

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

SPÉCIFICATIONS – MODELEUR 2.0

Rapport de recherche No R-720-b1

Janvier 2004

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

Projet SEGRI

Spécifications – Modeleur 2.0

Présenté au

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

20 janvier 2004

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
Ingénieur numérique

Pour les fins de citation : **Secretan Y., Larouche E. & coll. (2003).**

Système d'Évaluation et de Gestion des Risques d'Inondation en milieu fluvial (SEGRI) :
Spécifications – Modeleur 2.0. Québec, INRS-Eau, Terre & Environnement. 68 pages.
(INRS-Eau, Terre & Environnement, rapport de recherche 720 b1)

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

©INRS-Eau, Terre & Environnement, 2003
ISBN : 2-89146-515-6

AVANT PROPOS	1
1. INTRODUCTION	3
1.1 SOMMAIRE	3
1.2 SYSTÈME EXISTANT	3
1.3 STRUCTURE DU DOCUMENT.....	4
2 DESCRIPTION DES CONCEPTS	5
2.1 CONCEPTS CLÉS À LA BASE DE MODELEUR	5
2.1.1 <i>Cohérence</i>	5
2.1.2 <i>Souplesse</i>	5
2.2 BASE DE DONNÉES RELATIONNELLE.....	5
2.3 ZONES D'ÉTUDE	5
3 DESCRIPTION FONCTIONNELLE	7
3.1 ACTIVITÉ : GESTION DES DONNÉES.....	8
3.1.1 <i>Gestion de projets</i>	8
3.1.1.1 Nouveau	8
3.1.1.2 Ouvrir.....	8
3.1.1.3 Fermer.....	9
3.1.2 <i>Importer des données</i>	9
3.1.2.1 À partir de fichiers	9
3.1.2.2 À partir d'une base de données externe	9
3.1.3 <i>Exporter des résultats</i>	9
3.1.3.1 Vers une base de données externe	9
3.1.3.2 Vers des fichiers.....	9
3.1.4 <i>Voir le nom des données d'un projet</i>	9
3.1.5 <i>Effacer une donnée de projet</i>	9
3.1.6 <i>Voir le graphe des dépendances d'une donnée</i>	9
3.1.7 <i>Voir les informations sur une donnée</i>	9
3.1.8 <i>Visualiser les données (vue tabulée)</i>	9
3.1.9 <i>Trier les données</i>	10
3.2 ACTIVITÉ : VALIDATION DES DONNÉES DE TERRAIN	10
3.2.1 <i>Gestion</i>	12
3.2.1.1 Nouveau	12
3.2.1.2 Ouvrir.....	12
3.2.1.3 Fermer	12
3.2.1.4 Enregistrer.....	12
3.2.1.5 Enregistrer sous.....	12
3.2.2 <i>Points</i>	12
3.2.2.1 Sélectionner des points	12
3.2.2.2 Créer des points.....	13
3.2.2.3 Activer / désactiver des points	13
3.2.2.4 Déplacer des points	14
3.2.2.5 Effacer des points.....	14
3.2.2.6 Importer des points	14

3.2.3	<i>Information portée par les points</i>	14
3.2.3.1	Editer l'information	14
3.2.3.2	Filtrer l'information	14
3.2.3.3	Lisser l'information de jeux de données limitrophes.....	17
3.2.3.4	Afficher l'information portée par les points de terrain	18
3.2.3.5	Voir les données dans une table.....	19
3.2.4	<i>Gestion du temps</i>	19
3.2.4.1	Sélectionner le pas de temps actif.....	19
3.2.4.2	Comparer les données de différents pas de temps	19
3.2.4.3	Enchaîner les données sous forme d'animation.....	19
3.3	ACTIVITÉ : MNT	19
3.3.1	<i>Interface usager</i>	20
3.3.2	<i>Gestion d'un MNT</i>	21
3.3.2.1	Nouveau	21
3.3.2.2	Ouvrir.....	21
3.3.2.3	Fermer	21
3.3.2.4	Enregistrer.....	21
3.3.2.5	Enregistrer sous.....	22
3.3.3	<i>Gestion des partitions d'un MNT</i>	22
3.3.3.1	Nouvelle partition	22
3.3.3.2	Ajouter une partition existante.....	22
3.3.3.3	Enregistrer.....	22
3.3.3.4	Enregistrer sous.....	23
3.3.3.5	Retirer	23
3.3.4	<i>Partitions spécialisées</i>	23
3.3.4.1	Afficher les limites externes de jeux de données (peaux).....	25
3.3.4.2	Ajouter une couche de données	25
3.3.4.3	Éditer une couche de données.....	25
3.3.4.4	Copier et coller une couche.....	26
3.3.4.5	Retirer une couche de données	26
3.3.4.6	Modifier la couleur des jeux de données	26
3.3.4.7	Modifier la priorité d'une couche de données	26
3.3.5	<i>Partitions génériques</i>	26
3.3.5.1	Définir la partition générique.....	27
3.3.5.2	Construire la partition générique	27
3.3.6	<i>Édition de sous-domaines</i>	27
3.3.6.1	Créer un sous-domaine	27
3.3.6.2	Sélectionner un sous-domaine	28
3.3.6.3	Effacer un sous-domaine.....	28
3.3.6.4	Fusionner deux sous-domaine.....	28
3.3.6.5	Ajouter un sommet.....	28
3.3.6.6	Relier deux sommets.....	28
3.3.6.7	Enlever un sommet	28
3.3.6.8	Exemple	28
3.4	ACTIVITÉ : MAILLAGE	29
3.4.1	<i>Théorie relative aux partitions de maillage</i>	29

3.4.2	<i>Paramètres de maillage</i>	30
3.4.2.1	Taille des mailles	30
3.4.2.2	Champ d'erreur	30
3.4.3	<i>Interface usager pour la gestion d'une partition de maillage</i>	31
3.4.4	<i>Exemple d'une partition de maillage</i>	31
3.4.5	<i>Gestion d'une partition de maillage</i>	32
3.4.5.1	Nouveau	32
3.4.5.2	Ouvrir	32
3.4.5.3	Fermer	32
3.4.5.4	Enregistrer	32
3.4.5.5	Enregistrer sous	33
3.4.6	<i>Gestion des couches paramètres-maillage</i>	33
3.4.6.1	Ajouter une couche paramètres-maillage	33
3.4.6.2	Retirer une couche paramètres-maillage	33
3.4.6.3	Copier et coller une couche paramètres-maillage	33
3.4.6.4	Modifier la priorité d'une couche paramètres-maillage	33
3.4.6.5	Modifier la couleur d'affichage des couches paramètres-maillage	33
3.4.7	<i>Édition</i>	33
3.4.7.1	Afficher le squelette d'un MNT	33
3.4.7.2	Afficher les contours des autres couches	34
3.4.7.3	Éditer une couche paramètres-maillage	34
3.4.7.4	Assigner des paramètres de maillage à un sous-domaine	34
3.4.7.5	Trouver le niveau de tolérance optimal	35
3.4.7.6	Afficher les zones à mailler	35
3.4.7.7	Mailler une zone à mailler	35
3.4.8	<i>Génération</i>	35
3.4.8.1	Générer le maillage final	35
3.4.8.2	Générer les champs du MNT	35
3.5	ACTIVITÉ : SIMULATION	36
3.5.1	<i>Gestion d'une simulation</i>	36
3.5.1.1	Nouveau	36
3.5.1.2	Ouvrir	36
3.5.1.3	Fermer	36
3.5.1.4	Enregistrer	36
3.5.1.5	Enregistrer sous	36
3.5.2	<i>Définir les paramètres</i>	36
3.5.2.1	Paramètres simulation	36
3.5.2.2	MNT	36
3.5.2.3	Maillage	36
3.5.3	<i>Définir les conditions aux limites</i>	36
3.5.4	<i>Définir des résultats d'intérêt</i>	36
3.5.5	<i>Exécuter une simulation (calcul)</i>	36
3.5.5.1	Local	36
3.5.5.2	À distance	36
3.5.6	<i>Analyser l'historique de la simulation</i>	36
3.5.6.1	Vue d'un graphique de simulation	36

3.5.6.2	Vue des paramètres d'entrées/sortie à chacune des étapes	36
3.5.6.3	Modification de paramètres x itération en arrière.....	36
3.6	ACTIVITÉ : VISUALISATION DE DONNÉES.....	36
3.6.1	<i>Gestion de la visualisation</i>	36
3.6.1.1	Ouvrir.....	36
3.6.1.2	Fermer.....	37
3.6.1.3	Enregistrer.....	37
3.6.1.4	Enregistrer sous.....	37
3.6.2	<i>Visualiser graphiquement des données</i>	37
3.6.2.1	Afficher des maillages	37
3.6.2.2	Afficher des isolignes	37
3.6.2.3	Afficher des isosurfaces.....	37
3.6.2.4	Afficher des vecteurs	37
3.6.2.5	Afficher des nœuds en 2D.....	37
3.6.2.6	Afficher des nœuds en 3D (topographie).....	37
3.6.2.7	Afficher des profils curvilignes	37
3.6.3	<i>Construire des échelles de couleur (légendes)</i>	37
3.6.4	<i>Gestion de l'affichage</i>	37
3.6.4.1	Sélectionner un arrière plan	37
3.6.4.2	Modifier l'état d'une couche (affiché/non-affiché)	37
3.6.4.3	Modifier l'ordre des couches d'affichage.....	37
3.7	ACTIVITÉ : ANALYSE ET TRAITEMENT DES RÉSULTATS.....	37
3.7.1	<i>Sonde</i>	38
3.7.2	<i>Calculatrice</i>	38
3.7.3	<i>Profil en long</i>	38
3.7.4	<i>Projection</i>	38
3.7.5	<i>Outil de projection cartographique</i>	38
3.7.6	<i>Outil de ligne</i>	38
3.7.6.1	Étudier les lignes de courant.....	38
3.7.6.2	Trouver la trajectoire Lagrangienne.....	38
3.7.6.3	Mesurer des distances	38
3.7.7	<i>Outil de surface</i>	38
3.7.8	<i>Outil de calcul de débit</i>	38
3.8	ACTIVITÉ : CONSTRUCTION DE SÉRIES	38
3.8.1	<i>À définir</i>	38
3.9	ACTIVITÉ : PUBLICATION DE RÉSULTATS	38
3.9.1	<i>Impression</i>	38
3.9.2	<i>Rapports</i>	38
3.9.2.1	Gestion des rapports.....	38
3.9.2.2	Édition.....	39
3.9.3	<i>Habillage d'écran</i>	39
3.9.3.1	Gestion des habillages d'écran.....	39
3.9.3.2	Édition.....	39
3.9.4	<i>Animations</i>	39
3.9.4.1	Gestion des animations	39
3.9.4.2	Édition.....	39

3.10	ACTIVITÉ : GESTION DE MACROS	39
3.10.1	<i>Gestion des scripts globaux</i>	42
3.10.2	<i>Importer</i>	42
3.10.3	<i>Exporter</i>	42
3.10.4	<i>Rechercher</i>	42
3.10.5	<i>Modifier</i>	42
3.10.6	<i>Supprimer</i>	43
3.10.7	<i>Glisser-déposer et Raccourcis-clavier standards</i>	43
3.11	GESTION DES SCRIPTS LOCAUX	43
3.11.1	<i>Edition – Déboguage</i>	43
3.11.2	<i>Exécution</i>	43
3.11.3	<i>Raccourcis clavier</i>	43
3.11.4	<i>Macros</i>	44
3.11.4.1	Commencer l'enregistrement	44
3.11.4.2	Terminer l'enregistrement	44
3.11.4.3	Pause / reprise de l'enregistrement	44
3.12	IDE EXTERNE	44
3.12.1	<i>Editeur</i>	44
3.12.1.1	Caractéristiques nécessaires pour l'intégration dans Modeleur	44
3.12.1.2	Nouveau	44
3.12.1.3	Ouvrir	44
3.12.1.4	Editer	45
3.12.1.5	Enregistrer	45
3.12.1.6	Enregistrer sous	45
3.12.1.7	Fermer	45
3.12.1.8	Quitter	45
3.12.2	<i>Débogueur</i>	45
3.12.2.1	Caractéristiques nécessaires pour l'intégration dans Modeleur	45
3.12.2.2	Compiler	45
3.12.2.3	Exécuter pas à pas	45
3.12.2.4	Visualisation de l'état des variables	46
3.12.2.5	Exécuter	46
3.12.2.6	Affichage des erreurs	46
4	ARCHITECTURE LOGICIELLE	47
4.1	BUS D'ÉVÉNEMENTS	47
4.1.1	<i>Terminologie de base</i>	47
4.1.2	<i>Fonctionnement général</i>	48
4.1.2.1	La phase d'initialisation	48
4.1.2.2	La phase d'utilisation	48
4.1.2.3	La phase de fermeture	49
4.1.3	<i>Événements</i>	49
4.1.4	<i>Lien de type « direct »</i>	49
4.2	BASE DE DONNEES	50
4.3	SYSTEME DE VISUALISATION GRAPHIQUE	51
4.4	PLUG-IN	52
4.4.1	<i>Ajout d'un plug-in</i>	52

4.4.2	<i>Gestion des menus</i>	53
4.5	ENGIN DE SCRIPT.....	53
4.5.1	<i>Interface utilisateur (UI)</i>	53
4.5.2	<i>Calculatrice</i>	53
4.5.3	<i>Macros</i>	53
4.6	INSTALLATION VIA LE WEB	54
5	GUI	55
5.1	ORGANISATION DES FENÊTRES.....	55
5.2	MENU	55
6	AIDE, LANGUE ET DISTRIBUTION DU LOGICIEL	57
7	GLOSSAIRE	59

AVANT PROPOS

Ce document constitue une version préliminaire des spécifications de Modeleur 2.0. Le contenu et la structure de ce document sont sujets à des changements. Il est donc possible que certaines des fonctionnalités décrites dans ce document n'apparaissent pas dans la version finale.

Certaines sections ont été travaillées et révisées en détail. Dans ces sections, le texte apparaît en noir. D'autres sections sont beaucoup moins mûres, et le contenu devra être amélioré ou modifié. Dans ces sections, le texte apparaît en gris. À certains endroits, des remarques ont été insérées. Elles apparaissent en italique et sont encadrées.

Modeleur pourrait utiliser un modèle de données par copie ou par référence. La décision quant au mode à employer n'a pas encore été prise. C'est pourquoi dans le texte, il sera parfois question de ces modes. Les explications qui suivent aident à illustrer les avantages et les inconvénients de chacun des modes.

MODE REFERENCE

Lorsqu'une donnée A dépend d'une donnée B, elle prend une référence vers cette donnée pour l'utiliser. Un changement à la donnée B affectera la donnée A car elle détient une référence vers la donnée B.

Avantages : si l'on modifie une donnée au début de la chaîne de traitement, tous les maillons de la chaîne intégreront le changement. Ceci évite à l'utilisateur de répéter toutes les tâches intermédiaires. De plus, le mode par référence diminue grandement le nombre de données générées, ce qui facilite leur gestion.

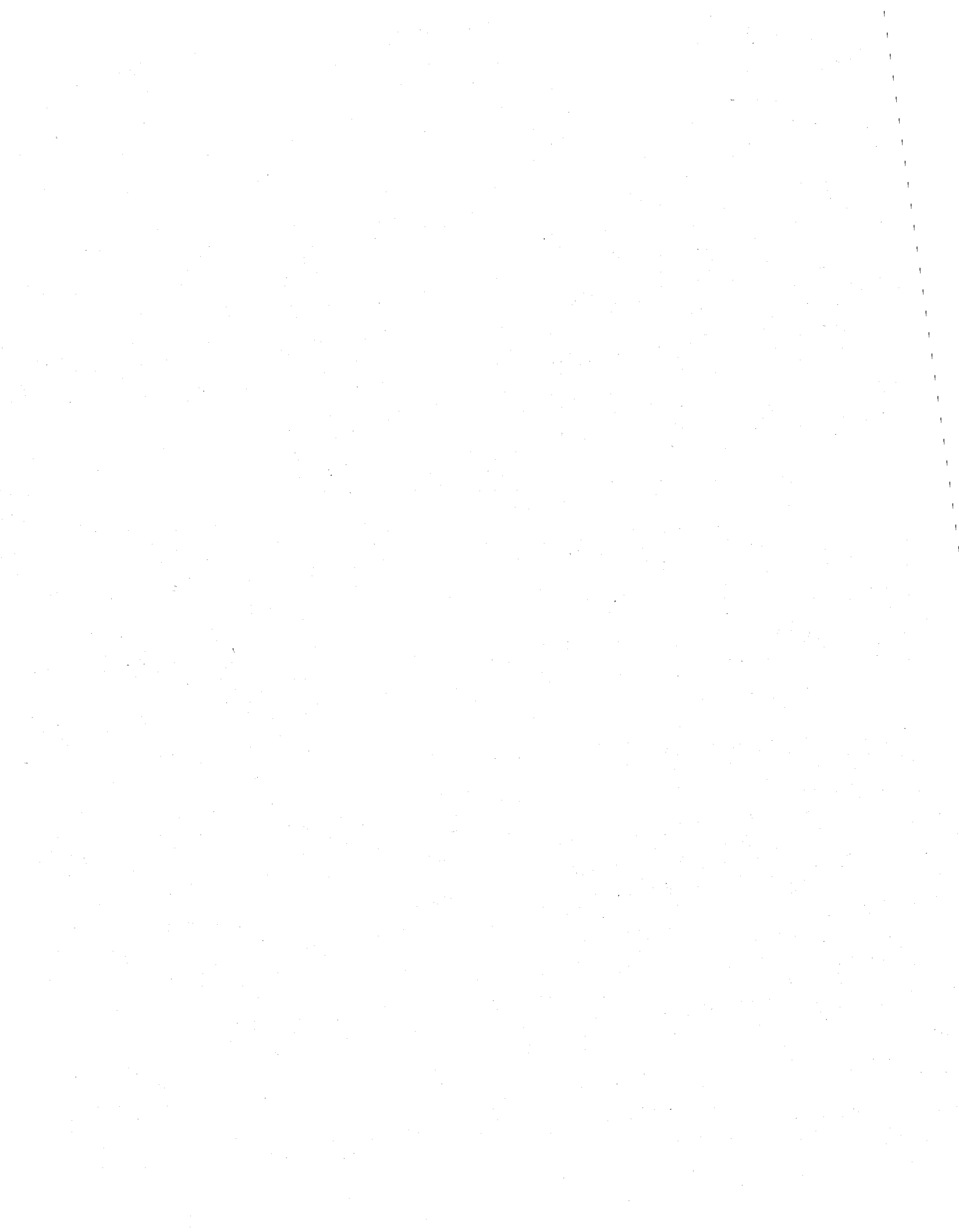
Désavantages : en modifiant une donnée, on peut introduire des effets néfastes sur toutes les données qui dépendent de cette donnée dans la chaîne de dépendance.

MODE COPIE

Lorsqu'une donnée A dépend d'une donnée B, elle fait une copie de cette donnée B pour l'utiliser. Un changement à la donnée originale n'affectera pas la donnée A.

Avantages : le lien direct entre les données est absent, ce qui évite le risque de modifier des données par erreur par l'intermédiaire d'un lien de dépendance.

Désavantages : le lien entre les données est absent, cela veut dire qu'un changement apporté à une donnée au début de la chaîne de traitement nécessite de ré-exécuter manuellement toutes les tâches intermédiaires. De plus, puisque le mode par copie génère une copie d'une donnée à chaque fois, la gestion des données devient difficile.



1. 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 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 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 : la 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 énorme (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 GIS permettant de faire des requêtes spatiales

- Les usagers ont demandé l'ajout de fonctionnalités de visualisation graphique avancées. Le module de visualisation de Modeleur n'offre pas d'extension facile.

Solution : l'utilisation d'un système de visualisation 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 Structure du document

Ce document contient cinq sections principales :

- description des concepts;
- description fonctionnelle;
- description technique;
- interface utilisateur;
- aide, langue et distribution du logiciel.

La section description des concepts discute des aspects suivants : les concepts clés à la source de Modeleur 2.0, l'arrivée d'une base de données relationnelle, les zones d'études, etc.

La section description fonctionnelle décrit les activités pour réaliser une étude. Chacune des activités est détaillée en explorant les fonctionnalités offertes par chacune d'elles.

La section description technique 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 interface utilisateur (UI) décrit les grandes lignes de l'interface graphique de Modeleur. Il est question de l'organisation des fenêtres et des menus, et de la façon d'interagir avec ceux-ci.

La section description de l'aspect gestion discute des sujets suivants: les langues, le support d'aide, la distribution et les logiciels externes à installer.

2 Description des concepts

2.1 Concepts clés à la base de Modeleur

Il sera question des concepts clés à partir desquels Modeleur 2.0 a été repensé.

2.1.1 Cohérence

Modeleur 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 important de renforcer au maximum la cohérence.

2.1.2 Souplesse

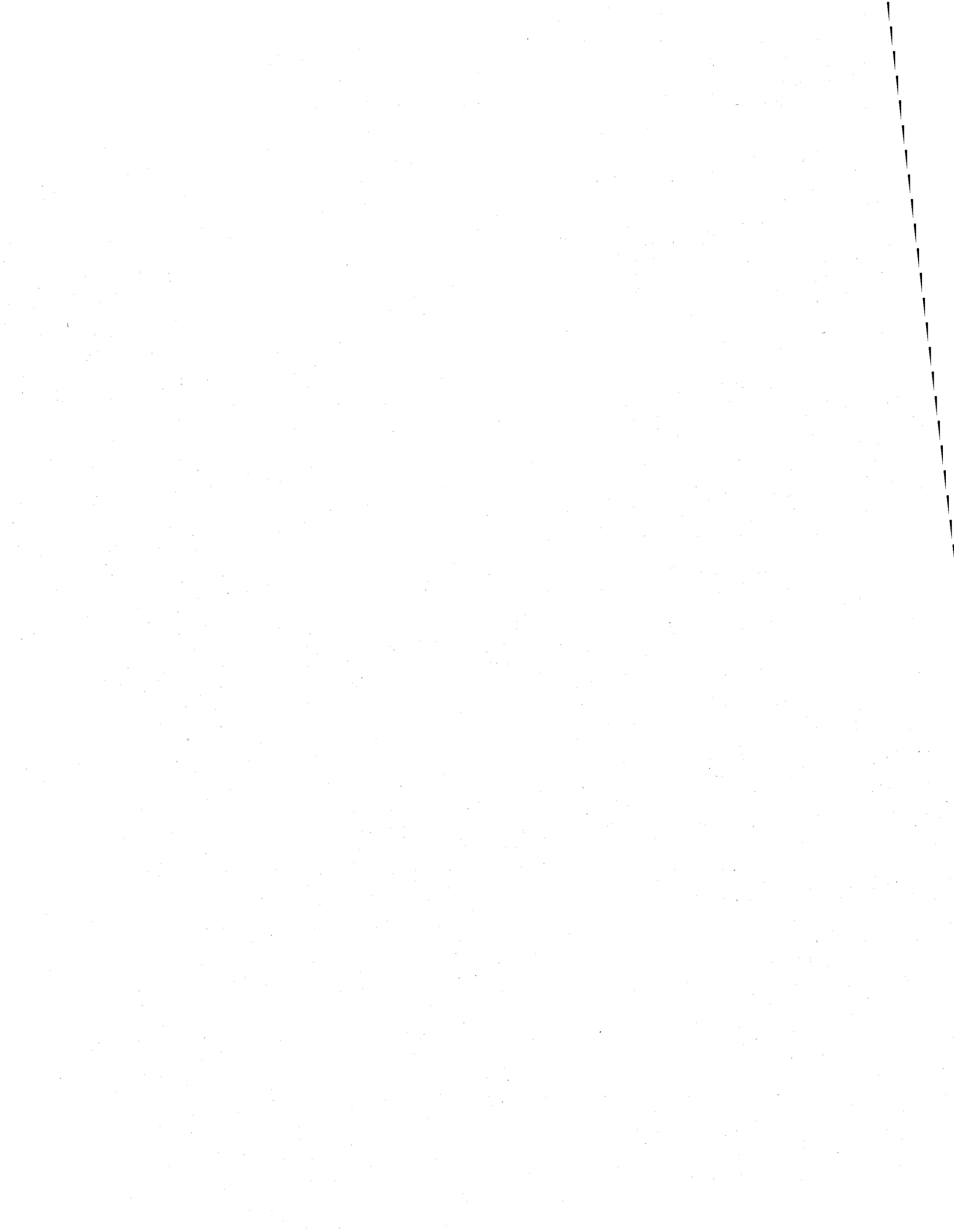
Modeleur a pour objectif de fournir aux usagers de la souplesse pour réaliser tout ce qu'ils veulent.

2.2 Base de données relationnelle

Modeleur aura besoin d'une base de données relationnelle pour fonctionner. Il devra également supporter la coexistence d'une base de données locale et d'une base de données globale partagée entre les utilisateurs de Modeleur.

2.3 Zones d'étude

L'utilisateur pourra déterminer une zone (ou région) d'intérêt sur laquelle il désire porter son étude.



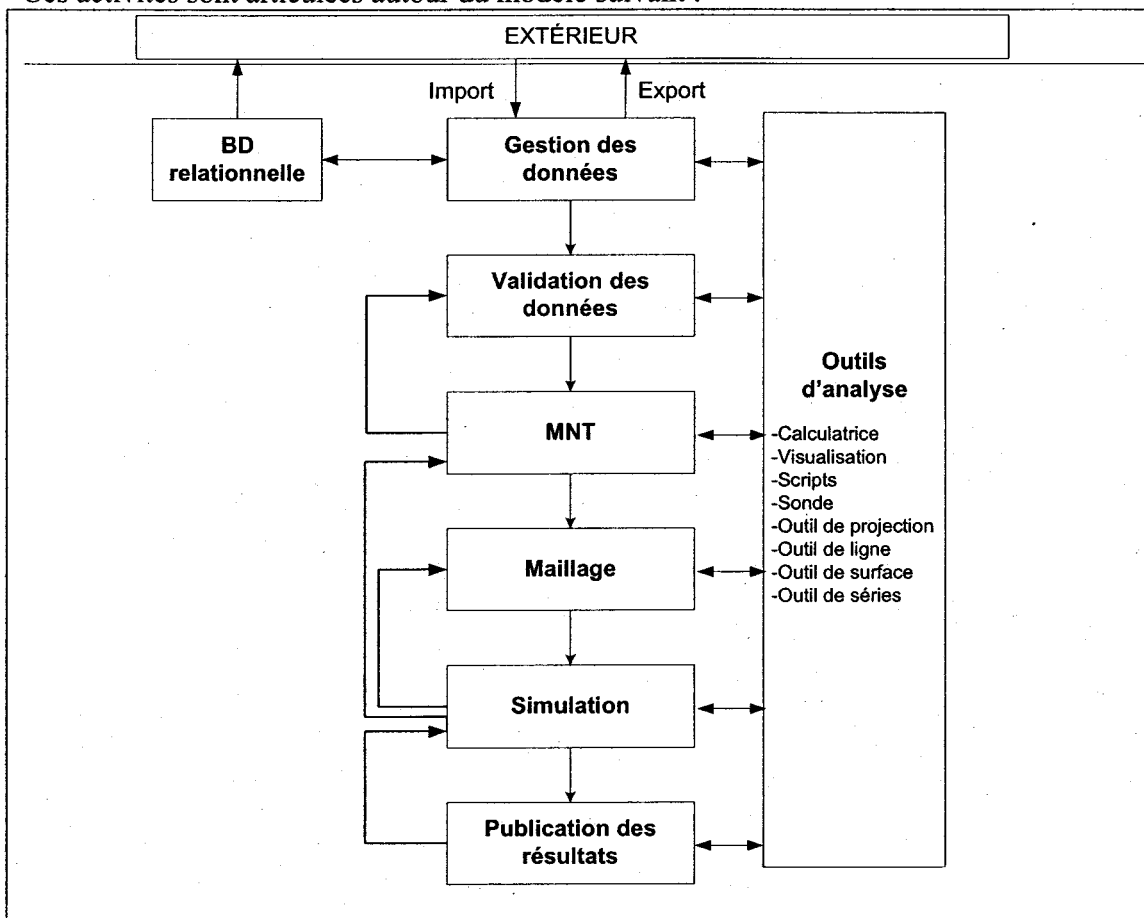
3 Description fonctionnelle

Modeleur 2.0 fonctionnera à l'aide d'activités liées typiquement aux tâches de modélisation. La plupart des tâches-clé de Modeleur pourront être regroupées sous l'une ou l'autre de ces activités. Ces activités suivent un flot particulier, mais qui implique parfois des retours en arrière. Une activité a pour but d'accomplir une certaine partie du travail de modélisation hydrodynamique.

Les activités que l'on retrouvera dans Modeleur 2.0 sont les suivantes :

- Gestion des données
- Validation des données
- Modélisation numérique de terrain (MNT)
- Maillage
- Simulation
- Publication des résultats

Ces activités sont articulées autour du modèle suivant :



Gravitant autour des activités qui viennent d'être mentionnées, un groupe d'outils qui peuvent être utilisés à tout moment pour interagir avec le système afin d'y effectuer des tâches d'analyse. Ces outils sont :

- les scripts;
- la calculatrice;
- la visualisation des données;
- la sonde;
- l'outil de ligne;
- l'outil de surface;
- la projection;
- le calcul de débit.

Une activité découpe un groupe de fonctionnalités, relativement distinctes. L'extrait du travail fait dans une activité peut être réutilisé dans une autre activité, si au préalable, il a été sauvegardé.

3.1 Activité : Gestion des données

3.1.1 Gestion de projets

3.1.1.1 Nouveau

Basé sur un projet dans la BD locale

Le projet peut être créé à partir d'un projet déjà présent dans la base de donnée locale.

MODE REFERENCE: Une référence est faite à l'autre projet. Les entités référencées ne peuvent pas être effacées. Elle sont uniquement disponible en mode lecture.

MODE COPIE: Une copie de tous les composants du projet source est faite. L'utilisateur peut donc modifier ses composants sans altérer le projet source.

Basé sur un projet dans la BD externe

L'utilisateur peut créer un projet à partir d'une base de données externe. Celle-ci doit avoir été configurée au préalable.

3.1.1.2 Ouvrir

L'utilisateur peut ouvrir un projet existant.

3.1.1.3 Fermer

L'utilisateur peut fermer un projet. L'utilisateur se verra demander s'il désire sauvegarder les données qui ne l'ont pas été.

3.1.2 Importer des données

3.1.2.1 À partir de fichiers

L'utilisateur peut effectuer des opérations d'importation à partir de fichiers. L'utilisateur peut importer plusieurs fichiers simultanément.

3.1.2.2 À partir d'une base de données externe

3.1.3 Exporter des résultats

3.1.3.1 Vers une base de données externe

3.1.3.2 Vers des fichiers

3.1.4 Voir le nom des données d'un projet

L'utilisateur peut voir via une arborescence les données d'un projet (incluant les résultats de calculs). L'arborescence affiche le nom des données. Cette vue peut être soumise à un filtre ou à une requête faite par l'utilisateur.

3.1.5 Effacer une donnée de projet

L'utilisateur peut effacer une donnée s'il n'y a aucune autre donnée qui dépend d'elle.

3.1.6 Voir le graphe des dépendances d'une donnée

L'utilisateur peut voir quelles sont les données qui dépendent d'une donnée particulière.

3.1.7 Voir les informations sur une donnée

Les informations sur les données d'un projet peuvent être vues et certaines peuvent être modifiées.

3.1.8 Visualiser les données (vue tabulée)

Une donnée de projet (incluant les résultats de calculs) peut être visualisée sous forme tabulée (style Excel).

3.1.9 Trier les données

À l'aide de requêtes SQL, les données peuvent être triées et filtrées selon différents critères.

3.2 *Activité : Validation des données de terrain*

Cette activité a pour but de valider les données de terrain qui ont été importées à l'Activité : Gestion des données.

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 Modeleur, notamment des données météo (vent, glace, précipitation, couverture nuageuse, etc.), des données de plantes ou d'utilisation du sol.

L'utilisateur aura à interagir 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 décrivent 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 possède comme attributs, sept valeurs numériques (répartition des catégories granulométriques en %).

Quand un point de terrain est importé, on importe du même coup toutes les valeurs de cette donnée $\{x, y\}$, \langle 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 accomplir est de désactiver certains points aberrant qui ne sont pas cohérents dans son système de validation. L'utilisateur peut faire appel à des filtres pour l'aider à déterminer quels doivent être les points à désactiver.

Les points désactivés ne sont pas effacés (perdus); l'utilisateur peut donc en tout temps revenir aux données qui ont été mesurées. 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 2.3), c'est-à-dire qu'il peut indiquer une région de l'espace délimitant le secteur de données avec lequel il désire travailler. Cette fonctionnalité est disponible car il arrivera très souvent que l'utilisateur ne sera pas intéressé à travailler sur son jeu de données en entier.

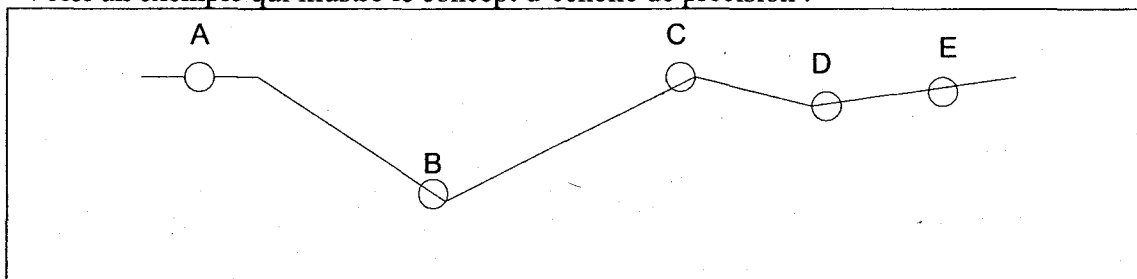
Les données recueillies peuvent varier dans le temps. C'est-à-dire qu'une donnée peut avoir été mesurée au même endroit {x,y}, mais à différents moments (différentes campagnes de terrain). Lors de l'importation de données réparties dans le temps, les points de chaque pas de temps sont importés. Ceci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

L'utilisateur doit spécifier quel est 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 à chacun des points de son semi. Cette échelle pourra être déterminée de plusieurs façons; par exemple, en fonction des points voisins. Le but de cette opération est de permettre aux usagers de sélectionner des points servant à représenter une certaine précision.

L'échelle de précision est une valeur qui qualifie le niveau de précision pour lequel le point peut être ignoré. L'échelle de précision sert à enlever 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 :



Si le point D est enlevé, il y a une perte de précision moindre que si 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.

- **MODE REFERENCE:** Les usagers utilisent directement les semis de points par référence à l'étape du MNT (voir activité 3.3). Ainsi, un changement fait dans un semi de points aura un impact sur les MNT auxquels il est relié.
- **MODE COPIE:** Une fois nettoyé, le semi de points de terrain pourra être généré pour devenir une composante de terrain. L'utilisateur pourra utiliser la composante de terrain ainsi générée pour monter son MNT (Modèle Numérique de Terrain) (voir activité 3.3). Une composante de terrain qui a été générée ne peut être modifiée par la suite.

Au moment de la génération de la composante de terrain, une échelle de précision est attribuée à chaque point.

3.2.1 Gestion

3.2.1.1 Nouveau

L'utilisateur a le loisir de construire un semi de points de terrain, à partir de zéro, c'est-à-dire en y spécifiant tous les points (ex : cas académique). L'utilisateur doit spécifier dans quel repère cartographique se situera 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.).

3.2.1.2 Ouvrir

L'utilisateur peut ouvrir un semi de points de terrain existant. En fonction du type de donnée à la source de ce semi de points de terrain, le type d'affichage sera différent. L'utilisateur peut spécifier une zone d'étude (voir 2.3).

3.2.1.3 Fermer

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

3.2.1.4 Enregistrer

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

3.2.1.5 Enregistrer sous

Le semi de points de terrain est sauvegardé 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).

3.2.2 Points

3.2.2.1 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 qui sont sélectionnés endossent une couleur différente de ceux qui ne le sont pas.

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.
- il peut également sélectionner plusieurs points en maintenant la touche CTRL enfoncée, et en cliquant sur les points à sélectionner.

Par zone

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 la zone.

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 retrouvent 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. L'utilisateur a alors la souplesse de contourner les points qu'il désire. Cette zone de sélection peut également être déplacée et éditée.

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

L'utilisateur peut écrire un script lui permettant de sélectionner des points.

3.2.2.2 Créer des points

L'utilisateur peut créer des points manuellement à l'aide de la souris. L'utilisateur peut également ajouter un ou plusieurs points à son semi de points de terrain, en spécifiant la coordonnée exacte où le point doit être ajouté.

Cette opération est également accessible via un script, ce qui permet à l'utilisateur de modifier son terrain automatiquement, ou même d'en créer un de toute pièce, sans avoir à manipuler l'interface graphique.

3.2.2.3 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 par la composante de terrain qui sera générée par la suite 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 garder que les points qui sont essentiels dans le cadre de son problème.

3.2.2.4 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.

3.2.2.5 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.

3.2.2.6 Importer des points

L'utilisateur peut importer une zone de points provenant d'un ou plusieurs autres jeux de données. L'utilisateur peut spécifier un sous-domaine des données qu'il désire intégrer dans son semi de points de terrain. Le sous-domaine pourra être spécifié à l'aide de l'outil de spécification des sous-domaines. Cette opération pourra être utilisée pour générer des semis de points combinés.

- **MODE REFERENCE:** Les points qui seront modifiés ou désactivés dans le semi de points combiné seront également modifiés ou désactivés dans le semi de points duquel ils originent.
- **MODE COPIE:** Les points qui seront modifiés ou désactivés dans le semi de points combiné ne le seront pas dans le semi de points duquel ils originent.

3.2.3 Information portée par les points

3.2.3.1 Editer l'information

L'utilisateur peut modifier les attributs des points du semi. Le format dans lequel les attributs sont présentés dépend du type des attributs. Les attributs 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 faits, il peut toujours remettre le point à sa valeur d'origine (valeur mesurée).

3.2.3.2 Filtrer l'information

L'utilisateur peut faire appel à des filtres afin de déterminer les points qui doivent être désactivés. Les filtres sont des algorithmes appelables à partir d'un script et qui répondent à un API particulier.

Algorithmes de filtrage

L'API que le script doit respecter est le suivant :

```
Filtre(Point[] in, Cote [] out)
```

En entrée, le script se voit passer la liste des points du semi de points courant (en fonction de la zone d'étude active s'il y a lieu), un point contenant une coordonnée {x,y} et des attributs. Les attributs peuvent être de différents types. En sortie, le filtre fournit une cote entre 0 (mauvais) et 1 (excellent) pour chacun des points. Cette cote sera utilisée pour déterminer si le point doit être désactivé. Par exemple, le calcul de la cote du point pourrait être basé sur l'écart de valeur de ce point par rapport aux points voisins.

Outil de filtrage

L'utilisateur peut faire appel à un outil pour enchaîner une suite de filtres pour aider à trouver les points qui doivent être désactivés dans un semi de points (Figure 3-1). L'utilisateur peut effectuer les actions suivantes sur un modèle de filtrage : créer, charger, sauvegarder, sauvegarder sous.

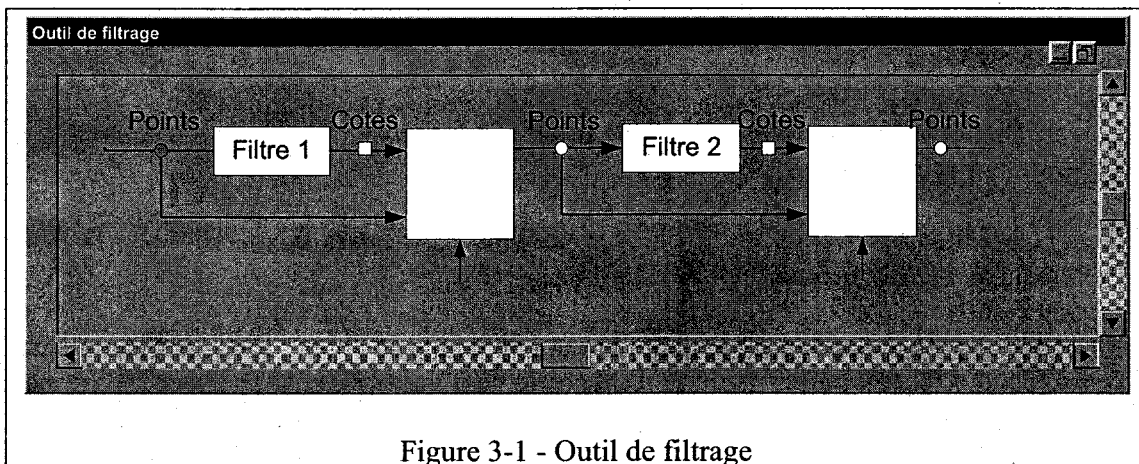
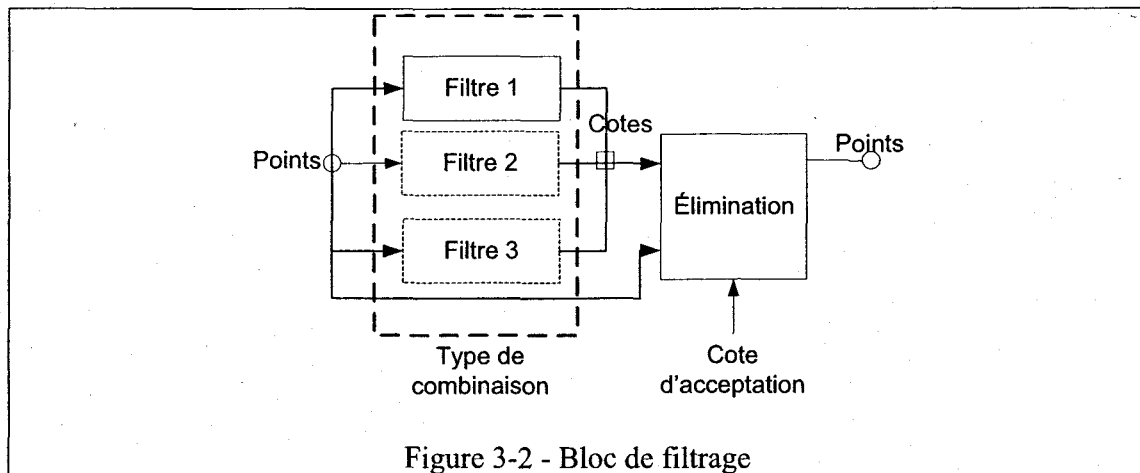


Figure 3-1 - Outil de filtrage

Cet outil fonctionne en mettant en série une suite de blocs de filtrage. Un bloc de filtrage est constitué d'un semi de points d'entrée, d'un semi de points de sortie, d'un ou plusieurs filtres en parallèle et d'un bloc d'élimination caractérisé par une cote d'acceptation, tel qu'illustré dans la Figure 3-2. La cote d'acceptation sert à déterminer si un point doit rester actif, ou être désactivé.



L'utilisateur peut placer plus d'un filtre dans un bloc de filtrage. Lorsque c'est le cas, il doit alors spécifier le type de combinaison qui sera utilisée pour produire la cote de sortie pour ses filtres. Par exemple, la combinaison pourrait être faite à partir de la moyenne des cotes produites par tous les filtres ou encore en donnant un poids prédominant à l'un des filtres.

Lors de l'ajout ou de l'insertion d'un bloc filtre, par défaut, le nombre de filtres est fixé à 1 et la cote d'acceptation est fixée à 0.5

L'utilisateur peut ajouter, insérer et enlever un bloc de filtrage à son modèle. Il peut également activer/désactiver un bloc de filtrage.

Le semi de points en entrée est toujours le semi de points en cours. L'utilisateur peut mettre des points d'arrêt aux points marqués d'un cercle ou d'un carré, pour analyser les cotes et l'état du semi de points à cette étape du processus. Cette fonctionnalité est intéressante car elle permet de prendre une décision sur la cote d'acceptation à fixer pour chaque bloc de filtrage. L'utilisateur peut également décider d'activer/désactiver un bloc de filtrage en fonction de ce qui a été analysé.

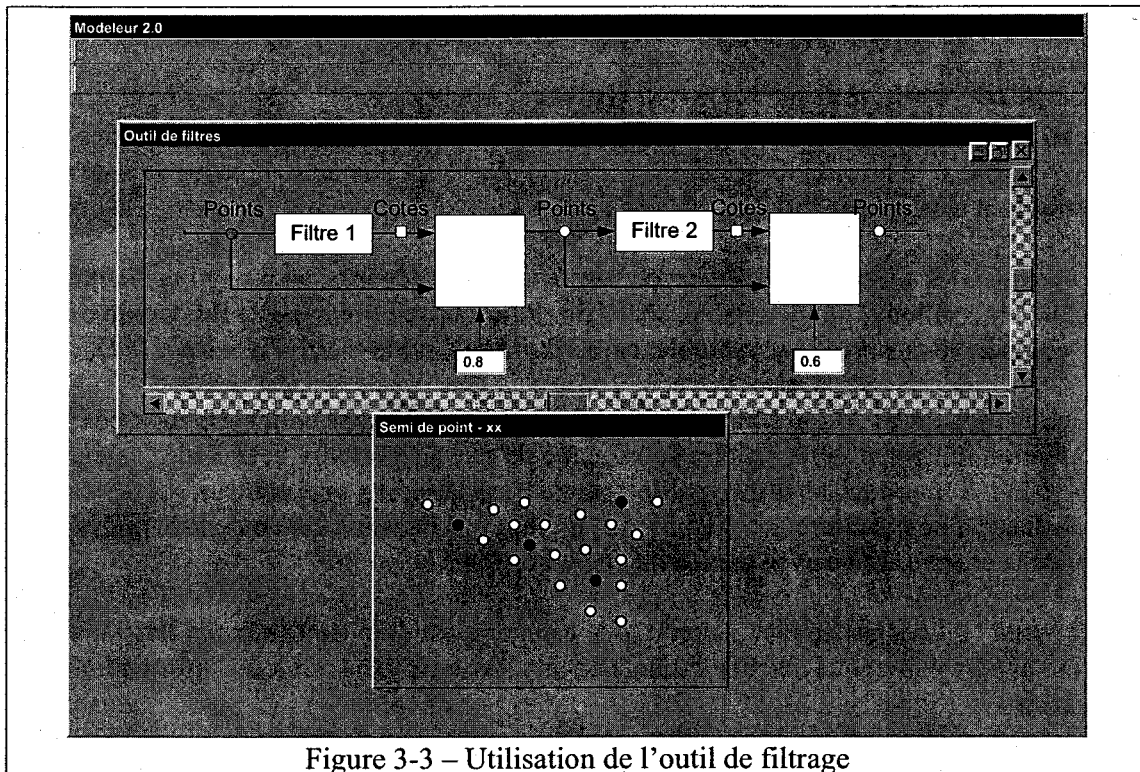


Figure 3-3 – Utilisation de l’outil de filtrage

Les points qui ont été désactivés dans le semi de point présent au début du modèle de filtrage sont marqués d’une couleur ou d’un symbole.

Aux points marqués d’un carré, l’usager peut obtenir les isolignes ou les isosurfaces de cotes pour les points de son domaine. Ceci pourra servir si l’usager désire déterminer comment la cote est répartie sur son domaine. Il pourra ainsi identifier les zones où la cote est basse et confirmer qu’elle correspondent effectivement aux zones problématiques de son domaine.

Aux points marqués d’un carré, l’usager peut avoir accès à un histogramme de répartition. Cet histogramme sert à illustrer la répartition des cotes entre 0 et 1. Cette information pourra aider l’usager à déterminer la cote d’acceptation.

L’usager peut également voir les données (cotes ou semis) sous forme tabulée.

3.2.3.3 *Lisser l’information de jeux de données limitrophes*

L’usager peut effectuer une opération d’interpolation entre deux groupes de points. Cette opération pourra être utilisée pour ajuster les points de deux jeux de données qui sont limitrophes. Le but de la manœuvre est de rendre la zone de transition entre les deux jeux de données plus fluide. L’usager peut spécifier l’algorithme d’interpolation utilisé.

3.2.3.4 **Afficher l'information portée par les points de terrain**

Affichage 2D – catégories avec vue non numérique

Cet affichage est utilisé pour représenter les données qui réfèrent à des catégories non numériques et à leur répartition sur chaque point $\{x, y\}$ du domaine. Les données de type substrat font partie de ce groupe.

Par exemple, les substrats peuvent se classer dans sept catégories ou plus : bloc métrique, bloc, galets, cailloux, gravier, sable, sable fin. Pour chaque point $\{x,y\}$ du domaine, une répartition a été mesurée (par exemple, cailloux = 80%, sable = 20%, le reste = 0%).

Chaque point $\{x,y\}$ d'un semi de points de terrain de type catégorie possède une information indiquant la répartition de chacune de ses catégories. Chacune des catégories correspond à une couche d'information auquel est associée une couleur. Ces couches d'information peuvent être activées ou désactivées par l'utilisateur. La couleur du point sera celle de la couche active avec la plus haute répartition à ce point.

Ceci permet à l'utilisateur de bien situer quelle catégorie est dominante sur son territoire. En jouant avec l'état (actif/inactif) d'une couche, l'utilisateur peut cibler l'information qui l'intéresse.

Affichage 2D – catégories avec vue numérique

Cet affichage est utilisé pour représenter certains aspects des données qui réfèrent à la valeur numérique d'une catégorie. Par exemple, les valeurs numériques des données de type substrats pourraient être : diamètre d'un bloc métrique = 10 m, diamètre d'un bloc = 1.0 m, diamètre d'un galet = 0.25 m, etc. Pour chaque point $\{x,y\}$ du domaine, une répartition a été mesurée (par exemple, cailloux = 80%, sable = 20%, le reste = 0%).

Avec cette information, il est possible de déterminer le diamètre moyen, le diamètre minimum/maximum de chacun des points. Une fois l'information recueillie, il est possible de reporter le tout sur une échelle de couleur pour représenter l'information sur le domaine entier.

Affichage 2D – avec vecteurs

L'affichage 2D avec vecteur pourrait permettre de visualiser les données de terrain de type vectoriel, comme le vent, la vitesse du courant ou les contraintes de frottement.

Affichage 3D

L'affichage 3D pour les données est utilisé pour représenter les données topographiques, mais pourrait être utilisé pour toute donnée qui s'exprime avec un scalaire. Avec cette vue, l'utilisateur pourra visiter son domaine, et plus facilement déceler les anomalies.

3.2.3.5 Voir les données dans une table

L'utilisateur a accès à une vue des données (ou une partie des données) sous forme tabulaire. Il peut ainsi voir les coordonnées {x,y} et l'information. Le tableur associé est alors démarré (ex : Microsoft Excel).

3.2.4 Gestion du temps

3.2.4.1 Sélectionner le pas de temps actif

Pour les données recueillies à différent pas de temps, l'utilisateur peut sélectionner quel pas de temps est le pas de temps actif. Le semi de points de terrain à ce pas de temps est alors affiché, et l'utilisateur peut interagir avec lui.

3.2.4.2 Comparer les données de différents pas de temps

L'utilisateur peut comparer ses données en ouvrant un semi de points de terrain à un temps 't1', et le même semi de points de terrain à un temps 't2', dans des fenêtres séparées.

3.2.4.3 Enchaîner les données sous forme d'animation

L'utilisateur peut afficher les pas de temps d'un même semi de points, à tour de rôle. En d'autres mots, il voit l'état de son semi de points à chacun des pas de temps, ce qui s'articule sous forme d'une animation. L'utilisateur peut fixer l'intervalle de temps entre chacune des « images » de l'animation. Ceci permet à l'utilisateur de visualiser les variations dans ses données à travers le temps.

Il est important de voir que lorsque des jeux de données de grande taille sont manipulés, l'utilisateur n'aura d'autre choix que de spécifier une zone d'étude ou utiliser l'échelle de précision pour ne garder que les points d'une certaine précision, s'il veut être en mesure d'utiliser la fonctionnalité « animation ».

3.3 Activité : MNT

L'extrait de cette étape 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. Le nombre est indéterminé a priori car c'est à l'utilisateur de déterminer de quoi sera construit son MNT.

Chaque partition décrit un aspect particulier du terrain (exemple : topographie, substrat, etc.). Il existe des partitions spécialisées et des partitions génériques, lesquelles seront décrites dans les pages subséquentes.

Un MNT peut contenir les types de partitions spécialisées suivants: topographie, substrat, glace, vent, frottement de Manning. Il est important de voir que la liste pourrait s'allonger au cours du projet.

Les partitions spécialisées sont construites en superposant des couches de données. Une couche de données est construite de sous-domaines auxquels est associé un même jeu de données. Les couches de données se voient donner des ordres de priorité. La couche de données se retrouvant à l'avant-plan est celle qui sera utilisée.

La zone commune à toutes les partitions d'un MNT est appelée « squelette du MNT ». 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 du MNT.

Il est possible de créer une couche de données qui couvre tout le domaine d'intérêt, à laquelle on associe la valeur « 0 ». La valeur « 0 » signifie qu'il n'y a pas de données, mais que le « squelette du MNT » ne doit pas considérer cela comme un trou.

3.3.1 Interface usager

À la création ou à l'ouverture du MNT, l'utilisateur se voit présenter une fenêtre contenant le gestionnaire de partitions avec une arborescence des partitions contenues dans ce MNT. On peut y voir toutes les partitions du MNT, et ainsi en ajouter, en enlever, en enregistrer.

Il y a également une vue avec onglets (voir Figure 3-4) qui permet de visiter chacune des partitions du MNT. L'utilisateur peut donc modifier chacune de ses partitions et naviguer d'une partition à l'autre. Quand il change d'onglet, les changements faits à la partition en cours sont gardés en mémoire, mais ne sont pas sauvegardés.

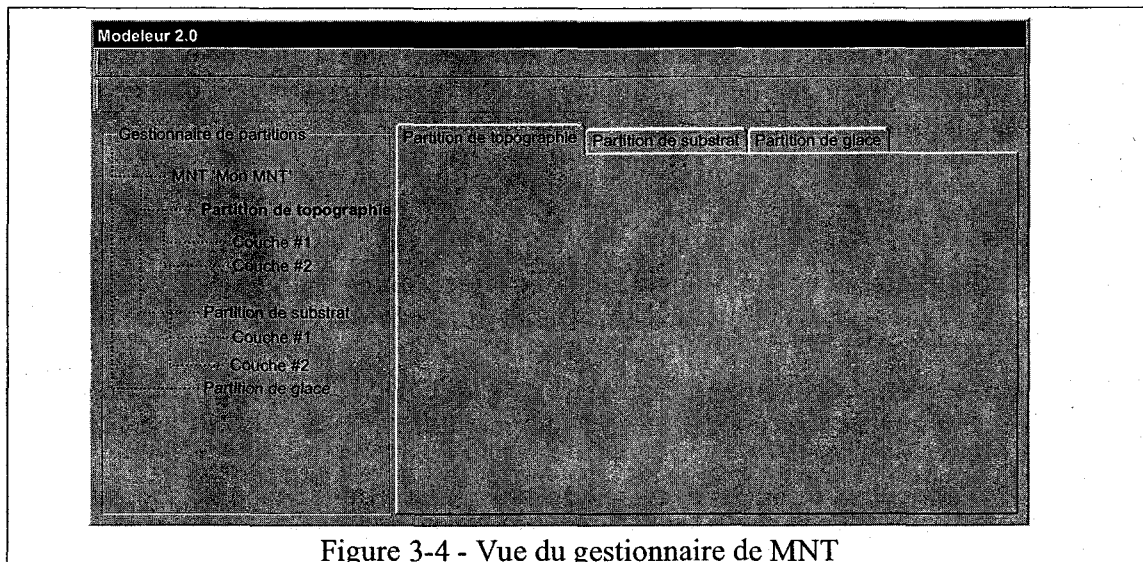


Figure 3-4 - Vue du gestionnaire de MNT

Remarque : L'utilisateur ne peut manipuler qu'un seul MNT à la fois. Lorsqu'il désire en ouvrir ou en créer un, il se voit demander de fermer le MNT en cours s'il y a lieu. Ce n'est pas certain que c'est la bonne solution, car il pourrait être intéressant d'avoir deux MNT et de pouvoir faire du copier / coller de partitions ou de couches.

3.3.2 Gestion d'un MNT

- **MODE REFERENCE:** Une même partition peut être liée à plus d'un MNT. Bien qu'une partition doit être créée dans le cadre d'un MNT, celle-ci peut continuer d'exister même si elle n'est reliée à aucun MNT.
- **MODE COPIE:** Une partition est obligatoirement liée à un MNT. Toutefois, l'utilisateur peut importer une partitions d'un autre MNT, ce qui a pour effet de copier la partition en question dans le MNT en cours. Aucun lien n'est gardé vers le MNT ou la partition source. Ceci fait que tout changement demeurera local au MNT en cours.

3.3.2.1 Nouveau

MNT vide

Il y a création d'un MNT vide. Tout ce que l'utilisateur peut faire à cette étape, c'est d'ajouter des partitions à son MNT.

MNT existant

- **MODE REFERENCE:** Il y a création d'un MNT qui fait référence aux mêmes partitions que le MNT existant. Les modifications faites sur ces partitions auront donc effet sur le MNT existant ou tout autre MNT faisant référence à ces partitions.
- **MODE COPIE:** Il y a création d'une copie d'un MNT existant et de ses partitions. Les modifications faites sur le nouvel MNT n'altéreront donc pas le MNT existant ou tout autre MNT.

3.3.2.2 Ouvrir

Il y a ouverture d'un MNT existant. Toutes les partitions contenues dans ce MNT sont également ouvertes. L'utilisateur peut donc naviguer d'une partition à l'autre en changeant d'onglet ou à l'aide du gestionnaire de partitions.

3.3.2.3 Fermer

Il y a fermeture du MNT courant. Au préalable, l'utilisateur se voit demander s'il désire enregistrer son MNT.

3.3.2.4 Enregistrer

Il y a sauvegarde du MNT en cours. Ceci a pour effet d'enregistrer toutes les partitions contenues dans ce MNT. À partir de toutes les partitions dans le MNT, il y a génération du « squelette du MNT ».

- **MODE REFERENCE:** Tous les MNT qui utilisent les partitions dans le MNT voient le changement les affecter car ils font référence aux mêmes partitions. L'utilisateur reçoit une mise en garde qui indique quels sont les MNT qui seront touchés.
- **MODE COPIE:** Le changement reste local au MNT en cours.

3.3.2.5 *Enregistrer sous*

Il y a sauvegarde du MNT en cours sous le nom spécifié par l'utilisateur. À partir de toutes les partitions dans le MNT, il y a génération du « squelette du MNT ».

- **MODE REFERENCE:** Les partitions contenues dans le MNT en cours sont sauvegardées. Tous les MNT qui utilisent ces partitions voient le changement les affecter. L'utilisateur reçoit une mise en garde qui indique quels sont les MNT qui seront touchés.
- **MODE COPIE:** Les partitions contenues dans le MNT en cours sont copiées avec le nouvel MNT. Le changement reste local et aucun autre MNT n'est affecté.

3.3.3 **Gestion des partitions d'un MNT**

3.3.3.1 *Nouvelle partition*

Lorsque l'utilisateur a créé ou a 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. L'ajout d'une partition a pour effet d'insérer un nouvel onglet.

3.3.3.2 *Ajouter une partition existante*

L'utilisateur peut choisir d'ajouter une partition existante à son MNT. L'utilisateur peut alors visualiser la partition qui a été chargée et la modifier en la sélectionnant.

- **MODE REFERENCE:** La partition ajoutée au MNT fait référence à une partition existante. Tout changement à cette partition sera répercuté sur tous les MNT utilisant cette partition.
- **MODE COPIE:** La partition ajoutée au MNT est une copie d'une partition existante. Tout changement restera local au MNT courant.

3.3.3.3 *Enregistrer*

L'utilisateur a la possibilité d'enregistrer une partition. Il est important de voir que c'est uniquement la partition qui est sauvegardée. Elle ne garde pas de liens avec le MNT courant.

3.3.3.4 Enregistrer sous

L'utilisateur a la possibilité d'enregistrer une partition sous un nouveau nom. Elle ne garde pas de liens avec le MNT courant.

3.3.3.5 Retirer

L'utilisateur peut retirer une partition du MNT. Ceci aura pour effet d'enlever le lien entre la partition et ce MNT. Après ce type d'opération, il est possible que la partition ne soit liée à aucun MNT. L'utilisateur se verra offrir le choix d'effacer de façon permanente la partition.

3.3.4 Partitions spécialisées

La gestion des partitions spécialisées est basée sur un principe de superposition de couches de données en fonction de leur priorité. Les jeux de données associés aux couches qui se retrouvent à l'avant plan seront ceux qui seront utilisés lorsque l'information sera portée sur un maillage.

L'action d'effectuer un double-clic sur une couche ou une partition, a pour effet de changer le contenu affiché dans la fenêtre « vue » (Figure 3-5).

Si le double-clic a eu lieu sur une couche, la fenêtre « vue » affiche une vue sur cette couche, permettant de la modifier.

Si le double-clic a eu lieu sur une partition, la fenêtre « vue » affiche 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, ajouter une nouvelle couche ou modifier une couche existante.

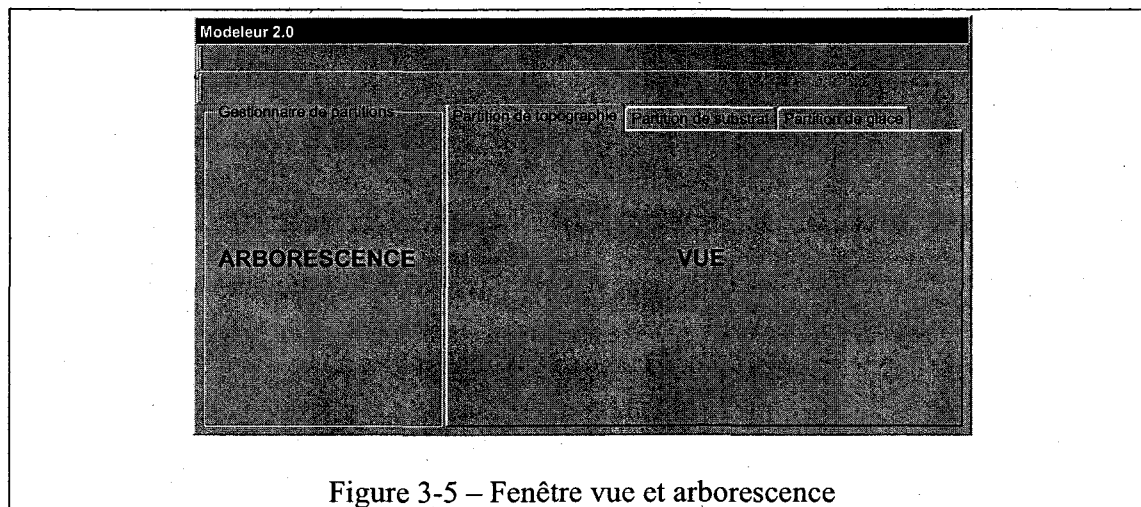


Figure 3-5 – Fenêtre vue et arborescence

L'utilisateur accorde une priorité aux couches de données en les déplaçant dans une arborescence. Par exemple, dans la première partie de la Figure 3-6, la couche A est prioritaire alors que dans la seconde partie de la couche B est prioritaire.

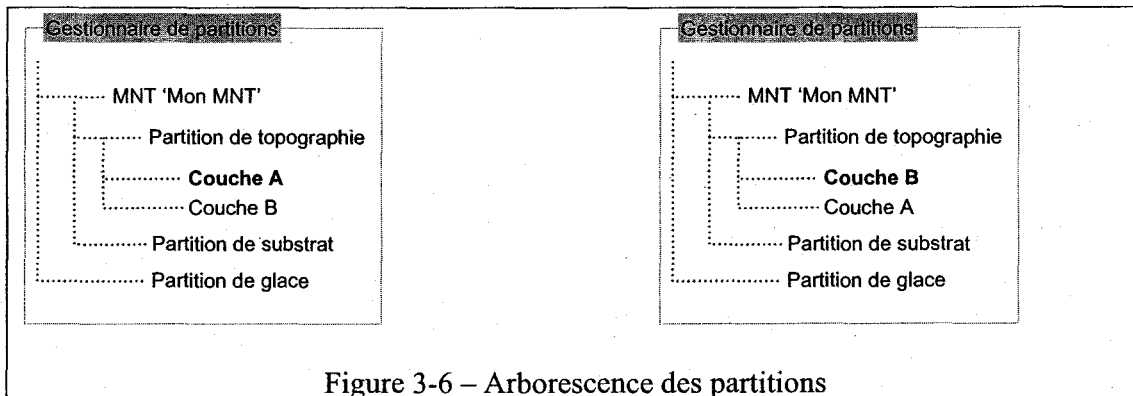


Figure 3-6 – Arborescence des partitions

Une couche de données est constituée d'un ou plusieurs sous-domaines qui se voient assignés un jeu de données commun. Le ou les sous-domaines d'une couches doivent obligatoirement se trouver dans la limite du jeu de données.

Dans la Figure 3-7, la couche A est associée au jeu de données J1. La couche B est associée au jeu de données J2. Ces deux couches ne sont constituées que d'un seul sous-domaine, qui dans ce cas particulier, correspond à la limite de validité des jeux de donnée.

La zone hachurée représente la zone couverte par chacune des couches de données. Les lignes pointillées représentent les limites de validité des jeux de données.

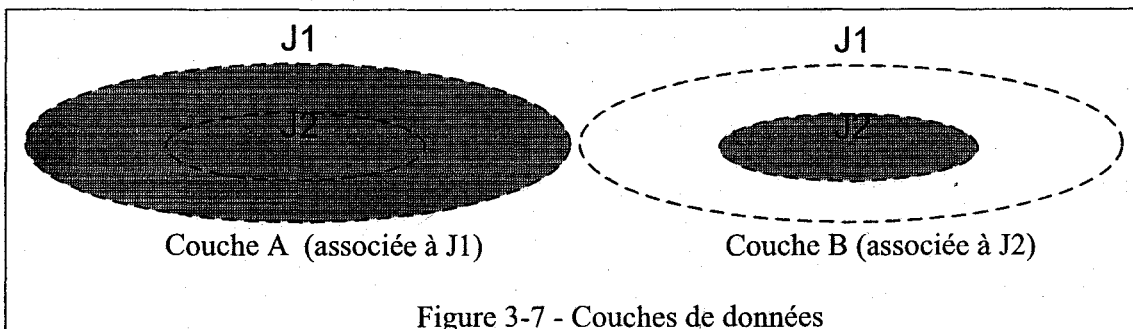
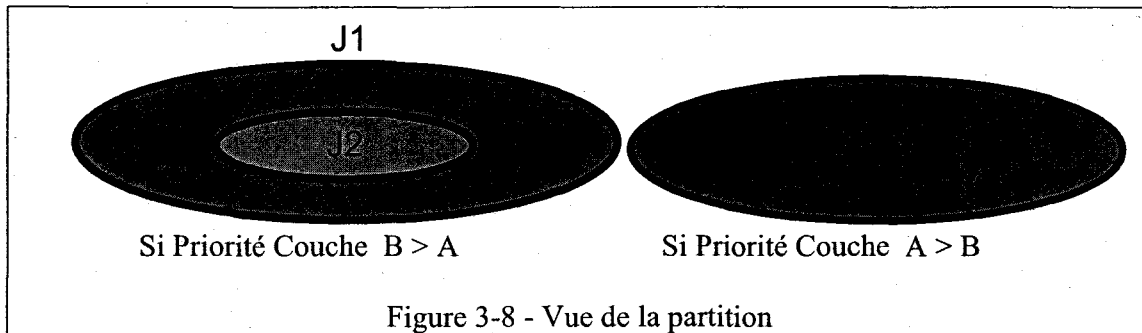
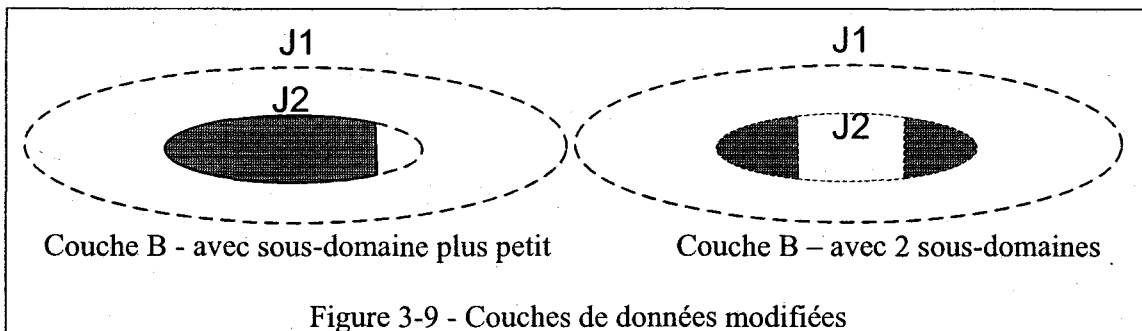


Figure 3-7 - Couches de données

La Figure 3-8 illustre les vues effectives des partitions que l'on obtiendrait en fonction des priorités assignées aux couches.



Dans l'exemple précédent, on constate qu'il n'y a qu'un seul sous-domaine et que celui-ci occupe toute la zone de validité du jeu de données. Il est important de voir que le sous-domaine aurait pu être plus petit ou qu'il aurait pu y avoir plus d'un sous-domaine tel qu'illustré dans la Figure 3-9.



3.3.4.1 *Afficher les limites externes de jeux de données (peaux)*

L'utilisateur peut afficher une ou plusieurs peaux de jeux de données. L'utilisateur ne pourra toutefois pas interagir avec cette couche d'information. Elle est présente uniquement à titre informatif et l'utilisateur pourra facilement la différencier de la couche active.

3.3.4.2 *Ajouter une couche de données*

L'utilisateur peut ajouter une couche de données à sa partition. L'utilisateur doit sélectionner quel jeu de données doit être assigné à cette couche. Après son ajout, la couche devient la couche active. C'est-à-dire que l'utilisateur peut l'éditer en créant un ou plusieurs sous-domaines. La couche se voit attribuer par défaut une priorité maximum.

3.3.4.3 *Éditer une couche de données*

L'utilisateur peut choisir entre deux modes d'édition des sous-domaines d'une couche de données: l'édition avec peau ou l'édition libre. Dans chacun des deux modes, l'utilisateur ne peut dépasser les limites couvertes par le jeu de données. 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'édition avec peau

L'outil affiche une ligne continue délimitant le contour du jeu de données, ligne à partir de laquelle l'utilisateur peut construire son sous-domaine. Il peut l'ajuster à sa guise à l'aide de l'éditeur de sous-domaines.

L'édition libre

L'utilisateur construit lui-même un polygone en reliant des segments de droites entre eux à l'aide de l'éditeur de sous-domaines.

3.3.4.4 Copier et coller une couche

L'utilisateur peut copier et coller une couche. L'opération aura pour effet de créer une nouvelle couche et d'y dupliquer le contenu de la couche source.

3.3.4.5 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 seront réordonnées en fonction de leur priorité.

3.3.4.6 Modifier la couleur des jeux de données

L'utilisateur peut modifier la couleur d'affichage d'un jeu de données. Deux jeux de données ne peuvent pas avoir la même couleur.

3.3.4.7 Modifier la priorité d'une couche de données

L'utilisateur peut modifier la priorité d'une couche de données à partir du gestionnaire de partitions.

3.3.5 Partitions génériques

Une partition générique sert à porter de l'information utilisateur. Cette information 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.

Voici un exemple qui illustre l'usage d'une partition générique où l'information ne doit pas être interpolée :

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

Voici un exemple qui illustre l'usage d'une partition générique où l'information doit être interpolée :

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

3.3.5.1 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 **Additionne** (struct_utilisateur a, struct_utilisateur b)
- struct_utilisateur **Multiplie** (réel x, struct_utilisateur a)

Il est important de noter que d'autres règles devront peut-être se voir définir par l'utilisateur. La liste exhaustive des règles sera faite à l'étape du design.

3.3.5.2 Construire la partition générique

L'utilisateur peut créer des partitions génériques en assemblant des polygones au moyen de lignes, à l'aide de l'éditeur de sous-domaines. Il forme ainsi une série de sous-domaines. Ces sous-domaines peuvent être reliés ou non. L'utilisateur peut ensuite assigner de l'information aux sommets ou aux sous-domaines qu'il a définis.

3.3.6 Édition de sous-domaines

3.3.6.1 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.

3.3.6.2 **Sélectionner un sous-domaine**

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

3.3.6.3 **Effacer un sous-domaine**

L'utilisateur peut effacer un sous-domaine.

3.3.6.4 **Fusionner deux sous-domaine**

L'utilisateur peut fusionner deux sous-domaines.

3.3.6.5 **Ajouter un sommet**

L'utilisateur peut ajouter un sommet sur une ligne fermée existante.

3.3.6.6 **Relier deux sommets**

L'utilisateur peut relier deux sommets pour créer une ligne.

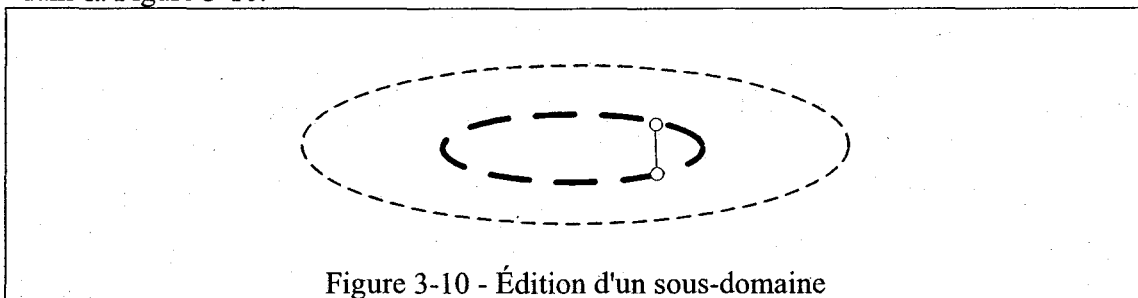
3.3.6.7 **Enlever un sommet**

L'utilisateur peut enlever un sommet.

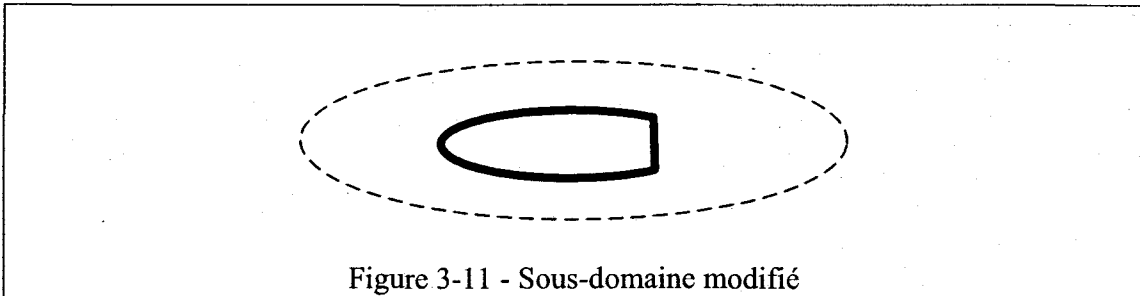
3.3.6.8 **Exemple**

Voici un exemple d'utilisation de l'éditeur de sous-domaines dans un contexte d'utilisation de construction d'un couche de données basée sur J2 (en gras).

En utilisant le mode édition avec peau, l'utilisateur part de la peau de J2. Il y ajoute deux sommets. Il relie ces deux sommets. Ceci a pour effet de créer le sous-domaine visible dans la Figure 3-10.



Finalement, il efface le sous-domaine à droite du segment et obtient le sous-domaine de la Figure 3-11.



3.4 Activité : Maillage

3.4.1 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.

Afin d'aider à la génération de son maillage, l'utilisateur est appelé à segmenter son maillage en sous-domaines fermés. La construction du maillage se fait en utilisant l'éditeur de sous-domaines (voir 3.3.6). L'utilisateur peut s'aider du « squelette de MNT » (voir 3.3) 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.

La construction des partitions de maillage est basée sur un principe de superposition de couches paramètres-maillage en fonction de leur priorité. Une couche paramètres-maillage est constituée d'un sous-domaine qui se voit assigné des paramètres de maillage

Les paramètres de maillage servent à spécifier le degré de raffinement du maillage. Ils peuvent être déterminés de deux manières :

- 1) par la taille des mailles
- 2) par le niveau de tolérance à l'erreur

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

Lors de l'étape de superposition, les paramètres de maillage associés aux couches qui se retrouvent à l'avant plan seront donc ceux qui seront utilisés lorsque le maillage sera généré, y compris, bien sûr, dans les sous-domaines adjacents.

A DEPLACER dans une autre section

L'activité « Maillage » permet d'associer le maillage à un MNT. L'objectif est de créer des champs pour chacun des aspects du terrain décrits dans le MNT. Par exemple, pour un MNT composé d'une partition de topographie et d'une partition de glace, il y aurait génération d'un champ de topographie et d'un champ scalaire de glace.

3.4.2 Paramètres de maillage

3.4.2.1 Taille des mailles

L'utilisateur peut spécifier la taille des mailles aux sommets de son sous-domaine, sur son sous-domaine en entier, ou encore à des points de contrôle quelconque sur le sous-domaine.

3.4.2.2 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 un sous-domaine. L'utilisateur spécifie également un niveau de tolérance à l'erreur acceptable pour ce sous-domaine. Il est possible d'assigner une valeur à plus d'un sous-domaine simultanément.

Une fois le niveau de tolérance fourni, le mailleur se charge de mailler ce sous-domaine 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 statiques. Dans le cas de données statiques, comme la topographie par exemple, 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.

3.4.3 Interface usager pour la gestion d'une partition de maillage

L'utilisateur se voit présenter une interface lui permettant d'interagir avec sa partition de maillage et de gérer la priorité des couches paramètres-maillage. L'utilisateur accorde une priorité aux couches paramètres-maillage en les déplaçant dans une arborescence.

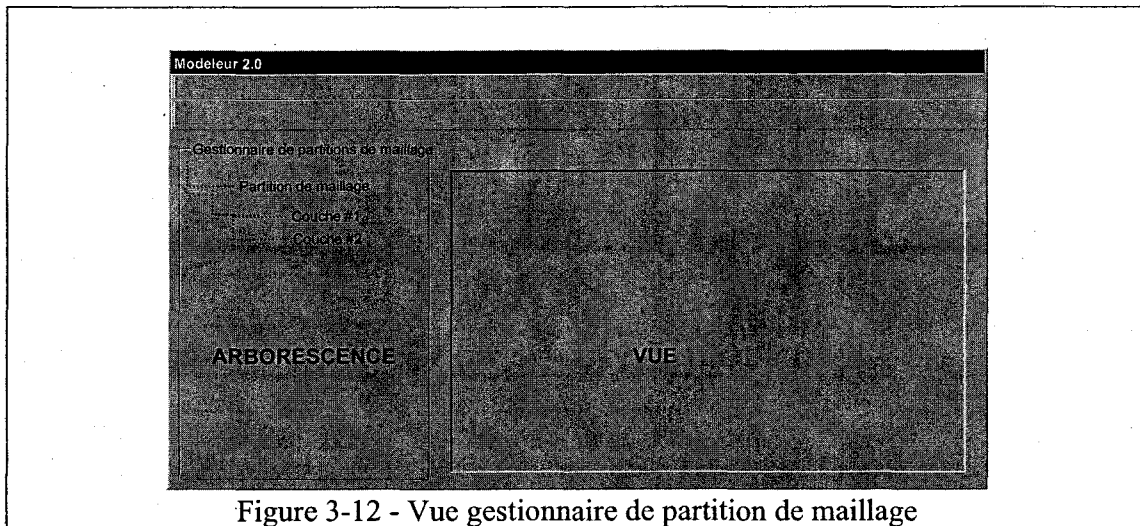


Figure 3-12 - Vue gestionnaire de partition de maillage

L'action d'effectuer un double-clic sur une couche paramètres-maillage, a pour effet de changer le contenu affiché dans la fenêtre « vue ». L'utilisateur peut alors modifier la couche, c'est-à-dire modifier le sous-domaine et ses paramètres. Les autres couches sont également affichées, mais ne peuvent être modifiées.

Si le double-clic a eu lieu sur l'item partition de maillage, la fenêtre « vue » affiche la vue effective de la partition de maillage. Les couches paramètres-maillage sont ainsi superposées l'une sur l'autre en ordre de priorité croissante. Ceci a pour effet de créer des « zones à mailler ». L'utilisateur peut sélectionner chacune d'elles et les mailler.

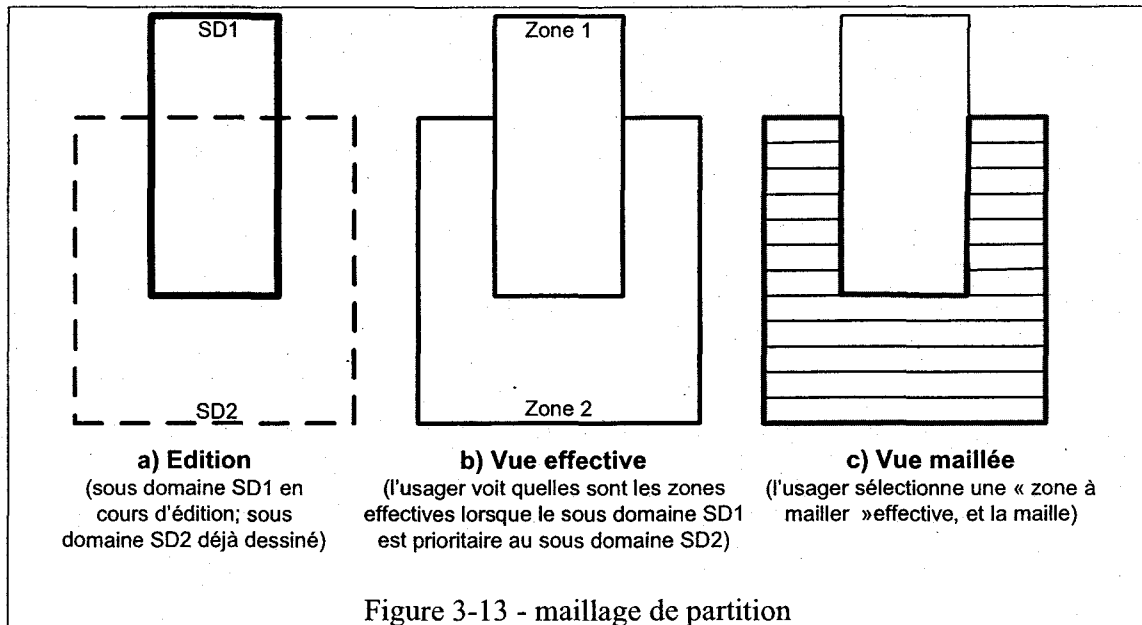
3.4.4 Exemple d'une partition de maillage

Dans la Figure 3-13 a), l'utilisateur est en train de construire une couche paramètres-maillage, et vient de terminer l'édition de son sous-domaine SD1 (dessiné en gras). La couche paramètres-maillage basée sur le sous-domaine SD2 a déjà été construite à une étape précédente. La couche basée sur SD2 est affichée en pointillé car l'utilisateur ne peut la modifier, car seule la couche active peut être modifiée.

Dans la Figure 3-13 b), la vue effective de la partition de maillage est présentée en assumant que la couche basée sur le sous-domaine SD1 est plus prioritaire que celle

basée sur le sous-domaine SD2. De cette figure, on perçoit deux « zones à mailler » effectives : Zone 1 et Zone 2.

Dans la Figure 3-13 c), l'utilisateur a sélectionné la « zone à mailler » effective (Zone 2), et l'a maillée. Les paramètres de maillage utilisés pour cette zone sont ceux qui ont été spécifiés pour le sous-domaine SD2.



3.4.5 Gestion d'une partition de maillage

3.4.5.1 *Nouveau*

L'utilisateur peut créer une nouvelle partition de maillage.

3.4.5.2 *Ouvrir*

L'utilisateur peut ouvrir une partition de maillage existante.

3.4.5.3 *Fermer*

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

3.4.5.4 *Enregistrer*

L'utilisateur peut enregistrer la partition de maillage en cours. Le fait de sauver une partition de maillage sauve la géométrie, les paramètres de maillage et les zones déjà maillées.

3.4.5.5 Enregistrer sous

L'utilisateur peut enregistrer la partition de maillage en cours sous un autre nom. Ceci a pour effet de créer une nouvelle partition de maillage.

3.4.6 Gestion des couches paramètres-maillage

3.4.6.1 Ajouter une couche paramètres-maillage

L'utilisateur peut ajouter une couche paramètres-maillage à sa partition de maillage. Après son ajout, la couche devient la couche active, c'est-à-dire que l'utilisateur peut l'éditer en créant un sous-domaine auquel il devra associer des paramètres de maillage. La couche se voit attribuer par défaut un ordre de priorité maximum.

3.4.6.2 Retirer une couche paramètres-maillage

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

3.4.6.3 Copier et coller une couche paramètres-maillage

L'utilisateur peut copier et coller une couche. L'opération aura pour effet de créer une nouvelle couche et d'y dupliquer le contenu de la couche source. L'utilisateur peut également copier et coller des sous-domaines entre des couches différentes.

3.4.6.4 Modifier la priorité d'une couche paramètres-maillage

L'utilisateur peut modifier la priorité d'une couche de données à partir du gestionnaire de couches paramètres-maillage.

3.4.6.5 Modifier la couleur d'affichage des couches paramètres-maillage

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

3.4.7 Édition

3.4.7.1 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.

3.4.7.2 Afficher les contours des autres couches

L'utilisateur peut afficher le contour du sous-domaine de chacune des autres couches paramètres-maillage de sa partition. L'utilisateur ne pourra toutefois pas interagir avec cette couche d'information. Elle est présente uniquement à titre informatif et l'utilisateur pourra facilement la différencier de la couche active.

3.4.7.3 Éditer une couche paramètres-maillage

L'utilisateur peut choisir entre deux modes d'édition pour le sous-domaine d'une couche paramètres-maillage: l'édition avec l'aide du « squelette de MNT » ou l'édition libre.

L'édition avec squelette du MNT

L'outil affiche une ligne continue délimitant le contour du MNT, ligne à partir de laquelle l'utilisateur peut construire son sous-domaine. Il peut l'ajuster à sa guise à l'aide de l'éditeur de sous-domaines.

L'édition libre

L'utilisateur construit lui même un polygone en reliant des segments de droite entre eux à l'aide de l'éditeur de sous-domaines. Les types d'éléments sont les suivants : T3, T6, T6L, (Q4).

3.4.7.4 Assigner des paramètres de maillage à un sous-domaine

À la partition

L'utilisateur doit spécifier le type de mailleur. Il doit également spécifier le type d'élément qui sera utilisé pour effectuer l'opération de maillage.

Aux sous-domaines

L'utilisateur doit spécifier si le sous-domaine en question sera maillé en spécifiant la taille des mailles ou à l'aide d'un champ d'erreur.

Pour ce qui est des sous-domaines maillés en fixant la taille des mailles, l'utilisateur peut procéder de trois façons :

- en spécifiant la taille des mailles sur le sous-domaine en entier
- en spécifiant la taille des mailles aux sommets du sous-domaine
- en spécifiant la taille des mailles à des points de contrôle qu'il ajoute à son sous-domaine (les points de contrôle deviennent des nœuds du maillage)

Il doit alors spécifier la taille de la maille sous la forme du grand axe, du petit axe et de l'inclinaison, ce qui permet de construire un maillage orienté.

Pour ce qui est des sous-domaines maillés à partir de champs d'erreur, l'utilisateur doit spécifier le champ d'erreur à utiliser. Il doit également fixer une marge d'erreur acceptable (niveau de tolérance).

3.4.7.5 Trouver le niveau de tolérance optimal

Dans le cas des sous-domaines maillés 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.

3.4.7.6 Afficher les zones à mailler

L'utilisateur peut visualiser les « zones à mailler » effectives issues de la superposition des couches paramètres-maillage.

3.4.7.7 Mailler une zone à mailler

L'utilisateur peut sélectionner une zone à mailler, résultat de la superposition des couches paramètres-maillage (et leur sous-domaine).

L'utilisateur peut obtenir de l'information sur sa partition de maillage, notamment le nombre de nœuds et d'éléments, par sous domaine ou globalement.

3.4.8 Génération

3.4.8.1 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 final.

Remarque: À déplacer dans une autre section

3.4.8.2 Générer les champs du MNT

L'utilisateur assemble son maillage avec un MNT. Ceci a pour effet de créer un champ pour chacune des partitions du MNT, à partir des données de cette partition. Si l'utilisateur a respecté la limite du « squelette du MNT », aucun problème ne devrait être rencontré à cette étape du processus.

3.5 Activité : Simulation

Une simulation contiendrait son historique, à chacune des étapes de calculs importante, pour pouvoir modifier des paramètres et vérifier l'effet de la modification.

3.5.1 Gestion d'une simulation

- 3.5.1.1 *Nouveau*
- 3.5.1.2 *Ouvrir*
- 3.5.1.3 *Fermer*
- 3.5.1.4 *Enregistrer*
- 3.5.1.5 *Enregistrer sous*

3.5.2 Définir les paramètres

- 3.5.2.1 *Paramètres simulation*
- 3.5.2.2 *MNT*

Ceci correspond à tous les champs qui seront utilisés pour le calcul.

- 3.5.2.3 *Maillage*

3.5.3 Définir les conditions aux limites

3.5.4 Définir des résultats d'intérêt

3.5.5 Exécuter une simulation (calcul)

- 3.5.5.1 *Local*
- 3.5.5.2 *À distance*

3.5.6 Analyser l'historique de la simulation

- 3.5.6.1 *Vue d'un graphique de simulation*
- 3.5.6.2 *Vue des paramètres d'entrées/sortie à chacune des étapes*
- 3.5.6.3 *Modification de paramètres x itération en arrière*

Si l'outil garde trace des simulations et de leur lien avec un composant (ex : MNT), il est possible d'avertir l'utilisateur que sa simulation pourrait ne plus fonctionner. La simulation pourrait avoir un statut d'intégrité. Si un ou plusieurs composants sont altérés, la simulation n'est plus garantie.

3.6 Activité : Visualisation de données

Ces outils devraient pouvoir être disponibles en tout temps.

3.6.1 Gestion de la visualisation

- 3.6.1.1 *Ouvrir*

L'utilisateur pourra ouvrir un espace, un plan ou une couche sauvegardée au préalable.

3.6.1.2 Fermer

L'utilisateur pourra fermer un espace, un plan ou une couche ouvert au préalable.

3.6.1.3 Enregistrer

L'utilisateur pourra enregistrer un espace, un plan ou une couche en cours.

3.6.1.4 Enregistrer sous

L'utilisateur pourra enregistrer un espace, un plan ou une couche en cours sous un autre nom.

3.6.2 Visualiser graphiquement des données**3.6.2.1 Afficher des maillages****3.6.2.2 Afficher des isolignes****3.6.2.3 Afficher des isosurfaces****3.6.2.4 Afficher des vecteurs****3.6.2.5 Afficher des nœuds en 2D****3.6.2.6 Afficher des nœuds en 3D (topographie)****3.6.2.7 Afficher des profils curvilignes****3.6.3 Construire des échelles de couleur (légendes)****3.6.4 Gestion de l'affichage****3.6.4.1 Sélectionner un arrière plan****3.6.4.2 Modifier l'état d'une couche (affiché/non-affiché)****3.6.4.3 Modifier l'ordre des couches d'affichage****3.7 Activité : Analyse et traitement des résultats**

Ces outils devraient pouvoir être disponibles en tout temps.

- 3.7.1 Sonde
- 3.7.2 Calculatrice
- 3.7.3 Profil en long
- 3.7.4 Projection
- 3.7.5 Outil de projection cartographique
- 3.7.6 Outil de ligne
 - 3.7.6.1 *Étudier les lignes de courant*
 - 3.7.6.2 *Trouver la trajectoire Lagrangienne*
 - 3.7.6.3 *Mesurer des distances*
- 3.7.7 Outil de surface
- 3.7.8 Outil de calcul de débit

3.8 Activité : Construction de séries

- 3.8.1 À définir...

3.9 Activité : Publication de résultats

- 3.9.1 Impression
- 3.9.2 Rapports
 - 3.9.2.1 *Gestion des rapports*

Nouveau

Ouvrir

Fermer

Enregistrer

Enregistrer sous

3.9.2.2 *Édition***3.9.3** **Habillage d'écran****3.9.3.1** *Gestion des habillages d'écran**Nouveau**Ouvrir**Fermer**Enregistrer**Enregistrer sous***3.9.3.2** *Édition***3.9.4** **Animations****3.9.4.1** *Gestion des animations**Nouveau**Ouvrir**Fermer**Enregistrer**Enregistrer sous***3.9.4.2** *Édition***3.10 Activité : Gestion de macros**

L'activité « Système de scripts » a pour but de permettre à l'utilisateur d'automatiser le fonctionnement de Modeleur et de fournir l'accès à la plupart des fonctionnalités « cachées » de Modeleur. L'utilisateur peut ainsi créer des scripts qu'il sera ensuite capable de ré-exécuter.

Pour écrire un script, l'utilisateur a deux choix : soit il édite un script et écrit lui-même le contenu, soit il crée une macro. Une macro est un script qui enregistre automatiquement toutes les actions produites dans Modeleur pendant une période de temps définie par l'utilisateur. Pour que l'utilisateur puisse éditer un script, la version 2.0 de Modeleur sera constituée d'un IDE (Integrated Development Environment) externe contenant un éditeur (et un débogueur).

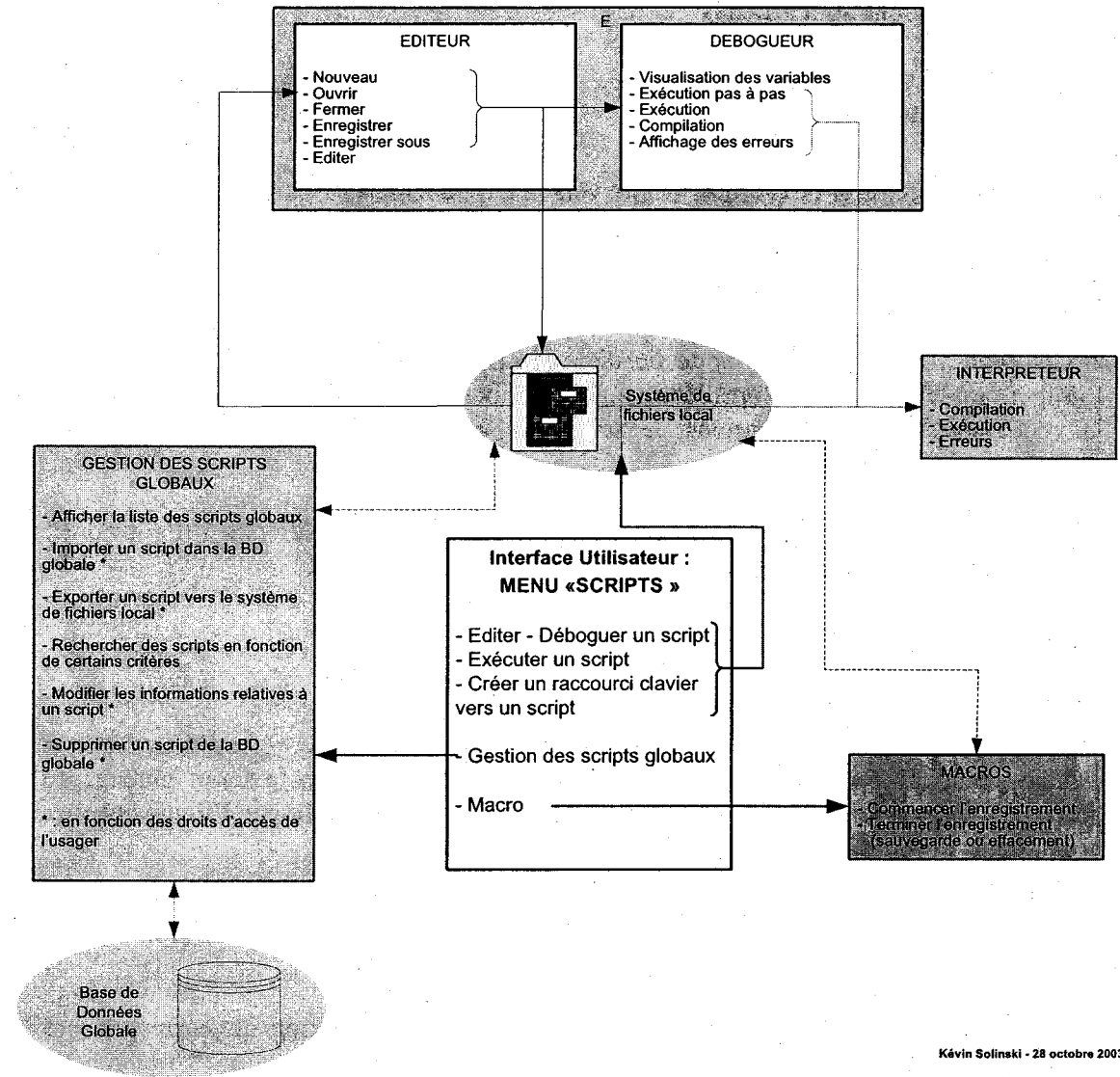
Après avoir sauvegardé ses scripts sur un disque dur, l'utilisateur peut les gérer à l'aide du système de fichiers de son système d'exploitation. Il bénéficie ainsi de toute la souplesse et la facilité que ce dernier peut lui procurer. Il peut également exporter des scripts globaux vers son système de 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 sont partagés entre plusieurs utilisateurs à l'aide de la base de données globale. Seuls les utilisateurs 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 qu'il crée. Ce débogueur externe interagit directement avec Modeleur 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 :

**SCHEMA DES SPECIFICATIONS FONCTIONNELLES
RELATIVES AU SYSTEME DE SCRIPTS**



Kévin Solinski - 28 octobre 2003

3.10.1 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.

3.10.2 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 sauvegardé 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.

3.10.3 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.

3.10.4 Rechercher

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

3.10.5 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 lui permet ensuite de modifier le nom, la date, le type, la description de l'élément sélectionné.

3.10.6 Supprimer

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

3.10.7 Glisser-déposer et Raccourcis-clavier standards

L'utilisateur peut utiliser le « glisser – déposer » ou 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 remplir les informations relatives à chaque script qu'il a décidé d'importer dans la base de données globale.

3.11 Gestion des scripts locaux

3.11.1 Edition – Déboguage

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 Modeleur pour pouvoir être débogué dans ce dernier.

3.11.2 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 Modeleur. L'interpréteur de Modeleur fournit à l'utilisateur des informations sur d'éventuelles erreurs dues à un script bogué.

3.11.3 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.

3.11.4 Macros

3.11.4.1 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.

3.11.4.2 Terminer l'enregistrement

De la même manière, l'utilisateur doit indiquer à Modeleur 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), ce script peut ensuite être modifié par l'éditeur de code précédemment cité.

3.11.4.3 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..

3.12 IDE externe

3.12.1 Editeur

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

3.12.1.1 Caractéristiques nécessaires pour l'intégration dans Modeleur

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.

3.12.1.2 Nouveau

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

3.12.1.3 Ouvrir

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

3.12.1.4 Editer

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

3.12.1.5 Enregistrer

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

3.12.1.6 Enregistrer sous

L'utilisateur peut sauvegarder sous un autre nom le script qu'il était en train de modifier. Cette fonctionnalité permet d'enregistrer les modifications dans un nouveau script sans avoir à écraser l'ancien.

3.12.1.7 Fermer

L'utilisateur peut fermer un script.

3.12.1.8 Quitter

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

3.12.2 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.

3.12.2.1 Caractéristiques nécessaires pour l'intégration dans Modeleur

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

3.12.2.2 Compiler

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

3.12.2.3 Exécuter pas à pas

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

3.12.2.4 *Visualisation de l'état des variables*

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

3.12.2.5 *Exécuter*

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

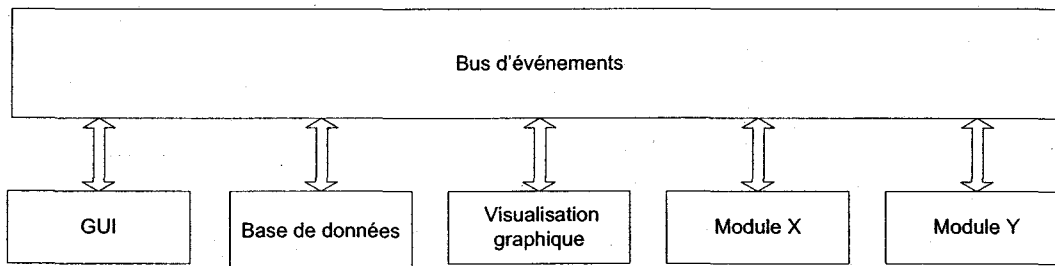
3.12.2.6 *Affichage des erreurs*

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

4 Architecture logicielle

4.1 Bus d'événements

L'architecture de Modeleur 2.0 sera centrée autour d'un bus d'événements qui aura comme fonctionnalité de transmettre des événements entre les modules du système.



Plusieurs raisons justifient le choix du bus logiciel comme pièce centrale du logiciel Modeleur 2.0. 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 qu'il y aie de répercussions 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 même 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.

4.1.1 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 liens 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.

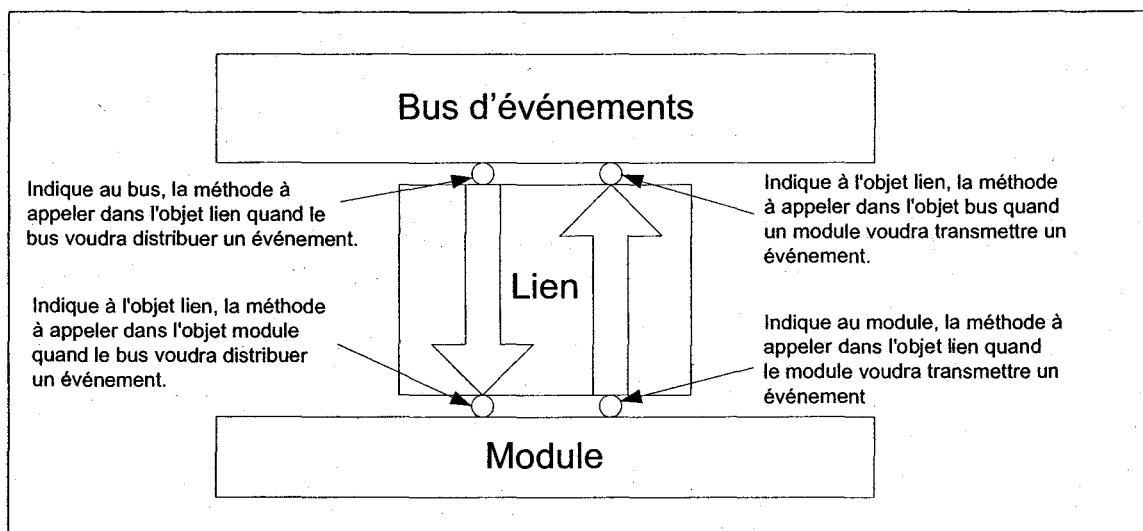
Événement : unité d'échange sur le bus d'événements. Un événement peut contenir à la fois de l'information, une commande ou encore rien.

4.1.2 Fonctionnement général

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

4.1.2.1 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.

4.1.2.2 La phase d'utilisation

Une fois le bus initialisé, tout objet ayant accès au bus pourra lancer des événements. Ces événements seront reçus par le bus. Le bus distribuera les événements à chaque module de sa liste de modules connectés. La distribution s'arrêtera lorsque la fin de la liste sera rencontrée ou lorsque que l'un des modules retournera un message d'erreur.

4.1.2.3 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 sauvegarder son travail avant de quitter.

4.1.3 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'un ancêtre commun ("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.

4.1.4 Lien de type « direct »

Dans un contexte d'utilisation où les liens sont de type « directs », comme c'est le cas dans Modeleur 2.0, 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.

4.2 Base de données

Modeleur 2.0 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 Modeleur 2.0.

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.

Modeleur 2.0 fonctionnera localement avec une base de données relationnelle. C'est dans cette base de données que l'utilisateur effectuera son travail de tous les jours. Il y aura également possibilité d'interagir avec une base de données relationnelle globale, dans laquelle les usagers pourront 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 GIS, 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 et Access. 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 tous les entités qui faisaient partie de Modeleur 1.0 (champs, maillage, partitions), mais permettra en plus la gestion des objets graphiques, de même que des séries.

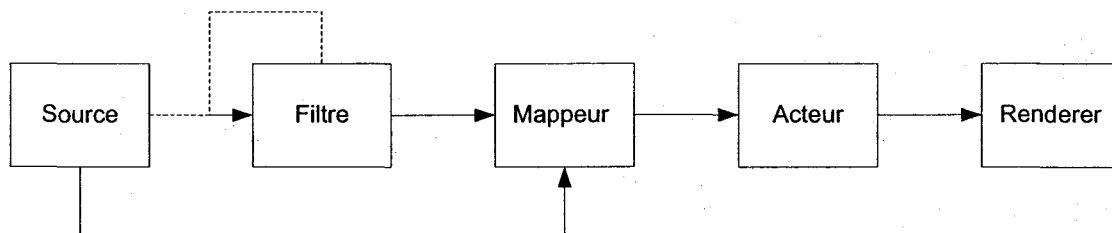
4.3 Système de visualisation graphique

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

Le moteur graphique de Modeleur 2.0 sera basé sur VTK (Visual ToolKit). Cette librairie graphique est relativement de haut niveau et s'appuie sur OpenGL. Le choix de la librairie 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 ».

4.4 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.

Modeleur 2.0 pousse plus loin cette notion de plug-in 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.

Pour que Modeleur fonctionne au minimum, seul le module de base de données doit être en place. Tout ce qui gravite autour est considéré comme un plug-in. Modeleur 2.0 est donc livré avec une série de plug-in.

Dans Modeleur 2.0, un plug-in est composé d'une ou plusieurs DLL, d'un ou plusieurs modules Python, de l'aide et de fichiers de langues.

Dans le cas d'un module ayant besoin d'interfacer avec le GUI (Graphical User Interface), celui-ci rend disponible un script Python qui configure les menus dont il a besoin.

4.4.1 Ajout d'un plug-in

Modeleur 2.0 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.

4.4.2 Gestion des menus

La notion de plug-in est indirectement reliée aux menus du logiciel. En effet les plug-in peuvent offrir leurs fonctionnalités au monde extérieur via les menus. Il est donc important d'avoir un mécanisme simple qui permet aux plug-in de venir s'intégrer à l'interface graphique du logiciel.

Chaque nouveau module agissant comme plug-in doit donc indiquer au gestionnaire de menus (noyau de l'interface graphique) quels sont les menus / sous menus qu'il désire ajouter. Le module doit également spécifier à quel endroit (chemin) dans l'arborescence de menus ceux-ci doivent être ajoutés.

Enfin, le module doit associer un callback à chaque entrée de menu qu'il ajoute. Quand l'utilisateur cliquera sur ce menu, le callback sera appelé et le travail s'effectuera.

4.5 *Engin de script*

4.5.1 Interface utilisateur (UI)

L'interface graphique (GUI) sera entièrement piloté à partir d'un engin de script. C'est donc dire qu'elle sera interprétée à l'exécution.

L'interface graphique de Modeleur pourra donc être modifiée en changeant le script, sans qu'on aie à recompiler quoi que ce soit.

Il sera ainsi possible d'étendre Modeleur et de lui ajouter de nouvelles fonctionnalités en modifiant le script. Le langage de script utilisé sera Python.

4.5.2 Calculatrice

Le langage de script sera également utilisé comme langage de programmation de la calculatrice. Ce langage de programmation permettra l'utilisation du code de Modeleur à partir du script écrit par l'utilisateur. Ceci permettra à l'utilisateur d'avoir accès à beaucoup de fonctionnalités internes de Modeleur.

4.5.3 Macros

Le langage de script sera utilisé pour générer l'envoi d'une suite d'événements sur le bus.

4.6 *Installation via le web*

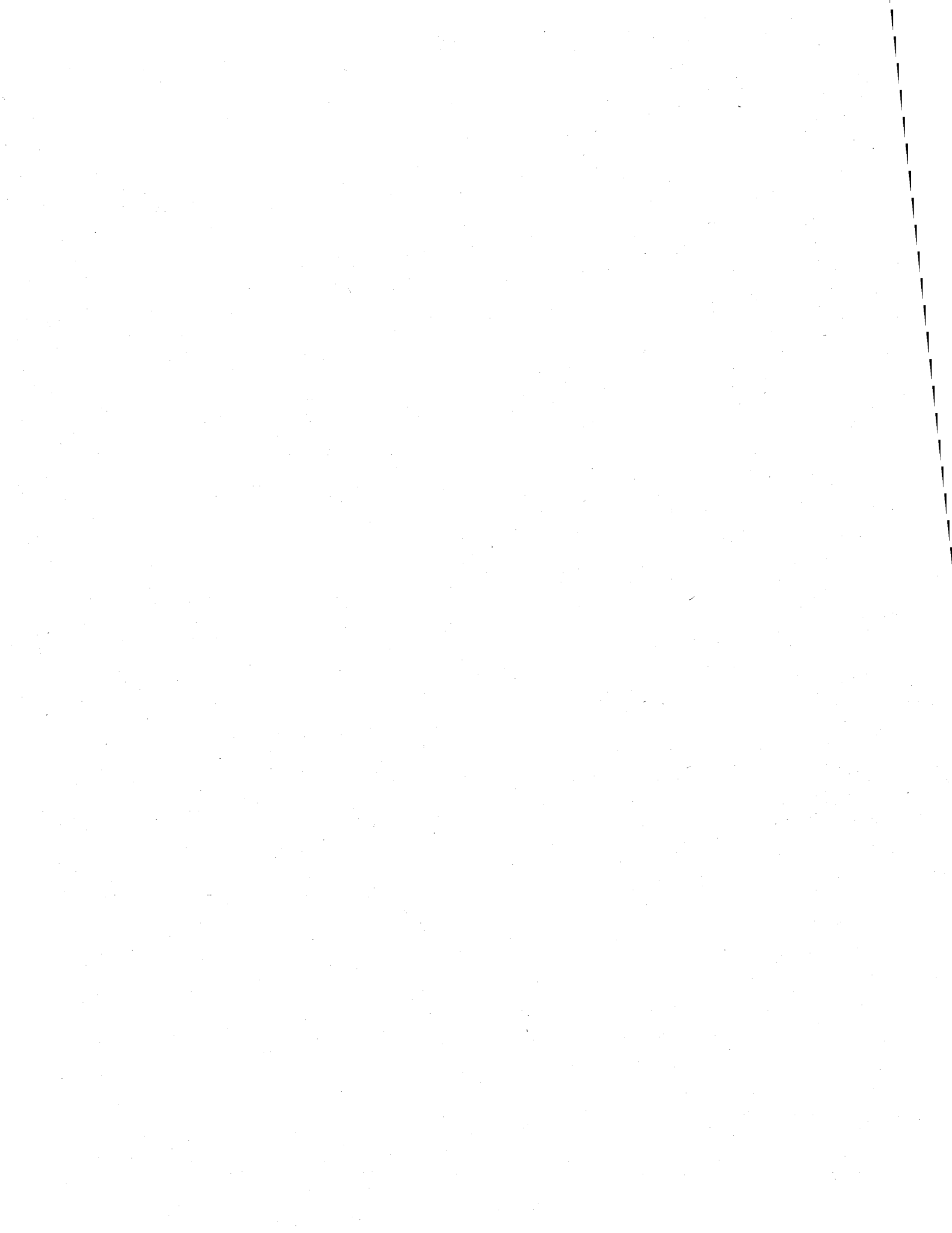
Modeleur devra pouvoir s'installer à partir du web. En d'autres mots, les usagers n'auront pas de livrable à proprement parler. Ils auront toujours accès à la plus récente version de Modeleur en effectuant une opération de mise à jour.

5 GUI

5.1 Organisation des fenêtres

Modeleur 2.0 est une application multi fenêtres. L'utilisateur crée ou charge une donnée, ce qui a pour effet d'ouvrir une fenêtre activité adaptée à cette donnée. Par exemple, si l'utilisateur ouvre un MNT, la fenêtre activité MNT sera chargée.

5.2 Menu



6 Aide, langue et distribution du logiciel

À venir.

7 Glossaire

Maillage
MNT
Partition
Champ
Série
Espace
Plan
Couche
Donnée

API : Application Programming Interface : Interface de programmation d'applications, contenant un ensemble de fonctions courantes de bas niveau, bien documentées, permettant de programmer des applications de « Haut Niveau ». On obtient ainsi des bibliothèques de routines, stockées par exemple dans des DLL.

C++ : Le C++ est le langage de programmation (compilé) utilisé dans l'écriture du code source de Modeleur. C'est un langage Orienté Objet, multi-paradigme, très puissant et complexe.

DLL : Dynamic Linked Library. bibliothèque de liens dynamiques. Ensemble de routines extraite d'un programme principal, pour être partagées par plusieurs programmes, ou pour optimiser l'occupation de la mémoire (les DLL pouvant être chargées et déchargées à volonté).

Langage interprété / langage de scripts : langage non compilé constitué d'une suite d'instructions interprétées qui sont exécutées de façon séquentielle par un programme intermédiaire appelé interpréteur de commandes.

Macro (Macro-commande) : commande formée par une succession d'autres commandes répétitives. Dans Modeleur 2.0, une macro est une suite d'événements enregistré dans un script Python.

Module C++ : Partie C++ d'un plug-in Modeleur 2.0.

Module Python : un module Python est un élément qui fournit de nouvelles fonctions à un interpréteur Python. Dans Modeleur 2.0, les scripts, les macros et les DLL Python (.pyd) sont considérés comme des modules Python.

Python : langage de script, libre, cousin de Perl dans son esprit, mais beaucoup plus fortement orienté objet, dont le développement a débuté en 1990.

Plug-in: extension à une application qui vient se loger dans l'application elle-même. Une fois installé, on peut utiliser le plug-in de façon tout à fait transparente.

RAD : Rapid Application Developpement. développement rapide d'application avec des outils modernes.

Script: Suite d'instructions simples, peu structurées, permettant d'automatiser certaines tâches en se passant d'un réel langage

Sources de certaines définitions :<http://www.linux-france.org/prj/jargonf/index.html>