

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

RAPPORT D'ÉTAPE #1

Rapport de recherche No R-720-e1

Janvier 2004

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

Projet SEGRI

Rapport d'étape #1

Présenté au

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

15 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 numéricien

Collaborateurs

Renaud Le Boulleur de Courlon
Samuel Ouellet
Maxime Derenne
Sébastien Bédard
Kevin Solinski
Olivier Kaczor
Francis Larrivée
Dominic Richard

Stagiaire
Stagiaire
Stagiaire
Stagiaire
Stagiaire
Stagiaire
Stagiaire
Stagiaire

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) :
Rapport d'étape #1. Québec, INRS-Eau, Terre & Environnement. 41 pages. (INRS-Eau,
Terre & Environnement, rapport de recherche 720 e1)

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

©INRS-Eau, Terre & Environnement, 2003
ISBN : 2-89146-514-8

Tables des matières

1	INTRODUCTION.....	1
2	ANALYSE PRÉLIMINAIRE.....	3
3	SPÉCIFICATIONS FONCTIONNELLES	7
4	PROGRESSION DES TRAVAUX – ANNÉE 2003.....	9
5	RESSOURCES HUMAINES.....	31
6	PLANIFICATION	35
6.1	TÂCHES.....	35
6.2	DIAGRAMME DES TÂCHES	38
7	CONCLUSION	41

1 Introduction

Ce premier rapport couvre la période allant du 6 janvier 2003 au 20 décembre 2003 et conclut la première année du projet SEGRI. L'objectif de ce rapport d'étape est de faire état de l'avancement du projet. Il y sera question des tâches qui ont été accomplies, mais aussi de la façon dont ces tâches s'insèrent dans le plan global du projet SEGRI.

La section 2 présente une analyse préliminaire où l'on discute des grandes décisions qui ont été prises et des raisons pour lesquelles elles l'ont été. La section 1 introduit les spécifications fonctionnelles et fait référence au document Spécifications – Modeleur 2.0. La section 1 décrit en détails quelles ont été les tâches accomplies pendant chaque mois du projet. La section 1 présente les ressources humaines du projet et les tâches que ces personnes ont accomplies. La section 6 discute de la planification du projet.

Avant d'aller plus loin, il est important de mentionner que ce document reprend les grandes lignes des dossiers qui ont été abordés au cours de l'année 2003. Un rapport scientifique a été rédigé pour la plupart de ces dossiers. Ces rapports scientifiques sont disponibles dans le document Rapports de recherche 2003.



2 Analyse préliminaire

Au début de l'année 2003, une phase d'analyse a été menée afin de déterminer comment organiser le travail de la manière la plus optimale en tenant compte des ressources disponibles et du temps alloué au projet.

Déterminer les changements nécessaires à Modeleur pour réaliser SEGRI

Afin d'avoir un modèle de terrain précis, prémisses de base à une analyse de dommage rigoureuse, il a été déterminé qu'une base de données relationnelle devait être mise en place. Ceci est devenu nécessaire afin d'être en mesure d'interagir avec les immenses jeux de données provenant par exemple du laser aéroporté.

Afin d'être en mesure de construire et analyser des courbes de dommage, il a été déterminé que les fonctionnalités offertes par la calculatrice devaient être augmentées. Il a été déterminé qu'un langage de script externe devait être intégré à Modeleur car la calculatrice actuelle aurait dû être réinventée pour répondre aux besoins. L'utilisation d'un langage de script externe permettra non seulement de rendre disponibles de nouvelles fonctionnalités, mais elle aura aussi pour effet d'ouvrir le logiciel vers le monde extérieur et d'offrir des possibilités d'extension pour l'avenir.

Afin d'être en mesure d'étudier les problèmes de risque d'inondation, il a été déterminé qu'un concept de série devait être développé. Ce concept touche de près la notion de champ et de maillage éléments finis, pièces essentielles qui, elles aussi, devront être revisitées.

Comme il vient d'être montré, plusieurs décisions majeures ont dû être prises afin de satisfaire aux besoins du projet SEGRI. Ce travail était nécessaire et devait être fait dès le début du projet étant donné son implication à tous les niveaux. Ces adaptations font que la mise à disposition de « nouvelles fonctionnalités » a été reportée dans l'échéancier.

Travailler avec une petite équipe

Le logiciel Modeleur 1 a été réalisé par une équipe relativement expérimentée, constituée d'environ huit personnes. Étant donné les ressources financières limitées dans le cadre du projet SEGRI, il a été établi que le développement devait se faire avec une équipe réduite. Le cœur de l'équipe sera constitué de deux développeurs seniors auxquels se grefferont des stagiaires. Pour compenser la diminution de la force de travail en comparaison avec Modeleur 1, il a donc fallu revisiter plusieurs des aspects du développement logiciel afin d'être en mesure de construire le successeur de Modeleur 1, avec un niveau de qualité élevé et un jeu de fonctionnalités qui répond aux besoins du projet et aux attentes des partenaires et des utilisateurs.

Développer avec des stagiaires

Pour que le projet avance au rythme désiré, il est impératif que les stagiaires qui viennent travailler au sein du groupe laissent une contribution tangible et réutilisable. Le mot « réutilisable » est important ici car il faut bien voir qu'après le départ du stagiaire, il doit être aisé de reprendre le travail là où il l'a laissé et de continuer à avancer de manière quasi-transparente.

Un stagiaire demeure typiquement environ quatre mois au sein du groupe. Il a donc fallu mettre en place une structure adaptée à ce très haut taux de roulement. Dans cette optique, les mandats de stage ont été découpés en tranches d'environ quatre mois. Pendant cette période, le stagiaire se voit confier un dossier complet (analyse, design, programmation, test) pour lequel il se voit remettre un échéancier. Il agit alors en tant que pilote. C'est à lui qu'il revient d'organiser son travail et de tenir des réunions d'analyse au besoin.

À chacune de ces réunions, il doit produire un rapport. Ces rapports de réunion sont importants car ils permettent de garder une trace des décisions qui ont été prises tout au long du dossier, et les raisons pour lesquelles elles ont été prises. À la fin de son stage, le stagiaire doit rédiger un rapport détaillant chacune des tâches effectuées dans le cadre de son dossier. Ce document sert par la suite de référence une fois que le stagiaire a quitté.

Sans l'ombre d'un doute, cette méthodologie de développement axée sur des stagiaires a des retombées fort positive pour le Québec de demain. Non seulement, un projet comme le projet SEGRI a pour but d'aider à résoudre des problématiques qui touchent de près la population, mais en plus il contribue à la formation de personnel hautement qualifié.

Construire autour d'un bus logiciel

Il a été décidé que l'architecture sera centrée autour d'un bus logiciel auquel sont connectés des modules. Les modules sont des pièces logicielles capables d'envoyer des événements sur le bus et d'en recevoir de celui-ci. Cette approche bus / modules a été retenue car elle minimise le couplage entre les modules eux-mêmes.

Il est donc possible d'en enlever, d'en ajouter, d'en interchanger sans qu'il y aie de répercussions sur le système en entier. Étant donné que tous les événements circulent sur le bus, il est possible d'enregistrer des macros en étudiant ce flot d'événements pour pouvoir le répéter par la suite.

L'approche « par module » permet de circonscrire clairement le rôle à jouer par chacun des modules, tout en gardant le minimum de dépendances avec les autres modules. Elle permet le développement de plusieurs modules de manière concurrente une fois l'interface « événementielle » déterminée. Ceci est très approprié à la définition des mandats de stage, car les liens de dépendance entre les tâches sont minimales. Il sera également possible de corriger des bogues qui pourraient éventuellement être trouvés dans un module sans avoir à toucher au reste du logiciel.

S'assurer une base fonctionnelle en tout temps

Le développement logiciel est souvent ralenti par l'étape d'intégration. L'approche bus / modules permet d'atténuer en partie les effets de cette étape.

L'autre moyen qui sera mis en œuvre est le suivant: s'assurer de toujours avoir une base de code solide dans le gestionnaire de code source CVS (Concurrent Version System), base avec laquelle tous les développeurs devront travailler. Les développeurs ajouteront leur contribution à cette base de code, mais à tout moment durant le développement il devra être possible de construire un version du logiciel fonctionnelle à partir des sources dans CVS.

Dans cette optique, il a été déterminé qu'un système automatisé de compilation et de test serait mis en place. La nuit, le système en question devra pouvoir extraire de CVS les modules nécessaires à la construction du logiciel et pouvoir les reconstruire pour une gamme élargie de compilateurs. Il devra également pouvoir les tester et enfin rendre disponible les résultats de compilation et de test via le web.

S'appuyer sur des composantes externes

Vu la gamme d'améliorations et d'ajouts à faire à Modeleur dans le cadre du projet SEGRI, il a fallu chercher un moyen de diminuer la banque de code à la charge du groupe.

Il a donc été décidé que le projet reposera, dans la mesure du possible, sur des bibliothèques externes gratuites et disponibles sur Internet sous des licences « open source ». C'est ainsi qu'il a été déterminé que la base de données, le système de visualisation et le système de script seront montés à partir de composantes externes. Tous les autres modules qui pourront tirer profit de bibliothèques externes devront le faire.

S'assurer d'être extensible pour l'avenir

Construire un outil qui comblera les besoins de tous et chacun n'est pas une tâche facile, surtout dans le cas d'un logiciel scientifique complexe comme l'est Modeleur. Pour répondre à cette problématique, il a été décidé que l'outil ne devra pas offrir directement des fonctionnalités permettant de faire tout ce qui est imaginable. Il devra plutôt fournir une structure flexible pour pouvoir le faire.

La distinction est importante car elle signifie que ce qui sera livré est une base de logicielle solide et fonctionnelle qui couvrira les besoins de la plupart des usagers. À celle-ci, il sera possible d'intégrer des ajouts externes afin d'augmenter les capacités du logiciel et d'ainsi augmenter sa vie utile. L'important est de mettre en place les mécanismes pour que le logiciel réponde aux besoins des usagers pour longtemps, en ajoutant toujours au même noyau principal des fonctionnalités additionnelles.

Configurer les fonctionnalités disponibles

Il est possible que certains usagers ne veuillent que certaines fonctionnalités du logiciel, et désirent que toutes les autres fonctionnalités soient retirées. Afin de répondre à ce besoin, il a été déterminé qu'il devra être possible de configurer le logiciel de façon à le personnaliser en ne gardant que les modules d'intérêt. L'architecture ouverte basée sur le bus est tout à fait cohérente avec ce besoin.

Rendre disponible le logiciel

Il été déterminé que la structure des licences telle qu'elle l'est actuellement pourrait être revue. En effet, puisque le logiciel sera construit d'éléments qui font partie de la communauté « open source », il est possible que Modeleur soit lui aussi intégré à cette communauté.

Dans cette optique, il a été envisagé de distribuer Modeleur à l'aide d'un système similaire à cygwin (émulation de la plateforme Unix sous Windows). Ce système permettrait la distribution et la mise à jour des différentes composantes de Modeleur à partir du web.

L'intérêt d'utiliser et de mettre en place un tel système est que cela ferait disparaître à toute fin pratique l'étape de distribution du logiciel. De plus, ceci éviterait d'avoir à effectuer les tâches de gestion reliées à l'émission et au contrôle des licences.

Si éventuellement des plug-in sont développés avec des intérêts commerciaux, un système de licences pourrait être mis en place pour combler ce besoin.

Faciliter l'accès au logiciel

Il était difficile d'utiliser efficacement le logiciel Modeleur 1.0 sans avoir suivi au préalable une formation. Il a donc été déterminé que le processus de travail dans Modeleur 2.0 devait être revu et simplifié, pour permettre aux nouveaux usagers de s'y retrouver plus facilement.

Dans l'optique d'adoucir la courbe d'apprentissage et de faciliter la distribution du logiciel dans le monde, il a été également déterminé qu'un tutoriel sera produit. Ce tutoriel expliquera un usage typique de Modeleur, pour un cas donné. À travers ce cas, l'utilisateur visitera chacune des étapes d'un problème à travers lesquelles il découvrira les fonctionnalités offertes par le logiciel.

3 Spécifications fonctionnelles

Afin de produire un logiciel qui répond aux besoins des utilisateurs, il importe de rencontrer ces utilisateurs et d'aller chercher leurs commentaires et suggestions.

Dans le cas de Modeleur 2, cette étape a été réalisée au moyen « d'entrevues » avec les utilisateurs / partenaires. C'est ainsi qu'ont été rencontrées les personnes suivantes :

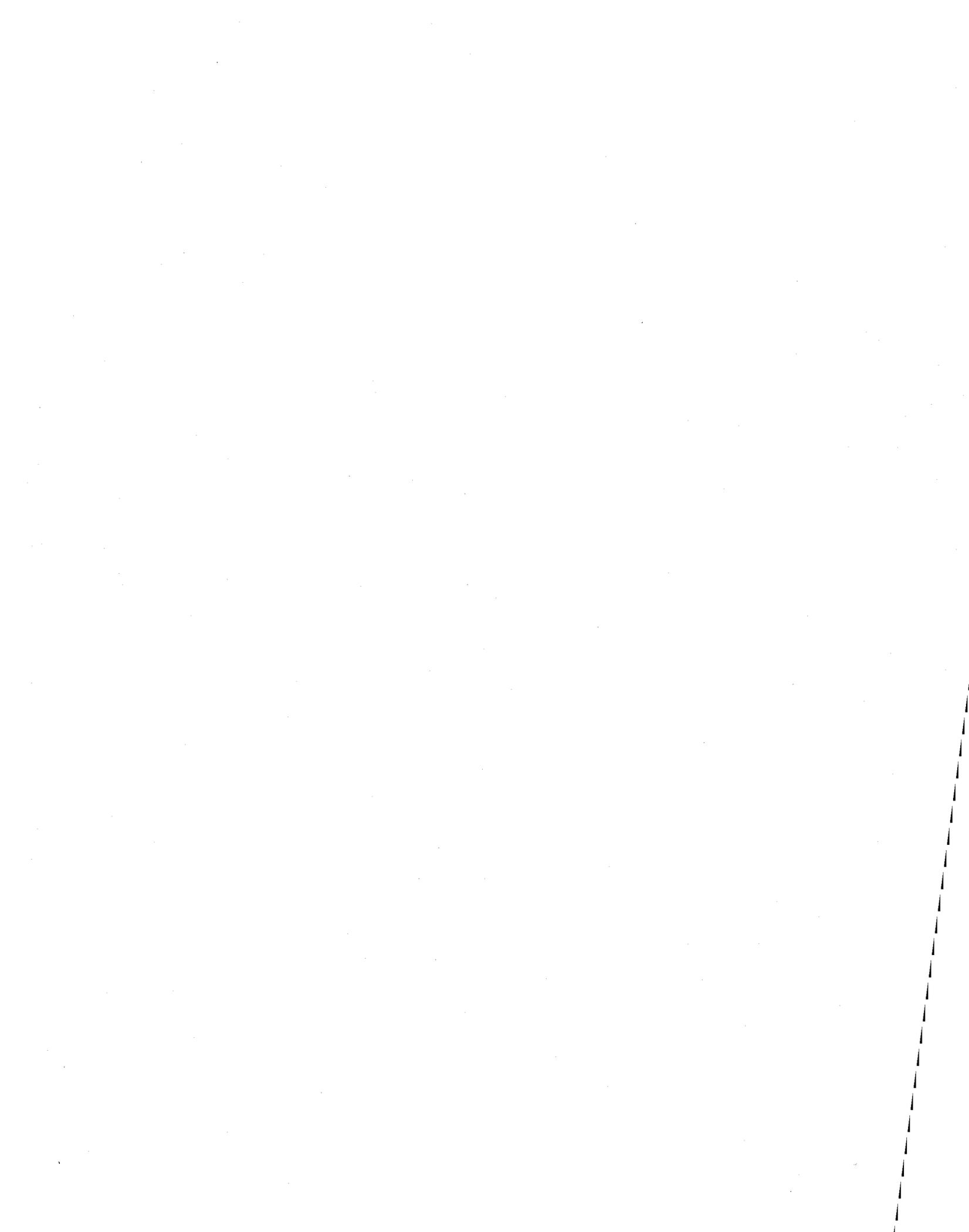
- Jose Bechara (Université Nationale du Nord-Est, Corrientes, Argentine)
- Jean Morin, Olivier Champoux, Daniel Rioux (Environnement Canada)
- Paul Boudreau, Michel Leclerc (INRS-ETE)
- Renaud le Boulleur de Courlon (Institut National Polytechnique de Toulouse)
- Jean Gauthier (BPR)
- Richard Frenette (Université d'Ottawa)
- Pierre Bélanger (GPR)
- André Robitaille, Michel Carreau (Synexus Global)

Des rapports d'entrevue ont été rédigés suite à ces rencontres. Ces rapports ont servi à la rédaction des spécifications fonctionnelles de Modeleur 2.0.

Les spécifications fonctionnelles incorporent également toute l'expérience de l'INRS-ETE tant dans l'utilisation de Modeleur que dans la formation des utilisateurs de Modeleur / Hydrosim.

Les spécifications fonctionnelles sont en cours de rédaction. Plusieurs sections sont assez avancées, notamment les sections sur la validation des données, sur le Modèle Numérique de Terrain (MNT), sur le maillage et sur le module de script.

Les spécifications fonctionnelles sont disponibles dans le document Spécifications – Modeleur 2.0.



4 Progression des travaux – année 2003

Janvier 2003

Arrivée d'un développeur senior au sein de l'équipe

Le début de l'année 2003 a salué l'arrivée au sein du groupe d'Eric Larouche, un ingénieur en informatique. Son rôle est d'agir en tant que développeur senior. Il a pour principales tâches de participer à la gestion du projet, d'aider au recrutement et à l'encadrement des stagiaires, de participer aux réunions d'analyse des différents dossiers et de paver la voie à suivre au moyen de prototypes.

Évaluation des besoins en logiciel / matériel

La première étape du projet visait à évaluer les besoins en ressources matérielles et logicielles pour répondre aux besoins des nouveaux développeurs (stagiaires) qui allaient se joindre à l'équipe au cours des mois suivants. Une analyse a été faite pour déterminer les besoins matériels / logiciels de chaque poste à combler jusqu'à la fin 2003, en cherchant à réutiliser le matériel existant pour minimiser les coûts.

Premier contact avec les cégeps

Des contacts ont été faits avec les différents Cégeps de la région de Québec, en vue d'accueillir des étudiants au début du mois de mars 2003. Peu d'institutions offraient des stages à cette période. Généralement, les stages ont lieu en janvier ou en mai.

Évaluation des besoins pour un environnement de compilation automatisé

Une recherche préliminaire a été menée afin de trouver des pistes de solution pour implanter un environnement de compilation automatisé. L'objectif était de pouvoir mettre en place un système permettant d'exécuter des compilations la nuit, lequel tiendrait compte des dépendances entre les modules, tout en supportant de multiples compilateurs. Un mandat de stage fut rédigé autour de cette problématique.

Analyse de la base de données Modeleur existante

La base de données de Modeleur 1.0 a été revue afin d'évaluer la pertinence de continuer à la supporter. Suite à cette analyse, il a été décidé qu'une base de données relationnelle devrait remplacer l'actuelle base de données pour être en mesure de répondre aux besoins futurs. Il a été déterminé que les données présentes dans l'ancienne base de données allaient devoir être portées vers la nouvelle base de données.

Réunion: présentation générale et gestion des données (Synexus Global)

Une réunion a été tenue dans les bureaux de Synexus Global à Montréal. Michel Carreau (Synexus Global), André Robitaille (Synexus Global), Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait pour but de présenter le projet en général, et de discuter de l'aspect gestion de données au niveau du logiciel Simsys, développé par Synexus Global. Un rapport détaillant le contenu de la réunion a été rédigé.

Réunion: environnement de développement (GIREF)

Une réunion a été tenue au GIREF (Groupe Interdisciplinaire de Recherche en Éléments Finis). Eric Chamberland (GIREF), Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait pour but de discuter du processus de développement en place au GIREF. Il a été principalement question des compilations et des tests automatisés, répétés chaque nuit pour une gamme étendue de compilateurs, de même que de la présentation des résultats de compilation / test sur une page web. Les pistes de solution trouvées lors des recherches préliminaires ont été présentées à M. Chamberland.

Achat de logiciels et d'ordinateurs

Un total de six ordinateurs P4-2.4 ont été achetés pour remplacer les P2-266 qui étaient désuets pour les besoins du projet. Ces achats ont été faits en collaboration avec d'autres projets. L'environnement Microsoft Visual Studio a été choisi comme environnement de développement. Des licences ont donc été achetées de manière à équiper toutes les stations des développeurs avec ce logiciel. Le logiciel Microsoft Visio a été choisi comme outil pour faire des diagrammes. Le logiciel Microsoft Project a été choisi pour gérer le projet. Finalement, les compilateurs Intel C++ et Intel Fortran ont été achetés pour les parties de code mixte C++/Fortran.

Mise à jour du site web

Le site web du groupe a été mis à jour (www.gre-ehn.inrs-ete.quebec.ca). Les pages Web ont été revues. L'objectif principal était de remettre le site sur pied avec un contenu à jour de façon à pouvoir y publier des offres de stages, et ainsi pouvoir rejoindre des candidats de partout dans le monde.

Février 2003**Installation et configuration des machines**

En plus des ordinateurs achetés, cinq machines ont été offertes par BPR, l'un des partenaires dans le projet. Tous les ordinateurs ajoutés au parc informatique ont dû être configurés pour être conformes aux procédures en place dans le groupe. Plusieurs logiciels ont également dû être réinstallés. D'autres ordinateurs ont été modifiés afin

d'augmenter leur durée de vie, et d'éviter d'avoir à acquérir des machines neuves. Cette étape a donné lieu à la mise à jour de certaines machines, à des reformatages et à de la réinstallation. Toutes les machines sont maintenant montées sur un système d'exploitation au moins égal à Windows NT 4.0.

Mise en place de SYGET

Un outil de gestion de temps nommé SYGET a été mis en place. L'objectif d'un tel outil est de colliger le temps passé sur chaque activité. Le projet a donc été découpé en plusieurs sous projets, et des tâches ont été spécifiées pour chacun d'eux. Différents bogues ont été rencontrés avec cet outil lors de la phase de démarrage. Ceux-ci ont été corrigés, et le logiciel fonctionne maintenant correctement. Le processus de cueillette des feuilles de temps s'est alors amorcé, et tous les participants au projet s'y conforment.

Réunion : base de données relationnelle (Environnement Canada)

Une réunion a été tenue dans les bureaux d'Environnement Canada. André Plante (EC), Sylvain Martin (EC), Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait pour but de discuter de la mise en place d'une base de données relationnelle dans Modeleur 2.0. M. Plante a alors fourni différents conseils en ce qui a trait aux aspects à prendre en compte avec les bases de données relationnelles. Un rapport de réunion faisant état des sujets discutés lors de cette réunion fut rédigé.

Recrutement de stagiaires techniciens

Une étape de sélection visant à recruter des stagiaires techniciens a été menée. L'objectif était de recruter un stagiaire pour mettre en place un système de compilation automatisé et un autre stagiaire pour migrer le code réutilisable de Modeleur 1.0 vers Modeleur 2.0. Le recrutement s'est fait dans deux institutions : le collège François-Xavier-Garneau et le Cégep de Lévis-Lauzon. Francis Larrivée et Dominic Richard ont ainsi été sélectionnés.

Planification des tâches dans MS Project

La planification du projet SEGRI pour la durée de l'année 2003 a été faite. Le tout a été réalisé à l'aide du logiciel Microsoft Project. Les mandats ont été découpés en tranches d'environ quatre mois de travail, de façon à pouvoir être conduits par des stagiaires. Au début d'un stage, le stagiaire se voit remettre un document décrivant le mandat de son stage. Ce mandat inclut le calendrier des tâches.

Migration d'Hydrosim vers H2D2

L'architecture d'Hydrosim a été revue afin de rendre le logiciel plus flexible au niveau de la gestion des données. Une approche orientée objet a été appliquée au code Fortran. Il est maintenant possible d'intégrer de nouvelles méthodes de résolution, de nouveaux algorithmes et de nouvelles méthodes de stockage, à partir de l'externe.

Arrivée d'un stagiaire pour travailler sur la partie application sur la Chaudière

Un stagiaire français (Institut National Polytechnique de Toulouse INP - ENSEEIHT), Renaud le Boulleur de Courlon, a été recruté par Michel Leclerc et s'est intégré à l'équipe en février. Son mandat était de collecter les données relatives à la rivière Chaudière et de monter un Modèle Numérique de Terrain (MNT) pour pouvoir mener des simulations hydrodynamiques sur le tronçon de Sainte-Marie de Beauce.

Mars 2003**Recrutement de stagiaires universitaires**

Une étape de sélection visant à recruter des stagiaires universitaires a été menée. L'objectif était de recruter un stagiaire pour le développement d'un module basé sur une base de données relationnelle, un stagiaire pour le module de visualisation et un stagiaire pour travailler sur H2D2. Le recrutement s'est fait dans quatre institutions : l'Université Laval, l'Université de Sherbrooke, l'UQTR et l'UQAC. Samuel Ouellet (Université de Sherbrooke), Maxime Derenne (Université de Sherbrooke) et Sébastien Bédard (Université Laval) ont ainsi été sélectionnés. Des trois stagiaires recrutés, deux ont bénéficié d'une bourse d'excellence du CRSNG.

Réunion: environnement de développement (BPR)

Une réunion a été tenue aux bureaux de BPR. Jean Gauthier (BPR), Marco Grondin (BPR), Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait pour but de visiter le processus de développement logiciel en place dans cette entreprise. M. Grondin a alors présenté différents aspects du processus en place chez eux. Il a été principalement question de la phase de test. Il a été discuté d'une contribution possible de la part de BPR au niveau d'un banc d'essai pour réaliser des tests unitaires.

Cours de Modeleur

Un cours de Modeleur 1.0 a été dispensé par Yves Secretan. Ceci a permis de présenter en détails Modeleur 1.0 à l'équipe de développement.

Réunion: spécifications fonctionnelles (Jose Bechara)

Une réunion a été tenue à l'INRS-ETE. Jose Bechara, Yves Secretan et Eric Larouche ont assisté à la réunion. M. Bechara est un professeur invité de l'Université Nationale du Nord-Est à Corrientes (Argentine). M. Bechara est biologiste et travaille sur la modélisation des micro-habitats. Dans le cadre de son travail, il a eu à utiliser Modeleur 1.0. La rencontre avait pour but de recueillir ses commentaires et suggestions afin de rédiger les spécifications fonctionnelles de Modeleur 2.0. M. Bechara a principalement

discuté de l'aspect gestion de données (import / export), principal problème auquel il a eu à faire face lors de son utilisation de Modeleur.

Analyse et recherche d'un outil de compilation automatisée

Une recherche a eu lieu afin d'expérimenter des outils pour la mise en place d'un système de compilation automatisé. Différents outils ont été évalués, notamment Tmake, Cmake, Jam et Boost-Jam. L'outil retenu fut Boost-Jam car il était le seul à offrir la gamme de fonctionnalités attendue, tout en étant configurable pour effectuer des tâches avancées.

Écriture du cahier de charge GUI (Graphical User Interface) pour Synexus Global

Un cahier de charge a été écrit afin de confier une tâche à un partenaire dans le projet, Synexus Global. Le mandat était de convertir le GUI de Modeleur 1.0 vers Modeleur 2.0, et de mettre en place la mécanique pour communiquer avec le bus logiciel à partir de VB.NET. Suivant la réception du mandat, un appel conférence a eu lieu afin de clarifier certains détails.

Collecte des données pour le MNT

Dans le cadre de la partie application sur la rivière Chaudière du projet SEGRI, une collecte de données a été menée auprès de différents intervenants afin de pouvoir construire le Modèle Numérique de Terrain (MNT). Les données ont été recueillies via le ministère de l'environnement du Québec, Ecce Terra Inc. (firme d'arpenteurs-géomètres de Sainte-Marie de Beauce) et le Centre d'Expertise Hydrique (CEH).

Réunion : application sur la Chaudière (Ste-Marie de Beauce)

Une réunion a été tenue à Sainte-Marie de Beauce. Jean Gauthier (BPR), Jean-Francois Cyr (Centre d'expertise hydrique), Simon Dubé (Centre d'expertise hydrique), Van Diem Hoang (Centre d'expertise hydrique), Bruno Gilbert (Ville de Sainte-Marie de Beauce), Robert Mathieu (Arpenteur), Michel Leclerc, Yves Secretan, Paul Boudreau et Renaud le Boulleur de Courlon ont assisté à la réunion. La rencontre avait pour but de discuter du cas de la rivière Chaudière et de la collecte des données.

Avril 2003

Mise en place d'un système de back up

Un plan de back up et un calendrier de back up ont été mis en place afin de s'assurer que le travail réalisé dans le cadre du projet soit gardé en sécurité. Un système de back up HP DAT 40x6E a ainsi été acheté, afin d'orchestrer efficacement le plan. Chaque fin de semaine une sauvegarde complète du répertoire de travail de chaque membre du groupe est faite, et les autres soirs, une sauvegarde incrémentale est exécutée. Le logiciel utilisé

pour les tâches de gestion de back up est BackupExec de la compagnie Veritas (www.veritas.com). Chaque semaine, les cassettes de back up doivent subir une rotation selon ce qui est spécifié dans le calendrier de back up.

Rédaction des normes de programmation

Des normes de programmation pour le langage C++ ont été adaptées à partir d'anciens documents. L'objectif poursuivi était de mettre en place les structures pour s'assurer de livrer du code standard et dépersonnalisé même si plusieurs développeurs auront à se succéder tout au long du projet. Une section détaillée sur la théorie du contrat a été ajoutée, laquelle est sous jacente au développement de code qui s'auto défend contre les erreurs de programmation. Dans l'optique d'offrir du code robuste, des séances de révision ont lieu afin de s'assurer que le code livré respecte bien les normes en vigueur.

Rédaction d'un document décrivant le processus de développement du groupe

Un document a été rédigé afin d'expliquer le processus de développement logiciel prescrit pour tous les membres du groupe. Dans ce document, il est question des responsabilités de chaque développeur lorsqu'un dossier leur est confié. Le document détaille également quels sont les mécanismes en place dans le groupe pour mener à bien le développement logiciel. Parmi celles-ci, on peut noter: CVS, Boost-Jam, environnement multi compilateurs, résultats de test / compilation sur le web, web++. Enfin, le document discute de différents aspects reliés aux tâches, aux rapports, aux tests et aux réunions.

Analyse des dépendances du code de Modeleur 1.0

Une étape d'analyse visant à déterminer les dépendances entre les classes C++ présentes dans Modeleur 1.0 a été menée à l'aide d'un outil développé par John Lakos, une référence dans le domaine de l'analyse de dépendances. L'objectif de cette démarche était d'arriver à briser les dépendances complexes entre les classes, afin de pouvoir les séparer en modules finis, réutilisables dans Modeleur 2.0. Un document a été écrit, lequel document explique formellement les changements à faire pour pouvoir construire ces modules utilitaires.

Nettoyage et support multi compilateur pour le code à récupérer de Modeleur 1.0

Le code à récupérer de Modeleur 1.0 a été nettoyé et standardisé. Il a été migré avec succès sur une gamme élargie de compilateurs. Les compilateurs supportés sont les suivants : Microsoft Visual C++ 7.0, Borland C++ 5.0.1 / 5.5, Intel C++ 7.0, gcc 3.2 et Watcom. Malgré le fait qu'il soit moins récent que les autres, Watcom fait partie de la liste de compilateurs car il constitue un couple Fortran/C++ solide. Afin de continuer à le supporter, il a toutefois été nécessaire d'utiliser une émulation des bibliothèques standards appelée STL Port. Le fait de supporter de multiples compilateurs permet de produire du code standard qui possède un haut niveau de qualité. Ceci permet également de déceler des erreurs qui seraient autrement très difficiles à détecter.

Mise en place de CVS

L'outil utilisé pour la gestion du code source était RCS dans la version précédente de Modeleur. RCS n'offre pas les facilités d'interface graphique et les fonctionnalités qu'offrent les nouveaux logiciels de gestion de code source. Le logiciel qui l'a remplacé dans le monde de la gestion de code source est CVS (www.cvshome.org). Cet engin cadre bien à l'approche « par module », prémisses de base à l'architecture de Modeleur 2.0. De plus, il est adapté au travail partagé. Tous les nouveaux segments de code développés dans le cadre du projet sont placés dans CVS, et c'est le seul endroit par où transige le code. WinCVS (www.wincvs.org) ou Tortoise (www.tortoisecvs.org) sont des outils GUI qui permettent aux développeurs d'interagir avec le serveur CVS.

Définition de la structure CVS

La structure des répertoires dans CVS a été déterminée. La structure adoptée permet d'héberger les projets, et d'isoler les sources et les inclusions. Elle inclut également un répertoire doc et un répertoire build où sont spécifiés les paramètres à utiliser pour la construction du module. Il a été déterminé que les fichiers binaires pré-compilés pourraient éventuellement être enregistrés dans CVS, par version et par compilateur, afin d'éviter d'avoir à les reconstruire. Il reste encore à définir comment gérer les sources et les bibliothèques externes dans CVS.

Mise en place de la solution Boost-Jam

Suite à la décision prise en ce qui a trait à l'outil pour mettre en œuvre le système de compilation automatisé, la solution Boost-Jam (www.boost.org/tools/build) a été implantée. Une analyse a été menée afin de bien intégrer le concept de dépendances au niveau des versions, dans le code de Boost-Jam. L'objectif était de pouvoir spécifier avec quelle version un module devait être construit.

Mise à jour du bus logiciel

La pièce fondamentale de l'architecture de Modeleur 2.0 est le bus logiciel. C'est sur ce bus qu'auront lieu tous les échanges entre les modules. Le bus logiciel a été mis à jour à partir d'un prototype antérieur. Le bus logiciel est maintenant en mesure de charger les modules sous forme de bibliothèques dynamiques (DLL). Les modules se connectent au bus avec un minimum de code à ajouter.

Simulation hydrodynamique pour le cas de la rivière Chaudière

Un maillage hydrodynamique a été construit afin d'y projeter les données du Modèle Numérique de Terrain (MNT). La partition de maillage utilisée est composée de plusieurs sous domaines afin de bien gérer le raffinement du maillage. La détermination des conditions limites et des paramètres d'initialisation s'est faite à l'aide de données recueillies en 2003. Une fois le modèle bien calibré, des simulations hydrodynamiques ont pu être réalisées à l'aide de l'outil Hydrosim. Il est à noter que cette application ne

fait pas encore appel aux fonctionnalités en développement et pour cause. Le but est de préparer le terrain en vue des tests à venir.

Parallélisation de H2D2

Afin de permettre d'exécuter des calculs sur des machines réparties, l'étape du calcul a été revisitée de manière à la segmenter et à pouvoir récupérer les résultats. On vise ainsi à permettre un accroissement et une accélération des tâches de simulation. Une librairie de résolution externe (PETSc - <http://www-unix.mcs.anl.gov/petsc/petsc-2/>) ainsi qu'un partitionneur externe (ParMETIS - <http://www-users.cs.umn.edu/~karypis/metis/parmetis/>) sont utilisés dans ce logiciel. Bien que cette tâche soit avancée, elle n'est toujours pas complétée. En effet, le code d'H2D2 doit être modifié pour permettre les échanges d'information entre les « processus » répartis.

Mai 2003

Construction des modules de bases

Suite à l'analyse des dépendances menée au mois précédent, les huit modules utilitaires suivants ont été construits: configuration, message d'erreur, outil mémoire, système d'exploitation, géométrie, collection, fichier, licence. Il a fallu renommer des classes, en modifier, en segmenter afin de casser les liens. Des projets Visual C++ ont été construits, et des tests ont été menés sur certains modules clés afin de valider que la migration s'est effectuée avec succès.

Réunion: spécifications fonctionnelles (Environnement Canada)

Une réunion a été tenue aux bureaux d'Environnement Canada. Jean Morin (EC), Olivier Champoux (EC), Daniel Rioux (EC), Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait pour but de rencontrer ce groupe d'utilisateurs experts de Modeleur 1.0. Les commentaires de M. Morin, M. Champoux, M. Rioux ont ainsi été recueillis. Un rapport a ensuite été produit, rapport dans lequel sont regroupées les suggestions d'améliorations et d'ajouts de caractéristiques pour la version 2.0 de Modeleur. Ce rapport a constitué un premier pas important vers la rédaction des spécifications fonctionnelles.

Évaluation et choix d'un outil pour le design de la base de données

Afin de lancer le design de la base de données relationnelle, un outil de conception de base de données a dû être sélectionné sur la base de différents critères. Les candidats étaient : Dezign for databases, Case Studio 2 Full, DDS, ERCreator. Le choix s'est arrêté sur Case Studio 2 Full (www.casestudio.com). Les principales raisons pour lesquelles le choix s'est arrêté sur cet outil sont les suivantes : son faible coût, le fait qu'il offre un

système de sauvegarde par version et le fait qu'il supporte et optimise chacun des systèmes de bases de données.

Analyse et choix d'un engin de base de données

Afin de mettre en place la base de données, un engin de base de données a dû être choisi. Les deux principaux critères de sélection étaient les suivants : supporter les coordonnées GIS (Geographic Information Systems) afin d'être ouvert aux échanges avec l'extérieur, et être gratuit afin de pouvoir être installé chez les utilisateurs de Modeleur sans frais. Les candidats étaient : MySQL et PostgreSQL. Le choix s'est arrêté sur PostgreSQL (www.postgresql.org), accompagné de son extension GIS : PostGIS, car il est le seul des deux à offrir un support complet pour les coordonnées GIS (MySQL n'offrant actuellement qu'un support limité). Il est toutefois possible d'utiliser une base de données Oracle car celle-ci supporte les coordonnées GIS et les tables peuvent être générées pour cette base de données.

Analyse et choix d'un outil pour l'affichage graphique

La composante graphique de Modeleur 1.0 avait été développée à l'aide de l'API Win32. Afin d'étendre les fonctionnalités au niveau de l'affichage, l'ancien système a été remplacé par un engin graphique externe. Les candidats étaient : VTK, OpenDX et Open Inventor. Le choix de l'engin graphique s'est arrêté sur VTK (Visual Tool Kit – www.vtk.org) car il est un outil indépendant de plateforme et du système de fenêtrage qui supporte le 2D et le 3D tout en offrant une interface (API) de haut niveau aux programmeurs.

Développement d'un prototype Visual Basic / VTK

Un prototype a été développé afin de valider le lien entre Visual Basic (VB) et VTK. Le prototype a démontré qu'il était faisable de construire une application avec le minimum de couplage entre Visual Basic et VTK. En effet, l'application VB crée une fenêtre, et passe l'adresse de cette fenêtre au module de visualisation via un événement. Le module de visualisation peut ensuite y afficher de l'information sans avoir à connaître autre chose de l'application. Le prototype a ensuite été amélioré au cours des mois de juin à août afin d'intégrer les modules développés au cours de l'été.

Conception d'un banc d'essai pour H2D2

Un banc d'essai a été développé afin de trouver les bogues dans la première version du logiciel H2D2 (Hydrosim 2 – Dispersim 2). La première phase de test a principalement permis de trouver les erreurs dues à l'interprétation des fichiers de commandes. Au cours de cette phase de test, différentes corrections ont été apportées au code Fortran.

Réunion: spécifications fonctionnelles (Michel Leclerc, Paul Boudreau)

Une réunion a été tenue à l'INRS-ETE. Michel Leclerc (INRS-ETE), Paul Boudreau (INRS-ETE), Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait

pour but de recueillir leurs commentaires et suggestions pour Modeleur 2.0. Leurs commentaires et suggestions ont été intégrés dans un rapport, lequel rapport a été utilisé pour construire les spécifications fonctionnelles.

Réunion: spécifications fonctionnelles (Renaud le Boulleur de Courlon)

Une réunion a été tenue à l'INRS-ETE. Renaud le Boulleur de Courlon (stagiaire, utilisateur de Modeleur 1.0), Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait pour but de recueillir ses commentaires et suggestions pour Modeleur 2.0. Ses commentaires et suggestions ont été intégrés dans un rapport, lequel rapport a été utilisé pour construire les spécifications fonctionnelles.

Rapport de développement sur l'environnement de compilation automatisé

Un rapport résumant les principaux sujets reliés au dossier de l'environnement de compilation automatisé a été rédigé. Il y est question des outils qui ont été explorés, de leur comparaison et d'une justification du choix qui a été fait.

Migration de classes maillage éléments finis de Modeleur 1.0

Les classes de maillage éléments finis ont été migrées de Modeleur 1.0. Plusieurs fonctions présentes dans ces classes ont été retirées. L'objectif était de simplifier les classes de maillage, d'enlever tout ce qui était relié à l'algorithmie pour ne laisser que ce qui a trait à sa fonctionnalité de « contenant ». Ces classes ont par la suite servi pour la partie gestionnaire de données et le module de visualisation.

Juin 2003

Évaluation et choix d'un outil de design UML (Unified Modeling Language)

Différents outils de design UML ont été évalués selon certains critères. Les candidats étaient : Posseidon, Visual Paradigm, Objectteering, Visio. L'outil sélectionné fut Visual Paradigm for UML, de la compagnie Visual Paradigm (www.visual-paradigm.com). Les principales raisons qui ont mené à ce choix sont les suivantes : sa gratuité, sa facilité d'utilisation, sa bonne documentation et sa stabilité. Lors des tâches de design, la construction de diagrammes de classes et de séquence est de rigueur. L'emploi de ces diagrammes permet d'avoir une base solide sur laquelle discuter et mettre à l'épreuve le design.

Évaluation et choix d'un outil de bug-tracking

Différents outils de bug-tracking ont été évalués selon certains critères. Les candidats étaient nombreux. Parmi ceux-ci, on peut noter GNATS, Bugzilla et Fast BugTrack. L'outil sélectionné fut Fast BugTrack (www.fastbugtrack.com), de la compagnie Alcea Technologies. Les principales raisons qui ont mené à ce choix sont les suivantes : sa

facilité d'installation / configuration, son faible coût, le fait que le nombre d'utilisateurs soit illimité et son utilisation via le web. L'outil est actuellement en place, et est configuré pour accueillir les rapports de bogues, les améliorations et les suggestions de tout membre de l'équipe de développement. L'accès au serveur Fast BugTrack n'est toutefois pas disponible à l'externe pour le moment.

Design des tables de la base de données relationnelle

Les tables de la base de données relationnelles ont été définies, de même que les liens entre chacune d'elles. Le design inclut les maillages, les champs, les séries, les partitions, et les composantes graphiques. L'aspect GIS a également été pris en compte dans le design. Le design est composé de plus d'une soixantaine de tables. Le design de l'entité maillage a été fait afin de supporter la notion d'arêtes et de faces des éléments. Le design des partitions a été fait en vue de supporter la notion de partition générique. Le design des séries a été fait en reproduisant sensiblement l'organisation des tables dans le maillage.

Installation de l'engin de base de données PostgreSQL

L'engin de base de données PostgreSQL a été installé. Certaines sections du code source de cet outil ont dû être modifiées pour prendre en compte l'aspect GIS et intégrer la librairie PostGIS. Un document décrivant les étapes de la procédure d'installation a été rédigé. La procédure d'installation est relativement ardue et demande l'utilisation de plusieurs logiciels et librairies externes. Lors de la distribution de Modeleur, cette procédure devra être simplifiée. Il est envisagé d'offrir aux utilisateurs une version pré compilée des DLL nécessaires au bon fonctionnement de la base de données.

Réunion: présentation du design de la base de données relationnelle

Une réunion a été tenue aux bureaux d'Environnement Canada. André Plante (EC), Yves Secretan, Samuel Ouellet (stagiaire INRS-ETE) et Eric Larouche ont assisté à la réunion. La rencontre avait pour but de présenter le design à M. Plante, spécialiste des bases de données relationnelles. Lors de cette réunion, M. Plante a commenté le design de manière très positive et suggéré quelques modifications mineures pour le solidifier.

Analyse et design de l'architecture du module de visualisation

Une étape d'analyse visant à élaborer une architecture qui la plus indépendante possible de la librairie graphique a été menée. Ceci a mené au design du module de visualisation, module composé d'une quinzaine de classes. Ce module gère l'affichage de différents plans (arrière, médian, avant, interactif) qui peuvent être composés de couches d'information. L'architecture du module de visualisation fait une distinction claire entre trois intervenants : les « contenants » (ex : espace, plans, couches), les algorithmes (ex : représentation des noeuds, des maillages) et l'implémentation (ex : implémentation spécifique à VTK). Ceci permet d'avoir un design solide même si à un certain moment la librairie graphique est changée. Le design pattern du « composite » et celui du « bridge » ont été utilisés pour mener à bien cette tâche.

Recrutement de stagiaires universitaires

Une étape de sélection visant à recruter des stagiaires universitaires a été menée. L'objectif était de recruter un stagiaire pour le développement d'un module sur les champs / séries et un stagiaire pour le module de script / macro. Le recrutement s'est fait dans une institution : l'Université de Sherbrooke, et via le site web. Olivier Kaczor (Université de Sherbrooke) et Kevin Solinski (ENIC Telecom, France) ont ainsi été sélectionnés.

Rapport de développement sur le module visualisation

Un rapport discutant du cas de la rivière Chaudière a été rédigé et distribué aux différents intervenants.

Réunion : présentation des résultats obtenus sur la rivière Chaudière

Une réunion a été tenue à l'INRS-ETE. Jean Gauthier (BPR), Jean-Francois Cyr (Centre d'expertise hydrique), Simon Dubé (Centre d'expertise hydrique), Van Diem Hoang (Centre d'expertise hydrique), Bruno Gilbert (ville de Ste-Marie de Beauce), Michel Leclerc, Yves Secretan, Paul Boudreau et Renaud le Boulleur de Courlon ont assisté à la réunion. La rencontre avait pour but de présenter les résultats de simulation obtenus à ce jour pour le cas de la rivière Chaudière.

Arrivée d'un étudiant à la maîtrise

Laurent Bonnifait, étudiant à la maîtrise à l'INRS-ETE, a été intégré à l'équipe de SEGRI et s'est vu confié le mandat commencé par M. Renaud le Boulleur de Courlon. Il a pour tâche de continuer le travail sur la rivière Chaudière.

Juillet 2003**Serveur web**

Un nouveau serveur web a été mis en place. Le nouveau serveur web est un serveur Apache (www.apache.org). Les requêtes faites vers l'ancien serveur web ont été redirigées vers le nouveau.

Construction d'un prototype de traducteur (maillage)

Un prototype de traducteur visant à traduire l'ancienne base de données (Modeleur 1) a été mis en place. La première tâche que le traducteur devait accomplir est de relire les informations reliées aux maillages (les noeuds et les éléments) et de les placer dans la nouvelle base de données relationnelle. Dans la version 2.0 de Modeleur, ce traducteur pourra être utilisé pour convertir les anciens projets.

Design d'un module de gestionnaire de données

Afin de pouvoir échanger des données entre les modules, il était nécessaire de construire un gestionnaire de données. Le gestionnaire de données devait être responsable d'héberger les données. Il devait être capable de répondre à des événements de création, de chargement, de sauvegarde et de destruction de données. Le design est donc articulé autour de classes d'algorithmes responsables d'exécuter ces opérations pour un type de données précis, et de faire le lien avec les tables reliées à ce type de données dans la base de données relationnelle. Le design pattern de « l'object factory » a été utilisé pour réaliser ce design.

Implantation d'un module de gestionnaire de base de données

Le module de gestionnaire de données a été implémenté selon le design. Le code qui a été développé utilise une librairie qui interface avec ODBC: la librairie OTL 4.0, à l'aide du langage SQL pour la création des requêtes à la base de données. L'implantation s'est faite à l'aide de la base de données Microsoft Access, ce qui a permis d'identifier certaines différences entre la base de données PostgreSQL. De plus, il est apparu que pour certaines tâches (notamment, la sauvegarde de maillages), il fallait faire appel aux fonctionnalités spécifiques aux engins de données, plutôt qu'aux requêtes SQL standards. Afin d'intégrer l'aspect base de données multiples, le design a dû être revisité quelque peu. Cette nouvelle version n'a toutefois pas été implantée.

Validation du modèle de température de H2D2

Différents tests et scénarios ont été mis en place afin de valider le modèle de température de H2D2. Certaines pistes de solution ont été trouvées pour expliquer des comportements du modèle.

Août 2003

Rapport de développement sur le dossier de la base de données relationnelle

Un rapport résumant les principaux sujets reliés au dossier de base de données a été rédigé.

Réunion : optimisation de la base de données relationnelle

Une réunion a été tenue aux bureaux d'Environnement Canada. André Plante (EC), Yves Secretan, Samuel Ouellet (stagiaire INRS-ETE) ont assisté à la réunion. La rencontre avait pour but de discuter de l'optimisation de la base de données, de chercher à expliquer certains problèmes rencontrés et de trouver des amorces de solution.

Implantation du module de visualisation

L'architecture du module de visualisation développée au cours du mois précédent a été implantée. Il persiste encore quelques lacunes, mais les éléments sont en place pour que ce module soit utilisé dans Modeleur 2.0.

Amélioration du prototype VB / VTK

Le prototype VB / VTK a été amélioré pour intégrer le module gestionnaire de données et le module de visualisation. Ce prototype a constitué le premier pas vers une version de Modeleur 2.0.

Rapport de développement sur le module visualisation

Un rapport résumant les principaux sujets reliés au dossier du module de visualisation a été rédigé.

Analyse et recherche d'un outil de script

Une recherche a été menée afin de trouver un langage de script. Ce langage devait pouvoir jouer plusieurs rôles. Il devait être en mesure de remplacer le langage de la calculatrice. C'est à dire qu'il devrait être possible de rendre une partie de la banque de code C++ accessible à partir de scripts. Il devait permettre la mise en place d'une mécanique de macros (à partir des événements qui circulent sur le bus). Le langage devait aussi servir comme langage pour la partie GUI. Finalement, il devait être cohérent avec la mise en place d'une mécanique de plug-in, qui allait permettre aux usagers d'ajouter des composants GUI à Modeleur 2.0. La recherche a fait ressortir deux candidats de choix : VB .NET et Python.

Rédaction du manuel d'H2D2

Une ébauche du manuel utilisateur d'H2D2 a été faite. Cette ébauche pourra servir à la version finale du manuel. La structure du document est la suivante : description du processus de travail, description des données, description du langage de contrôle.

Septembre 2003

Traduction des champs éléments finis

Le traducteur développé au mois de juillet a été amélioré afin d'y intégrer les champs éléments finis. Les champs éléments finis provenant de Modeleur 1.0 peuvent donc être traduits vers la nouvelle base de données.

Intégration des unités de mesure à la base de données

Une analyse a été menée afin d'intégrer la gestion des unités de mesure (mètre, kilogramme, secondes, etc) à même la base de données. De nouvelles tables ont donc été ajoutées pour couvrir cet aspect.

Analyse sur les champs

Une analyse a été menée afin de bien situer le concept de champ en tant que structure algébrique. Il a aussi été question des champs en tant que structure de données avec support spatial (ex : éléments finis, différences finies, raster, vecteur). Ce fut la première étape qui allait mener vers le design des champs et séries.

Révision des rapports de la base de données et du module de visualisation

Les rapports de la base de données et du module de visualisation ont été révisés et adaptés de manière à pouvoir les intégrer directement en tant qu'annexe dans le rapport d'étape 2003.

Construction du squelette des spécifications de Modeleur 2.0

Le squelette du document décrivant les spécifications de Modeleur 2.0 a été mis en place. Les spécifications de Modeleur sont subdivisées en cinq sections principales. La première section décrit les concepts clés à la base de Modeleur 2.0. La seconde présente les spécifications fonctionnelles. La troisième présente les spécifications techniques. La quatrième décrit les concepts de l'interface utilisateur. La dernière présente divers aspects tel que l'aide, les langues et la distribution du logiciel. Une approche « par activité » a été choisie pour mettre en oeuvre la partie « spécifications fonctionnelles ». L'objectif est de décrire les activités clés que l'utilisateur aura à accomplir avant d'arriver à une solution finale. À partir des réunions sur les spécifications fonctionnelles (réalisées sous forme d'entrevues), une gamme d'activités ont donc été extraites. Ces activités sont les suivantes : gestion des données, validation des données, MNT, maillage, simulation, publication, outils d'analyse, scripts.

Écriture des spécifications fonctionnelles (section : MNT)

Les spécifications fonctionnelles ont été rédigées pour ce qui est de l'aspect terrain (MNT). Le fait d'avoir une activité dédiée au MNT est un concept nouveau. Le concept décrit le modèle numérique de terrain (MNT) comme une superposition de couches de données. Ce concept a été présenté à Paul Boudreau, utilisateur avancé. Il a apprécié l'idée, et a fait ressortir quelques points qui ont permis l'amélioration du document.

Comparaison de la solution Microsoft VB .NET à Python

Une comparaison a été faite entre la solution VB .NET et la solution Python sur la base des critères suivants: interopérabilité avec le C++, niveau de couplage, courbe d'apprentissage, documentation, débogage, interface graphique, plug-in, portabilité, configuration. Suite à cette analyse, Python a été retenu comme langage de script principalement en raison de son interopérabilité avec le C++, de son faible niveau de couplage, de sa courbe d'apprentissage. Un rapport a été réalisé dans le cadre de cette analyse.

Prototypage avec Python, Boost-Python et wxPython

Suite au choix de Python comme langage des script, une phase d'apprentissage et de prototypage a eu lieu. Durant cette phase, le langage Python (www.python.org) et son extension GUI wxPython (www.wxpython.org) ont été étudiés attentivement. En parallèle, l'apprentissage de la librairie Boost-Python et de l'outil Pyste a été fait (voir le site suivant pour obtenir ces deux outils : www.boost.org/libs/python/doc/). Ces deux outils utilisés en combinaison permettent d'accéder aux classes développées en C++ de manière quasi-transparente.

Préparation pour la campagne de terrain sur la rivière Chaudière

Un échosondeur a été acheté en collaboration avec d'autres chercheurs du Centre INRS-ETE dans le cadre du projet. Un partenaire privé a été trouvé pour ce qui est de l'aspect GPS haute résolution : Richard Leclair (Geolocation). Une étape d'apprentissage de l'échosondeur et de mise au point de celui-ci a été réalisée. Un support a également été fabriqué de manière à pouvoir à pouvoir exécuter la campagne de terrain.

Octobre 2003

Écriture des spécifications fonctionnelles (sections: validation de données, maillage)

Les spécifications fonctionnelles ont été rédigées pour ce qui est des activités « validation de données » et « maillage ». La section sur la validation de données a été développée avec l'idée d'intégrer une mécanique basée sur des filtres, qui fournissent une cote à chacun des points, laquelle cote permettra de décider si le point doit être désactivé ou non. La section sur le maillage a été orientée autour du concept d'adaptation du niveau de raffinement du maillage en fonction de l'erreur admissible.

Écriture des spécifications fonctionnelles (section : script)

Les spécifications fonctionnelles ont été rédigées pour ce qui est de l'activité « script ». Lors de cette étape, il a fallu différencier le concept de script et de macro. Il a fallu présenter la notion de script local et script global. Enfin, via un diagramme, les interactions entre les différents intervenants ont été identifiées clairement.

Développement d'un démo illustrant le fonctionnement du bus (Bus View)

Afin d'illustrer le fonctionnement du bus aux futurs stagiaires et aux développeurs externes, une petite démo a été mise en place. Cette démo présente de manière visuelle l'échange d'événements entre les différents modules du système.

Design des champs

Le design des classes de champs a été fait. Ce design a permis d'isoler les types de fonctionnalités que doit supporter la classe mère, et de mettre en place une série de classes enfants (champs éléments finis, champs analytique, champ vectoriel, etc). La principale opération qu'un champ doit supporter est la suivante : être en mesure de retourner sa valeur en tout point. La classe champ est une classe « template » car un champ peut porter n'importe quel type de données qui possède les propriétés algébriques.

Design des séries

Une série temporelle 1D est composée d'un certain nombre de pas de temps. À chaque pas de temps est associé un champ de valeurs (ex : champ de vitesse) avec lequel il est possible de trouver la valeur en tout point $\{x,y\}$. Pour les pas de temps intermédiaires (pas de temps où il n'y a pas de champs associé), les valeurs sont interpolées. Le design des séries a été fait non seulement afin de bien représenter des champs dynamiques, mais aussi avec l'idée de développer un concept plus générique qui peut être réutilisé pour des problèmes plus complexes. Le design des séries a été fait de manière à ce qu'une série puisse être de n'importe laquelle dimension, porter tout type de données (pas seulement des champs) et qu'elle puisse être composée avec d'autres séries.

Transformation des classes de maillage en classes génériques

Conceptuellement, une série se situe très près de ce qu'est un maillage éléments finis dans la mesure où il s'agit du même type de discrétisation. Il a donc été décidé de rendre la classe de maillage générique afin de pouvoir s'en servir autant pour les maillages éléments finis que pour les séries. Les composants de la classe maillage ont donc été migrés afin de les rendre « template ».

Design des itérateurs spatiaux

Les itérateurs spatiaux permettent de visiter un champ en se déplaçant spatialement dans celui-ci. Le design des itérateurs spatiaux a amené le concept de région. Une région correspond à l'espace qui délimite un champ. Les itérateurs se déplacent dans cette région, et permettent d'obtenir la valeur du champ au point du curseur. Pour pouvoir visiter un champs 2D, un itérateur doit être configuré avec deux vecteurs de direction et deux pas. Le concept d'itérateur sera très utile pour balayer un champ. Les itérateurs spatiaux seront également applicables aux séries.

Recrutement de trois stagiaires universitaires pour l'hiver 2004

Une étape de sélection visant à recruter des stagiaires universitaires a été menée. L'objectif était de recruter un stagiaire pour le développement d'un module MNT, un stagiaire pour le développement d'un module validation de données et d'un dernier pour la seconde phase du développement du module de visualisation. Le recrutement s'est fait dans trois institutions : l'Université Laval, l'Université de Sherbrooke et l'UQAM, et via le site web. Cédric Caron (Université de Sherbrooke), Maude Giasson (Université Laval) et Benjamin Behaghel (EPITA, France) ont ainsi été sélectionnés.

Réunion : spécifications fonctionnelles et avancement du projet (Synexus Global)

Une réunion a été tenue chez Synexus Global. André Robitaille (Synexus Global), Michel Carreau (Synexus Global), Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait pour but d'identifier les attentes de Synexus Global envers Modeleur 2.0. La réunion avait également pour but de faire part de l'avancement du projet à M. Carreau, et de lui expliquer les raisons qui ont mené à choisir la solution Python au lieu de la solution .NET. Il a été également question d'une contribution de Synexus Global au projet, un outil nommé la « sentinelle » qui pourrait être utilisé dans Modeleur pour le calcul distant.

Réunion : spécifications fonctionnelles (BPR)

Une réunion a été tenue à l'INRS-ETE. Jean Gauthier (BPR), Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait pour but de recueillir les commentaires et suggestions de M. Gauthier pour le développement de Modeleur 2.0. Beaucoup des commentaires avancés par M. Gauthier recoupaient ceux recueillis jusqu'alors, mais il en est ressorti certaines suggestions intéressantes qui seront prises en compte dans Modeleur 2.0.

Réunion : spécifications fonctionnelles (Richard Frenette)

Une réunion a été tenue à l'INRS-ETE. Richard Frenette (Université d'Ottawa), Yves Secretan et Eric Larouche ont assisté à la réunion. M. Frenette est professeur de génie civil et utilise Modeleur dans le cadre d'un projet de recherche. La rencontre avait pour but de recueillir ses commentaires et suggestions pour le développement de Modeleur 2.0. M. Frenette a des connaissances avancées en éléments finis. Ses commentaires étaient donc plutôt reliés à des aspects bas niveau en ce qui a trait aux algorithmes ou aux méthodes utilisées. Il a ainsi fait ressortir des lacunes au niveau de la documentation.

Mise en place de la macro de compilation

Afin de bien contrôler la construction des modules, et les paramètres avec lesquels ceux-ci sont construits, une macro de compilation multi-compilateurs a été mise en place. Cette macro permet de gérer les paramètres de compilation avec un fichier dans CVS plutôt qu'à l'aide de l'interface usager de Microsoft Visual Studio.

Mise en place de la macro d'import / export

Afin de bien contrôler ce qui est exporté d'une DLL, une macro d'import / export a été mise en place. Ce mécanisme évite qu'une classe ou fonction soit exportée par deux modules différents. Ceci évite les erreurs à l'étape de « link ».

Novembre 2003

Réunion spécifications fonctionnelles (GPR)

Une réunion avec Pierre Bélanger a été tenue aux bureaux de GPR. Pierre Bélanger, Yves Secretan et Eric Larouche ont assisté à la réunion. La rencontre avait pour but de recueillir ses commentaires et suggestions pour le développement de Modeleur 2.0. M. Bélanger a manifesté un intérêt à ce que Modeleur 2.0, en ce qui concerne les inondations, puisse évaluer les risques humains, plutôt qu'uniquement les risques financiers. Il a été également question d'améliorer les techniques pour la validation des données, et d'un possible partenariat pour un projet à l'étranger.

Design / prototypage de la mécanique des plug-in

Une mécanique pour gérer le lien entre les menus dans le GUI, et les modules sur le bus a été mise en place. La partie GUI est responsable de configurer la partie C++ pour que le lien avec les menus se fasse correctement.

Implantation du module de script

À partir des spécifications fonctionnelles, le module de script a été implanté. Ce module intègre la gestion des macros (des scripts générés automatiquement à partir des actions de l'utilisateur).

Rédaction d'une procédure d'installation des logiciels

Une procédure détaillant les logiciels à installer pour un poste développeur a été rédigée. Cette procédure présente la procédure d'installation / configuration de CVS et des outils Python.

Implantation des champs / séries

Les champs et les séries ont été implantés suivant le design qui a été fait au mois précédent. Les classes ont ensuite été exportées en Boost-Python, de manière à ce qu'il soit possible de les utiliser à partir de Python.

Implantation d'une démo (champs / série)

Une démo qui illustre l'utilisation des séries et des champs à partir d'un script Python a été mise en place. Cette démo a pour but de démontrer le concept des séries, et celui d'itérateur sur un champ ou une série.

Mise en place d'une version alpha 0.xx de Modeleur

Afin d'illustrer les progrès et les étapes accomplies dans le cadre du projet Modeleur 2.0, une version alpha de Modeleur a été assemblée. Celle-ci démontre le lien avec la base de données relationnelle, l'affichage avec VTK, le module de script, les macros, la gestion des menus et le concept de plug-in.

Recherche d'une librairie

Dans le cadre de la partie sur les itérateurs spatiaux et les régions, une recherche a été menée pour trouver la meilleure librairie externe pour offrir un support géométrique. Les candidats étaient GEOS et Terralib. La librairie retenue a été Terralib (www.terralib.org), car elle offrait toutes les fonctionnalités recherchées en plus d'offrir une ouverture vers les GIS.

Campagne de terrain

Une campagne de terrain a été menée afin de recueillir des données sur la rivière Chaudière. La municipalité de Sainte-Marie a contribué à cette campagne en rendant disponible une équipe de pompiers.

Courbes de dommage générique

Dans le cadre de l'aspect évaluation de dommages, une étude a été entreprise afin de déterminer des courbes de dommages génériques.

Décembre 2003**Rapport de développement sur les champs et les séries**

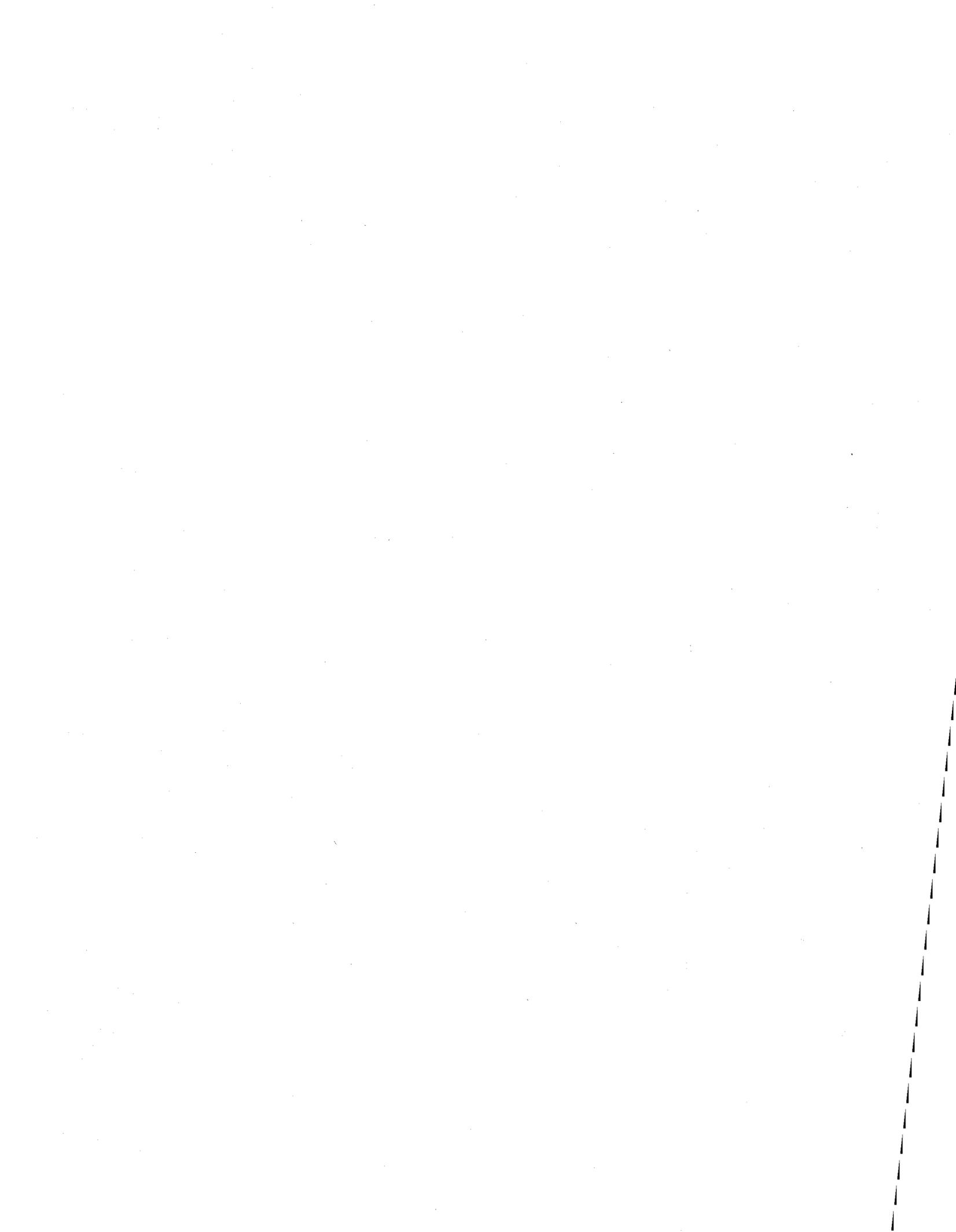
Un rapport résumant les principaux sujets reliés au dossier des champs et série a été rédigé.

Rapport de développement sur le module de script

Un rapport résumant les principaux sujets reliés au dossier du module de script a été rédigé.

Rapport d'étape #1

Le rapport d'étape #1 a été rédigé. Celui-ci inclut une section sur l'analyse préliminaire, une section sur les spécifications fonctionnelles, une section sur les tâches accomplies à chaque mois de l'année 2003, une section sur les ressources humaines et une section sur la planification du projet.



5 Ressources humaines

Les personnes suivantes ont été appelées à travailler sur le projet SEGRI au cours de l'année 2003. La contribution de chacune de ces personnes est détaillée dans les prochaines pages.

Yves Secretan

M. Secretan, professeur à l'INRS-ETE, est responsable du projet SEGRI - volet informatique. Il a participé à la planification du projet et au recrutement de stagiaires. Il a travaillé à l'encadrement de ceux-ci et a participé à toutes les réunions de design et d'analyse tenues au cours des stages.

En plus d'offrir un support aux stagiaires, M. Secretan a effectué diverses tâches de développement logiciel. Il a développé H2D2, la version 2 des logiciels Hydrosim et Dispersim. Il a travaillé au prototype d'environnement de compilation automatisé avec Boost-Jam. Il a travaillé à la mise en place de la mécanique des macros en Boost-Python / Pyste. Il a travaillé au développement du bus logiciel. Il a développé une macro de compilation et une macro d'import / export. Il a installé et configuré CVS et Apache.

M. Secretan a travaillé à la rédaction des spécifications fonctionnelles, et à la révision de nombreux rapports ou documents.

Michel Leclerc

M. Leclerc, professeur à l'INRS-ETE, est responsable du projet SEGRI – volet application sur la rivière Chaudière. Il a travaillé à l'encadrement du stagiaire Renaud le Boulleur de Courlon, et l'a aidé à mener à terme son mandat.

Paul Boudreau

M. Boudreau, agent de recherche, a travaillé à l'encadrement du stagiaire Renaud le Boulleur de Courlon, et l'a aidé à maîtriser le logiciel Modeleur. M. Boudreau a également été le premier à donner ses commentaires sur la version préliminaire des spécifications fonctionnelles.

Eric Larouche

M. Larouche, ingénieur en informatique, a comme tâche d'agir en tant que développeur senior au sein du groupe. Il a participé à la planification du projet et au recrutement de stagiaires. Il a travaillé à l'encadrement de ceux-ci et a participé à toutes les réunions de design et d'analyse tenues au cours des stages.

En plus d'offrir un support aux stagiaires, M. Larouche a effectué diverses tâches de développement logiciel. Il a développé le module de gestion de données. Il a mis en place une mécanique de plug-in permettant la gestion de menus. Il a monté une version alpha de Modeleur illustrant les progrès accomplis en cours d'année. Il a mis en place le plan de back up.

M. Larouche a travaillé à la rédaction des spécifications fonctionnelles. Il a également travaillé à la rédaction / révision de plusieurs documents d'analyse, documents de génie logiciel ou de rapports.

Renaud le Boulleur de Courlon

M. le Boulleur de Courlon, étudiant à l'Institut National Polytechnique de Toulouse (INP ENSEEIHT), a travaillé sur le projet SEGRI à titre de stagiaire en génie mécanique. Son stage s'est effectué du 14 février au 14 juin 2003.

M. le Boulleur de Courlon a travaillé à la cueillette des données sur la rivière Chaudière. Il a ensuite monté le Modèle Numérique de Terrain (MNT), construit un maillage et effectué des simulations.

Francis Larrivée

M. Larrivée, étudiant au Cégep de François-Xavier-Garneau, a travaillé sur le projet SEGRI à titre de stagiaire technicien en informatique. Son stage s'est effectué du 3 mars au 30 mai 2003.

M. Larrivée a travaillé à la recherche et à l'évaluation de solutions afin de mettre en place un système de compilation automatisé. M. Larrivée a également accompli diverses tâches d'installation / configuration de machines reliées à la mise et à jour du parc informatique.

Dominic Richard

M. Richard, étudiant au Cégep de Lévis-Lauzon, a travaillé sur le projet SEGRI à titre de stagiaire technicien en informatique. Son stage s'est effectué du 31 mars au 11 juillet 2003.

M. Richard a travaillé à la migration de certaines parties du code de Modeleur 1.0 vers des compilateurs plus récents. Avec ce code nettoyé, M. Richard a construit un certain nombre de modules de base qui sont réutilisés dans Modeleur 2.0.

Samuel Ouellet

M. Ouellet, étudiant à l'Université de Sherbrooke, a travaillé sur le projet SEGRI à titre de stagiaire en génie informatique. Son stage s'est effectué du 5 mai au 22 août 2003.

M. Ouellet s'est vu confié un dossier traitant de la base de données relationnelle. Il a évalué des outils de design et des engins de base de données. Il a ensuite effectué une recommandation. Il a mené le design des tables de la base de données relationnelle. Il a installé PostgreSQL / PostGIS et a rédigé un document explicatif. Il a démarré un projet de traducteur, dans lequel il a implanté la traduction des maillages. Enfin, il a rédigé un rapport décrivant en détails les tâches réalisées.

Maxime Derenne

M. Derenne, étudiant à l'Université de Sherbrooke, a travaillé sur le projet SEGRI à titre de stagiaire en informatique. Son stage s'est effectué du 5 mai au 22 août 2003.

M. Derenne s'est vu confié un dossier traitant de la visualisation. Il a évalué des bibliothèques graphiques et a effectué une recommandation. Il a mené le design de l'architecture du module de visualisation, qu'il a ensuite implémenté. Il a construit un prototype en Visual Basic qui intègre le module de visualisation et le module de gestion de données. Enfin, il a rédigé un rapport décrivant en détails les tâches réalisées.

Sébastien Bédard

M. Bédard, étudiant à l'Université Laval, a travaillé sur le projet SEGRI à titre de stagiaire en génie mécanique. Son stage s'est effectué du 5 mai au 15 août 2003.

M. Bédard a travaillé à la conception d'un banc de test pour le logiciel H2D2. Pendant cette phase, il a également apporté différentes corrections au logiciel. M. Bédard a travaillé à la validation du modèle de température d'H2D2. M. Bédard a construit une ébauche de manuel utilisateur pour H2D2. Enfin, il a rédigé un rapport décrivant en détails les tâches réalisées.

Laurent Bonnifait

M. Bonnifait, étudiant à la maîtrise à l'INRS-ETE, travaille actuellement sur le projet SEGRI. Son implication dans le projet a commencé au mois de juin 2003.

M. Bonnifait a effectué une revue de la littérature disponible au sujet de la ville de Ste-Marie de Beauce. Il a procédé à l'acquisition d'un échosondeur et à sa mise au point. Il a mené une campagne de terrain visant à recueillir des données sur la rivière Chaudière. M. Bonnifait a également effectué une analyse visant à déterminer une courbe de dommage générique.

Kevin Solinski

M. Solinski, étudiant à l'ENIC Telecom (France), travaille actuellement sur le projet SEGRI à titre de stagiaire en génie informatique. Son stage a débuté le 4 août 2003 et se terminera le 30 janvier 2004.

M. Solinski s'est vu confié un dossier traitant du module de script. M. Solinski a effectué une recherche afin de trouver un langage de script pour Modeleur 2.0. M. Solinski a travaillé à l'élaboration des spécifications fonctionnelles pour le module de script. Il a également travaillé à l'implémentation de celui-ci. Enfin, il a rédigé un rapport décrivant en détails les tâches réalisées.

Olivier Kaczor

M. Kaczor, étudiant à l'Université de Sherbrooke, a travaillé sur le projet SEGRI à titre de stagiaire en informatique. Son stage s'est effectué du 2 septembre au 19 décembre 2003.

M. Kaczor s'est vu confié un mandat traitant des champs et des séries. M. Kaczor a amélioré le traducteur pour y ajouter la traduction des champs. M. Kaczor a mené le design des champs, des séries et des itérateurs spatiaux, design qu'il a ensuite implémenté. M. Kaczor a également effectué une recherche pour trouver une librairie géométrique. Enfin, il a rédigé un rapport décrivant en détails les tâches réalisées.

Maude Giasson

Mme Giasson, étudiante à l'Université Laval, travaillera sur le projet SEGRI à titre de stagiaire en informatique. Son stage s'effectuera du 5 janvier au 16 avril 2004.

Cédric Caron

M. Caron, étudiant à l'Université de Sherbrooke, travaillera sur le projet SEGRI à titre de stagiaire en informatique. Son stage s'effectuera du 5 janvier au 16 avril 2004.

Benjamin Behaghel

M. Behaghel est étudiant à l'École Pour l'Informatique et les Techniques Avancées (EPITA) en France. Il travaillera sur le projet SEGRI à titre de stagiaire en informatique. Son stage s'effectuera du 12 janvier au 16 juillet 2004.

6 Planification

La section suivante présente le plan de développement pour toute la durée du projet SEGRI (2003-2005). Il couvre donc l'année écoulée, et une planification pour les deux années à venir. Dans un premier temps, les tâches-clés sont décrites brièvement. Dans un second temps, la planification du projet est présentée à l'aide d'un diagramme.

Les tâches sont découpées en tranches de quatre mois de manière à pouvoir être effectuées dans le cadre de stages. Les livrables logiciels sont prévus pour la fin 2004 et la fin 2005, et sont marqués à l'aide de jalons dans le diagramme.

6.1 Tâches

Environnement de développement

Cette tâche a pour but de mettre en place les mécanismes pour assurer le développement de Modeleur 2 dans les règles de l'art du génie logiciel. Cette tâche inclut une composante rédaction de documents de génie logiciel (normes de programmation, analyses, rapports, recommandations d'outils). Elle inclut également la mise en place d'un système automatisé de compilation et de tests.

Cette tâche est complétée à environ 80%. Le rapport détaillant le travail effectué sur cette tâche est disponible dans le document Rapports de recherche 2003.

H2D2

Cette tâche a pour but d'effectuer la migration d'Hydrosim / Dispersim. Cette tâche inclut également la parallélisation du logiciel afin de pouvoir distribuer la charge de calcul, et d'ainsi diminuer les temps de calcul.

Cette tâche est complétée à environ 35%.

Application Chaudière-MNT et maillage

Cette tâche a pour but de construire le Modèle Numérique de Terrain de la rivière Chaudière à l'aide de données provenant de différentes sources. Elle inclut également la conception d'un maillage adapté au cas de la rivière Chaudière.

Cette tâche est complétée à environ 75%. Le rapport détaillant le travail effectué sur cette tâche est disponible dans le document Rapports de recherche 2003.

Module de visualisation

Cette tâche a pour but de permettre la visualisation des maillages, des isolignes, des isosurfaces, des vecteurs, des semis de points, des nœuds, des profils en long et des lignes de courant.

Cette tâche est complétée à environ 50%. Le rapport détaillant le travail effectué sur cette tâche est disponible dans le document Rapports de recherche 2003.

Module de base de données

Cette tâche a pour but d'intégrer une base de données relationnelle à Modeleur 2, et de fournir des mécanismes pour que les autres modules connectés au bus logiciel puissent créer, charger, sauver et détruire des données.

Cette tâche est complétée à environ 65%. Le rapport détaillant le travail effectué sur cette tâche est disponible dans le document Rapports de recherche 2003.

Champs-série

Cette tâche a pour but de définir et implanter les champs et les séries, structures de données qui seront à la base de Modeleur 2.

Cette tâche est complétée à environ 80%. Le rapport détaillant le travail effectué sur cette tâche est disponible dans le document Rapports de recherche 2003.

Module de script

Cette tâche a pour but d'intégrer un langage interprété externe à Modeleur 2 afin de remplacer la calculatrice, de permettre l'ajout de plug-in et de mettre en place un système de macro.

Cette tâche est complétée à environ 50%. Le rapport détaillant le travail effectué sur cette tâche est disponible dans le document Rapports de recherche 2003.

Spécifications fonctionnelles

Cette tâche a pour but de rencontrer les utilisateurs / partenaires et d'ensuite rédiger les spécifications fonctionnelles de Modeleur 2.

Cette tâche est complétée à 50%.

Module GUI

Cette tâche a pour but d'intégrer à Modeleur 2 les mécanismes permettant l'ajout de plugin.

Cette tâche est complétée à 90%.

Module de maillage

Cette tâche a pour but d'intégrer à Modeleur 2 l'activité maillage. La notion de maillage adaptatif, de même que la génération de transects pour les méthodes 1D sont inclus dans cette activité.

Module de modèle numérique de terrain (MNT)

Cette tâche a pour but d'intégrer à Modeleur 2 une activité permettant de définir le Modèle Numérique de Terrain. Les partitions génériques sont incluses dans cette activité.

Module de validation des données

Cette tâche a pour but d'intégrer à Modeleur 2 l'activité permettant d'effectuer la validation des données. Cette tâche inclut la mise en place de mécanismes permettant l'ajout d'algorithmes de filtrage externe.

Installation et distribution

Cette tâche a pour but de définir et de mettre en place les mécanismes afin de pouvoir effectuer la distribution du logiciel par le web.

Module de gestion de données

Cette tâche a pour but d'intégrer à Modeleur 2 l'activité permettant de gérer l'import et export de données. Cette activité inclut également l'équivalent du gestionnaire d'entités de Modeleur 1.

Outils d'analyse

Cette tâche a pour but d'intégrer à Modeleur 2 une gamme d'outils d'analyse. Parmi ceux-ci, on retrouvera des outils de ligne, de surface, de projection et de calcul de débit.

Module de simulation

Cette tâche a pour but d'intégrer à Modeleur 2 une activité simulation. La connexion entre Modeleur et les simulateurs (H2D2, par exemple) sera faite à cette étape.

Aide et tutoriel

Cette tâche a pour but de définir et de mettre en place les mécanismes permettant de gérer l'aide et les tutoriels au niveau de chaque plug-in. Elle couvre également la rédaction du contenu de l'aide et des tutoriels.

Modèle 1 D ½

Cette tâche a pour but de mettre en place un modèle de simulation hydrodynamique 1D ½ dans H2D2.

Module de publication

Cette tâche a pour but d'intégrer à Modeleur 2 les aspects suivants : génération de rapports, habillage d'écran et montage de simulation.

Application Chaudière-Simulation

Cette tâche a pour but d'étudier le cas de la rivière Chaudière à l'aide de Modeleur 2.

Évaluation des dommages

Cette tâche a pour but d'intégrer à Modeleur 2 une composante « évaluation des dommages ».

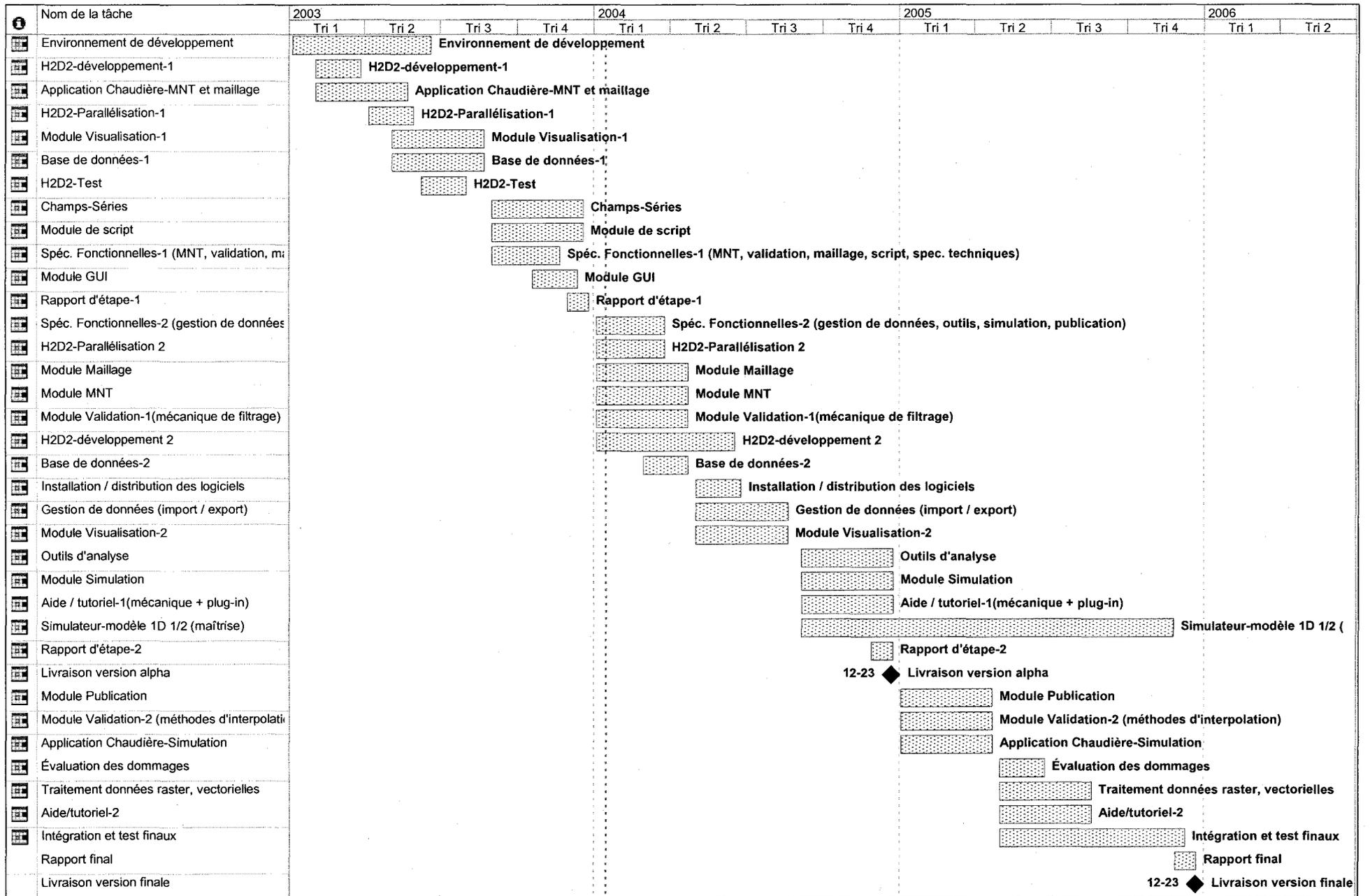
Traitement des données raster, vectorielles

Cette tâche a pour but d'intégrer à Modeleur 2 un support pour les données raster et vectorielles.

Intégration et test finaux

Cette tâche a pour but de s'assurer que toutes les composantes développées dans le cadre du projet s'intègrent bien. Des tests au niveau système seront développés afin de garantir un logiciel fiable.

6.2 Diagramme des tâches



Projet : calendrier 2004-2005 Date : Jeu Jan 15	Tâche		Jalon		Tâches externes	
	Fractionnement		Récapitulative		Jalons externes	
	Avancement		Récapitulative de projet		Échéance	

7 Conclusion

Le projet SEGRI offre des défis très stimulants au niveau du développement logiciel. De ce fait, il s'avère qu'il est facile de motiver les étudiants car les stages offerts sont riches et variés. Ces stages constituent une occasion unique pour eux de travailler sur un projet logiciel d'envergure et de se voir confier une gamme importante de responsabilités.

En ce qui a trait à l'aspect planification, le projet avance tel que prévu. Les décisions prises en cours d'année nous donne une plate forme solide sur laquelle construire pour les années à venir. Au cours de l'année 2004, des fonctionnalités additionnelles seront peu à peu ajoutées au noyau principal afin de pouvoir livrer une version alpha de Modeleur 2.0 à la fin de l'année.

Les contacts avec les partenaires externes se sont avérés fort productifs. L'expertise recueillie auprès d'eux a servi à articuler plusieurs sections du logiciel. Leurs commentaires et suggestions ont grandement aidé à la rédaction de la première version des spécifications fonctionnelles de Modeleur 2.0.