

**SYSTÈME D'ÉVALUATION ET DE GESTION  
DES RISQUES D'INONDATION EN MILIEU FLUVIAL**

**PROJET SEGRI**

**Spécifications – Modeleur2**

*Rapport de recherche No R-720-b2*

*Janvier 2005*

**Système d'Évaluation et de Gestion  
des Risques d'Inondation en milieu fluvial**

**Projet SEGRI**

**Spécifications – Modeleur2**

**Présenté au**

**Fonds des Priorités Gouvernementales en Science et en  
Technologie – volet Environnement (FPGST-E)**

**18 janvier 2005**

## Équipe de réalisation

### Institut National de la Recherche Scientifique – Eau, Terre et Environnement

Yves Secretan  
Michel Leclerc  
Eric Larouche  
Paul Boudreau

Professeur, PhD  
Professeur, PhD  
Ingénieur en informatique  
Agent de recherche, M. Sc.-Eau

Pour les fins de citation : **Secretan Y., Larouche E. & coll. (2004).**

Système d'Évaluation et de Gestion des Risques d'Inondation en milieu fluvial (SEGRI) :  
Spécifications – Modeleur2. Québec, INRS-Eau, Terre & Environnement. 126 pages.  
(INRS-Eau, Terre & Environnement, rapport de recherche 720 b2)

Pour: Fonds des Priorités Gouvernementales en Science et en Technologie – volet  
Environnement (FPGST-E).

©INRS-Eau, Terre & Environnement, 2004  
ISBN : 2-89146-525-3

## TABLE DES MATIÈRES

INTRODUCTION.....	1
1.1    SOMMAIRE .....	1
1.2    SYSTÈME EXISTANT .....	1
1.3    CONCEPTS CLÉS .....	2
1.4    STRUCTURE DU DOCUMENT.....	2
<b>2    DESCRIPTION DES ZONES DE L'INTERFACE UTILISATEUR .....</b>	<b>5</b>
2.1    ORGANISATION DE LA FENÊTRE PRINCIPALE .....	5
2.1.1    Zone de contrôle .....	5
2.1.2    Zone des fenêtres.....	6
2.1.3    Zone d'Information .....	7
2.2    BARRE DE MENUS.....	7
2.2.1    Menu Fichier.....	7
2.2.2    Menu Édition.....	8
2.2.3    Menu Affichage .....	8
2.3    BARRE D'OUTILS.....	8
2.3.1    Barre d'outils standard.....	8
2.3.2    Barre d'outils spécialisée .....	9
2.3.3    Barre d'outils partagée.....	9
<b>3    DESCRIPTION FONCTIONNELLE .....</b>	<b>11</b>
<b>4    ACTIVITÉS .....</b>	<b>13</b>
4.1    ACTIVITÉ : GESTION DES PROJETS .....	13
4.1.1    Nouveau projet.....	13
4.1.2    Ouvrir projet.....	13
4.1.3    Fermer projet.....	14
4.1.4    Enregistrer projet.....	14
4.1.5    Enregistrer projet sous.....	14
4.1.6    Modifier les paramètres du projet .....	14
4.2    GESTION DES DONNÉES .....	15
4.2.1    Créer des liens externes vers des données.....	15
4.2.2    Consulter / modifier les métadonnées d'une donnée .....	15
4.2.2.1    Profil de l'utilisateur .....	16
4.2.2.2    Informations générales.....	16
4.2.2.3    Informations de distribution.....	16
4.2.3    Consulter / trier les données d'un projet .....	16
4.2.4    Raffiner la liste des données par critères .....	17
4.2.5    Consulter les dépendances d'une donnée .....	17
4.2.5.1    Type : parent dépend de l'enfant .....	17
4.2.5.2    Type : parent est à la source de l'enfant .....	18
4.2.6    Supprimer une donnée .....	18
4.2.7    Importer / Exporter des données.....	19
4.2.7.1    À partir / vers des fichiers.....	19

4.2.7.2	À partir / vers une base de données .....	19
4.3	ACTIVITÉ : VALIDATION DES DONNÉES DE TERRAIN .....	20
4.3.1	<i>Gestion</i> .....	22
4.3.1.1	Nouveau .....	22
4.3.1.2	Ouvrir .....	22
4.3.1.3	Fermer .....	22
4.3.1.4	Enregistrer .....	22
4.3.1.5	Enregistrer sous .....	23
4.3.2	<i>Semi de points</i> .....	23
4.3.2.1	Publier des semis de points en champ .....	23
4.3.2.2	Combiner des semis de points .....	23
4.3.2.3	Visualiser l'information de semis de points .....	23
4.3.3	<i>Points</i> .....	24
4.3.3.1	Sélectionner des points .....	24
4.3.3.2	Créer des points .....	25
4.3.3.3	Éditer l'information portée par les points .....	25
4.3.3.4	Activer / désactiver des points .....	25
4.3.3.5	Déplacer des points .....	25
4.3.3.6	Effacer des points .....	25
4.3.4	<i>Filtrage</i> .....	26
4.3.4.1	Filtrer les points à désactiver .....	26
4.3.4.2	Filtres primaires .....	26
4.3.4.3	Filtres secondaires .....	27
4.4	ACTIVITÉ : MODÉLISATION NUMÉRIQUE DE TERRAIN .....	27
4.4.1	<i>Interface usager</i> .....	29
4.4.2	<i>Gestion d'un MNT</i> .....	30
4.4.2.1	Nouveau .....	30
4.4.2.2	Ouvrir .....	30
4.4.2.3	Fermer .....	31
4.4.2.4	Enregistrer .....	31
4.4.2.5	Enregistrer sous .....	31
4.4.3	<i>Fonctionnalités du MNT</i> .....	31
4.4.3.1	Ajouter une nouvelle partition .....	31
4.4.3.2	Ajouter une partition existante .....	31
4.4.3.5	Retirer une partition .....	31
4.4.3.6	Ajouter une nouvelle couche de données .....	32
4.4.3.7	Ajouter une couche de données existante .....	32
4.4.3.8	Retirer une couche de données .....	32
4.4.3.9	Afficher les limites de jeux de données .....	32
4.4.3.10	Éditer une couche de données .....	32
4.4.3.11	Copier et coller une couche de données .....	33
4.4.3.12	Modifier la couleur des couches .....	33
4.4.3.13	Modifier le nom d'une couche ou d'une partition .....	33
4.4.3.14	Modifier la priorité d'une couche de données .....	33
4.4.3.15	Annexer des commentaires .....	33
4.4.3.16	Générer le squelette du MNT .....	33

4.4.4	<i>Partitions génériques</i> .....	33
4.4.4.1	Partition générique sans interpolation.....	34
4.4.4.2	Partition générique avec interpolation.....	34
4.4.4.3	Définir la partition générique.....	35
4.4.4.4	Construire la partition générique.....	35
4.4.5	<i>Édition de sous-domaines</i> .....	36
4.4.5.1	Créer un sous-domaine.....	36
4.4.5.2	Sélectionner un sous-domaine.....	36
4.4.5.3	Effacer un sous-domaine.....	36
4.4.5.4	Fusionner deux sous-domaine.....	36
4.4.5.5	Enlever un sommet d'un sous-domaine.....	36
4.4.5.6	Scinder une ligne d'un sous-domaine.....	36
4.5	ACTIVITÉ : CONSTRUCTION D'UN MAILLAGE.....	36
4.5.1	<i>Théorie relative aux partitions de maillage</i> .....	36
4.5.2	<i>Paramètres de maillage</i> .....	38
4.5.2.1	Taille des mailles.....	38
4.5.2.2	Champ d'erreur.....	38
4.5.3	<i>Ajout de nouveaux mailleurs</i> .....	39
4.5.4	<i>Interface usager pour la gestion d'une partition de maillage</i> .....	39
4.5.5	<i>Gestion d'une partition de maillage</i> .....	40
4.5.5.1	Nouveau.....	40
4.5.5.2	Ouvrir.....	40
4.5.5.3	Fermer.....	40
4.5.5.4	Enregistrer.....	41
4.5.5.5	Enregistrer sous.....	41
4.5.6	<i>Fonctionnalités des partitions de maillage</i> .....	41
4.5.6.1	Ajouter une nouvelle couche de maillage.....	41
4.5.6.2	Ajouter une couche de maillage existante.....	41
4.5.6.3	Retirer une couche de maillage.....	41
4.5.6.4	Copier et coller une couche de maillage.....	41
4.5.6.5	Modifier le nom d'une couche de maillage.....	42
4.5.6.6	Modifier la priorité d'une couche de maillage.....	42
4.5.6.7	Modifier la couleur d'affichage des couches de maillage.....	42
4.5.6.8	Afficher le squelette d'un MNT.....	42
4.5.6.9	Éditer une couche de maillage.....	42
4.5.6.10	Mailler une couche.....	42
4.5.6.11	Voir le maillage précédent (maillage invalidé).....	43
4.5.6.12	Modifier les paramètres d'une partition.....	43
4.5.6.13	Modifier les paramètres d'une couche.....	43
4.5.6.14	Trouver le niveau de tolérance optimal.....	43
4.5.6.15	Obtenir de l'information sur le maillage généré.....	43
4.5.6.16	Générer le maillage final.....	44
4.6	PARTITION DE CONDITIONS AUX LIMITES.....	44
4.6.1	<i>Interface usager</i> .....	45
4.6.1.1	Partition de conditions aux limites.....	45
4.6.2	<i>Gestion des partitions de conditions aux limites</i> .....	46

4.6.2.1	Nouveau .....	46
4.6.2.2	Ouvrir .....	47
4.6.2.3	Fermer .....	47
4.6.2.4	Enregistrer.....	47
4.6.2.5	Enregistrer sous.....	47
4.6.3	<i>Fonctionnalités reliées aux partitions de conditions aux limites</i> .....	47
4.6.3.1	Ajouter une nouvelle condition.....	47
4.6.3.2	Ajouter une condition existante .....	48
4.6.3.3	Assigner une condition .....	48
4.6.3.4	Modifier une condition .....	48
4.6.3.5	Effacer une condition .....	48
4.7	ACTIVITÉ : ASSEMBLAGE DES DONNÉES À PARTIR DU MAILLAGE .....	48
4.7.1	<i>Interface usager</i> .....	49
4.8	ACTIVITÉ : SIMULATION .....	49
4.8.1	<i>Simulation</i> .....	49
4.8.2	<i>Simulateur</i> .....	50
4.8.3	<i>Interface usager</i> .....	51
4.8.3.1	Simulation .....	51
4.8.4	<i>Gestion d'une simulation</i> .....	53
4.8.4.1	Nouveau .....	53
4.8.4.2	Ouvrir .....	53
4.8.4.3	Fermer .....	53
4.8.4.4	Enregistrer.....	53
4.8.4.5	Enregistrer sous.....	53
4.8.5	<i>Fonctionnalités reliées aux simulations</i> .....	53
4.8.5.1	Créer un calcul .....	53
4.8.5.2	Modifier les paramètres d'un calcul .....	54
4.8.5.3	Exécuter un calcul.....	54
4.8.5.4	Importer les résultats d'un calcul .....	54
4.8.5.5	Interrompre un calcul .....	54
4.8.5.6	Renommer un calcul .....	55
4.8.5.7	Naviguer l'historique de la simulation.....	55
4.9	PUBLICATION DES RÉSULTATS (HABILLAGE D'ÉCRAN).....	55
4.9.1	<i>Gestion des habillages d'écran</i> .....	55
4.9.1.1	Nouveau .....	55
4.9.1.2	Ouvrir .....	55
4.9.1.3	Enregistrer.....	55
4.9.1.4	Enregistrer sous.....	55
4.9.2	<i>Fonctionnalités reliées aux habillages d'écran</i> .....	56
4.9.2.1	Ajouter .....	56
4.9.2.2	Effacer .....	56
4.9.2.3	Déplacer .....	56
4.9.2.4	Copier / coller .....	56
5	OUTILS D'ANALYSE .....	57
5.1	L'AFFICHAGE .....	57

5.1.1	<i>Interface usager</i> .....	58
5.1.2	<i>Gestion de l’affichage</i> .....	59
5.1.2.1	Espace de représentation .....	59
5.1.2.2	Plan .....	59
5.1.2.3	Objet graphique .....	60
5.1.3	<i>Fonctionnalités de la liste d’affichage</i> .....	61
5.1.3.1	Modifier un objet graphique .....	61
5.1.3.2	Retirer un objet graphique .....	61
5.1.3.3	Copier / coller / couper un objet graphique .....	61
5.1.3.4	Déplacer un objet graphique .....	61
5.1.4	<i>Construire des échelles de couleur (légendes)</i> .....	61
5.1.5	<i>Gestion de la caméra</i> .....	61
5.1.5.1	Configurer les paramètres de gestion de caméra .....	61
5.1.5.2	Effectuer une opération de zoom .....	61
5.1.5.3	Effectuer une opération de zoom de façon interactive .....	62
5.1.5.4	Se déplacer avec des translations (haut / bas, gauche / droite) .....	62
5.1.5.5	Se déplacer de façon interactive .....	62
5.1.5.6	Réinitialiser la zone d’affichage .....	62
5.2	LA VISUALISATION .....	62
5.2.1	<i>Visualiser un objet graphique de type maillage</i> .....	62
5.2.2	<i>Visualiser un objet graphique de type point</i> .....	63
5.2.2.1	Source d’où peuvent provenir les points .....	63
5.2.2.2	Information portée par les points .....	64
5.2.3	<i>Visualiser un objet graphique de type géométrie</i> .....	65
5.3	LES SÉRIES .....	65
5.3.1	<i>Théorie sur les séries</i> .....	65
5.3.2	<i>Utilisation des séries</i> .....	68
5.3.3	<i>Construction d’une série</i> .....	68
5.3.4	<i>Gestion d’une série</i> .....	70
5.3.4.1	Nouveau .....	70
5.3.4.2	Ouvrir .....	71
5.3.4.3	Fermer .....	71
5.3.4.4	Enregistrer .....	71
5.3.4.5	Enregistrer sous .....	71
5.3.5	<i>Fonctionnalités reliées aux séries</i> .....	71
5.3.5.1	Créer la structure d’une série .....	71
5.3.5.2	Modifier la structure d’une série .....	71
5.3.5.3	Ajouter un ou des nœuds à une série .....	71
5.3.5.4	Retirer un ou des nœuds à une série .....	72
5.3.5.5	Visualiser la série .....	72
5.3.5.6	Modifier les valeurs des nœuds d’une série .....	72
5.4	LES SCRIPTS ET LES MACROS .....	72
5.4.1	<i>Gestion des scripts globaux</i> .....	75
5.4.1.1	Importer .....	75
5.4.1.2	Exporter .....	75
5.4.1.3	Rechercher .....	75

5.4.1.4	Modifier .....	75
5.4.1.5	Supprimer.....	75
5.4.1.6	Glisser-déposer et raccourcis clavier standards .....	76
5.4.2	<i>Gestion des scripts locaux</i> .....	76
5.4.2.1	Edition – Débogage.....	76
5.4.2.2	Exécution .....	76
5.4.2.3	Raccourcis clavier .....	76
5.4.3	<i>Macros</i> .....	76
5.4.3.1	Commencer l’enregistrement.....	76
5.4.3.2	Terminer l’enregistrement .....	76
5.4.3.3	Pause / reprise de l’enregistrement .....	77
5.4.4	<i>IDE externe - Editeur</i> .....	77
5.4.4.1	Caractéristiques nécessaires pour l’intégration dans Modeleur2.....	77
5.4.4.2	Nouveau .....	77
5.4.4.3	Ouvrir .....	77
5.4.4.4	Editer .....	77
5.4.4.5	Enregistrer.....	77
5.4.4.6	Enregistrer sous.....	77
5.4.4.7	Fermer .....	77
5.4.4.8	Quitter .....	78
5.4.5	<i>IDE externe - Débogueur</i> .....	78
5.4.5.1	Caractéristiques nécessaires pour l’intégration dans Modeleur2.....	78
5.4.5.2	Compiler .....	78
5.4.5.3	Exécuter pas à pas.....	78
5.4.5.4	Visualisation de l’état des variables.....	78
5.4.5.5	Exécuter .....	78
5.4.5.6	Affichage des erreurs .....	78
5.5	LA CALCULATRICE .....	79
5.5.1	<i>Description et rôle</i> .....	79
5.5.1.1	Interface usager.....	80
5.5.2	<i>Fonctionnalités de la calculatrice - Barre d’outils</i> .....	82
5.5.2.1	Charger une ou plusieurs variables .....	82
5.5.2.2	Enregistrer une variable .....	82
5.5.2.3	Importer un module dans la calculatrice .....	82
5.5.2.4	Éditer un module.....	83
5.5.2.5	Rafraîchir un module .....	83
5.6	L’OUTIL DE SONDE .....	83
5.6.1.1	Ajouter un champ.....	83
5.6.1.2	Retirer un champ.....	83
5.6.1.3	Changer le repère .....	84
5.6.1.4	Placer la sonde dans la zone Information .....	84
5.6.1.5	Changer de mode .....	84
5.7	L’OUTIL DE CALCUL D’ISOLIGNES ET D’ISOSURFACES .....	84
5.7.1.1	Calculer des isolignes .....	84
5.7.1.2	Calculer des isosurfaces.....	84
5.7.1.3	Enregistrer des isolignes .....	84

5.7.1.4	Enregistrer des isosurfaces.....	85
5.8	L'OUTIL DE LIGNES .....	86
5.8.1	<i>Théorie</i> .....	86
5.8.2	<i>Construction</i> .....	87
5.8.3	<i>Interface usager : gestion des multipolygones</i> .....	87
5.8.4	<i>Gestion des lignes</i> .....	88
5.8.4.1	Nouveau .....	88
5.8.4.2	Ouvrir.....	88
5.8.4.3	Enregistrer.....	88
5.8.4.4	Enregistrer sous.....	89
5.8.5	<i>Analyse</i> .....	89
5.8.6	<i>Interface usager : analyse des lignes</i> .....	90
5.9	L'OUTIL DE POLYGONES.....	91
5.9.1.1	Théorie .....	91
5.9.1.2	Analyse .....	92
5.10	L'OUTIL DE VUE TABULÉE.....	92
5.11	L'OUTIL DE PROJECTION .....	92
5.12	L'OUTIL DE PROJECTION CARTOGRAPHIQUE.....	93
<b>6</b>	<b>ARCHITECTURE LOGICIELLE .....</b>	<b>95</b>
6.1	BUS D'ÉVÉNEMENTS .....	95
6.1.1	<i>Terminologie de base</i> .....	96
6.1.2	<i>Fonctionnement général</i> .....	96
6.1.2.1	La phase d'initialisation.....	96
6.1.2.2	La phase d'utilisation.....	97
6.1.2.3	La phase de fermeture .....	97
6.1.3	<i>Événements</i> .....	97
6.1.4	<i>Lien de type « direct »</i> .....	97
6.2	BASE DE DONNEES .....	98
6.3	SYSTEME DE VISUALISATION GRAPHIQUE.....	98
6.4	PLUG-IN .....	99
6.4.1	<i>Ajout : partie C++</i> .....	100
6.4.2	<i>Ajout : partie BD</i> .....	100
6.4.3	<i>Ajout : partie GUI</i> .....	101
6.4.3.1	Gestion des menus .....	101
6.4.3.2	Gestion des zones.....	101
6.4.3.3	Gestion des barre d'outils .....	101
6.4.3.4	Gestion des événements .....	101
6.5	LANGAGE DE SCRIPT .....	102
6.5.1	<i>Interface utilisateur (UI)</i> .....	102
6.5.2	<i>Calculatrice</i> .....	102
6.5.3	<i>Macros</i> .....	102
6.6	INSTALLATION VIA LE WEB .....	102
<b>7</b>	<b>CONFIGURATION, AIDE, LANGUE ET DISTRIBUTION .....</b>	<b>103</b>
7.1	AIDE .....	103

7.2	LANGUE.....	103
7.3	DISTRIBUTION.....	103
<b>8</b>	<b>GLOSSAIRE.....</b>	<b>105</b>

## LISTE DES FIGURES

Figure 2.1-1 - Organisation de la fenêtre principale .....	5
Figure 2.3-1 - Flot d'activité de Modeleur2 .....	12
Figure 4.4-1 - Limites de validité des jeux de données .....	28
Figure 4.4-2 - Description des couches et priorités .....	29
Figure 4.4-3 - Interface usager du MNT .....	30
Figure 4.4-4 - Partition générique sans interpolation.....	34
Figure 4.4-5 - Partition générique interpolée.....	35
Figure 4.5-1 - Couche non maillées .....	37
Figure 4.5-2 - Couches maillées .....	38
Figure 4.5-3 - Interface usager partition de maillage.....	40
Figure 4.6-1 - Conditions aux limites .....	44
Figure 4.6-2 - Arborescence des partitions de conditions aux limites.....	45
Figure 4.6-3 - Fenêtre Partition de conditions aux limites.....	46
Figure 4.7-1 - Assemblage sur un maillage .....	49
Figure 4.8-1 - Exemple d'une simulation avec retour en arrière.....	50
Figure 4.8-2 - Arborescence d'une simulation .....	52
Figure 4.8-3 - Fenêtre historique de la simulation.....	52
Figure 5.1-1 - Liste d'affichage et fenêtres graphiques.....	58
Figure 5.2-1 - Principe des itérateurs .....	64
Figure 5.3-1 - Série de niveaux d'eau en fonction du débit.....	67
Figure 5.3-2 - Grille régulière avec pas constant.....	69
Figure 5.3-3 - Grille régulière avec pas variable .....	69
Figure 5.4-1 - Schéma des composants des <i>scripts</i> et <i>macros</i> .....	74
Figure 5.5-1 - Zones de la calculatrice mode normal .....	80
Figure 5.5-2 - Arborescence des modules et des fonctions de la calculatrice .....	81
Figure 5.5-3 - Calculatrice - Zone de travail avancée.....	81
Figure 5.8-1 Extraire une polyligne d'une polyligne existante .....	87
Figure 5.8-2 - Gestion des lignes .....	88
Figure 5.8-3 - Interface usager de l'analyse d'une multipolyligne.....	90
Figure 5.8-4 - Analyser l'interaction entre deux lignes .....	90



# Introduction

## 1.1 Sommaire

Dans le cadre du projet SEGRI, une version 2.0 de Modeleur sera développée. La nouvelle version implantera les fonctionnalités nécessaires au projet, et aussi, dans la mesure du faisable, toute une gamme de fonctionnalités additionnelles qui ont été demandées par les usagers de la version 1.0.

## 1.2 Système existant

Plusieurs raisons justifient le développement d'une nouvelle version de Modeleur. En voici une liste abrégée:

- Le flot de travail dans Modeleur n'est pas aussi simple et fluide qu'il devrait l'être. Il n'est pas évident de pouvoir utiliser efficacement Modeleur sans être étroitement encadré lors des premières utilisations.

**Solution** : révision et simplification du flot d'activités afin de faciliter l'usage du logiciel. Ajout de fonctionnalités visant à rendre le contact avec Modeleur plus convivial. Développement d'un tutoriel.

- Les usagers de Modeleur ont demandé à ce que le logiciel puisse échanger des données provenant de bases de données relationnelles (ex : Oracle).

**Solution** : migration du système de donnée actuel (hybride objet/relationnel) vers une base de données relationnelle basée sur un langage commun à toutes les bases de données relationnelles : le SQL.

- Les jeux de données (projet) que doit manipuler le logiciel peuvent atteindre une taille considérable (plusieurs millions de points). Il devient parfois irréaliste de charger entièrement un jeu de données en mémoire.

**Solution** : utilisation d'une base de donnée relationnelle, jumelée à l'intégration du support des données SIG (Système d'Information Géographique) permettant de faire des requêtes spatiales.

- Les usagers ont demandé l'ajout de fonctionnalités de visualisation graphique avancées. Le module de visualisation de Modeleur 1.0 n'est pas facilement extensible.

**Solution** : l'utilisation d'un système d'affichage externe qui offre les fonctionnalités recherchées, et des possibilités d'expansion futures.

- Avec l'architecture du logiciel actuel, il est très ardu, voire impossible d'ajouter des composants externes, sans avoir à rouvrir le logiciel et le reconstruire.

**Solution** : une architecture sous forme de bus logiciel, beaucoup plus ouverte à l'ajout de modules externes (plug-in).

### **1.3 Concepts clés**

Tout comme l'était sa version 1.0, Modeleur2 a comme prémisses de base d'offrir un modèle de données cohérent. Étant donné le nombre élevé de données qui ont un lien entre elles, il est essentiel de renforcer au maximum la cohérence et les liens des données entre elles.

Modeleur2 a aussi pour objectif de fournir aux usagers de la souplesse pour réaliser tout ce qu'ils veulent. Modeleur2 propose donc une architecture sous forme de plug-in. Il est ainsi très facile d'ajouter ou d'enlever une fonctionnalité, sans avoir à ouvrir tout le logiciel.

Modeleur2 introduit le concept de « Zone d'étude ». C'est-à-dire que l'utilisateur pourra déterminer une zone (ou région) d'intérêt sur laquelle il désirera porter son étude. Cette zone d'étude sera proposée à l'utilisateur lorsqu'il procédera au chargement d'une donnée. Cette fonctionnalité est rendue possible par le fait qu'une base de données relationnelle avec support SIG chapeaute maintenant Modeleur.

### **1.4 Structure du document**

Ce document contient cinq sections principales :

- description des concepts;
- description des zones de l'interface utilisateur;
- description fonctionnelle;
- description des activités;
- description des outils d'analyse;
- description technique;
- description des aspects généraux (aide, langue et distribution du logiciel).

La section Description des zones de l'interface utilisateur décrit les grandes lignes de l'interface graphique de Modeleur2. Il y est question de l'organisation des fenêtres et des menus, et des barres d'outils.

La section Description fonctionnelle présente le flot d'activités typique lors de la réalisation d'une étude. La section décrit les grandes activités à réaliser lors d'une étude. Chacune des activités est détaillée en explorant les fonctionnalités qu'elle offre. décrit une gamme d'outils qui aideront l'utilisateur à réaliser une étude et à tirer le meilleur bénéfice des résultats.

La section Architecture logicielle présente les choix qui ont été faits à tous les niveaux en ce qui a trait à l'architecture du logiciel et aux aspects techniques du développement.

La section Configuration, aide, langue et distribution discute des sujets suivants: les langues, le support d'aide, la distribution et les logiciels externes à installer.



## 2 Description des zones de l'interface utilisateur

### 2.1 Organisation de la fenêtre principale

L'application Modeleur2 est divisée en trois zones:

- la zone de Contrôle;
- la zone des Fenêtres;
- la zone d'Information.

Les trois zones peuvent être redimensionnées et réorganisées par l'utilisateur. La Figure 2.1-1 illustre la disposition standard de chacune des zones. Une description sommaire de chacune d'elles est présentée ci bas.

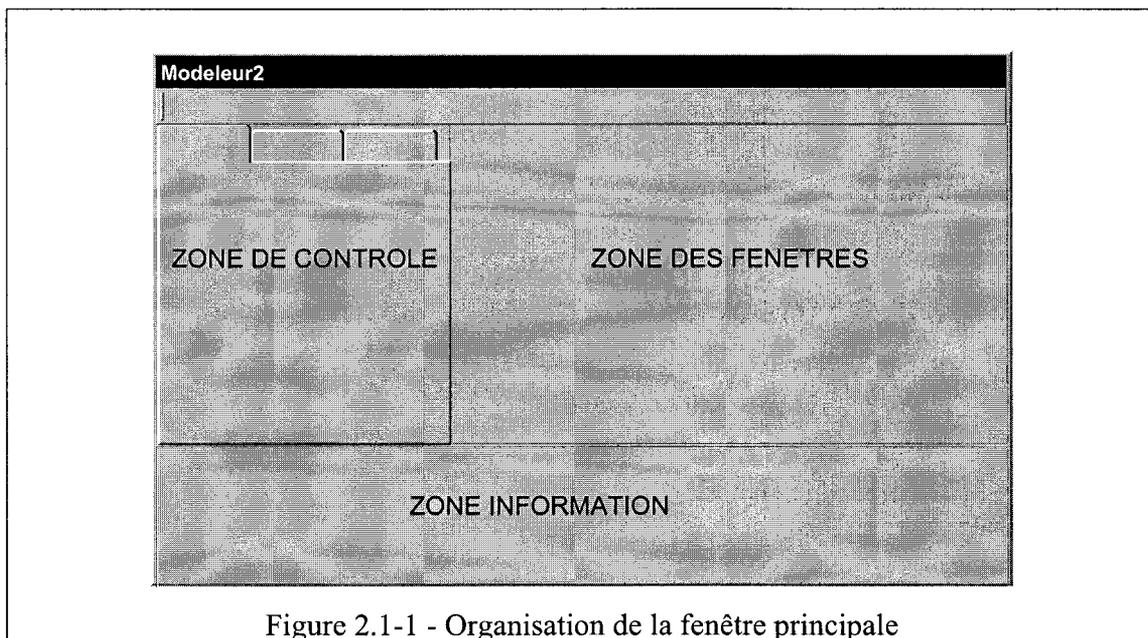


Figure 2.1-1 - Organisation de la fenêtre principale

#### 2.1.1 Zone de contrôle

La zone de contrôle est une zone à onglets typiquement située à gauche dans la fenêtre principale de Modeleur2. Chaque onglet héberge habituellement la zone de contrôle d'un plug-in. Dans la plupart des cas, ceci correspond à un composant de type « arborescence », lequel est utilisé pour explorer et gérer les différentes données offertes par le plug-in.

Par exemple, le plug-in MNT (voir 4.4) offre un explorateur de MNT (Modèle Numérique de Terrain) qui permet de naviguer à travers les différents modèles ouverts / créés par l'utilisateur. Pour chacun d'eux, il est alors possible de naviguer à travers ses partitions et ses couches.

Chaque plug-in a la responsabilité de gérer lui-même les interactions de l'utilisateur avec sa zone de contrôle (son arborescence), c'est-à-dire qu'il détermine comment sont gérés les clics sur les items dans cette zone.

Bien qu'en général un plug-in procède à l'ajout d'un composant de type arborescence à la zone de contrôle, il est important de préciser que rien n'empêche un plug-in d'ajouter un autre type de contrôle, ou même d'ajouter plus d'un composant dans la zone de contrôle.

Un seul plug-in de la zone de contrôle est actif à la fois. C'est celui qui est sélectionné dans la zone de contrôle qui est considéré comme le plug-in actif. Quand un plug-in devient actif, ses entrées de menus sont activées, et celles des autres plug-in sont désactivées. Le plug-in a pour tâche de gérer lui-même ses entrées de menus en fonction de ce que l'utilisateur sélectionne dans la zone de contrôle.

Le plug-in a la possibilité de déterminer quelle fenêtre il veut active. Par exemple, en effectuant un double-clic sur un élément de l'arborescence d'un MNT, ceci a pour effet de donner le focus à la fenêtre associée à ce MNT. Tout plug-in a la possibilité de donner le focus à la fenêtre de son choix dans la zone de fenêtres.

Bien qu'une fenêtre ait été créée par un plug-in en particulier, le fait de la sélectionner n'a pas d'effet sur le plug-in actif dans la zone de contrôle. Par exemple, si le plug-in actif est le plug-in de lignes (voir 5.8), le fait de sélectionner une fenêtre construite par le MNT (voir 4.4) ne change pas le plug-in actif.

### **2.1.2 Zone des fenêtres**

Modeleur2 est une application multifenêtres; c'est-à-dire qu'en cours d'utilisation plusieurs fenêtres peuvent être ouvertes simultanément à l'intérieur de la zone des fenêtres. Il y a des fenêtres de type graphique (2D ou 3D) et des fenêtres spécialisées. Des fenêtres spécialisées viennent s'ajouter au besoin pour supporter les différents plug-in qui sont intégrés au logiciel.

Certains plug-in de Modeleur2 offrent des fonctionnalités qui mènent à l'ouverture ou à la création de fenêtres. En d'autres termes, un plug-in appartenant à cette catégorie ouvre une toute nouvelle fenêtre et y affiche du contenu propre à l'activité qu'il implémente. Par exemple, le plug-in MNT crée une fenêtre graphique dans laquelle il affiche ses partitions et ses couches. Il peut y avoir plusieurs instances de fenêtres créées par le même plug-in.

Les fenêtres dans la zone de fenêtres ne peuvent être déplacées à l'extérieur de cette zone. Il est important de voir que bien que le plug-in crée la fenêtre, il n'est pas le seul à pouvoir y afficher du contenu. En effet, d'autres plug-in n'ont besoin que d'une fenêtre graphique pour travailler, et cette fenêtre peut être n'importe laquelle des fenêtres graphiques dans la zone de fenêtres. Par exemple, le plug-in de lignes (voir 5.8), et le plug-in de visualisation (voir 5.2) appartiennent à cette catégorie.

### **2.1.3 Zone d'Information**

Typiquement située au bas de la fenêtre principale, on retrouve une seconde zone à onglets. Dans cette zone, les plug-in peuvent ajouter des composants de type informatif. De base Modeleur2 offre un composant de log qui est placé dans cette zone. D'autres plug-in peuvent y ajouter d'autres composants par la suite.

Par exemple, le plug-in de script vient ajouter un interpréteur en langage Python dans cette zone alors que le plug-in de lignes vient ajouter un composant permettant d'analyser une ligne (géométrie et valeurs).

## **2.2 Barre de menus**

Modeleur2 contient bien sûr une barre de menus. Les items de menus hébergés dans cette barre peuvent varier en fonction des plug-in présents au démarrage du logiciel. Les plug-in configurent eux-mêmes les entrées de menu requis dans cette barre. Ils sont également responsables d'indiquer l'état du menu (actif / inactif) en fonction du contexte.

Le nom des items et leur disposition dans la barre de menus ne varie pas une fois l'application lancée. Ceci est fait afin d'habituer l'utilisateur à une interface stable et de l'aider à se retrouver à travers l'arborescence des menus.

La section courante n'a pas pour objectif de présenter l'arborescence complète des menus, car la dite arborescence dépend en fait des plug-in qui sont connectés au logiciel.

Les menus Fichier, Edition et Affichage sont présents quel que soit le contexte. Les autres menus sont plutôt la propriété des plug-in et ne seront pas expliqués en détail dans cette section.

### **2.2.1 Menu Fichier**

Dans le menu Fichier, les plug-in qui permettent la création d'une donnée se voient suggérés d'inscrire sous les items de menu Nouveau et Ouvrir les types de données dont ils désirent prendre charge de la création / ouverture.

Par exemple, le plug-in MNT inscrit: Fichier→Nouveau→MNT et Fichier→Ouvrir→MNT. Tout nouveau plug-in pour lequel cette façon de faire est viable

devrait procéder ainsi car cela permet d'avoir une façon de faire uniforme à toutes les activités dans le logiciel.

L'entrée de menu Fichier→Enregistrer est contextuelle, c'est-à-dire que l'action prise dépend de ce qui est sélectionné dans la zone de contrôle.

Le menu Fichier héberge des entrées de menu permettant de gérer les projets (création, ouverture, enregistrement, fermeture, paramètres) et les données contenues dans le projet.

### **2.2.2 Menu Édition**

Les plug-in qui le désirent peuvent prendre en charge les items de menu Édition→copier / Édition→couper / Édition→coller.

### **2.2.3 Menu Affichage**

Les plug-in qui offrent une barre d'outils devraient aller ajouter une entrée de menu : Affichage→Barre d'outils pour la barre d'outils qu'ils publient. Par exemple, le module MNT irait ajouter Affichage→Barre d'outils→MNT.

## **2.3 Barre d'outils**

Trois types de barres d'outils décrites ci-après se greffent à la barre d'outils principale de Modeleur2. L'utilisateur peut configurer lesquelles de ces barres d'outils sont affichées à l'aide de l'item de menu Affichage→Barre d'outils.

### **2.3.1 Barre d'outils standard**

La barre d'outil standard contient les items Nouveau, Ouvrir et Enregistrer. Les boutons Nouveau et Ouvrir sont des « boutons menus ». Cela signifie qu'ils offrent un menu dans lequel l'utilisateur sélectionne le type de donnée à créer ou charger.

Ce sont les mêmes fonctionnalités qui sont rendues disponibles dans les items de menu Fichier→Nouveau et Fichier→Ouvrir. Le bouton Enregistrer est contextuel.

La barre d'outils standard contient également les bouton Imprimer, Copier, Couper et Coller. Chaque plug-in ayant à prendre en charge les mêmes items doit fournir une action pour ces boutons et leur état. Si le plug-in actif ne requiert pas offert ce type de fonctionnalité, les boutons sont désactivés.

### 2.3.2 Barre d'outils spécialisée

La barre d'outils spécialisée peut être configurée par l'utilisateur (i.e. l'utilisateur peut décider quels boutons il veut voir afficher dans la barre). Elle contient des boutons accélérateurs vers des fonctionnalités disponibles ailleurs dans l'interface usager du logiciel.

Dans la mesure du possible, chacun des boutons dans la barre d'outil a son équivalent dans la barre de menu sous le nom de l'activité. Par exemple, sous le menu Maillage, on retrouve les items de menu qui peuvent être ajoutés à la barre de menu spécialisée pour l'activité Maillage. Évidemment, ce ne sont pas toutes les fonctionnalités disponibles via les menus qui sont dans la barre d'outils.

La barre d'outil spécialisée est la responsabilité du plug-in qui implante cette activité. Le plug-in est donc chargé de gérer les différents états des boutons de sa barre d'outils spécialisée.

Autrement dit, si le plug-in MNT devient actif et qu'auparavant c'était le plug-in Partition de maillage qui l'était, les deux plug-in impliqués doivent faire la mise à jour de leur barre d'outils. C'est au plug-in MNT d'indiquer les boutons de sa barre d'outils spécialisée à activer ou désactiver et c'est au plug-in de Partition de maillage d'indiquer que sa barre doit être désactivée (en tout ou en partie).

### 2.3.3 Barre d'outils partagée

Certaines des barres d'outils ont la possibilité d'être activées et désactivées par plus d'un plug-in.

Le plug-in « Éditeur » est responsable de l'édition géométrique. Le plug-in « Éditeur » est proposé avec une barre d'outil d'édition qui s'intègre à la barre principale de Modeleur2. Bien qu'il puisse être présent seul, ce plug-in sera plutôt inséré à l'intérieur d'autres fenêtres.

Les plug-in qui intègrent le plug-in « Éditeur » doivent prendre en compte la présence de la barre d'outils Édition. En effet, les plug-in utilisateurs ont la responsabilité d'activer cette barre d'outil quand ils sont actifs, et de la désactiver quand ils sont inactifs. Le plug-in Éditeur a de son côté pour rôle de gérer les différents états des boutons de la barre d'outils.



### 3 Description fonctionnelle

Modeleur2 fonctionne à l'aide d'activités liées typiquement aux tâches de modélisation. La plupart des tâches-clés de Modeleur2 peuvent être regroupées sous l'une ou l'autre de ces activités. Bien que chacune de ces activités s'enchaîne aux autres pour proposer un flot de travail à l'utilisateur, très souvent des retours en arrière doivent être effectués pour améliorer ou corriger une donnée. Les principales activités que l'on retrouve dans Modeleur2 sont les suivantes :

- gestion des projets;
- gestion des données;
- validation des données;
- construction d'un Modèle Numérique de Terrain (MNT);
- construction d'une partition de conditions aux limites;
- construction d'un maillage;
- assemblage des données sur un maillage;
- simulation hydrodynamique;
- publication des résultats (habillage d'écran).

L'extrait d'une activité peut être réutilisé dans une autre activité, si au préalable, il a été enregistré.

Gravitant autour des activités qui viennent d'être mentionnées, on retrouve une activité d'analyse des résultats qui peut être réalisée avec toute une gamme d'outils d'analyse. En voici la liste :

- l'affichage;
- la visualisation;
- les séries;
- les scripts & macros;
- la calculatrice;
- l'outil de sonde;
- l'outil de lignes;
- l'outil de polygones;
- l'outil de projection de champs;
- l'outil de projection cartographique;
- l'outil de vue tabulée.

La Figure 2.3-1 illustre le flot d'activité suggéré dans l'utilisation de Modeleur2. Ces activités sont articulées autour du modèle suivant :

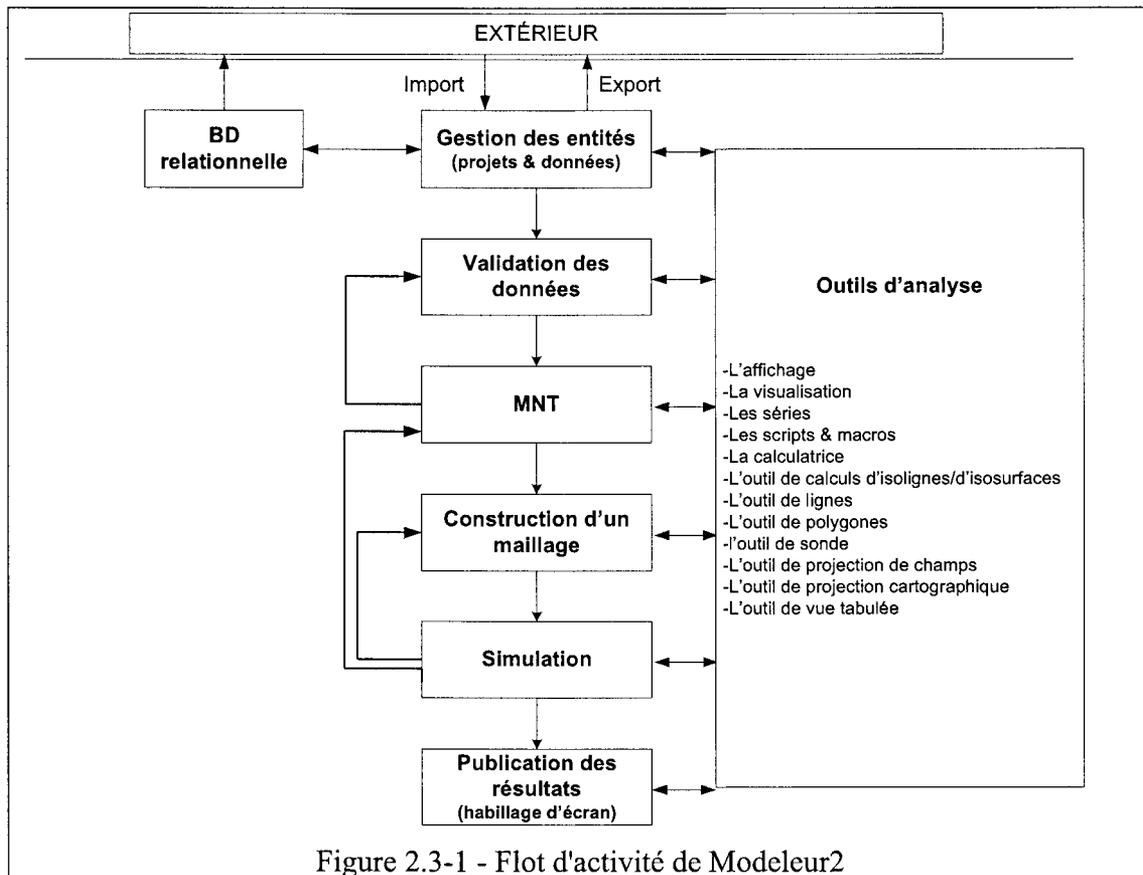


Figure 2.3-1 - Flot d'activité de Modeleur2

Les deux prochains chapitres présentent chacune des activités et chacun des outils d'analyse mentionnés dans la présente section.

## **4 Activités**

### **4.1 Activité : Gestion des projets**

La notion de Projet peut être vue comme étant un contenant qui contient les données créées ou importées par l'utilisateur.

Sous Modeleur2, les données sont toutes emmagasinées dans une base de données. Cette base de données emploie le concept de « schéma ». De façon simplifiée, un schéma peut être vu comme étant un contenant qui maintient des tables d'informations, chaque donnée étant elle-même modélisée à l'aide d'une ou plusieurs tables.

Sous Modeleur2, un projet est l'équivalent d'un schéma, dans le sens où chaque projet possède ses propres tables. Un projet est toujours constitué du même jeu de tables.

#### **4.1.1 Nouveau projet**

L'utilisateur peut créer un nouveau projet (i.e. un nouveau schéma). Lors de la création d'un nouveau projet, l'utilisateur doit spécifier le nom de son projet. Le nom spécifié par l'utilisateur doit être unique.

L'utilisateur doit spécifier la projection cartographique de son projet. La projection est utilisée comme référence pour toutes les données extraites de la base de données. En d'autres mots, toutes les données, peu importe leur projection propre, sont portées dans la projection projet.

L'utilisateur doit spécifier une zone d'affichage par défaut pour les fenêtres graphiques. La zone d'affichage par défaut sert lors de la création de fenêtres graphiques. La zone d'affichage est spécifiée en fonction de la projection retenue pour le projet. Toute nouvelle fenêtre graphique est créée à partir de ces paramètres.

La zone d'affichage permet également de déterminer un paramètre de décalage pour la projection. Ce paramètre est utilisé pour remettre les valeurs de coordonnées  $x$ ,  $y$  dans un ordre de grandeur plus petit, de façon à éviter les pertes de précision.

#### **4.1.2 Ouvrir projet**

L'utilisateur peut ouvrir un projet existant dans une base de données. La liste des projets disponibles pour l'ouverture est extraite à partir de la connexion aux bases de données configurées dans Modeleur2. Cette connexion peut pointer vers des bases de données locales ou bien distantes.

Le fait d'ouvrir un projet a pour effet de rendre disponibles toutes les données contenues dans ce projet, de même que toutes celles provenant de liens externes (voir section 4.2.1).

Pour pouvoir utiliser la plupart des fonctionnalités de Modeleur2, l'utilisateur doit avoir un projet ouvert. Le projet ouvert correspond au schéma d'où les données sont tirées<sup>1</sup> ou bien où elles sont enregistrées.

### **4.1.3 Fermer projet**

L'utilisateur peut fermer un projet. L'utilisateur se voit demander s'il désire enregistrer les données qui ne l'ont pas été. Toutes les fenêtres ouvertes dans la zone des fenêtres (voir 2.1.2) et tous les éléments dans la zone de contrôle (voir 2.1.1) sont fermés dans l'opération.

L'utilisateur doit fermer un projet avant de pouvoir en ouvrir un autre. S'il ne le fait pas, il se voit demander de le faire, ce qui peut occasionner une demande d'enregistrement des données.

### **4.1.4 Enregistrer projet**

L'utilisateur a la possibilité d'enregistrer le projet actuellement ouvert. Ceci a pour effet d'enregistrer chacune des données qui n'était pas déjà sauvegardée.

### **4.1.5 Enregistrer projet sous...**

L'utilisateur peut enregistrer un projet ouvert sous un autre nom. L'utilisateur se voit demander de fournir un nom de projet qui n'existe pas déjà. S'il entre un nom de projet déjà existant, il se voit demander de choisir un autre nom.

Comme pour l'item Enregistrer, cette opération a pour effet d'enregistrer chacune des données qui n'était pas déjà sauvegardée.

L'opération a également pour effet de copier entièrement le contenu des tables du projet ouvert dans un tout nouveau projet (schéma) créé avec le nom fourni par l'utilisateur. L'opération provoque également la mise à jour des tables communes à tous les projets de manière à ce que le nouveau projet soit fonctionnel.

### **4.1.6 Modifier les paramètres du projet**

Les paramètres de projection cartographique et de zone d'affichage discutés en 4.1.1 peuvent être modifiés. Les métadonnées (voir 4.2.2) du projet peuvent également l'être.

---

<sup>1</sup> À l'exception des liens externes.

## 4.2 Gestion des données

L'utilisateur a accès à une boîte de dialogue qui lui permet d'interagir avec les données de son projet. Chacune des fonctionnalités offertes *via* cette boîte de dialogue est décrite ci-après.

### 4.2.1 Créer des liens externes vers des données

L'utilisateur peut créer des liens externes vers une ou plusieurs données qui sont hébergées dans un projet autre que celui actuellement ouvert. Lors de la création d'un lien externe, la donnée continue de résider dans le projet d'où elle origine, mais un lien de référence est fait du projet ouvert vers le projet d'origine. On désigne ce projet d'origine avec l'expression « Projet externe ».

L'utilisateur a la possibilité de créer des liens externes vers un projet hébergé dans la même base de données que le projet ouvert ou vers des projets situés dans d'autres bases de données. Toutefois, pour être en mesure d'avoir accès à des projets situés dans une base de données autre que celle du projet ouvert, l'utilisateur doit au préalable configurer la connexion vers cette base de données.

L'accès aux données par un lien externe est toujours en lecture seule, c'est-à-dire qu'il n'est pas possible de modifier ou de détruire les données d'un projet externe. L'accès aux données peut se faire sur le projet en entier ou seulement sur une partie. Il est possible cependant que certaines données ou certains projets ne soient pas disponibles à l'utilisateur si celui-ci n'a pas les droits d'accès nécessaires pour les lire.

### 4.2.2 Consulter / modifier les métadonnées d'une donnée

Chaque donnée de Modeleur2 possède toute une gamme de métadonnées. Une métadonnée est une information qui décrit un aspect (paramètre global) de la donnée. Sous Modeleur2, les métadonnées suivent le standard FGDC (Federal Geographic Data Committee).

Dans Modeleur2, l'utilisateur a accès aux métadonnées. Il peut les consulter et en modifier certaines. Ceci est fait à l'aide d'un outil externe d'édition de métadonnées.

Les sections suivantes rapportent la liste d'éléments à inclure pour répondre aux critères minimaux de conformité au standard FGDC.

NOTE : pour ne pas perdre le sens de la terminologie utilisée par le FGDC (en anglais), les termes ont été temporairement listés en langue anglaise.

#### 4.2.2.1 Profil de l'utilisateur

Chaque personne qui ajoute des données doit posséder un profil personnel dans la base de données avec les informations suivantes :

*metadata organization, metadata contact position/person, address type, address, city, state or province, postal code, country, phone, fax, e-mail address.*

#### 4.2.2.2 Informations générales

Chaque donnée entrée doit comporter les informations générales suivantes:

*identity of this entry (for future update), originator, publication date (YYYYMMDD), title of data set, edition, presentation form, publication place, publisher, online linkage (URL), abstract, purpose, supplemental information, beginning date: (YYYYMMDD), ending date: (YYYYMMDD), currentness reference, progress, intended data set maintenance and update frequency, west bounding coordinate (-DDD.XXX), east bounding coordinate (-DDD.XXX), north bounding coordinate (DD.XXX), south bounding coordinate (DD.XXX), theme keywords, theme Reference, place keywords, place reference, limits on data accessibility, limits on use of data, browse graphic URL, browse graphic caption, browse graphic file type, spatial data type: (vector, raster, point).*

#### 4.2.2.3 Informations de distribution

Chaque donnée entrée doit comporter les informations de distribution suivantes :

*distribution organization, distribution contact position/person, address type, address, city: state or province, postal code, country, phone, fax, e-mail, dataset name as known by distributor, Liability held by distributor, date of last metadata entry or update (YYYYMMDD).*

### 4.2.3 Consulter / trier les données d'un projet

L'utilisateur peut voir toutes les données qui font partie du projet ouvert. Les données sont identifiées par le nom que l'utilisateur leur assigne. Le nom d'une donnée n'est qu'une métadonnée parmi toutes celles présentées à la section précédente. C'est avec cette métadonnée que l'utilisateur interagira tout au long de son utilisation de Modeleur2.

L'utilisateur voit les données possédées par le projet qu'il a ouvert, mais il voit également celles provenant de liens externes (voir section 4.2.1). Celles provenant de liens externes sont identifiées de façon spéciale pour que l'utilisateur puisse les repérer. Toutes les données sont affichées dans un composant de type arborescence.

Le nom et la date de création sont les éléments qui sont affichés par défaut dans la boîte de dialogue de gestion des données. Il est important de noter qu'on peut présenter

n'importe laquelle des métadonnées dans cette boîte de dialogue. C'est l'utilisateur qui détermine *via* une configuration quelles sont celles qu'il désire voir affichées. Par exemple, l'utilisateur pourrait déterminer qu'il désire afficher la métadonnée URL (voir section 4.2.2) en plus du nom et de la date.

La liste des données peut être triée. Par défaut, le tri peut être fait à l'aide des critères suivants: date, nom, type. Toutefois, l'utilisateur peut effectuer le tri à partir de n'importe laquelle des métadonnées qu'il a demandé de voir afficher (ex : il peut trier les données en fonction de leur URL).

#### **4.2.4 Raffiner la liste des données par critères**

L'utilisateur peut raffiner la liste des données affichée dans la boîte de dialogue de gestion des données. Ceci permet à l'utilisateur de ne travailler qu'avec les données qui l'intéressent.

Afin de réaliser cette opération, l'utilisateur a la possibilité de se définir des critères. Un critère est en fait une requête SQL qui limite les entrées affichées dans la boîte de dialogue. Les critères que l'utilisateur se donne peuvent être enregistrés pour un usage futur. Ces mêmes critères peuvent également servir dans les boîtes de dialogue de chargement pour raffiner la liste des sélections disponibles à plusieurs autres endroits dans le logiciel.

Pour l'aider à construire ses critères, l'utilisateur a accès à un outil de construction de requêtes SQL. Cet outil permet à l'utilisateur de construire de telles requêtes (sur les tables de la base de données) sans avoir à connaître en détails la structure de la base de données.

#### **4.2.5 Consulter les dépendances d'une donnée**

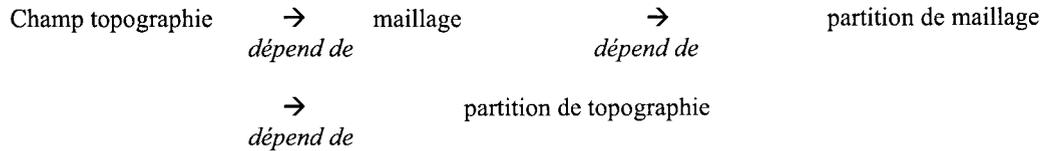
L'utilisateur peut consulter les dépendances des données contenues dans son projet. Il a accès à deux points de vue pour voir ces liens de dépendance . Il a accès à une vue qui décrit un lien: « parent dépend de l'enfant » et à une vue qui décrit un lien « parent est à la source de l'enfant ».

Les deux types d'illustration des dépendances se basent sur une arborescence. Le texte qui suit présente brièvement chacun des deux types de liens de dépendance.

##### **4.2.5.1 Type : parent dépend de l'enfant**

L'utilisateur peut voir quelles sont les dépendances d'une donnée x, et pour cette donnée x, toutes les données dont dépend cette donnée et ainsi de suite. Ceci est représenté sous forme d'une arborescence où pour chaque étage de l'arborescence, l'utilisateur se voit indiquer toutes les données enfant dont dépend le parent.

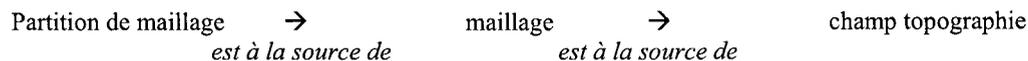
Par exemple, on peut illustrer ceci avec un champ de topographie créé à l'aide d'une partition de topographie et d'un maillage qui lui, fut généré préalablement à l'aide d'une partition de maillage.



#### 4.2.5.2 Type : parent est à la source de l'enfant

L'utilisateur peut voir quelles sont les données qui dépendent d'une donnée  $x$ , et pour cette donnée  $x$ , toutes les données qui en dépendent et ainsi de suite. Ceci est représenté sous forme d'une arborescence où pour chaque étage de l'arborescence, l'utilisateur se voit indiquer toutes les données enfant qui ont comme source la donnée parent.

Par exemple, on peut illustrer ceci avec une partition de maillage, à partir de laquelle on génère un maillage, sur lequel on fait porter un champ de topographie.



#### 4.2.6 Supprimer une donnée

L'utilisateur peut supprimer une donnée s'il n'y a aucune autre donnée qui dépend d'elle dans le projet ouvert. S'il tente de supprimer une donnée dans le projet ouvert, l'utilisateur se voit indiquer quelles sont les dépendances qui font que l'opération de suppression n'est pas acceptée par le système. En d'autres mots, cela permet à l'utilisateur de voir quelles données doivent être supprimées pour que soit permise l'opération demandée.

L'utilisateur peut effacer des données en cascade c'est-à-dire qu'il peut effacer une donnée et toutes les données qui dépendent de cette donnée. L'utilisateur ne peut en aucun temps supprimer les données provenant de liens externes. S'il tente de le faire, un message d'avertissement est affiché.

Toutefois, l'utilisateur peut supprimer une donnée dans un projet même si d'autres projets y font référence à titre de lien externe. Cette opération est permise car il est pratiquement impensable de pouvoir déterminer tous les projets qui ont créé un lien externe vers une donnée du projet, particulièrement dans le cas de projets distants.

Cette façon de procéder peut donc causer l'invalidation de liens externes. Quand un projet est ouvert, il doit donc y avoir vérification de ses liens externes afin de valider qu'ils sont tous disponibles. L'utilisateur est alors averti de l'absence de l'un ou plus des liens externes sur lesquels son projet est basé.

Il y a alors désactivation de toutes les données dépendant du lien externe manquant. Toute donnée du projet en cours qui s'appuie sur le lien externe est donc également rendue indisponible et ainsi de suite jusqu'à la fin de la chaîne de dépendance. Toutefois, tant que l'utilisateur n'a pas besoin d'accéder aux données manquantes, cela ne cause pas de problèmes.

Il est possible pour l'utilisateur de remplacer les liens externes manquant en tout temps. Cette option altère toutefois le projet original, dans le sens où les anciens liens externes sont retirés du projet même s'ils peuvent éventuellement redevenir accessibles. Cette option doit donc être utilisée judicieusement.

## **4.2.7 Importer / Exporter des données**

### **4.2.7.1 À partir / vers des fichiers**

L'utilisateur peut importer / exporter des données à partir de / vers des fichiers. Pour chaque type de donnée qu'il est possible d'importer / exporter, un format intermédiaire a été défini.

À l'importation, Modeleur2 est en mesure d'amener le contenu de ce fichier au format intermédiaire dans les tables du projet ouvert. L'utilisateur importateur doit au préalable appliquer un prétraitement à ses données pour les traduire dans ce format.

À l'exportation, Modeleur2 est en mesure d'écrire le contenu de ses tables dans un fichier avec le format intermédiaire. L'utilisateur exportateur peut par la suite appliquer un post-traitement pour traduire le contenu de ce fichier dans un autre format dont il est propriétaire. L'utilisateur peut exporter autant les données locales que celles provenant de liens externes.

L'utilisateur peut importer / exporter plusieurs fichiers simultanément. La liste des types qu'il est possible d'importer reste à définir.

### **4.2.7.2 À partir / vers une base de données**

L'utilisateur peut importer / exporter des données à partir d'une / vers une base de données (BD). Ladite base de données peut être la même que celle hébergeant le projet ouvert ou une autre base de données, du format Modeleur2 ou d'un autre format.

Contrairement aux liens externes (voir 4.2.1) qui créent une référence vers des tables externes, l'opération d'importation / exportation effectue plutôt une copie du contenu de chaque table, une à une, de la source vers la destination.

Pour pouvoir importer / exporter des données à partir d'une / vers une base de données, le format des tables source / destination doit être compatible avec le format attendu par Modeleur2. L'utilisateur importateur / exportateur doit donc fournir une « vue BD » (*view*)

qui corresponde à ce format. Cette vue BD est en fait une table dans sa bases de données remplie avec l'information contenues dans d'autres tables.

L'opération d'importation se réduit à une copie directe du contenu de la vue BD vers les tables de Modeleur2. Réciproquement, l'exportation peut être vue comme une copie directe du contenu des tables de Modeleur2 vers la vue BD.

L'opération d'exportation nécessite que l'utilisateur définisse des mécanismes pour pouvoir écrire dans sa vue BD car typiquement l'insertion dans une vue BD n'est pas une opération standard. Il devra donc définir des « balises » ou « *triggers* » sur sa vue BD, c'est-à-dire, des règles de distribution de l'information vers les tables appropriées.

### **4.3 Activité : Validation des données de terrain**

Cette activité a pour but de valider les données de terrain qui ont été importées lors de l'activité précédente.

Dans un premier temps, l'utilisateur pourra importer les types de données suivants: topographie, substrat. Par la suite, d'autres types de données pourront être intégrés à Modeleur2, notamment des données météorologiques ou hydrologiques (vents, couvert de glace, précipitations, couverture nuageuse, etc.), des données de plantes aquatiques ou d'utilisation du sol.

L'utilisateur interagit avec des points de mesure. Un point possède une coordonnée  $\{x,y\}$  et des attributs. Les attributs d'un point sont constitués de variables qui constituent l'information en ce point. Ces variables peuvent être des valeurs numériques, des chaînes de caractères, ou autres.

Par exemple, un point de topographie possède au moins comme attribut, une valeur numérique  $z$  (altitude), alors qu'un point de substrat peut posséder comme attributs, sept valeurs numériques (pondération des catégories granulométriques dans l'assemblage).

Quand un point de terrain est importé, on importe du même coup toutes les valeurs de cette donnée  $\{x, y\}$ ,  $\langle \text{attributs} \rangle$ . L'ensemble des points forme un semi de points de terrain.

Il y a deux types de points :

- 1) ceux qui proviennent d'une mesure sur le terrain
- 2) ceux qui éventuellement ont été ajoutés par l'utilisateur

La principale tâche que l'utilisateur doit d'abord accomplir est de désactiver certains points aberrants, c'est-à-dire qui ne sont pas cohérents dans son système de validation. Pour ce faire, il peut faire appel à des filtres pour déterminer les points à désactiver. Les points désactivés ne sont pas effacés (perdus) car l'utilisateur peut en tout temps revenir aux données initiales .

L'utilisateur peut également ajouter des points manuellement ou modifier la valeur des attributs de certains points.

L'utilisateur peut spécifier une zone d'étude (voir 1.3), c'est-à-dire qu'il peut indiquer une région de l'espace délimitant le sous-ensemble de données avec lequel il désire travailler. Cette fonctionnalité est rendue disponible car il arrive fréquemment que l'utilisateur ne soit pas, pour une raison quelconque, intéressé à travailler sur son jeu de données complet.

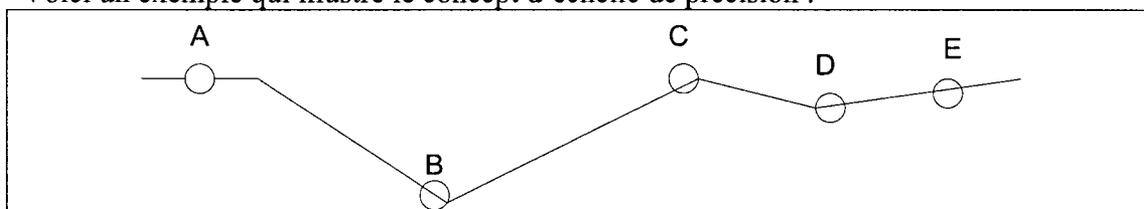
Les données recueillies à un endroit peuvent varier dans le temps (topographie vs érosion), c'est-à-dire qu'une donnée peut avoir été mesurée au même endroit  $\{x,y\}$ , mais à différents moments (différentes campagnes de terrain). Lors de l'importation de données réparties dans le temps, les points de chaque pas de temps sont importés. Ceci a pour effet de produire un semi de points pour chaque pas de temps (ou campagne de terrain).

À l'étape du chargement, l'utilisateur doit spécifier le pas de temps actif (c'est-à-dire le semi de point à afficher). Un point désactivé dans un pas de temps est désactivé dans tous les pas de temps. En contrepartie, l'utilisateur peut modifier la valeur des attributs d'un point dans un semi de points à un pas de temps précis, opération qui n'affecte pas les autres pas de temps.

L'utilisateur peut faire calculer et associer automatiquement une *échelle de précision*<sup>2</sup> à chacun des points de son semi. L'échelle de précision est une valeur qui qualifie le niveau de précision pour lequel le point peut être ignoré. Cette échelle pourra être déterminée de plusieurs façons, par exemple en fonction des valeurs des points voisins (variabilité locale du champ). Le but de cette opération est de permettre la sélection uniquement des points servant à l'atteinte d'une certaine précision lors d'un affichage ou d'un calcul.

L'échelle de précision sert donc à désactiver des points redondants ou superflus, tout en atteignant un niveau de précision demandé. Ainsi, tous les points qui ont une valeur d'échelle de précision plus petite que le niveau demandé pour un calcul / affichage peuvent être ignorés. Les intervalles de valeur pour l'échelle de précision sont à déterminer.

Voici un exemple qui illustre le concept d'échelle de précision :



<sup>2</sup> L'échelle de précision doit être comprise au sens cartographique. Ainsi, à une échelle de représentation donnée (ex : 1/2000) est normalement associée une précision intrinsèque à chaque variable cartographiée.

Si le point D est enlevé, la perte de précision est moindre que si l'on retire le point B. L'échelle sera utilisée par les autres modules, lorsqu'on voudra calculer / afficher en acceptant une marge d'erreur sur les données.

Une fois nettoyé, le semi de points de terrain peut être publié pour devenir un nouveau champ. L'utilisateur pourra utiliser le champ ainsi publié pour monter son MNT (Modèle Numérique de Terrain) (voir activité 4.4). Un champ qui a été publié ne peut être modifié par la suite.

L'utilisateur peut combiner un ou plusieurs semis de points (ou parties de semi de points) dans un même semi. Le semi obtenu peut être également publié au même titre que tout autre semi.

### **4.3.1 Gestion**

#### **4.3.1.1 Nouveau**

L'utilisateur a la possibilité de construire un semi de points de terrain, à partir de rien, c'est-à-dire en y spécifiant tous les points (ex : cas académique). L'utilisateur doit spécifier dans quel repère cartographique (projection) se rattache son semi de points de terrain. Il doit également spécifier quel type de semi de points de terrain il désire créer (topographie, substrat, vent, etc.).

#### **4.3.1.2 Ouvrir**

L'utilisateur peut ouvrir un semi de points de terrain existant. En fonction du type de donnée à la source de ce semi l'utilisateur peut spécifier une zone d'étude pour limiter la taille du jeu de données chargées (voir 1.3).

Pour les données recueillies à différent pas de temps, l'utilisateur peut sélectionner quel pas de temps il désire charger. Le semi de points de terrain à ce pas de temps est alors chargé, et l'utilisateur peut interagir avec lui. L'utilisateur peut comparer des semis de points de terrain de pas de temps différents dans des fenêtres séparées.

#### **4.3.1.3 Fermer**

L'utilisateur peut fermer un semi de points de terrain. L'utilisateur se voit alors demander s'il désire enregistrer son semi de points de terrain, si celui-ci a été modifié.

#### **4.3.1.4 Enregistrer**

Le semi de points de terrain est sauvegardé sous son nom actuel.

#### **4.3.1.5 Enregistrer sous**

Le semi de points de terrain est enregistré sous le nom spécifié par l'utilisateur. Un nouveau semi de points est alors créé.

L'utilisateur peut spécifier une zone d'étude pour ne créer son nouveau semi de points qu'à partir d'une portion du domaine (sous-ensemble de points).

### **4.3.2 Semi de points**

#### **4.3.2.1 Publier des semis de points en champ**

Une fois que l'utilisateur a validé un semi de point, il peut le publier. Le semi de points est alors transformé en *champ* (ex : champ de topographie). Ce champ sera ensuite utilisé pour monter un MNT. Un champ qui a été publié ne peut être modifié par la suite.

#### **4.3.2.2 Combiner des semis de points**

L'utilisateur peut combiner un seul ou plusieurs semis de points, ou encore une ou des parties de semi de points dans un même semi de points. L'utilisateur peut spécifier une zone d'importation dans le semi de points source. Seuls les points contenus dans cette zone sont alors intégrés au semi de points courant.

Les points importés ne maintiennent aucun lien vers leur source. Ainsi, tout changement fait sur les points importés n'affectera pas le semi d'origine, et inversement, tout changement dans le semi d'origine n'affectera pas le semi combiné.

Un attribut permet tout de même d'identifier la source d'origine. Il est donc possible d'utiliser ce critère pour faire un traitement particulier. Par exemple, un utilisateur pourrait ajuster les paramètres des filtres (voir section 4.3.4.1) en fonction du semi de points d'où l'origine du point en traitement.

L'utilisateur peut bénéficier de la fonctionnalité de combinaison de semi de points *via* les actions copier / coller. Il est ainsi possible de copier une partie d'un semi de points ou un semi de points en entier, et de le coller dans un autre semi de points. L'opération de copier / coller doit se faire avec des points du même type. Les points peuvent être sélectionnés dans le semi source à l'aide de l'interface graphique tel décrit à la section 4.3.3.1.

#### **4.3.2.3 Visualiser l'information de semis de points**

Il est possible de visualiser l'information (i.e. les valeurs) de certains types de semis de points (ex : semi de substrat) sous forme graphique. L'utilisateur peut définir une règle de représentation graphique liant les valeurs d'un point à un scalaire qui sera utilisé pour déterminer la couleur lors de la visualisation d'isolignes et d'isosurfaces.

Par exemple, une règle simple pour déterminer la présence de sable au fond d'une rivière pourrait être la suivante : si on retrouve plus de 50% de sable, on retourne le scalaire 1 et sinon 0. Le scalaire 0 (absence de sable) pourrait être associé au rouge et le scalaire 1 au bleu (présence de sable). La représentation sous forme d'isosurfaces indiquerait alors la présence ou l'absence de sable.

Il est important de noter que la règle de représentation pourrait être tout autre. Elle dépend de ce que l'utilisateur recherche comme information. Modeleur2 offre à l'utilisateur la possibilité de définir ses propres règles et d'ainsi pouvoir représenter ce dont il a besoin.

### **4.3.3 Points**

#### **4.3.3.1 Sélectionner des points**

L'utilisateur peut sélectionner un ou des points, individuellement ou par zone. Il peut alors éditer ou effacer ce ou ces points. Les points sélectionnés arborent une couleur différente des autres.

##### **4.3.3.1.1 Individuellement**

L'utilisateur se voit offrir deux moyens pour sélectionner les points individuellement :

- il peut sélectionner un point à la fois en cliquant sur le point en question. Un seul point demeure alors dans la sélection;
- il peut étendre la sélection à plusieurs points en maintenant la touche CTRL enfoncée, et en cliquant individuellement sur les points à ajouter à la sélection.

##### **4.3.3.1.2 Par zone**

L'utilisateur peut aussi sélectionner une zone contenant des points. Différents procédés sont expliqués ci-après. En sortie, la requête de points ainsi délimités (sélection) sera faite sur le polygone SIG défini par l'utilisateur.

En maintenant le bouton de la souris enfoncé, il peut agrandir et rapetisser une zone de sélection rectangulaire. Ceci a pour effet de sélectionner tous les points contenus dans le rectangle.

Une fois la boîte de sélection construite, celle-ci reste active et l'utilisateur peut alors la déplacer et la redimensionner. Ceci a pour effet de sélectionner tous les points qui se trouvent compris dans les limites de la boîte. La boîte de sélection peut également être déplacée à l'aide du clavier.

L'utilisateur peut également construire une zone de sélection en construisant un polygone fermé de forme irrégulière. Il a alors la latitude de contourner les points qu'il désire

exclure de la sélection. Cette zone de sélection peut également être déplacée et éditée sommet à sommet.

L'utilisateur peut sélectionner un point, appuyer sur SHIFT et sélectionner un autre point. Ceci aura pour effet de sélectionner tous les points compris entre ces deux repères.

#### **4.3.3.2 Créer des points**

L'utilisateur peut créer des points manuellement à l'aide de la souris. Il peut également ajouter un ou plusieurs points à son semi de points de terrain, en spécifiant la coordonnée exacte où l'ajout est effectué.

Cette opération est également accessible *via* un *script*, ce qui permet de modifier le modèle de terrain automatiquement, ou même d'en créer un de toutes pièces, sans mobiliser l'interface graphique.

#### **4.3.3.3 Éditer l'information portée par les points**

L'utilisateur a accès à l'information portée par le point *via* une bulle d'aide qu'il obtient en positionnant le curseur dessus. Un double clic sur le point permet d'en modifier les attributs. Les types d'attribut sont les mêmes pour chacun des points du domaine.

Les points qui proviennent d'une mesure gardent en mémoire leur valeur originale. Si l'utilisateur n'est pas satisfait des changements qu'il a effectués, il peut toujours réinitialiser la valeur du point (valeur mesurée).

#### **4.3.3.4 Activer / désactiver des points**

L'utilisateur peut activer / désactiver un point, que ce soit un point usager ou un point mesuré. Un point inactif sera ignoré lors de la publication du semi de points en champ au même titre que s'il avait été effacé.

L'utilisateur peut désactiver des points en fonction de leur échelle de précision. Ceci permet à l'utilisateur de ne conserver que les points essentiels à son analyse.

#### **4.3.3.5 Déplacer des points**

L'utilisateur peut déplacer les points qu'il a lui-même ajoutés. Ceux qui sont issus d'une mesure ne peuvent être déplacés.

#### **4.3.3.6 Effacer des points**

L'utilisateur peut effacer des points qui ont été ajoutés en les sélectionnant à l'aide de la souris ou en spécifiant la coordonnée exacte du point à effacer. Ceux qui sont issus d'une mesure ne peuvent être effacés.

## 4.3.4 Filtrage

### 4.3.4.1 Filtrer les points à désactiver

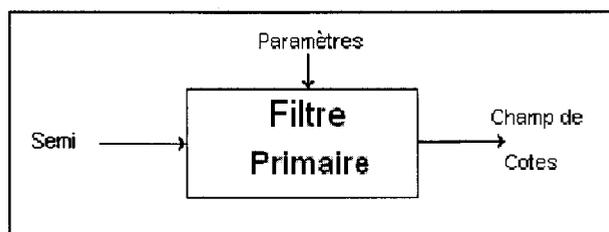
Avant de pouvoir utiliser les semis de points, il est important de les nettoyer des points indésirables qu'ils peuvent contenir. Il est donc nécessaire d'effectuer une opération de filtrage permettant de distinguer les points qui doivent être conservés de ceux qui doivent être désactivés.

Les semis de points (ex : obtenus par balayage laser aéroporté) contiennent habituellement un nombre très élevé d'entrées. Il est donc impensable de laisser l'utilisateur faire ce travail de filtrage manuellement. Modeleur2 propose donc un plug-in de filtrage qui permet d'accueillir des filtres. Les filtres peuvent être développés par l'utilisateur de Modeleur2 en langage C++ ou en Python.

La tâche de filtrage est accomplie en combinant deux types de filtre: les filtres *primaires* et les filtres *secondaires*. Ces deux types de filtre sont discutés dans la prochaine section.

### 4.3.4.2 Filtres primaires

Les filtres primaires sont utilisés pour calculer une cote de qualité à chaque point du semi introduit à l'entrée du filtre. Ces filtres peuvent baser leur appréciation sur différents paramètres, et ils attribuent à chaque point du semi une cote entre 0 et 1.

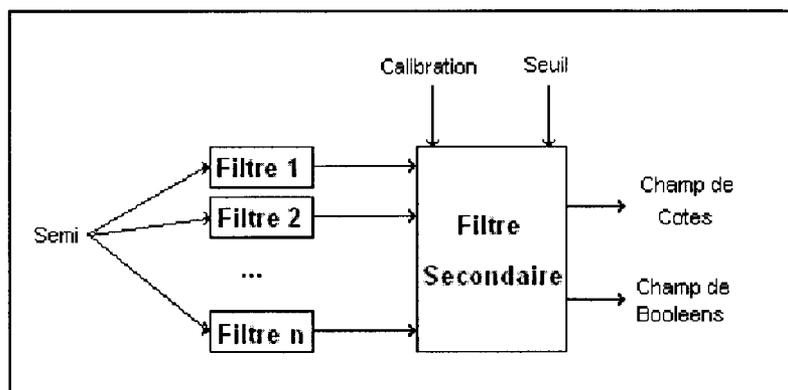


Par exemple, un filtre primaire très simple pourrait renvoyer une cote 0 pour tous les points d'altitude de plus de 1500 mètres et une cote 1 pour tous les autres. Les filtres primaires peuvent évidemment être plus complexes, notamment en se basant sur leur voisinage (i.e. les valeurs des points proches) et en combinant les différentes informations portées par les points.

Les filtres primaires sont adaptés au type de points qu'ils traitent. Par exemple, un filtre primaire de topographie ne pourra être utilisé sur un semi de points de substrat car il a été conçu pour traiter des points de topographie.

#### 4.3.4.3 Filtres secondaires

Les filtres secondaires sont les filtres qui vont travailler à partir des cotes produites par les filtres primaires. Ils ont comme rôle de ressortir un champ de booléens déterminé à partir des cotes combinées de filtres primaires et d'une valeur seuil.



Les filtres secondaires permettent d'effectuer une pondération pour augmenter / diminuer la valeur relative de l'un des filtres primaires impliqué dans le système. Par exemple, un filtre primaire peut très bien fonctionner pour un type de terrain, mais être beaucoup moins efficace sur un autre type. On peut alors jouer sur le poids relatif du filtre dans le système en fonction du type de terrain considéré.

Les combinaisons appliquées peuvent être de simples pondérations ou peuvent être mises en place à partir de techniques plus évoluées comme la logique floue, les réseaux de neurones, les méthodes statistiques, etc.)

### 4.4 Activité : Modélisation numérique de terrain

L'extrait de cette activité est un Modèle Numérique de Terrain (MNT). Le MNT décrit chacun des aspects du terrain sur lequel l'utilisateur désire baser son étude. Un MNT contient un nombre non déterminé de partitions de données. Une partition est un processus déclaratif permettant d'associer à un domaine ou un ensemble de sous-domaines une valeur ou un jeu de données ou de paramètres quelconque. Cette notion a fait largement ses preuves dans la version 1.0 de Modeleur. Le nombre de partitions est indéterminé *a priori* car c'est à l'utilisateur à déterminer de quoi sera constitué son MNT.

Il existe des *partitions spécialisées* et des *partitions génériques*. Les partitions *spécialisées* décrivent un aspect particulier prédéfini du terrain (exemple : topographie, substrat, glace, etc.) et sont décrites dans les pages qui suivent.

Les partitions *génériques*, elles, servent à porter de l'information définie par l'utilisateur et sont présentées à la section 4.4.4.

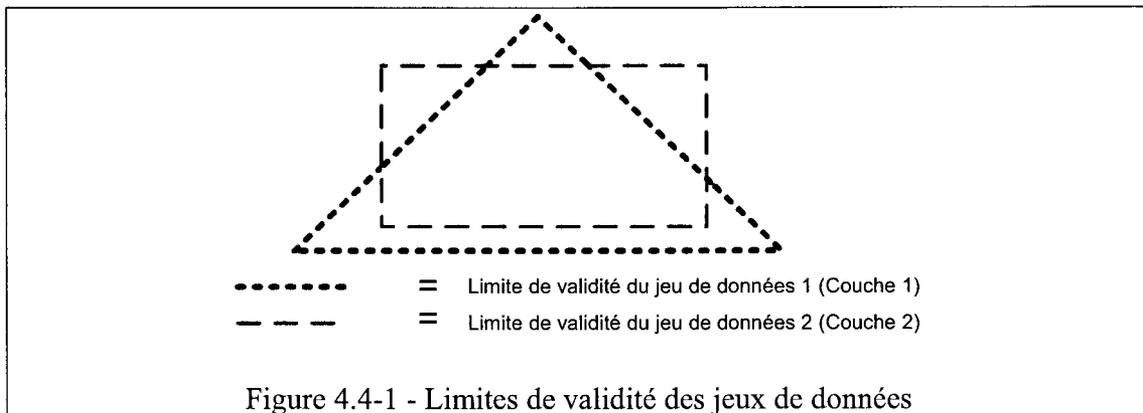
Les partitions spécialisées sont construites en superposant des *couches de données*. Une couche de données est constituée de sous-domaines auxquels est associé un même jeu de données. Un jeu de données fait référence à ce que l'on a nommé *champ publié* (voir 4.3.2.1) et correspond en fait à un semi de points qui a été publié à l'activité validation des données.

Les couches de données se voient allouer des ordres de priorité. La couche de données se retrouvant au sommet des priorités (à l'avant-plan) est celle qui sera utilisée lorsque l'information sera transportée (projetée) sur un maillage.

L'utilisateur peut amener une couche en mode d'édition afin d'en modifier le contour. Pendant l'édition, la plupart des fonctionnalités du logiciel sont suspendues. L'utilisateur doit indiquer lorsqu'il désire sortir du mode d'édition.

Les modifications faites peuvent alors être conservées ou rejetées. Lorsqu'une couche passe en mode d'édition, elle se retrouve affichée par dessus toutes les autres couches sans que soit modifié son ordre de priorité. À la sortie du mode d'édition, elle reprend sa position dans l'ordre de priorité.

Le ou les sous-domaines d'une couche doivent obligatoirement se trouver dans la limite du jeu de données. La limite du jeu de données correspond au contour du jeu de données, c'est-à-dire les points (x,y) qui délimitent l'enveloppe où le jeu de données peut fournir une valeur par interpolation. La Figure 4.4-1 introduit un exemple, où l'on peut voir les limites de validité des jeux de données, et donc, la limite d'édition des couches bâties sur ceux-ci.



La Figure 4.4-2 propose une description des couches, en fonction des limites de validité de leur jeu de données. Le cas a) illustre un cas où la couche est construite d'un seul sous-domaine à partir du jeu de données 1 alors que le cas b) illustre une situation avec deux sous-domaines à partir du jeu de données 2. Dans le cas c) la couche 1 est plus prioritaire que la couche 2. À l'inverse, dans le cas d), la couche 2 est plus prioritaire que la couche 1.

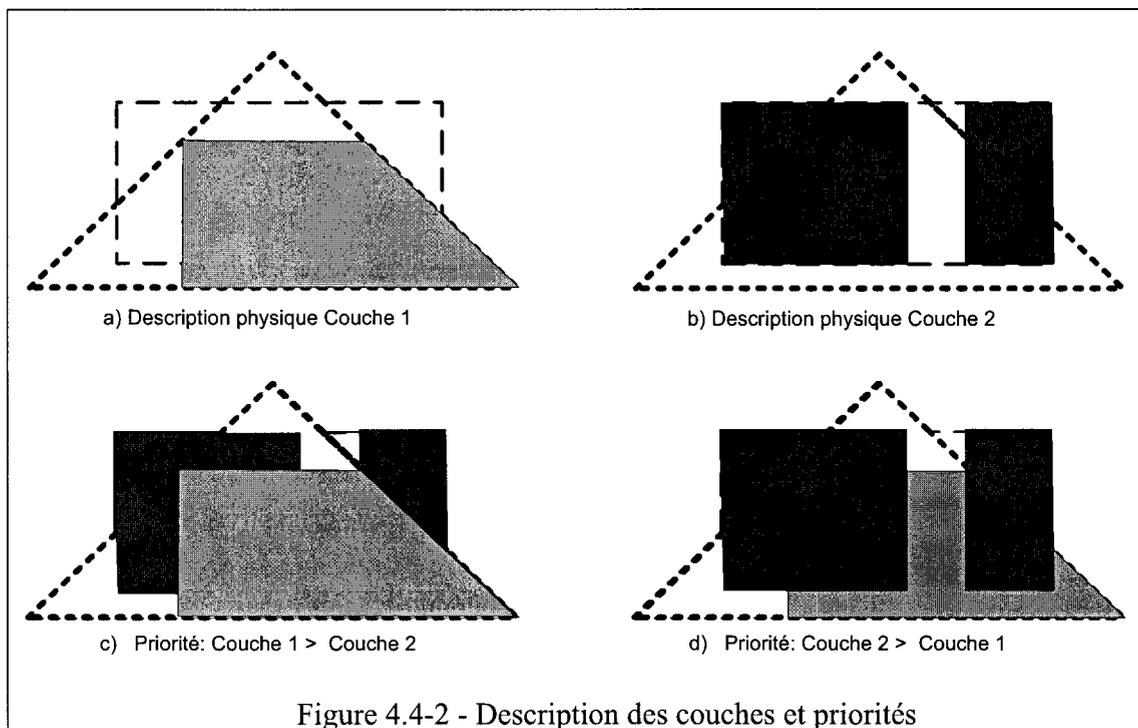


Figure 4.4-2 - Description des couches et priorités

Dans une partition, il est possible de créer une couche de données qui couvre tout le domaine d'intérêt, et à laquelle on associe une valeur par défaut ou une valeur facilement reconnaissable (ex : -99) indiquant l'absence de données. Cette couche se voit attribuer la priorité minimale. Elle a pour but de s'assurer que la partition ne causera pas de trous dans le squelette du MNT aux endroits où il n'y a pas de données.

#### 4.4.1 Interface usager

À la création ou à l'ouverture du MNT, un composant de type « arborescence » est ajouté à la zone de contrôle et une fenêtre graphique est créée. La Figure 4.4-3 donne un aperçu de l'interface usager présentée à l'utilisateur.

L'arborescence héberge toutes les partitions du MNT et les couches contenues par ces partitions. En sélectionnant ces éléments de l'arborescence, l'utilisateur peut avoir accès aux fonctionnalités qu'offre l'élément sélectionné.

Lorsqu'il ajoute une partition ou une couche, l'utilisateur voit le changement immédiatement reproduit dans l'arborescence. L'utilisateur peut sélectionner une couche dans la fenêtre

MNT tout comme il peut le faire à l'aide de l'arborescence. Le jeu de fonctionnalités est identique qu'une couche soit sélectionnée dans la fenêtre ou dans l'arborescence.

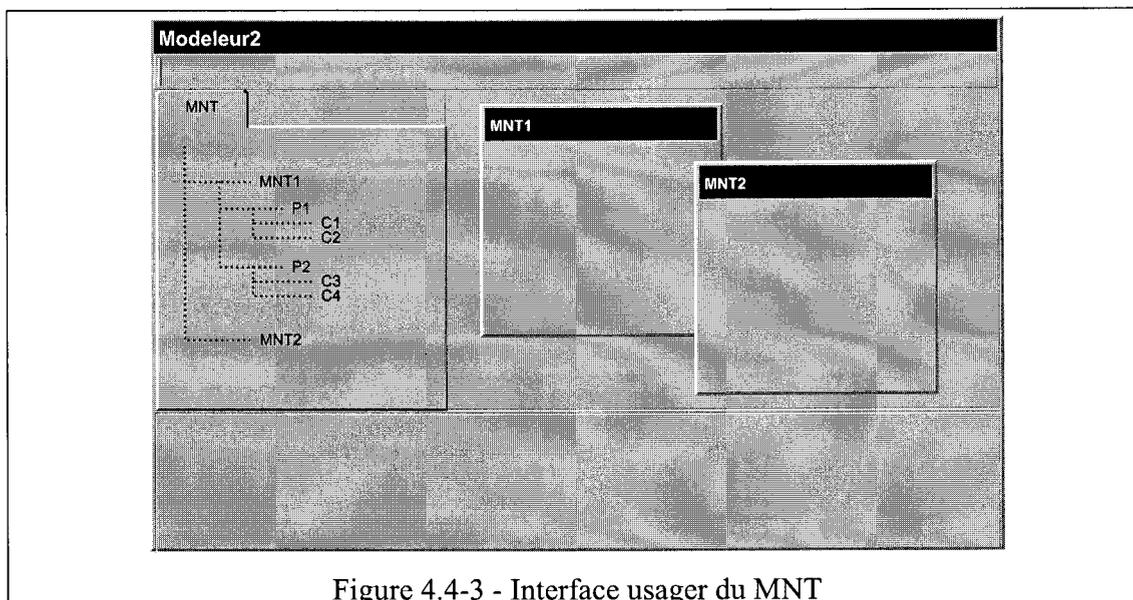


Figure 4.4-3 - Interface usager du MNT

L'action d'effectuer un double-clic sur une couche ou une partition a pour effet de rendre active la fenêtre du MNT qui possède ces couches et partitions. Le contenu de la fenêtre correspond alors à la vue effective de la partition. Les couches sont ainsi superposées l'une sur l'autre en ordre de priorité croissante.

Si l'utilisateur n'est pas satisfait du résultat obtenu, il peut changer l'ordre de priorité des couches en les déplaçant dans l'arborescence. Les couches qui sont dans le bas de la partition ont le niveau de priorité le plus élevé. Par exemple, dans la Figure 4.4-3, la couche C2 est prioritaire sur la couche C1.

## 4.4.2 Gestion d'un MNT

### 4.4.2.1 Nouveau

L'utilisateur peut créer un MNT vide (sans partitions et sans couches). Tout ce que l'utilisateur peut faire à cette étape, c'est d'ajouter des partitions à son MNT.

### 4.4.2.2 Ouvrir

L'utilisateur peut ouvrir un MNT existant. Toutes les partitions contenues dans ce MNT sont également ouvertes ainsi que les couches contenues dans ces partitions. L'utilisateur peut alors naviguer dans son MNT et interagir avec les partitions et couches qu'il contient.

#### **4.4.2.3 Fermer**

L'utilisateur peut fermer le MNT courant. Au préalable, l'utilisateur se voit demander s'il désire l'enregistrer.

#### **4.4.2.4 Enregistrer**

L'utilisateur peut enregistrer le MNT courant. Ceci a pour effet d'enregistrer toutes les partitions contenues dans ce MNT, et toutes les couches contenues dans ces partitions.

L'opération s'effectue sur les entrées existantes de la base de données. Donc, du point de vue de la base de données, on a affaire à une mise à jour (*update*). Si certaines partitions ou couches ne sont pas présentes, elles sont ajoutées à la base de données dans l'opération.

#### **4.4.2.5 Enregistrer sous...**

L'utilisateur peut enregistrer le MNT en cours sous un nom qu'il spécifie. Il doit également spécifier la projection SIG de son MNT.

Il y a création de nouvelles entrées pour le MNT, ses partitions et ses couches. Du point de vue de la base de données, on a affaire à un ajout (*insert*). Les partitions et couches contenues dans le MNT en cours sont copiées dans le nouveau MNT.

### **4.4.3 Fonctionnalités du MNT**

#### **4.4.3.1 Ajouter une nouvelle partition**

Lorsque l'utilisateur a créé ou ouvert un MNT, il peut y ajouter une partition. L'utilisateur doit spécifier le type de partition qu'il désire ajouter. Un même type de partition peut être présent plusieurs fois dans un MNT.

#### **4.4.3.2 Ajouter une partition existante**

Bien qu'une partition ait obligatoirement à être liée à un MNT, l'utilisateur peut choisir d'ajouter à son MNT une partition existante issue d'un autre MNT. La partition ajoutée au MNT est une copie de la partition existante. En ce sens, aucun lien n'est gardé vers la partition source et tout changement fait à cette partition restera local au MNT en cours.

#### **4.4.3.5 Retirer une partition**

L'utilisateur peut retirer une partition du MNT. Aucune partition ne peut exister à l'extérieur d'un MNT. L'utilisateur se verra donc indiquer que la partition sera effacée de la base de données. Il aura alors le choix de confirmer cette opération ou de demander à ce qu'un MNT temporaire soit créé pour héberger cette partition orpheline.

#### **4.4.3.6 Ajouter une nouvelle couche de données**

L'utilisateur peut ajouter une couche de données à une partition. L'utilisateur doit sélectionner quel jeu de données doit être assigné à cette couche. La couche se voit attribuer par défaut une priorité maximum.

#### **4.4.3.7 Ajouter une couche de données existante**

Bien qu'une couche de données ait obligatoirement à être liée à une partition d'un MNT, l'utilisateur peut choisir d'ajouter à son MNT une couche existante issue d'un autre MNT. La couche ajoutée au MNT est une copie de la couche existante. En ce sens, aucun lien n'est gardé vers la couche source et tout changement fait à la nouvelle couche restera local au MNT en cours.

#### **4.4.3.8 Retirer une couche de données**

L'utilisateur peut choisir d'enlever une couche de données de sa partition. Lors du retrait d'une couche de données, les couches de données de la partition sont réordonnées en fonction de leur priorité.

Aucune couche ne peut exister à l'extérieur d'une partition d'un MNT. L'utilisateur se verra donc indiquer que la couche sera effacée de la base de données. Il aura alors le choix de confirmer cette opération ou de demander à ce qu'une partition temporaire dans un MNT temporaire soit créée pour héberger cette couche orpheline.

#### **4.4.3.9 Afficher les limites de jeux de données**

L'utilisateur peut afficher les limites des différents jeux de données. L'utilisateur ne peut toutefois pas interagir avec cette composante visuelle. Elle est présente uniquement à titre informatif pour aider l'utilisateur à dessiner ses couches.

#### **4.4.3.10 Éditer une couche de données**

Lors de l'édition d'une couche, l'utilisateur ne peut dépasser les limites couvertes par le jeu de données associé à cette couche. Le contour du jeu de données actif reste d'ailleurs toujours affiché pour permettre à l'utilisateur de situer les limites où il peut éditer.

L'utilisateur construit lui-même un polygone en reliant des segments de droites entre eux à l'aide de l'éditeur de sous-domaines. L'utilisateur peut construire son sous domaine en utilisant comme point de départ la ligne de contour du jeu de données.

L'utilisateur a la possibilité d'annuler toute modification géométrique faite pendant l'édition d'une couche à sa sortie du mode édition.

#### **4.4.3.11 Copier et coller une couche de données**

L'utilisateur peut copier une couche d'une partition et la coller sur une autre ou bien sur la partition courante. L'opération aura pour effet de créer une nouvelle couche, d'y dupliquer le contenu de la couche source et d'ajouter celle-ci à la partition cible. L'opération n'est permise que sur une partition du même type que la couche copiée.

#### **4.4.3.12 Modifier la couleur des couches**

L'utilisateur peut modifier la couleur d'affichage d'une couche. Deux couches ne peuvent pas avoir la même couleur.

#### **4.4.3.13 Modifier le nom d'une couche ou d'une partition**

L'utilisateur peut modifier le nom d'une couche ou d'une partition. Le changement affecte la métadonnée associée à la couche ou à la partition. Quand l'utilisateur visite l'activité gestion des données (voir 4.2), le nom apparaît également comme étant modifié.

#### **4.4.3.14 Modifier la priorité d'une couche de données**

L'utilisateur peut modifier la priorité d'une couche de données. L'opération peut être réalisée en incrémentant / décrémentant la priorité d'une couche, ou bien en effectuant une opération de «glisser – déposer» (*drag and drop*).

#### **4.4.3.15 Annexer des commentaires**

L'utilisateur peut associer des commentaires aux partitions et aux couches. Ces commentaires sont enregistrés dans la base de données lorsque la partition ou la couche est enregistrée.

#### **4.4.3.16 Générer le squelette du MNT**

L'utilisateur peut générer le squelette de son MNT. Le squelette d'un MNT correspond à la plus petite zone commune à un ensemble de partitions spécifié par l'utilisateur. Le squelette du MNT pourra servir lors de la construction d'un maillage pour s'assurer que le maillage ne dépasse pas la zone de données partagée par toutes les partitions d'intérêt.

### **4.4.4 Partitions génériques**

La section précédente traitait d'une catégorie de partitions dites spécialisées, c'est-à-dire d'un type prédéfini dans Modeleur2. Toutes ces partitions ont le point commun d'être associé à un jeu de donnée (topographie, substrat, glace, etc.)

Afin de laisser le plus de latitude possible aux usagers, un autre type de partition a été défini : le type *partition générique*. Une partition générique sert à porter de l'information

utilisateur dont l'utilisateur définit la structure et le contenu. L'utilisateur peut donc faire porter tout type d'information à la partition afin d'être en mesure de trouver la valeur de son information en tout point  $\{x,y\}$  de son domaine.

L'information utilisateur sera interpolée si l'information est portée aux sommets, et ne le sera pas si l'information est portée par les sous-domaines. L'utilisateur définit et utilise ses partitions à l'aide du langage de la calculatrice (Python). Afin de clarifier quelque peu l'utilisation de partitions génériques, deux exemples simples sont présentés dans les prochaines lignes : l'un sans interpolation et l'autre avec interpolation.

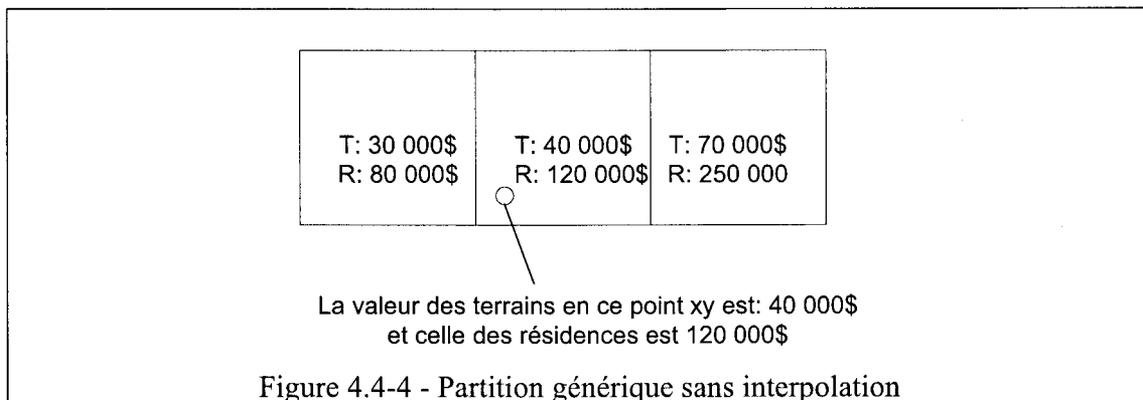
#### 4.4.4.1 Partition générique sans interpolation

Un usager divise un territoire inondable en sous-domaines d'évaluation foncière et assigne une valeur moyenne (en \$) aux groupes de résidences et de terrains qui occupent chacun des sous-domaines.

L'utilisateur définit l'information portée en indiquant que la partition porte deux entiers, l'un correspond à la valeur moyenne des résidences et l'autre à la valeur moyenne des terrains.

À l'aide cette partition générique, l'utilisateur est à même de pouvoir trouver la valeur moyenne des terrains et des résidences en tout point  $\{x,y\}$  de son domaine.

La Figure 4.4-4 illustre cet exemple.

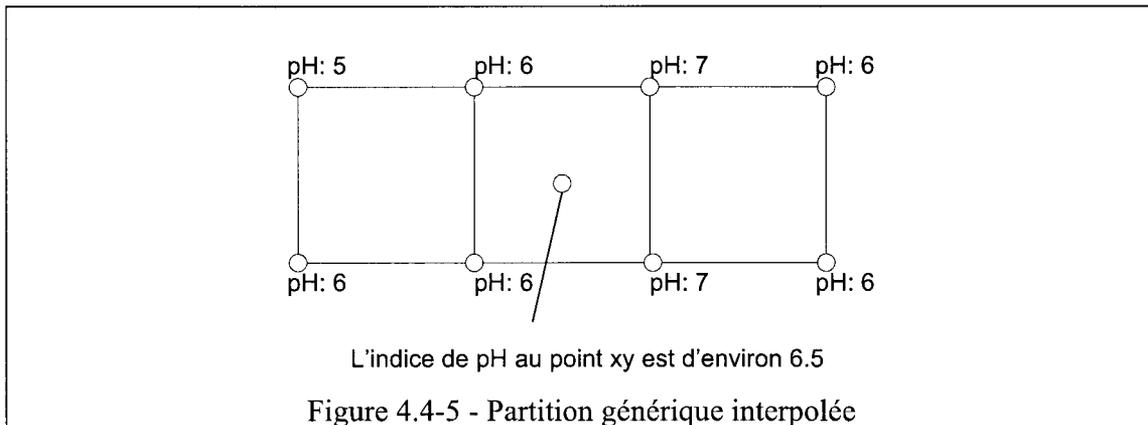


#### 4.4.4.2 Partition générique avec interpolation

Un usager divise un domaine fluvial en sous-domaines représentant des attributs continus de l'habitat (ex : gradients de qualité de l'eau).

L'utilisateur définit l'information portée en indiquant que la partition porte un entier, lequel correspond à la valeur du pH. À l'aide de cette partition générique, l'utilisateur est à même de pouvoir trouver la valeur du pH en tout point  $\{x,y\}$  de son domaine.

La Figure 4.4-4 illustre cet exemple.



#### 4.4.4.3 Définir la partition générique

L'utilisateur doit spécifier si l'information sera portée aux sommets ou par les sous-domaines.

L'utilisateur doit définir le type d'information que la partition hébergera. L'information portée par la partition générique doit être définie en spécifiant une structure de données à l'aide du langage de la calculatrice.

Si l'utilisateur a spécifié que l'information est portée aux sommets, il doit définir les règles d'interpolation pour sa structure de données.

Il doit ainsi définir une règle pour effectuer l'addition et une pour effectuer la multiplication :

- `struct_utilisateur + (struct_utilisateur a, struct_utilisateur b)`
- `struct_utilisateur * (réel x, struct_utilisateur a)`

Il est important de noter que d'autres règles devront peut-être se voir définies par l'utilisateur. La liste exhaustive des règles reste à définir.

#### 4.4.4.4 Construire la partition générique

Comme dans le cas de partitions spécialisées, l'utilisateur peut créer ses partitions génériques en assemblant des polygones au moyen de lignes, à l'aide de l'éditeur de sous-domaines.

L'utilisateur peut ensuite assigner de l'information aux sommets ou aux sous-domaines qu'il a définis.

## **4.4.5 Édition de sous-domaines**

### **4.4.5.1 Créer un sous-domaine**

L'utilisateur peut créer un sous-domaine en ajoutant des sommets jusqu'à la construction d'une forme fermée. Si l'utilisateur ne ferme pas la forme en reliant le dernier sommet au premier, elle sera effacée.

### **4.4.5.2 Sélectionner un sous-domaine**

L'utilisateur peut sélectionner un sous-domaine. Toute région fermée est considérée comme un sous-domaine.

### **4.4.5.3 Effacer un sous-domaine**

L'utilisateur peut effacer un sous-domaine.

### **4.4.5.4 Fusionner deux sous-domaine**

L'utilisateur peut fusionner deux sous-domaines.

### **4.4.5.5 Enlever un sommet d'un sous-domaine**

L'utilisateur peut enlever un sommet au sous-domaine. Ceci a pour effet de relier les deux sommets qui étaient les voisins du sommet enlevé.

### **4.4.5.6 Scinder une ligne d'un sous-domaine**

L'utilisateur peut scinder une ligne existante d'un sous-domaine, ce qui a pour effet d'ajouter un nouveau sommet au centre de cette ligne. L'utilisateur peut alors déplacer le nouveau sommet.

## **4.5 Activité : Construction d'un maillage**

### **4.5.1 Théorie relative aux partitions de maillage**

Cette activité a pour but de construire un maillage d'éléments finis. Pour arriver à compléter cette tâche, l'utilisateur doit se construire une partition de maillage. À l'aide de la partition de maillage, l'utilisateur peut spécifier la géométrie et le degré de raffinement local de son maillage.

La construction des partitions de maillage est basée sur un principe de superposition de couches de maillage en fonction de leur priorité, principe similaire à celui employé pour le MNT (voir 4.4). Ladite construction est réalisée à l'aide de l'éditeur de sous-domaines (voir 4.4.5).

L'utilisateur peut s'aider du « squelette de MNT » (voir 4.4) pour construire sa partition de maillage. Cela lui permet d'identifier les zones où il y a des données pour toutes les partitions du MNT afin de construire un maillage qui ne déborde pas de la couverture des données.

Une couche de maillage est constituée d'un ou plusieurs sous-domaine qui se voient assignés des paramètres de maillage. Les paramètres de maillage des couches servent à spécifier le degré de raffinement du maillage. De base, dans Modeleur2, le degré de raffinement peut être contrôlé par les maillages d'au moins deux manières :

- 1) Directement par la taille des mailles
- 2) Indirectement par un critère de tolérance à l'erreur

D'autres maillages, et par le fait même d'autres façons de contrôler le degré de raffinement, pourront être intégrés à Modeleur2 par des développeurs externes.

Chaque couche de maillage se voit assigner un ordre de priorité, que l'utilisateur peut changer par la suite. Lorsqu'il y a des zones d'intersection entre deux ou plusieurs couches, les paramètres de la couche avec la plus haute priorité sont employés pour la génération du maillage.

L'utilisateur peut mailler les couches de sa partition une à une, plusieurs à la fois, ou toutes en même temps. Le résultat sera le même peu importe l'ordre dans lequel les couches sont maillées, car la frontière entre deux couches est toujours maillée avec les paramètres de la couche la plus prioritaire.

Dans la Figure 4.5-1, une partition de maillage constituée de trois couches est présentée. La couche C1 est plus prioritaire que C2 qui elle est plus prioritaire que C3.

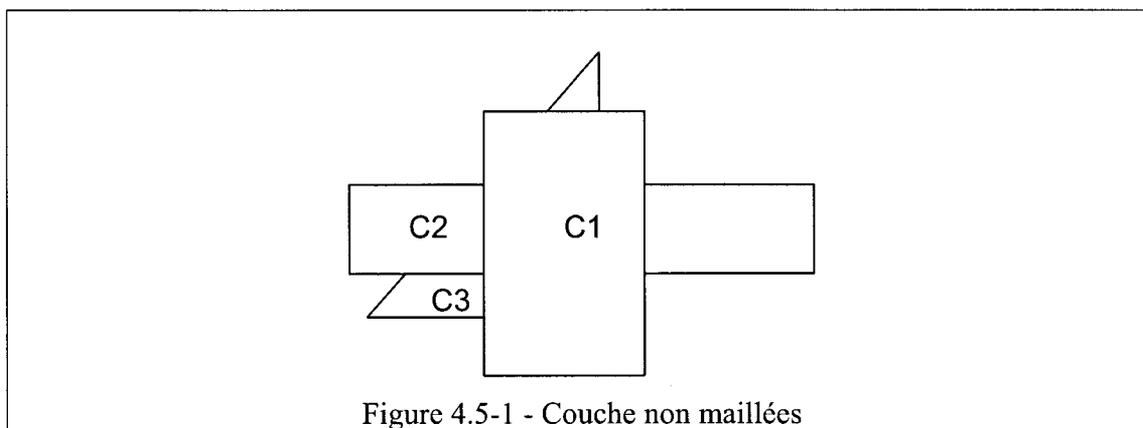
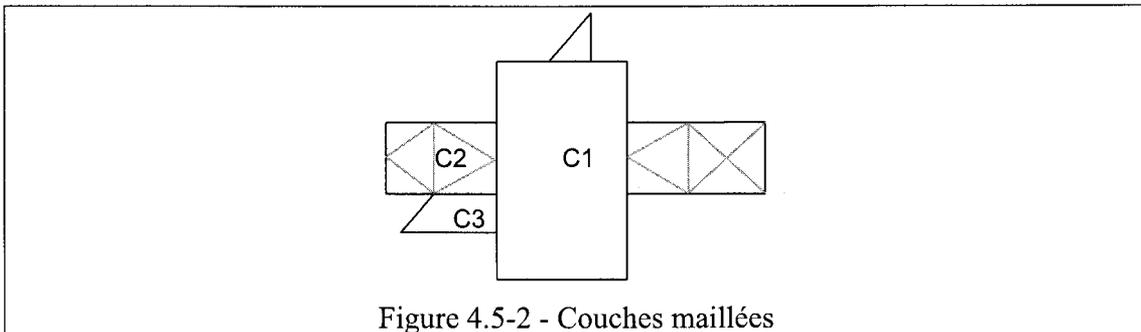


Figure 4.5-1 - Couche non maillées

Lorsque l'on demande de mailler C2, on obtient le maillage présenté à la Figure 4.5-2. C2 est maillée avec les paramètres de C1 aux endroits où les deux couches s'intersectent.



Ce que l'utilisateur voit apparaître dans les couches maillées n'est qu'un artifice visuel. C'est-à-dire que les maillages à la source des représentations visuelles affichées sur les couches ne sont pas rattachés aux couches et ne seront pas enregistrés quand la partition le sera.

L'objectif recherché est d'aider l'utilisateur à construire un maillage avec une taille de mailles qui lui convient. Quand l'utilisateur est satisfait du maillage qu'il a obtenu, il peut l'assembler pour obtenir un résultat qui, cette fois, pourra être enregistré. Il peut par la suite réutiliser ce maillage dans d'autres sections du logiciel.

## 4.5.2 Paramètres de maillage

### 4.5.2.1 Taille des mailles

L'utilisateur peut spécifier la taille des mailles aux sommets des sous-domaines de ses couches, sur sa couche en entier, ou encore à des points de contrôle quelconques sur sa couche (qui deviendront alors des nœuds du maillage).

### 4.5.2.2 Champ d'erreur

L'utilisateur peut demander à ce que la taille des mailles soit automatiquement ajustée par le système en fonction d'un champ d'erreur numérique. L'utilisateur assigne un champ d'erreur à une couche de maillage.

L'utilisateur spécifie également un niveau de tolérance (critère) à l'erreur acceptable pour cette couche. Il est possible d'assigner un critère de tolérance à plus d'une couche de maillage simultanément.

Une fois le critère de tolérance fourni, le mailleur se charge de mailler la couche en répondant au niveau de tolérance prescrit, c'est-à-dire qu'il ajoute des mailles jusqu'à ce que le niveau d'erreur soit acceptable.

Les champs d'erreur peuvent provenir de résultats de simulation (champ d'erreur de niveau d'eau, de débit, etc.) Dans ce cas, une simulation est nécessaire à l'adaptation du maillage afin d'obtenir un champ existant avec lequel travailler.

Les champs d'erreur peuvent également être déterminés à partir de données mesurées (ex : topographie). Dans ce cas, la projection des données de la partition de topographie sur un maillage avec des paramètres par défaut permet d'obtenir un champ de topographie, avec lequel on peut trouver le champ d'erreur.

Bien que ce soit un cas plus académique, il est également possible d'obtenir un champ d'erreur à partir d'une équation analytique.

### **4.5.3 Ajout de nouveaux mailleurs**

Bien que l'utilisateur puisse choisir le mailleur dans une liste contenant les mailleurs livrés avec Modeleur2, le logiciel offre toute la flexibilité nécessaire à l'intégration de nouveaux mailleurs. Ces mailleurs pourront être développés ultérieurement.

Tous les mécanismes sont offerts à l'utilisateur pour qu'il puisse aisément intégrer un mailleur de sa conception. Comme seule contrainte, le mailleur doit être encapsulé dans une DLL qui réponde à l'interface de mailleur commune à tous les mailleurs et implanter des fonctions pour mailler une ligne 1D et pour mailler une zone 2D.

Les mailleurs doivent connaître le domaine à mailler, les contraintes aux frontières et des paramètres de maillage. De base, les paramètres de maillage sont centrés autour du duo de paramètres suivants: un champ d'ellipse erreur et un niveau de tolérance.

Toutefois, afin d'offrir le maximum d'ouverture vers l'extérieur, l'interface des mailleurs propose en plus un paramètre optionnel qui se présente sous forme d'une chaîne de caractères. Ce paramètre peut héberger n'importe quel contenu et se veut un moyen pour configurer chaque mailleur avec des paramètres propriétaires.

### **4.5.4 Interface usager pour la gestion d'une partition de maillage**

À la création ou à l'ouverture d'une partition de maillage, un composant de type « arborescence » est ajouté à la zone de contrôle et une fenêtre graphique est créée.

Le comportement est très similaire à ce qui a été décrit à la section 4.4.1 pour l'activité MNT. La principale différence est que l'arborescence possède deux niveaux de profondeur au lieu de trois comme c'était le cas pour le MNT.

Le texte n'est donc pas inutilement repris ici. Le lecteur est invité à consulter cette section pour obtenir l'information souhaitée. La Figure 4.5-3 donne tout de même un aperçu de l'interface usager présentée à l'utilisateur avec l'arborescence à deux niveaux.

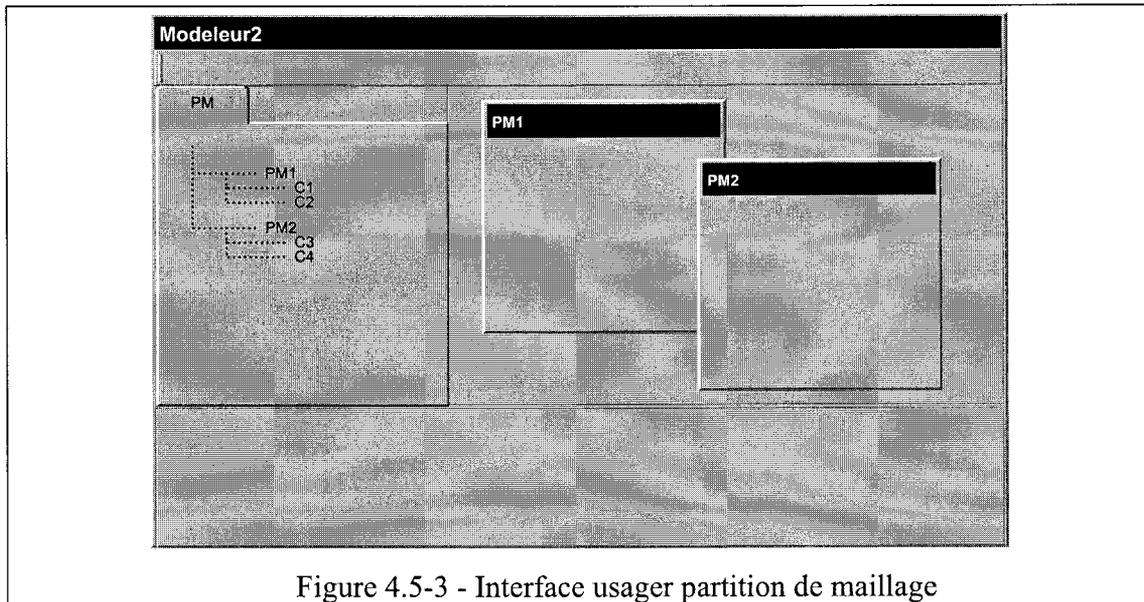


Figure 4.5-3 - Interface usager partition de maillage

## 4.5.5 Gestion d'une partition de maillage

### 4.5.5.1 Nouveau

L'utilisateur peut créer une nouvelle partition de maillage vide (sans couches). Tout ce que l'utilisateur peut faire à cette étape, c'est d'ajouter des couches à sa partition de maillage.

### 4.5.5.2 Ouvrir

L'utilisateur peut ouvrir une partition de maillage existante. Toutes les couches de maillage contenues dans cette partition sont également ouvertes. L'utilisateur peut alors naviguer dans sa partition et interagir avec son contenu.

### 4.5.5.3 Fermer

L'utilisateur peut fermer une partition de maillage. Si celle-ci a été modifiée, l'utilisateur se voit demander s'il désire l'enregistrer.

#### **4.5.5.4 Enregistrer**

L'utilisateur peut enregistrer la partition de maillage courante, ce qui a pour effet d'enregistrer les paramètres de la partition et de ses couches ainsi que la géométrie des couches. Les maillages qui sont affichés dans les couches ne sont pas enregistrés.

L'opération s'effectue sur les entrées existantes de la base de données. Donc, du point de vue de la base de données, on a affaire à une mise à jour (*update*). Si certaines couches ne sont pas présentes, elles sont ajoutées à la base de données dans l'opération.

#### **4.5.5.5 Enregistrer sous**

L'utilisateur peut enregistrer la partition de maillage en cours sous un nom qu'il spécifie. Il doit également spécifier la projection SIG de sa partition de maillage.

Il y a création de nouvelles entrées pour la partition de maillage et de ses couches. Du point de vue de la base de données, on a affaire à un ajout (*insert*). Les couches contenues dans la partition de maillage en cours sont copiées dans la nouvelle partition de maillage.

### **4.5.6 Fonctionnalités des partitions de maillage**

#### **4.5.6.1 Ajouter une nouvelle couche de maillage**

L'utilisateur peut ajouter une couche à sa partition de maillage. Celle-ci se voit attribuer par défaut un ordre de priorité maximum.

#### **4.5.6.2 Ajouter une couche de maillage existante**

Bien qu'une couche de maillage ait obligatoirement à être liée à une partition de maillage, l'utilisateur peut choisir d'ajouter à sa partition de maillage en cours une couche existante issue d'une autre partition de maillage. La couche ajoutée à la partition de maillage est une copie de la couche existante. En ce sens, aucun lien n'est gardé vers la couche source et tout changement fait à la nouvelle couche restera local à la partition de maillage en cours.

#### **4.5.6.3 Retirer une couche de maillage**

L'utilisateur peut choisir d'enlever une couche de sa partition de maillage. Lors du retrait d'une couche, les couches de la partition sont réordonnées en fonction de leur priorité.

#### **4.5.6.4 Copier et coller une couche de maillage**

L'utilisateur peut copier une couche d'une partition et la coller sur une autre ou bien sur la partition courante. L'opération aura pour effet de créer une nouvelle couche, d'y dupliquer le contenu de la couche source et d'ajouter celle-ci à la partition cible.

#### **4.5.6.5 Modifier le nom d'une couche de maillage**

L'utilisateur peut modifier le nom d'une couche de maillage. Le changement affecte la métadonnée associée à la couche. Quand l'utilisateur visite l'activité Gestion des données (voir 4.2) le nom apparaît également comme étant modifié.

#### **4.5.6.6 Modifier la priorité d'une couche de maillage**

L'utilisateur peut modifier la priorité d'une couche de maillage. L'opération peut être réalisée en incrémentant / décrémentant la priorité d'une couche, ou bien en effectuant une opération de «glisser – déposer».

#### **4.5.6.7 Modifier la couleur d'affichage des couches de maillage**

L'utilisateur peut modifier la couleur d'affichage d'une couche de maillage.

#### **4.5.6.8 Afficher le squelette d'un MNT**

L'utilisateur a la possibilité de voir le « squelette d'un MNT ». Cela peut lui permettre de définir son maillage de manière cohérente avec la zone de données couverte par ce MNT. Il faut bien voir que cette fonctionnalité est optionnelle et que l'utilisateur n'est aucunement tenu de l'utiliser pour construire sa partition de maillage.

#### **4.5.6.9 Éditer une couche de maillage**

L'utilisateur construit lui-même un polygone en reliant des segments de droite entre eux à l'aide de l'éditeur de sous-domaines. L'utilisateur peut construire son sous-domaine en utilisant comme point de départ la ligne de contour du squelette de MNT.

L'utilisateur a la possibilité d'annuler toute modification géométrique faite pendant l'édition d'une couche à sa sortie du mode édition.

Lorsque l'utilisateur modifie la géométrie d'une couche, toutes les couches touchées par la modification géométrique voient leur maillage invalidé.

#### **4.5.6.10 Mailler une couche**

L'utilisateur peut mailler une ou plusieurs couches. Suite à cette action, un maillage est affiché dans la couche ou les couches qui ont été maillées. S'il y avait un maillage dans la couche, celui-ci est invalidé.

L'utilisateur a accès en temps réel à des informations durant l'opération de maillage, notamment le nombre de nœuds et le nombre d'éléments.

#### **4.5.6.11 Voir le maillage précédent (maillage invalidé)**

L'utilisateur peut encore voir les maillages qui deviennent invalidés, mais en arrière-plan, c'est-à-dire que ceux qui étaient affichés en avant plan sont maintenant affichés en arrière-plan. Ils sont alors représentés par une couleur à être déterminée.

#### **4.5.6.12 Modifier les paramètres d'une partition**

L'utilisateur doit spécifier le mailleur à utiliser. Le mailleur en question peut également nécessiter l'assignation des paramètres propriétaires. Lorsque l'utilisateur change le mailleur actif, toutes les couches qui contenaient des maillages sont invalidées.

#### **4.5.6.13 Modifier les paramètres d'une couche**

L'utilisateur peut spécifier les paramètres qui sont propres à sa couche. Certains de ses paramètres sont dictés par le choix du mailleur.

Lorsque l'utilisateur modifie les paramètres d'une couche, toutes les couches touchées par la modification géométrique voient leur maillage invalidé.

Certains mailleurs fonctionnent avec une taille de maille spécifiée à l'aide du grand axe, du petit axe et de l'inclinaison, ce qui permet de construire un maillage orienté. La taille des mailles peut être spécifiée globalement sur la couche ou bien aux sommets de chacun des sous domaines de cette couche. L'utilisateur a également la possibilité d'ajouter des points de contrôle sur sa couche, lesquels deviennent des nœuds du maillage. Les points peuvent être ajoutés de façon interactive par l'utilisateur ou encore via un moyen externe (ex : fichier de points).

D'autres mailleurs fonctionnent à l'aide d'un champ d'ellipse erreur jumelé à un critère de niveau de tolérance (marge d'erreur acceptable). La taille des mailles est ainsi ajustée jusqu'à ce qu'à ce soit atteint le niveau de tolérance spécifié par l'utilisateur.

#### **4.5.6.14 Trouver le niveau de tolérance optimal**

Dans le cas de couches maillées à partir du champ d'erreur, l'utilisateur peut utiliser un histogramme de répartition d'erreur pour l'aider à choisir un niveau de tolérance à l'erreur acceptable. Cela lui permet de trouver la combinaison niveau d'erreur / nombre de mailles optimal.

#### **4.5.6.15 Obtenir de l'information sur le maillage généré**

L'utilisateur peut obtenir de l'information sur sa partition de maillage, notamment le nombre de nœuds et d'éléments. L'information peut être obtenue individuellement pour chaque couche ou bien pour un groupe de couches.

#### 4.5.6.16 Générer le maillage final

Quand l'utilisateur est satisfait de la partition de maillage qu'il a construite, il peut procéder à la génération du maillage. Les maillages partiels associés à chacune des couches sont alors rassemblés pour former un maillage complet. Le maillage résultant est cohérent, c'est-à-dire qu'il ne contient pas de nœuds dédoublés (doublons) et que ses éléments sont valides.

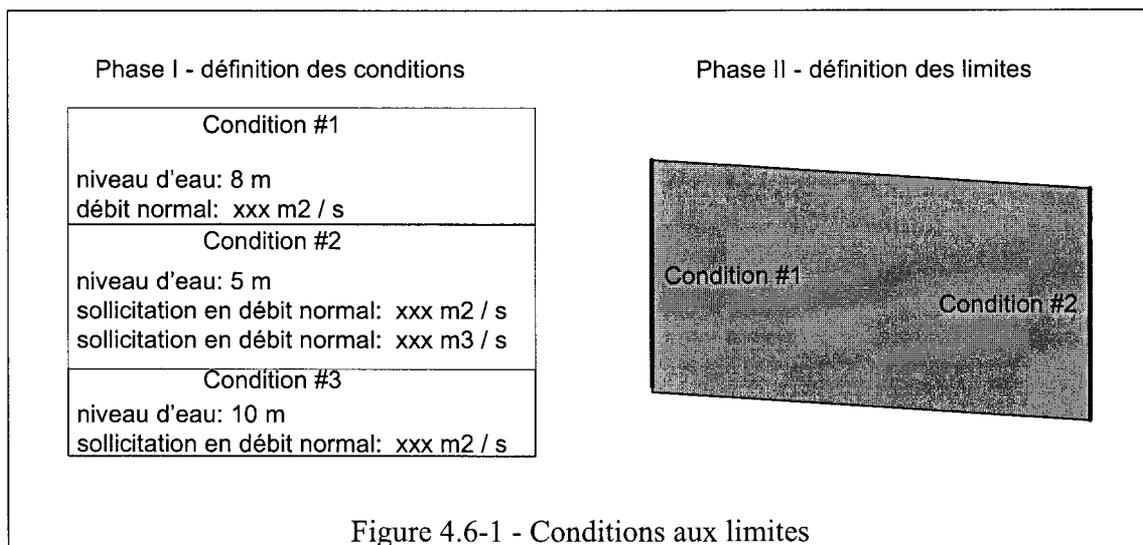
### 4.6 Partition de conditions aux limites

L'utilisateur spécifie des conditions aux limites de son domaine. Le modèle de conditions aux limites est spécifié en deux phases.

La première phase vise à spécifier la partie « condition » et la seconde vise à spécifier la partie « limites ». Ces deux phases amènent la construction d'une partition de conditions aux limites.

L'utilisateur peut spécifier la partie condition en ajoutant, enlevant et modifiant des conditions. Ces conditions se voient attribuer un nom, lequel nom est utilisé pour associer les dites conditions avec les limites de la partition.

La Figure 4.6-1 illustre le modèle des conditions aux limites.



Les conditions sont bien sûr couplées de très près avec le simulateur hydrodynamique, c'est-à-dire que chaque simulateur (ou plutôt chaque application d'un simulateur) a son jeu de conditions possibles.

L'utilisateur peut créer des conditions non stationnaires sous forme de graphe. Il peut également construire des conditions en débit réparties sur plusieurs arêtes.

Les conditions aux limites du domaine peuvent être spécifiées soit par arête, soit par sommets. Quand les conditions sont spécifiées sur les sommets, il y a interpolation de la condition sur l'arête reliant les sommets. Quand la condition est spécifiée par arête, la valeur est constante partout sur l'arête.

## 4.6.1 Interface usager

### 4.6.1.1 Partition de conditions aux limites

À la création ou à l'ouverture d'une partition de conditions aux limites, un composant de type « arborescence » est ajouté à la zone de contrôle. L'arborescence héberge toutes les partitions de conditions aux limites ouvertes par l'utilisateur. L'utilisateur peut consulter chacune des conditions aux limites contenue dans une partition. La Figure 4.6-2 donne un aperçu de l'interface usager proposée.

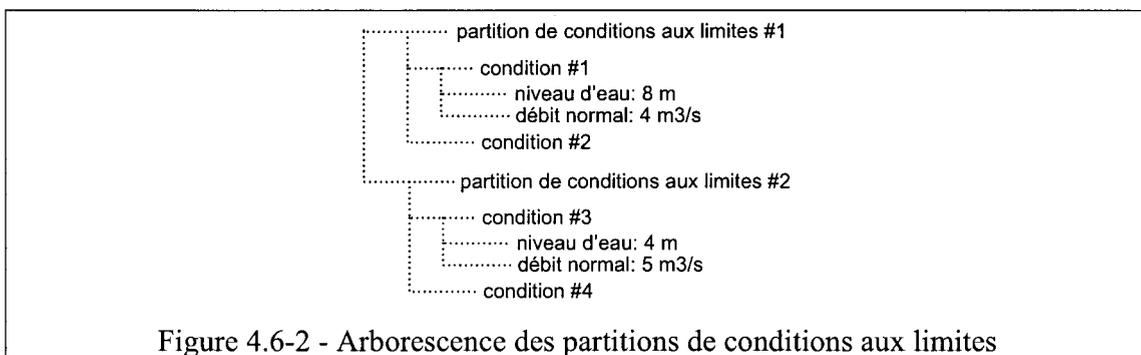


Figure 4.6-2 - Arborescence des partitions de conditions aux limites

Lors de la création d'une partition de conditions aux limites, une fenêtre graphique est également créée. La fenêtre affiche le contour de la partition et les conditions que l'utilisateur a associées à chacune des limites.

La Figure 4.6-3 donne un aperçu de l'interface proposée à l'utilisateur.

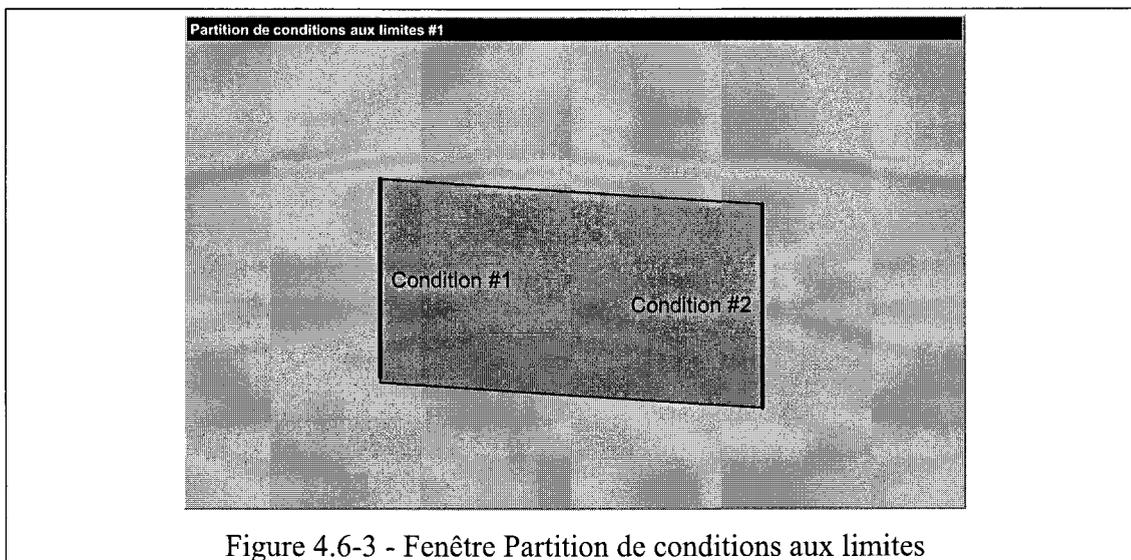


Figure 4.6-3 - Fenêtre Partition de conditions aux limites

Quand l'utilisateur sélectionne une condition dans l'arborescence, ceci a pour effet de sélectionner les limites de la partition sur lesquelles cette condition s'applique.

Lorsque l'utilisateur effectue un double clic sur une des limites de sa partition, il provoque l'affichage d'une boîte de dialogue. Dans cette boîte de dialogue, l'utilisateur voit toutes les conditions connues de sa partition de conditions aux limites. Les conditions proposées à l'utilisateur sont celles hébergées dans le projet et qui sont compatibles avec le simulateur associé à cette partition de conditions aux limites.

S'il avait auparavant assigné une condition à cette limite, celle-ci apparaît dans l'état « sélectionné » dans la liste des conditions.

L'utilisateur peut créer de nouvelles conditions ou encore en ajouter en provenance d'autres partitions.

## 4.6.2 Gestion des partitions de conditions aux limites

### 4.6.2.1 Nouveau

L'utilisateur peut créer une partition de conditions aux limites. Il doit spécifier la partition de maillage qui sera utilisée comme contour de la partition de conditions aux limites.

La partition de maillage qui est spécifiée doit être la même qui est à l'origine du maillage sur lequel seront portées les données de la partition de conditions aux limites.

L'utilisateur doit également spécifier le simulateur pour lequel il crée cette partition de conditions aux limites.

#### **4.6.2.2 Ouvrir**

L'utilisateur peut ouvrir une partition de conditions aux limites existante. À l'ouverture, la partition de condition aux limites est affichée dans une fenêtre graphique.

L'utilisateur peut alors sélectionner chacune des limites de la partition, et lui assigner une condition.

#### **4.6.2.3 Fermer**

L'utilisateur peut fermer une partition de conditions aux limites. Si elle n'a pas encore été enregistrée, l'utilisateur a l'opportunité de le faire.

#### **4.6.2.4 Enregistrer**

L'utilisateur peut enregistrer une partition de conditions aux limites. Toutes les conditions contenues dans la partition de conditions aux limites sont également enregistrées. Le lien entre chacune des limites et la condition qui lui est associé est également enregistré.

#### **4.6.2.5 Enregistrer sous...**

L'utilisateur peut enregistrer une partition de conditions aux limites sous un autre nom.

### **4.6.3 Fonctionnalités liées aux partitions de conditions aux limites**

#### **4.6.3.1 Ajouter une nouvelle condition**

L'utilisateur peut ajouter de nouvelles conditions à sa partition de conditions aux limites. Les types de conditions permis dépendent du simulateur et du type de la simulation.

Par exemple, pour une simulation hydrodynamique avec H2D2, une condition peut être du type « niveau d'eau » alors que pour une simulation de température, la condition est plutôt de type « température ». C'est au plug-in du simulateur d'offrir les moyens de construire les conditions aux limites qu'il supporte.

Les conditions créées par l'utilisateur ne sont pas associées aux limites de la partition de conditions aux limites tant que l'utilisateur n'effectue pas une opération d'assignation.

#### **4.6.3.2 Ajouter une condition existante**

L'utilisateur peut ajouter une condition existante tirée d'une autre partition de conditions aux limites. La condition est alors copiée dans la partition courante. Toute modification à cette condition n'entraînera aucun changement dans l'autre. La seule contrainte est que la condition doit être compatible avec le simulateur et le type de simulation associé à la partition de conditions aux limites en cours.

#### **4.6.3.3 Assigner une condition**

En double-cliquant sur une limite d'une partition de conditions aux limites, l'utilisateur peut spécifier les conditions qu'il désire appliquer sur cette limite. Les limites correspondent aux arêtes ou aux sommets de la partition. Les conditions qui sont offertes à l'utilisateur sont toutes celles qui ont été créées ou ajoutées dans cette partition.

#### **4.6.3.4 Modifier une condition**

L'utilisateur peut modifier une condition qu'il a créée ou ajoutée auparavant. Les paramètres des conditions aux limites dépendent du simulateur et du type de la simulation.

#### **4.6.3.5 Effacer une condition**

L'utilisateur peut effacer une condition parmi celles connues par sa partition de conditions aux limites. Si l'utilisateur efface une condition utilisée dans l'une des limites de sa partition, l'utilisateur se voit averti, et s'il désire tout de même procéder, les limites qui étaient associées à cette condition sont désormais sans condition. L'opération entraîne également l'effacement de la condition dans la base de données.

### **4.7 Activité : Assemblage des données à partir du maillage**

Les données spécifiées lors des différentes activités de construction de partitions peuvent être portées sur un maillage lors de l'activité d'assemblage.

Les partitions d'un MNT peuvent être portées sur un maillage de façon à obtenir des champs pour chacun des aspects du MNT (topographie, substrat, etc.). Pour le MNT, l'opération d'assemblage devrait pouvoir s'exécuter avec succès si le maillage a été construit en respectant les limites du squelette du MNT.

Les partitions de conditions aux limites, elles, peuvent être portées sur un maillage pour produire des conditions aux limites.

L'opération d'assemblage peut se faire sur plusieurs données simultanément. Par exemple, elle peut être réalisée sur plusieurs partitions d'un même MNT, sur des MNT différents ou encore sur des partitions de conditions aux limites.

L'utilisateur choisit d'abord son maillage. Il choisit ensuite les partitions qu'il désire assembler. Finalement, il attribue un nom à chacune des données qui résultera de l'opération d'assemblage. L'utilisateur peut spécifier ces informations pour tous les champs qu'il désire assembler et lancer l'opération d'assemblage en une seule étape.

Si, pour une raison ou une autre, il y a une erreur lors de l'assemblage d'une ou plusieurs partitions, l'utilisateur peut aller corriger sa partition ou son maillage et répéter l'opération sans que la boîte de dialogue n'ait été fermée. L'utilisateur se voit fournir de l'information pouvant l'aider à identifier le problème rencontré. Les données qui ont été assemblées correctement sont enregistrées dans la base de données et peuvent être consultées. Les données qui ont été assemblées avec succès sont marquées d'une façon différente de celles pour laquelle il y a eu une erreur lors de l'assemblage.

### 4.7.1 Interface usager

L'utilisateur doit préciser le maillage sur lequel il désire porter ses données. Il doit également spécifier les partitions qu'il désire assembler parmi toutes celles disponibles dans son projet.

Finalement, il doit spécifier le nom qu'il désire donner à chaque donnée qui sera issue de l'assemblage de la partition sur le maillage. La Figure 4.7-1 illustre sommairement le concept d'interface qui s'offre à l'utilisateur lors du processus d'assemblage.

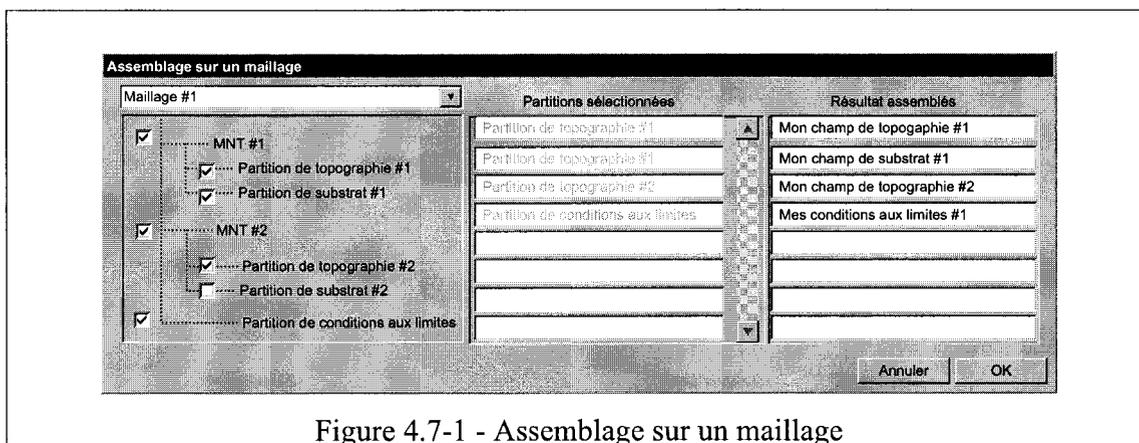


Figure 4.7-1 - Assemblage sur un maillage

## 4.8 Activité : Simulation

### 4.8.1 Simulation

Une simulation est une suite d'étapes de calcul (ex : hydrodynamique) qui a pour but d'obtenir une solution finale satisfaisante. Chaque calcul est réalisé à l'aide de paramètres

d'entrée et produit une série de résultats (ex : champ de niveaux d'eau et champs de vitesses). Les résultats d'une étape de calcul peuvent à leur tour servir de paramètres d'entrée (i.e. solution initiale) à une autre étape de calcul.

Une simulation n'est pas nécessairement continue, c'est-à-dire que le chemin à suivre pour arriver à la solution finale n'est pas nécessairement exempt de retours en arrière. Pour arriver à gérer ceux-ci, l'utilisateur a accès à un historique sous forme d'un graphe présentant toutes les étapes de calcul qui furent suivies.

Si, pour une raison quelconque, un chemin de simulation ne mène nul part, il est possible pour l'utilisateur de repartir à une étape de calcul précédente. De là, il est possible de revisiter les paramètres d'entrée de ce calcul et de le reprendre. Il est également possible de reprendre le calcul à partir de toutes nouvelles conditions initiales. La Figure 4.8-1 illustre un cas où après un calcul #3 insatisfaisant, l'utilisateur décide de reprendre à l'étape suivant la fin du calcul #2 pour finalement arriver à une solution finale au calcul #4.

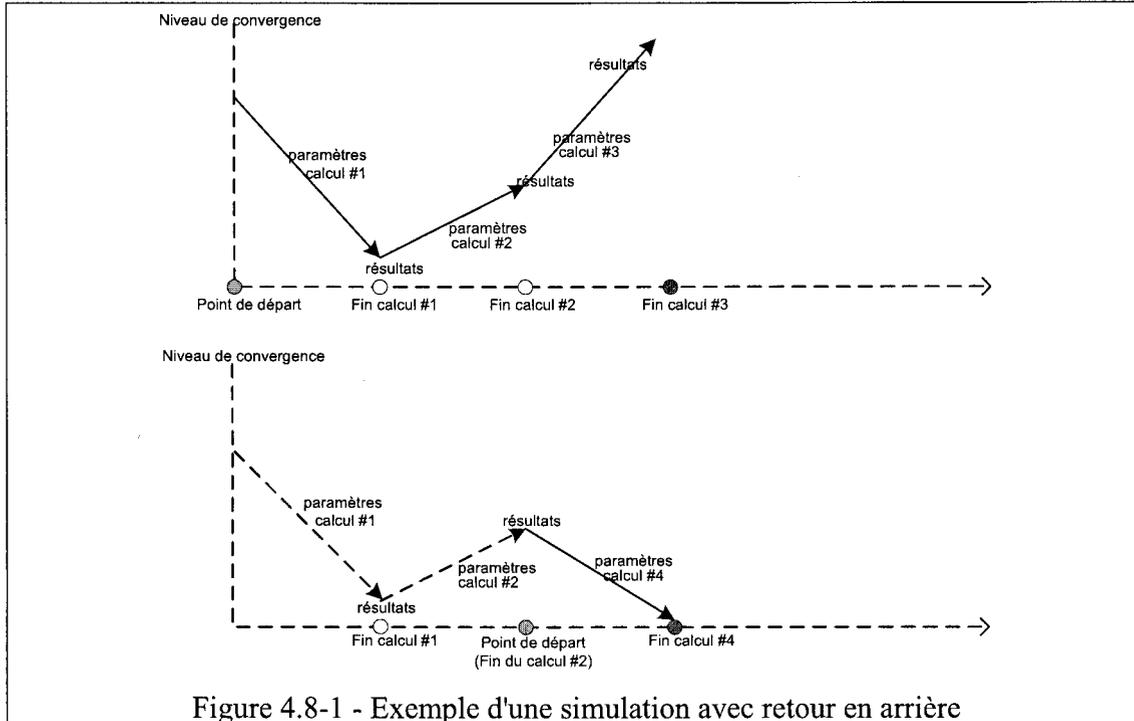


Figure 4.8-1 - Exemple d'une simulation avec retour en arrière

Un des objectifs poursuivis est de pouvoir analyser une simulation et ses étapes de calculs clés afin de déceler les raisons d'une divergence possible. Un autre objectif est de pouvoir aider les usagers externes à distance en analysant leur simulation.

## 4.8.2 Simulateur

Un calcul de simulation est réalisé à l'aide d'un simulateur. Un simulateur peut être vu comme une boîte noire qui reçoit des paramètres en entrée et qui en ressort des résultats.

Les simulateurs doivent fournir à Modeleur2 un moyen d'exporter les données d'entrée au simulateur et d'importer les résultats vers Modeleur2.

Le même simulateur peut être utilisé de plusieurs manières différentes, c'est-à-dire dans plusieurs types d'application. Par exemple, H2D2, le simulateur de base fourni avec Modeleur2, peut être utilisé pour faire des calculs hydrodynamiques, des calculs de température, des calculs de transport-diffusion, etc. Chaque utilisation d'un simulateur peut demander des paramètres d'entrée et produire des résultats différents l'un à l'autre.

Les simulateurs sont des plug-in que l'on peut rattacher à Modeleur2. Le logiciel offre tous les mécanismes pour pouvoir intégrer plusieurs simulateurs, et n'est donc pas spécifiquement dédié à un simulateur particulier. Pour cette raison, le présent document n'explique pas les paramètres propres à chaque simulateur. C'est donc aux plug-in qui implémentent les simulateurs de proposer des moyens à l'utilisateur pour faire connaître les paramètres d'entrée.

Les simulateurs peuvent être contrôlés pour effectuer des calculs à distance. Pour cela, une configuration doit être faite pour faire connaître les machines distantes à utiliser et leurs paramètres.

Les calculs des simulateurs peuvent être exécutés dans des processus détachés avec possibilité d'en diminuer la priorité, ce qui permet à l'utilisateur de pouvoir continuer à travailler sans voir sa machine bloquée ou indûment ralentie. Des mécanismes sont également mis en place pour que chaque simulateur puisse faire retourner des informations sur l'avancement de son calcul. L'utilisateur a donc accès à cette rétroaction (*feedback*) en temps réel lors de l'exécution d'un calcul.

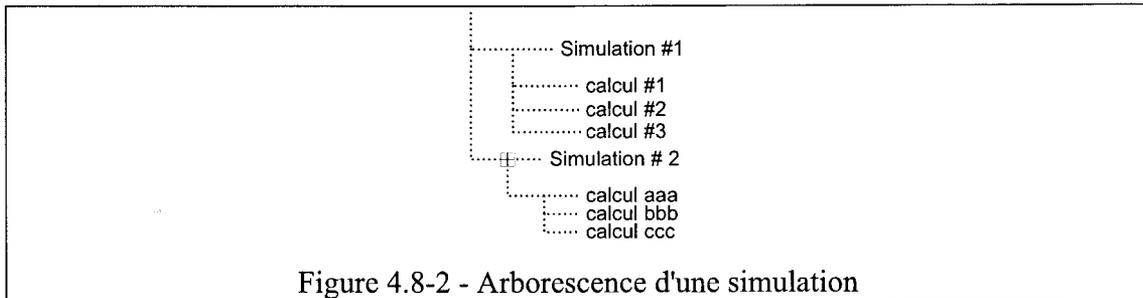
Les calculs maintiennent des dépendances sur les données qui sont nécessaires à leur exécution. Si une donnée dont dépend une simulation est sur le point d'être modifiée ou détruite, l'utilisateur se voit averti que la chaîne de dépendance de la simulation sera brisée.

### **4.8.3 Interface usager**

#### **4.8.3.1 Simulation**

À la création ou à l'ouverture d'une simulation, un composant de type « arborescence » est ajouté à la zone de contrôle et une fenêtre graphique est créée. L'arborescence héberge l'historique des calculs de la simulation.

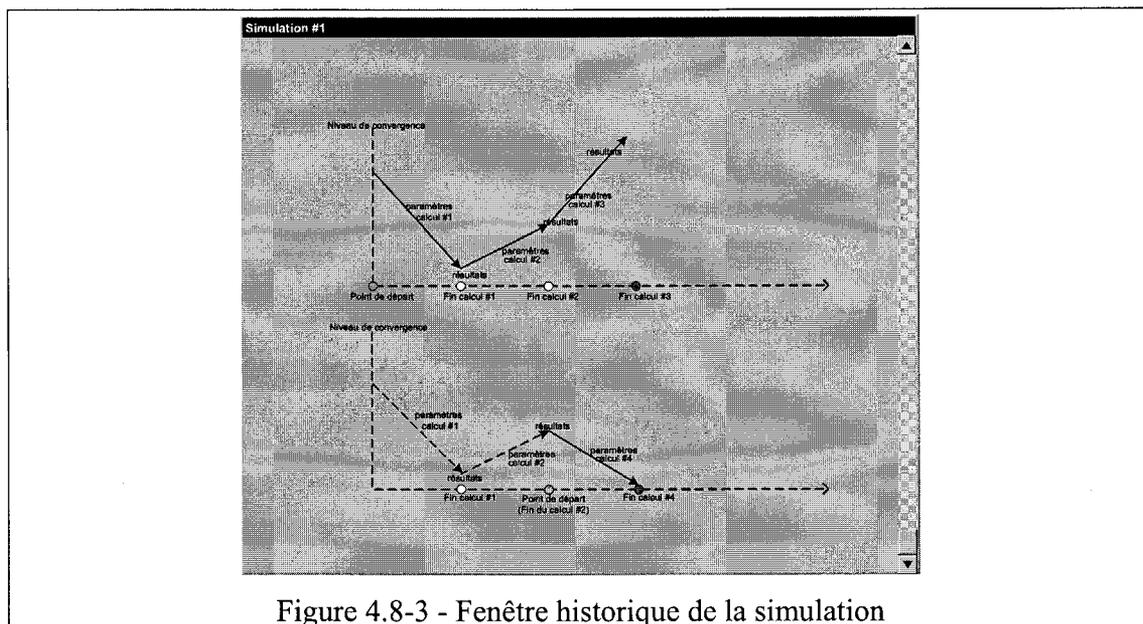
La Figure 4.8-2 donne un aperçu de l'interface usager proposée.



L'utilisateur a accès aux paramètres de chacun des calculs réalisés jusqu'ici dans le cadre de sa simulation. Il ne peut toutefois plus les modifier une fois l'étape de calcul exécutée.

Lorsque l'utilisateur double-clique sur la zone Calculs, il a accès à une vue de l'historique de sa simulation. La

Figure 4.8-3 donne un aperçu de l'interface usager offerte alors. Il est à noter que certaines fonctionnalités illustrées (ex : le niveau de convergence) dépendent de ce que le simulateur peut offrir comme donnée correspondante.



## **4.8.4 Gestion d'une simulation**

### **4.8.4.1 Nouveau**

L'utilisateur peut créer une nouvelle simulation. Il doit alors spécifier le simulateur.

### **4.8.4.2 Ouvrir**

L'utilisateur peut ouvrir une simulation existante. Il peut alors naviguer à travers tous les calculs contenus dans cette simulation.

### **4.8.4.3 Fermer**

L'utilisateur peut fermer une simulation en tout temps, même lorsqu'un calcul est en cours d'exécution. Si, à sa fermeture, une simulation est en cours d'exécution d'un calcul, les résultats associés à cette simulation pourront être importés à la prochaine ouverture de la simulation. L'utilisateur se verra alors demander s'il désire importer les résultats maintenant rendus disponibles.

À la fermeture d'une simulation, l'utilisateur se voit demander s'il désire enregistrer cette simulation.

### **4.8.4.4 Enregistrer**

L'utilisateur peut enregistrer une simulation. Ceci aura pour effet d'enregistrer les calculs et leurs paramètres, de même que l'historique de la simulation.

### **4.8.4.5 Enregistrer sous**

L'utilisateur peut enregistrer une simulation sous un autre nom. Ceci a pour effet de dupliquer la simulation. Ceci aura pour effet d'enregistrer les calculs et leurs paramètres, de même que l'historique de la simulation.

## **4.8.5 Fonctionnalités liées aux simulations**

### **4.8.5.1 Créer un calcul**

L'utilisateur peut créer un nouveau calcul pour sa simulation. Il se voit alors offrir toute la gamme des paramètres nécessaires à la préparation de ce calcul. Les paramètres en question dépendent du simulateur et du type de calcul.

Parmi les paramètres d'entrée, l'utilisateur doit spécifier les conditions aux limites qui ont été assemblées (voir Figure 4.6-1) de son domaine. Il a alors le choix parmi toutes les conditions aux limites disponibles dans son projet.

L'utilisateur est appelé à nommer son calcul et c'est ce nom qui sera réutilisé dans l'arborescence et dans l'historique de la simulation. L'utilisateur peut également déterminer les résultats qu'il désire importer suite à ce calcul.

L'item calcul qui vient d'être créé sera dûment identifié comme n'ayant pas encore été exécuté.

L'utilisateur a la possibilité de créer un calcul à partir des résultats d'un calcul précédent. Il peut choisir la fin d'un calcul antérieur comme point de démarrage pour tout nouveau calcul qu'il crée. L'utilisateur peut interagir avec les points de l'historique, choisir un point et modifier les paramètres d'entrée du calcul correspondant à ce point.

L'utilisateur peut distinguer les calculs qui ont été déjà exécutés de ceux qui ne l'ont pas encore été. Le moyen utilisé pour distinguer les calculs « exécutés » de ceux « non exécutés » reste à déterminer (couleur, symbole ou autre).

#### **4.8.5.2 Modifier les paramètres d'un calcul**

Tant qu'un calcul n'a pas été exécuté, l'utilisateur peut modifier les paramètres d'entrée de ce calcul.

#### **4.8.5.3 Exécuter un calcul**

L'utilisateur peut démarrer tout calcul qui n'a pas encore été exécuté. Ceci produit éventuellement une série de résultats en sortie.

Tout au long du calcul, l'utilisateur reçoit un *feedback* sur son déroulement (ex : la valeur du résidu). Une fois le calcul terminé, l'item calcul dans l'arborescence est identifié comme « exécuté », et ne peut désormais qu'être consulté ou servir de point de départ pour un nouveau calcul. Le calcul qui vient d'être exécuté apparaît désormais dans l'historique.

#### **4.8.5.4 Importer les résultats d'un calcul**

L'utilisateur peut importer les résultats d'un calcul dans Modeleur2 (ex : champ scalaire de niveau d'eau). L'utilisateur doit spécifier le nom à assigner au résultat qu'il importe. Les résultats qui ne sont pas importés ne peuvent être consultés ou utilisés par l'utilisateur *via* les fonctionnalités de Modeleur2. L'utilisateur peut à tout moment importer les résultats d'un calcul passé. Si ces résultats ont été effacés, l'utilisateur en est prévenu.

#### **4.8.5.5 Interrompre un calcul**

En tout temps, l'utilisateur peut interrompre un calcul en cours d'exécution. Ceci a pour effet d'ignorer tout résultat intermédiaire issu de ce calcul, et de revenir au point de départ, avant le début du calcul. Le calcul est alors considéré comme « non exécuté », au même titre qu'un calcul venant d'être créé.

#### **4.8.5.6 Renommer un calcul**

L'utilisateur peut renommer un calcul. Ceci aura pour effet de modifier le nom utilisé pour référer à ce calcul dans l'historique.

#### **4.8.5.7 Naviguer l'historique de la simulation**

L'utilisateur peut naviguer dans l'historique d'une simulation et ainsi visualiser les paramètres des calculs passés et l'évolution du niveau de convergence à travers chacune des étapes de calcul.

Lorsque l'utilisateur double-clique sur un item calcul dans l'arborescence, la fenêtre Historique est activée et le point correspondant au calcul est sélectionné.

### **4.9 Publication des résultats (habillage d'écran)**

L'utilisateur a accès à des fonctionnalités pour ajouter des composants visuels à son affichage pour fin de présentation (publication) par exemple.

#### **4.9.1 Gestion des habillages d'écran**

##### **4.9.1.1 Nouveau**

L'utilisateur peut créer un nouvel habillage d'écran. À la création, l'habillage d'écran est vide. L'utilisateur doit donc par la suite y ajouter des composants.

##### **4.9.1.2 Ouvrir**

L'utilisateur peut ouvrir un habillage d'écran enregistré au préalable.

##### **4.9.1.3 Enregistrer**

L'utilisateur peut enregistrer un habillage d'écran. Tous les composants qu'il a créés sont enregistrés dans l'opération.

##### **4.9.1.4 Enregistrer sous...**

L'utilisateur peut enregistrer un habillage d'écran sous un autre nom. Tous les composants qu'il a créés sont enregistrés dans l'opération.

## **4.9.2 Fonctionnalités reliées aux habillages d'écran**

### **4.9.2.1 Ajouter**

L'utilisateur peut donc ajouter les composants suivants :

- flèche du nord;
- légende;
- barre d'échelle;
- texte;
- grille de repérage.

Certains éléments peuvent être en coordonnées écran et d'autres être géoréférencés. Un élément en coordonnées écran suivra le déplacement lors d'une translation (ex : la flèche du nord), alors qu'un élément géoréférencé restera à une position fixe (ex : un toponyme).

### **4.9.2.2 Effacer**

L'utilisateur peut effacer chaque composant de son habillage d'écran.

### **4.9.2.3 Déplacer**

L'utilisateur peut déplacer à sa guise les composants de son habillage d'écran.

### **4.9.2.4 Copier / coller**

L'utilisateur peut copier et coller les composants de son habillage d'écran d'une fenêtre graphique à l'autre.

## 5 Outils d'analyse

### 5.1 L'affichage

Un *espace de représentation* correspond au contenu d'une fenêtre graphique. Un espace de représentation est composé de trois plans : plan arrière, plan médian et plan interactif. Un plan peut héberger un nombre indéterminé d'objets graphiques.

Un objet graphique correspond à la représentation visuelle d'une donnée. Il possède les moyens de convertir en primitives graphiques la donnée à laquelle il est lié. Ces primitives graphiques peuvent ensuite être envoyées à l'affichage. Un objet graphique possède les attributs suivants: *affichable*, *imprimable* et *une échelle de couleurs*. Ces attributs peuvent être modifiés par l'utilisateur. L'utilisateur interagit avec une liste d'affichage qui contient tous les espaces de représentation qui ont été créés (i.e. toutes les fenêtres). L'utilisateur peut y voir les objets graphiques contenus dans chaque plan et interagir avec eux.

Un objet graphique maintient une référence vers une donnée, mais ne contient pas cette donnée. La donnée en question peut avoir été modifiée entre l'étape de l'enregistrement et de la réouverture de l'objet graphique. Il n'y a donc pas de garanties d'obtenir le même résultat visuel puisque la donnée source peut être différente.

Par exemple, lors l'enregistrement d'un objet graphique représentant une couche d'une partition de maillage, la géométrie de la couche n'est pas enregistrée. Il est possible que cette géométrie soit modifiée par la suite lors d'une étape d'édition de la couche. Si l'utilisateur tente de recharger l'objet graphique qu'il a enregistré au tout début, celui-ci sera chargé avec la géométrie courante, et non pas celle qu'il avait au moment de l'enregistrement.

La liste d'objets graphiques est non exhaustive car chaque plug-in peut ajouter de nouveaux objets graphiques au système. Les prochaines lignes présentent néanmoins quelques types d'objets graphiques retrouvés dans le logiciel. Il existe des types d'objet graphique pour représenter des maillages, des géométries et des points. Ces objets sont utilisés dans certaines des activités de Modeleur2. Il existe un type d'objet graphique dit « composite » qui est en fait un assemblage d'autres objets graphiques. Par exemple, ce type particulier d'objet graphique est utilisé dans la partition de maillage pour héberger les objets graphiques qui correspondent aux géométries des couches de maillage.

Il existe un objet graphique pour représenter une image raster géoréférencée. Cet objet graphique est typiquement utilisé pour contenir des arrière plans (ex : carte des routes, photographie aérienne). Il est ainsi possible de faire afficher cet objet par défaut dans le plan arrière de toutes les fenêtres graphiques ou encore de le faire afficher dans une fenêtre spécifique. L'image géoréférencée peut également provenir d'une capture d'écran fait dans une fenêtre graphique.

### 5.1.1 Interface usager

La présente section illustre l'interface de la *Liste d'affichage* avec laquelle l'utilisateur interagit pour gérer le contenu de ses fenêtres graphiques. Les objets graphiques qui se retrouvent au bas de la liste sont ceux qui seront affichés en dernier, et donc qui apparaîtront visuellement par dessus les autres (avant-plan). Voici un exemple d'une liste d'affichage contenant deux fenêtres graphiques (Figure 5.1-1). Les deux fenêtres graphiques contiennent un plan arrière contenant un objet graphique de type image géoréférencée représentant une carte routière. Dans la fenêtre #1, l'objet graphique est actif. On voit donc la carte routière. Dans la fenêtre #2, l'objet graphique est grisé (i.e. non affiché). On ne voit donc pas la carte de routes.

Dans la plan médian de la fenêtre #1, on retrouve en premier lieu un objet graphique composite associé à une partition. Étant donné sa position, celui-ci est affiché en premier. Dans cet objet graphique composite, on voit que l'objet graphique représentant la couche #1 (triangle) est affiché en premier, alors que celui représentant la couche #2 (quadrilatère) est affiché par dessus, car il se retrouve plus bas dans la liste d'affichage. Enfin, il y a également un objet graphique maillage qui, lui, est affiché par dessus l'objet graphique de la partition étant donné sa position dans la liste d'affichage. Dans la fenêtre #2, l'objet graphique maillage est affiché en premier, et ceux des couches sont affichés par dessus dans l'ordre suivant : a (quadrilatère) et b (triangle).

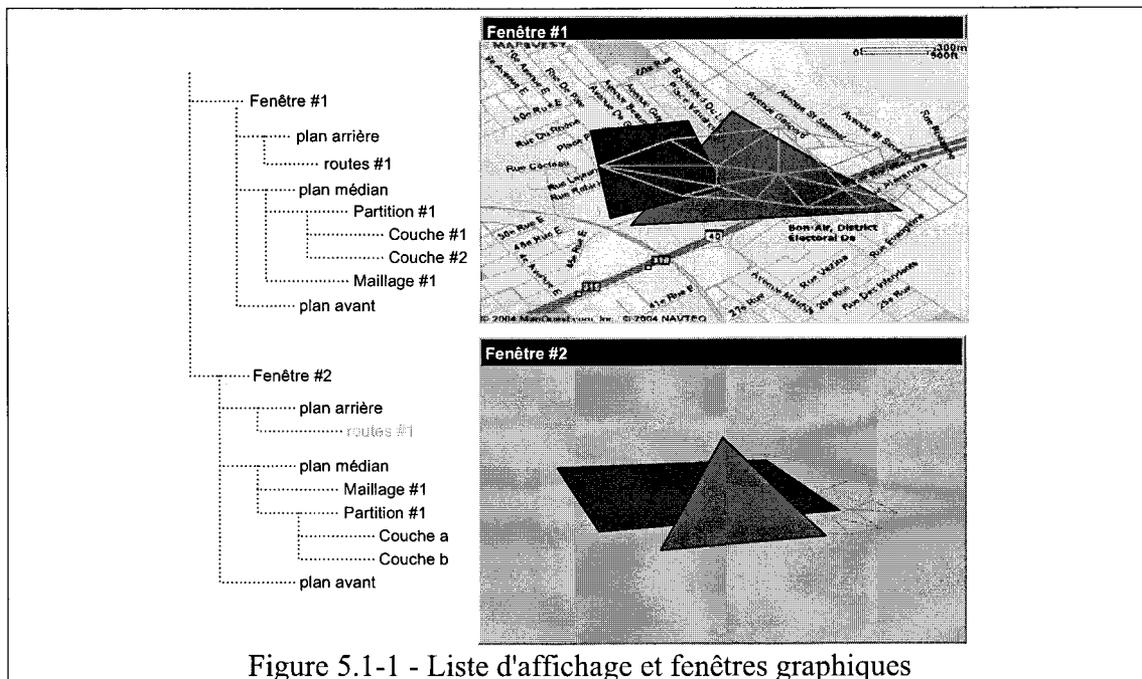


Figure 5.1-1 - Liste d'affichage et fenêtres graphiques

## **5.1.2 Gestion de l’affichage**

### **5.1.2.1 Espace de représentation**

#### **5.1.2.1.1 Ouvrir**

L’usager peut remplacer l’espace de représentation contenu dans une fenêtre graphique par un autre préalablement enregistré. L’espace de représentation courant est alors déchargé de ses plans et de ses objets graphiques. L’usager est averti au préalable, et peut donc, s’il le désire, enregistrer son espace de représentation courant.

#### **5.1.2.1.2 Fermer**

L’usager ferme un espace de représentation implicitement en fermant une fenêtre graphique.

#### **5.1.2.1.3 Enregistrer**

L’usager peut enregistrer un espace de représentation (i.e., le contenu d’une fenêtre graphique). Ceci a pour effet d’enregistrer les plans et les objets graphiques rattachés à cet espace de représentation.

Tel qu’il a été spécifié au début de la section, les données à la source des objets graphiques ne sont pas enregistrées dans l’opération.

#### **5.1.2.1.4 Enregistrer sous...**

L’usager peut enregistrer un espace de représentation sous un nouveau nom. Ceci a pour effet de dupliquer l’espace de représentation dans la base de données, ses plans et les objets graphiques contenus par ses plans.

Tel qu’il a été spécifié au début de la section, les données à la source des objets graphiques ne sont pas enregistrées dans l’opération.

### **5.1.2.2 Plan**

#### **5.1.2.2.1 Ouvrir**

L’usager peut remplacer un plan d’un espace de représentation par un plan du même type qui fut préalablement enregistré. Le plan courant est alors déchargé de ses objets graphiques. L’usager est averti au préalable, et peut donc, s’il le désire, déplacer temporairement ses objets graphiques dans un autre plan par exemple.

#### **5.1.2.2.2 Enregistrer**

L'utilisateur a la possibilité d'enregistrer un plan qui contient des objets graphiques. Ceci a pour effet d'enregistrer les objets graphiques rattachés à ce plan.

Tel qu'il a été spécifié au début de la section, les données à la source des objets graphiques ne sont pas enregistrées dans l'opération.

#### **5.1.2.2.3 Enregistrer sous...**

L'utilisateur a la possibilité d'enregistrer un plan sous un nouveau nom. Ceci a pour effet de dupliquer le plan dans la base de données ainsi que les objets graphiques qu'il contient.

Tel qu'il a été spécifié au début de la section, les données à la source des objets graphiques ne sont pas enregistrées dans l'opération.

### **5.1.2.3 Objet graphique**

#### **5.1.2.3.1 Ouvrir**

L'utilisateur peut ouvrir dans le plan de son choix, un objet graphique qui a été préalablement enregistré. L'objet graphique est alors chargé avec tous ses paramètres. Il est ici question des attributs suivants : *affichable*, *imprimable* et de son *échelle de couleur (légende)*.

#### **5.1.2.3.2 Enregistrer**

L'utilisateur a la possibilité d'enregistrer des objets graphiques.

Tel qu'il a été spécifié au début de la section, les données à la source des objets graphiques ne sont pas enregistrées dans l'opération.

#### **5.1.2.3.3 Enregistrer sous...**

L'utilisateur a la possibilité d'enregistrer un objet graphique sous un nouveau nom. Ceci a pour effet de dupliquer l'objet graphique dans la base de données.

Tel qu'il a été spécifié au début de la section, les données à la source des objets graphiques ne sont pas enregistrées dans l'opération.

### **5.1.3 Fonctionnalités de la liste d’affichage**

#### **5.1.3.1 Modifier un objet graphique**

L’usager peut modifier les attributs d’un objet graphique : *échelle de couleurs*, état *affichable / non affichable*, état *imprimable / non imprimable*. L’usager interagit avec les objets graphiques contenus dans ces plans *via* la liste d’affichage.

#### **5.1.3.2 Retirer un objet graphique**

L’usager peut retirer des objets graphiques des plans de son espace de représentation.

#### **5.1.3.3 Copier / coller / couper un objet graphique**

L’usager peut copier / coller des objets graphiques. L’opération peut se faire localement, à l’intérieur de la même fenêtre ou vers une autre fenêtre.

#### **5.1.3.4 Déplacer un objet graphique**

L’usager peut déplacer les objets graphiques. L’opération peut se faire localement, à l’intérieur de la même fenêtre ou vers une autre fenêtre.

### **5.1.4 Construire des échelles de couleur (légendes)**

L’usager peut construire des échelles de couleur ou légendes. Ces échelles peuvent être utilisées par la suite dans le logiciel à tout endroit où une échelle de couleur est requise. Une échelle de couleur est constituée d’une couleur minimum / maximum, d’une valeur scalaire minimum / maximum et d’une règle qui permet de répartir (mapper) chaque scalaire dans l’intervalle. Par exemple, une échelle de couleur peut être linéaire ou logarithmique.

### **5.1.5 Gestion de la caméra**

#### **5.1.5.1 Configurer les paramètres de gestion de caméra**

L’usager peut configurer le facteur de *zoom-in* et le facteur de *zoom-out*. Il peut également configurer la distance appliquée lors d’une translation.

#### **5.1.5.2 Effectuer une opération de zoom**

L’usager peut effectuer un *zoom-in* ou un *zoom-out*. Le niveau de zoom appliqué dépend des paramètres que l’usager a spécifiés.

### **5.1.5.3 Effectuer une opération de zoom de façon interactive**

L'utilisateur peut activer l'équivalent du mode « loupe » dans certains logiciels pour effectuer des zoom-in ou des zoom-out de façon intuitive. Un facteur de zoom différent peut être spécifié pour cette opération.

### **5.1.5.4 Se déplacer avec des translations (haut / bas, gauche / droite)**

L'utilisateur peut se déplacer dans sa fenêtre graphique en effectuant des translations. Par exemple, en faisant une translation vers le haut l'utilisateur voit une partie de son domaine qui était invisible auparavant. La partie de son domaine qui était au bas de la fenêtre graphique n'est désormais plus visible. Le déplacement dans l'espace d'affichage dépend des paramètres que l'utilisateur a spécifiés.

### **5.1.5.5 Se déplacer de façon interactive**

L'utilisateur peut se déplacer de façon interactive (panoramique) dans la fenêtre d'affichage à l'aide de l'équivalent de « la petite main » façon de faire très utilisée dans les logiciels de dessin. Il peut ainsi naturellement se déplacer dans son domaine sans avoir à toujours faire des opérations de translations.

### **5.1.5.6 Réinitialiser la zone d'affichage**

L'utilisateur peut demander à ce que la zone d'affichage soit remise à la valeur par défaut qui est spécifiée dans les paramètres du projet.

## **5.2 La visualisation**

### **5.2.1 Visualiser un objet graphique de type maillage**

L'utilisateur peut visualiser un nouvel objet graphique de type maillage en sélectionnant un maillage parmi la liste des maillages connus du projet. L'objet graphique est construit à partir de paramètres fournis par l'utilisateur et peut ensuite être affiché.

L'utilisateur peut sélectionner une couleur constante pour son maillage. Il peut également faire porter de l'information par la couleur. Pour cela, l'utilisateur doit spécifier un champ scalaire d'où est extraite l'information sur la couleur. L'utilisateur peut également faire porter un champ sur la coordonnée « z » (ex : champ de niveau d'eau) ce qui permet d'afficher le maillage en trois dimensions.

L'utilisateur peut demander à ce que les numéros des éléments soient affichés et sélectionner la couleur qui sera employée pour ce faire.

Le type de tracé du maillage peut être réduit, c'est-à-dire qu'au lieu de dessiner le maillage tel quel, on dessine en lieu et place l'intérieur de chaque élément, mais sans que

les éléments ne se touchent. Il est ainsi possible de détecter les trous indus dans le maillage.

En spécifiant un champ scalaire, il est également possible de demander l'affichage des isolignes. L'engin graphique derrière Modeleur2 se charge alors de les calculer, pour ensuite les afficher. Contrairement aux isolignes qui sont calculées avec l'outil de calcul d'isolignes dans Modeleur2 (voir 5.7), ces isolignes ne peuvent être enregistrées et analysées car elles sont générées uniquement pour une représentation visuelle par l'engin graphique.

## **5.2.2 Visualiser un objet graphique de type point**

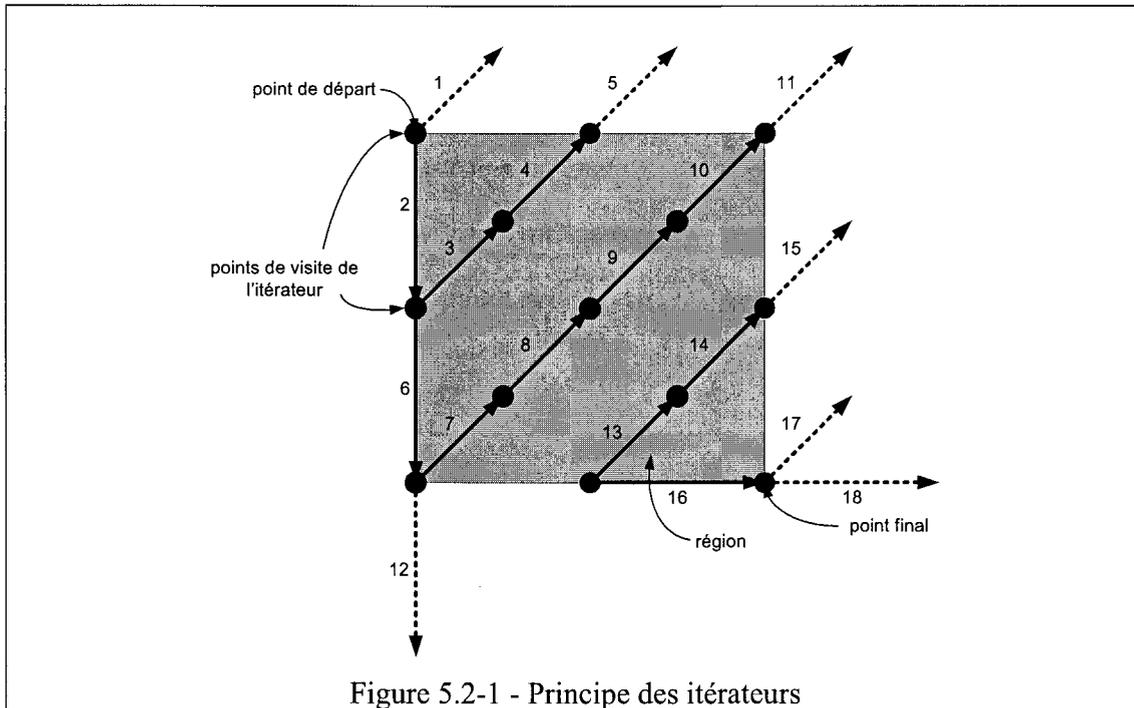
L'utilisateur peut visualiser des points qui sont spécifiés par une coordonnée  $x$  et une coordonnée  $y$ . Il a plusieurs façons de faire pour générer les points à tracer, et pour chacune d'elles, il y a différentes façons de faire porter de l'information sur les points.

### **5.2.2.1 Source d'où peuvent provenir les points**

L'utilisateur peut générer les points à tracer à partir des nœuds d'un maillage. L'utilisateur sélectionne alors le maillage source. L'utilisateur peut également changer le marqueur et faire afficher les numéros des nœuds.

L'utilisateur peut demander à ce que les points tracés soient obtenus en itérant sur une région. Il spécifie alors la région à parcourir, le pas d'itération et la direction.

La Figure 5.2-1 illustre le principe de parcours d'une région à l'aide d'itérateurs et du tracé des points.



L'utilisateur peut générer les points à tracer à partir d'un semi de points. Il sélectionne alors le semi de points d'origine. Il peut également changer le marqueur.

### 5.2.2.2 Information portée par les points

Sur les points générés, il est possible de faire porter de l'information de différentes façons. En voici quelques-unes :

- en associant un champ scalaire à la troisième coordonnée ( $z$ ), on ajoute une troisième dimension à l'affichage des points;
- en associant un champ vectoriel aux points, on peut illustrer un vecteur pour chacun des points;
- en associant un champ scalaire, on peut faire porter de la couleur par chacun des points.

### 5.2.3 Visualiser un objet graphique de type géométrie

L'utilisateur peut visualiser des objets graphiques de type *géométrie* qui ont été préalablement calculés dans une autre activité de Modeleur2. Par exemple, le calcul d'isosurfaces ou d'isolignes produit des objets géométriques.

Le résultat d'un calcul d'isosurfaces est hébergé dans un multipolygone (ensemble de polygones). Le résultat d'un calcul d'isolignes est hébergé dans une multipolyligne (ensemble de polylignes).

Ces objets géométriques peuvent être représentés à l'aide d'objets graphiques de type géométrie.

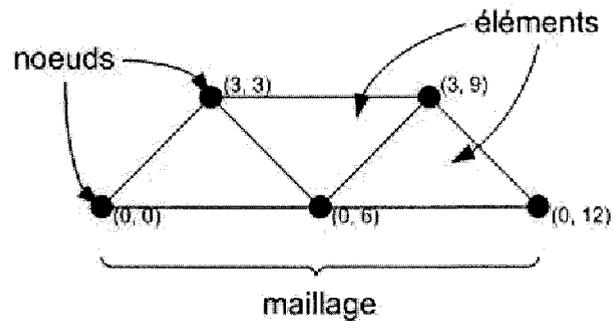
L'utilisateur peut modifier l'échelle de couleur utilisée pour représenter les isolignes / isosurfaces. La couleur peut être mappée en utilisant un champ scalaire. Ce champ peut être le même qui a servi à la construction des isolignes / isosurfaces ou peut être tout autre. L'utilisateur ne peut modifier le nombre d'isolignes / isosurfaces ou l'écart entre celles-ci car cette opération demande un calcul.

## 5.3 Les séries

### 5.3.1 Théorie sur les séries

La *série* est une structure de données avec laquelle les utilisateurs de Modeleur2 vont interagir. Dans le présent contexte, cette série est essentiellement basée sur les structures de maillage et de champ éléments finis, deux structures de données fondamentales dans Modeleur2 qui permettent une représentation numérique des données de terrain et de simulation. Pour cette raison, il apparaît pertinent d'effectuer le rappel suivant au sujet des éléments finis.

La *méthode des éléments finis* est une technique d'interpolation qui consiste à discrétiser le domaine de calcul en un ensemble de sous domaines de géométrie simple, les éléments. Sur chaque élément, on définit une fonction d'interpolation qui prend en compte la disposition des nœuds de l'élément (géométrie) et les valeurs qu'ils portent.



L'ensemble des nœuds et des éléments forme un maillage. Le champ éléments finis est la structure de données qui rassemble les valeurs nodales sur un maillage éléments finis. Une valeur en tout point de la région spatiale du champ est obtenue par interpolation au sein des éléments.

La différence entre un champ éléments finis conventionnel et la *série* telle que définie ici se situe au niveau du domaine auquel la structure est associée. Les domaines auxquels appartiennent les séries ne sont pas des régions spatiales (définies par des coordonnées  $\{x,y,z\}$  de l'espace géographique), ce sont plutôt des régions d'un espace conceptuel de solution.

Tout comme le maillage, la série possède des nœuds et des éléments, lesquels sont utilisés pour interpoler à partir de valeurs portées par les nœuds. L'emplacement de chacun de nœuds est spécifié *via* une coordonnée.

Typiquement, la coordonnée d'une série est entière ou réelle. Elle pourrait toutefois être un nombre complexe ou encore avoir été définie par l'utilisateur (à condition qu'elle implémente certaines opérations nécessaires).

Sous Modeleur2, les maillages sont de type bi-dimensionnel (2D) avec des coordonnées spatiales  $\{x,y\}$ . Les séries, elles, peuvent être en 1D et 2D, et théoriquement, pourraient même être en  $n$  dimensions. Chacun des axes d'une série peut être basé sur n'importe quel type de coordonnées (ex : paramètre de calcul, débit, temps, etc.)

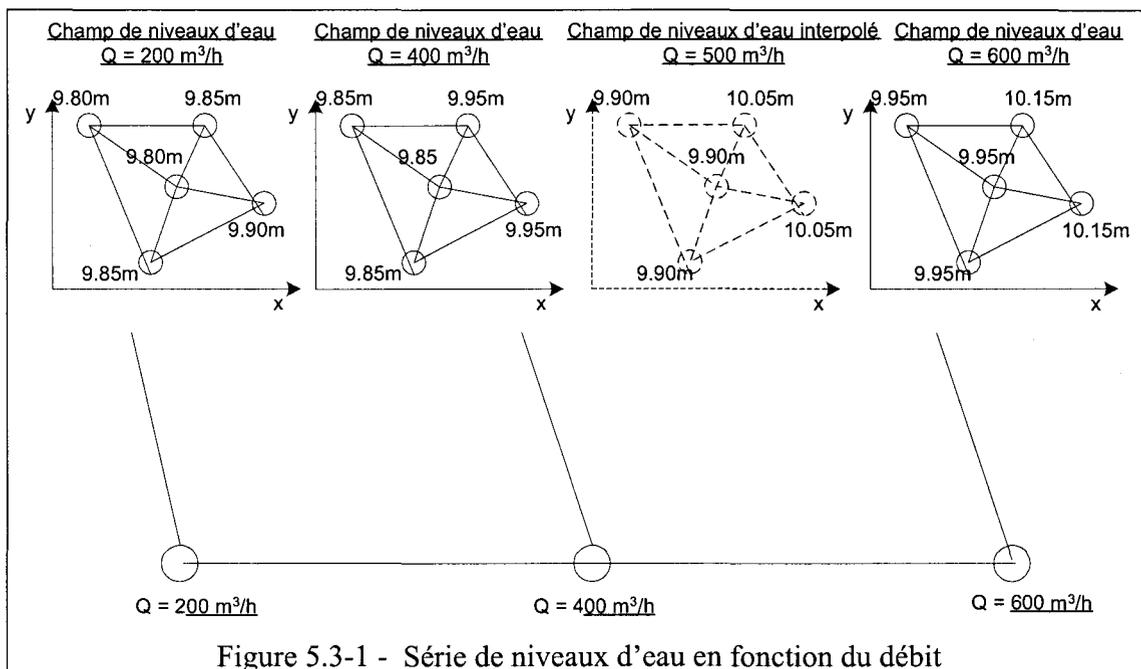
Tout comme les maillages éléments finis, les nœuds d'une série portent des valeurs. Dans le cas des séries, les valeurs peuvent être simples comme des scalaires (ex : niveau d'eau), des vecteurs (ex : vitesse) mais également des champs répartis spatialement.

Les données d'une série doivent être en relation. Par exemple, une série temporelle doit contenir des données qui diffèrent selon le temps. Il est possible de créer une série temporelle où les valeurs nodales de la série sont des champs de niveaux d'eau ou de vitesses à différents moments de l'année. La région d'une telle série est le domaine du temps. Un champ de niveaux d'eau ou de vitesses peut alors être interpolé pour connaître le niveau d'eau ou la vitesse à tout moment durant la période de temps couverte (région de la série).

L'espace solution que décrit une série peut avoir plusieurs coordonnées, chronologique ou paramétrique. Par exemple, une série de champs de température pourrait être définie en fonction du temps et de la force du vent.

La Figure 5.3-1 présente le cas d'une série 1D sur l'axe  $Q$  (débit en  $\text{m}^3/\text{h}$ ). Les coordonnées des nœuds sont des entiers avec les valeurs suivantes :  $q_0 = 200 \text{ m}^3/\text{h}$ ,  $q_1 = 400 \text{ m}^3/\text{h}$ ,  $q_2 = 600 \text{ m}^3/\text{h}$ .

La région de la série couvre la plage  $Q = 200 \text{ m}^3/\text{h}$  à  $Q = 600 \text{ m}^3/\text{h}$  dans l'espace solution. La valeur portée par chacun des nœuds est un champ de niveaux d'eau (variable physique qui dépend du débit). Avec cette série, l'utilisateur est en mesure d'obtenir un champ de niveaux d'eau interpolé pour n'importe quel débit  $Q$  dans l'intervalle  $200 \text{ m}^3/\text{h}$  à  $600 \text{ m}^3/\text{h}$ ). L'exemple suivant illustre le champ obtenu à  $Q = 500 \text{ m}^3/\text{h}$ . L'élément qui englobe  $Q = 500 \text{ m}^3/\text{h}$  est celui constitué des nœuds avec les coordonnées  $q_1$  et  $q_2$ . Le champ interpolé est donc obtenu à l'aide des champs portés par ces deux nœuds.



Les séries peuvent être combinées entre elles de manière à créer de nouvelles séries. Pour être combinées, deux séries doivent être compatibles. Des séries sont compatibles si et seulement si elles possèdent un axe commun. Par exemple, une série de débits dans le temps est compatible avec une série de températures dans le temps pour créer une série de débits en fonction de la température.

Il est possible de créer une série temporelle où les valeurs nodales sont des champs qui dépendent du débit d'une rivière (ex : les champs de vitesse et de niveaux d'eau), donc d'une série hydrologique. Si le débit peut être interpolé pour en connaître la valeur en un moment particulier, il en est de même des champs de valeurs correspondant au débit.

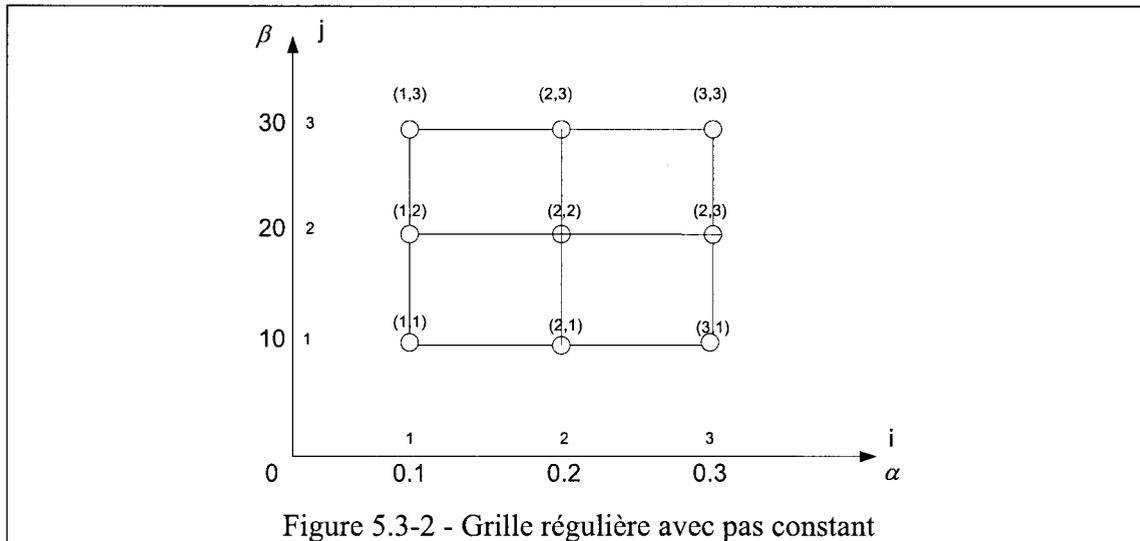
### 5.3.2 Utilisation des séries

Dans Modeleur2, l'utilisateur interagit avec les champs de résultats par l'intermédiaire des séries plutôt que directement avec les champs eux-mêmes comme c'était le cas dans Modeleur 1.0. Partout où l'utilisateur est appelé à travailler sur un champ, qu'il soit déjà défini dans l'espace de solution ou non, l'utilisateur le fait *via* une série (temporelle ou paramétrique) qui caractérise son évolution ou sa variabilité. Cela permet à l'utilisateur de sélectionner un point quelconque dans la série et d'obtenir par interpolation un champ à ce point. L'avantage de cette approche est de limiter le nombre de calculs à réaliser (en termes statistiques, on parlerait d'échantillon) pour caractériser l'espace de solution (similairement, la population) et d'exploiter à leur maximum les analyses de sensibilité paramétriques, courantes en modélisation.

### 5.3.3 Construction d'une série

Pour construire une série, l'utilisateur doit spécifier des couples coordonnées / valeur. La méthode employée pour la construction des séries est basée sur une simple grille dont les axes sont des indices (ex :  $i, j$ ) servant au repérage de l'information (ex : champs). L'utilisateur effectue dans un premier temps la correspondance entre les indices et les coordonnées des nœuds dans son espace de solution. Il spécifie ensuite les valeurs portées en fonction des indices de cette grille, et non pas en fonction des coordonnées des déterminants qui la sous-tendent.

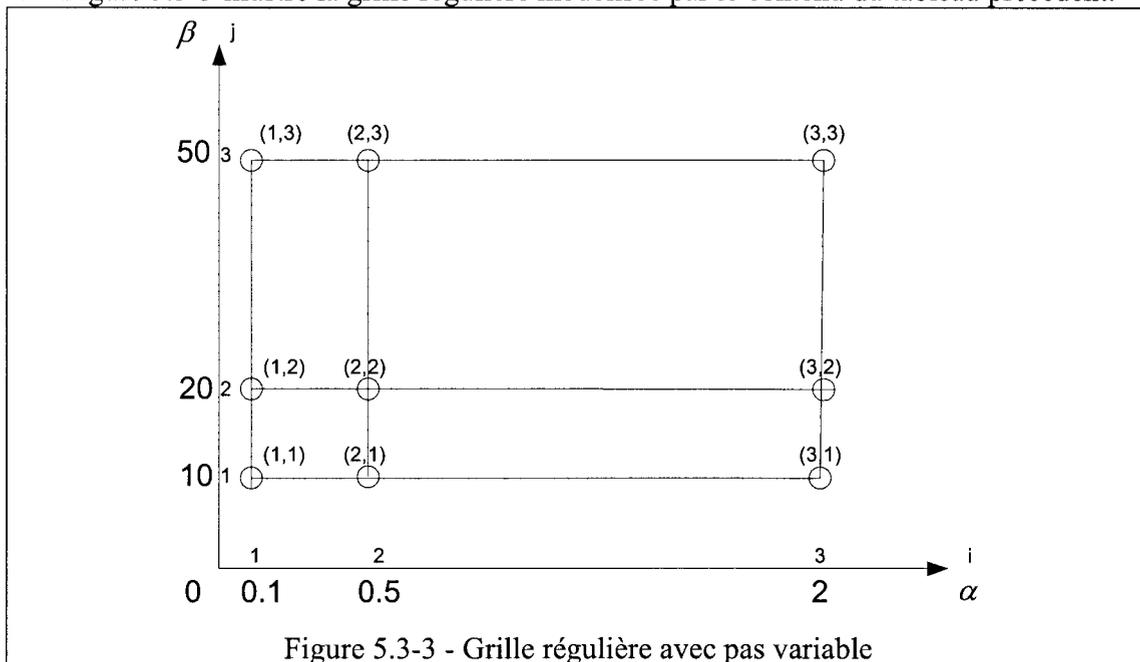
Pour définir l'espace de solution, l'utilisateur a la possibilité de construire une *grille régulière* simple en fournissant une coordonnée d'origine et un pas de coordonnée constant entre les nœuds de chacun des axes. Par exemple, si l'utilisateur spécifie que le pas de coordonnée sur l'axe  $i$  est de 0.1 et celui sur l'axe  $j$  est de 10, tous les nœuds peuvent ainsi être déterminés automatiquement. La Figure 5.3-2 illustre un cas où une série 2D avec neuf nœuds est proposée.



L'utilisateur a également la possibilité de construire une grille régulière plus complexe en spécifiant pour chaque axe un pas de coordonnée variable pour chaque axe (indice). La spécification de la coordonnée sur chaque axe se fait à l'aide d'un tableau qui indique la coordonnée sur l'axe en fonction de l'indice du noeud.

Coordonnée(Indice)	Indices <i>i</i> et <i>j</i>		
	1	2	3
$\alpha(i)$	0.1	0.5	2
$\beta(j)$	10	20	50

La Figure 5.3-3 illustre la grille régulière modélisée par le contenu du tableau précédent.



Une fois la grille régulière spécifiée (avec pas constant ou avec pas variable), l'utilisateur peut procéder à l'assignation des valeurs. L'assignation se fait, rappelons-le, *via* les indices  $i$  et  $j$  qui correspondent aux nœuds sur la grille régulière, et non pas à l'aide des coordonnées elles-mêmes. Les valeurs de l'espace de solution doivent toutes être du même type (condition d'homogénéité), par exemple des champs de niveaux d'eau issus de simulations sur le même maillage éléments finis.

Ainsi, si l'utilisateur décide de construire une série portant des champs de niveaux d'eau, chacun des neuf nœuds de la grille régulière doit porter un champ de niveaux d'eau.

Indice $i$	Indice $j$	Valeur
1	1	champ de niveaux d'eau #1
	2	champ de niveaux d'eau #2
	3	champ de niveaux d'eau #3
2	1	champ de niveaux d'eau #4
	2	champ de niveaux d'eau #5
	3	champ de niveaux d'eau #6
3	1	champ de niveaux d'eau #7
	2	champ de niveaux d'eau #8
	3	champ de niveaux d'eau #9

Cet exemple représente la construction d'une série dans un espace de solution à deux dimensions, mais il peut s'appliquer aussi pour une série en 1D ou 3D dépendant du nombre d'axes définissant la solution du problème. Dans le cas d'une série 1D, on ne peut pas strictement parler de grille, mais l'utilisateur entre tout de même les valeurs en fonction de l'indice et non pas de la coordonnée. Ceci permet de construire toutes les sortes de séries avec la même façon de faire peu importe si l'on a affaire à une série 1D ou nD.

### 5.3.4 Gestion d'une série

#### 5.3.4.1 Nouveau

L'utilisateur peut créer une nouvelle série. L'utilisateur doit spécifier le type des valeurs portées par les nœuds de la série.

L'utilisateur a également la possibilité de créer une série en combinant deux séries qui possèdent un axe commun. Par exemple, une série hydrologique et une série de champs de niveaux d'eau en fonction du débit ont en commun la valeur du débit; il en résulte un champ chronologique de niveau d'eau.

#### **5.3.4.2 Ouvrir**

L'utilisateur peut ouvrir une série existante. Il peut alors modifier les valeurs ou ajouter / retirer des nœuds de la série.

#### **5.3.4.3 Fermer**

L'utilisateur peut fermer une série. À la fermeture d'une série, l'utilisateur se voit demander s'il désire l'enregistrer.

#### **5.3.4.4 Enregistrer**

L'utilisateur peut enregistrer une série pour laquelle les valeurs aux nœuds ont toutes été initialisées (spécifiées).

Pour les séries qui portent des champs, seule l'identification du champ est enregistrée à titre de valeur. Le champ lui-même n'est pas enregistré dans l'opération. Pour les séries qui portent des valeurs simples (ex : débit), la valeur est par contre enregistrée dans l'opération.

#### **5.3.4.5 Enregistrer sous...**

L'utilisateur peut enregistrer une série sous un autre nom. Ceci a pour effet de la dupliquer.

### **5.3.5 Fonctionnalités reliées aux séries**

#### **5.3.5.1 Créer la structure d'une série**

L'utilisateur crée la structure d'une série en spécifiant la dimension (portée) de ses axes et le type des coordonnées correspondantes. Il doit ensuite spécifier la grille régulière qui réglera la disposition des nœuds de sa série.

#### **5.3.5.2 Modifier la structure d'une série**

L'utilisateur peut modifier les dimensions d'une série et la définition de sa grille de repérage indicelle. La modification de la structure d'une série entraîne l'invalidation de toutes ou certaines des valeurs portées par cette série.

Lorsqu'une invalidation survient, le champ valeur est remis à une valeur non initialisée. L'utilisateur ne peut enregistrer une série qui contient des valeurs non valides.

#### **5.3.5.3 Ajouter un ou des nœuds à une série**

L'utilisateur peut ajouter un nœud à une grille régulière 1D ou une ligne de nœuds à une grille régulière 2D.

L'utilisateur doit spécifier l'axe et la coordonnée à laquelle ajouter le nœud / la ligne. À partir de la valeur de la coordonnée, l'indice où insérer le nœud / la ligne sera automatiquement calculé. Tous les autres éléments affectés par cet ajout verront leur indice incrémenté d'une unité.

Une fois l'ajout complété, l'utilisateur doit spécifier la valeur ou le champ associé à chaque nœud créé dans l'opération. Une série qui contient des nœuds sans valeur ou champ associé ne peut être enregistrée.

#### **5.3.5.4 Retirer un ou des nœuds à une série**

L'utilisateur peut retirer un nœud à une grille régulière 1D ou une ligne de nœuds à une grille régulière 2D.

L'utilisateur doit spécifier l'axe et la coordonnée à laquelle retirer son nœud / sa ligne / son plan. À partir de la valeur de la coordonnée, l'indice du ou des nœuds qu'il désire retirer peut être trouvé. L'utilisateur peut également spécifier directement l'indice.

Tous les autres éléments affectés par ce retrait verront leur indice soustrait d'une unité.

#### **5.3.5.5 Visualiser la série**

L'utilisateur peut visualiser la série. Il voit alors les nœuds, disposés selon la grille régulière qu'il a choisie. Les valeurs aux nœuds sont affichées. Si la valeur est un champ, le nom du champ est affiché. L'utilisateur ne peut interagir graphiquement avec la série.

#### **5.3.5.6 Modifier les valeurs des nœuds d'une série**

L'utilisateur peut modifier les valeurs ou champs portés par la série. Les valeurs possibles doivent uniquement être du même type que celui de la série.

## **5.4 Les scripts et les macros**

L'activité « Gestion des scripts et des macros » a pour but de permettre à l'utilisateur d'automatiser le fonctionnement de Modeleur2 et de fournir l'accès à la plupart de ses fonctionnalités « cachées ». L'utilisateur peut ainsi créer des scripts en Python qu'il sera ensuite capable d'exécuter à répétition.

Pour écrire un script, l'utilisateur a deux choix : soit il édite un *script* en spécifiant (rédigeant) lui-même le contenu, soit il crée une *macro*. Une *macro* est un script qui enregistre automatiquement toutes les actions produites dans Modeleur2 pendant une période de temps définie par l'utilisateur. Pour que l'utilisateur puisse éditer un script, Modeleur2 comprend un IDE (Integrated Development Environment) externe contenant un éditeur ( et un débogueur).

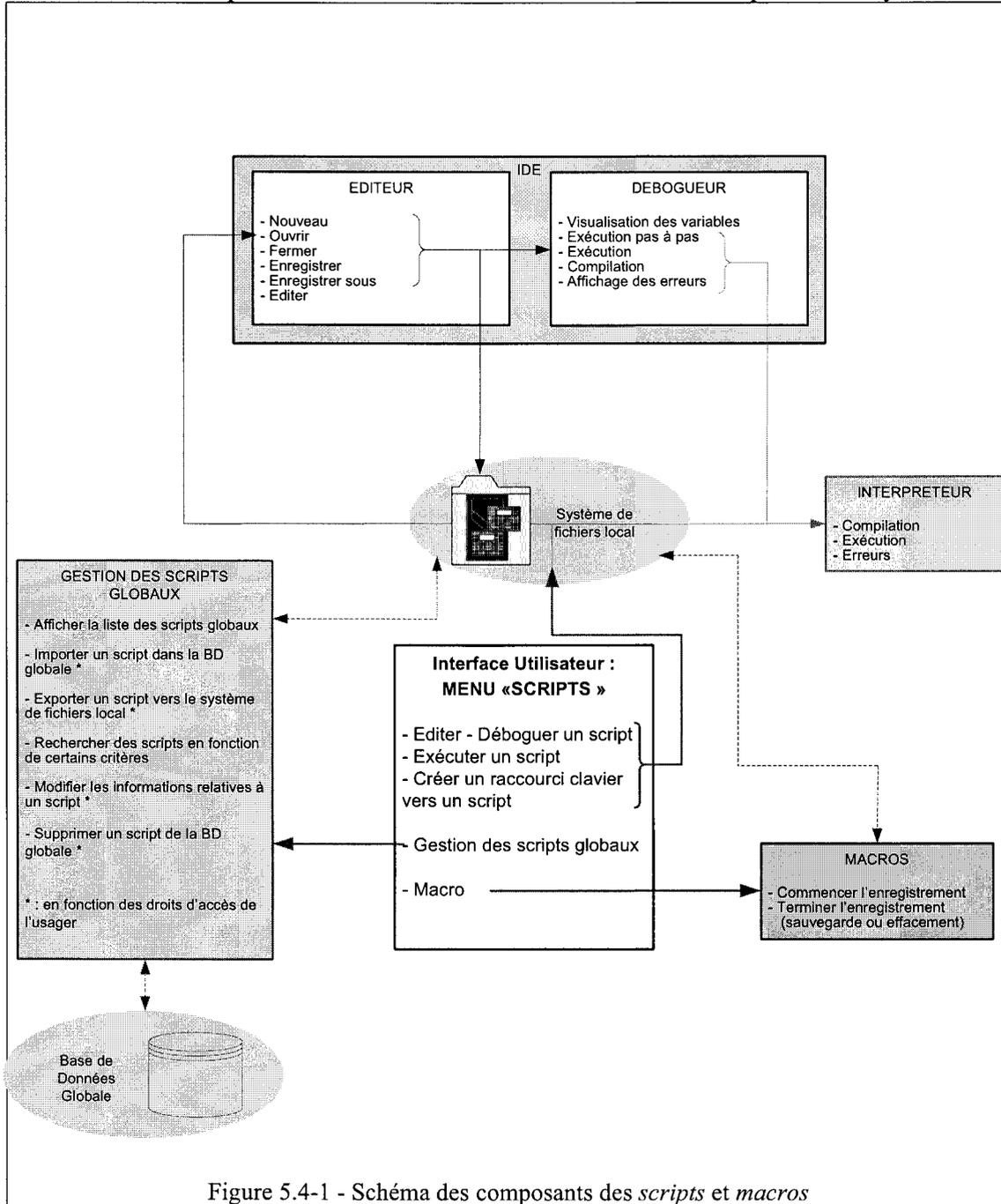
Après avoir enregistré ses scripts sur disque, l'utilisateur peut les gérer à l'aide du système de fichiers de son système d'exploitation. Il bénéficie ainsi de toute la souplesse et la facilité que ce dernier peut lui procurer.

L'utilisateur peut exporter les scripts qu'il crée dans une base de données partagée par plusieurs utilisateurs. Cette base de données est désignée par le terme base de données globale dans les lignes qui suivent. Une telle base de données doit être disponible pour que les fonctionnalités qui en font usage soient disponibles.

L'utilisateur peut exporter des scripts globaux vers son système de fichiers local, s'il est autorisé à le faire (droits d'accès). Il peut s'il le souhaite et s'il y est autorisé importer des scripts locaux vers la base de données globale. Ainsi, ses scripts deviennent globaux et peuvent être partagés entre plusieurs usagers. Seuls les usagers qui y sont autorisés peuvent importer ou détruire des scripts dans la base de données globale.

Enfin, l'utilisateur bénéficie d'un débogueur qui lui permet de tester les scripts créés. Ce débogueur externe interagit directement avec Modeleur2 et fait partie d'un IDE qui contient également l'éditeur de code.

Le schéma suivant présente les interactions entre les différents composants du système :



### **5.4.1 Gestion des scripts globaux**

La gestion des scripts globaux se fait à l'aide d'une liste qui affiche les différents scripts qui se trouvent dans la base de données globale. Seuls les scripts correspondant à certains critères de recherche sont affichés.

#### **5.4.1.1 Importer**

L'utilisateur peut importer des scripts locaux vers la base de données globale, il est pour cela invité à choisir un script Python à l'aide d'une boîte de dialogue de sélection de fichiers Python. Ce script est ensuite compilé pour avoir l'assurance de ne stocker que des scripts Python dans la base de données globale.

Après cette vérification, une boîte de dialogue de modification est affichée pour que l'utilisateur puisse entrer des informations sur son script (nom, date, type, description). Après la validation de ces informations, le script est enregistré dans la base de données globale.

L'utilisateur ne peut importer un script dans la base de données globale que s'il a les droits d'accès qui permettent cette opération.

#### **5.4.1.2 Exporter**

L'utilisateur peut exporter un script de la base de données globale vers son système de fichiers local, pour cela il sélectionne dans une liste d'affichage le script global qu'il souhaite exporter vers son espace de travail local. Une boîte de dialogue l'invite ensuite à sélectionner le répertoire dans lequel son script sera copié.

#### **5.4.1.3 Rechercher**

L'utilisateur peut entrer une requête SQL pour pouvoir afficher les scripts qui correspondent à certains critères de recherche.

#### **5.4.1.4 Modifier**

L'utilisateur peut modifier un élément de la base de données globale en le sélectionnant, puis soit en double-cliquant dessus, soit en cliquant sur le bouton Modifier de la boîte de dialogue de gestion des scripts globaux.

La boîte de dialogue de modification des informations relatives à un script permet ensuite de modifier le nom, la date, le type, la description de l'élément sélectionné.

#### **5.4.1.5 Supprimer**

L'utilisateur peut supprimer un ou plusieurs script(s), s'il dispose des droits d'accès nécessaires.

#### **5.4.1.6 Glisser-déposer et raccourcis clavier standards**

L'utilisateur peut « glisser – déposer » ou utiliser les raccourcis classiques de son système d'exploitation (ctrl+v, ctrl+c, ctrl+x sous Windows) pour copier des scripts locaux vers la base de données globale.

L'utilisateur est ensuite interrogé par une boîte de dialogue qui lui demande de fournir les informations relatives à chaque script qu'il a décidé d'importer dans la base de données globale.

### **5.4.2 Gestion des scripts locaux**

#### **5.4.2.1 Edition – Débogage**

L'utilisateur sélectionne un script local à l'aide d'une boîte de dialogue classique de sélection de fichiers. Ce script est ensuite chargé dans un éditeur externe qui reste en liaison avec Modeleur2 pour pouvoir être débogué dans ce dernier.

#### **5.4.2.2 Exécution**

L'utilisateur sélectionne un script local à l'aide d'une boîte de dialogue classique de sélection de fichiers. Ce script est ensuite exécuté dans l'interpréteur de scripts de Modeleur2. L'interpréteur de Modeleur2 fournit à l'utilisateur des informations sur d'éventuelles erreurs dues à un script bogué.

#### **5.4.2.3 Raccourcis clavier**

L'utilisateur peut choisir un raccourci clavier pour l'exécution d'un script qu'il aura sélectionné à l'aide d'une boîte de dialogue de sélection de fichiers. Ainsi, à chaque appel du raccourci clavier, un certain script sera appelé.

### **5.4.3 Macros**

#### **5.4.3.1 Commencer l'enregistrement**

L'utilisateur peut enregistrer les actions qu'il produit dans une macro. Pour cela, il doit débiter l'enregistrement de ces actions.

#### **5.4.3.2 Terminer l'enregistrement**

De la même manière, l'utilisateur doit indiquer à Modeleur2 quand il souhaite que l'enregistrement de ces actions prenne fin. Quand l'enregistrement est terminé, il est invité à nommer la macro qu'il vient de créer. S'il le désire, il peut décider d'effacer cette macro (annulation de l'enregistrement).

Cette capture des actions de l'utilisateur crée un script automatiquement (macro), lequel peut ensuite être modifié par l'éditeur de code précédemment cité.

#### **5.4.3.3 Pause / reprise de l'enregistrement**

L'utilisateur peut interrompre momentanément l'enregistrement de la macro lorsqu'il ne souhaite pas enregistrer certaines de ses actions, et le reprendre ultérieurement..

### **5.4.4 IDE externe - Editeur**

Un éditeur externe est intégré à Modeleur2 pour faciliter la modification des scripts par l'utilisateur.

#### **5.4.4.1 Caractéristiques nécessaires pour l'intégration dans Modeleur2**

Cet éditeur doit être capable de se lancer avec en argument le script à ouvrir, il doit également pouvoir recevoir à tout moment le nom d'un script à ouvrir.

#### **5.4.4.2 Nouveau**

L'éditeur permet à l'utilisateur de créer un nouveau script.

#### **5.4.4.3 Ouvrir**

L'éditeur permet à l'utilisateur d'ouvrir un script existant.

#### **5.4.4.4 Editer**

L'éditeur fournit à l'utilisateur toutes les fonctionnalités d'un éditeur de code standard: auto indentation, coloration des mots clé, auto complétion, etc...).

#### **5.4.4.5 Enregistrer**

L'utilisateur peut enregistrer les modifications apportées à son script lorsqu'il le souhaite.

#### **5.4.4.6 Enregistrer sous...**

L'utilisateur peut enregistrer sous un autre nom le script en cours de modification. Cette fonctionnalité permet d'enregistrer les modifications dans un nouveau script sans avoir à écraser l'ancien.

#### **5.4.4.7 Fermer**

L'utilisateur peut fermer un script.

#### **5.4.4.8 Quitter**

L'utilisateur peut quitter l'éditeur de script externe à tout moment pour revenir dans Modeleur2.

#### **5.4.5 IDE externe - Débogueur**

Après avoir créé ou modifié un script, l'utilisateur peut tester le bon fonctionnement de son script à l'aide d'un débogueur externe intégré à l'éditeur de code.

##### **5.4.5.1 Caractéristiques nécessaires pour l'intégration dans Modeleur2**

Ce débogueur doit être capable d'exécuter les « scripts à déboguer » dans l'interpréteur Python de Modeleur2.

##### **5.4.5.2 Compiler**

L'utilisateur peut compiler pour voir s'il n'y a pas d'erreurs de syntaxe.

##### **5.4.5.3 Exécuter pas à pas**

L'utilisateur peut exécuter son script ligne par ligne et voir les problèmes éventuels.

##### **5.4.5.4 Visualisation de l'état des variables**

L'utilisateur peut, pendant le débogage, voir la valeur (l'état) des différentes variables Python.

##### **5.4.5.5 Exécuter**

L'utilisateur peut exécuter le script en entier, pour éventuellement s'arrêter sur une ligne fautive lors de l'apparition d'un bogue.

##### **5.4.5.6 Affichage des erreurs**

L'utilisateur bénéficie d'informations utiles lors du débogage telles qu'un bon affichage des erreurs que renvoie l'interpréteur.

## 5.5 La calculatrice

### 5.5.1 Description et rôle

Modeleur2 offre aux usagers une *calculatrice* qui permet de manipuler des types de données simples (ex : entier ou double précision), mais également de manipuler certaines structures de données de Modeleur2 (ex : les champs et les séries). La calculatrice sous Modeleur2 utilise l'interpréteur Python comme plaque tournante.

Le calculatrice inclut un clavier ayant une apparence similaire à celui de la calculatrice sous Windows. Le clavier de la calculatrice est là pour éviter d'obliger l'utilisateur à rédiger du code (Python) manuellement. On veut éviter le « syndrome de la page blanche » car beaucoup d'utilisateurs ne sont pas à l'aise avec l'idée de devoir programmer. Il faut donc les placer dans un contexte de fonctionnement convivial auquel ils sont plus habitués et dans lequel ils pourront effectuer un travail plus efficace.

Dans cette optique, la calculatrice de Modeleur2 propose aux usagers la liste des variables et des fonctions connues de la calculatrice (i.e. variables / fonctions connues de l'interpréteur Python). En effectuant un double-clic sur une variable ou une fonction, celle-ci sera affichée dans la zone <commandes> (voir Figure 5.5-1) comme si l'utilisateur l'avait écrite manuellement.

L'utilisateur peut enchaîner séquentiellement des opérations sur la ligne de commande comme dans une calculatrice standard. La ligne de commande est ensuite transférée à l'interpréteur Python pour qu'il l'évalue. Il est important de voir qu'il n'y a aucun pré-traitement fait par la calculatrice avant l'expédition à l'interpréteur Python.

Si l'utilisateur enchaîne des symboles ou des expressions qui n'ont pas de sens, c'est l'interpréteur Python qui va réagir en disant que la syntaxe est incorrecte au moment de l'exécution de la ligne. L'utilisateur se verra alors transmettre le message d'erreur généré par Python.

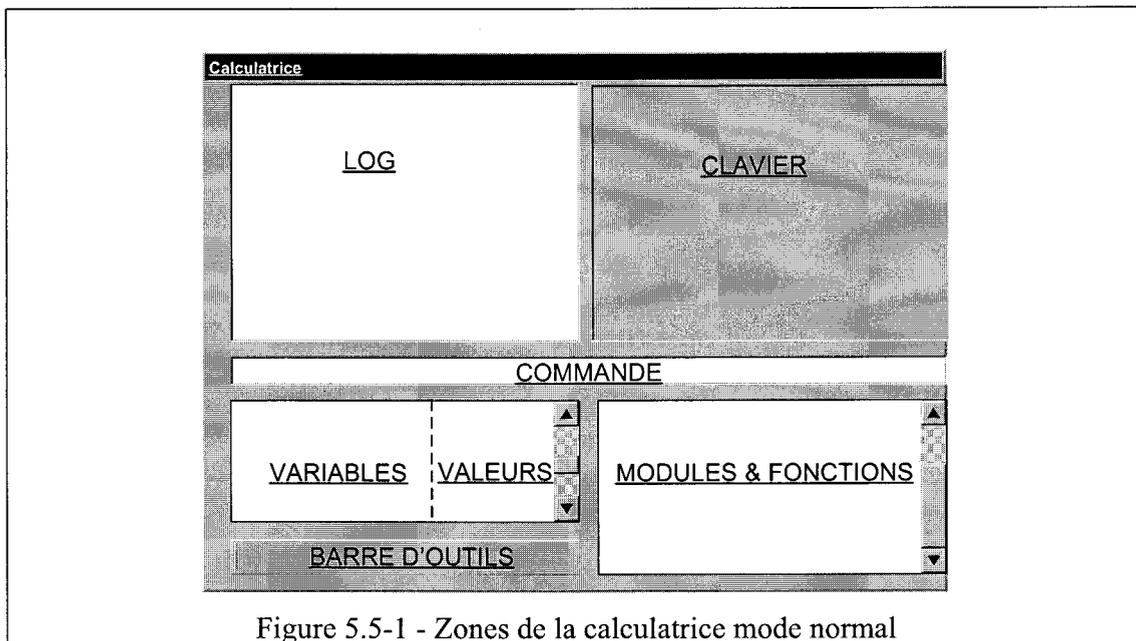
Le création de variables dans la calculatrice se fait de manière implicite en écrivant une ligne d'assignation en Python. Par exemple, `myvar = 1` aura pour effet de créer la variable `myvar` et de la charger avec la valeur 1. Python n'est pas un langage typé, alors si on écrit `myvar = monchamp`, `myvar` contiendra un champ.

Le chargement de variables peut se faire en appelant une fonction de chargement de données parmi celles connues dans la calculatrice (i.e. les modules importés – voir 5.5.2.3). Par exemple, l'appel suivant : `myvar = chargeChampAnalytique(id)` permet de charger le champ analytique à partir de son *id*.

### 5.5.1.1 Interface usager

Voici à quoi correspond chacune des zones de l'interface-usager de la calculatrice Figure 5.5-1) :

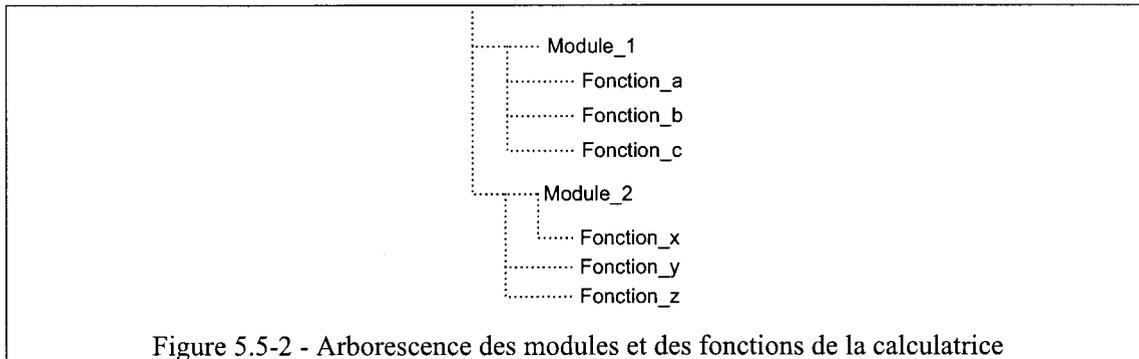
- <log>: code Python qui a été exécuté
- <clavier>: clavier de calculatrice standard
- <commande>: ligne de commandes entrée par l'utilisateur
- <variables> : liste des variables disponibles
- <valeurs>: valeur de la variable sélectionnée
- <fonctions>: liste des fonctions disponibles pour chaque module
- <barre d'outils> fonctionnalités disponibles



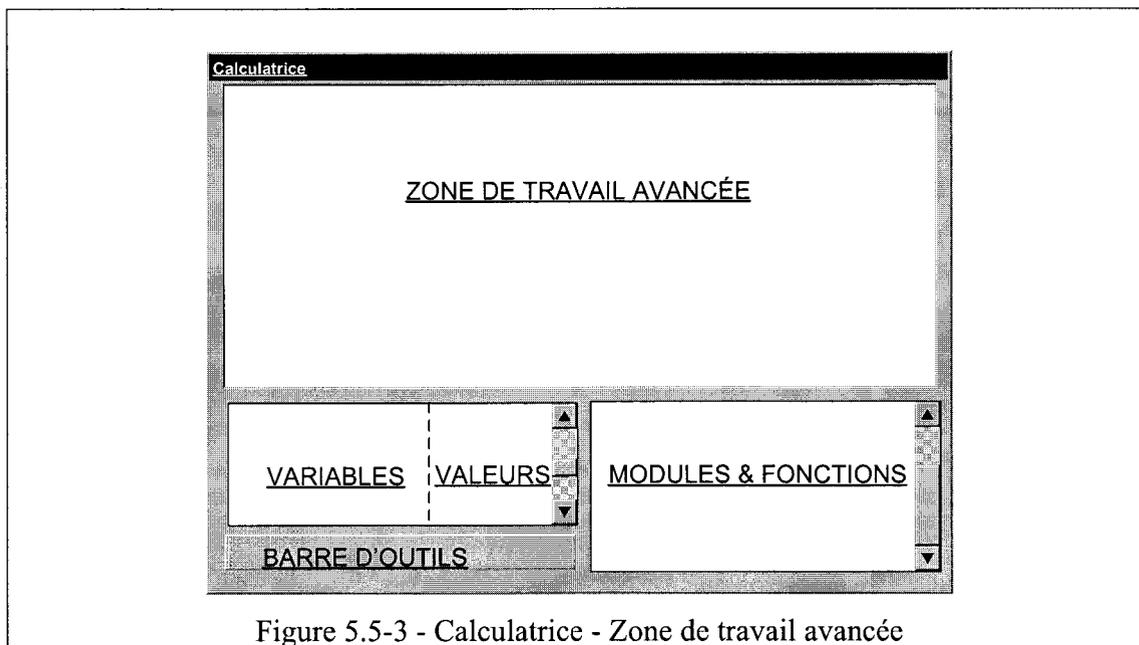
La zone <variables> contient toutes les variables qui ont été créées ou chargées par l'utilisateur. Lorsqu'une variable est sélectionnée, son contenu est affiché dans la zone <valeurs>.

La zone <fonctions> contient toutes les fonctions des modules (ex : statistiques, algébriques, différentielles, intégrales ou autres) connus de la calculatrice. Les fonctions sont regroupées dans une arborescence ayant comme racine le nom de ce module.

La Figure 5.5-2 illustre la vue qui est proposée à l'utilisateur dans la zone <fonctions>.



Le clavier de la calculatrice peut être caché, ce qui permet d'agrandir la zone d'édition. La zone <log>, la zone <clavier> et la zone <commande> se combinent alors pour former une « zone de travail avancée ». Les autres zones demeurent au même endroit. La illustre la vue alors proposée à l'utilisateur.



Les modules (et leurs fonctions) peuvent avoir été définis par l'utilisateur dans le projet en cours via la création d'un script Python (voir 5.4.4.2).

Ils peuvent également être importés d'un projet Modeleur2 existant ou directement de Python (ex : module Math). Le choix des modules disponibles par défaut dans la calculatrice se fait *via* une configuration du plug-in. Il est possible d'extraire la description de chacune des fonctions disponibles. L'utilisateur a accès à cette description *via* une bulle d'aide.

## 5.5.2 Fonctionnalités de la calculatrice - Barre d'outils

La zone <barre d'outils> de la calculatrice contient les fonctionnalités suivantes :

- charger une ou plusieurs variables;
- enregistrer une variable;
- importer un module;
- rafraîchir un module;
- créer un module.

La section suivante décrit chacune de ces fonctionnalités.

### 5.5.2.1 Charger une ou plusieurs variables

L'utilisateur peut charger une ou plusieurs variables simultanément *via* une boîte de dialogue. La boîte de dialogue permet à l'utilisateur de sélectionner les variables sources parmi les données disponibles dans Modeleur2 et déterminer dans quelles variables les charger.

Une fois ces variables chargées, l'utilisateur peut les consulter dans la zone <variables>. La liste des types listés correspond à ceux pour lesquels la fonctionnalité de chargement a été publiée en Python.

### 5.5.2.2 Enregistrer une variable

Cette fonctionnalité permet à l'utilisateur d'enregistrer des variables de sa calculatrice dans le projet courant.

L'opération exécutée est toujours un « Enregistrer sous... ». L'utilisateur doit donc spécifier un nom pour la donnée qu'il s'apprête à sauvegarder. Lorsque l'utilisateur quitte la calculatrice, il se voit proposer d'enregistrer ses variables.

### 5.5.2.3 Importer un module dans la calculatrice

L'utilisateur peut importer un module Python qu'il a codé lui-même à l'aide de l'éditeur de scripts fourni avec Modeleur2 (voir 5.4.4). Le script peut faire parti du projet courant ou d'un autre projet. S'il fait parti d'un autre projet, il ne pourra être édité.

S'il y a une erreur à l'importation, le message détaillant l'erreur est affiché. L'importation d'un module fait apparaître le module et ses fonctions dans la zone <fonctions> de la calculatrice.

Certains modules sont importés par défaut. Les modules qui sont importés par défaut sont ceux qui sont spécifiés lors de la configuration du plug-in Calculatrice. Si des modules

importés par défaut ne sont pas présents ou s'ils contiennent des erreurs, l'utilisateur en est averti. Ces modules ne sont alors pas affichés dans la zone <fonction> mais l'utilisateur peut continuer de travailler.

#### **5.5.2.4 Éditer un module**

L'utilisateur peut éditer certains modules Python présents dans la calculatrice, à partir de l'interface de la calculatrice. Ceci a pour effet d'appeler l'éditeur externe de scripts couplé à Modeleur2. Ce ne sont pas tous les modules qui peuvent être édités. Ainsi, les modules Python comme l'item module Math ne peuvent être édités. Les modules qui proviennent d'autres projets peuvent être ouverts en lecture seulement. Il est alors possible « d'enregistrer sous... » le module en question dans le projet courant pour le modifier à sa guise. Les modules éditables sont identifiés de façon spéciale pour que l'utilisateur puisse les reconnaître.

#### **5.5.2.5 Rafraîchir un module**

Si l'utilisateur sait qu'un module a été modifié, il doit procéder à un rafraîchissement de la calculatrice. Ceci provoque la réimportation des modules. Cette opération est nécessaire pour que les changements soient pris en compte.

### **5.6 L'outil de sonde**

Comme dans Modeleur 1.0, cet outil permet d'obtenir la position géographique d'un point et d'interpoler les valeurs d'un ou plusieurs champs à cette position géographique. La sonde peut être utilisée dans n'importe laquelle des fenêtres graphiques de Modeleur2. La sonde permet de voir les coordonnées en fonction de la position de la souris dans la fenêtre graphique.

#### **5.6.1.1 Ajouter un champ**

L'utilisateur peut suivre les valeurs de plus d'un champ à la fois. Il peut ajouter à tout moment un nouveau champ. Les champs ajoutés peuvent être de type scalaire, vectoriel ou ellipse erreur.

L'utilisateur peut sélectionner le champ à partir d'une série qui porte des champs. L'utilisateur n'a alors qu'à sélectionner les coordonnées de la série pour lesquelles il désire obtenir le champ.

#### **5.6.1.2 Retirer un champ**

L'utilisateur peut retirer un champ de la liste de ceux qu'il désire suivre.

### **5.6.1.3 Changer le repère**

L'utilisateur peut changer le repère (projection spatiale) à partir duquel l'information est affichée

### **5.6.1.4 Placer la sonde dans la zone Information**

L'utilisateur peut placer la sonde dans la zone information (voir 2.1.3). Ceci lui permet de toujours avoir l'information sur sa position géographique dans son domaine et les valeurs aux points d'intérêt.

### **5.6.1.5 Changer de mode**

L'utilisateur peut passer en mode différé ou en mode immédiat. Le mode immédiat permet d'obtenir la position en temps réel lors des déplacements de la souris. Le mode différé demande à ce que l'utilisateur effectue un clic de souris pour que la position soit mise à jour.

## **5.7 L'outil de calcul d'isolignes et d'isosurfaces**

Les résultats provenant de calculs d'isolignes et de calculs d'isosurfaces sont des *objets géométriques* qui portent des valeurs. Bien qu'ils en aient l'air, ils ne sont pas seulement reliés à une représentation visuelle. Il est donc possible d'analyser ces objets avec des outils qui permettent d'en étudier la géométrie ou encore les valeurs portées.

### **5.7.1.1 Calculer des isolignes**

L'utilisateur peut procéder à un calcul d'isolignes à l'aide d'un maillage et d'un champ scalaire. L'utilisateur doit spécifier le nombre d'isolignes qu'il désire ou encore un écart de valeur qui en détermine le nombre. Une fois ces informations spécifiées, le calcul peut être exécuté et le résultat consiste en un objet géométrique de type *isoligne*.

### **5.7.1.2 Calculer des isosurfaces**

L'utilisateur peut procéder à un calcul d'isosurfaces à l'aide d'un maillage et d'un champ scalaire. L'utilisateur doit spécifier le nombre d'isosurfaces qu'il désire ou encore un écart de valeur qui en détermine le nombre. Une fois ces informations spécifiées, le calcul peut être exécuté et le résultat consiste en un objet géométrique de type *isosurface*.

### **5.7.1.3 Enregistrer des isolignes**

Les résultats d'un calcul d'isolignes peuvent être enregistrés dans la base de données. Les isolignes ainsi enregistrées pourront être visualisées (voir 5.2.3) ou pourront être analysées en tant que lignes avec l'outil de lignes.

#### **5.7.1.4 Enregistrer des isosurfaces**

Les résultats d'un calcul d'isosurfaces peuvent être enregistrés dans la base de données. Les isosurfaces ainsi enregistrées pourront être visualisées (voir 5.2.3) ou pourront être analysées en tant que surfaces avec l'outil de surfaces.

## 5.8 L'outil de lignes

### 5.8.1 Théorie

Afin de bien situer le lecteur en ce qui a trait à la nomenclature utilisée dans cette section, une brève description des concepts de multipolygones et de polygones est présentée dans l'encadré ci-après.

#### ***Polyligne :***

Une *polyligne* repose sur une liste ordonnée de coordonnées. La polyligne est l'union des segments de droite compris entre chacune des paires de coordonnées consécutives dans cette liste. Les coordonnées représentent les *sommets* de la polyligne.

Les divers segments de droite formant la polyligne peuvent se recouper ou se superposer sans aucune restriction.

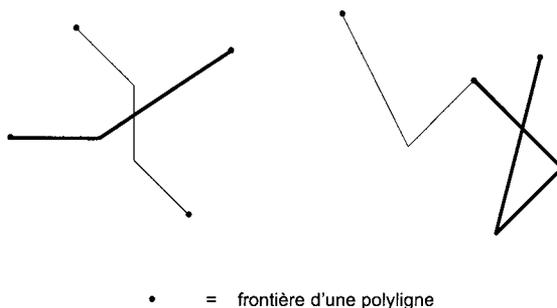
La *frontière* d'une polyligne est constituée des deux sommets aux extrémités de la polyligne. Dans le cas où ces sommets sont les mêmes, la polyligne a une frontière vide et est dite *fermée*.

#### ***Multipolyligne :***

La *multipolyligne* est définie comme une collection de polygones qui ne peuvent se toucher entre elles qu'en leur frontière (contrainte SIG). Elles peuvent aussi être distinctes.

Exemple :

La première multipolyligne dans la figure ci-après est invalide (SIG) car les deux polygones se croisent ailleurs qu'en leur frontière. La seconde multipolyligne, elle, est valide car les deux polygones ne se croisent qu'en leur frontière.

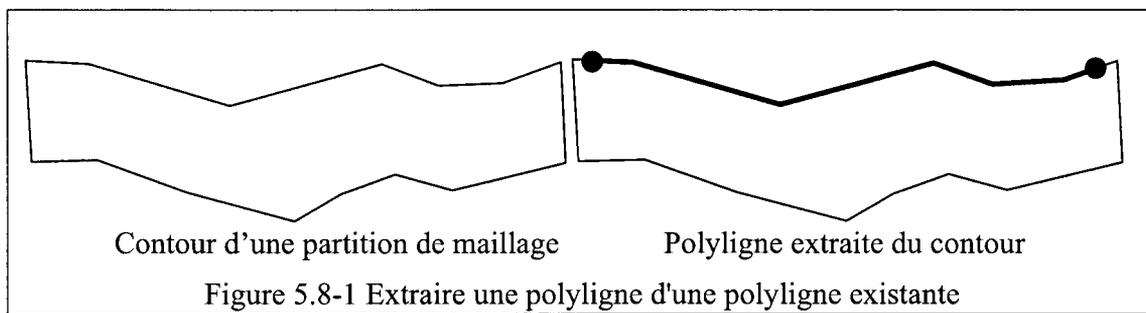


## 5.8.2 Construction

L'outil de lignes offre à l'utilisateur la possibilité de construire des multipolygones qu'il peut par la suite analyser.

Par exemple, l'utilisateur peut construire une multipolygone regroupant trois polygones (ex : contour de trois îles). Évidemment, il peut construire une multipolygone constituée d'une seule polygone (cas le plus simple).

L'utilisateur peut construire une polygone qui suit le contour d'une polygone ou d'une partie d'une polygone générée dans d'autres modules de Modeleur2. Par exemple, la Figure 5.8-1 illustre le contour d'une partition (construite dans le module partition de maillage). De cette polygone, il est possible d'extraire une polygone ou un segment d'une polygone.



L'utilisateur peut construire une polygone à partir d'une polygone existante en tentant de suivre le contour de la polygone existante. Quand l'outil de construction détecte cela, il propose à l'utilisateur de construire un segment de polygone à partir de la géométrie de la polygone existante. C'est un comportement typique dans un logiciel comme Microsoft Visio<sup>MD</sup>. L'utilisateur peut également cliquer sur deux points d'une polygone existante et l'outil de construction repère automatiquement la polygone qui contient ces points. Il est à même de construire une polygone identique à celle d'où proviennent les points sur lesquels ont eu lieu les deux clics.

## 5.8.3 Interface usager : gestion des multipolygones

Une multipolygone qui vient d'être créée est liée à la fenêtre active. La Figure 5.8-3 illustre la vue d'une arborescence de multipolygones où il y a une multipolygone à deux polygones (multipolygone #1) qui correspond par exemple à la ligne de berge. Il y a également une multipolygone contenant une seule polygone (multipolygone #2).

Quand le contrôle « Arborescence des lignes » est le contrôle actif dans la zone de contrôle, l'utilisateur peut éditer et modifier ses multipolygones. Quand c'est un autre contrôle qui est actif, l'utilisateur continue de les voir dans la fenêtre graphique, mais ne peut les modifier.

En double-cliquant sur un élément multipolyligne dans l'arborescence, l'utilisateur voit la fenêtre hébergeant cette multipolyligne devenir active. S'il n'y a pas encore de représentation associée à la multipolyligne, l'utilisateur peut poser des sommets pour la construire car il est implicitement en édition.

S'il y a déjà une ou des polygones tracés, celles-ci sont sélectionnées et l'utilisateur peut les modifier, c'est-à-dire qu'il peut ajouter et enlever des sommets à une polygone existante. Il peut également effacer une polygone existante ou en construire une autre.

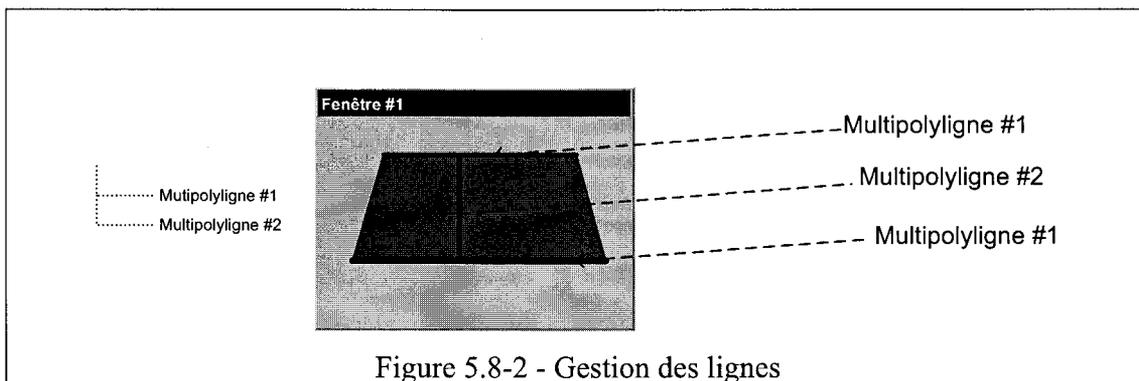


Figure 5.8-2 - Gestion des lignes

## 5.8.4 Gestion des lignes

### 5.8.4.1 Nouveau

L'utilisateur peut créer une nouvelle multipolyligne. Ceci a pour effet d'ajouter un élément à l'arborescence. Visuellement, aucune polygone n'est toutefois ajoutée à la fenêtre graphique. L'utilisateur doit lui-même poser ses sommets pour construire effectivement des lignes dans sa multipolyligne.

### 5.8.4.2 Ouvrir

L'utilisateur peut ouvrir une multipolyligne qu'il a préalablement enregistrée. La géométrie de la multipolyligne est alors chargée.

L'utilisateur peut également ouvrir des isolignes qui ont été préalablement calculées (voir 5.7.1.1). Une isoligne est une multipolyligne qui porte une valeur.

### 5.8.4.3 Enregistrer

L'utilisateur peut enregistrer une multipolyligne. Seule la géométrie de la multipolyligne est enregistrée.

#### 5.8.4.4 Enregistrer sous

L'utilisateur peut enregistrer une multipolyligne sous un autre nom. Seule la géométrie de la multipolyligne est enregistrée.

#### 5.8.5 Analyse

La composante « Analyse » de l'outil de lignes permet à l'utilisateur d'effectuer de l'analyse sur une ou des multipolylignes, la ligne en question servant en quelque sorte de coordonnée curviligne. L'analyse peut se faire sur la composante géométrique (ex : trouver la distance à la berge la plus proche). Elle peut également se faire sur une composante valeur (ex : trouver la valeur minimale de température le long d'une multipolyligne). Lorsque l'utilisateur désire effectuer une analyse sur les valeurs, il doit assigner une source d'information sur laquelle se fera l'analyse.

L'utilisateur peut extraire de l'information géographique d'une multipolyligne. Voici certaines des informations que l'utilisateur peut obtenir à partir d'une multipolyligne :

- la longueur totale;
- la position de chacun de ses sommets;
- la distance directe entre le début et la fin de la ligne (applicable uniquement aux multipolylignes qui contiennent une seule polyligne).

L'utilisateur peut également extraire de l'information de type « valeur » sur les lignes qu'il crée. Pour cela, il doit sélectionner la source d'où proviendra l'information (ex : champ de niveaux d'eau). Voici les informations que l'utilisateur peut ainsi obtenir .

- la valeur minimum;
- la valeur maximum;
- coordonnée  $(x,y)$  de la valeur minimale;
- coordonnée  $(x,y)$  de la valeur maximale;
- moyenne des valeurs;
- écart entre les valeurs minimales et maximales.

L'utilisateur peut également extraire de l'information de type « calcul » sur les lignes qu'il crée. Par exemple, en spécifiant un champ de vitesse (débit spécifique en l'occurrence), l'utilisateur peut faire calculer le débit qui traverse la ligne.

Pour les multipolylignes contenant une seule polyligne, il est possible d'extraire de l'information sur la polyligne à un intervalle spatial régulier (ex : profil longitudinal), et d'afficher l'information dans un graphe. Un composant externe est greffé à Modeleur2 pour effectuer cette tâche de visualisation.

### 5.8.6 Interface usager : analyse des lignes

Le partie interface usager utilisée pour l'analyse des lignes se situe dans la zone <Information> de Modeleur2. En cliquant sur une multipolyligne dans la zone de contrôle (i.e. l'arborescence des multipolylignes), l'utilisateur voit instantanément l'information sur sa multipolyligne mise à jour. L'interface usager de la partie analyse de l'outil de lignes reste à finaliser. La Figure 5.8-3 présente une vue préliminaire de ce qui vient d'être discuté.

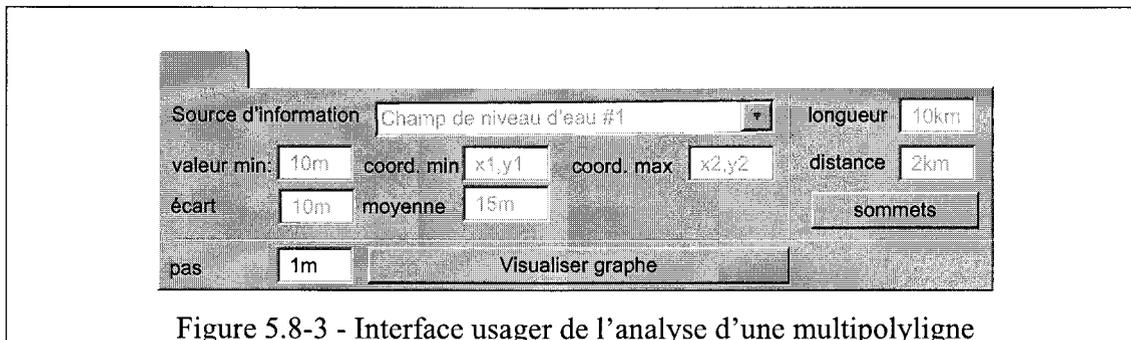


Figure 5.8-3 - Interface usager de l'analyse d'une multipolyligne

L'utilisateur peut sélectionner deux multipolylignes qu'il a construites et les analyser entre elles. Voici certaines des informations que l'utilisateur peut être intéressé à recueillir :

- distance minimale entre les deux lignes;
- distance maximale entre les deux lignes;
- coordonnées  $(x,y)$  de distance minimale entre les deux lignes;
- coordonnées  $(x,y)$  de distance maximale entre les deux lignes.

Par exemple, dans une voie navigable, l'utilisateur pourrait être intéressé à extraire la distance à la berge la plus proche, à partir du chenal de navigation (multipolyligne située au centre dans la Figure 5.8-4).

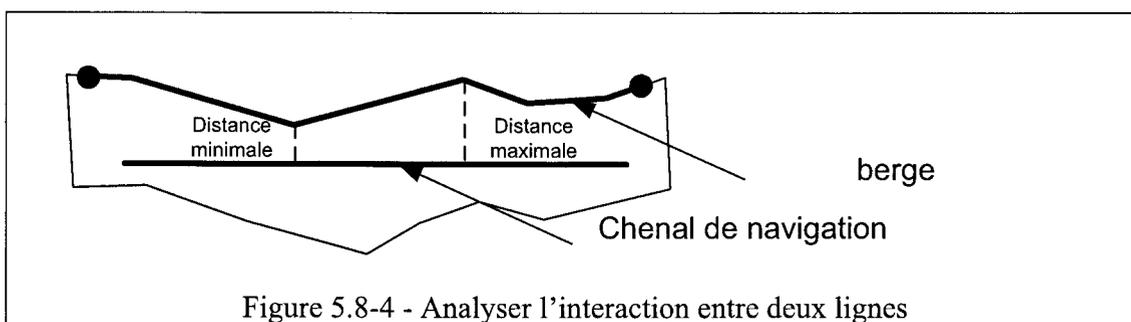


Figure 5.8-4 - Analyser l'interaction entre deux lignes

## 5.9 L'outil de polygones

### 5.9.1.1 Théorie

Afin de bien situer le lecteur en ce qui a trait à la nomenclature utilisée dans cette section, une brève description des concepts de multipolygone et de polygone est présentée dans l'encadré ci-après.

#### **Polygone :**

Un polygone est une surface plane, définie par une frontière extérieure et/ou plusieurs frontières intérieures (des trous) respectant certaines assertions. Pour une liste exhaustive de ces assertions, consulter le document « *OpenGIS Simple Features Specifications for SQL* ».

Parmi les plus importantes, notons que les frontières d'un polygone forment un ensemble de polygones simples et fermés, ne se croisant pas et que l'intérieur de tout polygone forme un ensemble connexe de points.

On dit d'un ensemble de points qu'il est *connexe* s'il est possible de relier toute paire de points de l'ensemble par une polygones continue ne sortant pas de l'ensemble.

Les *sommets* d'un polygone correspondent aux sommets des polygones simples et fermés formant ses frontières.

#### **MultiPolygone :**

Le multiPolygone est défini comme une collection de polygones qui ne peuvent s'intersecter qu'en un nombre fini de points (contrainte SIG).

Exemple :

Le premier multipolygones dans la figure ci-après est invalide (SIG) car l'intersection entre ces deux polygones contient un nombre infini de points. Le second multipolygones, lui, est valide car l'intersection entre les deux polygones contient un nombre fini de points..



L'outil de polygone est très similaire à l'outil de lignes qui fut présenté à la section 5.8 à l'exception que l'on construit des multipolygones au lieu de construire des multipolygones. Pour cette raison, la partie construction et la partie interface usager n'est pas ré expliquée inutilement ici.

### **5.9.1.2 Analyse**

La composante « analyse » de l'outil de polygones permet à l'utilisateur de réaliser des analyses sur un ou des multipolygones. Il peut ainsi trouver la surface d'un polygone.

L'utilisateur peut se servir de cet outil pour analyser des isosurfaces qui ont été préalablement calculées (5.7) et ainsi extraire des statistiques spatiales simples. Les isosurfaces sont constituées de polygones chacun associé à une valeur.

## **5.10 L'outil de vue tabulée**

L'utilisateur peut visualiser certains types de données avec une vue sous forme tabulée (comme il est possible de le faire avec un tableur de style Microsoft Excel<sup>MD</sup>). Les types de données qui se prêtent à ce type de visualisation sont ceux où chaque élément peut être exprimé sous la forme d'une ligne et chaque attribut sous la forme d'une colonne.

Dans Modeleur2, les données de type champ scalaire et champ vectoriel se prêtent bien à la tabulation. L'opération de visualisation tabulée ne sera pas rendue disponible à l'utilisateur pour les types de données sur lesquels elle n'est pas applicable (ex : sur une couche de maillage).

Pour exécuter la tâche de visualisation sous forme tabulée, un outil externe est couplé à Modeleur2. La tâche de Modeleur2 se limite donc à transmettre les lignes de données au logiciel externe. L'utilisateur a alors accès à toutes les fonctionnalités de ce logiciel pour effectuer les traitements qu'il désire sur ces données. L'utilisateur peut par la suite récupérer les résultats modifiés et les réimporter dans Modeleur2.

## **5.11 L'outil de projection**

Comme dans Modeleur 1.0, l'outil de *projection* sert à effectuer le transfert d'un champ appartenant à un maillage vers un autre maillage. La projection est une opération d'interpolation qui exige que le maillage récepteur soit inclus dans le maillage fournisseur. Il y a alors création d'un nouveau champ du même type que celui projeté.

Cette opération est conceptuellement assez simple bien qu'elle comporte certaines contraintes, Nous nous en tenons à cette courte explication pour les fins de ce document.

## **5.12 L'outil de projection cartographique**

Cet outil permet à l'utilisateur de spécifier une projection SIG source, d'entrer une coordonnée dans cette projection, et de faire calculer la dite coordonnée dans la projection destination.

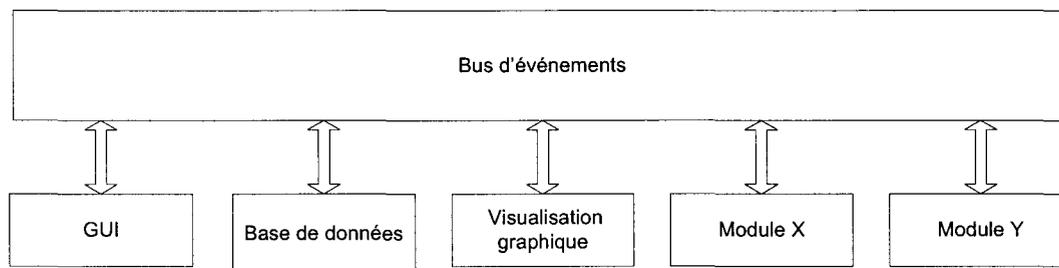


## 6 Architecture logicielle

Lors de la ré-ingénierie de Modeleur2, différentes décisions ont été prises sur le plan technique, notamment l'architecture logicielle. Ce chapitre fait état des différents choix qui ont dicté le développement.

### 6.1 Bus d'événements

L'architecture de Modeleur2 est centrée autour d'un bus d'événements qui a comme fonction de transmettre des événements entre les modules du système.



Plusieurs raisons justifient le choix du bus logiciel comme plaque tournante du logiciel. Voici les principales :

- Le bus d'événements permet de minimiser le couplage (interaction directe) entre les différents modules. Il est donc possible d'en enlever et d'en ajouter sans répercussions indésirables sur le système en entier.
- Il est possible d'interchanger un module par un autre à condition qu'ils répondent tout deux aux mêmes événements. Par exemple, il serait possible de changer l'interface graphique par une autre si on s'assure que les deux répondent aux mêmes événements.
- Il est possible d'enregistrer et d'exécuter des macros en étudiant le flot d'événements et en le reproduisant lors d'une nouvelle exécution. De ce fait, il sera possible de simuler une interaction avec l'interface graphique lors de l'étape des tests.
- Les différents types de liens permettent une extensibilité multi-tâches ou multiprocesseurs.
- La possibilité de construire toute une gamme de logiciels « différents » en déterminant les modules à ajouter.

### 6.1.1 Terminologie de base

**Bus d'événements:** mécanisme ayant pour but d'effectuer la distribution d'événements aux différents modules qui y sont reliés. Le bus d'événements contient une liste des modules et distribuera les événements à chaque module dans cette liste.

**Module:** ensemble de fonctionnalités logicielles, connectées au bus d'événements afin de pouvoir y envoyer des événements et en recevoir. Cet ensemble sera représenté par un objet qui possédera une méthode spécialisée dans la réception d'événements et aura accès à une méthode d'envoi d'événement.

**Lien :** intermédiaire liant le bus et un module. Le lien peut être de différents types : direct, par tâches, distant, etc. La notion de lien permet de garder un faible couplage entre le bus et le module. Ceci permet d'ajouter un type de lien sans avoir à modifier le bus ou les modules.

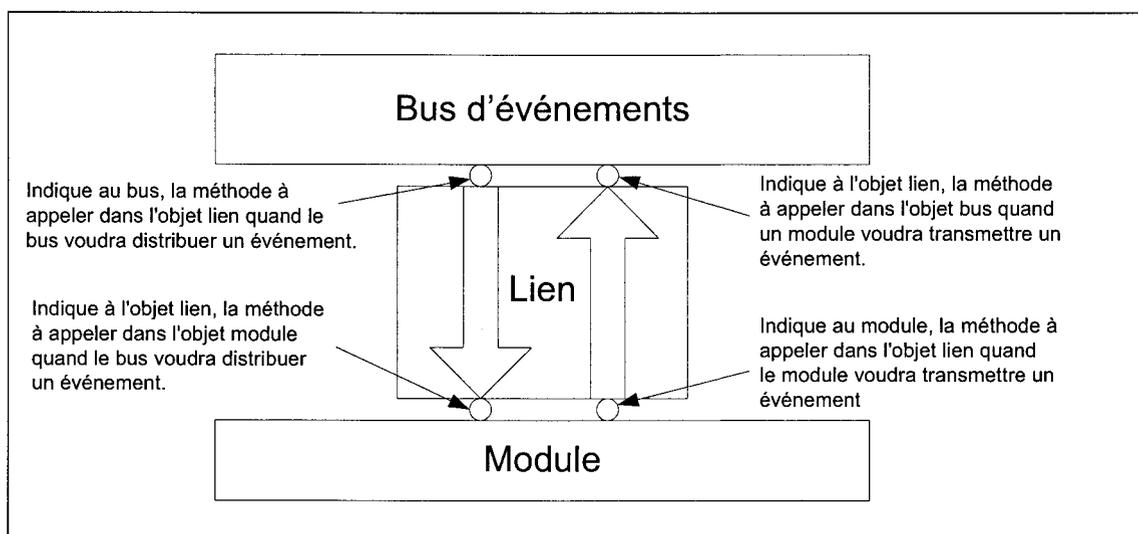
**Événement:** unité d'échange sur le bus d'événements. Un événement peut contenir à la fois de l'information, une commande ou encore rien.

### 6.1.2 Fonctionnement général

L'exécution d'une application utilisant le bus d'événements peut se diviser en différentes étapes:

#### 6.1.2.1 La phase d'initialisation

L'objet « bus d'événements » et tous les modules sont créés. Ensuite, un lien est construit pour chaque module, lien qui est constitué de quatre connections.



A la fin de cette phase, chaque module créé a le pouvoir d'envoyer et de recevoir des événements par l'intermédiaire du bus.

### **6.1.2.2 La phase d'utilisation**

Une fois le bus initialisé, tout objet ayant accès au bus peut lancer des événements. Ces événements sont reçus par le bus. Le bus distribue les événements à chaque module de sa liste de modules connectés. La distribution s'arrête lorsque la fin de la liste est rencontrée ou lorsque que l'un des modules retourne un message d'erreur.

### **6.1.2.3 La phase de fermeture**

À la fermeture de l'application, il est nécessaire d'avertir les différents modules ouverts *via* un événement de fermeture. Ceci doit être fait afin de s'assurer qu'ils sont prêts à être fermés. Suite à l'événement de fermeture, certains modules peuvent à leur tour envoyer un événement sur le bus. Par exemple, un module pourrait envoyer un événement visant à demander à l'utilisateur s'il désire enregistrer son travail avant de quitter.

## **6.1.3 Événements**

Les événements sont à la base de la communication dans le modèle "Module-Événements"; il s'agit en fait de petites classes C++ dérivant d'une classe de base ("l'événement de base").

Les événements sont divisés selon une hiérarchie de classes parents abstraites héritant toutes de l'événement de base. Un événement peut contenir tout type de données et sera transmis tel quel sur le bus. En fait, le bus donne un pointeur vers l'événement au module récepteur; c'est le même événement qui est donné successivement à chaque récepteur. Tout module récepteur sera à même d'accéder à l'événement et d'y appeler les méthodes de requête et d'assignation voulues.

### **6.1.4 Lien de type « direct »**

Dans un contexte d'utilisation où les liens sont de type « directs », comme c'est le cas dans Modeleur2, les échanges s'effectuent via un mécanisme de callback (mécanisme de rappel). Lors de la création d'un module, celui-ci se voit assigné un callback (c'est-à-dire un pointeur vers une méthode) lui permettant d'envoyer de nouveaux événements au bus. Il y a également création d'un callback vers la méthode de réception d'événement du module (*recoisEvenement()*). Ce callback est alors assigné au bus.

À l'intérieur d'un module, il sera donc possible d'utiliser en tout temps le "callback" vers le bus d'événements pour envoyer des événements. L'appel de cette méthode sera fait avec un pointeur d'événement en paramètre.

## 6.2 Base de données

Modeleur2 sera basé sur une base de données relationnelle. Les données contenues dans la base de données au cœur de Modeleur 1.0 pourront être traduites vers la nouvelle base de données avec un outil distribué avec Modeleur2.

L'utilisation d'une base de données relationnelle comporte plusieurs avantages que voici :

- possibilité d'échanger des données avec d'autres bases de données relationnelles ;
- possibilité de faire des requêtes basées sur les données ;
- possibilité de faire du travail partagé sur la même base de données.

Modeleur2 fonctionne localement avec une base de données relationnelle. C'est dans cette base de données que l'utilisateur effectue son travail de tous les jours. Il y a également possibilité d'interagir avec une base de données relationnelle globale, dans laquelle les usagers peuvent publier leurs travaux, et ainsi les rendre accessibles aux autres usagers.

L'engin de base de données utilisé pour les tests est PostgreSQL, et son extension PostGIS. Cette extension permet de manipuler les objets SIG, support standard pour les données de type géographique. Cette base de données sera recommandée aux usagers, et l'utilisateur se verra offrir les facilités nécessaires à sa mise en opération.

D'autres bases de données seront également supportées, notamment Oracle. Les usagers pourront donc opter pour la base de données qui leur convient. La base de données sera développée à l'aide du logiciel Case Studio 2 qui permet de générer des scripts SQL pour la génération des tables. Ces scripts peuvent être optimisés pour chaque base de données.

Pour interagir avec les bases de données de manière transparente, un lien ODBC est utilisé. Pour faciliter la communication avec la base de données, le développement se fera avec les bibliothèques OTL qui permettent d'effectuer la communication *via* des « streams » (flots) en encapsulant le lien ODBC.

La base de données permettra de gérer toutes les données qui faisaient partie de Modeleur 1.0 (champs, maillage), mais permettra en plus la gestion de structures de données propres à Modeleur2 (ex : les MNT, les objets graphiques, les séries, etc).

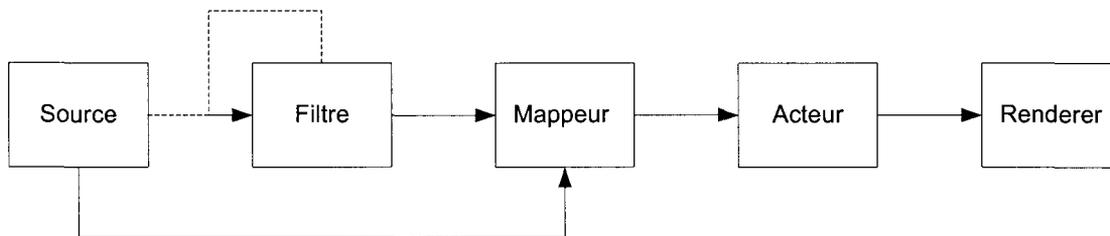
## 6.3 Système de visualisation graphique

Afin de permettre une expansion plus aisée des capacités de représentation graphique, le moteur graphique de Modeleur a été changé.

Le moteur graphique de Modeleur2 sera basé sur VTK (Visual ToolKit). Cette bibliothèque graphique est relativement de haut niveau et s'appuie sur le standard OpenGL. Le choix de la bibliothèque graphique devait respecter les critères suivants:

- être indépendant du système de fenêtrage ;
- être indépendant de la plate-forme ;
- permettre de faire du « picking » ;
- fonctionner sous Windows ou sous cygwin ;
- supporter le 2D et le 3D ;
- faible coût, ou gratuit.

VTK est basé sur le paradigme par flux de données. Dans ce paradigme, les données circulent dans un réseau de modules. Ces modules sont en fait des algorithmes qui modifient les données qui entrent dans ceux-ci. Le diagramme suivant illustre la structure typique d'un programme utilisant VTK (les éléments en pointillé dans le schéma sont facultatifs) :



Voici la signification de chacune des composantes du diagramme :

Source :	correspond à la source des données caractérisant l'objet à afficher. Par exemple, si on veut afficher un cône, une instance de la classe <code>vtkConeSource</code> serait la source.
Filtre	correspond à un filtre qui traite les données qui y entrent pour obtenir des données modifiées en sortie.
Mappeur	sert à interpréter les données qui entrent en primitives graphiques comme des points, des lignes et des polygones.
Acteur	correspond à l'élément de la scène. C'est avec ce dernier qu'il est possible d'interagir, de positionner et de modifier la couleur et l'opacité de l'objet à afficher.
« Renderer »	est la scène contenant tous les acteurs. Il s'agit ici d'une vue (« viewport ») qui sera contenue dans une fenêtre. Une fenêtre peut ainsi contenir plusieurs « renderers ».

## 6.4 Plug-in

Un *plug-in* est une extension d'un logiciel permettant d'ajouter des fonctionnalités à celles qui sont offertes dans sa version de base.

Modeleur2 pousse plus loin cette notion en considérant comme « plug-in » tout ce qui n'est pas essentiel à son fonctionnement « minimal ». L'architecture sous forme de bus

d'événements permet de voir le logiciel comme une charpente à laquelle on peut ajouter des fonctionnalités.

Dans Modeleur2, un plug-in est composé d'une ou plusieurs DLL, d'un ou plusieurs modules Python, de l'aide et de fichiers de langues.

Certains modules ont besoin d'interagir (interfacer) avec l'interface graphique (GUI ou Graphical User Interface). Ceux-ci ont la possibilité d'ajouter des contrôles dans la zone de contrôle, d'ajouter des menus dans la zone de menus, d'ajouter une barre d'outils dans la barre d'outils principale du logiciel, et de s'enregistrer pour l'écoute d'événements provenant du C++ recevoir des événements du C++.

L'utilisateur se voit offrir le moyen de configurer les plug-in à connecter au bus au démarrage de l'application.

### **6.4.1 Ajout : partie C++**

Modeleur2 offre des mécanismes pour intégrer dynamiquement à l'exécution les nouveaux modules (plug-in). Pour ce faire, l'utilisateur doit encapsuler son module dans une DLL.

Un module doit dériver du module de base (`TEModuleBase`). Il doit également spécialiser la fonction virtuelle d'envoi et de réception d'événements.

Une fois ces étapes réalisées, le module se voit octroyer les capacités pour transmettre des événements sur le bus et en recevoir. Le module de l'utilisateur peut donc interagir avec tout le reste du système *via* l'échange d'événements.

L'utilisateur peut ajouter ses événements au système en respectant l'interface C++ prescrite. L'utilisateur doit également publier ceux-ci dans Python. Les événements doivent être publiés dans Python car le module de macro doit être en mesure de capter et reproduire tous les événements qui circulent sur le bus. La génération des événements C++ vers Python peut être faite en utilisant la librairie Boost.Python.

### **6.4.2 Ajout : partie BD**

Certains plug-in offrent la possibilité de créer, charger et enregistrer des données dans la base de données. Pour cela, ils doivent aller inscrire les algorithmes (`GDAIo`) pour effectuer ces opérations. Les algorithmes incluent des requêtes SQL pour effectuer la lecture et l'écriture dans une base de données. Tout plug-in qui désire utiliser les fonctionnalités offertes par ces algorithmes y aura accès car elles sont disponibles à tous.

## **6.4.3 Ajout : partie GUI**

### **6.4.3.1 Gestion des menus**

Les plug-in qui le désirent peuvent venir ajouter des menus à la barre de menus de Modeleur2. Ils peuvent ainsi rendre disponibles leurs fonctionnalités au monde extérieur

Tout plug-in intéressé à ajouter des menus doit indiquer au gestionnaire de menus quels sont les menus / sous-menus qu'il désire ajouter. Le plug-in doit également spécifier à quel endroit (chemin) dans l'arborescence de menus ceux-ci doivent être ajoutés. Enfin, le plug-in doit associer un callback (fonction à appeler dans le script Python) à chaque entrée de menu qu'il ajoute. Quand l'utilisateur clique sur ce menu, le callback est appelé et le travail s'effectue.

Le plug-in doit également être en mesure de répondre à un événement qui est régulièrement envoyé par l'interface graphique afin de savoir quelles commandes doivent être actives. Le plug-in doit alors indiquer toutes les commandes qu'il désire rendre disponible à l'utilisateur (entrée de menu non grisée dans la barre de menu).

### **6.4.3.2 Gestion des zones**

L'interface graphique (GUI) de Modeleur2 est constituée de trois zones. Deux de ces zones sont des zones de contrôle, c'est-à-dire que des contrôles (ex : arborescence) peuvent y être ajoutés. Chaque module peut inscrire les contrôles qu'il désire dans le gestionnaire de contrôles. Quand un plug-in a besoin de ce contrôle, le gestionnaire le construit et l'ajoute dans la zone de contrôle.

### **6.4.3.3 Gestion des barre d'outils**

Le GUI de Modeleur2 est constitué d'une zone de barres d'outils. Dans cette zone, les plug-in peuvent insérer une barre d'outils. L'état des boutons de la barre d'outils doit être géré par les plug-in.

### **6.4.3.4 Gestion des événements**

La partie GUI de certains plug-in doit parfois être en mesure de recevoir des événements du code C++. Ces plug-in doivent donc s'inscrire comme « écouteur » pour un type d'événement particulier. Quand le gestionnaire d'événements recevra ce type d'événement, il appellera tous les callbacks qui auront été enregistrés par les plug-in.

## **6.5 Langage de script**

### **6.5.1 Interface utilisateur (UI)**

L'interface graphique (GUI) est entièrement pilotée à partir de Python (et wxPython). C'est donc dire qu'elle est interprétée à l'exécution. L'interface graphique de Modeleur2 peut donc être modifiée en changeant le script, sans qu'on ait à recompiler quoi que ce soit.

### **6.5.2 Calculatrice**

Le langage de script est également utilisé comme langage de programmation de la calculatrice. Ce langage de programmation permet l'utilisation du code de Modeleur2 à partir du script écrit par l'utilisateur. Ceci permet à l'utilisateur d'avoir accès à beaucoup de fonctionnalités internes de Modeleur2. L'utilisation de la librairie externe Boost-Python permet également de publier aisément toutes les structures de données internes à Modeleur2 et qui sont intéressantes pour l'utilisateur.

### **6.5.3 Macros**

Le langage de script est utilisé pour enregistrer les événements qui circulent sur le bus dans un script. Il est alors possible de rejouer la suite d'événement.

## **6.6 Installation via le web**

Modeleur2 s'installe à partir du web. En d'autres mots, les usagers n'ont pas de livrable à proprement parler. Ils ont toujours accès à la plus récente version de Modeleur2 en effectuant une opération de mise à jour. L'installation tient compte de la version de chaque composant et de ses dépendances.

## **7 Configuration, aide, langue et distribution**

### **7.1 Aide**

L'utilisateur a accès à une aide contextuelle pour la plupart des fonctionnalités de Modeleur2. L'utilisateur a également accès à des bulles d'aide quand la situation s'y prête. Un tutoriel sera également développé pour permettre de diminuer la courbe d'apprentissage.

### **7.2 Langue**

Le logiciel Modeleur2 est neutre quant à la langue, dans le sens où tous les messages d'erreur qui circulent sont identifiés par une chaîne de caractères. Celle-ci est par la suite traduite dans la langue qui a été choisie lors de l'installation du logiciel. Le logiciel est d'abord livré en français. Une version anglaise sera faite par la suite.

### **7.3 Distribution**

Le logiciel est livré sous la forme d'une série de DLL dont plusieurs correspondent aux plug-in disponibles. Une série de scripts Python accompagnent également le logiciel. Des fichiers de configuration permettent à l'utilisateur de déterminer les plug-in qu'il veut voir rattacher à Modeleur2.

Toute une gamme de composants externes (exécutables, DLL) qui sont nécessaires au bon fonctionnement de Modeleur2 sont également inclus dans la distribution.



## 8 Glossaire

**API :**

Application Programming Interface : Interface de programmation d'applications. Ensemble de fonctions qui répondent à une interface. Le programmeur utilise l'API sans avoir à se soucier de l'implémentation sous-jacente.

**Arborescence :**

Représentation organisationnelle selon une structure arborescente, qui établit la stricte hiérarchie entre les divers composants, de façon que toute information, sauf la première, procède d'une seule autre, mais peut en engendrer plusieurs.

**Base de données relationnelle :**

Base de données organisée en fonction des relations qui existent entre les données. Le principal langage utilisé dans les bases de données relationnelles est le SQL.

**Bus d'événements:**

Mécanisme propriétaire à Modeleur2 ayant pour but d'effectuer la distribution d'événements aux différents modules qui y sont reliés. Le bus d'événements contient une liste des modules et distribue les événements à chaque module dans cette liste.

**Calcul:**

Composante de base d'une simulation dans Modeleur2. Chaque calcul est réalisé à l'aide de paramètres d'entrée et produit une série de résultats (ex : champ de niveaux d'eau et champs de vitesses). Les résultats d'un calcul peuvent à leur tour servir de paramètres d'entrée (i.e. solution initiale) à un autre calcul.

**Callback :**

Dans le contexte de la programmation informatique, procédure appelée lorsque certains événements préalablement définis se produisent.

**Champ :**

Dans Modeleur2, un champ est une structure de données qui possède en plus des propriétés algébriques de type algèbre, des propriétés spatiales. Le champ porte des données qui répondent elles aussi aux structures algébriques et sont associées à une position dans l'espace. Il est possible d'interroger le champ au sujet de sa valeur en tout point compris dans la région spatiale qu'il englobe. Sous Modeleur2, les champs peuvent être qualifiés par la valeur qu'ils portent (ex : valeur scalaire, vectorielle ou ellipse d'erreur) et peuvent être qualifiés par la méthode qu'ils utilisent pour arriver à fournir la valeur en tout point (ex : éléments finis, analytique, etc).

**Champ analytique :**

Le champ analytique est un champ défini par une équation mathématique. (ex :  $f(x,y) = 2x + y$ ).

**Champs éléments finis :**

Le champ éléments finis est la structure de données rassemblant les valeurs nodales sur un maillage éléments finis. Une valeur en tout point de la région spatiale du champ est obtenue par interpolation d'éléments finis.

**Champ d'ellipse erreur :**

Type de champ pour lequel la donnée portée correspond aux paramètres d'une ellipse décrivant l'erreur à ce point.

**Champ publié :**

Dans Modeleur2, le terme champ publié fait référence à un semi de points qui a été validé par l'utilisateur, et que l'on ensuite transformé en champ en générant un maillage avec les points du semi validé.

**Champ scalaire :**

Type de champ pour lequel la donnée portée correspond à une valeur réelle.

**Champ vectoriel :**

Type de champ pour lequel la donnée portée correspond à un vecteur.

**Condition (d'une partition de conditions aux limites) :**

Une condition d'une partition de conditions aux limites correspond à une valeur de paramètre (ex : niveau d'eau, débit, etc) qui doit prévaloir à cette limite du domaine. Les conditions dépendent du simulateur utilisé.

**Connexion ODBC :**

Opération qui permet d'établir une liaison entre une application et une base de données via une interface ODBC.

**Condition aux limites :**

Dans Modeleur2, le terme « condition aux limites » correspond au résultat d'une opération d'assemblage d'une partition de conditions aux limites sur un maillage. Il est bien important de distinguer ce terme de celui de condition qui lui est une composante d'une partition de conditions aux limites (voir Condition).

**Couche :**

Dans Modeleur2, une couche est une construction géométrique constituée d'un ou plusieurs sous-domaines fermés. Chaque couche est associée à un jeu de données ou à des paramètres. Chaque couche se voit assigner un ordre de priorité, qui permet de déterminer de quelle couche le jeu de donnée ou les paramètres doivent être extraits lorsque deux ou plusieurs couches se chevauchent dans une partition.

**Couche de maillage:**

Couche appartenant à une partition de maillage. La donnée portée par la couche correspond aux paramètres de maillage utilisés pour mailler.

**Couche de données d'un MNT (ex : couche de topographie):**

Couche appartenant à une partition dans un MNT (ex : partition de topographie). La donnée portée par la couche correspond au champ qui décrit la couche.

**C++ :**

Langage de programmation de haut niveau défini comme une version orientée objet du langage C. Le C++ est le principal langage de programmation utilisé dans l'écriture du code source de Modeleur2.

**Degré de raffinement d'un maillage :**

Correspond à la grosseur des mailles d'un maillage. Un haut niveau de raffinement implique des mailles plus petites, et un faible degré de raffinement, des mailles plus grosses.

**DLL :**

Dynamic Linked Library (bibliothèque de liens dynamiques). Ensemble de routines qui sont disponibles aux applications au moment de l'exécution.

**Échelle de précision :**

L'échelle de précision est une valeur qui qualifie l'erreur générée par le retrait d'un point. Par exemple, dans un semi de points de topographie, un terrain de soccer parfait peut être décrit par quatre points. Tous les autres points peuvent avoir un échelle de précision inférieure à ces quatre points car leur retrait n'enlève rien à la description du terrain.

**Espace de représentation :**

Dans Modeleur2, un espace de représentation correspond au contenu d'une fenêtre graphique. Un espace de représentation est composé de trois plans : plan arrière, plan médian et plan interactif.

**Événement :**

Dans Modeleur2, ceci correspond à l'unité d'échange sur le bus d'événements. Un événement peut contenir à la fois de l'information, une commande ou encore rien.

**Fenêtre graphique :**

Fenêtre dans laquelle il y peut y avoir affichage de représentations graphiques.

**Filtrage :**

Dans Modeleur2, le terme filtrage fait référence à l'opération qui permet distinguer les points d'un semi qui doivent être conservés de ceux qui doivent être désactivés. L'opération de filtrage permet donc de nettoyer le semi brut des points indésirables qu'il peut contenir.

**Filtre primaire :**

Les filtres primaires sont utilisés pour calculer une cote de qualité à chaque point du semi introduit à l'entrée du filtre. Ces filtres peuvent baser leur appréciation sur différents paramètres, et ils attribuent à chaque point du semi une cote entre 0 et 1.

**Filtre secondaire :**

Les filtres secondaires sont les filtres qui travaillent à partir des cotes produites par les filtres primaires. Ils ont comme rôle d'indiquer si un point doit être gardé ou retiré en se basant sur les cotes combinées des filtres primaires et d'une valeur seuil.

**Focus :**

Zone d'une fenêtre active où l'utilisateur fera sa prochaine intervention. Ceci coïncide avec la position du curseur, après que l'utilisateur a cliqué sur la souris.

**GIS :**

Geographic Information System. Voir SIG.

**GUI:**

Graphical User Interface (interface utilisateur graphique). Interface utilisateur basée sur l'emploi d'éléments graphiques tels que les fenêtres, les icônes et les menus, qui visent la simplicité d'emploi et qui créent un environnement de travail convivial.

**IDE :**

Integrated Development Environment (environnement de développement intégré). Dans un système de développement, ensemble d'outils intégrés qui sont utilisés pour le développement de logiciels et qui sont directement accessibles à partir de l'interface utilisateur.

**Interpolation :**

Estimation ou calcul de valeurs intermédiaires dans une série de valeurs connues.

**Interpréteur :**

Programme qui traduit les instructions d'un langage évolué en langage machine et les exécute au fur et à mesure qu'elles se présentent.

**Isolignes :**

Ligne qui joint des points correspondants à une même valeur. Comme exemple d'isolignes, on peut mentionner les courbes de niveau.

**Isosurfaces :**

Surface qui englobe des points correspondant tous à une même plage de valeurs.

**Jeu de données :**

Dans Modeleur2, le terme jeu de données fait référence aux champs publiés à l'étape de validation, et que l'utilisateur doit spécifier lors de la construction d'un MNT.

**Langage interprété (langage de scripts) :**

Langage de programmation évolué qui traduit en langage machine le code source d'un programme (d'un script), pour l'exécuter immédiatement, et ce, ligne par ligne.

**Liens externes :**

Dans Modeleur2, un lien externe correspond à un lien établi avec une donnée appartenant à un projet autre que le projet courant. Par exemple, le projet 'a' peut avoir comme lien externe, le maillage du projet 'b'.

**Liste d'affichage :**

Représentation sous forme d'une arborescence du contenu de chacune des fenêtres graphiques, et pour chacune d'elles du contenu de ses plans, et des objets graphiques contenus par ceux-ci.

**Log :**

Relevé chronologique des opérations informatiques constituant un historique de l'utilisation. Dans Modeleur2, le log (journal) fournit à l'utilisateur de l'information sur le déroulement des opérations exécutées. Le log peut être configuré pour afficher différents niveaux de détail.

**Macro :**

Séquence de commandes, de touches de fonction et d'instructions enregistrée sous un nom, qu'on peut rappeler et exécuter par une commande unique ou par le nom qui lui a été attribué. Dans Modeleur2, une macro correspond en fait à une suite d'événements enregistré dans un script Python.

**Maillage:**

Surface ou volume généré par un ensemble de nœuds et de lignes de contrôle dont les caractéristiques sont définies par les points ou les courbes qui ont servi à la création de l'élément.

**Mailleur :**

Dans Modeleur2, le terme mailleur correspond à la partie de logiciel utilisée pour générer un maillage (i.e. générer les nœuds et les éléments).

**Métadonnées :**

Donnée qui renseigne sur la nature de certaines autres données et qui permet ainsi leur utilisation pertinente.

**Méthode des éléments finis :**

La méthode des éléments finis est une technique d'interpolation qui consiste à discrétiser le domaine de calcul en un ensemble de sous domaines de géométrie simple, les éléments. Sur chaque élément, on définit une fonction d'interpolation qui prend en compte la disposition des nœuds de l'élément (géométrie) et les valeurs qu'ils portent.

**MNT :**

Modèle Numérique de Terrain. Dans Modeleur2, le MNT correspond à un ensemble de partitions qui décrivent le terrain (ex : partition de topographie, partition de substrat, etc).

**Module C++ :**

Partie C++ d'un plug-in Modeleur2 connectée au bus d'événements afin de pouvoir y envoyer des événements et en recevoir.

**Module Python :**

Un module Python est un élément qui fournit de nouvelles fonctions à un interpréteur Python. Dans Modeleur2, les scripts, les macros et les DLL Python (.pyd) sont considérés comme des modules Python.

**Multipolygone :**

Ensemble de polygones.

**Multipolyligne :**

Ensemble de polylignes.

**Niveau de tolérance à l'erreur :**

Dans Modeleur2, le niveau de tolérance à l'erreur correspond au niveau d'erreur acceptable pour lequel une couche est considérée maillée avec un degré de raffinement acceptable.

**Objet graphique :**

Dans Modeleur2, toute représentation affichée dans une fenêtre graphique est considérée comme un objet graphique. Par exemple, le contour d'une couche est un objet graphique de même que la représentation graphique d'un maillage.

**ODBC :**

Interface standardisée, définie par Microsoft, permettant d'accéder à des bases de données ayant des formats différents.

**Onglet :**

Partie qui semble dépasser du bord d'un élément affiché dans une fenêtre et qui sert à accéder facilement à des fonctions, des réglages, des rubriques d'aide, des dossiers ou des documents, en cliquant dessus avec un dispositif de pointage tel que la souris.

**OTL :**

Librairie qui permet d'effectuer la communication avec la base de données via les opérateur de flot (>> et <<) pour lire de la base de données et écrire dans la base de données) en encapsulant le lien ODBC.

**Partition :**

Sous Modeleur2, le terme partition fait référence à un ensemble de couches.

**Partition de conditions aux limites :**

Type de partition qui décrit les conditions aux limites d'un domaine. Ce type de partition sera utilisé pour procéder à l'assemblage des conditions aux limites sur un maillage.

**Partition générique :**

Type de partition qui portent un type de donnée qui a été défini par l'utilisateur. La partition générique peut supporter l'interpolation ou non.

**Partition de maillage :**

Type de partition dont les couches servent à générer un maillage.

**Partition spécialisée :**

Type de partition prédéfini dans Modeleur2 (topographie, substrat, etc) par opposition avec les partitions générique qui elle sont définies par l'utilisateur.

**Plug-in :**

Extension à une application qui vient se loger dans l'application elle-même. Une fois installé, on peut avoir accès aux fonctionnalités du plug-in de façon tout à fait transparente.

**Polygone :**

Figure formée par une suite ordonnée de segments (côtés), dont chacun a une extrémité commune (sommet) avec le précédent et le suivant.

**Polylignes :**

Entité géométrique constituée d'une succession de lignes parfaitement reliées les unes aux autres par leurs extrémités.

**Projection :**

Transposition d'une portion de l'ellipsoïde de référence géodésique représentant la surface terrestre, sur une surface plane, à l'aide d'un modèle mathématique.

**Projet :**

Dans Modeleur2, le projet est le contenant qui héberge toutes les données et résultat de l'utilisateur dans le cadre d'une étude.

**Python :**

Un langage de programmation interprété, interactif, orienté-objet comparable à Tcl, Perl ou Java.

**RAD :**

Rapid Application Development. Développement rapide d'application avec des outils modernes.

**Région spatiale:**

Région définie par des coordonnées  $\{x,y,z\}$  de l'espace géographique.

**Représentation graphique :**

Dans Modeleur2, le terme représentation graphique fait référence au contenu d'une fenêtre graphique. Tout affichage dans une fenêtre graphique est une représentation graphique (ex : affichage d'un maillage, affichage d'une partition, etc.).

**Schéma :**

Dans un système de gestion de base de données (SGBD), description d'une base de données créée au moyen du langage de définition de données proposé par le SGBD. Un schéma permet de regrouper un ensemble de tables. Dans Modeleur2, chaque projet est associé à un schéma.

**Script :**

Série d'instructions servant à accomplir une tâche particulière.

**Semi de points :**

Sous Modeleur2, le terme semi de points correspond à un ensemble de points  $\{xy, valeur\}$ . À titre d'exemple, pour un semi de points de topographie, la valeur correspond à l'altitude (z).

**Série :**

Une série peut être vue comme un champ éléments finis, mais pour lequel le domaine n'est pas une région spatiale mais plutôt une région d'un espace conceptuel de solution. Tout comme le maillage, la série possède des nœuds et des éléments, lesquels sont utilisés pour interpoler à partir de valeurs portées par les nœuds. L'emplacement de chacun de nœuds est spécifié via une coordonnée. Typiquement, la coordonnée d'une série est entière ou réelle. Une série peut être en 1D et 2D, et théoriquement, pourrait même être en n dimensions. Les nœuds d'une série portent des valeurs. Les valeurs peuvent être simples comme des scalaires (ex : niveau d'eau), des vecteurs (ex : vitesse) mais peuvent être

également des champs répartis spatialement. Les données d'une série doivent être en relation.

**SIG :**

Système d'information portant sur des données géographiques.

**Simulateur :**

Dans Modeleur2, le terme simulateur fait référence au logiciel utilisé pour effectuer un calcul, en fonction de paramètres d'entrée.

**Simulation :**

Une simulation est une suite d'étapes de calcul (ex : hydrodynamique) qui a pour but d'obtenir une solution finale satisfaisante

**Sous-domaine :**

Polygone géoréférencé qui définit une couche ou une partie d'une couche.

**Squelette du MNT :**

Correspond à la plus petite zone commune à un ensemble de partitions spécifié par l'utilisateur. Il peut servir lors de la construction d'un maillage pour s'assurer que le maillage ne dépasse pas la zone de données partagée par toutes les partitions d'intérêt.

**SQL :**

Langage d'interrogation, de mise à jour et de gestion des bases de données relationnelles.

**Table :**

Mode d'organisation des données dans une base de données, composé de colonnes et de lignes, dont chaque cellule contient des informations et des liens qui existent entre les données. La table constitue la structure de base de la relation entre les données.

**Trigger :**

Mécanisme qui amène l'exécution automatique d'une procédure lorsqu'un utilisateur s'apprête à modifier le contenu d'une base de données. En mettant en marche une procédure déterminée lorsqu'un événement suspect se présente, le déclencheur assure le maintien de l'intégrité des données dans la base de données.

**VTK :**

Visual ToolKit. Librairie graphique qui offre des fonctionnalités haut niveau. Cette librairie est utilisée dans Modeleur2 pour effectuer toute tâche de représentation graphique.

**Vue BD :**

Dans les bases de données relationnelles, table logique créée par la spécification d'opérations relationnelles reliant plusieurs fichiers. Dans de nombreux systèmes, une vue peut être manipulée comme s'il s'agissait d'une table réelle.

**Zone d'affichage :**

Dans Modeleur2, la zone d'affichage correspond à la zone délimitée par une fenêtre graphique. Plus l'utilisateur zoom in, plus la zone d'affichage est réduite, et inversement, plus il zoom out, plus elle est agrandie.

**Zone de contrôle :**

Dans Modeleur2, la zone de contrôle correspond à la partie de l'interface utilisateur où l'utilisateur peut gérer ses données. Cette zone contient entre autres les arborescences des différents types de données, mais pourrait également accueillir tout type de contrôle offert par un plug-in.

**Zone d'étude :**

Zone (ou région) d'intérêt sur laquelle l'utilisateur désire porter son étude. Cette zone d'étude limitera les données chargées lors des requêtes de chargement.

Sources de certaines définitions : <http://www.grandictionnaire.com>