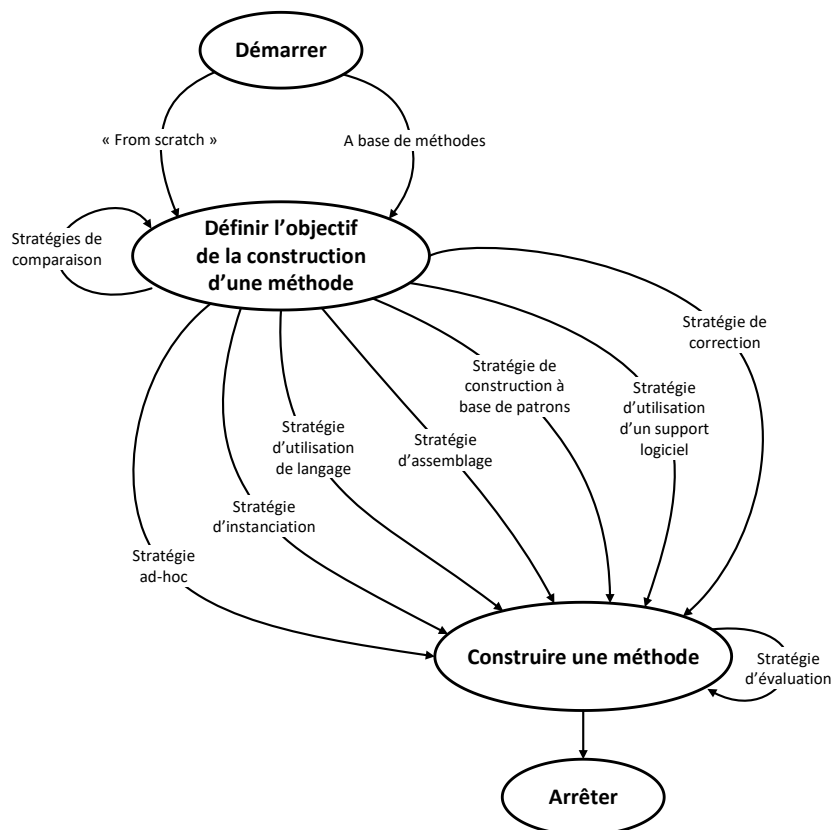


Figure 31 : Cadre de référence du processus d'ingénierie des méthodes (Ralyte 2001)

3.2.3.2. Première intention : définir l'objectif de la construction d'une méthode

La première intention peut être atteinte grâce à trois stratégies : **From scratch**, **A base de méthodes** et/ou **Stratégie de comparaison**.

- <Démarrer, Définir l'objectif de la construction d'une méthode, **From scratch**>
Cette stratégie consiste à créer de toutes pièces une nouvelle méthode, si le concepteur ne trouve pas de méthode adaptée à son problème. La méthode qu'il propose peut ensuite être généralisée si « elle a fait ses preuves lors de son application dans le projet pour lequel elle a été créée » (Ralyte 2001)
- <Démarrer, Définir l'objectif de la construction d'une méthode, **A base de méthodes**>
Cette stratégie est une application directe du principe de réutilisation. Il s'agit de construire sa méthode sur la base de morceaux de méthodes existantes. L'ingénierie des méthodes est d'ailleurs parfois décrite comme une approche de construction de méthode combinant différentes parties de méthodes (Punter et Lemmen 1996).
- <Définir l'objectif de la construction d'une méthode, Définir l'objectif de la construction d'une méthode, **Stratégie de comparaison**>
Cette stratégie est ce qu'on appelle une stratégie de raffinement. Elle permet d'aller d'une intention vers elle-même en affinant le produit obtenu. Ici, il s'agit de comparer les différentes possibilités qui s'offrent à nous (autres méthodes notamment), afin de sélectionner celles qui correspondent le plus à nos besoins.

3.2.3.3. Deuxième intention : Construire une méthode

La seconde intention peut être atteinte grâce à huit stratégies : **Stratégie ad hoc**, **Stratégie d'instanciation**, **Stratégie d'utilisation de langage**, **Stratégie d'assemblage**, **Stratégie de construction à base de patrons**, **Stratégie d'utilisation d'un support logiciel**, **Stratégie de correction**, **Stratégie d'évaluation**. Nous avons choisi de ne décrire ici que quatre des huit stratégies, qui correspondent à celles que nous avons effectivement mises en œuvre dans notre processus d'ingénierie de méthode.

- <Définir l'objectif de la construction d'une méthode, Construire une méthode, **Stratégie d'instanciation**>
Il s'agit de concevoir la méthode au travers de l'instanciation de modèles. Dans le cadre de méthodes de construction de systèmes d'information on parle de l'instanciation d'un modèle de produit et de l'instanciation d'un modèle de processus. Ces modèles doivent alors eux-mêmes être l'instance de métamodèles de processus et de produit.
- <Définir l'objectif de la construction d'une méthode, Construire une méthode, **Stratégie d'utilisation de langage**>
Cette stratégie propose d'appliquer un langage de modélisation à la méthode. Ce langage s'exprime de manière formelle au travers d'une syntaxe abstraite, d'une syntaxe concrète et d'une sémantique.
- <Définir l'objectif de la construction d'une méthode, Construire une méthode, **Stratégie d'assemblage**>
Cette stratégie consiste à sélectionner et réutiliser des bouts de méthodes existantes en les assemblant. Il s'agit d'une application du principe de réutilisation de l'ingénierie des méthodes. La structure des composants de méthodes peut se faire selon trois types de décomposition (Brinkkemper et Harmsen 1998) : dimension de perspective (correspondant au découpage produit/processus), dimension d'abstraction (correspondant au découpage

conceptuel/technique) et dimension de granularité (correspondant à une décomposition par niveau de détail : processus, activités, tâches par exemple)

- *<Construire une méthode, Construire une méthode, **Stratégie d'évaluation**>*

La stratégie d'évaluation est ici encore une stratégie de raffinement car elle permet d'apporter des corrections à la construction de la méthode. Les stratégies d'évaluation sont variées, mais consistent principalement en une application de la méthode à différents cas, et en une évaluation par les participants. « *Les expériences, remarques et propositions* » (Ralyte 2001) des participants sont en effet cruciales dans l'évolution de la méthode. La méthode peut aussi être évaluée en expérimentant les capacités de l'outil développé.

Cette seconde partie d'état de l'art sur l'ingénierie des méthodes nous a permis de mettre en lumière les « attendus » lorsque l'on souhaite appliquer un processus de création de méthodes. Au travers des définitions et des principes sous-jacents à l'ingénierie des méthodes, nous avons pu positionner les pré-requis que nous nous sommes fixés pour la création de DEMOS.

Le cadre de référence « adaptable » proposé par Jolita Ralyte et décrivant le processus d'ingénierie des méthodes nous permet également de positionner notre propre processus en termes d'intentions et de stratégies.

Nous avons souhaité, avec la méthode DEMOS, répondre aux critères d'exigence qu'impose la discipline de l'ingénierie des méthodes, mais également formaliser non seulement notre méthode (cf. partie 4.1) mais aussi le processus de conception et d'adaptation de cette méthode. Ces éléments sont des piliers sur lesquels repose le chapitre 5 de notre thèse, qui présente la méthode DEMOS et son processus de conception dans son intégralité.

3.3. Conclusion du chapitre

L'objectif de ce chapitre était de présenter les travaux existants dans le domaine de l'ingénierie des méthodes afin de pouvoir par la suite positionner notre méthode : DEMOS. La définition du concept de méthode et la présentation du processus d'ingénierie des méthodes nous ont permis d'explicitier les concepts de modèle de produit et de modèle de processus comme éléments fondateurs d'une méthode.

Une méthode peut ainsi être décomposée de la manière suivante (cf. Figure 32) :

- Son modèle de produit, décrit par une syntaxe abstraite, une syntaxe concrète et une sémantique et représentant différents aspects du système d'information à construire (statiques, dynamiques, fonctionnels, etc). Ce modèle de produit est l'instance d'un métamodèle de produit.
- Son modèle de processus conforme à une orientation choisie par le concepteur et représenté conformément à un métamodèle de processus.
- Sa plateforme support, c'est-à-dire l'outil dédié au répertoriage et à la traçabilité de tous les éléments du produit.

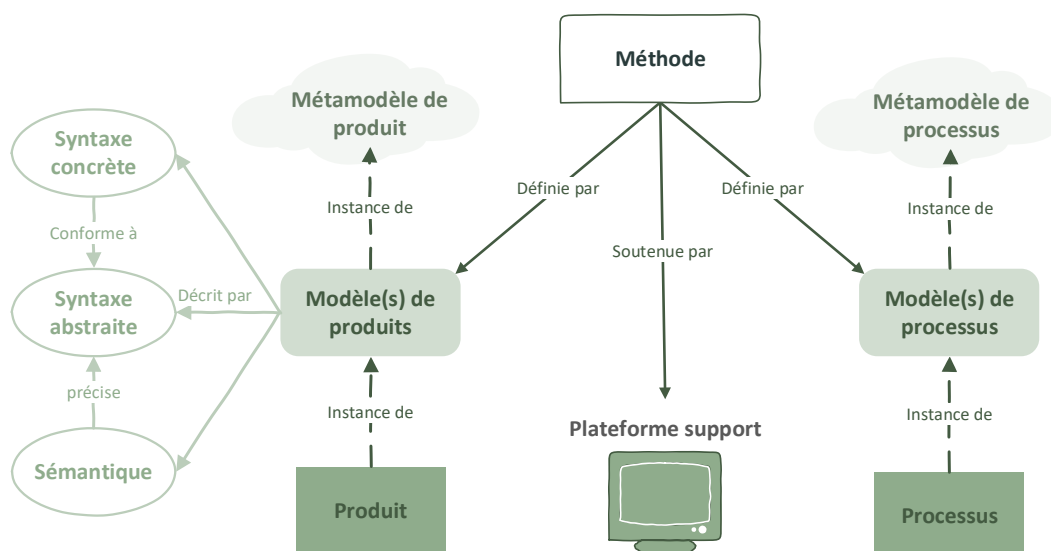


Figure 32 : Vision complète de la description d'une méthode

L'état de l'art sur l'ingénierie des méthodes nous a quant à lui permis de formaliser le processus de conception d'une méthode, sous la forme d'une MAP, qui permet de diviser en deux étapes cruciales la conception d'une méthode

- La définition de l'objectif de la construction d'une méthode, via des stratégies telles que l'étude de méthodes existantes ou la création d'une nouvelle méthode.
- La construction de la méthode, via des stratégies telles que la métamodélisation et l'instanciation, l'utilisation d'un langage de modélisation, l'assemblage de composants et l'évaluation.

Ces éléments sont ceux sur lesquels nous nous appuyons pour décrire notre contribution dans le chapitre suivant.

Partie II

DEMOS : a DDesign Method for demOcratic information System

Chapitre 4

POSITIONNEMENT DE NOTRE CONTRIBUTION

Nous avons développé la méthode DEMOS afin de contribuer à soutenir la démocratie dans les organisations via les systèmes d'information. C'est cet objectif qui nous a guidés depuis nos réflexions préalables, jusqu'à la mise en œuvre de la méthode dans le cadre d'expérimentations. Ce *pourquoi ?* a été pour nous structurant, car il est la ligne d'horizon qui ne nous a pas quittés durant tout ce travail de thèse. C'est cela que nous expliquons dans une première partie, où nous positionnons notre méthode par rapport au contexte de la démocratie dans les organisations (cf. Chapitre 2). Notre seconde interrogation a tout naturellement été *quoi ?* Qu'est-ce qui peut permettre de construire des systèmes d'information soutenant la démocratie des organisations ? Le rôle actif de l'utilisateur dans la construction de son système d'information nous a semblé être une condition sine qua none pour un système d'information démocratique. C'est ce que nous montrons dans une deuxième partie, en liant nos propositions au chapitre d'état de l'art que nous avons consacré à la place de l'utilisateur dans la construction de systèmes d'information (cf. Chapitre 3). Enfin, et afin d'appliquer les grands principes que nous nous sommes fixés, nous avons tâché de répondre à la question *Comment ?* Nous avons répondu à cette question en proposant une nouvelle méthode de conception de système d'information nommée DEMOS pour D^Esign Method for demOcratic information System. Nous avons structuré cette méthode en nous appuyant sur les grands principes de l'ingénierie des méthodes (cf. Chapitre 4), c'est ce que nous montrons dans la troisième partie de ce chapitre (cf. Figure 33).

DEMOS : a D^Esign Method for demOcratic information System

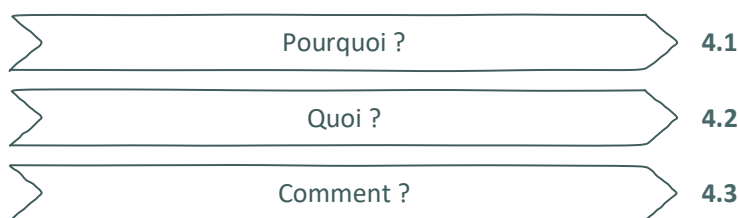


Figure 33 : Plan du chapitre 4

4.1. Les objectifs de DEMOS

La démocratie dans la société nous semble être un sujet majeur de préoccupation aujourd'hui. C'est un enjeu de société, et il nous semble opportun de revenir aux origines de la démocratie, et aux définitions qui en sont proposées par les philosophes pour nous éclairer sur ce concept, couramment utilisé, et parfois galvaudé. La section 1.1 de cette thèse nous a permis de proposer une vision large de

la démocratie dans la société, que nous avons finalement choisi de caractériser par trois éléments : la séparation des trois pouvoirs, le droit à la remise en cause des lois par les citoyens et la participation de tous à la prise de décision.

Ce sujet de la démocratie, bien que très vaste et pouvant sembler un peu vaporeux, impalpable, prend aussi racine dans les organisations. La démocratie des organisations peut être vue comme un objectif à atteindre, et pour lequel un certain nombre de stratégies peuvent être mises en place : respect du droit du travail, participation collective à la prise de décision et limitation de la gouvernementalité (c'est-à-dire des moyens permettant de cadrer le travail, de le contraindre, sans éliciter les règles sous-jacentes).

Notre thèse s'inscrivant dans le domaine des systèmes d'information, il nous a semblé que nos contributions pouvaient poursuivre l'objectif de soutenir la démocratie des organisations. Pour cela, nous avons décrit notre vision du système d'information : un prisme permettant à l'organisation de se voir elle-même et de voir son environnement. Nous avons ensuite cherché à comprendre quels pouvaient être les leviers permettant aux systèmes d'information de soutenir la démocratie des organisations (cf. section 1.2.). Nous avons identifié des éléments fondateurs, qui sont les objectifs poursuivis par notre contribution afin de garantir qu'elle soutienne la démocratie des organisations. Les premiers éléments ont trait au processus de construction de système d'information. Afin qu'il soit en lui-même le plus démocratique, mais aussi qu'il permette de soutenir la démocratie de l'organisation, nous avons décidé que ce processus de construction devait impliquer les futurs utilisateurs du système d'information.

Cette implication active doit être garantie par la démarche proposée. L'utilisateur ne doit pas avoir uniquement un rôle consultatif ou informationnel mais bien un rôle de co-constructeur de son système d'information. Il doit pouvoir participer au débat autour des normes et des valeurs qui sont encapsulées dans le système, afin à minima de les éliciter, et au mieux de participer à leur établissement. Il s'agit dans un premier temps de limiter la gouvernementalité dans l'organisation, mais aussi de permettre aux utilisateurs de maîtriser les tenants et aboutissants d'un projet qui les concerne.

Cette implication des utilisateurs a également pour but de faire émerger des points de vue divergents et complémentaires. Ce respect d'une pluralité de points de vue étant l'une des garanties de la démocratie, la méthode que nous proposons poursuit l'objectif de permettre de capter, représenter, implanter et faire évoluer ces points de vue. Ainsi, le système n'est pas le reflet d'un point de vue dominant, mais bien un système construit sur une pluralité de points de vue.

Pour synthétiser, et pour répondre à la question : *pourquoi DEMOS* ? Nous pouvons dresser une liste de quatre objectifs qui ont guidé notre contribution :

- Pour permettre aux utilisateurs de participer au processus de construction de leur système d'information, en ayant un rôle actif dans la prise de décision notamment.
- Pour permettre aux utilisateurs de rediscuter et d'éliciter les « lois », c'est-à-dire les valeurs et principes qui sous-tendent la construction du système d'information.
- Pour considérer les utilisateurs comme de véritables co-constructeurs en ne les écartant pas du processus, notamment lors des phases qui sont habituellement réservées aux « experts » : conception et développement.
- Pour prendre en compte et respecter une pluralité de points de vue, et laisser la possibilité d'avoir un système d'information « multi-facettes », qui n'est pas uniquement construit sur le point de vue dominant.

Tous ces éléments sont synthétisés dans la Figure 34 qui suit.

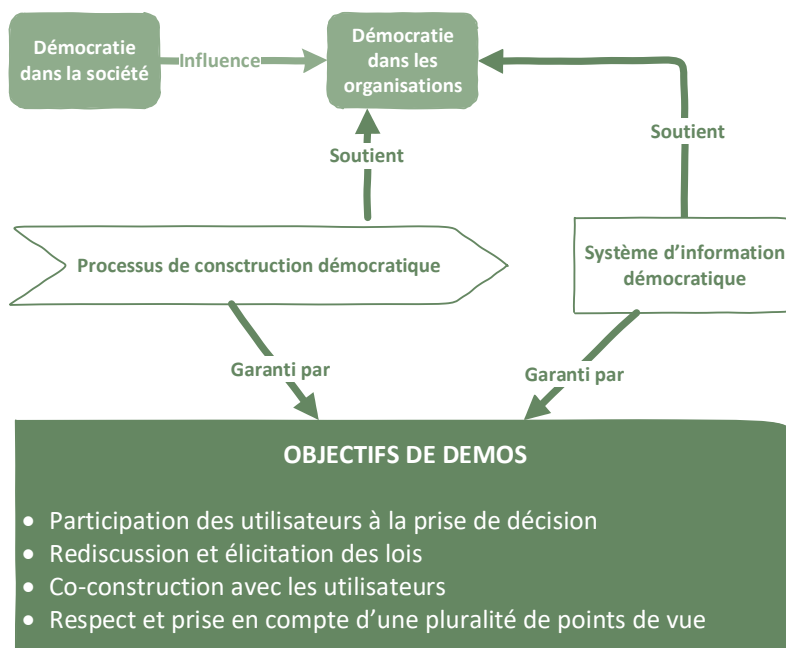


Figure 34 : Les objectifs de DEMOS

4.2. Les principes de DEMOS

Que voulons-nous proposer afin de satisfaire les objectifs que nous nous sommes fixés ? Cette question nous a poussés à affiner notre proposition en la faisant reposer sur de grands principes permettant de garantir un processus de construction démocratique et un système d'information démocratique. La problématique de l'empowerment des utilisateurs de système d'information a été au cœur de nos réflexions, et notre état de l'art sur l'implication des utilisateurs dans la construction de leur système d'information s'est articulé autour de cela.

Nous avons repéré, parmi les travaux existants, qu'un certain nombre de démarches de construction de systèmes d'information proposaient une implication des utilisateurs limitée (cf. section 3.1.). Au niveau de l'ingénierie des exigences, une limite majeure nous est apparue : l'outillage méthodologique (permettant de garantir que les éléments issus du processus de recueil des exigences soient effectivement pris en compte dans la suite du processus de construction), est largement insuffisant. En effet, cette étape étant cruciale au sens où elle est celle durant laquelle les valeurs et principes sont débattus, élicités, il nous a semblé qu'il était primordial que les éléments qui en résultent soient conservés et tracés durant tout le processus. Pour ce qui est de la phase de conception, l'implication des utilisateurs se fait à des degrés divers. Il peut avoir un rôle informatif, consultatif ou réellement participatif. Nous avons pu voir que les démarches de conception participatives étaient une bonne « base » pour notre contribution, à condition qu'elles respectent des critères tels que : une participation directe, individuelle (ou par le biais de représentants) et conduisant notamment à un empowerment démocratique des utilisateurs. Pour le processus de développement, il nous a semblé intéressant d'explorer le champ des méthodes agiles, qui s'affichent comme participatives. Nous avons pu identifier deux risques que nous souhaitons dépasser au travers de notre contribution : la participation des utilisateurs finaux (et pas seulement des clients) au processus de développement et l'établissement d'une intelligibilité mutuelle entre développeurs et utilisateurs.

Toujours dans la poursuite de nos objectifs, nous avons également exploré la littérature portant sur la question des points de vue dans les systèmes d'information (cf. section 2.2.). Notre objectif est

clairement de proposer de développer des systèmes respectant une pluralité de points de vue, et les implantant. Dans le champ de l'ingénierie des exigences, nous avons pu observer que les utilisateurs étaient considérés comme un groupe homogène avec un point de vue unique. Il nous a semblé que cette vision devait être actualisée pour arriver à prendre en compte des points de vue divergents entre utilisateurs dès le début de la construction du système d'information. Au niveau de la conception, nous nous sommes intéressés à l'exercice de modélisation des points de vue, et avons pu observer que la question du point de vue était toujours considérée sous un angle technique et non dans une visée « démocratique », c'est-à-dire de respect des points de vue des utilisateurs. Il nous a semblé ici encore que cette vision pouvait être dépassée grâce à notre contribution, en proposant de modéliser les points de vue utilisateurs de façon claire et appréhendable par ces utilisateurs eux-mêmes.

A partir de cet état de l'art, nous pouvons à présent définir les principes de notre contribution (cf. Figure 35) (Bour, Soule-Dupuy et Vallès-Parlangeau 2019a, 2019b). Nous avons souhaité proposer un outillage méthodologique, autrement dit une méthode de construction de systèmes d'information. Cette méthode, nommée DEMOS doit reposer sur les principes suivants :

- Impliquer les utilisateurs finaux dans un processus participatif et démocratique.
- Favoriser un débat démocratique permettant l'émergence de points de vue.
- Concevoir un système d'information démocratique qui prend en compte ces points de vue.
- Assurer une traçabilité des points de vue dans le système d'information.
- Permettre l'évolution des points de vue dans le cycle de vie du système.

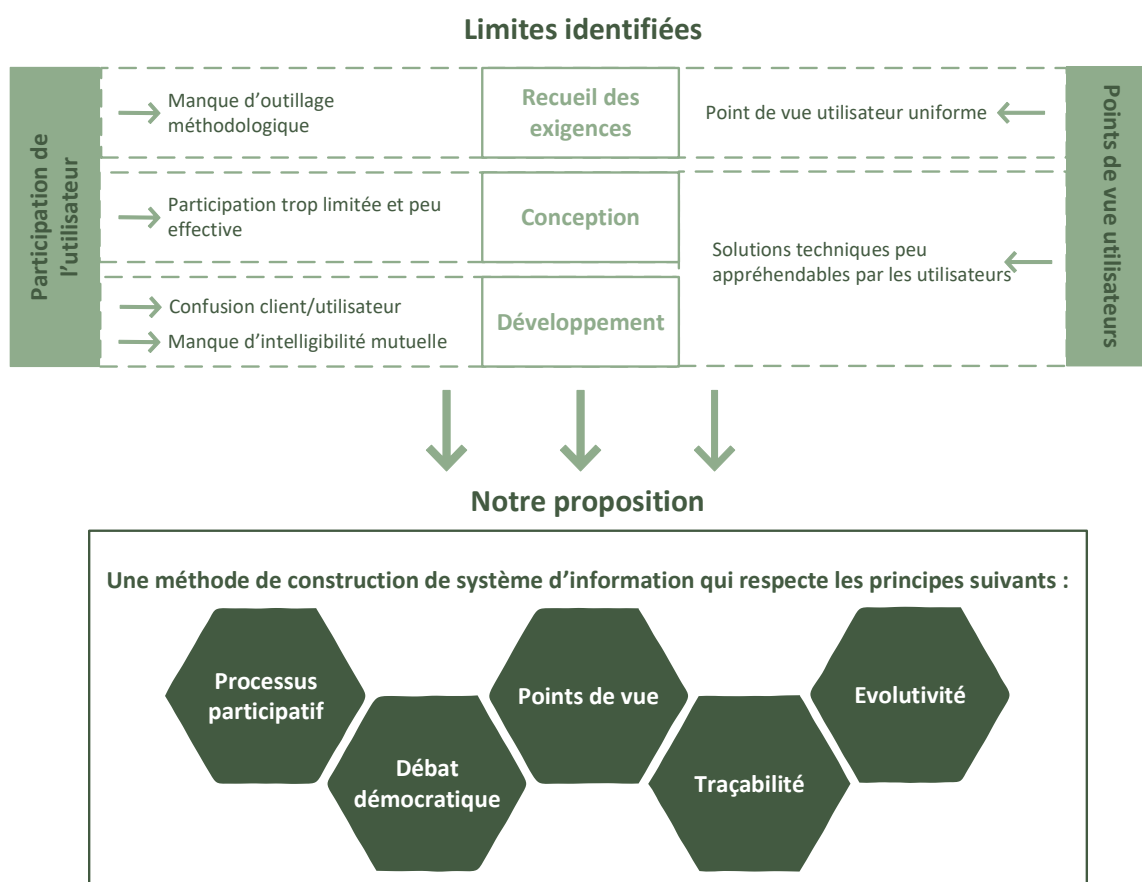


Figure 35 : Les principes de DEMOS

4.3. Composition de la méthode DEMOS

Les grands principes de DEMOS étant établis, nous nous sommes appuyés sur un état de l'art du domaine de l'ingénierie des méthodes afin de construire notre méthode. En réponse aux buts généraux assignés au domaine de l'ingénierie des systèmes d'information (cf. section 3.1.1), et en regard des objectifs propres que nous nous sommes fixés, il nous a semblé opportun de définir un certain nombre de règles :

- La méthode doit permettre à toutes les parties prenantes du projet d'appréhender et de limiter le domaine métier dans lequel s'inscrit le projet.
- La méthode doit permettre de spécifier les exigences du projet, notamment du point de vue stratégique, ainsi que les besoins fonctionnels attendus. Elle doit permettre le débat autour des valeurs et des normes qui conditionneront le système.
- La méthode doit servir à concevoir et réaliser une solution informatique correspondant aux besoins et exigences exprimés. Les modèles utiles à cette construction doivent être au plus proche des représentations cognitives des utilisateurs afin de rester compréhensibles par ceux-ci. Pour cela, les modèles doivent être adaptés aux points de vue des utilisateurs.
- La méthode est un vecteur de communication entre toutes les parties prenantes, présentes tout au long du projet : utilisateurs finaux, clients, développeurs, animateur de la méthode.

Nous nous sommes ensuite appuyés sur la décomposition : modèle de produit, modèle de processus afin de proposer une décomposition formelle de DEMOS (cf. sections 3.1.2 et 3.1.3). Les sections suivantes sont consacrées à une description des caractéristiques principales des modèles de DEMOS.

4.3.1. Les caractéristiques du modèle de produit de DEMOS

Nous avons fait le choix de la simplicité avec un modèle de produit unique. Ce modèle propose de représenter les aspects statiques, dynamiques et fonctionnels du système d'information. Dit autrement, notre modèle de produit propose une représentation des éléments suivants :

- Le vocabulaire, c'est à dire les données pour la partie statique.
- Les processus et les activités pour la partie dynamique.
- Les fonctionnalités attendues pour la partie fonctionnelle.

La particularité du modèle de produit de DEMOS est que chacune de ses facettes est directement liée au concept de point de vue qui est essentiel compte tenu de l'un des objectifs que nous nous sommes fixés : respecter une pluralité de points de vue dans le système d'information. Nous avons donc choisi d'ajouter une quatrième dimension aux trois citées précédemment : l'aspect perspectif, autrement dit l'aspect relatif aux points de vue.

Le modèle de produit de DEMOS est une instance de diagramme de classes. Nous avons choisi ce formalisme de métamodèle largement éprouvé pour sa simplicité et sa complétude vis-à-vis des éléments que nous souhaitions intégrer.

Pour approfondir la description du modèle de produit, nous avons choisi de le définir en respectant la décomposition suivante : syntaxe abstraite, syntaxe concrète et sémantique. La syntaxe abstraite est représentée sous la forme d'un diagramme de classes présentant tous les concepts manipulés dans la méthode. La syntaxe concrète est composée d'un ensemble d'éléments de notation visuelle, reliés chacun à l'un des éléments de la syntaxe abstraite. Elle sera évaluée au regard des critères de Moody

précédemment mentionnés (cf. 3.1.2.2). La sémantique du langage quant à elle est précisée sous la forme d'un dictionnaire des propriétés accompagnant la lecture de la syntaxe abstraite.

Pour synthétiser notre approche du modèle de produit de DEMOS, on peut dire que la méthode DEMOS est conforme aux propositions que nous avons énoncées à partir de la littérature dans le domaine de l'ingénierie des méthodes. Le modèle de produit représente les aspects statique, dynamique, fonctionnel et perspectif du système d'information à construire. Le modèle de produit est une instance de diagramme de classes pour sa syntaxe abstraite, et cette dernière est complétée par une syntaxe concrète et une sémantique (cf. Figure 36).

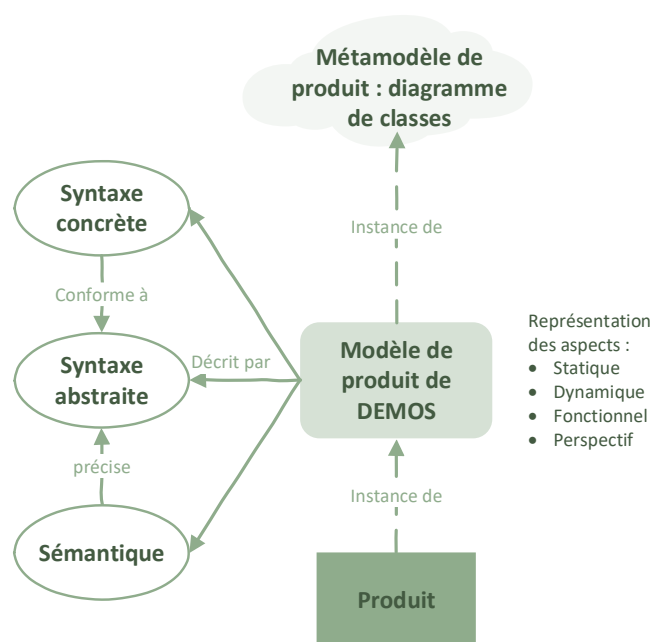


Figure 36 : Formalisation du modèle de produit de DEMOS

4.3.2. Les caractéristiques des modèles de processus de DEMOS

DEMOS est formalisée au travers de deux modèles de processus : l'un décrit le processus de construction du système d'information, l'autre décrit la boucle d'amélioration continue de ce système (Bour, Soule-Dupuy et Vallès-Parlangeau 2015). Ces modèles décrivent les différentes activités de la démarche. Chaque activité des modèles de processus est reliée directement à un élément du modèle de produit, afin de s'assurer d'une cohérence totale dans la mise en œuvre de la méthode. Les modèles de processus de DEMOS sont des modèles à « granularité variable ». Ils peuvent en effet se lire de façon tout à fait macroscopique, ou en zoomant sur des parties afin d'y découvrir les activités détaillées, et complétées par les outils mis en œuvre.

Nous avons choisi, dans une perspective démocratique, de continuer de considérer la méthode comme un soutien, et non comme un cadre normatif, et de nous appuyer par conséquent sur des modèles de processus orientés stratégie. Nos modèles de processus sont des instances du métamodèle MAP, qui formalise la construction de processus autour d'intentions et de stratégies pour les atteindre. Cela permet de ne ni normer ni rigidifier la démarche, mais au contraire de garantir une flexibilité permettant d'atteindre les buts fixés : débattre autour des normes et des valeurs, faire participer tous les utilisateurs en tant que co-constructeurs, prendre en compte des points de vue divergents, permettre à ces points de vue d'évoluer.

Notre modèle de processus de construction peut être lu selon différents niveaux de granularité, nous permettant de gérer la clarté du modèle dans une MAP générale, tout en en proposant des versions très détaillées dans des sous-map (cf. Figure 37).

Le modèle de processus d'amélioration continue est lui aussi représenté dans une MAP, et accompagné d'une description de chacune des stratégies possiblement mises en œuvre.

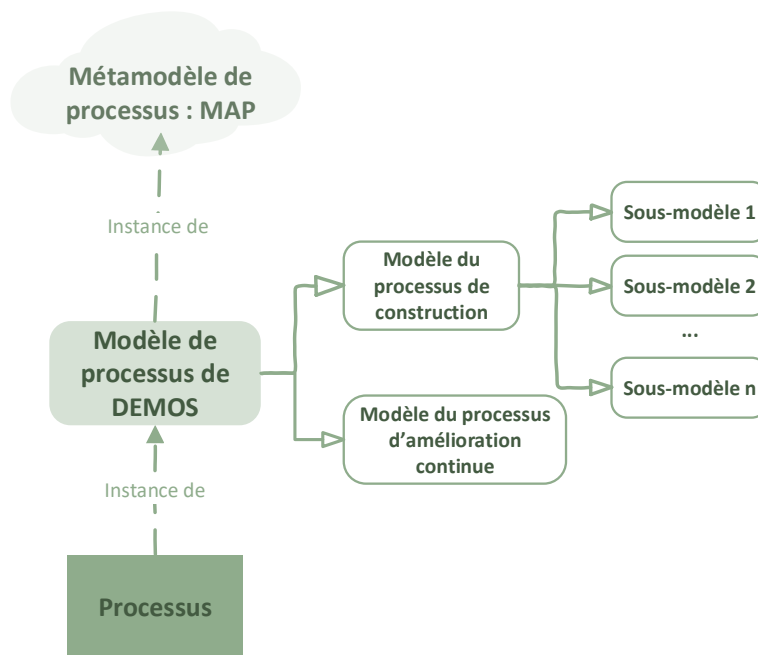


Figure 37 : Formalisation du modèle de processus de DEMOS

4.3.3. Méta-métamodélisation de la méthode DEMOS

Le fait de définir une méthode revient à définir un produit. Parallèlement, la construction d'une méthode peut être synthétisée dans un modèle de processus, afin de formaliser les aspects procéduraux pour la construction et l'évolution de ce produit. Ces deux éléments nous permettent de dresser une méta-métamodélisation de la méthode.

Le méta-métamodèle de produit de DEMOS est constitué de (cf. Figure 38) :

- Son modèle de produit, décrit par une syntaxe abstraite, une syntaxe concrète et une sémantique et représentant les aspects statiques, dynamiques, fonctionnels et prescriptifs du système d'information à construire. Ce modèle de produit est l'instance d'un diagramme de classe.
- Ses modèles de processus à « granularité variable », orientés stratégie et donc représentés grâce au formalisme MAP.
- Sa plateforme support, c'est-à-dire l'outil dédié au répertoriage et à la traçabilité de tous les éléments du produit.

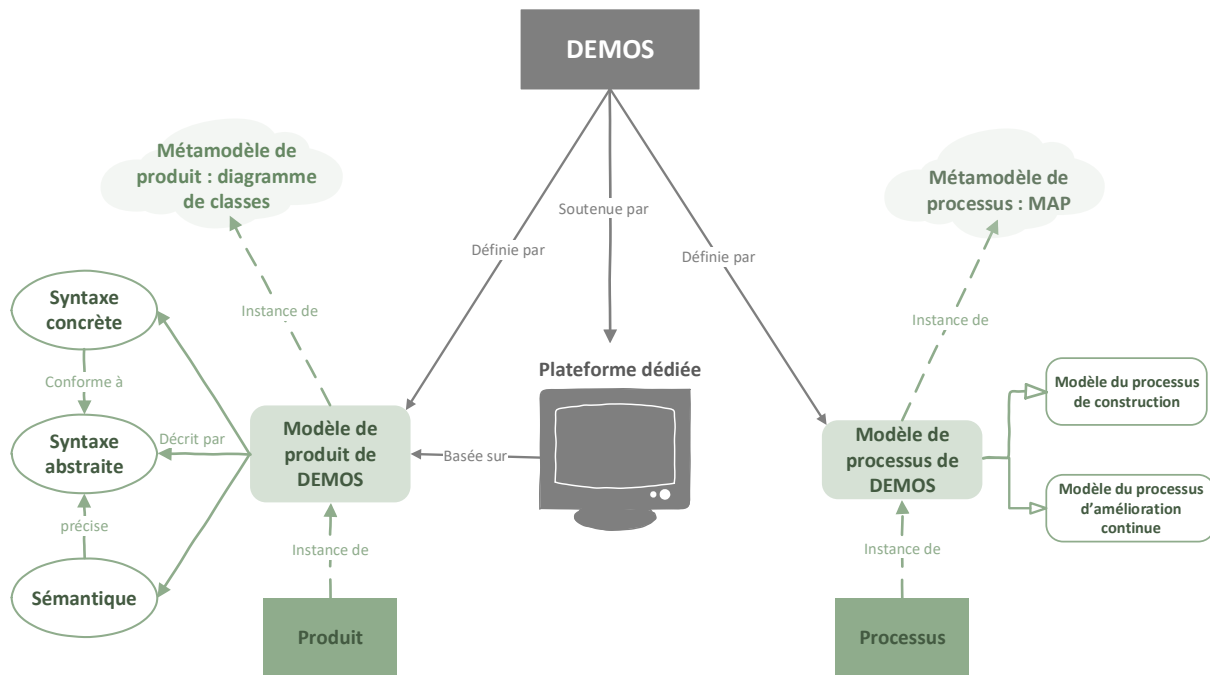


Figure 38 : Méta-métamodèle de produit pour la conception de la méthode DEMOS

Pour l'élaboration de DEMOS, nous avons également choisi de formaliser tous les aspects de la méthode, tout en conservant son caractère adaptable. Nous avons respecté les trois principes d'ingénierie des méthodes qui nous semblaient les plus opportuns : métamodélisation, réutilisation et modularité (cf. partie 4.2.2).

Le principe de métamodélisation est absolument central dans l'élaboration de la méthode DEMOS puisque celle-ci se structure autour de deux modèles : modèle de produit et modèle de processus, qui sont liés entre eux afin de garantir la plus grande cohérence. Le choix des métamodèles de produit et de processus a été éclairé par les objectifs assignés à chacun de ces modèles : souplesse pour le métamodèle de processus (MAP) et représentation simple et complète pour le métamodèle de produit (diagramme de classes).

Pour ce qui est de la réutilisation et de la modularité, nous avons mis en œuvre chacun de ces deux principes lors de la création de la méthode DEMOS. D'une part, nous avons choisi de réutiliser des morceaux de méthode qui nous semblaient en cohérence avec nos besoins. C'est le cas par exemple du formalisme MAP qui nous permet de représenter notre modèle de processus, mais aussi du formalisme de diagramme de classes proposé par UML qui nous permet de représenter notre modèle de produit. Certaines de nos activités sont également des emprunts à des méthodes existantes : User Stories empruntées aux méthodes agiles pour la description des fonctionnalités, principes de la méthode METAPLAN pour les ateliers participatifs, etc. Tous ces aspects sont détaillés dans le chapitre 5.

D'autre part, nous avons découpé notre méthode en intentions afin de la rendre modulaire. Chaque intention, c'est-à-dire chaque objectif de la méthode est lié à des stratégies à mettre en œuvre. Ces intentions, prises séparément, peuvent avoir une existence propre et être réutilisées dans le cadre d'autres projets.

Le méta-métamodèle de processus de DEMOS est représenté au travers d'une MAP (cf. Figure 39) :

- La définition de l'objectif de la construction de la méthode DEMOS a été atteinte grâce à deux stratégies : l'étude de méthodes existantes, la création d'une nouvelle méthode.

- La construction de la méthode s’est articulée autour de quatre stratégies : la métamodélisation et l’instanciation, l’utilisation d’un langage de modélisation, l’assemblage de composants et l’évaluation.

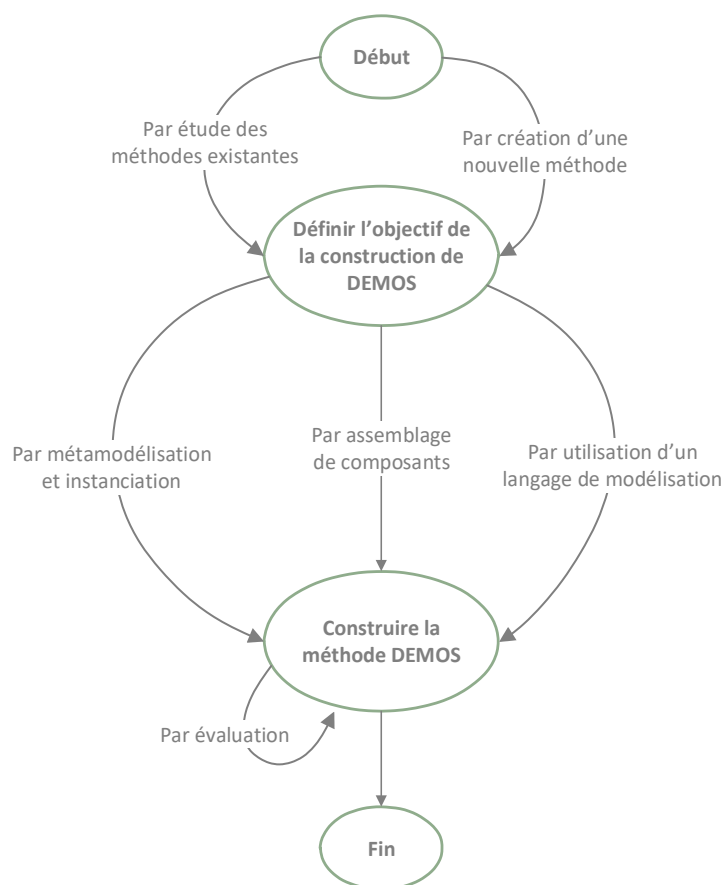


Figure 39 : Méta-métamodèle de processus pour la conception de la méthode DEMOS

Cette formalisation de notre méthode et de son processus de conception ancre véritablement notre contribution dans le domaine de l’ingénierie des méthodes. Le chapitre qui suit propose donc de présenter DEMOS conformément aux éléments de formalisme que nous avons établis dans ce quatrième chapitre.

4.4. Conclusion du chapitre

Ce chapitre de positionnement introduit notre contribution. Il s’agissait ici pour nous de replacer notre proposition autour de trois questions : Pourquoi ? Quoi ? Comment ?

La réponse à la première question est la réaffirmation de notre positionnement : notre contribution se propose de venir soutenir la démocratie dans les organisations via la construction de systèmes d’information. Nous avons défini plusieurs objectifs à notre contribution qui ont constitué pour nous à la fois un socle sur lequel nous appuyer, et une ligne d’horizon qui nous a guidés même dans les aspects les plus techniques de notre travail de recherche.

La réponse à la deuxième question nous a poussés à formaliser clairement les principes de notre contribution. Le choix de nous inscrire dans la lignée des méthodes participatives et de faire une proposition allant du recueil des exigences au développement, en passant par la conception, a été précisé et clarifié. L’empowerment démocratique des utilisateurs, leur position centrale dans la

construction du système d'information et les points de vue pluriels qu'ils peuvent avoir sont les piliers sur lesquels repose notre méthode.

La question du comment : comment construit-on une méthode ? nous a conduit à nous positionner dans le domaine de l'ingénierie des méthodes, et c'est dans ce domaine que nous avons donc puisé toute la méthodologie structurante de la conception d'une méthode. Le méta-métamodélisation que nous avons présentée garantit que nous proposons une méthode structurée, formalisée, précise et la plus complète possible. C'est cette méthode que nous présentons dans le chapitre suivant.

Chapitre 5

PRESENTATION DE DEMOS

La méthode DEMOS est un outil méthodologique. Elle a pour vocation de permettre la construction de systèmes d'information soutenant la démocratie des organisations. La méthode repose sur cinq grands principes, et c'est autour de ceux-ci que s'est articulée sa conception :

- Impliquer les utilisateurs finaux dans un processus participatif et démocratique,
- Favoriser un débat démocratique permettant l'émergence de points de vue,
- Concevoir un système d'information démocratique qui prend en compte ces points de vue,
- Assurer une traçabilité des points de vue dans le système d'information,
- Permettre l'évolution des points de vue dans le cycle de vie du système.

Nous avons choisi de décomposer la méthode en trois grandes parties, respectant ainsi les bonnes pratiques de l'ingénierie des méthodes. Dans un premier temps, nous présentons le modèle de produit de la méthode, c'est-à-dire le modèle que nous souhaitons instancier dans sa totalité lors de la mise en œuvre de la méthode. Dans un second temps, nous présentons les modèles de processus de la méthode, autrement dit les aspects « démarche » de notre méthode afin de définir comment parvenir à un modèle de produit complet et comment le faire évoluer. Dans un troisième et dernier temps, nous présentons la plateforme qui soutient la méthode et l'outil, de façon notamment à garantir sa cohérence (cf. Figure 40).

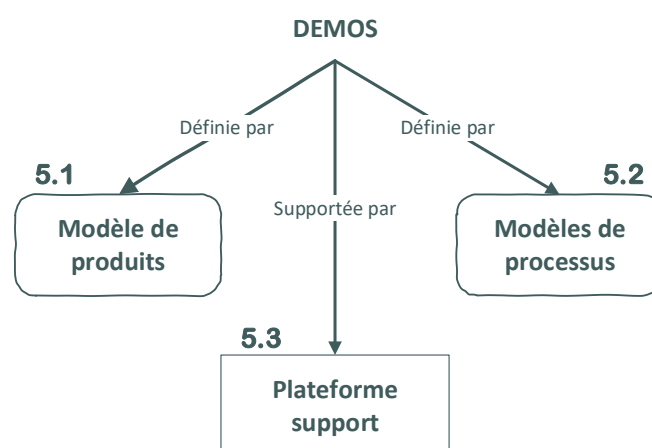


Figure 40 : Plan du chapitre 5

5.1. Le modèle de produit

Le modèle de produit formalise les aspects représentationnels de DEMOS. La méthode DEMOS propose de fournir à des développeurs un modèle complet permettant d'implanter une solution ; c'est

la forme de ce résultat attendu qui est décrite par le modèle de produit. Selon nous, la question de la démocratie passe avant tout par la question du langage, c'est-à-dire du vocabulaire, c'est pourquoi la méthode s'attache à fournir un modèle de données implémentable. Ce modèle est complété par une description fonctionnelle du domaine concerné, c'est ce que nous abordons dans un premier temps. Chaque concept du modèle de produit est ensuite défini et décrit au travers de sa syntaxe abstraite, de sa sémantique et de sa syntaxe concrète. Une évaluation du langage de modélisation que nous proposons vient compléter cette description.

5.1.1. Les aspects représentationnels du modèle de produit

Comme nous l'avons mentionné dans le chapitre 4, le modèle de produit de DEMOS couvre les aspects statiques, dynamiques, fonctionnels et prescriptifs du système d'information à construire (cf. Figure 41). Nous avons choisi de regrouper tous ces aspects dans un seul et même modèle de produit dans un but de simplicité tout d'abord, mais également afin que celui-ci soit appréhendable par les participants de la méthode que sont les futurs utilisateurs du système d'information (Bour, Soule-Dupuy et Vallès-Parlangeau 2019c).

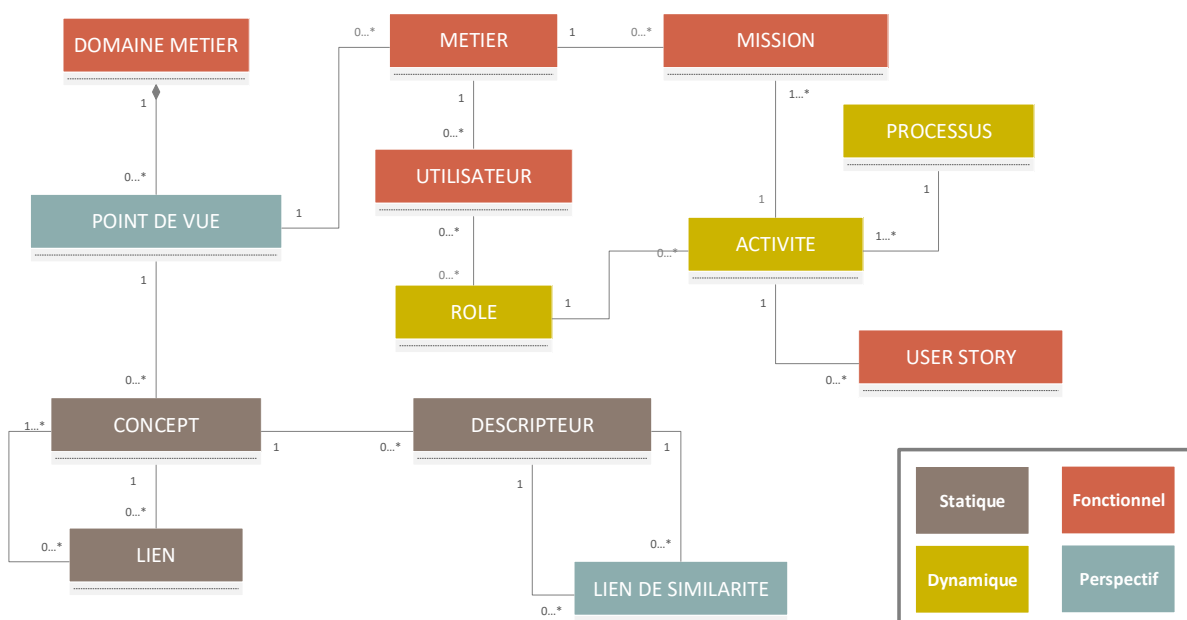


Figure 41 : Les aspects représentationnels du modèle de produit

5.1.1.1. L'aspect fonctionnel du modèle de produit

L'aspect fonctionnel du modèle de produit est décrit au travers de cinq concepts : le domaine métier, l'utilisateur, le métier, la mission et la User Story. Le **domaine métier** désigne le cadre fonctionnel dans lequel s'inscrit le système d'information en construction. Ce domaine métier permet de délimiter conceptuellement le périmètre du projet. Ainsi, si l'on s'intéresse par exemple au domaine « enfance » d'une mairie, on considère implicitement que les domaines « espaces verts » ou encore « troisième âge » sont hors du périmètre. Le domaine métier est le plus souvent issu d'un découpage par l'organisation elle-même des grands secteurs qu'elle recouvre.

L'utilisateur est au cœur de la méthode DEMOS. Il désigne toute personne de l'organisation qui est concernée par la construction du système d'information en cours. Il peut s'agir d'un utilisateur direct ou indirect, c'est-à-dire d'une personne qui utilisera le système d'information en produisant les

données ou d'une personne qui sera bénéficiaire de ces données, au travers de tableaux de bord par exemple. Les utilisateurs composent l'ensemble des participants de la méthode.

Le **métier** correspond à une nomenclature de l'organisation. Les utilisateurs sont regroupés dans l'organisation par métiers, qui sont le plus souvent décrits au travers de fiches de poste. Dans le cadre du modèle de produit de DEMOS, la notion de métier est forte car elle suppose, pour les utilisateurs qui le composent, un partage de visions sur les missions attendues notamment.

La **mission** désigne, de façon plus ou moins formelle, les tâches que les utilisateurs regroupés par métier doivent assurer. Les missions peuvent être celles qui sont attendues des utilisateurs par l'organisation, mais également celles qui leur incombent de façon plus officieuse. La méthode DEMOS doit permettre de faire apparaître l'ensemble de ces missions, qui sont partagées par l'ensemble des utilisateurs d'un métier.

La **User Story**⁹ est un terme emprunté à la méthode agile Scrum. Elle permet de définir les fonctionnalités attendues par les utilisateurs quant à leur système d'information. Elle est décrite en langage naturel par l'utilisateur lui-même. L'ensemble des User Stories délimite la couverture fonctionnelle du système d'information en construction.

L'aspect fonctionnel que le modèle de produit de DEMOS révèle permet de comprendre tout l'aspect métier que le système d'information viendra soutenir. Il est primordial de le définir de façon complète afin de s'assurer notamment que le périmètre du projet est conforme aux attentes des utilisateurs. L'aspect fonctionnel est déterminant pour le reste du modèle de produit de DEMOS puisqu'il est en lien direct avec les aspects dynamiques et prospectifs que nous allons maintenant décrire.

5.1.1.2. L'aspect dynamique du modèle de produit

L'aspect dynamique du modèle de produit est représenté au travers de trois concepts : le processus, l'activité et le rôle. Ces concepts sont ceux qui sont utilisés principalement dans le formalisme BPM (Business Process Modeling) et qui permettent de formaliser les processus de travail qui animent le domaine métier : le processus, l'activité et le rôle. Le **processus** de travail dans un premier temps consiste en une succession d'activités et peut impliquer un ou plusieurs utilisateurs. Chaque processus est mis en œuvre dans le but d'accomplir une mission de l'un des métiers de l'organisation. Le système d'information en construction est un support à ces processus métier qui méritent par conséquent d'être clairement identifiés et décrits.

L'activité est une décomposition du processus de travail. Chaque activité correspond à une tâche prise en charge par des utilisateurs afin de servir l'objectif du processus. Le processus est par conséquent composé d'une multitude d'activités qui nécessitent un recours au système d'information. Les activités doivent être décomposées de la façon la plus fine qu'il soit afin de comprendre au mieux les étapes et le séquençement du processus.

Le **rôle** est une façon de désigner un ensemble d'acteurs qui accomplissent une ou plusieurs activités du processus. Ces rôles peuvent correspondre de façon simple aux métiers, mais peuvent aussi être constitués d'un ensemble de métiers ou d'une sous partie d'un métier, c'est-à-dire de quelques utilisateurs seulement.

⁹ Le terme User Story peut être traduit en français par l'expression « récit utilisateur ». Nous avons choisi de ne pas traduire ce terme dans le cadre de cette thèse car il est utilisé et connu en anglais dans les organisations pratiquent les méthodes agiles.

L'aspect dynamique du modèle de produit de DEMOS permet de comprendre finement les différentes tâches concrètes effectuées au sein de l'organisation, mais aussi et surtout de percevoir leur enchaînement et leurs liens. C'est cette partie du modèle de produit qui sert à identifier les dépendances entre les métiers et l'impact que cela aura sur le système d'information en construction.

5.1.1.3. L'aspect perspectif du modèle de produit

L'aspect perspectif du modèle de produit est une particularité de la méthode que nous proposons. En effet, cet aspect n'est jamais représenté dans les méthodes de conception (cf. section 2.2) et est absolument déterminant dans la méthode que nous proposons. Il permet la prise en compte des points de vue, et donc le respect des principes que nous nous sommes fixés tels que le respect d'une pluralité de points de vue ou encore la traçabilité de ces derniers. L'aspect perspectif du modèle de produit se traduit par deux concepts : le point de vue et le lien de similarité. Le **point de vue** permet de regrouper des utilisateurs partageant une vision commune du domaine métier. Cette vision commune porte sur les missions de l'organisation et les rôles dans les processus de travail. Ainsi, dans une université par exemple, des points de vue divers tels que le point de vue « administratif », le point de vue « enseignement » ou encore le point de vue « apprenant » pourraient apparaître. En poursuivant l'objectif que DEMOS permette aux utilisateurs de ne pas se conformer à un point de vue dominant incarné par le système d'information lui-même, ce concept est absolument central.

Ceci étant, les points de vue n'existent pas de façon indépendantes et autonomes. Comme nous avons pu le voir à travers la description de l'aspect dynamique, les processus de travail dans l'organisation peuvent être partagés par des utilisateurs ayant des points de vue différents. De la même façon, comme nous le verrons dans la section suivante, les points de vue correspondent aussi à des langages différents, un usage du vocabulaire différent, mais qui pourtant n'exclut pas l'idée que les utilisateurs doivent pouvoir se comprendre entre eux. Pour cela un référentiel commun doit être au cœur du système d'information. Les **liens de similarité** ont pour but de dresser des ponts entre les différents vocabulaires des points de vue afin de définir les éléments de langage qu'ils partagent.

L'aspect perspectif du modèle de produit de DEMOS constitue l'une des particularités de cette méthode et doit être décrit de façon claire afin que tous les utilisateurs puissent se retrouver dans le point de vue qui leur correspond. Les principes de DEMOS reposent en grande partie sur cet aspect du modèle de produit et la mise en œuvre du processus DEMOS s'attache à parvenir à une définition des points de vue la plus correcte qu'il soit.

5.1.1.4. L'aspect statique du modèle de produit

L'aspect statique du modèle de produit est incarné par trois éléments : le concept, le descripteur et le lien. Ces éléments permettent d'obtenir la représentation conceptuelle du domaine métier et constituent le socle de la base de données du système d'information en construction. Chaque point de vue utilisant un langage, un vocabulaire propre, les concepts sont directement rattachés au point de vue, pour constituer une représentation conceptuelle par point de vue. Le **concept** peut être défini comme la représentation d'une entité du monde réel. Il peut être rapproché des notions de classe (en modélisation objet) ou d'entité (en modélisation relationnelle).

Les concepts sont décrits avec des **descripteurs**. Ceux-ci peuvent être rapprochés de la notion d'attribut. C'est le regroupement de ces descripteurs qui constitue le concept à proprement parler. Les concepts sont également mis en relation grâce à des **liens** qui portent en eux-mêmes une sémantique. Ces liens constituent une manière supplémentaire de décrire un concept.

L'aspect statique du modèle de produit de DEMOS est donc le révélateur des différents points de vue. C'est au travers de cette description du vocabulaire que les utilisateurs peuvent exprimer leurs différences, que ce soit au niveau du vocabulaire employé ou de la façon dont ils articulent les différents concepts composant leur langage.

Cette présentation des différents aspects du modèle de produit nous a permis d'introduire plusieurs concepts, mais également de comprendre comment la notion de point de vue émerge grâce à l'aspect fonctionnel du modèle, et comment elle se répand ensuite dans une grande partie du modèle de produit, au niveau des aspects statiques et dynamiques notamment.

5.1.2. Syntaxe abstraite et sémantique du modèle de produit

Maintenant que nous avons introduit les différentes facettes du modèle de produit en fonction des aspects qu'il recouvre, nous pouvons décrire de façon plus détaillée la syntaxe abstraite de DEMOS. Cette syntaxe abstraite est en quelque sorte le langage de la méthode, et nous avons choisi de la représenter comme une instance de diagramme de classes simplifié (seule la partie statique des classes est représentée). La Figure 42 ci-dessous présente cette syntaxe abstraite du modèle de produit que nous décrivons ensuite.

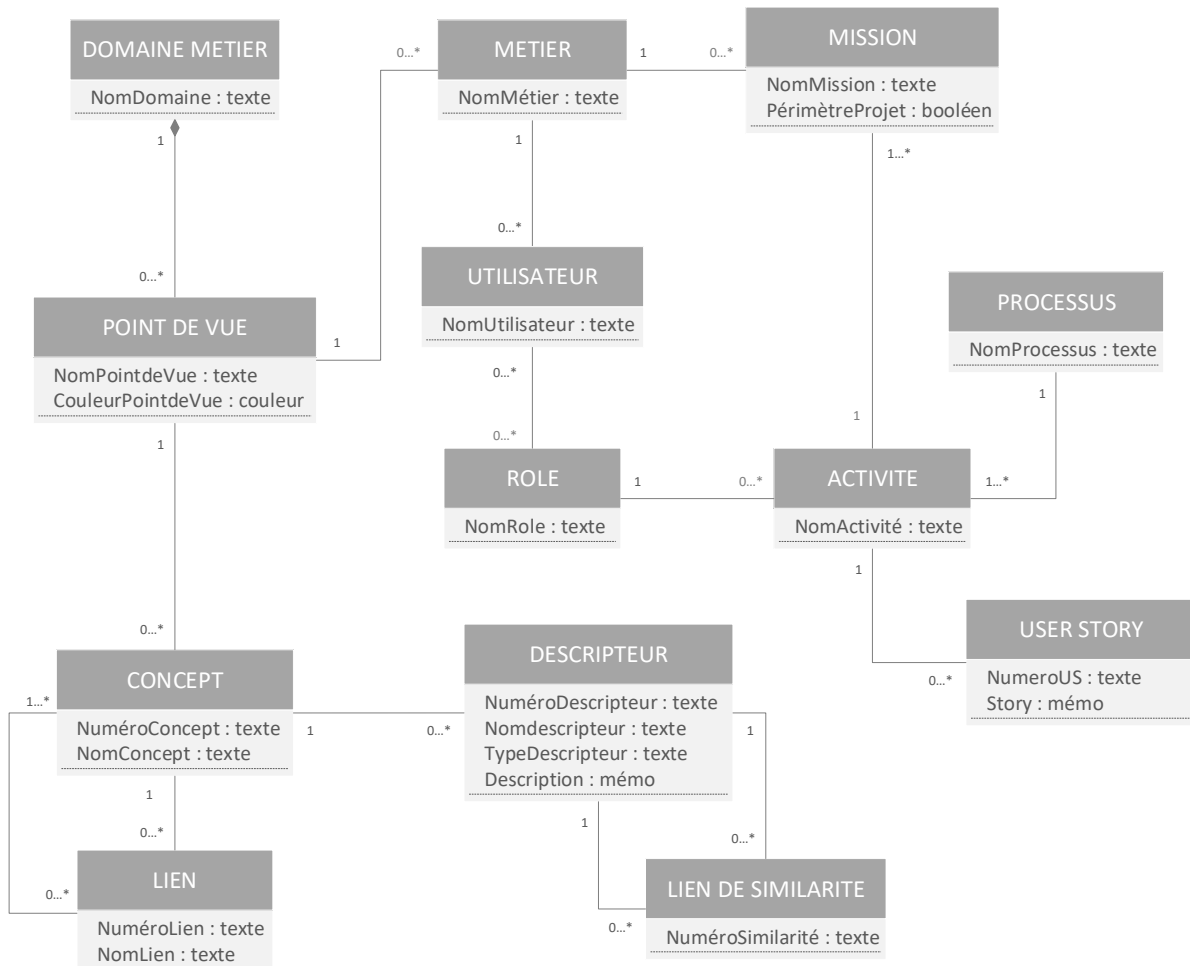


Figure 42 : Syntaxe abstraite du modèle de produit de DEMOS

Dans la section 3.1.2.2, nous avons présenté les caractéristiques de Krogstie (Krogstie 2003), qui sont les caractéristiques que doit respecter un langage de modélisation. Nous avons cherché à être conforme à chacun de ces points lors de la création de cette syntaxe abstraite :

- « *Le langage doit être en adéquation avec le domaine métier* ». Notre proposition est en adéquation avec le contexte des organisations et de leurs processus de travail. Notre méthode de conception s'attachant à permettre de construire des systèmes d'information soutenant la démocratie des organisations, nous avons choisi de l'axer principalement sur les processus de travail. Pour cette raison, des éléments de la syntaxe abstraite tels que **métier, mission, processus, activité** nous ont semblé pertinents.
- « *Le langage doit être en adéquation avec les connaissances des concepteurs du domaine* ». Pour nous, les concepteurs du domaine sont les utilisateurs eux-mêmes, qui sont considérés comme les co-constructeurs de leur système afin de respecter les principes de démocratie que nous nous sommes fixés. Le modèle de produit que nous proposons est par conséquent simple et synthétique car notre volonté était de le rendre appréhendable par des non informaticiens. Il comprend 13 classes, avec un maximum de quatre attributs par classe, et les termes que nous utilisons ne sont jamais techniques. C'est pour cette raison notamment que nous avons choisi d'utiliser les notions de **concepts** et **descripteurs** qui nous semblent appartenir à un langage naturel et facilement manipulable (en lieu et place des notions de classe et d'attribut qui portent en eux une acception bien plus technique).
- « *Le langage doit être en mesure de capter « toutes » les connaissances des participants* ». Le modèle de produit que nous décrivons ici au travers de sa syntaxe abstraite se veut suffisamment complet pour capter les connaissances nécessaires à l'élaboration d'un système d'information support aux processus de travail. La notion de **point de vue** nous permet de nous assurer que « toutes » les connaissances seront captées, au sens où ce ne sont pas seulement les connaissances émergeant d'un point de vue dominant qui seront prises en compte.

La syntaxe abstraite du modèle de produit de DEMOS ne saurait être appréhendable sans y ajouter des éléments permettant de préciser la sémantique de chacun des attributs. Pour cette raison, nous présentons dans la Table 3 le dictionnaire des attributs de la syntaxe abstraite.

Table 3 : Sémantique du modèle de produit de DEMOS

Attribut	Type	Description
CouleurPointdeVue	Couleur	Couleur de chacun des points de vue choisie par les participants, qui sera ensuite répercutée sur les modèles conceptuels.
Description	Memo	Texte rédigé par les utilisateurs et le modérateur de la méthode pour définir la sémantique d'un descripteur.
NomActivité	Texte	Nom des activités composant chaque processus.
NomConcept	Texte	Nom donné par les utilisateurs à un concept.
NomDescripteur	Texte	Nom donné par les utilisateurs aux descripteurs correspondant à un concept.
NomDomaine	Texte	Nom du domaine métier concerné par le projet.
NomLien	Texte	Nom du lien entre deux concepts attribué par les utilisateurs.
NomMétier	Texte	Nom des métiers des utilisateurs participant à la méthode.
NomMission	Texte	Nom des missions choisi par les participants et correspondant à leur métier.
NomPointdeVue	Texte	Nom de chacun des points de vue du domaine métier choisi par les participants.
NomProcessus	Texte	Nom des processus défini par les participants.
NomRôle	Texte	Nom de chacun des rôles défini dans le processus et effectuant des activités.




NomUtilisateur	Texte	Prénom et nom de chacun des participants de la méthode.
NuméroConcept	Texte	Numéro défini de manière aléatoire pour identifier un concept.
Numérodescripteur	Texte	Numéro défini de manière aléatoire pour identifier un descripteur.
NuméroLien	Texte	Numéro défini de manière aléatoire pour identifier un lien entre deux concepts.
NuméroSimilarité	Texte	Numéro défini de manière aléatoire pour identifier un lien de similarité entre deux descripteurs de modèles conceptuels différents.
NuméroUS	Texte	Numéro défini de manière aléatoire pour identifier une User Story.
PérimètreProjet	Booléen	Booléen permettant d'identifier si la mission correspond ou non au périmètre du projet.
Story	Mémo	Texte rédigé par les participants pour définir la User Story. Il se décompose en trois parties : <i>en tant que ...</i> , <i>je veux ...</i> , <i>afin de ...</i>
TypeDescripteur	Texte	Type de chacun des descripteurs. Ce type peut être : texte, numérique, mémo, date, image.



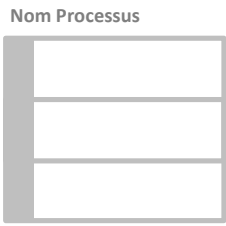
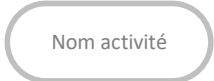



La syntaxe abstraite et la sémantique du modèle de produit de DEMOS nous permettent d'identifier de façon claire le produit attendu de la méthode. Afin que cette syntaxe abstraite soit manipulable par les utilisateurs co-constructeurs de leur système d'information, elle doit être complétée par une syntaxe concrète du modèle de produit, c'est ce que nous abordons dans la section suivante.


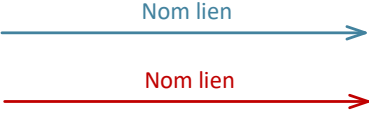

5.1.3. Syntaxe concrète du modèle de produit

La syntaxe concrète décrit la représentation, c'est-à-dire la notation des éléments de syntaxe abstraite du modèle de produit. Nous avons souhaité respecter les critères de Moody (présentés en section 3.1.2.2) afin que ces éléments de notation visuelle soient pertinents. Nous présentons dans la Table 4 chacune des notations visuelles attachées aux classes de la syntaxe abstraite, et nous évaluons ensuite dans la Table 5 tous ces éléments de syntaxe concrète.

Table 4 : Syntaxe concrète du modèle de produit de DEMOS

Classe	Notation visuelle	Justification
Domaine métier		Le domaine métier est représenté par un grand rectangle gris. Il symbolise le cadre du projet et porte une couleur neutre. Son nom est noté au-dessus du cadre.
Point de vue		Le point de vue est représenté par un rectangle aux coins arrondis. Cela symbolise le cadre perspectif dans lequel s'inscrivent les utilisateurs, mais également son caractère souple et évolutif au fil de la méthode. Chaque point de vue est identifié par une couleur et son nom est noté au-dessus du cadre.
Métier		Chaque métier est représenté par un ovale avec le nom du métier noté au milieu. Cela symbolise le fait qu'un métier est un regroupement d'utilisateurs.

<p>Mission</p>		<p>Chaque mission est notée dans un rectangle plein et peut être barrée si la mission est hors du périmètre du projet.</p>
<p>Utilisateur</p>		<p>Chaque utilisateur est représenté par un bonhomme avec son nom écrit au-dessus. Le symbole du bonhomme permet de personifier les utilisateurs.</p>
<p>Processus</p>		<p>Le processus est représenté par un cadre avec le nom du processus noté au-dessus. Le cadre est divisé en lignes, qui permettent de noter les activités correspondant à chacun des rôles. Cette notation des processus correspond à la norme BPMn.</p>
<p>Activité</p>		<p>Les activités sont représentées par des bulles avec le nom de l'activité noté à l'intérieur. Ces bulles sont placées à l'intérieur de chaque ligne du processus, en face du rôle correspondant à celui qui effectue cette activité.</p>
<p>Rôle</p>		<p>Le rôle est écrit à la verticale au début de chacune des lignes du processus, en blanc sur fond gris.</p>
<p>User Story</p>		<p>Chaque User Story est représentée suivant le formalisme utilisé dans la méthode SCRUM. La User Story est numérotée en haut à droite et est décrite par trois éléments : <i>en tant que</i> (pour noter le nom du métier), <i>je veux</i> (pour décrire la fonctionnalité souhaitée), <i>afin de</i> (pour noter la justification du choix de la fonctionnalité). Chaque User Story correspondant à un point de vue, elle porte la couleur de celui-ci.</p>
<p>Concept</p>		<p>Chaque concept est représenté par un rectangle et le nom du concept est noté dans un cadre plein. Cette notation se rapproche du diagramme de classe, bien qu'elle soit simplifiée. Le concept étant un élément constitutif de la représentation conceptuelle par point de vue, il porte la couleur de celui-ci.</p>

Descripteur		<p>Le nom de chacun des descripteurs est écrit à l'intérieur du cadre du concept correspondant. Le descripteur porte lui aussi la couleur du point de vue correspondant</p>
Lien		<p>Le lien entre deux concepts est représenté par une flèche orientée sur laquelle le nom du lien est noté. Chaque lien porte la couleur du point de vue associé au modèle conceptuel auquel il appartient. Le sens de la flèche indique le sens de lecture du lien.</p>
Lien de similarité		<p>Le lien de similarité entre descripteurs de modèles conceptuels différents est représenté par une flèche bilatérale qui symbolise la similarité entre les descripteurs. Cette flèche est de couleur neutre car elle n'appartient pas à un point de vue particulier mais fait le lien entre deux points de vue. Le lien de similarité porte un numéro noté sur la flèche.</p>

Les notations visuelles que nous présentons ici sont celles qui sont utilisées durant la mise en œuvre concrète de la méthode que nous aborderons dans la section sur le modèle de processus de DEMOS.

Dans le dictionnaire des données, les concepts et descripteurs sont représentés de la façon suivante :

N° concept Nom concept	N° - Nom descripteur	<u>Type descripteur</u>	<i>Description</i>
	N° - Nom descripteur	Type descripteur	<i>Description</i>
	N° - Nom descripteur	Type descripteur	<i>Description</i>
	N° - Nom descripteur	Type descripteur	<i>Description</i>
N° concept Nom concept	N° - Nom descripteur	<u>Type descripteur</u>	<i>Description</i>
	N° - Nom descripteur	Type descripteur	<i>Description</i>
	N° - Nom descripteur	Type descripteur	<i>Description</i>
	N° - Nom descripteur	Type descripteur	<i>Description</i>

Figure 43 : Représentation des concepts et descripteurs dans le dictionnaire des données

Tous ces éléments de notation visuelle ont fait l'objet d'une évaluation au regard des critères de Moody (cf. Table 5). Ces critères nous ont semblé pertinents pour juger de l'utilité et de l'utilisabilité de la syntaxe concrète de notre modèle de produit. L'évaluation est pour nous d'autant plus importante que tous ces éléments de notation visuelle sont manipulés par les utilisateurs eux-mêmes, c'est à dire par des non experts de la conception de systèmes d'information, afin de respecter les principes de démocratie de DEMOS.

Table 5 : Evaluation de la syntaxe concrète de DEMOS

Critère de Moody	Validation pour DEMOS	Justification
Clarté sémiotique	oui	Le modèle de produit de DEMOS compte autant d'éléments de syntaxe abstraite (classes) que d'éléments de syntaxe concrète.
Discriminabilité perceptuelle	oui	Chaque élément de la syntaxe concrète de DEMOS est unique. Lorsque les formes sont les mêmes, elles peuvent être distinguées par les couleurs ou par des nuances (telles que les coins arrondis pour un rectangle).
Transparence sémantique	oui	Un effort a été fait dans la conception de la syntaxe concrète pour qu'un maximum d'éléments soit porteur d'une symbolique (par exemple : les rectangles représentent des cadres symbolisant des éléments de cadrage du projet).
Management de la complexité	non	Dans le cadre du modèle de produit de DEMOS, cette question n'est pas abordée.
Intégration cognitive	oui	Le processus de la méthode DEMOS permet de guider les utilisateurs dans l'usage des éléments de notation visuelle.
Expressivité visuelle	oui	Les couleurs et les tailles des éléments permettent d'améliorer l'expressivité des éléments de notation.
Double codage	oui	Du texte accompagne et complète tous les éléments de notation visuelle.
Economie graphique	oui	Un effort particulier a été fait pour parvenir à limiter le nombre d'éléments de la syntaxe abstraite, et par conséquent de la syntaxe concrète.
Ajustement cognitif	non	Pour le moment la notation graphique est identique pour tous les participants. Une évolution possible serait d'adapter cette notation aux publics concernés.

La présentation du modèle de produit de DEMOS est essentielle pour comprendre le résultat attendu de la méthode. Nous avons montré, au travers de la syntaxe abstraite, de la sémantique et de la

syntaxe concrète que notre volonté est d'avoir une représentation globale du système d'information à construire, dans ses aspects fonctionnels, dynamiques, perspectifs et statiques. Ainsi, la méthode peut permettre une implémentation directe du modèle des données notamment (c'est-à-dire de la partie statique de notre modèle de produit), tout en respectant les principes que nous nous sommes fixés : le respect des points de vue et leur traçabilité dans le système. Afin de comprendre maintenant le cheminement nécessaire à l'obtention de ce modèle de produit dans sa globalité, nous présentons dans la section suivante les modèles de processus de DEMOS.

5.2. Les modèle de processus

Les modèles de processus de la méthode nous permettent de présenter la/les façons dont DEMOS peut être mise en œuvre, l'objectif étant d'instancier le modèle de produit que nous avons présenté dans la section précédente. Comme expliqué dans le chapitre 4, nous avons choisi de représenter ces modèles de processus sous la forme de MAP, qui peuvent ensuite être décomposées en sous-map. Nous garantissons ainsi que, dans une visée démocratique, la méthode est adaptable au contexte dans lequel elle est déroulée, et qu'elle peut être enrichie au fil du temps. L'essentiel est que les intentions que nous nous sommes fixées soient atteintes, par des stratégies diverses, en appliquant ces dernières de façon totale ou partielle, ou en proposant de nouvelles stratégies si besoin.

Nous présentons dans un premier temps les deux MAP générales de DEMOS : MAP du processus de construction et MAP du processus d'amélioration continue, afin de dévoiler les principales intentions de la méthode, et les grandes stratégies permettant de les atteindre. Les deux MAP forment une démarche complète allant de la construction de système d'information à son amélioration et son adaptation à l'évolution des points de vue. Ainsi, alors que la MAP de construction est mise en œuvre une seule fois pour créer le système, la MAP d'amélioration continue peut être mise en œuvre autant de fois que nécessaire durant le cycle de vie du système (cf. Figure 44).

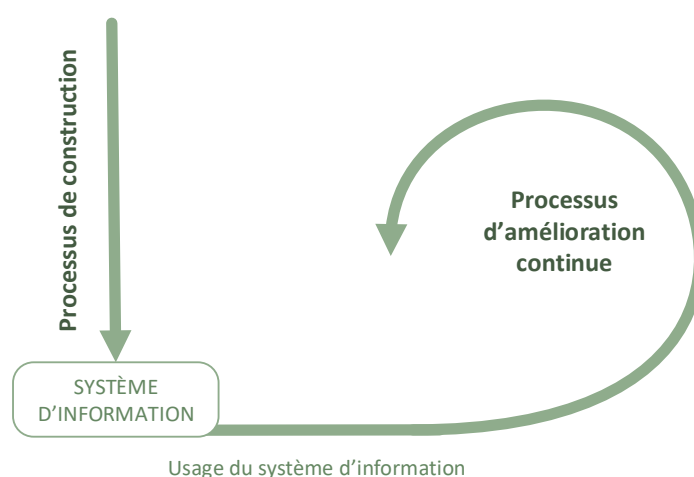


Figure 44 : Enchaînement des processus de DEMOS

Bien que la MAP ne soit pas un cadre chronologique, nous proposons d'y ajouter un séquençement en grandes phases de projet, c'est ce que nous décrivons dans un second temps. Dans un troisième temps, nous décomposons chacune des sections de la MAP de construction afin de présenter toutes les stratégies envisageables de façon plus précise. Dans un quatrième temps, nous décrivons en détail chacun des éléments de la MAP d'amélioration continue. Dans un dernier temps, nous détaillons les principaux outils utilisés durant les ateliers.

5.2.1. MAP générales

Les modèles de processus de DEMOS permettent de formaliser les aspects procéduraux de la méthode, c'est-à-dire la démarche qui s'applique. La Figure 45 présente la MAP générale du processus de construction de système d'information de DEMOS, tandis que la Figure 46 présente la MAP générale du processus d'amélioration continue de système d'information de DEMOS.

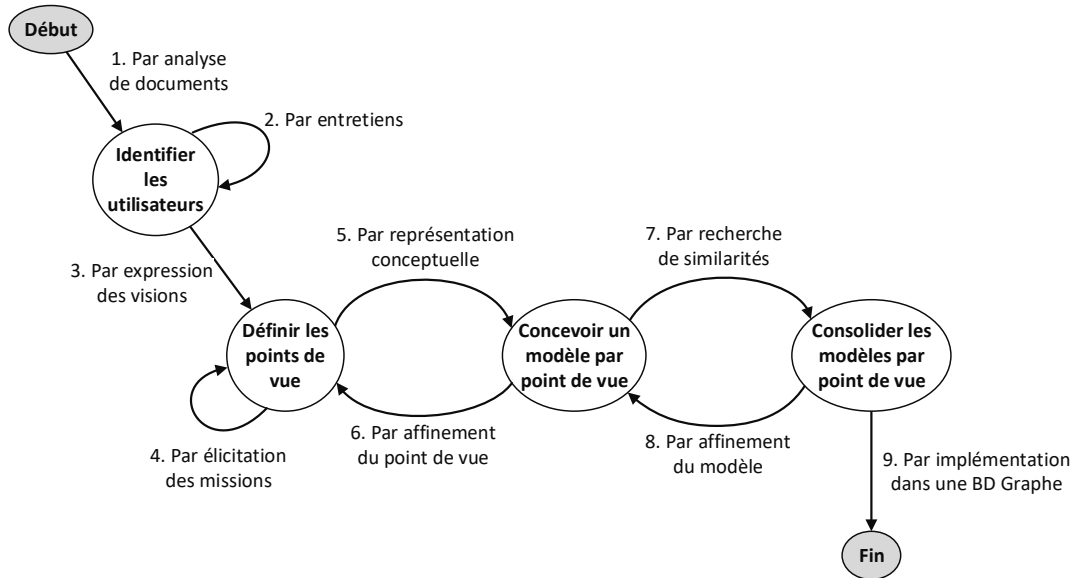


Figure 45 : MAP générale du processus de construction

Par souci de clarté, nous avons choisi de commencer la numérotation de la MAP d'amélioration continue à partir de la fin de la numérotation de la MAP de construction. Cela nous permet par la suite de faire référence à des sections de chacune des MAP sans qu'il n'y ait de confusion.

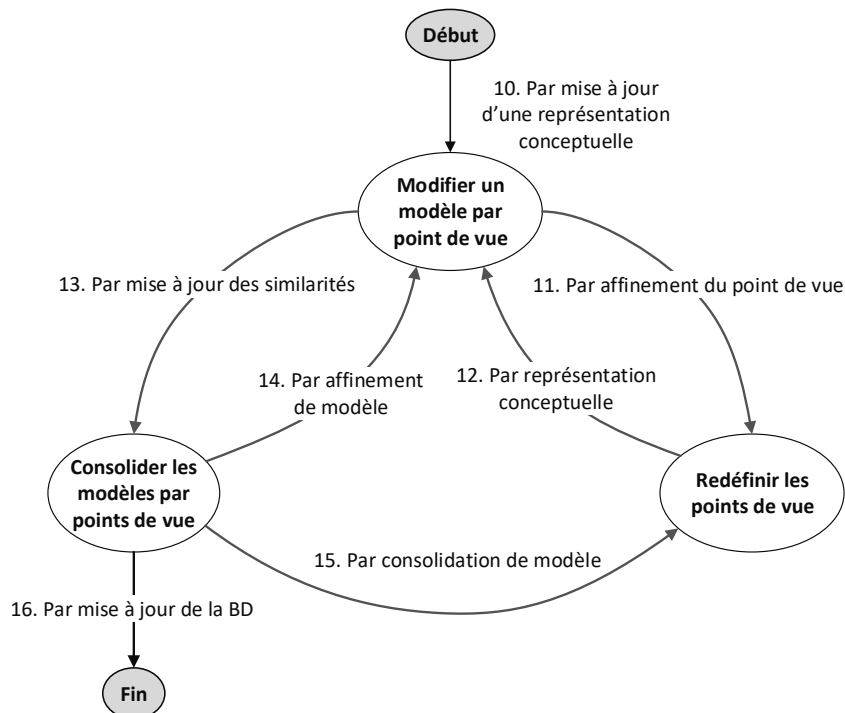


Figure 46 : MAP générale du processus d'amélioration continue

Les objectifs de démocratie que nous avons définis ont été les lignes directrices de la conception de ces MAP générales du processus de DEMOS. Chaque intention des MAP est donc directement liée aux principes de DEMOS, comme le présente la Table 6.

Table 6 : Lien entre principes et intentions de DEMOS

Principes de DEMOS	Intentions de la MAP	MAP correspondante
Impliquer les utilisateurs finaux dans un processus participatif et démocratique	Identifier les utilisateurs	MAP du processus de construction
Favoriser un débat démocratique permettant l'émergence de points de vue	Définir les points de vue	
Concevoir un système d'information démocratique qui prend en compte ces points de vue	Concevoir un modèle par point de vue	
Assurer une traçabilité des points de vue dans le système d'information	Consolider les modèles par points de vue	
Permettre l'évolution des points de vue dans le cycle de vie du système	Toutes les intentions	MAP du processus d'amélioration continue

Les deux MAP générales que nous avons présentées constituent, comme nous l'avons dit, un cadre procédural souple pour DEMOS. Elles méritent bien entendu d'être affinées, afin de décrire la façon dont les stratégies peuvent être concrètement mises en œuvre. C'est ce que nous présentons dans les sections 5.2.3. et 5.2.4. Avant cela, il nous a semblé nécessaire de présenter le séquençement général de DEMOS en grandes phases. C'est cette description qui fait l'objet de la section suivante.

5.2.2. Séquençement de la méthode

Bien que le métamodèle MAP permette de s'affranchir d'une vision chronologique d'un processus, nous avons choisi de définir les grandes lignes d'un séquençement de la méthode, en lien direct avec les MAP générales proposées (cf. Table 7). Cela nous permet de définir un cadre clair à DEMOS. Notre volonté étant que la méthode soit reproductible, il nous a semblé indispensable de rattacher nos MAP générales aux grandes lignes des phases d'un projet informatique et de découper sa mise en œuvre en ateliers en identifiant les participants à solliciter. On retrouve, parmi les participants, le rôle de modérateur. En effet, la description qui est faite de la méthode aujourd'hui, le langage simple qu'elle mobilise, et les étapes formalisées qui la structurent permettent de mettre la méthode dans les mains d'une personne externe ou d'une personne membre de l'organisation pour assurer ce rôle de modérateur, sans qu'elle soit un « expert-méthode » (Front et al. 2017).

Table 7 : Séquencement de la méthode DEMOS

Phase	Atelier	Participants	Sections concernées
Phase préparatoire		<ul style="list-style-type: none"> • Modérateur • Parties prenantes de l'organisation (client, DSI, DRH, syndicats, etc.) 	<p>Extrait de la MAP de construction :</p>
Phase de recueil des exigences	Atelier de cadrage	<ul style="list-style-type: none"> • Modérateur • Tous les utilisateurs 	<p>Extrait de la MAP de construction :</p>
Phase de conception	Ateliers point de vue	<ul style="list-style-type: none"> • Modérateur • Utilisateurs regroupés par points de vue 	<p>Extrait de la MAP de construction :</p>
	Atelier de mise en commun	<ul style="list-style-type: none"> • Modérateur • Tous les utilisateurs 	<p>Extrait de la MAP de construction :</p>

Phase d'implémentation		<ul style="list-style-type: none"> • Modérateur • Utilisateurs • Développeurs 	<p>Extrait de la MAP de construction :</p> <pre> graph TD A((Consolider les modèles par point de vue)) -- "9. Par implémentation dans une BD Graphe" --> B((Fin)) </pre>
Phase d'amélioration continue	<p>Atelier Point de vue Et / ou Atelier de mise en commun Et / ou Atelier de cadrage</p>	<ul style="list-style-type: none"> • Modérateur • Utilisateurs • Développeurs 	<p>Toute les sections de la MAP d'amélioration continue</p>

La méthode DEMOS repose sur le séquençage que nous venons de présenter. Ceci étant, en poursuivant l'objectif d'une méthode adaptable, chacune des sections des MAP de construction peut ensuite être affinée, et c'est dans la présentation des sous-map de DEMOS que l'on peut percevoir les possibilités d'adaptation de stratégies. Cela permet de proposer une méthode souple dans un cadre clairement formalisé.

5.2.3. Décomposition du processus de construction

Le processus de construction de la méthode DEMOS couvre les phases préparatoires, de recueil des exigences, de conception et d'implémentation d'un projet de construction de système d'information. Cette richesse rend nécessaire le fait de détailler chacune des étapes, afin de s'assurer que la méthode soit reproductible et appréhendable par n'importe quelle personne qui assurerait le rôle de modérateur de la méthode.

Pour cette raison, chaque section de la MAP (intention de départ, intention d'arrivée, stratégie mise en œuvre) est précisée sous la forme d'une sous-map. Pour les sections de la phase préparatoire et de la phase d'implémentation, la sous-map est accompagnée d'un tableau synthétique présentant une description de chacune des sous-sections. Pour les sections qui sont mises en œuvre durant les ateliers (atelier de cadrage, ateliers point de vue et atelier de mise en commun) le tableau synthétique accompagnant la sous-map précise en plus les moyens mis en œuvre et les résultats attendus.

5.2.3.1. Section 1 : <Début, Identifier les utilisateurs, Par analyse de documents>

La section 1 de DEMOS permet de débiter le processus, elle fait donc partie de ce que l'on nomme la phase préparatoire. Le but de cette section est pour le modérateur d'identifier tous les futurs utilisateurs qui seront les participants de la méthode. Cette section comporte cinq stratégies qui

peuvent être totalement ou partiellement mises en œuvre pour identifier les utilisateurs qui participeront au projet (cf. Figure 47).

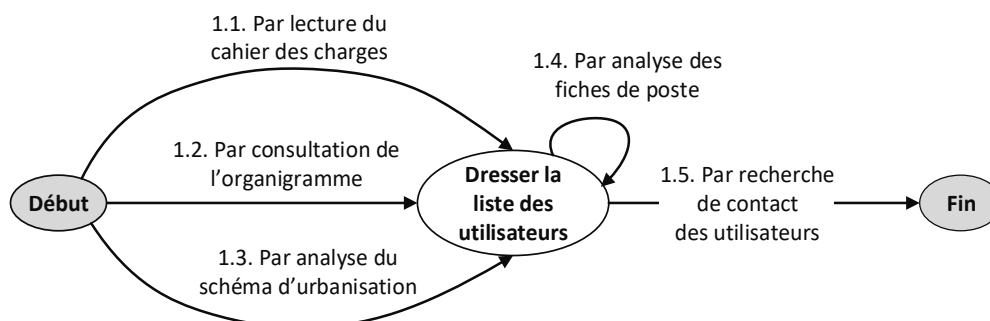


Figure 47 : Sous-map de la section 1

Cette section a pour but de permettre une première instanciation du modèle de produit de DEMOS en identifiant le **domaine métier**¹⁰ et les **utilisateurs**. Chaque stratégie mise en œuvre dans ce but est décrite dans la Table 8 suivante.

Table 8 : Synthèse de la section 1 de DEMOS

N°	Nom de section	Description
1.1	<Début, Dresser la liste des utilisateurs, Par lecture du cahier des charges >	Les projets informatiques font généralement l'objet d'un cahier des charges préalable proposé par le client. Ce cahier des charges mentionne un certain nombre d'acteurs pouvant faire partie de la liste des futurs utilisateurs.
1.2	<Début, Dresser la liste des utilisateurs, Par consultation de l'organigramme >	L'organigramme de l'organisation concernée par le projet peut permettre d'établir une cartographie des services et personnes concernés par le projet.
1.3	<Début, Dresser la liste des utilisateurs, Par analyse du schéma d'urbanisation >	Le schéma d'urbanisation de l'organisation, s'il existe, peut donner des indications sur les services ou personnes concernés par le projet dans sa couche fonctionnelle.
1.4	<Dresser la liste des utilisateurs, Dresser la liste des utilisateurs, Par analyse des fiches de poste >	La consultation des fiches de poste des premiers utilisateurs identifiés peut permettre de venir affiner la liste d'utilisateurs participant au projet.
1.5	<Dresser la liste des utilisateurs, Fin, Par recherche de contacts des utilisateurs >	La liste d'utilisateurs doit être complétée avec les contacts de toutes les personnes identifiées.

5.2.3.2. Section 2 : <Identifier les utilisateurs, Identifier les utilisateurs, Par entretiens>

La section 2 a pour but de venir affiner la liste d'utilisateurs par le biais d'entretiens. Cette section implique de nombreux acteurs de l'organisation, et des stratégies peuvent venir compléter cette section s'il semble pertinent que d'autres acteurs de l'organisations soient entendus. On compte ici

¹⁰ Durant toute la présentation du modèle de processus de DEMOS, chaque référence à un élément du modèle de produit sera reconnaissable par la police que nous avons choisie : gras et italique.

cinq stratégies, qui une fois encore peuvent être totalement ou partiellement mises en œuvre afin d'affiner la liste d'utilisateurs qui devront participer au projet (cf. Figure 48).

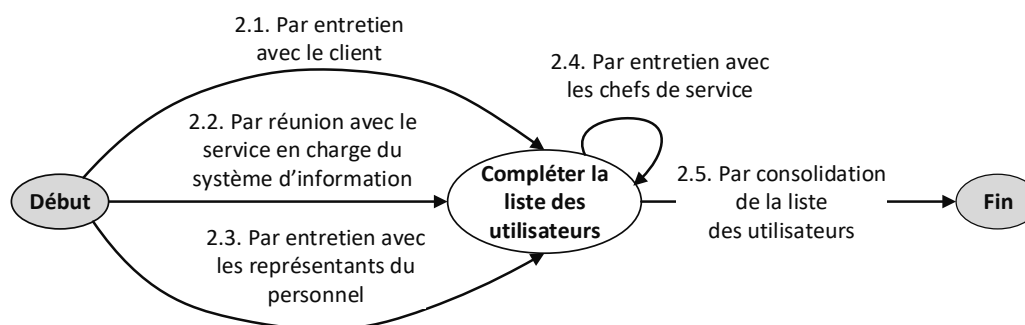


Figure 48 : Sous-map de la section 2

La mise en œuvre de cette section permet de confirmer ou de modifier la première instanciation du modèle de produit qui aura été faite : **domaine métier** et **utilisateurs**. Chaque stratégie mise en œuvre est décrite dans la Table 9 suivante.

Table 9 : Synthèse de la section 2 de DEMOS

n°	Nom de section	Description
2.1	<Début, Compléter la liste des utilisateurs, Par entretien avec le client >	La définition par le client du domaine métier concerné par le projet peut permettre de faire apparaître de nouveaux utilisateurs. La vision globale du projet de ce dernier peut également lui permettre de valider la liste précédemment établie.
2.2	<Début, Compléter la liste des utilisateurs, Par réunion avec le service en charge du système d'information >	Le service en charge du système d'information, qui connaît la liste des utilisateurs actuels des outils informatiques de l'organisation peut venir compléter la liste d'utilisateurs à convier dans le processus de conception.
2.3	<Début, Compléter la liste des utilisateurs, Par entretien avec les représentants du personnel >	Les responsables du personnel (ex : les responsables syndicaux) ont une vision interne des positionnements des services/des personnes quant aux outils informatiques. Ils peuvent venir compléter la liste d'utilisateurs de façon pertinente.
2.4	<Compléter la liste des utilisateurs, Compléter la liste des utilisateurs, Par entretien avec les chefs de service >	Les chefs de service qui connaissent les équipes "à jour" peuvent venir affiner la liste en y ajoutant notamment des utilisateurs qui sont actuellement en cours de recrutement par exemple. Ils peuvent également compléter la liste de contacts.
2.5	<Compléter la liste des utilisateurs, Fin, Par consolidation de la liste des utilisateurs >	La liste des utilisateurs est consolidée après ces entretiens et les utilisateurs peuvent alors être contactés et conviés pour la suite du processus.

5.2.3.3. **Section 3 : <Identifier les utilisateurs, Définir les points de vue, Par expressions des visions>**

La section 3 du modèle de processus de DEMOS est la première étape de la phase de recueil des exigences et est appliquée lors d'un atelier de cadrage où tous les utilisateurs sont conviés. Durant cet atelier, le modérateur de la méthode tâchera de mettre en œuvre tout ou partie des sept stratégies proposées (cf. Figure 49) afin d'obtenir une première identification des différents points de vue des utilisateurs.

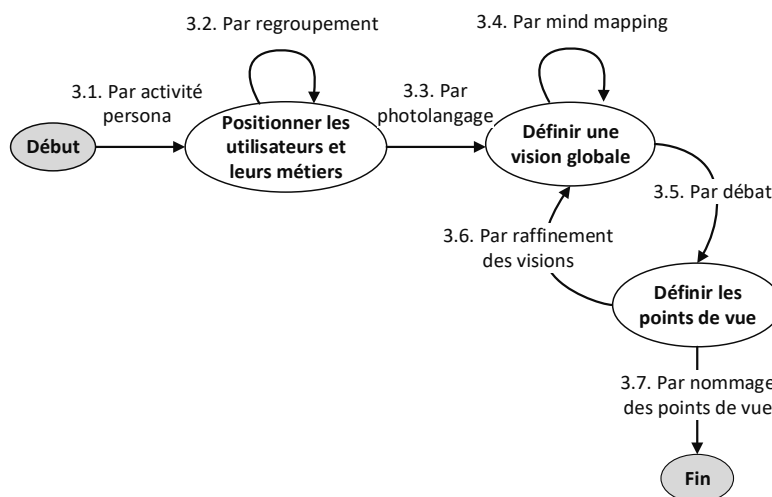


Figure 49 : Sous-map de la section 3

La sous-map que nous avons présentée doit permettre d'instancier de nouvelles classes du modèle de produit de DEMOS : **métier** et **point de vue**. Cette instanciation provisoire sera mise à jour par la suite avec la section 4. Les stratégies mises en œuvre sont détaillées dans la Table 10 et les outils utilisés tels que le photolangage feront l'objet d'une description plus précise dans la section 5.2.4.

Table 10 : Synthèse de la section 3 de DEMOS

n°	Nom de section	Description	Moyens mis en œuvre	Résultats attendus
3.1	<Début, Positionner les utilisateurs et leur métier, Par activité Persona >	Chacun des utilisateurs note son nom sur un persona et vient le positionner sur le tableau blanc. En face de son persona, il ajoute un post-it avec le nom de son métier.	Post-its Post-its persona	Liste d'utilisateurs Liste de métiers
3.2	<Positionner les utilisateurs et leur métier, Positionner les utilisateurs et leur métier, Par regroupement >	Lorsque plusieurs personas correspondent au même métier, ils sont regroupés et on ne laisse au tableau qu'un seul post-it par métier. Si pour un même métier plusieurs noms/plusieurs orthographes sont proposés, un consensus doit permettre de n'en garder qu'un.	Post-its	Liste d'utilisateurs et liste de métiers consolidés

3.3	<Positionner les utilisateurs et leurs métiers, Définir une vision globale, Par photolangage >	Chaque utilisateur participe à une activité de photolangage pour exprimer sa vision du domaine métier, les normes et valeurs qu'il lui attache. Le modérateur note les termes clés sur des post-it en face de chacun des persona.	Post-its Cartes de photolangage	Liste de visions individuelles
3.4	<Définir une vision globale, Définir une vision globale, Par mind mapping >	Les utilisateurs et le modérateur regroupent les termes clés notés afin de constituer une mind map représentant les différentes visions du domaine métier classées par métier. Cela permet d'obtenir une vision globale des utilisateurs.	Feutres et post-its	Mind map de la vision globale
3.5	<Définir une vision globale, Définir les points de vue, Par débat >	Les utilisateurs débattent afin d'identifier quels métiers peuvent être regroupés du fait d'une similarité dans les visions du domaine métier exprimées.	Feutres et post-its	Regroupement de métiers
3.6	<Définir les points de vue, Définir une vision globale, Par raffinement des visions >	Quelquefois la définition des points de vue fait émerger de nouveaux termes clés à ajouter à la mind map. Dans ce cas un nouveau post-it est créé et ajouté au tableau blanc.	Post-its	Mind map consolidée de la vision globale
3.7	<Définir les points de vue, Fin, Par nommage des points de vue >	Les métiers qui ont été regroupés autour d'une vision partagée proposent un nom au point de vue qui permettra de les identifier par la suite.	Feutres	Liste de points de vue

5.2.3.4. Section 4 : <Définir les points de vue, Définir les points de vue, Par élicitation des missions>

La section 4 est la deuxième étape de la phase de recueil des exigences, et se déroule durant l'atelier de cadrage. C'est en mettant en œuvre cette section du processus que l'on obtient une définition complète des points de vue qui structurera la suite de la méthode. Neuf stratégies peuvent être appliquées (cf. Figure 50), et permettent d'aborder une partie plus dynamique du modèle de produit. La description de ces éléments dynamiques poursuit deux objectifs : consolider la liste des points de vue et préparer les ateliers points de vue qui suivront en décrivant de manière macroscopique les processus de travail.

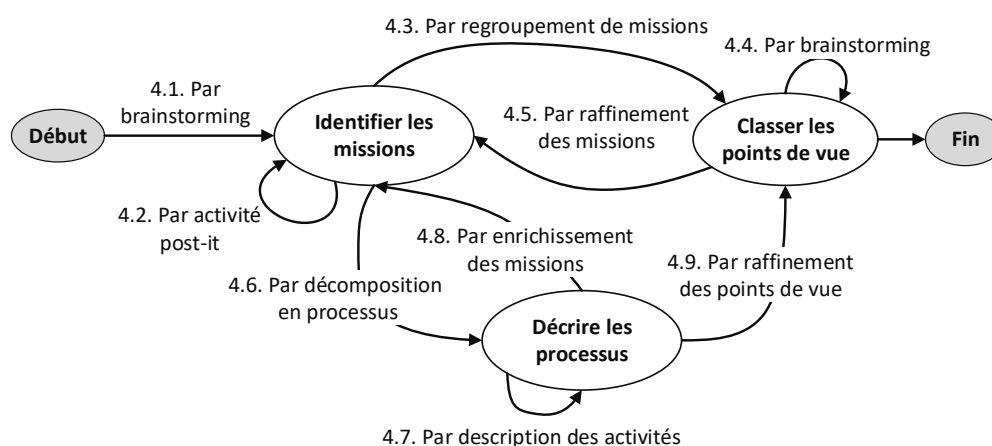


Figure 50 : Sous-map de la section 4

Cette section 4 est l’occasion d’instancier de nouvelles classes du modèle de produit : **mission**, **processus**, **activité**, **rôle** et de venir mettre à jour l’instanciation de la classe **point de vue**. Les stratégies nécessaires à cela sont présentées dans la Table 11 suivante.

Table 11 : Synthèse de la section 4 de DEMOS

n°	Nom de section	Description	Moyens mis en œuvre	Résultats attendus
4.1	<Début, Identifier les missions, Par brainstorming >	Les utilisateurs identifient les missions qu'ils accomplissent dans la cadre de la vision globale identifiée. Ils notent chacune de leurs missions sur un post-it.	Post-its	Liste de missions par personne
4.2	<Identifier les missions, Identifier les missions, Par activité post-it >	Chaque utilisateur va placer ses missions en face de son métier, si la mission n'est pas déjà positionnée sur le tableau. Les missions qui sont hors du périmètre du projet sont barrées mais laissées au tableau.	Post-its Feutres	Liste des missions
4.3	<Identifier les missions, Classer les points de vue, Par regroupement de missions >	Le modérateur de la méthode regroupe ensuite les missions de chaque participant par métier, et donc par point de vue en s'assurant que l'adéquation entre point de vue et mission est assurée.	Post-its Feutres	Liste de missions par point de vue
4.4	<Classer les points de vue, Classer les points de vue, Par brainstorming >	Les points de vue sont à nouveau discutés, et leur nom est éventuellement changé.	Feutres	Liste consolidée de points de vue
4.5	<Classer les points de vue, Identifier les missions, Par raffinement des missions >	Lors de la discussion portant sur les points de vue, des missions peuvent éventuellement être ajoutées.	Post-its	Liste de missions consolidée
4.6	< Identifier les missions, Décrire les processus, Par décomposition en processus >	Pour chaque mission, des noms de processus sont identifiés et notés sur des post-its par le modérateur.	Post-its Feutres	Liste des processus métier

4.7	<Décrire les processus, Décrire les processus, Par description des activités >	Les processus sont détaillés sous la forme de modèles BPM simplifiés. Les activités et les rôles sont notés par le modérateur qui guide les utilisateurs dans leur description.	Feutres	Description des processus avec une notation BPM simplifiée
4.8	<Décrire les processus, Identifier les missions, Par enrichissement des missions >	La description des processus peut permettre d'enrichir la liste des missions. Dans ce cas, chaque nouvelle mission est notée sur un post-it et ajoutée au tableau.	Post-its	Liste de missions consolidée
4.9	<Décrire les processus, Classer les points de vue, Par raffinement des points de vue >	La description des processus peut permettre de compléter la description des points de vue, et éventuellement de venir les modifier.	Feutres	Liste de points de vue consolidée

5.2.3.5. **Section 5 : <Définir les points de vue, Concevoir un modèle par point de vue, Par représentation conceptuelle>**

La section 5 est la première étape de la phase de conception du système d’information. Elle est mise en œuvre lors des ateliers point de vue qui sont aussi nombreux que les points de vue identifiés. Les utilisateurs sont donc regroupés par point de vue et l’objectif est de créer une représentation conceptuelle du domaine métier conforme à leur vocabulaire et à la vision qu’ils ont de ce domaine. Dix stratégies peuvent être totalement ou partiellement mises en œuvre pour atteindre les intentions décrites en Figure 51.

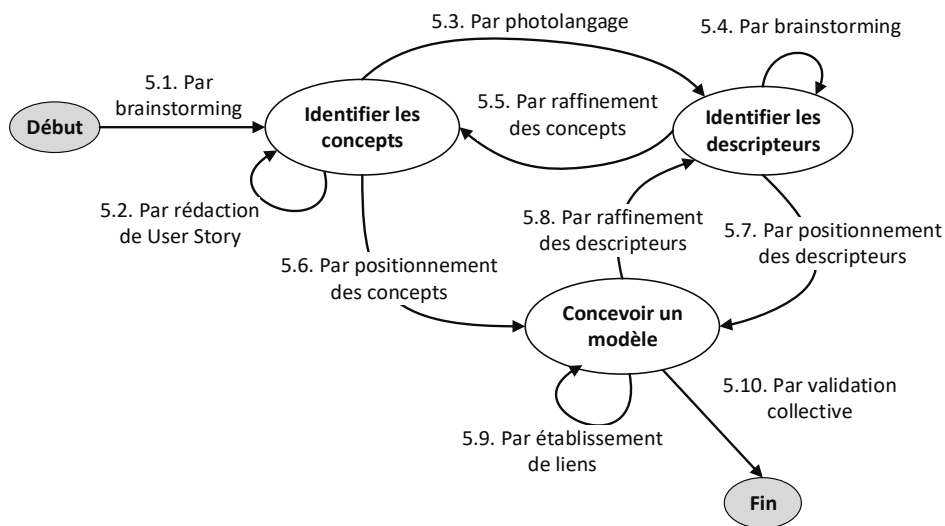


Figure 51 : Sous-map de la section 5

L’objectif de cette section est en grande partie de venir instancier la partie statique du modèle de produit de DEMOS. Ainsi, les classes **concept**, **descripteur** et **lien** seront instanciées pour chacun des points de vue, et la partie fonctionnelle sera également complétée avec l’instanciation des **User Stories**. Chacune des stratégies qui peut être mise en œuvre est présentée dans la Table 12 suivante.

Table 12 : Synthèse de la section 5 de DEMOS

n°	Nom de section	Description	Moyens mis en œuvre	Résultats attendus
5.1	<Début, Identifier les concepts, Par brainstorming >	Les utilisateurs identifient les concepts principaux qu'ils manipulent pour mener à bien leur mission. Ils utilisent un vocabulaire qui leur est propre. Ils notent chacun des concepts sur des post-its et viennent ensuite les positionner au tableau	Post-its	Liste de concepts
5.2	<Identifier les concepts, Identifier les concepts, Par rédaction de User Stories >	Les utilisateurs rédigent les User Stories correspondant à chaque activité des processus les concernant. Ils complètent ainsi la liste des concepts.	User Stories Post-its	Liste consolidée de concepts Liste de User Stories
5.3	<Identifier les concepts, Identifier les descripteurs, Par photolangage >	Les utilisateurs décrivent en quelques mots les principaux concepts qu'ils manipulent en participant à une activité de photolangage. Ainsi, les premiers descripteurs des concepts sont identifiés.	Post-its	Liste de concepts avec leurs descripteurs
5.4	<Identifier les descripteurs, Identifier les descripteurs, Par brainstorming >	Les utilisateurs ajoutent des descripteurs aux concepts, en accord avec le vocabulaire qu'ils utilisent pour la rédaction de leurs User Stories.	Post-its	Liste consolidée de concepts avec leurs descripteurs
5.5	<Identifier les descripteurs, Identifier les concepts, Par raffinement des concepts >	La définition des descripteurs peut venir modifier la liste des concepts établie. En effet, certains descripteurs deviennent des concepts, et inversement. Les noms des concepts peuvent également être modifiés.	Post-its	Liste complétée de concepts
5.6	<Identifier les concepts, Concevoir un modèle, Par positionnement des concepts >	Les utilisateurs et le modérateur de la méthode écrivent le nom de chacun des concepts sur une fiche cartonnée et la positionnent au tableau blanc.	Tableau blanc Fiches cartonnées	Modèle avec les concepts
5.7	<Identifier les descripteurs, Concevoir un modèle, Par positionnement des descripteurs >	Les utilisateurs et le modérateur de la méthode écrivent le nom de chacun des descripteurs sous le concept correspondant sur la fiche cartonnée et repositionnent la fiche au tableau blanc.	Tableau blanc Fiches cartonnées	Modèle avec les concepts et les descripteurs

5.8	<Concevoir un modèle, Concevoir un modèle, Par établissement des liens >	Les utilisateurs et le modérateur tracent les liens entre concepts en les nommant et en leur donnant une direction.	Tableau blanc Fiches cartonnées Feutres	Modèle avec les concepts, les descripteurs et les liens.
5.9	<Concevoir un modèle, Identifier les descripteurs, Par raffinement des descripteurs >	L'établissement de liens entre concepts peut amener à supprimer certains descripteurs dont le sens est contenu dans le lien lui-même.	Tableau blanc Fiches cartonnées	Modèle avec les concepts, les descripteurs et les liens consolidé
5.10	<Concevoir un modèle, Fin, Par validation collective >	Une fois le modèle terminé au tableau, les utilisateurs doivent valider la représentation obtenue à l'unanimité.		Modèle validé

5.2.3.6. **Section 6 : <Concevoir un modèle par point de vue, Définir les points de vue, Par affinement du point de vue>**

La conception des modèles conceptuels par point de vue peut venir impacter la définition antérieure des points de vue. En effet, l'étape de modélisation conceptuelle est une étape de description du vocabulaire et il peut apparaître différents cas de figure représentés dans la Figure 52. Ainsi, le point de vue peut être renommé, ou bien encore un point de vue peut être ajouté ou supprimé. Les missions, processus décrits précédemment peuvent aussi être mis à jour. Toutes ces modifications doivent en tous cas faire l'objet d'une validation collective.

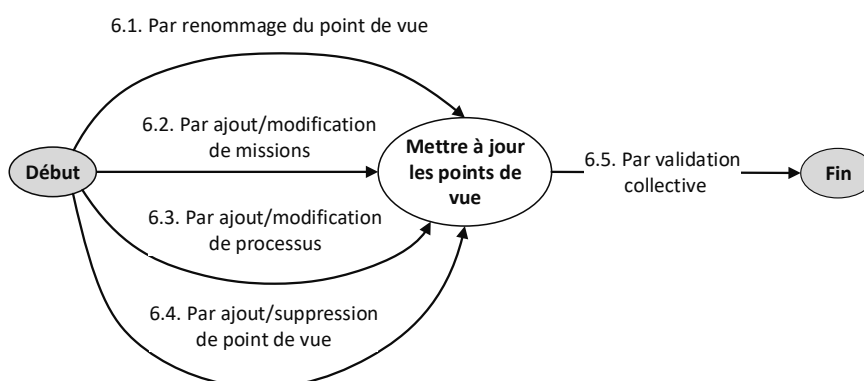


Figure 52 : Sous-map de la section 6

Cette section n'a pas pour but d'instancier de nouvelles classes du modèle de produit mais bien de venir modifier certaines instanciations précédemment faites. La Table 13 précise les différents cas de figure pouvant se présenter.

Table 13 : Synthèse de la section 6 de DEMOS

n°	Nom de section	Description
6.1	<Début, Mettre à jour les points de vue, Par renommage du point de vue >	Lors de la conception du modèle par point de vue, les utilisateurs peuvent souhaiter modifier le nom du point de vue qui les représente afin qu'il soit plus en adéquation avec le vocabulaire qu'ils ont employé.

6.2	<Début, Mettre à jour les points de vue, Par ajout/modification missions >	La description du vocabulaire dans le modèle conceptuel par point de vue peut faire apparaître de nouvelles missions des utilisateurs, ils peuvent alors les ajouter à la description de leur point de vue.
6.3	<Début, Mettre à jour les points de vue, Par ajout/modification processus >	La rédaction des User Stories et la création du modèle conceptuel peut entraîner des modifications de la représentation des processus précédemment définis, ou peut éventuellement venir enrichir la liste des processus.
6.4	<Début, Mettre à jour les points de vue, Par ajout/suppression de point de vue >	Les expressions de chacun des utilisateurs peuvent faire apparaître des dissonances entre points de vue. Si aucun consensus ne peut être opéré pour s'accorder sur un vocabulaire commun, cela peut signifier que l'ajout d'un nouveau point de vue est nécessaire. A l'inverse, le modérateur présent à chacun des ateliers point de vue peut considérer que deux points de vue s'expriment dans un vocabulaire quasi-identique et proposer alors une fusion de point de vue, conduisant ainsi à la suppression de l'un des points de vue fusionné.
6.5	<Mettre à jour les points de vue, Fin, Par validation collective >	Les utilisateurs regroupés par point de vue doivent valider collectivement et par consensus tous les changements qui auront été faits.

5.2.3.7. **Section 7 : <Concevoir un modèle par point de vue, Consolider les modèles par point de vue, Par recherche de similarités>**

La section 7 marque la deuxième étape de la phase de conception du processus DEMOS. L'étape précédente consistant à créer autant de modèles conceptuels que de points de vue, cette étape a pour but de lier ces modèles entre eux. Pour cela, un atelier de mise en commun est organisé afin de partager entre tous les utilisateurs les modèles, et d'identifier les similarités entre ces modèles. Cinq stratégies sont mises en œuvre pour atteindre cet objectif, et sont décrites dans la Figure 53 suivante.

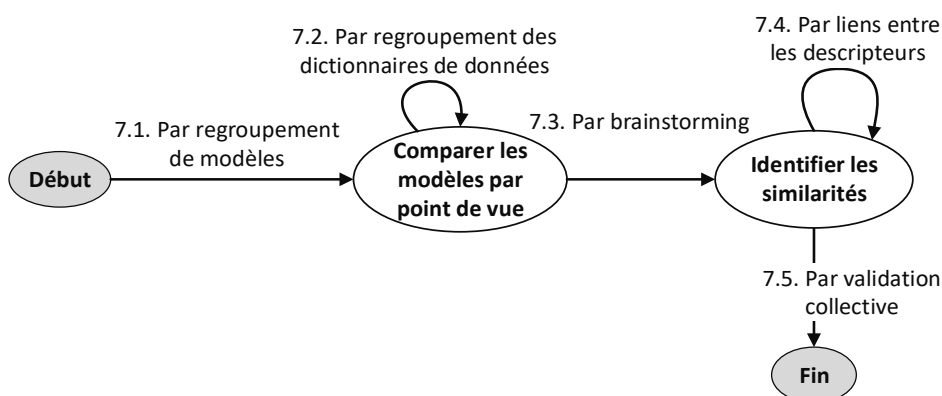


Figure 53 : Sous-map de la section 7

La section 7 permet d'instancier un nouvel élément du modèle de processus : le **lien de similarité**, et de compléter ainsi l'aspect perspectif du modèle. Cette étape nécessite des discussions et débats entre tous les utilisateurs, c'est ce que nous présentons dans la Table 14 suivante.

Table 14 : Synthèse de la section 7 de DEMOS

n°	Nom de section	Description	Moyens mis en œuvre	Résultats attendus
7.1	<Début, Comparer les modèles par point de vue, Par regroupement de modèles >	Les utilisateurs confrontent les modèles par point de vue. Ils débattent des différences entre ces modèles.	Tableau blanc avec les modèles par point de vue	
7.2	<Comparer les modèles par point de vue, Comparer les modèles par point de vue, Par regroupement des dictionnaires de données >	Les utilisateurs comparent les dictionnaires des données élaborés par le modérateur. Ils débattent des différences entre définitions.	Dictionnaires des données	
7.3	<Comparer les modèles par point de vue, Identifier les similarités, Par brainstorming >	Les utilisateurs identifient les similarités entre les modèles et entre les dictionnaires des données.	Tableau blanc avec modèles par point de vue Dictionnaires des données	
7.4	<Identifier les similarités, Identifier les similarités, Par liens entre les descripteurs >	Les utilisateurs tracent des liens entre les dictionnaires des données. Ils lient les descripteurs qui ont la même définition/le même sens.	Tableau blanc avec modèles par point de vue Dictionnaires des données feutres	Liens de similarités entre modèles et/ou dictionnaires de données
7.5	<Identifier les similarités, Fin, Par validation collective >	Une fois les modèles reliés au tableau, les utilisateurs doivent valider la représentation globale obtenue par consensus.	Tableau blanc avec modèles par point de vue Dictionnaires des données	Modèles par point de vue consolidés Dictionnaires des données consolidés

5.2.3.8. Section 8 : <Consolider les modèles par point de vue, Concevoir un modèle par point de vue, Par affinement du modèle>

La consolidation des modèles par point de vue via la recherche de similarités peut faire apparaître des éléments inopportuns dans les modèles conceptuels par point de vue. Le cas échéant, les modèles par point de vue doivent être modifiés, soit par l'ajout, soit par la modification, soit par la suppression de concepts, descripteurs et/ou liens. Les différentes stratégies applicables alors sont représentées dans la Figure 54 suivante.

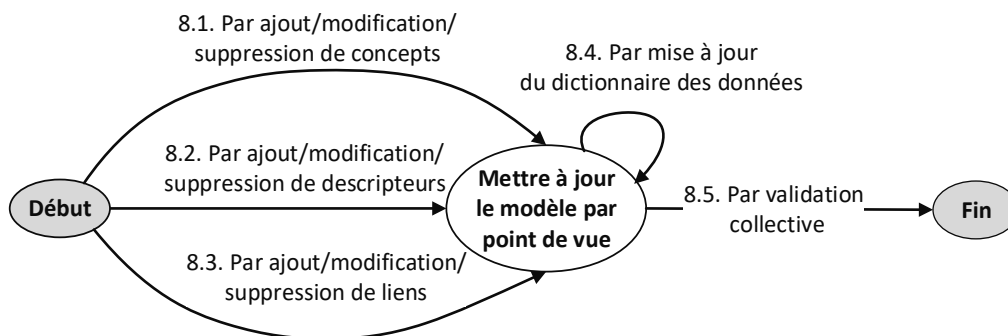


Figure 54 : Sous-map de la section 8

Dans le cadre de cette section, l’objet n’est pas d’instancier des classes du modèle de produit mais bien de venir modifier une instanciation déjà opérée. Ici, ce sont les instances des classes **concept**, **descripteur** et **lien** qui peuvent être impactées, comme présenté dans la Table 15 suivante.

Table 15 : Synthèse de la section 8 de DEMOS

n°	Nom de section	Description
8.1	<Début, Mettre à jour le modèle par point de vue, Par ajout/ modification/ suppression de concepts >	Lors de la mise en commun des modèles, certains utilisateurs regroupés par point de vue peuvent choisir de venir mettre à jour les concepts de leur modèle, en les renommant, en en ajoutant ou en en supprimant.
8.2	<Début, Mettre à jour le modèle par point de vue, Par ajout/ modification/ suppression de descripteurs >	La mise en commun des modèles peut aussi entraîner une mise à jour des descripteurs du modèle par point de vue.
8.3	<Début, Mettre à jour le modèle par point de vue, Par ajout/ modification/ suppression de liens >	Enfin certains liens peuvent être ajoutés, modifiés ou supprimés.
8.4	<Mettre à jour le modèle par point de vue, Mettre à jour le modèle par point de vue, Par mise à jour du dictionnaire des données >	Lorsque des changements sont apparus sur les modèles par point de vue, les dictionnaires des données correspondant doivent être mis à jour.
8.5	<Mettre à jour le modèle par point de vue, Fin, Par validation collective >	La modification d’un modèle par point de vue doit être validée par consensus par les utilisateurs regroupés sous ce point de vue.

5.2.3.9. Section 9 : <Consolider les modèles par point de vue, Fin, Par implémentation dans une BD graphe>

La section 9 est l’ultime étape de la méthode et correspond à la phase d’implémentation des modèles conceptuels de données. L’objectif de l’étape précédente était de fournir des modèles implémentables et la présente étape consiste donc en une succession de sept stratégies permettant de créer le référentiel de données support du système d’information (cf. Figure 55). Ces stratégies sont celles correspondant à une instanciation dans une base de données NoSQL orientée graphe.

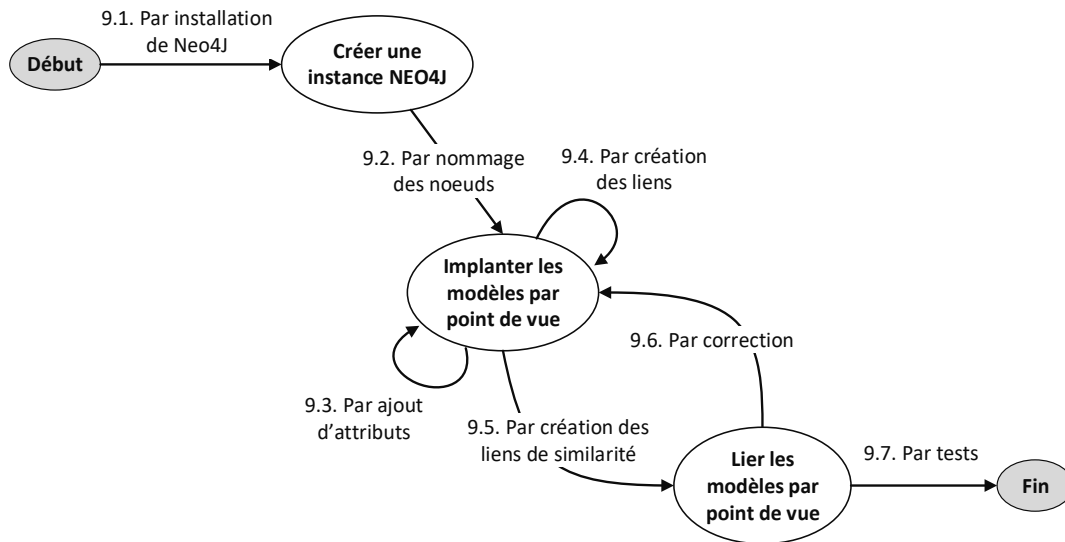


Figure 55 : Sous-map de la section 9

Cette section du modèle de processus de DEMOS n’instancie pas de nouvelle classe du modèle de produit. Elle se nourrit de l’instanciation globale du modèle de produit effectuée précédemment pour l’implanter en partie (les parties statiques et perspectives en l’occurrence). Chacune des stratégies est détaillée dans la Table 16 suivante.

Table 16 : Synthèse de la section 9 de DEMOS

n°	Nom de section	Description
9.1	<Début, Créer une instance Neo4J, Par installation de Neo4J >	Une instance de Neo4J doit être installée sur un serveur afin de pouvoir avoir accès à la base de données
9.2	<Créer une instance Neo4J, Implanter les modèles par point de vue, Par nommage des nœuds >	Les nœuds doivent être créés. Chaque nœud correspond à un concept.
9.3	<Implanter les modèles par point de vue, Implanter les modèles par point de vue, Par ajout d'attributs >	Sur chaque nœud les descripteurs sont ajoutés comme une liste d'attributs.
9.4	<Implanter les modèles par point de vue, Implanter les modèles par point de vue, Par création de liens >	Des arcs sont tracés entre chacun des nœuds. Ils correspondent aux liens entre concepts d'un même modèle. Ils sont nommés.
9.5	<Implanter les modèles par point de vue, Lier les modèles par point de vue, Fin, Par création de liens de similarité >	Des arcs sont établis entre les nœuds de chaque modèle et sont nommés "similaire à". Ils sont la représentation des liens de similarité.
9.6	<Lier les modèles par point de vue, Implanter les modèles par point de vue, Par correction >	L'établissement de liens peut faire apparaître un certain nombre d'erreurs.
9.7	<Lier les modèles par point de vue, Fin, Par tests >	La base de données est testée.

Nous avons choisi de proposer cette première forme d’implantation car elle est celle qui correspond le mieux à notre besoin : la prise en compte des points de vue multiples. L’un des principes de base qui

régit le fonctionnement des bases de données relationnelles (ou relationnelles objet) est le suivant : « *tout objet est instance d'une classe et une seule* » (Carré 1989). Ce principe a été remis en cause il y a 25 ans par Dominique Rieu lorsqu'elle propose : « *Une instance peut appartenir à plusieurs classes simultanément [...] une instance n'est plus forcément un moulage parfait des entités conceptuelles qui la décrivent* ». (Rieu, Nguyen, et Culet 1991). La voie de l'instanciation multiple, ouverte dans les années 90 nous offre aujourd'hui une solution pour la représentation des points de vue.

L'instanciation multiple peut être définie comme le rattachement d'un objet à plusieurs classes (Riet 1989). Avec DEMOS, ce cas peut se présenter de manière très fréquente. En effet, lorsque deux points de vue créent leur modèle conceptuel, et les lient ensuite par des liens de similarité, cela implique que dans la réalité, un même objet pourra être l'instance de deux classes (cf. Figure 56).

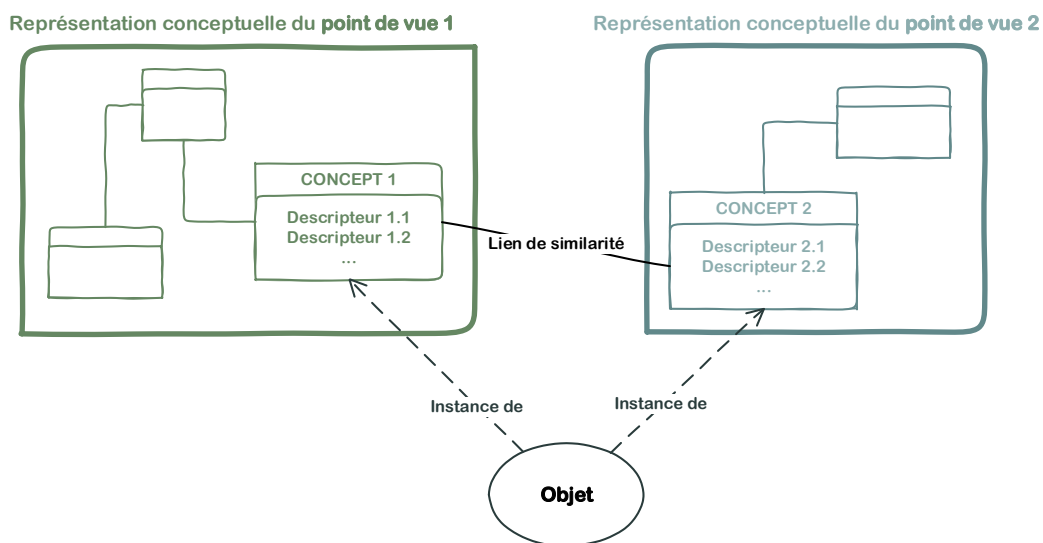


Figure 56 : Instanciation multiple avec DEMOS

Dans les années 90, aucune solution technique ne permettait de répondre à ce principe de multi-instanciation. Ainsi, les solutions permettant de représenter la catégorisation d'objet dans deux classes différentes étaient des adaptations du principe de spécialisation multiple (cf. Figure 57). Pour cela, une sous-classe pouvait être créée et implantée dans une base de données relationnelle ou objet.

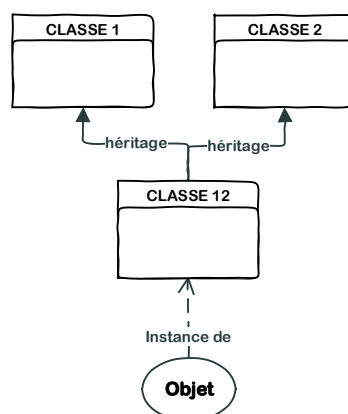


Figure 57 : Principe de spécialisation multiple

Ce type de solution présente selon nous deux désavantages :

- D'une part cette sous-classe ne fait partie de la représentation conceptuelle de personne. Notre objectif avec DEMOS est de permettre aux utilisateurs de créer leur propre

représentation conceptuelle, sans laisser cette tâche à des experts techniques. Or dans ce cas de spécialisation multiple, la création de sous-classes représente un choix de concepteur qui n'est pas en accord avec l'expression des utilisateurs.

- D'autre part, la sous-classe vient surcharger cette représentation en y apportant de la « *complexité hiérarchique* » (Bihanic et al. 2013). Or il nous semble indispensable que la représentation conceptuelle soit compréhensible, appréhendable par les utilisateurs. Pour cela, la plus grande simplicité est recherchée dans la représentation conceptuelle, et cette solution ne répond donc pas à ce besoin.

L'instanciation multiple nous semble être la solution à adopter dans notre cas. Elle répond aux problématiques que nous venons d'exposer en permettant « *d'éviter la prolifération des sous-classes de méthodes due à la spécialisation multiple.* » (Escamilla 1993), mais également de préserver « *l'identité de l'objet à ses différents points de vue* » (Rieu, Nguyen, et Culet 1991).

Aujourd'hui, les bases de données NoSQL orientées graphe nous offrent la possibilité d'implanter ce principe de multi-instanciation, grâce à leur souplesse notamment. En effet, la liberté permise dans le typage des liens est une opportunité pour nous de faire une distinction entre les trois types de liens nécessaires à la gestion de la multi-instanciation : les liens d'association, les liens de similarité et les liens d'instanciation. Un exemple permettant d'illustrer cela nous est fourni par les résultats de l'expérimentation de notre méthode présentée dans le chapitre suivant.

Cependant, cette stratégie d'implantation n'est certainement pas unique. Conformément à nos principes d'adaptabilité, il est tout à fait envisageable de proposer d'autres stratégies d'implémentation au fil du temps si des choix techniques différents devaient se présenter.

5.2.4. Décomposition du processus d'amélioration continue

Le processus d'amélioration continue de la méthode est déroulée aussi souvent que nécessaire au cours du cycle de vie du système d'information. En effet, afin de respecter le principe d'évolutivité des points de vue nécessaire à la démocratie, il nous a semblé indispensable de proposer aux utilisateurs des solutions d'adaptation du système. Ce processus reprenant certains éléments du processus de construction, il ne nous a pas semblé opportun de le décrire de façon aussi approfondie. Cependant, nous allons en détailler chacune des sections de manière précise, afin de comprendre comment il peut être effectivement mis en œuvre.

5.2.4.1. Section 10 : <Début, Modifier un modèle par point de vue, Par mise à jour d'une représentation conceptuelle>

Cette première section du processus d'amélioration continue est en quelque sorte l'opportunité de démarrer le processus. Il s'agit d'un besoin de modification du modèle par point de vue d'un ou plusieurs utilisateurs (cf. Figure 58).

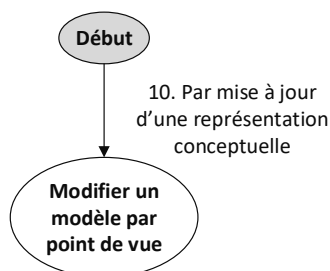


Figure 58 : Zoom sur la section 10

La mise à jour d'une représentation conceptuelle est une étape qui peut être enclenchée à n'importe quel moment du cycle de vie du système. Il s'agit dans tous les cas de répondre à un besoin exprimé par un ou des utilisateurs, et qui émerge de son usage du système. Cela s'explique par une inadéquation entre le modèle et « sa » réalité, qui peut se traduire par un ou des éléments du modèle manquant ou inexact. Cela peut engendrer des difficultés dans l'utilisation du système, que l'utilisateur doit « tordre » pour qu'il réponde à son besoin, ou qu'il doit tout simplement compléter par un autre outil. Ce changement peut avoir différentes causes : oubli d'éléments du modèle lors de la construction du système, évolution du point de vue des utilisateurs, changements dans les procédures de travail, etc. Dans le cas le moins formel, l'utilisateur peut faire part de ses besoins de modification aux autres utilisateurs partageant le même point de vue, et ils se mettent d'accord sur un besoin commun. Dans le cas le plus formalisé (et le plus souhaitable), les utilisateurs d'un même point de vue se regroupent pour un atelier « point de vue » et redéterminent ensemble les modifications à apporter à leur représentation conceptuelle, avec ou sans l'aide du modérateur.

A l'issue de la mise en œuvre de cette section, le modèle par point de vue de l'un des groupes utilisateurs est modifié, et cela peut donner lieu à deux étapes différentes pour la suite : un nouvel atelier de cadrage pour redéfinir plus en profondeur le point de vue (section 11), ou un atelier de mise en commun pour mettre en cohérence le nouveau modèle conceptuel avec les modèles des autres points de vue (section 13).

5.2.4.2. **Section 11 : <Modifier un modèle par point de vue, Redéfinir les points de vue, Par affinement du point de vue>**

La deuxième section de cette MAP du processus d'amélioration continue fait suite à une modification de la représentation conceptuelle de l'un des points de vue. Elle consiste en une redéfinition des points de vue par une stratégie d'affinement (cf. Figure 59).

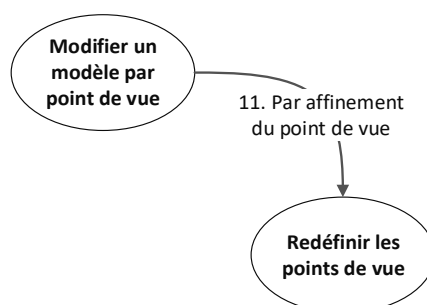


Figure 59 : Zoom sur la section 11

Lorsque la mise à jour d'une représentation conceptuelle donne lieu à de nouveaux débats sur le langage : les concepts, les descripteurs, les liens, cela entraîne parfois des débats de plus « haut niveau » sur la vision partagée au sein du point de vue et peut nécessiter de remettre en place un atelier de cadrage. Ainsi, les utilisateurs peuvent à nouveau discuter des valeurs qu'ils partagent, des missions qu'ils ont en commun et de leurs processus de travail afin de redéfinir éventuellement les points de vue. Ils peuvent alors ajouter ou supprimer un ou des points de vue afin qu'ils soient plus en accord avec leur vision du domaine métier couvert par le système.

5.2.4.3. **Section 12 : <Redéfinir les points de vue, Modifier un modèle par point de vue, Par représentation conceptuelle>**

La section 12 fait suite à un atelier de cadrage durant lequel les points de vue auront été redéfinis. Il s'agit alors de proposer une nouvelle représentation conceptuelle par point de vue (cf. Figure 60).

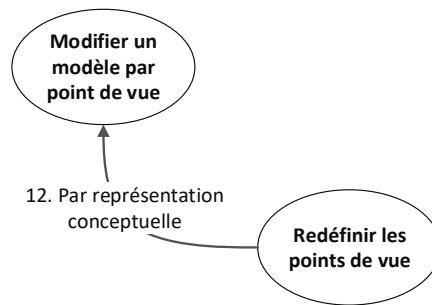


Figure 60 : Zoom sur la section 12

Lorsqu'un nouveau point de vue a été défini, les utilisateurs partageant ce point de vue doivent établir une nouvelle représentation conceptuelle, ou adapter une représentation conceptuelle existante. Pour cela, ils participent à un nouvel atelier point de vue, durant lequel ils mettront en place les stratégies proposées dans la section 5 du processus de construction de DEMOS. Ainsi, ils utiliseront des outils tels que le brainstorming, le photolangage ou encore la rédaction de User Stories afin d'instancier les classes concept, descripteur et lien du modèle de produit.

5.2.4.4. Section 13 : <Modifier un modèle par point de vue, Consolider les modèles par point de vue, Par mise à jour de similarités>

La section 13 fait suite à la modification d'un modèle par point de vue, ou à la création d'un nouveau modèle. Elle permet de s'assurer de la cohérence des points de vue du système par mise à jour des liens de similarités (cf. Figure 61).

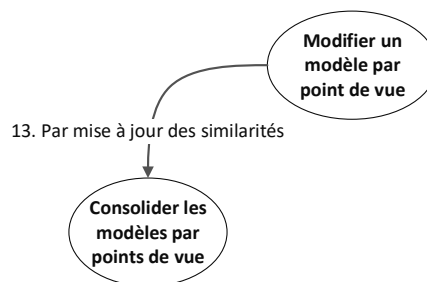


Figure 61 : Zoom sur la section 13

Lorsque de nouvelles propositions sont faites quant aux représentations conceptuelles par point de vue, il est indispensable de s'assurer que ces mises à jour sont cohérentes. Pour cela, tous les utilisateurs doivent être regroupés lors d'un atelier de mise en commun afin de débattre de leurs modèles, et d'identifier les similarités qui permettent de les relier. Ce sont alors les stratégies de la section 7 du modèle de processus de construction de DEMOS qui sont mises en œuvre. A l'issue de l'atelier, des modèles consolidés sont proposés, grâce à la modification des liens de similarité du modèle de produit.

5.2.4.5. Section 14 : <Consolider les modèles par point de vue, Modifier un modèle par point de vue, Par affinement du modèle>

La section 14 fait suite à un atelier de mise en commun durant lequel les modèles par points de vue auront été affinés (cf. Figure 62).

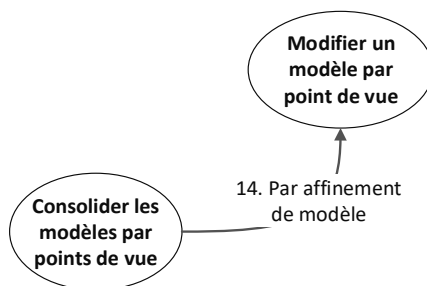


Figure 62 : Zoom sur la section 14

Cette section est identique à la section 8 du processus de construction de DEMOS. Il s’agit ici de venir modifier des représentations conceptuelle par point de vue.

5.2.4.6. Section 15 : <Consolider les modèles par point de vue, Redéfinir les points de vue, Par consolidation de modèle>

La consolidation des modèles peut donner lieu à de nouveaux débats et discussions sur la définition même des points de vue dans le cadre de cette boucle d’amélioration continue. C’est alors la section 15 du modèle de processus qui est mise en œuvre (cf. Figure 63).

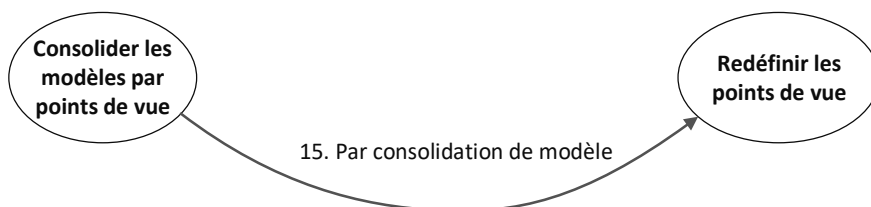


Figure 63 : Zoom sur la section 15

La boucle d’amélioration continue suit le chemin « inverse » de la boucle de construction. En effet, l’élément déclencheur est l’apparition d’un besoin dans la réalité qui entraîne une ascension jusqu’au niveau le plus abstrait de la méthode : la définition des points de vue. Ainsi, lorsqu’un groupe d’utilisateurs apporte des modifications à sa représentation conceptuelle et que celle-ci doit alors être mise en cohérence avec les autres modèles, tous les points de vue peuvent être touchés par les mises à jour. Les débats et discussions de l’atelier de mise en commun peuvent alors montrer la nécessité d’une mise à jour de l’ensemble des points de vue. Dans ce cas, un nouvel atelier de cadrage peut être proposé.

5.2.4.7. Section 16 : <Consolider les modèles par point de vue, Fin, Par mise à jour de la BD>

Suite à la consolidation des modèles par point de vue, une mise à jour de la base de données est nécessaire. La Figure 64 nous présente cette dernière étape de la boucle d’amélioration continue.

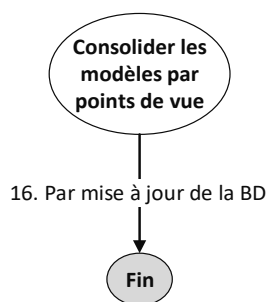


Figure 64 : Zoom sur la section 16

Cette mise à jour de la base de données consiste en fait en une mise à jour des stratégies de la section 9 du modèle de processus de construction. Ainsi, les nœuds, les liens et les liens de similarité seront modifiés afin de s’assurer en continu d’une traçabilité des points de vue dans le système. Pour le moment, la stratégie que nous proposons porte donc sur une mise à jour de base de données NoSQL orientée graphe.

5.2.5. Les outils de la méthode

Comme nous l’avons expliqué dans les descriptions détaillées des sections des modèles de processus de DEMOS, de nombreux outils sont utilisés. Ils permettent de faire participer tous les utilisateurs, de rythmer les ateliers de travail, et de dynamiser les réflexions avec des moyens ludiques. Les deux principaux outils que nous utilisons sont le jeu de photolangage et le tableau blanc sous une forme « Metaplan », c’est ce que nous décrivons à présent.

5.2.5.1. Le photolangage

Le photolangage est une technique consistant à proposer aux participants une centaine de photos, éparpillées sur une table, et à leur poser une question à laquelle ils répondent par le choix d’une photo (cf. Figure 65). Lorsque la question est posée, les participants prennent le temps qu’ils souhaitent pour choisir une photo en silence et ils la prennent dans leurs mains. Ensuite, chacun répondra à tour de rôle à la question posée en expliquant le choix de la photo.

Photolangage est avant tout un outil qui facilite la prise de parole. En effet, dans le contexte d’un atelier participatif, chacun des participants n’a pas le même rapport à l’expression en public, et certains doivent être aidés. L’objectif est de faire tourner la parole et de permettre à chacun de s’exprimer avec la même importance accordée à chaque discours. Le choix des photos permet, d’une part de personnaliser un point de vue individuel et d’autre part de relativiser sa propre position par rapport à celle d’un autre. On sort des discours généralisant pour aller vers des discours plus intéressants (Demorgon 1987).

La conceptualisation ou la description abstraite (d’une stratégie, d’une vision) est elle aussi facilitée par l’usage du photolangage. En effet, il est parfois complexe d’exprimer un avis sur des sujets dont on parle rarement : manque d’idées, peur de se tromper, etc. Le choix d’une photo permet dans un premier temps de faire naître une inspiration pour le participant, mais également de reconnaître le caractère partiel de ce que l’on exprime : on s’exprime selon une photo, donc un point de vue et une subjectivité. L’objectif n’est pas de faire « le tour du sujet » mais bien d’apporter sa contribution.



Figure 65 : Activité de photolangage lors d'un atelier DEMOS

L'activité de photolangage recouvre plusieurs intérêts dans le cadre des ateliers de conception de DEMOS, c'est pourquoi nous l'avons choisie. D'une part, il a un rôle de brise-glace lors des ateliers participatifs. Loin d'être anecdotique, cette fonction que remplit le jeu de photolangage est primordiale. Les participants de la méthode sont en effet parfois des inconnus les uns pour les autres, et l'objectif de la méthode est en partie de constituer un groupe où tous les points de vue sont représentés, mais où l'écoute et le partage sont essentiels. L'activité de photolangage permet d'introduire cette idée là sans passer par un discours introductif pesant. D'autre part et comme nous l'avons évoqué précédemment, le photolangage est un soutien pour les réflexions pouvant sembler abstraites telles que la définition d'une vision d'un domaine métier. Elle apporte de la subjectivité aux propos, qui est essentielle notamment pour l'identification de points de vue divergents.

5.2.5.2. Le tableau blanc ou Metaplan

Metaplan est une technique d'animation de réunion développée dans les années 70. Elle permet de structurer une réunion, pour laquelle l'objectif est que chacun puisse s'exprimer, donner un avis, exposer éventuellement des points de vue divergents (Beywl et Potter 1998). Le tout est noté sur un grand paper board, qui sera le seul support (cf. Figure 66).

La technique d'animation de réunion Metaplan, ou « Moderation method » peut être comprise selon deux perspectives. Dans un premier temps une perspective politique, au sens où elle relève de l'exercice du pouvoir dans un contexte organisé. En effet, la distribution de la parole lors d'ateliers Metaplan se fait de manière égale entre les participants. De plus, toutes les paroles étant considérées comme bonnes à donner et à entendre, elles sont notées sans distinction au tableau. Ainsi, la hiérarchie, l'appartenance à des services différents n'apparaît pas en fin d'atelier sur le tableau Metaplan. « *Si tout va bien, ce qui reste comme résultat d'une réflexion commune est la représentation des idées de tous les participants sur un grand panneau, qui reflète également le consensus et les divergences qui ont surgi* » (Le Rider et Levy-Hillerich 2004).

Dans un second temps, on peut appréhender Metaplan comme ayant une perspective créative. Lors d'ateliers, « *toutes les ressources intellectuelles disponibles sont utilisées pour inventer des solutions* » (Lauche 2002). Les discussions, les réflexions sont notées et on mobilise l'intelligence collective à la résolution de problèmes ou à l'émergence de solutions d'amélioration. L'idée est aussi que le formalisme de Metaplan ne prenne pas le dessus et ne vienne pas contraindre la créativité, c'est pourquoi il est relativement simple et laisse les participants s'exprimer dans un langage naturel.



Figure 66 : L'usage du tableau blanc dans les ateliers de DEMOS

Sans appliquer au sens strict la méthode METAPLAN et donc faire usage de tous les artefacts que celle-ci propose, nous nous sommes très largement inspirés de cette technique d'animation pour la construction de nos ateliers. Nous avons notamment fait le choix d'afficher au tableau tous les résultats de l'atelier. Cela nous permet dans un premier temps de garder une trace visuelle de ce qui se dit en le projetant aux yeux de tous, sans apartés, mais également de permettre de valider chacun des travaux faits en atelier en nous appuyant uniquement sur le tableau blanc. Ainsi, les comptes-rendus issus des ateliers sont des reproductions des tableaux obtenus.

5.3. La plateforme dédiée

La plateforme support à la méthode DEMOS¹¹ a pour but de recueillir l'instanciation du modèle de produit lors de la mise en œuvre de la méthode dans le cadre d'un projet de construction de système d'information. Outre la possibilité de formaliser les produits de la méthode, la plateforme permet d'assurer une traçabilité de chacun des éléments lorsque les ateliers ont lieu. La plateforme est utilisée par le modérateur de la méthode qui, à l'issue de chacune des phases, peut venir la renseigner.

L'objectif de cette plateforme est d'offrir une véritable interface graphique au(x) modérateur(s) afin qu'ils puissent répertorier les instances de modèles de produit de chacun des projets dans lesquels ils interviennent. Aujourd'hui, la plateforme est opérationnelle mais offre une interface graphique limitée ne permettant pas de réaliser un véritable modèle conceptuel graphiquement par exemple. Il s'agit pour nous d'une piste d'amélioration à creuser prochainement.

La plateforme est structurée autour du modèle de produit et du modèle de processus, c'est ce que nous détaillons dans un premier temps. Nous présentons ensuite les écrans principaux de la plateforme afin d'en décrire les fonctionnalités majeures.

5.3.1. Structure fonctionnelle de la plateforme DEMOS

La plateforme DEMOS propose au modérateur de la méthode de venir instancier le modèle de produit à l'issue des trois grands ateliers qui séquent le modèle de processus de DEMOS. Ainsi, le modérateur a accès, dès l'écran d'accueil, aux trois ateliers : atelier de cadrage, atelier point de vue et atelier de mise en commun. La structure fonctionnelle de la plateforme est présentée en Figure 67.

¹¹ La plateforme support à la méthode DEMOS a été développé en 2019 dans le cadre d'un projet du Master 1 MIAGE Parcours Ingénierie Métier

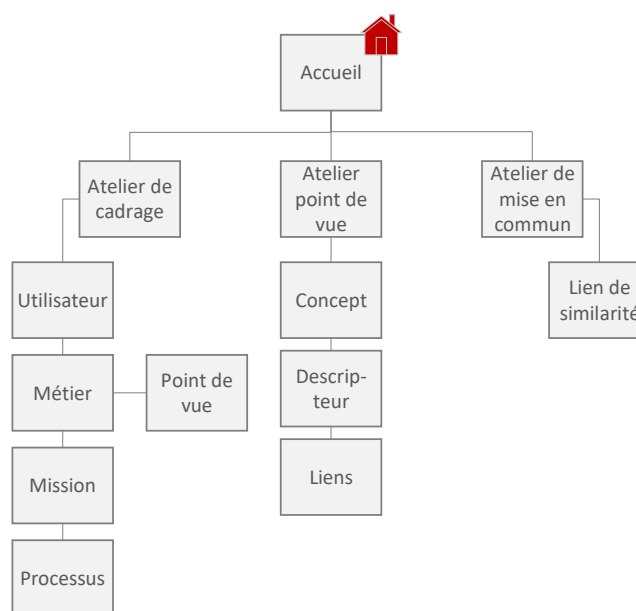


Figure 67 : Structure fonctionnelle de la plateforme DEMOS

A partir de l'accueil, le modérateur accède donc à l'atelier concerné afin de venir renseigner chacun des éléments du modèle de produit. Etant donné le temps limité que nous avons pour réaliser ce premier prototype de plateforme, nous n'avons pas été en capacité de proposer une instanciation des processus métier. Ayant dû faire un choix, nous avons préféré faire l'impasse sur les éléments du modèle de produit liés au processus car il nous semblait que cela ne représenterait pas un frein à la création du modèle conceptuel par point de vue qui constitue l'objectif premier. C'est pour cette raison que les parties Processus, Activité, Rôle et User Story ne sont pas incluses dans la plateforme. Elles le seront bien entendu dans une prochaine version.

5.3.2. Usage de la plateforme à la suite de l'atelier de cadrage

Suite à l'atelier de cadrage, le modérateur peut venir saisir dans un premier temps les noms des utilisateurs participant à l'atelier. Les métiers peuvent ensuite être créés, et affectés aux utilisateurs. Dans un second temps, il peut saisir les différentes missions notées durant l'atelier, et les affecter à un métier. Enfin, les points de vue peuvent être créés, et on leur affectera une couleur qui sera automatiquement associée aux utilisateurs de ce point de vue (cf. Figure 68).

Figure 68 : Création des points de vue sur la plateforme

La liste des utilisateurs avec leurs métiers, missions et point de vue est ensuite disponible sur la plateforme en consultation. Chacun des éléments peut être modifié ou supprimé.

5.3.3. Usage de la plateforme à la suite des ateliers point de vue

L'atelier point de vue est celui durant lequel les modèles conceptuels par point de vue sont créés. Par conséquent, l'entrée vers l'atelier sur la plateforme est conditionnée au choix d'un point de vue (cf. Figure 69).

Figure 69 : Choix du point de vue sur la plateforme

Pour le moment, la plateforme permet de répertorier l'ensemble des concepts, descripteurs et liens du modèle par point de vue. Pour cela, l'utilisateur saisit dans un premier temps les concepts, puis leur attribue des descripteurs qu'il peut typer (cf. Figure 70). Il peut ensuite lier les concepts entre eux afin de finaliser cette étape.

Figure 70 : Création des concepts et descripteurs sur la plateforme

5.3.4. Usage de la plateforme à la suite de l'atelier de mise en commun

Suite à l'atelier de mise en commun, il reste au modérateur à renseigner les liens de similarité entre les descripteurs des modèles par point de vue. Sur la plateforme, il doit alors sélectionner deux points de vue, puis un descripteur de chaque point de vue et indiquer leur similarité (cf. Figure 71).

Lien de similarité			
		Formation	Gestion
Lien 1	➤	NomE	↔ NomE
Lien 2	➤	Nom	↔ Prénom
Lien 3	➤	Type cours	↔ NomD
Lien 4	➤	Date	↔ Prénom
Lien 5	➤	Nom	↔ NomD
Lien 6	➤	Date	↔ NomD

[Revenir à la page de consultation](#)

Figure 71 : Création des liens de similarité sur la plateforme

Après ces trois ateliers, la plateforme aura permis de venir instancier le modèle de produit de DEMOS que nous avons construit. Dans l'avenir, cette plateforme devra être étendue afin, dans un premier temps de prendre en compte l'ensemble des éléments du modèle de produit (processus compris), mais également de proposer des visualisations graphiques plus proches des représentations issues des ateliers. Aujourd'hui, on peut dire que la plateforme propose de venir instancier la syntaxe abstraite de notre méthode, mais n'intègre pas encore les principes de notre syntaxe concrète.

Lorsque ces éléments complémentaires auront été proposés, les utilisateurs eux-mêmes pourront utiliser la plateforme afin de venir consulter les résultats des ateliers, en lieu et place des comptes-rendus proposés aujourd'hui par le modérateur. L'ambition est qu'à l'avenir, cette plateforme puisse permettre aux utilisateurs de suggérer des modifications, visibles et validables par les autres.

Enfin, notre dernière ambition est que cette plateforme permette une implémentation automatisée du modèle de données grâce à un lien vers une BD graphe NoSQL (type Neo4J). L'ensemble des stratégies de notre modèle de processus serait alors couvert par la plateforme support.

5.4. Conclusion du chapitre

Nous avons présenté dans ce chapitre la méthode DEMOS dans son ensemble. Les principes de démocratie que nous avons fixés au précédent chapitre ont ainsi pu être opérationnalisés au travers d'un modèle de produit, d'un modèle de processus et d'une plateforme dédiée.

Pour décrire chacune de ces composantes, nous nous sommes attachés à les détailler au maximum afin que cette méthode puisse être placée entre les mains de différents modérateurs. Elle a en effet vocation à être appliquée, puis adaptée pour répondre au mieux aux besoins de l'organisation pour laquelle elle est mise en œuvre.

Pour le moment, nous avons pu expérimenter l'instanciation de cette méthode lors d'un premier projet, et une seconde expérimentation est actuellement en préparation. C'est ce que nous allons maintenant décrire dans le chapitre 6.

Chapitre 6

EXPERIMENTATIONS ET EVALUATIONS DE LA METHODE

DEMOS est une méthode qui se veut opérationnelle. Elle a été expérimentée en 2018, dans le cadre d'un projet du service de la Formation Continue de l'Université Toulouse 1 Capitole. La seconde expérimentation est actuellement en préparation et aura lieu courant 2020 à l'Université Laval au Québec, pour un projet de conception d'une plateforme d'enseignement en tandem.

6.1. Première expérimentation : outil pour un service de Formation Continue

En 2018, le service de la Formation Continue de l'Université Toulouse 1 Capitole (FCV2A) a souhaité se doter d'un outil de gestion des présences des étudiants. Ce projet nécessitait la réalisation d'un logiciel adapté aux besoins particuliers de ce service, et qui impliquait de nombreuses parties prenantes aux points de vue divers. Nous détaillons ce contexte dans une première partie. L'expérimentation nous a permis d'instancier le modèle de produit en appliquant une grande partie du modèle de processus de la méthode que nous avons présenté dans le chapitre précédent. Le déroulement complet de l'expérimentation est détaillé dans la section 6.1.2. de ce chapitre. L'expérimentation a ensuite pu donner lieu à une évaluation par les participants, et nous complétons celle-ci par notre propre analyse dans un dernier temps.

6.1.1. Contexte de l'expérimentation et besoins exprimés

C'est dans le cadre du projet de développement d'un outil de gestion des présences pour la FCV2A que nous avons effectué une première expérimentation de la méthode DEMOS. Lorsque le service de Formation Continue a fait part du besoin d'un tel outil, la DSI de l'Université a proposé un module « sur étagère » à ce service, qui ne pouvait leur donner satisfaction pour deux raisons :

- D'une part le module proposé était mal adapté aux besoins d'un service aux problématiques particulières. En effet, la FCV2A gère le cas d'étudiants salariés, stagiaires de la Formation Continue, et dont le suivi des présences est l'un des éléments conditionnant la rémunération.
- D'autre part la mise en place de cet outil impliquait de nombreuses parties prenantes aux problématiques différentes : étudiants, gestionnaires de scolarité de la Formation Continue, enseignants de la Formation Continue (mais aussi enseignants des étudiants de Formation Initiale), organismes financeurs.

Pour ces deux raisons notamment, il nous a semblé opportun de mener une expérimentation. Le but était de conduire une démarche participative de conception afin de fournir au développeur (intégré au service de Formation Continue) des modèles implémentables pour la création de l’outil. Il nous a semblé que DEMOS pouvait, au travers de la démarche participative que la méthode propose, être une possibilité de faire travailler ensemble tous ces acteurs tout en respectant leurs points de vue éventuellement divergents.

6.1.2. Déroulement de l’expérimentation

L’expérimentation a eu lieu entre février et juin 2018. L’implémentation de la solution a ensuite été effectuée après un délai d’un an en juin 2019. Le prototype produit n’a pour le moment pas été testé par les utilisateurs eux-mêmes, mais l’ensemble du modèle de processus de construction de DEMOS a pu être instancié.

Parmi les neuf stratégies que propose le modèle de processus de construction de DEMOS, huit stratégies ont effectivement été mises en œuvre, elles sont représentées dans la Figure 72 qui suit¹².

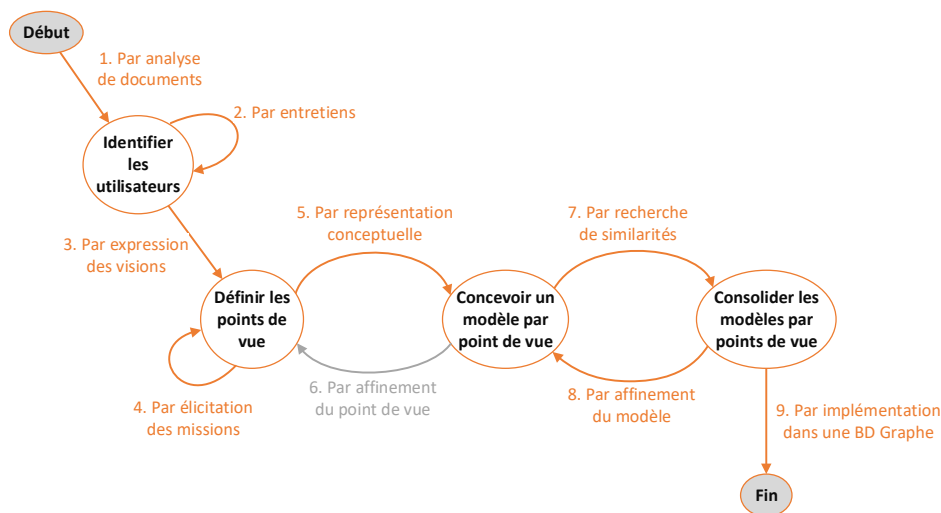


Figure 72 : Chemin dans la MAP pour l’expérimentation FCV2A

Respectant le séquençement de la méthode que nous avons présenté en section 5.2.2, une phase préparatoire nous a permis de mettre en œuvre les deux premières stratégies de la méthode. Les stratégies 3 et 4 ont ensuite été déroulées dans le cadre d’un atelier de cadrage impliquant un maximum de futurs utilisateurs de l’outil. Les ateliers point de vue nous ont permis d’instancier la cinquième stratégie du modèle de processus de DEMOS, tandis que les stratégies 7 et 8 ont été mises en œuvre dans la cadre d’un atelier de mise en commun. La stratégie 9 a été appliquée plus récemment, lors du développement du prototype de l’outil de gestion des présences.

6.1.2.1. Phase préparatoire

La phase préparatoire du projet s’est déroulée en février 2018. L’objectif était alors d’identifier l’ensemble des utilisateurs à impliquer pour l’expérimentation, mais également de délimiter clairement le domaine métier qui serait supporté par l’outil.

¹² Pour les figures qui suivent, le principe du chemin sélectionné dans la MAP sera toujours mis en avant avec la couleur orange. Ici, les stratégies en orange sont celles que l’on a effectivement mises en œuvre, celles en gris sont celles qui n’ont pas été appliquées.

La première section de la méthode DEMOS propose de dresser une première liste d'utilisateurs par lecture des documents de l'organisation (cf. Figure 73). Nous avons à notre disposition le cahier des charges rédigé depuis quelques mois, et c'est donc ce document qui nous a servi de support dans un premier temps. Les contacts des participants nous ont ensuite été fournis par le développeur du service de Formation Continue qui avait une bonne connaissance du projet.

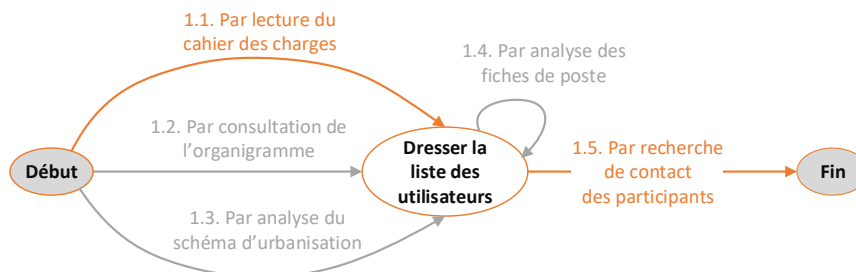


Figure 73 : Section 1 pour l'expérimentation FCV2A

A l'issue de cette section, nous avons identifié 18 acteurs parmi lesquels des enseignants ainsi que des gestionnaires de scolarité. Nous avons créé un fichier avec leurs contacts.

Nous avons ensuite mis en œuvre la deuxième section du modèle de processus (cf. Figure 74) en prenant rendez-vous avec un gestionnaire de scolarité (chef de service) et le développeur de la Formation Continue (le client dans notre cas).

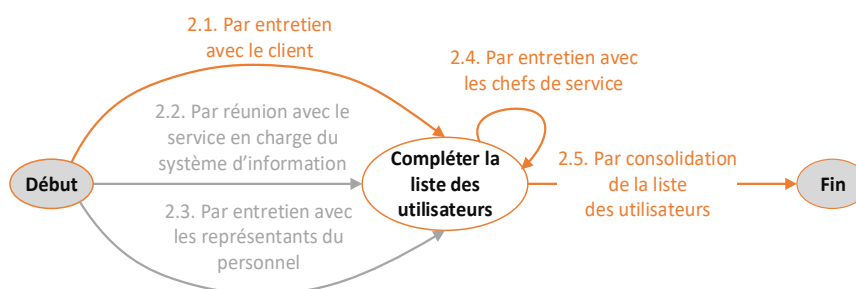


Figure 74 : Section 2 pour l'expérimentation FCV2A

Ces entretiens nous ont permis d'affiner la liste des utilisateurs en délimitant de manière plus claire le domaine métier concerné par le projet. Ainsi, nous avons décidé de concevoir dans un premier temps un prototype de l'outil, qui serait à tester par les enseignants d'un UFR particulier : l'UFR d'informatique. Nous avons donc retiré de la liste les enseignants d'autres UFR, et avons ajouté un gestionnaire de scolarité supplémentaire.

La liste finale des utilisateurs se composait de 12 personnes, conviées à participer au premier atelier : l'atelier de cadrage (cf. Figure 75).

Type d'utilisateur	Structure de rattachement	Prénom - nom	Adresse mail	Atelier de cadrage Invité
Enseignants	UFR Informatique - M2 IPM	Alain Berro	Alain.Berro@ut-capitole.fr	x
	UFR Informatique - M2 IPM	Frédéric Amblard	frederic.amblard@ut-capitole.fr	x
	UFR Informatique - M2 IPM	David Navarre	David.Navarre@ut-capitole.fr	x
	UFR Informatique - M2 IPM	Jean-Marc Thévenin	Jean-Marc.Thevenin@ut-capitole.fr	x
	UFR Informatique - M2 IPM	Eric Andonoff	Eric.Andonoff@ut-capitole.fr	x
Gestionnaires scolarité Responsables	FCV2A	Dominique Zou	dominique.zou@ut-capitole.fr	x
		Sébastien Garcia	sebastien.garcia@ut-capitole.fr	x
		Isabelle Paysan-Rouquairol	isabelle.paysan-rouquairol@ut-capitole.fr	x
		Véronique Lemozy	veronique.lemozy@ut-capitole.fr	x
		Delphine Renaud	delphine.renaud@ut-capitole.fr	x
		Sylvie Cardoso	sylvie.cardoso@ut-capitole.fr	x
		Adrien Blazy	adrien.blazy@ut-capitole.fr	x

Figure 75 : Liste des utilisateurs pour l'expérimentation FCV2A

6.1.2.2. Atelier de cadrage

L’atelier de cadrage s’est déroulé en mars 2018. Deux stratégies devaient nous permettre d’identifier les points de vue des utilisateurs du système d’information. Dans un premier temps, la troisième section du modèle de produit a été mise en œuvre dans sa quasi-globalité (cf. Figure 76).

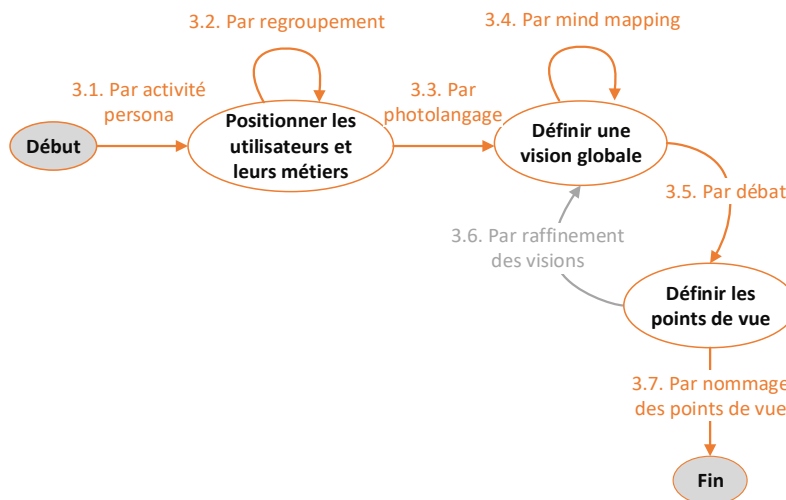


Figure 76 : Section 3 pour l'expérimentation FCV2A

Les utilisateurs, identifiés par leur nom et leur métier ont pu, au travers d’une activité de photolangage, donner leur vision du domaine concerné par le projet. Chacun des éléments qu’ils ont exprimé à partir des cartes de photolangage a été noté sur un post-it et positionné au tableau (cf. Figure 77).



Figure 77 : Illustration de la stratégie 3.3

Nous avons ensuite réalisé une mind map faisant apparaître deux points de vue divergents : le point de vue « enseignement » et le point de vue « administration » (cf. Figure 78). En effet, alors que certains acteurs ont insisté sur les aspects financiers, règlementaires et administratifs du service de Formation continue, d’autres ont insisté sur la formation des étudiants, les parcours et la pédagogie. De plus, des débats ont eu lieu sur l’équilibre rigueur administrative/adaptation des enseignements.

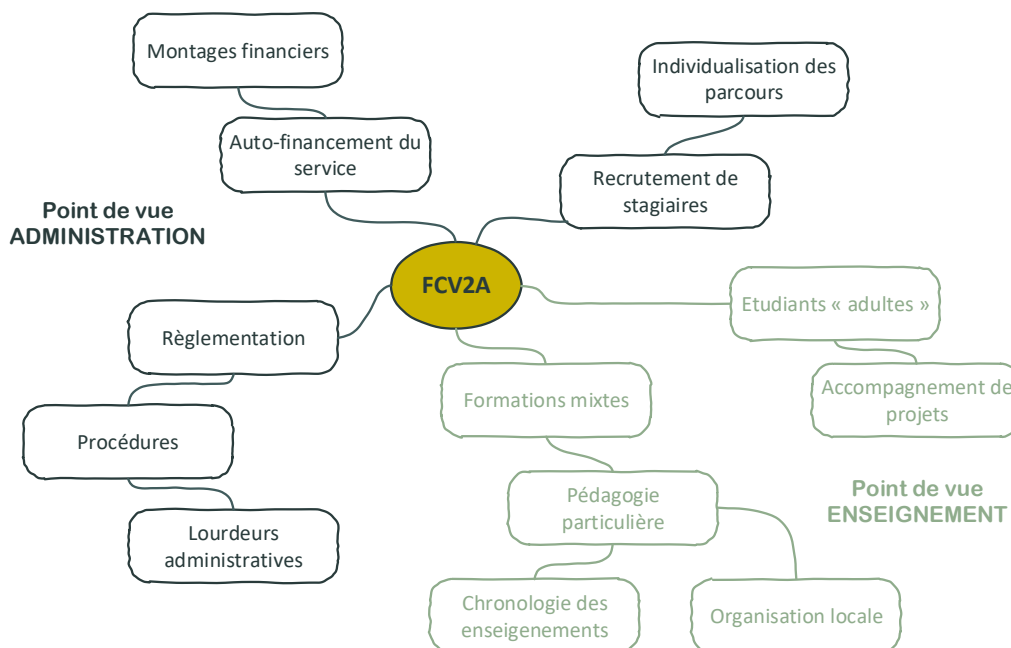


Figure 78 : Mind map des visions

Suite à cela, les participants ont mis en œuvre une partie des stratégies de la section 4 afin d’affiner la liste de points de vue obtenue (cf. Figure 79).

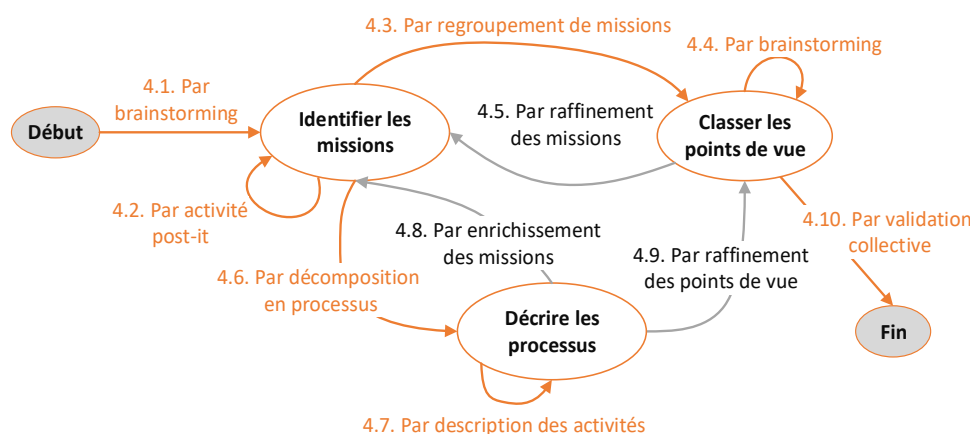


Figure 79 : Section 4 pour l’expérimentation FCV2A

Ils ont décrit leurs missions, qu’ils ont ensuite décomposées en processus. Suite au regroupement des missions, il a été considéré que certaines ne rentraient pas dans le périmètre du projet. Nous les avons laissées au tableau mais les avons barrées. Un processus principal de gestion des présences a été défini et détaillé au travers d’activités et de rôles. Suite à ces activités, le nom du point de vue « administration » a été modifié par les utilisateurs qui ont choisi de le nommer « gestion ». Le tableau blanc représentant les utilisateurs, les métiers, la vision des utilisateurs du domaine métier, les missions, les processus et les points de vue a été collectivement validé par l’ensemble des participants.

Suite à l’atelier, j’ai effectué en tant que modératrice de la méthode le compte-rendu de celui-ci. J’ai recopié chacun des éléments du tableau sur un document informatique, afin de le diffuser à tous les utilisateurs. La Figure 80 présente un extrait du document, et nous l’avons lié au modèle de produit de DEMOS afin de montrer de quelle façon il a pu être instancié. Par souci de clarté, seules certaines instances du modèle de produit sont données à titre d’exemple.

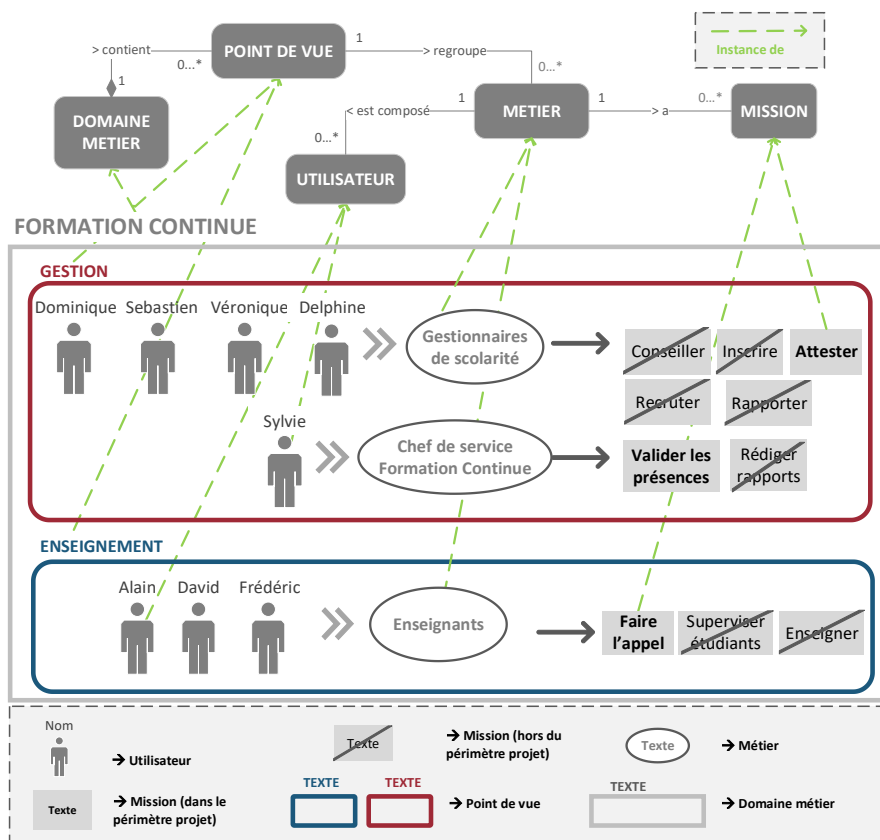


Figure 80 : Extrait d'instanciation d'une première partie du modèle de produit

6.1.2.3. Ateliers point de vue

Suite à l’atelier de cadrage, deux ateliers points de vue ont eu lieu : l’un avec les utilisateurs du point de vue « enseignement », l’autre avec les utilisateurs du point de vue « gestion ». Ces ateliers se sont déroulés au mois d’avril 2018.

Pour ces ateliers, nous avons préparé le tableau blanc en y notant les activités du processus qui étaient liées aux utilisateurs du point de vue convié. L’activité de photolangage a également été préparée en positionnant sur une table l’ensemble des photos (cf. Figure 81)



Figure 81 : Préparation de la salle pour l'atelier point de vue

Durant chacun des ateliers, toutes les stratégies de la section 5 ont été mises en œuvre (cf. Figure 82). L’objectif était alors de définir un modèle conceptuel par point de vue à l’issue de l’atelier.

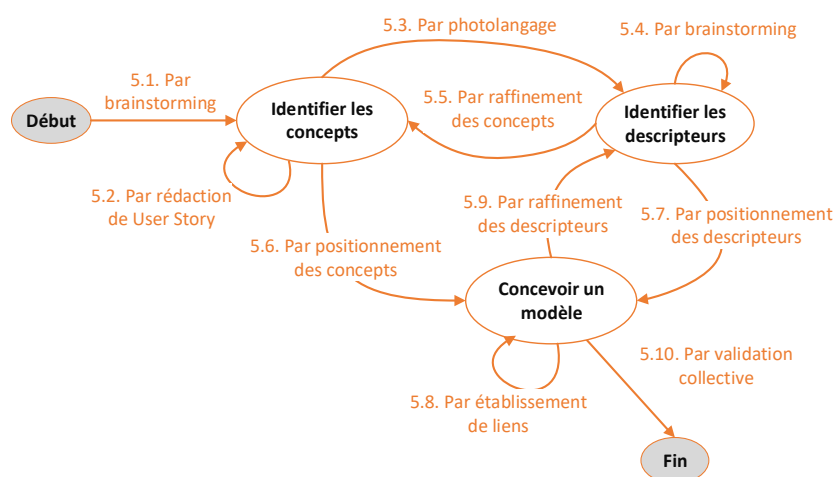


Figure 82 : Section 5 pour l'expérimentation FCV2A

Les utilisateurs ont dans un premier temps discuté des concepts clés qu'ils mettaient en œuvre dans le cadre de leur travail concernant la gestion des présences. Ils ont ensuite rédigé des User Stories afin de définir les besoins qu'ils avaient quant au futur outil informatique, et cela leur a permis d'identifier de façon claire les premiers concepts. Ils ont ensuite, via une activité de photolangage, décrit les concepts et je me suis chargée en tant que modératrice de noter les termes employés pour identifier les premiers descripteurs des concepts. Les utilisateurs ont complété cette liste et ont pu positionner les concepts et descripteurs au tableau pour esquisser un modèle conceptuel. Ils ont ensuite lié les concepts entre eux et ajusté la description de ceux-ci, et ont validé collectivement le modèle obtenu, ainsi que la liste de User Stories (cf. Figure 83).

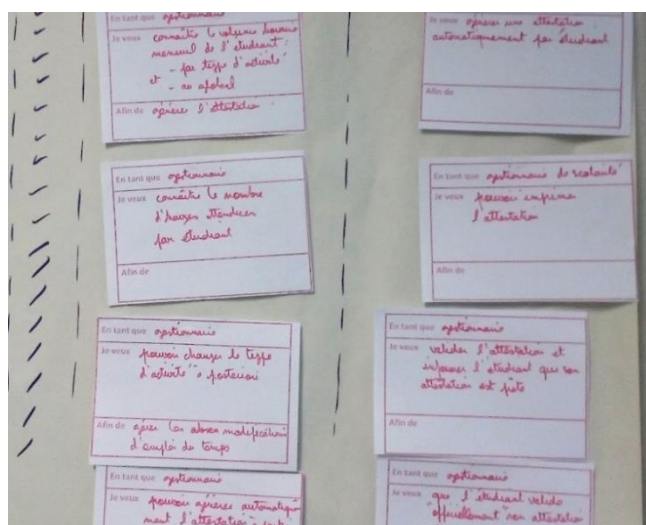


Figure 83 : Illustration de la stratégie 5.2

La Figure 84 présente un extrait de l'instanciation du modèle de produit obtenue à l'issue des deux ateliers point de vue. Nous avons choisi de présenter la modélisation conceptuelle de la notion de présence afin de montrer les différences explicites entre points de vue.

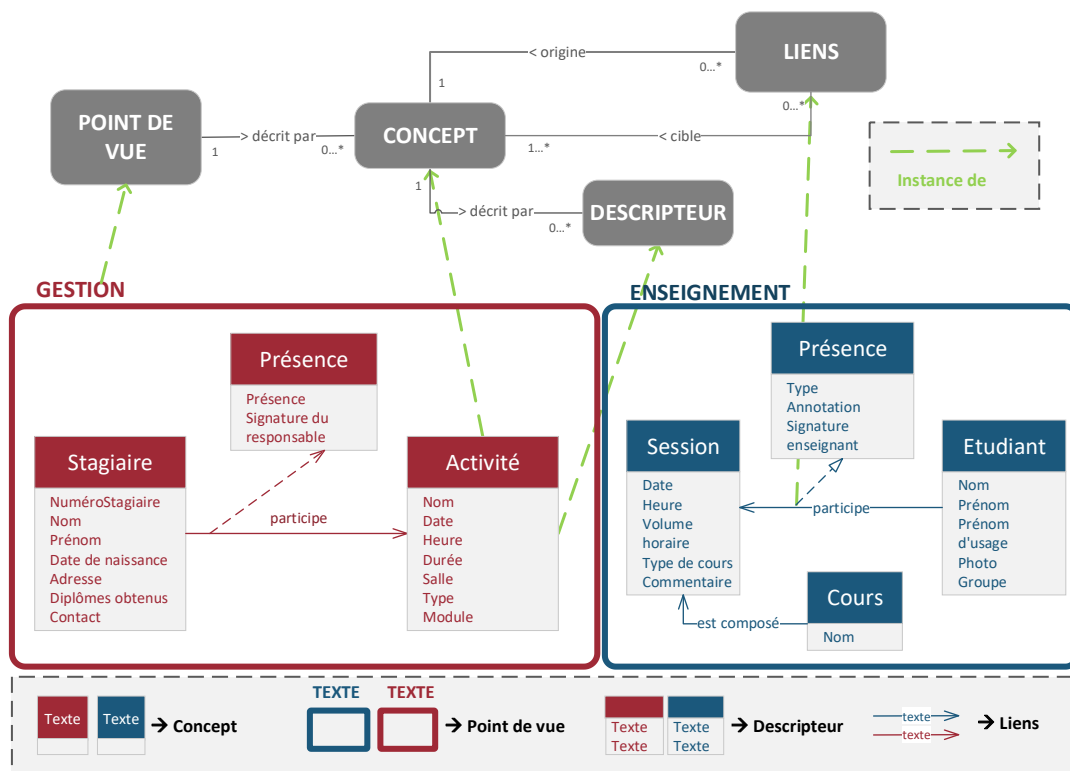


Figure 84 : Extrait d'instanciation d'une deuxième partie du modèle de produit

Il apparaît très clairement que des différences de vocabulaire existent entre les points de vue. Alors que le point de vue « gestion » parle de stagiaires, le point de vue « enseignement » parle lui d'étudiants. Leurs considérations sont également différentes quant à la notion de cours. Alors que pour le point de vue « enseignement » le cours est noté comme un concept, pour le point de vue « gestion » le cours est une activité parmi d'autres (stages, travail en bibliothèque, etc.). Ces différences confirment l'idée qu'il aurait été biaisé de demander à tous ces participants de réaliser une modélisation conceptuelle unique. Le fait de leur proposer une modélisation par point de vue leur a permis de s'exprimer dans leur vocabulaire courant, et nous permet d'obtenir une vision plus riche du domaine métier.

6.1.2.4. Atelier de mise en commun

En mai 2018 l'atelier de mise en commun a eu lieu. L'objectif était ici d'identifier les similarités entre les modèles afin de s'assurer que l'outil serait partagé entre les points de vue, supportant ainsi leur processus de travail commun. Toutes les stratégies de la section 7 ont donc été mises en œuvre, comme présenté en Figure 85.

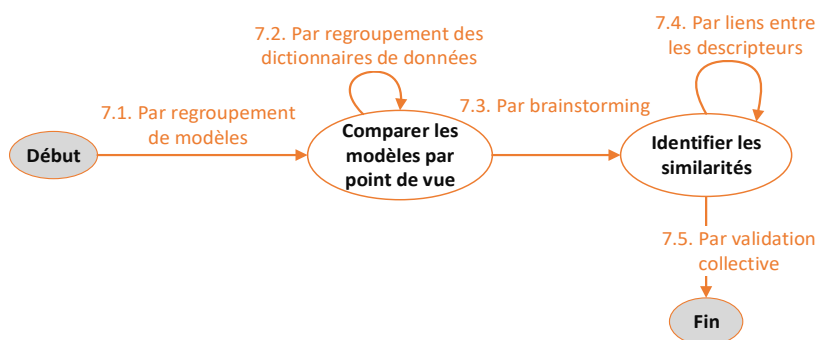


Figure 85 : Section 7 pour l'expérimentation FCV2A

Les utilisateurs ont longuement débattu des représentations de chacun de leur point de vue, et ont tracé les liens de similarité au tableau sur lequel les modèles et dictionnaires des données étaient notés (cf. Figure 86).



Figure 86 : Illustration de l'activité 7.3

La Figure 87 présente un extrait de l'instanciation du modèle de produit, qui est en lien direct avec l'extrait de modèle conceptuel présenté précédemment. On voit ici que sur 16 descripteurs du premier modèle et 14 descripteurs du second, 9 ont été identifiés comme présentant les liens de similarité.

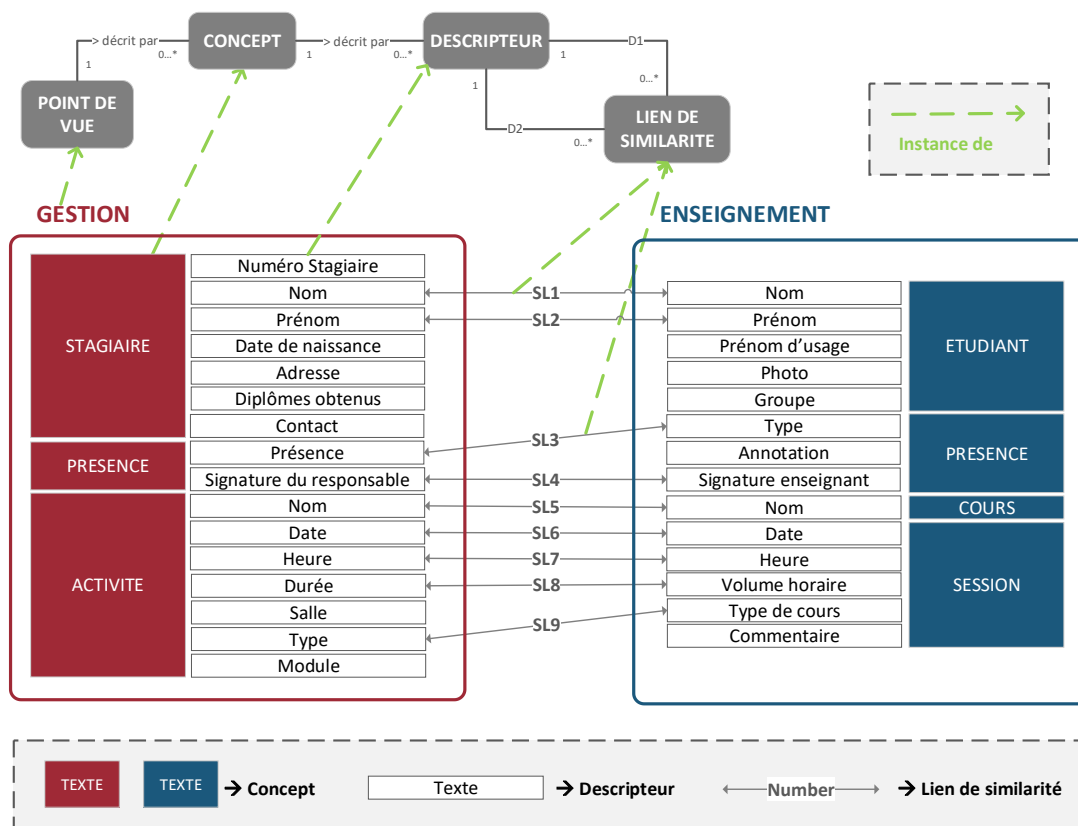


Figure 87 : Extrait d'instanciation d'une troisième partie du modèle de produit

Les discussions autour des liens de similarité ont poussé les utilisateurs à débattre de leurs représentations. Ils ont alors mis en œuvre la section 8 de notre modèle de processus permettant d’enrichir un modèle conceptuel par point de vue (cf. Figure 88). Certains descripteurs qui apparaissaient dans un seul des deux modèles ont pu être adoptés par l’autre point de vue qui les a alors ajoutés à son modèle conceptuel.

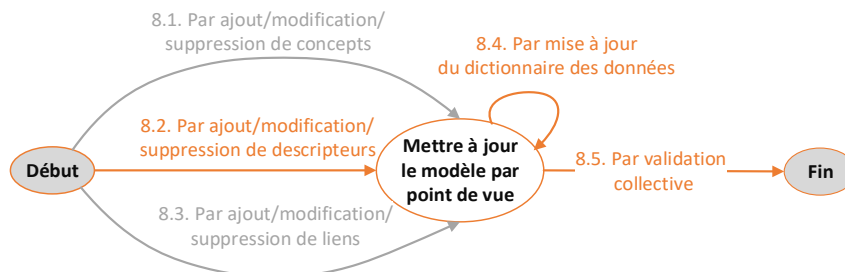


Figure 88 : Section 8 pour l'expérimentation FCV2A

6.1.2.5. Implémentation

Pour des raisons extérieures à l’expérimentation, l’implémentation des modèles conceptuels par point de vue et, par la suite, le développement du prototype de l’outil de gestion des présences a été fait un an après l’atelier de mise en commun. C’est alors la 9^{ème} et dernière section du modèle de processus qui a été appliquée (cf. Figure 89).

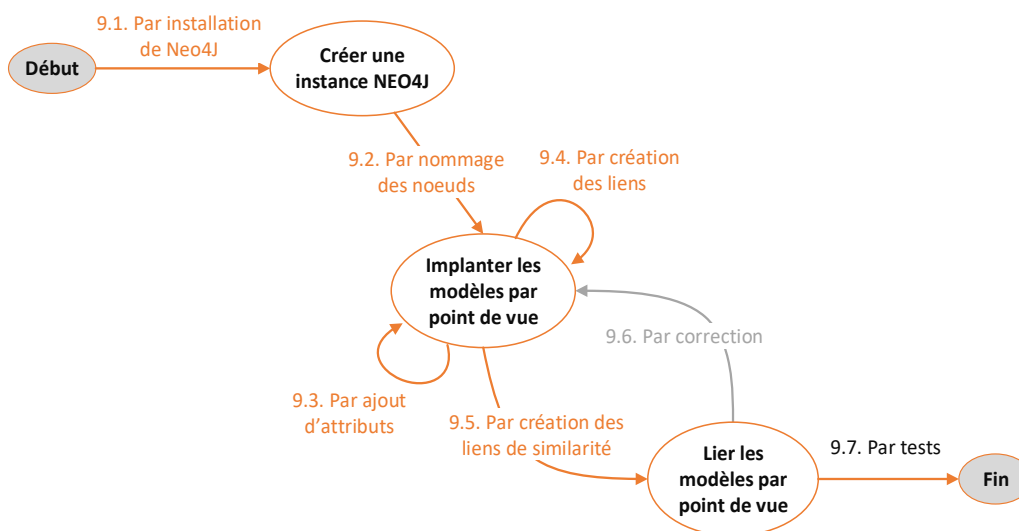


Figure 89 : Section 9 pour l'expérimentation FCV2A

Une instance de Neo4J a été ouverte afin d’implanter les modèles consolidés. Dans un premier temps les nœuds représentant les concepts ont été créés, puis des attributs leur ont été ajoutés pour implanter les descripteurs. Les nœuds ont été liés entre eux et nommés afin de représenter les liens du modèle de produit. Chaque modèle par point de vue s’est vu attribuer une couleur. Les modèles ont ensuite été consolidés grâce à l’ajout de nouveaux liens correspondant à nos liens de similarité. Les informations sur la similarité ont été ajoutées comme attributs du lien (cf. Figure 90).

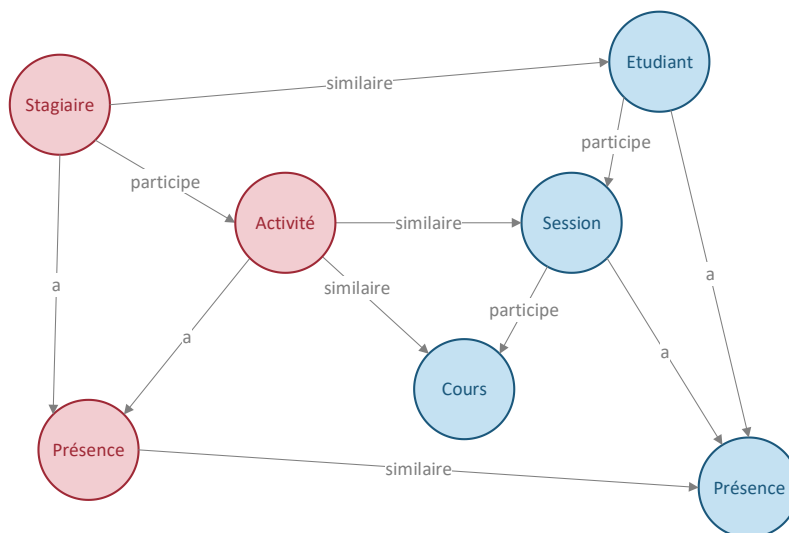


Figure 90 : Extrait de la base de données Neo4J

Suite à l’implantation de la base de données graphe et à l’insertion d’un jeu de données, un prototype de l’application a été développé. Les fonctionnalités choisies sont celles correspondant aux User Stories écrites par les utilisateurs eux-mêmes. A titre d’exemple, les figures suivantes (cf. Figure 91, Figure 92) présentent quelques écrans de l’application, avec la User Story correspondante.

La capture d'écran montre l'interface web de l'application 'Gestion des étudiants'. Le titre de la page est 'Gestion de Présences - Point de Vue Gestion'. Le formulaire principal est intitulé 'Entrez les données du nouveau Stagiaire' et contient des champs pour : Numéro Étudiant, Nom, Prénom, Date de Naissance (format jj/mm/aaaa), Coordonnées, Adresse (exemple: 1234 Main St), Dernier diplôme obtenu, Formation (menu déroulant avec 'STI' sélectionné), et Photo. Un bouton 'Inscrire Stagiaire' est visible en bas à gauche.

En tant que gestionnaire de scolarité

Je veux créer la liste de stagiaires (avec des caractéristiques particulières telles que leur nom d’usage)

Afin de les inscrire à des activités et consulter leur présence

Figure 91 : Création d'un stagiaire sur l'application

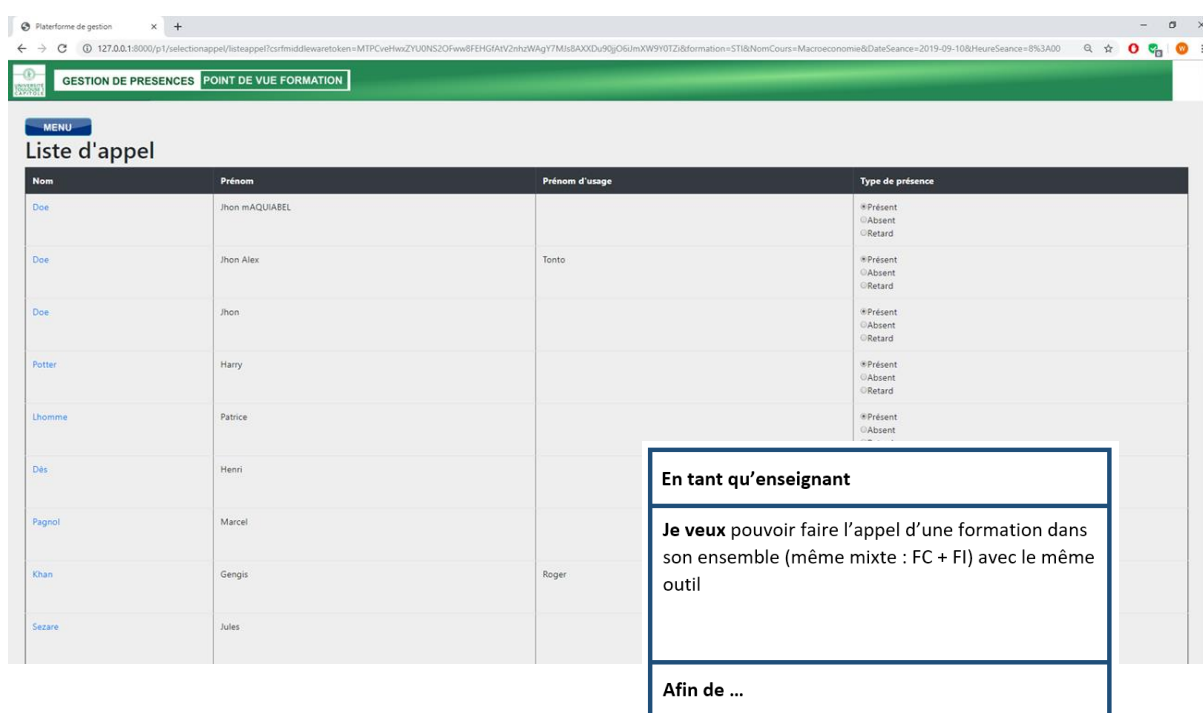


Figure 92 : Liste d'appel sur l'application

6.1.3. Evaluation et retours d'expérience

L'expérimentation ne saurait être complète sans une évaluation de la part des participants. Le prototype ayant été développé tardivement, nous n'avons pu évaluer qu'une partie de nos objectifs : ceux ayant trait à l'aspect démocratique de la démarche. En effet, les participants n'ont pas encore pu tester l'outil, ils ne peuvent se prononcer sur le caractère démocratique du système d'information lui-même, ni sur son caractère évolutif. Cette évaluation avait pour nous deux buts : d'une part recueillir les avis des participants de la méthode afin de nous assurer que l'objectif que nous nous étions fixés (proposer une méthode démocratique) soit atteint. D'autre part, envisager les modifications et adaptations à effectuer sur le modèle de produit et/ ou les modèles de processus en vue de la seconde expérimentation. C'est pour cette raison que nous proposons notre propre analyse de l'expérimentation suite à l'évaluation.

6.1.3.1. Evaluation par les participants

Nous avons sollicité les participants à l'issue de l'application de la méthode pour qu'ils évaluent DEMOS. Tous n'étant pas disponibles, il ne s'agit ici que d'une évaluation partielle. Nous avons mis en place un protocole d'évaluation afin de proposer aux participants de se positionner sur trois aspects : les intentions, les stratégies, les résultats de la méthode. Ce protocole est présenté dans la Table 17 suivante.

Table 17 : Protocole d'évaluation de la méthode

Hypothèse
DEMOS permet de concevoir un système d'information de façon démocratique
Revue de la méthode (intentions et stratégies) et des résultats obtenus durant l'expérimentation ~ 15 mn
Evaluation ~ 20 mn
Evaluation des intentions de la méthode
<i>Les participants ont-ils compris les intentions de la méthode ?</i>
<i>Les participants ont-ils trouvé l'enchaînement des étapes de la méthode pertinent ?</i>
Evaluation des stratégies de la méthode
<i>Les participants considèrent-ils que les stratégies aient permis d'atteindre les intentions ?</i>
<i>Les participants ont-ils trouvé les outils et techniques utilisés pertinents pour chaque stratégie ?</i>
Evaluation des résultats de la méthode
<i>Les résultats obtenus sont-ils en cohérence avec les intentions de la méthode ?</i>

Les évaluations ont révélé que les participants ont compris les intentions de la méthode : identifier les utilisateurs, définir les points de vue, concevoir un modèle par point de vue et consolider les modèles par point de vue. La notion de point de vue n'était pas « évidente » pour eux jusqu'à la fin de l'atelier de cadrage mais s'est « peu à peu clarifiée » et leur a semblé « pertinente ». Ils ont trouvé l'ordonnement des étapes « logique » et il leur a semblé que cette organisation par ateliers était « très structurante ». Les délais entre chacun des ateliers ont cependant semblé déstabiliser les utilisateurs qui ont eu du mal à se « replonger dans le sujet » après un mois.

Les stratégies de la méthode avaient pour but d'atteindre les intentions définies. Huit stratégies ont été mises en œuvre dans le cadre de la phase préparatoire, de l'atelier de cadrage, des ateliers point de vue, de l'atelier de mise en commun et de la phase d'implémentation. Dans l'ensemble, les participants ont considéré que les stratégies étaient « adaptées » aux intentions qu'elles permettaient d'atteindre. Les outils et techniques leur ont semblé « ludiques », « accessibles », « originaux ». Le photolangage notamment a été particulièrement apprécié car il laissait « l'occasion à chacun de participer, même ceux qui sont les plus réservés ». Les activités de brainstorming ont permis « aux gens de se parler entre eux » et les éléments au tableau blanc ont permis à chacun de « mieux se comprendre ». Ceci étant, les nombreux outils et techniques utilisés dans un « format court » d'ateliers ont donné lieu à des « ateliers très denses, où l'on a peur d'oublier des choses ».

Les résultats de la méthode ont été jugés « très satisfaisants » par les participants. Ils ont trouvé les comptes-rendus tout à fait « fidèles à ce qui avait été dit et écrit pendant l'atelier ». La modélisation par point de vue a surpris les utilisateurs eux-mêmes qui ne pensaient pas qu'ils pouvaient « représenter si simplement » et de manière « si complète » leur domaine métier. Les participants ont aussi évoqué leur regret de ne pouvoir évaluer l'outil, car ils craignent qu'il ne soit pas « conforme à ce qui a été fait lors des ateliers ».

6.1.3.2. Notre analyse de l'expérimentation

Notre analyse de cette expérimentation s'appuie sur l'évaluation qui en a été faite par les participants, mais aussi sur mon expérience en tant que modératrice de la méthode. D'un point de vue général, l'expérimentation s'est bien déroulée, mais l'évaluation s'est faite de façon trop informelle, et trop tardivement. Cela nous a empêché d'avoir des retours plus complets, et surtout plus « à chaud ». C'est

un point auquel nous allons veiller pour l'expérimentation suivante en incluant dans le processus une véritable phase d'évaluation, avec des entretiens individuels de chacun des participants.

Un premier biais manifeste a été introduit par le fait de solliciter des enseignants en informatique pour cette expérimentation. Ceux-ci ont des notions plus ou moins poussées en modélisation conceptuelle, et cette activité centrale n'a donc pas été faite de façon totalement neutre. L'expérimentation suivante nous permettra d'être face à des participants non-initiés, réduisant ainsi ce biais.

L'enchaînement des ateliers s'est fait avec un délai d'un mois à chaque fois, cela a été dommageable pour le projet selon nous. Nous avons par conséquent réduit ces délais dans l'expérimentation suivante, afin de dérouler le processus sur un mois maximum. Cela permet en outre de réduire la durée de chacun des ateliers, dont une partie non négligeable était consacrée à un rappel du travail préalablement effectué. On peut ainsi diminuer la densité des ateliers, et inclure dans ceux-ci de véritables moments de pause permettant de rediscuter d'éléments éventuellement oubliés.

Un dernier point nous semble important à prendre en compte pour une prochaine expérimentation : la phase d'implémentation doit être plus proche de celle de conception, et les participants doivent avoir une vision sur cette phase afin de s'assurer que l'outil développé l'est en accord avec ce qui a été décrit lors des ateliers. Cela permet de plus de proposer une évaluation de l'outil lui-même, et de son impacts sur la démocratie dans l'organisation : les points de vue sont-ils respectés dans l'outil ? Peuvent-ils évoluer ?

De manière générale, ce qui est ressorti de cette expérimentation est qu'elle a permis de faire travailler ensemble des personnes qui, bien que partageant un processus commun, ne dialoguaient que très peu. Cela avait pour cause des incompréhensions, et parfois des tensions entre enseignants et gestionnaires. Le fait de rassembler ces acteurs, et de leur permettre de concevoir un logiciel selon leurs points de vue respectifs a été l'apport majeur de la méthode.

6.2. Deuxième expérimentation : Plateforme Tandem Canada

En 2019, une collaboration avec l'université Laval au Québec a donné lieu à la planification d'une expérimentation pour la réingénierie d'une plateforme d'apprentissage des langues en tandem. Le projet « Tandem Canada » implique de nombreux acteurs de l'Université tels que des experts pédagogiques, des enseignants et des apprenants qui ont pour le moment développé une plateforme d'apprentissage en tandem sans véritable méthode de conception. Le résultat ne leur donnant pas satisfaction, ils ont souhaité concevoir une nouvelle plateforme avec une démarche formalisée. L'expérimentation aura lieu dans le courant de l'année 2020 et est donc actuellement en préparation. Nous présentons dans les sections suivantes le contexte de l'expérimentation à venir, ainsi que la phase préparatoire incluant les pistes d'amélioration évoquées avec la première expérimentation (cf. section 6.1.3).

6.2.1. Contexte de l'expérimentation et besoins exprimés

Le projet de plateforme Tandem Canada est porté par des enseignantes et des conseillers pédagogiques de l'université Laval au Québec. Le bilinguisme qui a cours au Canada aujourd'hui a été à l'origine de cette innovation pédagogique, qui consiste à faire travailler en tandem une classe de francophones apprenant l'anglais et une classe d'anglophones apprenant le français. Le but est de leur permettre, via une plateforme numérique, de communiquer, d'échanger, de travailler ensemble afin de progresser dans l'apprentissage de langues hors du contexte de la classe. Le système d'information soutenant cette innovation pédagogique est central, puisque c'est au travers de ce système que les outils d'échange sont proposés.

En 2018, la plateforme Tandem Canada a été développée avec l'appui de plusieurs enseignants et d'experts pédagogiques de l'Université Laval. Ce développement « au fil de l'eau » s'est fait sans véritable démarche de conception et de recueil des exigences préalable. Aujourd'hui, l'usage de la plateforme par les apprenants souffre de ce manque de conception, puisque la plateforme ne donne pas satisfaction.

Plusieurs raisons peuvent être invoquées pour expliquer cet « échec » :

- D'une part le développement de la plateforme a correspondu à l'adaptation d'une instance de Moodle. Il s'agissait ici de customiser une structure très rigide, et il n'a donc pas été permis de réfléchir à une véritable architecture innovante. Cette dernière est pourtant indispensable (Bour et al. 2019).
- D'autre part, tous les utilisateurs de la plateforme n'ont pas été impliqués dans le processus de développement. De fait, cela a empêché de développer une plateforme qui convienne aux acteurs qui n'avaient pas participé. Il s'agit ici principalement des apprenants, qui considèrent aujourd'hui la plateforme comme complexe et difficilement utilisable.
- Enfin, il n'y a pas eu de méthode structurée, et l'aspect trop peu formel du développement ne permet pas aujourd'hui d'expliquer certains des choix qui ont été faits, et de venir les modifier.

C'est dans ce contexte que nous avons souhaité proposer une expérimentation de la méthode DEMOS. En nous appuyant sur les pistes d'amélioration identifiées suite à la première expérimentation que nous avons menée, il nous semble que DEMOS peut être une réponse aux problématiques de la plateforme Tandem Canada. Le but est de proposer des modèles conceptuels permettant le développement d'une plateforme sur mesure ou une réingénierie de la plateforme actuellement en place si cela est possible.

6.2.2. Préparation de l'expérimentation

Cette seconde expérimentation aura lieu dans le courant de l'année 2020. Un travail préliminaire nous a permis de la préparer en identifiant les deux principales problématiques du projet :

- Multiplicité d'acteurs aux préoccupations différentes impliqués : enseignants, apprenants, conseillers pédagogiques, etc. Cela signifie que des points de vue différents sur le projet sont à prévoir.
- Système à développer en support à une innovation pédagogique partiellement formalisée, en cours de conception. Cela signifie que la plateforme devra être en capacité d'évoluer avec l'apparition de besoins complémentaires.

La première expérimentation que nous avons menée dans le cadre du projet avec la FCV2A nous a également permis de préparer l'expérimentation en termes de risques. Nous avons identifié plusieurs points à ne pas négliger. Dans un premier temps, la participation de tous les utilisateurs est indispensable à la bonne réussite de l'expérimentation. Pour cela, les ateliers doivent être planifiés suffisamment en amont pour que chacun puisse se dégager une disponibilité suffisante. Ces utilisateurs seront les testeurs du premier prototype de l'outil. Dans un second temps, la démarche doit pouvoir être déroulée dans un délai d'un mois (avec une semaine entre chaque atelier par exemple). Cela permet de ne pas avoir de longs rappels à effectuer au début de chaque atelier, les rendant ainsi moins denses. Ainsi, des pauses en cours d'ateliers peuvent être prévues afin que les utilisateurs puissent prendre du recul sur le travail effectué. La boucle d'amélioration continue permettra éventuellement par la suite de revenir sur des éléments oubliés ou qu'il est nécessaire de faire évoluer. Dans un troisième temps, le prototype doit pouvoir être développé rapidement à l'issue de la phase de conception. Ainsi, les utilisateurs peuvent être impliqués dans la phase de

développement en validant les incréments livrés, selon leur conformité aux propositions faites en phase de conception. Enfin, deux évaluations formelles doivent être prévues : l'une à l'issue de la démarche de conception, pour juger de sa pertinence au regard de l'objectif poursuivi (une méthode de conception participative et démocratique) ; l'autre après une première phase de test de l'outil pour juger du second objectif que nous nous sommes fixés : un système d'information support à la démocratie (c'est-à-dire intégrant et respectant une pluralité de points de vue).

6.3. Conclusion du chapitre

Depuis le début du développement de DEMOS, nous avons souhaité que notre démarche soit opérationnelle. Les expérimentations ont donc fait partie intégrante de notre travail, afin de valider nos objectifs, et de nous assurer que DEMOS pouvait être transférée au monde des organisations. La première expérimentation que nous avons menée avec le FCV2A nous a permis d'affiner notre formalisation de DEMOS, et a été l'occasion pour nous de repérer les différentes pistes d'amélioration à mettre en œuvre. L'expérimentation prochaine pour la plateforme Tandem Canada est de plus grande ampleur. Elle nous permettra de valider notre travail dans un nouveau domaine et avec un nombre plus important d'acteurs.

CONCLUSION ET PERSPECTIVES

Bilan des contributions

Nous avons montré, au travers de cette thèse, que le système d'information pouvait être un soutien à la démocratie des organisations. Pour cela, nous avons identifié les caractéristiques d'un tel système, et avons proposé une méthode de conception participative de système d'information reposant sur **cinq principes** :

- Impliquer les utilisateurs finaux dans un processus participatif et démocratique,
- Favoriser un débat démocratique permettant l'émergence de points de vue,
- Concevoir un système d'information démocratique qui prend en compte ces points de vue,
- Assurer une traçabilité des points de vue dans le système d'information,
- Permettre l'évolution des points de vue dans le cycle de vie du système.

Le modèle de produit et les modèles de processus sont des garanties du respect de ces principes. En effet, la syntaxe abstraite permet de représenter les aspects statiques, dynamiques, fonctionnels et prospectifs du modèle. Nous garantissons ainsi que les modèles issus de la phase de conception intègrent la notion de point de vue, et que ceux-ci sont traçables ensuite. La syntaxe concrète liée à la syntaxe abstraite est elle une garantie d'intelligibilité du modèle par des utilisateurs « non experts » et par des techniciens. Les modèles de processus, avec leurs intentions et stratégies prévoient de construire et de faire évoluer le système d'information de façon démocratique, c'est-à-dire en proposant une dispute démocratique autour des visions et valeurs portées par le système entre tous les utilisateurs, mais également en outillant une boucle d'amélioration continue du système permettant l'évolution des points de vue. Les utilisateurs sont au cœur de la méthode. Ils sont impliqués dans sa conception, depuis le recueil des exigences jusqu'au développement, et ensuite durant tout le cycle de vie du système. Ils participent en tant que co-constructeurs, et ont à ce titre un véritable pouvoir de décision quant au système d'information à développer.

Ces contributions ont été proposées en suivant un axe structurant : le respect de la démocratie, tel que nous avons défini ce concept. Ainsi, nous proposons une méthode adaptable, extensible, à laquelle des intentions et stratégies peuvent être ajoutées. Les expérimentations de DEMOS nous permettent ainsi dans un premier temps de valider la méthode, mais aussi de découvrir les pistes d'amélioration qui peuvent être envisagées. Actuellement, la méthode et sa plateforme support sont opérationnelles. DEMOS peut être mis entre les mains de différents modérateurs, qui auront un guide formalisé pour la mise en œuvre de DEMOS.

Nous avons annoncé que DEMOS serait une réponse à des enjeux organisationnels d'aujourd'hui : gestion du Shadow IT, innovation et capitalisation de la connaissance selon le principe des Communs. Maintenant que nous avons détaillé DEMOS, nous pouvons répondre à ces enjeux de manière plus

précise. Pour ce qui est du Shadow IT, nous considérons que DEMOS est une façon de limiter les causes de recours au Shadow IT, tout en nous appuyant sur les leviers qui rendent possible cette pratique. En effet, la cause principale du Shadow IT est l'inadéquation des outils informatiques aux besoins réels. Cette inadéquation peut être due à une mauvaise conception des outils, ou à une impossibilité de les faire évoluer. DEMOS est selon nous une réponse à ces deux points. En proposant une méthode de construction de systèmes d'information participative, impliquant les utilisateurs et leur permettant de débattre, elle garantit une adéquation la plus grande possible entre besoins utilisateurs et système développé. La boucle d'amélioration continue quant à elle permet de répondre aux besoins d'évolution des utilisateurs, en formalisant un processus reproductible autant de fois que nécessaire pour adapter le système à de nouveaux besoins. Mais DEMOS s'appuie également sur les leviers qui font exister le Shadow IT : d'un côté la volonté des utilisateurs de rechercher des outils adaptés, afin de faire correctement leur travail et d'améliorer leurs pratiques ; de l'autre côté leur capacité d'abstraction leur permettant de concevoir une solution informatique.

L'usage des systèmes d'information peut être un frein à l'innovation, mais aussi un véritable levier. Notre contribution répond à ces deux aspects : d'une part en permettant de ne pas freiner l'innovation dans les organisations, d'autre part en la soutenant au travers des outils informatiques. Si l'innovation émerge des réponses nouvelles, inventives, créatives des travailleurs qui se confrontent à des situations inhabituelles, alors les outils informatiques qui normalisent les activités et contraignent certains dans leur travail peuvent représenter un frein. Avec DEMOS, nous proposons de construire des systèmes d'information plus souples, plus adaptables et plus évolutifs afin de répondre à cette problématique, grâce notamment au processus d'amélioration continue que nous proposons. Dans le prolongement de cela, si l'on s'accorde à dire que l'innovation doit reposer sur de nouvelles représentations, on peut considérer que DEMOS concoure à cet objectif. En effet, le processus de construction propose d'intégrer de véritables débats et discussions sur la vision que les utilisateurs ont de leur domaine métier, et sur les missions qu'ils doivent mettre en œuvre au service de cette vision. Cette étape peut être une opportunité de « sortir du cadre », afin d'envisager des représentations innovantes du métier sur lesquelles faire reposer le système d'information.

Enfin, le dernier enjeu que nous avons identifié porte sur la capitalisation des connaissances au travers du Commun. Il nous semble aujourd'hui indispensable que, pour partager et diffuser les savoirs dans l'organisation, le système d'information devienne un véritable Commun : évolutif, complet et aux mains d'une Communauté. Avec DEMOS, les connaissances « encapsulées » dans le système d'information ne sont pas figées, et peuvent évoluer grâce à la boucle d'amélioration continue. De plus, la représentation de connaissances, sans être absolument complète, est riche grâce à la prise en compte d'une multiplicité de points de vue depuis le recueil des exigences jusqu'à la conception, puis ensuite dans les structures du système d'information lui-même. Enfin, la Communauté qui gère le Commun peut être incarnée par les utilisateurs, présents tout au long de la méthode, et par les « experts techniques », qui peuvent être intégrés tôt dans le projet. L'intelligibilité mutuelle qui peut alors s'établir entre eux est une garantie que la Communauté n'est pas dépossédée du Commun.

Perspectives

Le sujet de recherche que nous avons adressé dans cette thèse est large et dense. Notre contribution pourrait être étendue dans une perspective de court terme, en poursuivant notre travail sur une lancée très opérationnelle. En revanche, l'inscription de cette contribution dans le contexte des organisations et de la démocratie amène à envisager des perspectives de plus long terme.

Perspectives de court terme

La dispute démocratique et le droit à la remise en cause des règles, des normes établies est au cœur du concept de démocratie. Nous ne pouvons donc que souhaiter que la méthode proposée soit adaptée, modifiée, améliorée, afin qu'elle continue de servir l'objectif que nous nous sommes fixé : le soutien à la démocratie des organisations. Parmi les pistes d'amélioration, plusieurs ont été évoquées dans les chapitres précédents, et notamment à la suite des évaluations de notre première expérimentation. Nous en développons quelques-unes dans les paragraphes suivants.

Dans un premier temps, il nous semble souhaitable de poursuivre les expérimentations sur des cas plus diversifiés de construction de systèmes d'information. Parmi ces cas, une expérimentation de plus grande ampleur nous permettrait par exemple d'éprouver la méthode avec la présence de représentants des utilisateurs, élus ou tirés au sort, afin de vérifier que le passage à l'échelle de DEMOS est possible. D'autre part, dans certaines organisations les conflits autour des normes, valeurs et visions sont plus forts, et l'apport d'une méthode par point de vue pourrait permettre de résoudre des problématiques complexes. Cela permettrait de mettre davantage en avant le concept de point de vue utilisateurs qui est pour nous central. En effet, pour le moment les expérimentations nous conduisent à définir les points de vue de manière relativement « naturelle » (gestion VS enseignement par exemple). Nous souhaiterions aussi pouvoir évaluer la boucle d'amélioration continue avec un cas concret. Il nous faudrait pour cela travailler à une expérimentation sur une longue durée, permettant d'avoir une vision sur le système lors de son développement, puis durant les premières années de son utilisation. De telles expérimentations nous permettraient très certainement de venir enrichir la méthode par l'ajout de stratégies (en diversifiant l'offre d'activités et d'outils par exemple), voire d'intentions. Le fait que notre méthode soit modulaire et adaptable permet en effet de venir la compléter, la personnaliser à des cas particuliers sans pour autant en faire une méthode complexe. Chaque équipe projet (modérateur et utilisateurs) peut ainsi faire ses choix parmi les stratégies à mettre en œuvre, et décider de n'en appliquer qu'une partie.

Dans un second temps, la plateforme support de la méthode pourrait être étendue. Pour cela, nous avons identifié trois pistes à explorer. La première serait celle de l'industrialisation de la plateforme, qui consiste tout d'abord à garantir un accès distant à celle-ci, afin qu'elle soit utilisable par plusieurs modérateurs, et consultable par de nombreux utilisateurs en même temps. Il serait également bénéfique d'améliorer l'interface graphique actuelle, afin de la rapprocher des représentations obtenues en atelier. Une autre façon d'industrialiser la plateforme pourrait être d'étendre ses fonctionnalités afin qu'elle couvre l'ensemble du modèle de produit, et notamment les aspects dynamiques de celui-ci qui ne sont actuellement pas pris en compte. Une deuxième piste relative à la plateforme serait une extension de celle-ci jusqu'à une implémentation des modèles de données consolidés. En effet, la méthode prévoit aujourd'hui de parvenir à une modélisation conceptuelle aboutie des données, et la plateforme pourrait intégrer des passages automatisés jusqu'à une implantation dans une base de données graphe par exemple, puisque c'est actuellement la stratégie envisagée. La troisième piste serait de prolonger l'usage de la plateforme jusqu'à la boucle d'amélioration continue. En intégrant des mécanismes de détection de modifications des modèles par point de vue, la plateforme pourrait venir soutenir les ateliers de mise à jour, et assurer un rôle d'archivage des différentes versions de systèmes.

Dans un troisième temps, notre méthode s'attache aujourd'hui à une description approfondie des modèles de données et l'implantation proposée est d'ailleurs actuellement focalisée sur cette question. Pour autant, les aspects dynamiques et fonctionnels ne sont pas laissés de côté et font eux aussi l'objet d'une description (au travers du Business Process Model et de la rédaction des User Stories notamment). Cependant, la façon dont ils doivent être développés est laissée à l'appréciation des

développeurs eux-mêmes puisque la méthode ne fait pas de préconisations à ce sujet. Dans le futur, nos stratégies d'implantation pourront être étendues à ces aspects fonctionnels et dynamiques en proposant des stratégies formalisées de développement. Il faudrait pour cela ajouter au modèle de processus de construction des stratégies renforçant davantage la description des processus et activités, mais également des stratégies relatives au Story mapping (organisation des User Stories autour des priorités et des risques identifiés).

Une dernière perspective à court terme serait le lien entre notre méthode et une méthode de développement agile comme SCRUM. Comme nous l'avons vu, la plupart des méthodes agiles souffrent d'un manque de formalisation des phases de recueil des besoins et de conception générale permettant d'aboutir à l'élaboration du backlog de produit. DEMOS formalise ces phases amont et conduit à la rédaction de User Stories qui constituent ce backlog. Cela dit, les méthodes agiles fonctionnent selon des cycles itératifs et incrémentaux, tant du point de vue du développement que du cycle de vie du backlog de produit. Afin que notre méthode soit totalement compatible avec une méthode de développement telle que Scrum, il nous faudrait « l'agiliser » afin qu'elle réponde à l'exigence d'un backlog de produit vivant. La forme, pour le moment Bottom-up de notre démarche pourrait être envisagée dans une version itérative et incrémentale, prévoyant des ateliers « point de vue » à chaque début de cycle de développement. La présence d'équipes techniques durant ces ateliers serait en outre la garantie d'un partage de visions entre utilisateurs et techniciens lors des phases de conception.

Perspectives de plus long terme

Avant tout, il nous semble nécessaire de poursuivre le travail engagé sur l'étude des systèmes d'information avec une vision pluridisciplinaire. Les échanges entre communautés d'horizons différents nourrissent les réflexions des chercheurs en informatique en les inscrivant dans une perspective sociétale et, à l'inverse, les informaticiens ont à apporter aux chercheurs d'autres disciplines afin d'ancrer leurs raisonnements dans des problématiques très concrètes, parfois techniques. Ces recherches, que nous menons aujourd'hui au sein de l'atelier « Système d'information et Démocratie » devront, dans l'avenir, renforcer leur ancrage dans les organisations. Pour cela, des entretiens avec des travailleurs ou leurs représentants sont envisagés, pour baser notre travail sur des cas concrets et des retours d'expérience. Notre objectif est également de diffuser ces recherches dans le monde industriel, afin de partager les constats que nous faisons et de proposer des solutions opérationnelles. Enfin, nous souhaitons que notre approche trouve un écho dans l'enseignement des systèmes d'information en proposant des modules s'intégrant dans le parcours pédagogique des étudiants en informatique.

Nous voyons aujourd'hui que les discussions autour de l'éthique et du numérique se focalisent principalement sur des problématiques d'Intelligence Artificielle (IA). Cette préoccupation grandissante vient naturellement avec une diffusion de l'IA dans le monde de la recherche et dans la société, et est donc totalement justifiée. De même, les différentes législations qui émergent, telles que la RGPD (Règlementation Générale sur la protection des Données) nous semblent nécessaires pour protéger et sécuriser chacun, dans sa vie personnelle et sa vie professionnelle, quant à ses données personnelles. Ceci étant, l'éthique informatique ne saurait pour nous se réduire à ces situations. Avec des principes majeurs tels que la démocratie, l'éthique doit être un sujet dans les organisations, dans la vie au travail. Pour cela, il faut révéler les visions et valeurs qui sous-tendent les systèmes, et démontrer leur traduction directe sur les pratiques de travail supportées par le système d'information. Il nous faut également continuer de montrer en quoi une plus grande éthique dans la conception de systèmes d'information permettrait de construire des environnements de travail plus adaptés, servant

l'intérêt collectif et moins contraignant. Ces considérations pourraient être intégrées dans un projet plus global de Green IT ou de systèmes durables.

Enfin, et dans le prolongement de ce que nous venons d'évoquer, il nous semble que la recherche en système d'information doit se faire en lien étroit avec les recherches en management, afin d'accompagner les mutations organisationnelles que vivent les entreprises aujourd'hui. En effet, la promotion du management bienveillant, de l'entreprise libérée qui ont cours actuellement écartent pour le moment, dans leurs pratiques et dans leur recherche, la question du système d'information. Or, sans remettre en cause les bienfaits des sessions de team building ou les avantages à mettre plus en avant l'intelligence collective, il nous semble qu'une remise en question des outils informatiques est nécessaire à tout changement organisationnel profond. Ici encore, une analyse des systèmes d'information et de leur impact sous un angle éthique permettrait de rechercher des solutions, telles que DEMOS, qui permettraient *profondément* et *concrètement* d'améliorer le bien-être au travail.

Bibliographie

A

Aggeri, Franck. 2016. « L'obsession de la productivité et la fabrique du chercheur publiant ». *Le Libellio d'ÆGIS*, Ecrire et publier, 12 (2).

Aubry, Claude. 2018. *Scrum: Pour une pratique vivante de l'agilité*. 5ème édition. Dunod.

Aubry, Claude, et François Beauregard. 2013. *Scrum: le guide pratique de la méthode agile la plus populaire*. Dunod.

B

Bachet, Daniel. 2019. « Système d'information, outil comptable et démocratie dans l'entreprise ». In *3ème édition de l'atelier Système d'information et Démocratie dans les organisations - Inforsid*. Paris.

Barcellini, Flore, Laurent Van Belleghem, et François Daniellou. 2013. « Les projets de conception comme opportunité de développements des activités ». In *Ergonomie constructive*, 191-206. PUF.

Beck, Kent. 2003. *Test-driven development: by example*. The Addison-Wesley signature series. Boston: Addison-Wesley.

Berardi, Daniela, Diego Calvanese, et Giuseppe De Giacomo. 2005. « Reasoning on UML class diagrams ». *Artificial Intelligence* 168 (1-2): 70-118.

Beywl, Wolfgang, et Philip Potter. 1998. « Renomo - a Design Tool for Evaluations ». *Evaluation* 4 (1): 53-71.

Bihanic, David, Max Chevalier, Sophie Dupuy-Chessa, Xavier Le Pallec, et Thierry Morineau. 2013. « Modélisation graphique des SI ». In *Inforsid*. Paris.

Boboc, Anca. 2017. « Numérique et travail : quelles influences ? » *Sociologies pratiques* 34 (1): 3.

Boissin, Olivier, Hervé Charmettant, Jean-Yves Juban, et Yves Renou. 2015. « Les dynamiques plurielles d'innovation au sein des SCOP : les conditions d'un entrepreneuriat d'utilité sociale ». In *XVe rencontres du Réseau interuniversitaire de l'économie sociale et solidaire « La créativité de l'Économie sociale et solidaire est-elle soluble dans l'entrepreneuriat ? »* Reims, France.

Bour, Raphaëlle, Chantal Soule-Dupuy, et Nathalie Vallès-Parlangeau. 2015. « Les outils informatiques, un frein à l'innovation ? Proposition d'une démarche et d'une plateforme pour libérer l'innovation participative ». *Revue Génie Logiciel*, no 115: 26-35.

Bour, Raphaëlle, Chantal Soule-Dupuy, et Nathalie Vallès-Parlangeau. 2019a. « DEMOS : a Design Method for democratic information System ». In *13th International Conference on Research Challenges in Information Science (RCIS)*. Bruxelles, Belgique.

Bour, Raphaëlle, Chantal Soule-Dupuy, et Nathalie Vallès-Parlangeau. 2019b. « DEMOS: a participatory design approach for democratic empowerment of IS users ». In *38th International Conference, ER 2019*. Salvador de Bahia, Brésil. [à paraître]

- Bour, Raphaëlle, Chantal Soule-Dupuy, et Nathalie Vallès-Parlangeau. 2019c. « DEMOS : une méthode de conception participative pour un empowerment démocratique des utilisateurs de SI ». In *Inforsid*. Paris.
- Bour, Raphaëlle, Chantal Soule-Dupuy, et Nathalie Vallès-Parlangeau. 2019c. « DEMOS: a participatory design approach for democratic empowerment of IS users ». In *38th International Conference, ER 2019*. Salvador de Bahia, Brésil. [à paraître]
- Bour, Raphaëlle. 2017. « Les processus de conception des systèmes d'information sont-ils des processus démocratiques ? » In *1ère édition de l'atelier Système d'information et Démocratie dans les organisations - Inforsid*. Toulouse.
- Bour, Raphaëlle. 2018. « Les démarches agiles rendent-elle le processus de conception et développement plus démocratique ? » In *2ème édition de l'atelier Système d'information et Démocratie dans les organisations - Inforsid*. Nantes.
- Bour, Raphaëlle, Maryse Salles, et Claude Paraponaris. 2019. « Les systèmes d'information numériques, instruments de la gouvernementalité ? » In *3ème édition de l'atelier Système d'information et Démocratie dans les organisations - Inforsid*. Paris.
- Bour, Raphaëlle, Nathalie Vallès-Parlangeau, Chantal Soule-Dupuy, et Laurence Capus. 2019. « Towards a co-design approach to digital device for learning ». In *Edulearn19 Proceedings*. Palma de mallorqua.
- Boyle, James. 2003. « The Second Enclosure Movement and the Construction of the Public Domain ». *SSRN Electronic Journal*.
- Brière, Thibaud, et Thibault Le Texier. 2018. « Introduction : démocratiser l'entreprise ». *Esprit : Démocratiser l'entreprise* 2018/3: 40-51.
- Brinkkemper, Sjaak. 1996. « Method engineering : Engineering of information systems development methods and tools ». *Information & Software Technology* 38: 275-80.
- Brinkkemper, Sjaak, et Frank Harmsen. 1998. « Assembly techniques for method engineering ». In *Active Flow and Combustion Control*, 141:381-400. Cham: Springer International Publishing.
- Bruchez, Rudi. 2015. *Les bases de données NoSQL et le big data: comprendre et mettre en œuvre*. Paris: Eyrolles.

C

- Caelen, Jean. 2004. *Le consommateur au coeur de l'innovation*. Paris: CNRS éditions.
- Caron, Olivier, Alexis Muller, et Gilles Vanwormhoudt. 2003. « A Framework for Supporting Views in Component Oriented Information Systems ». In *Object-Oriented Information Systems, 9th International Conference*, 2817:164-78. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Carré, Bernard. 1989. « Méthodologie orientée objet pour la représentation des connaissances : concepts de point de vue, de représentation multiple et évolutive d'objet ». Thèse de doctorat en informatique, Université Lille 1.
- Castoriadis, Cornelius. 1975. *L'institution imaginaire de la société*. Edition du Seuil. Points Essais 383. Paris.

- Caudron, Fabrice. 2008. « Lier pratiques démocratiques et efficacité économique : la rationalisation démocratique ». *Revue internationale de l'économie sociale: Recma*, n° 308: 56.
- Cazals, Marie de. 2010. « La saisine du Conseil économique, social et environnemental par voie de pétition citoyenne : gage d'une Ve République « plus démocratique » ? » *Revue française de droit constitutionnel* 82 (2): 289.
- Cervone, Frank. 2011. « Understanding Agile Project Management Methods Using Scrum ». *OCLC Systems & Services: International Digital Library Perspectives* 27 (1): 18-22.
- Chamberlain, Stephanie, Helen Sharp, et Neil Maiden. 2006. « Towards a Framework for Integrating Agile Development and User-Centred Design ». In *Extreme Programming and Agile Processes in Software Engineering*, Springer Berlin Heidelberg, 4044:143-53.
- Clement, Andrew. 1994. « Computing at work: empowering action by "low-level users" ». *Communications of the ACM* 37 (1): 52-ff.
- Cohn, Mike. 2004. *User stories applied: for agile software development*. Addison-Wesley signature series. Addison-Wesley.
- Colletis, Gabriel. 2014. « Le capitalisme cognitif à l'ère du numérique ». *Mondes sociaux*, 2014.

D

- Darses, Françoise. 2004. « Chapitre premier. La conception participative : vers une théorie de la conception centrée sur l'établissement d'une intelligence mutuelle ». In *Le consommateur au coeur de l'innovation*. Paris: CNRS Editions.
- Darses, Françoise. 2009. « Résolution collective des problèmes de conception ». *Le travail humain* 72 (1): 43.
- Darses, Françoise, Françoise Détienne, et Willemien Visser. 2001. « A Method for Analysing Collective Design Processes ». 4258. Rapport INRIA.
- Dearden, Andy, et Haider Rizvi. 2008. « Adapting Participatory and Agile Software Methods to Participatory Rural Development ». In *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008*, 221–225. USA.
- Dell'Era, Claudio, et Paolo Landoni. 2014. « Living Lab: A Methodology between User-Centred Design and Participatory Design: Living Lab ». *Creativity and Innovation Management* 23 (2): 137-54.
- Demorgon, Jacques. 1987. « Approche de l'altérité et communication non-verbale dans la rencontre internationale: Trois exemples. Le photo-langage - le placement symbolique – le film en V.O. » *Études de communication*, n° 9: 59-68.
- Deneulin, Séverine, et Nicholas Townsend. 2007. « Public Goods, Global Public Goods and the Common Good ». *International Journal of Social Economics* 34 (1/2): 19-36.
- Desreumaux, Alain, et Jean-Pierre Brechet. 2013. « L'entreprise comme bien commun ». *RIMHE Rev. Interdiscip. Manag. Hommes Entrep* 7 (3): 77.

Dupuy-Chessa, Sophie, Benoit Combemale, Marie-Pierre Gervais, Thierry Nodenot, Xavier Le Pallec, et Laurent Wouters. 2014. « Vers une approche centrée humain pour la définition de langages de modélisation graphiques ». In *Inforsid 2014*. Lyon.

Dupuy-Chessa, Sophie, Nadine Mandran, Guillaume Godet-Bar, et Dominique Rieu. 2011. « A Case Study for Improving a Collaborative Design Process ». In *Engineering Methods in the Service-Oriented Context*, Springer Berlin Heidelberg, 351:97-101.

E

Erickson, John, Kalle Lyytinen, et Keng Siau. 2005. « Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research ». *Journal of Database Management* 16 (4): 88-100.

Escamilla, José. 1993. « Shood : un modele meta-circulaire de representation de connaissances ». INP de Grenoble.

Espana, Sergio, Niels Bik, et Sietse Overbeeke. 2019. « Model-driven engineering support for social and environmental accounting ». In *2019 13th International Conference on Research Challenges in Information Science (RCIS)*. Bruxelles, Belgique.

F

Feenberg, Andrew. 2004. *(Re)penser la technique: vers une technologie démocratique*. Paris: Découverte / M.A.U.S.S.

Ferrario, Maria Angela, Will Simm, Peter Newman, Stephen Forshaw, et Jon Whittle. 2014. « Software Engineering for "Social Good": Integrating Action Research, Participatory Design, and Agile Development ». In *Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014*, 520-23. Hyderabad, India: ACM Press.

Fieux, Etienne. 2018. « Quantifier, c'est déjà décider ». In *2ème édition de l'atelier Système d'information et Démocratie - Inforsid*. Nantes.

Floridi, Luciano. 2010. *The Cambridge Handbook of Information and Computer Ethics*. Cambridge University Press.

Fondement, Frédéric, et Thomas Baar. 2005. « Making Metamodels Aware of Concrete Syntax ». In *Model Driven Architecture – Foundations and Applications*, Springer Berlin Heidelberg, 3748:190-204. Berlin, Heidelberg.

Fresso, Françoise. 2019. « La démocratie, en danger, doit se réinventer ». *Le Monde*, 11 mars 2019.

Fricker, Samuel. 2010. « Requirements Value Chains: Stakeholder Management and Requirements Engineering in Software Ecosystems ». In *Requirements Engineering: Foundation for Software Quality*, Springer Berlin Heidelberg, 6182:60-66.

Friedman, Batya, Peter H. Kahn, Alan Borning, et Alina Huldtgren. 2013. « Value Sensitive Design and Information Systems ». In *Early engagement and new technologies: Opening up the laboratory*, 16:55-95. Springer Netherlands.

Front, Agnès, Dominique Rieu, Ornela Cela, et Fatemeh Movahedian. 2017. « Les méthodes d'évolution continue au sein des organisations : le cadre As-Is/As-If ». In *Actes du XXXVème Congrès INFORSID*, Toulouse, France, May 30 - June 2, 2017, 311–326.

G

- Gaborieau, David. 2012. « Le nez dans le micro ». Répercussions du travail sous commande vocale dans les entrepôts de la grande distribution alimentaire ». *La Nouvelle Revue du Travail*, n° 1.
- Gori, Roland, et Marie-José Del Volgo. 2009. « De la société de la norme à une conception managériale du soin ». *Connexions* 91 (1): 123.
- Gourgues, Guillaume, et Julien O'miel. 2019. « Qui a peur de l'initiative citoyenne ? » *Le Monde Diplomatique*, 2019, Février 2019 édition.
- Greenbaum, Joan M. 1991. *Design at work: cooperative design of computer systems*. L. Erlbaum Associates. Hillsdale, N.J.
- Grundstein M., Rosenthal-Sabroux C., Arduin P-E. (2013), « Evaluer la prise en compte des connaissances tacites dans un système d'information ». *Vers un système d'information et de connaissance, Ingénierie des systèmes d'information* 18 (3), 121-148.

H-I

- Harmsen, Frank, Sjaak Brinkkemper, et J L. Han Oei. 1994. « Situational method engineering for information system project approaches ». In *IFIP Transactions A: Computer Science and Technology*, 169-94.
- Hess, Charlotte, et Elinor Ostrom. 2007. *Understanding knowledge as a commons: from theory to practice*. Cambridge, Mass: MIT Press.
- Highsmith, J., et A. Cockburn. 2001. « Agile software development: the business of innovation ». *Computer* 34 (9): 120-27.
- Hug, Charlotte. 2009. « Method, models and tool for information systems engineering process metamodeling ». Thèse de doctorat en informatique, Université Joseph-Fourier - Grenoble I.
- IEEE Computer Society. 1996. *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*. Institute of Electrical and Electronics Engineers. New York.

J-K

- Jardat, Rémi. 2012. « De la démocratie en entreprise. Quelques résultats empiriques et propositions théoriques ». *Revue française de gestion* 38 (228-229): 167-84.
- Jardat, Rémi. 2019. « Une organisation démocratique est-elle compétitive ? Compte rendu d'une recherche en sciences de gestion ». In *3ème édition de l'Atelier Système d'information et Démocratie - Inforsid*. Paris.
- Kanstrup, Anne Marie, et Pernille Bertelsen. 2016. *User Innovation Management*. Aalborg University Press.
- Kautz, Karlheinz. 2011. « Investigating the Design Process: Participatory Design in Agile Software Development ». *Information Technology & People* 24 (3): 217-35.
- Kensing, Finn, et Jeanette Blomberg. 1998. « Participatory Design: Issues and Concerns ». *Computer Supported Cooperative Work (CSCW)* 7 (3-4): 167-85.

- Klinkenberg, Jean-Marie. 2015. « La langue dans la cité. Vivre et penser l'équité culturelle ». *Les impressions nouvelles*, 313.
- Kotonya, Gerald, et Ian Sommerville. 1996. « Requirements Engineering with Viewpoints ». *Software Engineering Journal* 11 (1): 5.
- Krogstie, John. 2003. « Evaluating UML Using a Generic Quality Framework ». In *UML and the unified process*, IGI Publishing Hershey, 1-22.
- Kujala, Sari. 2003. « User Involvement: A Review of the Benefits and Challenges ». *Behaviour & Information Technology* 22 (1): 1-16.
- Kushniruk, André, et Christian Nohr. 2016. « Participatory Design, User Involvement and Health IT Evaluation ». *Studies in Health Technology and Informatics*, 139–151.

L

- Lamsweerde, A. van. 2009. *Requirements engineering: from system goals to UML models to software specifications*. Chichester, England ; Hoboken, NJ: John Wiley.
- Lamsweerde, Axel van. 2000. « Goal-oriented requirements engineering: a guided tour ». In *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, 249-62.
- Lauche, Kristina. 2002. « Facilitating Creativity and Shared Understanding in Design Teams ». In *Proceedings of DESIGN 2002, the 7th International Design Conference, Dubrovnik*, 823-28.
- Le Moigne, Jean Louis. 1994. *La théorie du système général: théorie de la modélisation*. Paris: Les Éditions G. Crès et Cie.
- Le Moigne, Jean-louis. 1973. *Les systèmes d'information dans l'organisation*. Presses universitaires de France. Systèmes-décisions. Paris.
- Le Moigne, Jean-Luc. 1994. « L'information forme l'organisation qui la forme ». *Sciences et Société*, n° 33.
- Leclercq-Vandelannoitte, Aurélie. 2017. « Victime ou coupable ? Repenser le rôle du contrôlé dans la relation entre contrôle, information et technologies de l'information ». *Systèmes d'information & management* 22 (2): 49.
- Le Rider, Brigitte, et Dorothea Levy-Hillerich. 2004. « Der Metaplan im Deutsch als Fremdsprache Unterricht: ein Beispiel ». *Recherche et pratiques pédagogiques en langues de spécialité - Cahiers de l'APLIUT*, n° Vol. XXIII N° 1: 140-48.
- Lin, Angela, et Leiser Silva. 2005. « The Social and Political Construction of Technological Frames ». *European Journal of Information Systems* 14 (1): 49-59.
- Lukes, Steven. 2004. *Power: a radical view*. 2nd ed. Houndmills, Basingstoke, Hampshire : New York: Palgrave Macmillan.
- Lyons, Michal, Carin Smuts, et Anthea Stephens. 2001. « Participation, Empowerment and Sustainability: (How) Do the Links Work? » *Urban Studies* 38 (8): 1233-51.

M

- Mahaux, Martin, et Annick Castiaux. 2015. « Participation and Open Innovation for Sustainable Software Engineering ». In *Green in Software Engineering*, Springer International Publishing, 301-23. Cham.
- Mahaux, Martin, Lemai Nguyen, Oily Gotel, Luisa Mich, Alistair Mavin, et Klaus Schmid. 2013. « Collaborative creativity in requirements engineering: Analysis and practical advice ». In *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, 1-10.
- Mallard, Alexandre. 2014. « Métamorphoses d'une question scientifique: Trente ans de recherches sur l'inscription des TIC dans les univers productifs ». *Réseaux* n° 184-185 (2): 33.
- Maugeri, Salvatore. 2007a. « Travail, dispositif de gestion et domination ». In *Xème journées internationales de sociologie du travail*.
- McConnell, Steve. 2006. *Rapid Development: Taming Wild Software Schedules*. Microsoft Press. Redmond, Wash.
- Messaoudi, Sofiane, Régis Meissonier, et Claudio Vitari. 2019. « Contribution du shadow IT à la construction de la légitimité de l'acteur métier ». In *Transformation digitale*. El Jadida, Maroc.
- Mélèse, Jacques. 1979. *Approches systémiques des organisations: vers l'entreprise à complexité humaine*. Paris: Éditions d'organisation.
- Milne, Alastair, et Neil Maiden. 2012. « Power and Politics in Requirements Engineering: Embracing the Dark Side? » *Requirements Engineering* 17 (2): 83-98.
- Moody, Daniel. 2009. « The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering ». *IEEE Transactions on Software Engineering* 35 (6): 756-79.
- Morand, Bernard. 2007. « Le logiciel, sujet et objet de la norme ». In *Représentations, modèles et normes pour l'entreprise*. Paris: LGDJ : 2RJ.

N-O

- Nassar, Mahmoud. 2003. « VUML: a viewpoint oriented UML extension ». In *18th IEEE International Conference on Automated Software Engineering*, 373-376.
- Ogien, Albert, et Sandra Laugier. 2011. *Pourquoi désobéir en démocratie?* Paris: La Découverte.
- Ostrom, Elinor. 2007. « Challenges and Growth : The Development of the Interdisciplinary Field of Institutional Analysis ». *Journal of Institutional Economics* 3 (03).

P

- Palterini, Lucas, et Prévost-Carpentier. 2019. *Le travail en mouvement*. Mines-ParisTech. Paris.
- Peugeot, Valérie. 2011. *Libres savoirs: les biens communs de la connaissance: produire collectivement, partager et diffuser les connaissances au XXIe siècle*. Caen: C&F éditions.
- Pichler, Roman. 2010. *Agile product management with Scrum: creating products that customers love*. Addison-Wesley. The Addison-Wesley signature series. Upper Saddle River, NJ.

Piketty, Thomas. 2019. *Capital et idéologie*. Paris: Seuil.

Punter, Teade, et Karel Lemmen. 1996. « The MEMA-Model: Towards a New Approach for Method Engineering ». *Information and Software Technology* 38 (4): 295-305.

R

Raffenne, Marc. 2007. « Vers une sociologie de la logistique ? Tentative de réconciliation de la sociologie du travail avec son objet ». In *Le Travail à l'épreuve des paradigmes sociologiques*, Octares. Toulouse.

Ralyte, Jolita. 2001. « Vue stratégique sur l'ingénierie des méthodes ». In *Actes du XIXème Congrès INFORSID*, 43-66.

Ralyte, Jolita. 2006. « Méthodes et Ingénierie de Méthodes ». In *Encyclopédie de l'informatique et des systèmes d'information*, 1437-47.

Ráth, István, András Ökrös, et Dániel Varró. 2010. « Synchronization of Abstract and Concrete Syntax in Domain-Specific Modeling Languages: By Mapping Models and Live Transformations ». *Software & Systems Modeling* 9 (4): 453-71.

Rentrop, Christophe, et Stephan Zimmermann. 2012. « Shadow IT - Management and control of unofficial IT ». In *ICDS 2012 : The Sixth International Conference of Digital Society*.

Ricœur, Paul. 1985. « Éthique et politique ». *Autres Temps. Les cahiers du christianisme social* 5 (1): 58-70.

Riet, R.P. van de. 1989. « MOKUM: An Object-Oriented Active Knowledge Base System ». *Data & Knowledge Engineering* 4 (1): 21-42.

Rieu, Dominique, Toan Gia Nguyen, et Annie Culet. 1991. « Instanciation multiple et classification d'objet ». RR-1480. INRIA.

Rolland, C., N. Prakash, et A. Benjamen. 1999. « A Multi-Model View of Process Modelling ». *Requirements Engineering* 4 (4): 169-87.

Rolland, Colette. 2005. « L'ingénierie des méthodes : une visite guidée ». *e-revue en Technologies de l'Information*.

Ross, D. T., et K. E. Schoman Jr. 1979. *Classics in Software Engineering*. Yourdon Press Upper Saddle River.

S

Salais, Robert. 2010. « La donnée n'est pas un donné: Pour une analyse critique de l'évaluation chiffrée de la performance ». *Revue française d'administration publique* 135 (3): 497.

Salles, Maryse. 2007. *Représentations, modèles et normes pour l'entreprise*. Paris: LGDJ : 2RJ.

Salles, Maryse. 2013. « Ingénierie de méthodes d'ingénierie des exigences pour l'aide à la décision ». Mémoire HDR en informatique, Université Toulouse 1.

Salles, Maryse. 2015. « Responsabilité économique et sociale des concepteurs de systèmes d'information : contribution à une éthique appliquée ». *Innovations* 46 (1): 197.

- Salles, Maryse, et Gabriel Colletis. 2007. « TIC - systèmes d'information et choix de représentation du réel ». In *EUTIC 2007 - Enjeux et usages des TIC*. Athènes, Grèce.
- Salles, Maryse, et Gabriel Colletis. 2013. « Déconstruire la doxa dominante, construire une pensée politique alternative. Du lien entre les représentations, les principes et les normes ». *LoSguardo* Vol. 2013 (III) (N. 13): 391-414.
- Sanders, Elizabeth B.-N., et Pieter Jan Stappers. 2008. « Co-Creation and the New Landscapes of Design ». *CoDesign* 4 (1): 5-18.
- Schön, Eva-Maria, Jörg Thomaschewski, et María José Escalona. 2017. « Agile Requirements Engineering: A Systematic Literature Review ». *Computer Standards & Interfaces* 49: 79-91.
- Schwaber, Ken, et Mike Beedle. 2002. *Agile software development with Scrum*. Series in agile software development. Prentice Hall.
- Schwartz, Lou, Laurent Vergnol, Guillaume Gronier, Alain Vagner, Thomas Altenburger, et Sophie Battisti. 2009. « Comment Concilier Agilité et Conception Centrée Utilisateurs Dans Un Projet de Développement? » In *Proceedings of the 21st International Conference of the Association Francophone d'Interaction Homme-Machine*, 337. Grenoble.
- Sharp, H., A. Finkelstein, et G. Galal. 1999. « Stakeholder identification in the requirements engineering process ». In *Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99*, 387-91.
- Silva da Silva, Tiago, Angela Martin, Frank Maurer, et Milene Silveira. 2011. « User-Centered Design and Agile Methods: A Systematic Review ». In *2011 AGILE Conference*, 77-86. Salt Lake City, UT, USA: IEEE.
- Simon, Herbert A. 2004. *Les sciences de l'artificiel*. Paris: Gallimard.
- Simonsen, Jesper. 2013. *Routledge International Handbook of Participatory Design*. Édité par Toni Robertson. Routledge international handbooks. London.
- Sommerville, Ian, et Pete Sawyer. 1997. « Viewpoints: Principles, Problems and a Practical Approach to Requirements Engineering ». *Annals of Software Engineering* 3 (0): 101-30.
- Sverrisdottir, Hrafnhildur Sif, Helgi Thor Ingason, et Haukur Ingi Jonasson. 2014. « The Role of the Product Owner in Scrum-Comparison between Theory and Practices ». *Procedia - Social and Behavioral Sciences* 119: 257-67.

T-V

- Tiffon, Guillaume. 2013. « La mise au travail des clients ». Paris: Économica.
- Tissier, Dominique. 2018. *Management situationnel: vers l'autonomie et la responsabilisation*. Eyrolles.
- Vickoff, Jean-Pierre. 1996. *RAD: développement rapide d'application*. Paris: Simon & Schuster Macmillan.
- Vickoff, Jean-Pierre. 2003. *Systèmes d'information et processus agiles*. Paris: Lavoisier : Hermès Science.

Vilela, Rodolfo Andrade De Gouveia, Marco Antônio Pereira Querol, Laura Seppänen, Francisco De Paula Antunes Lima, Renata Wey Berti Mendes, Manoela Gomes Reis Lopes, Ildeberto M. Almeida, et José Marçal Jackson Filho. 2015. « Work Ergonomic Analysis and Change Laboratory: Similarities and Complementarities Between Interventionist Methods ». Report.

Von Hippel, Eric. 1995. *The Sources of Innovation*. New York Oxford: Oxford University Press.

W-Z

Walsham, Geoff. 2012. « Are We Making a Better World with ICTs? Reflections on a Future Agenda for the IS Field ». *Journal of Information Technology* 27 (2): 87-93.

Wanderley, Fernando, Nicolas Belloir, Jean-Michel Bruel, Nabil Hameurlin, et Joao Araujo. 2014. « Des buts à la modélisation système : une approche de modélisation des exigences centrée utilisateurs ». In *Actes du congrès Inforsid*. Lyon.

Wieggers, Karl Eugene. 2006. *More about software requirements: thorny issues and practical advice*. Microsoft Press. Best practices. Redmond, WA.

Wood, Jane, et Denise Silver. 1995. *Joint application development*. 2nd ed. New York: Wiley.

Zarifian, Philippe. 1996. *Travail et communication: essai sociologique sur le travail dans la grande entreprise industrielle*. 1. éd. Sociologie d'aujourd'hui. Paris: Presses Univ. de France.

Zehnder, Eloïse, Valérie Saint-Dizier De Almeida, Jérôme Dinet, Guillaume Gronier, et Nicolas Mayer. 2017. « Design of visual syntax based on cognitive effectiveness for modeling language ». In *29ème conférence francophone sur l'Interaction Homme-Machine*. Poitiers, France: AFIHM.

Webographie

Agile Manifesto. 2001. « *Agile manifesto* ».

<http://agilemanifesto.org/iso/fr/manifesto.html>.

Date de dernière consultation : 26 juin 2019

Chejfec, Thomas. 2012. « *Résultats de l'enquête sur le phénomène de Shadow IT* ».

<https://i8080.files.wordpress.com/2012/12/rc3a9sultats-enquete.pdf>.

Date de dernière consultation : 25 juin 2017

CIGREF. 2018. « *Ethique et numérique - un référentiel pratique pour les acteurs du numérique* ».

<https://www.cigref.fr/wp/wp-content/uploads/2018/10/Cigref-Syntec-Numerique-Referentiel-pratique-Ethique-et-Numerique-2018.pdf>.

Date de dernière consultation : 10 septembre 2019

Dupuy, François. 2015. « *La multiplication des chefs de projet est une catastrophe managériale* ».

<https://www.usinenouvelle.com/article/la-multiplication-des-chefs-de-projet-est-une-catastrophe-manageriale-majeure-affirme-le-sociologue-francois-dupuy.N307730>.

Date de dernière consultation : 6 février 2016

Jeffries, Ron. 2018. « *Developers Should Abandon Agile* ».

<https://ronjeffries.com/articles/018-01ff/abandon-1/>.

Date de dernière consultation : 9 février 2019

Le Crosnier, Hervé. 2013. « *Biens communs : de la nature à la connaissance* ».

<https://www.youtube.com/watch?v=RCaYuINVCpQ>.

Date de dernière consultation : 10 juin 2016

Ministère du travail. 2018. « *Prévention des risques pour la santé au travail* ».

<https://travail-emploi.gouv.fr/sante-au-travail/prevention-des-risques-pour-la-sante-au-travail/article/risques-psychosociaux>.

Date de dernière consultation : 15 janvier 2019

Patton, Jeff. 2008. « *Don't Know What I Want, But I Know How to Get It* ».

https://www.jpattonassociates.com/dont_know_what_i_want/.

Date de dernière consultation : 25 septembre 2019

Rambaud, Alexandre. 2018. « *Introduction aux enjeux de la comptabilité socio-environnementale des organisations* ».

https://www.canalu.tv/video/canal_ued/introduction_aux_enjeux_de_la_comptabilite_socio_environnementale_des_organisations.40117.

Date de dernière consultation : 10 juillet 2019

Standish Group. 2015. « *Standish Group Chaos Report* ».

https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf.

Date de dernière consultation : 25 septembre 2019

Stiegler, Bernard, et Laurent Ricard. 2014. « *Révolution numérique, fablab et économie de la contribution* ».

<https://www.youtube.com/watch?v=xsWs8tSAGds>.

Date de dernière consultation : 28 juillet 2016

Sutherland, Jeff, et Ken Schwaber. 2011. « *The Scrum Papers : Nut, Bolts, and Origins of an Agile Framework* ».

<https://klevas.mif.vu.lt/~adamonis/pkp/1415p/lit/ScrumPapers.pdf>

Date de dernière consultation : 26 septembre 2019

Villani, Cédric. 2018. « *Donner un sens à l'intelligence artificielle - pour une stratégie nationale et européenne* ». AI For Humanity.

https://www.aiforhumanity.fr/pdfs/9782111457089_Rapport_Villani_accessible.pdf

Date de dernière consultation : 30 septembre 2019

Résumé

Nos sociétés font face aujourd'hui à un enjeu majeur de démocratie. Les organisations, au cœur de la société, peuvent prendre leur part dans le traitement de cette problématique. Cela passe par une analyse introspective de chaque organisation quant aux possibilités de participation à la prise de décision collective qu'elle propose, mais aussi quant à la place qu'occupent les outils de gouvernementalité dans leur fonctionnement. Afin de soutenir la démocratie des organisations, les systèmes d'information peuvent jouer un rôle majeur à condition que, dès leur conception, la question démocratique soit posée. Pour cela, ils doivent être construits grâce à des méthodes de conception participatives qui interrogent la question des valeurs, des normes, des visions du monde que soutiennent ces systèmes. Ces systèmes d'information doivent en outre être la représentation d'une pluralité de points de vue, et par conséquent ne pas contraindre chacun à se conformer à un point de vue dominant.

Notre contribution : DEMOS, est une méthode de conception participative de systèmes d'information soutenant la démocratie des organisations. Elle repose sur des principes structurants, qui ont constitué le fil conducteur de sa construction. La méthode intègre elle-même des caractéristiques démocratiques au travers de son modèle de produit et de son modèle de processus. Elle permet également de créer des systèmes d'information plus démocratiques qui intègrent une diversité de points de vue mis en cohérence. Cette méthode est formalisée, détaillée, la rendant ainsi reproductible et adaptable à tout projet de construction de système d'information dans une organisation.

Mots clés : système d'information, démocratie, ingénierie des méthodes, démarches participatives, point de vue, modélisation conceptuelle