

Méthodologie de montage d'un SIG-3D par des Etudiants

■ François BOUILLÉ

MOTS-CLÉS

JAVA-3D, méthodologie de conception, projet étudiant, réalité virtuelle, SIG-4D, simulation, structuration HBDS

L'objectif de cette brève communication, d'ordre méthodologique, est de présenter la réalisation d'un modèle 3-D destiné à des Etudiants (Bac+5), prenant en compte le temps, d'où le nom de "projet SIG-4D", modèle permettant l'immersion de l'utilisateur, selon les techniques maintenant classiques de réalité virtuelle. Ce projet est avant tout un projet d'enseignement, visant à former des spécialistes en Géomatique et SIG, capables aussi bien de prendre en charge un système informatique complet, notamment dans le domaine des SIG, que d'appliquer leurs compétences à un domaine précis (la gestion d'un réseau par exemple), ou enfin de développer de nouvelles applications de pointe utilisant pleinement la 3D.

Ce projet a été intégré depuis 1998 dans l'enseignement du DESS double compétence IAST¹, cursus devenu, dans le cadre de la réforme LMD, un Master2 intitulé "Informatique Appliquée aux SIG", co-habilité UPMC-ENSG.

Il est important dès à présent de préciser que le développement est limité à une durée de six semaines, en fin d'année, et qu'aucun logiciel SIG n'est utilisé. Seul l'accès à un environnement de développement est fourni, en l'occurrence du JAVA 3D.

Les objectifs du projet SIG4D

Deux objectifs sont à la base de ce projet :

- réaliser un SIG immersif,
- acquérir et maîtriser une méthodologie de développement sur des SIG innovants.

La modélisation au sein du projet "SIG-4D" prend en charge les composants suivants :

- le MNT et l'orthophoto,

- un nombre quelconque de réseaux aériens, de surface, ou souterrains,
- le cadastre et le bâti,
- des photographies de mobiliers,

et adjoint des composants articulés mobiles au sein du modèle, avec la possibilité de synchroniser leurs déplacements (par exemple des trains), d'où l'intervention de la 4^e dimension : le temps.

Une fois composé, le modèle complet, appelé ici "univers" doit pouvoir être manipulé selon six degrés de liberté (translations et rotations). Enfin, le géomaticien doit pouvoir s'immerger au sein du modèle :

- soit pour un parcours au sol : déplacement d'un véhicule piloté en tout-terrain, ou déplacement asservi le long d'un graphe (par exemple le graphe des routes),
- soit pour un vol tactique, par exemple un parcours enchaînant des talwegs, automatique ou piloté.

Pour réaliser ce projet, une méthodologie a été définie, décomposant la conception en plusieurs étapes, chacune reposant sur l'emploi d'outils théoriques expérimentés de longue date. La première étape consiste à structurer, la deuxième à définir une architecture logicielle allant du stockage jusqu'aux IHM³, la troisième à effectuer les développements algorithmiques, la quatrième, très brève, à programmer et tester. Ces étapes sont rigoureusement distinctes. Un retour est possible en cas d'erreur.

L'étape de structuration

Elle fait appel à un modèle, une méthode, un catalogue de structures préexistantes, et bien évidemment à des composants géographiques à organiser.

■ L'outil de modélisation

La modélisation du monde réel pris en compte dans "l'univers" est fondée sur le modèle HBDS³. Celui-ci repose sur six TAD⁴ persistants élémentaires, complétés par diverses extensions :

- des ensembles appelés "classes",

(1) IAST : Informatique Appliquée aux Sciences de la Terre.

(2) IHM : Interface Homme-Machine.

(3) HBDS : Hypergraph-Based Data Structure (Bouillé, 1977)

(4) TAD : Type Abstrait de Donnée, traduction de Abstract Data Type, quoiqu'il s'agisse ici de manipuler des choses fort concrètes...

- portant un nombre quelconque d'"attributs de classe", représentant leurs propriétés,
- présentant entre eux un nombre quelconque de "liens entre classes", porteurs de relations potentielles, que leurs éléments peuvent éventuellement vérifier (mais non obligatoirement),
- contenant un nombre quelconque (parfois très grand) d'éléments appelés "objets",
- ces objets possèdent des "attributs d'objets" correspondant à leurs attributs de classe respectifs,
- présentent éventuellement des "liens entre objets" qui sont les réalisations effectives de leurs liens entre classes.

Un attribut n'est pas "atomique" (au sens du modèle relationnel), mais peut être d'un genre autre qu'un simple scalaire : vecteur, matrice, tenseur, liste, groupe... Le typage offert pour les attributs est assez riche : numérique (entier, flottant, complexe, quaternion, bi-quaternion, octonion, rationnel), booléen ou chaîne de caractères. Un attribut numérique peut être muni d'une unité, permettant de vérifier les équations aux dimensions. On peut lui appliquer diverses conditions et contraintes.

Ces six TAD de base sont complétés par des "hyperclasses", qui sont des regroupements de classes, des "hyperattributs", des "hyperliens", et par d'autres TAD obtenus par auto-extensibilité, tels que les "prototypes", que nous présentons plus loin.

La méthode de modélisation

Ayant inventorié tous les composants pouvant intervenir dans "l'univers", il s'agit de composer celui-ci. Pour tout concept, l'on cherche s'il s'agit d'un ensemble (ce sera donc une classe), d'une propriété (donc un attribut), d'une relation (donc un lien) ou d'un élément (donc un objet).

Pour toute classe, l'on cherche tous les attributs à prendre en compte, tous les liens avec d'autres classes. L'on cherche de même quelles sont les classes qui sont regroupables en plus grands ensembles constituant des hyperclasses. Tout composant de "l'univers" à modéliser est donc représenté par un TAD.

L'utilisation de prototypes

Un prototype est un modèle de structure représentant un type de phénomène que l'on retrouve applicable en diverses occasions. On le définit alors une fois pour toutes, et l'on en utilise une copie spécifique dédiée à chaque application. C'est un peu du "légo" pour géomaticiens...

L'on dispose déjà d'un certain nombre de prototypes réutilisables : graphe, réseau, maillage, isoligne, polyèdre, etc...

Tout réseau repose d'abord sur le concept de graphe, constitué de sommets et d'arcs, auxquels nous adjoignons des domaines représentant les entités surfaciques. Ces classes élémentaires présentent divers liens qui assurent la topologie du phénomène géographique. Un arc peut supporter des points annexes qui modifient son modelé et le divisent en

segments successifs. Les attributs que sont les coordonnées des sommets et celles des points annexes fournissent aisément la métrique du graphe et permettent d'obtenir la longueur d'un arc, le périmètre et la surface d'un domaine.

Un réseau est d'abord un graphe, complété par des réalisations matérielles modélisées comme suit :

- deux hyperclasses de mobiliers associées respectivement aux sommets et aux points annexes,
- une hyperclasse de composants linéaires matérialisant les segments.

Les segments ne sont toutefois pas toujours homogènes ; par exemple, ils peuvent être constitués de tubes de nature différente, et cette hétérogénéité doit être prise en compte en localisant le point de changement ; il en est de même de la localisation de soudures, ou de mobiliers dédiés : protection cathodique, capteurs divers, boîtiers de relais, etc...

Nous prenons donc en compte ces "points supplémentaires" et les "tronçons" (sous-segments qui en découlent) c'est-à-dire deux nouvelles classes. La figure 1 montre les diverses classes d'un réseau, certaines figurant en abrégé : S, A, D, PA, PC (pour "sommet", "arc", "domaine", "point annexe", "point complémentaire"), ainsi que diverses hyperclasses : CL, MPA, MPC, MS (pour "composant linéaire", "mobilier de point annexe", "mobilier de point complémentaire", "mobilier de sommet"). Les diverses hyperclasses de mobilier sont elles-mêmes regroupées au sein d'une hyperclasse commune,

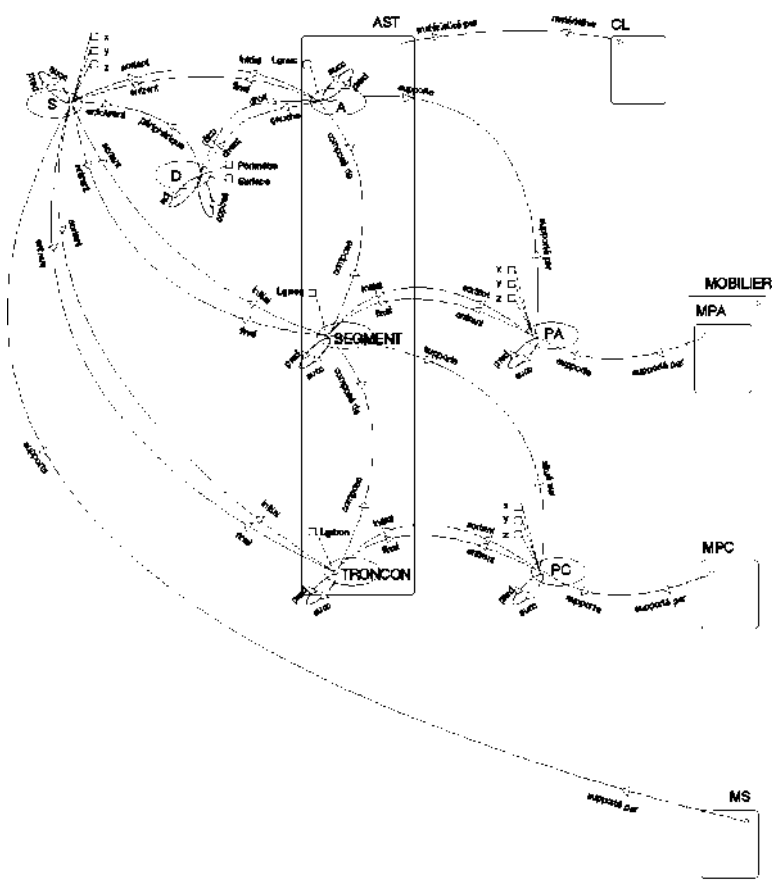


Figure 1. Le prototype réseau.

de même que l'hyperclasse AST regroupe les classes "arc", "segment" et "tronçon".

L'on peut ainsi, avec ce prototype, modéliser n'importe quel réseau, à n'importe quelle échelle.

Afin de contrôler les interactions entre réseaux, dans le cadre de la 3D, il faut tenir compte des superpositions et intersections ; l'on montre aisément que deux réseaux représentent neuf cas de superposition, ce qui conduit à neuf classes, et autant pour les cas d'intersection. Avec plusieurs réseaux, le nombre des classes augmente notablement, mais le fait de ranger les divers cas dans des classes distinctes ne consomme pas plus de place ; à l'inverse, ce rangement raccourcit considérablement le temps d'accès à l'information.

Par ailleurs, il faut pouvoir localiser instantanément tout ce qui peut être présent dans une zone de "l'univers". L'on applique donc un maillage rectangulaire (ou carré) auto-adaptatif, intervenant au sein de chaque dalle. Chaque maille doit être mise en relation avec tous les composants se trouvant au sein de celle-ci (sommets, points annexes, points complémentaires) ou la traversant en tout ou partie (arcs, segments, tronçons) ; l'on relie de même la maille aux nombreux points de superposition et d'intersection. La classe maille et ses caractéristiques sont prévues à l'étape de structuration. Les objets des classes de superpositions, intersections, et les liens avec la classe maille ne seront construits qu'à l'étape algorithmique.

■ Les composants du modèle

Ce sont, entre autres, un MNT, une orthophoto, des réseaux, du bâti, des photos de mobiliers. Le MNT peut être de grande taille. On lui associe un ensemble orthophotographique. Le MNT est divisé en grandes dalles rectangulaires de même taille.

A l'intérieur de chaque dalle, tous les quadrilatères gauches sont systématiquement scindés en deux triangles. L'orthophoto couvrant chaque dalle (ou la combinaison d'orthophotos) est plaquée sur le MNT, et chaque triangle reçoit ainsi une texture.

Divers réseaux sont alors pris en compte :

- des réseaux de surface : par exemple les routes ; ces réseaux sont plaqués directement sur le MNT, en superposition avec le fragment d'orthophoto,
- des réseaux souterrains : ils sont généralement représentés par un enchaînement de cylindres, une couleur spécifique caractérisant le réseau ; les jonctions sont réalisées par des sphères de diamètre légèrement supérieur à celui des cylindres.
- des réseaux aériens : par exemple les lignes à haute tension ; les pylônes sont réalisés en 3D et les lignes sont matérialisées non par des segments, mais bien sûr par des chaînettes. Un réseau peut évidemment changer d'état, et partiellement souterrain, peut devenir de surface ou aérien, ou vice-versa.

Architecture du modèle

Le système est développé selon une architecture multicouche. La couche la plus interne est la base de données, la plus externe étant l'IHM. Les interfaces sont définies en précisant à chaque fois, le nom de la primitive, les spécifications des paramètres, les messages de retour qui permettent de contrôler et de "rendre compte". Au sein de chaque couche, les divers modules sont définis, eux aussi avec leurs noms, leurs paramètres spécifiés et leurs messages. Un graphe des appels de modules est géré au sein de chaque couche. Au total, les modules sont quelques centaines. Les effectifs Etudiants assurant ce projet sont montés certaines années jusqu'à 86 personnes, les équipes étant constituées de trinômes devant s'interfacer. Il convient donc de ne pas aboutir à un développement anarchique. Une grande discipline de travail est imposée. Une équipe de coordination gère les noms des modules, le contrôle des paramètres, et surtout la définition des messages échangés entre les modules.

Une attention particulière est prêtée à bien séparer les modules de calcul des modules d'affichage. Dans la mesure où divers thèmes géographiques viennent se connecter facultativement, il est nécessaire de développer autant d'IHM. Les divers thèmes doivent pouvoir cohabiter. Cette possibilité implique de ne pas récrire des combinaisons d'IHM, mais de connecter toute IHM spécifique à une IHM d'accueil. Sous la couche des IHM, l'on trouve toutes les "moulinettes" géomatiques de calculs divers. La figure 2, extrêmement simplifiée, donne une idée de l'architecture multicouche.

Lorsque l'architecture a été bien spécifiée dans le détail, l'on peut passer à l'étape algorithmique.

L'étape algorithmique

Tous les algorithmes écrits doivent être dûment commentés. Ils doivent s'insérer dans le graphe d'appel précédemment évoqué. Ils représentent quelques centaines de modules.

Nous disposons d'un langage algorithmique ADL⁵, indépendant de tout langage de programmation, de tout système opératoire et de tout matériel. Constitué d'une cinquantaine de symboles mathématiques, et muni de théorèmes d'optimisation, il permet une écriture condensée, rationnelle et très rapide. Ce langage est sans étiquette et permet donc le développement d'un code sans "zone d'ombre". Il contient tous les éléments permettant d'écrire des algorithmes aussi bien pour machines séquentielles, vectorielles, et parallèles. Il existe un excellent éditeur algorithmique, développé par un de nos anciens Etudiants, M. Boutard, et utilisé en libre service par nos actuels Etudiants.

Les algorithmes relèvent de diverses catégories, dont les objectifs résumés sont les suivants :

- prendre en charge les thèmes de base :
 - effectuer le découpage en dalles,



(5) ADL : Algorithm Description Language (Bouillé, 1977).

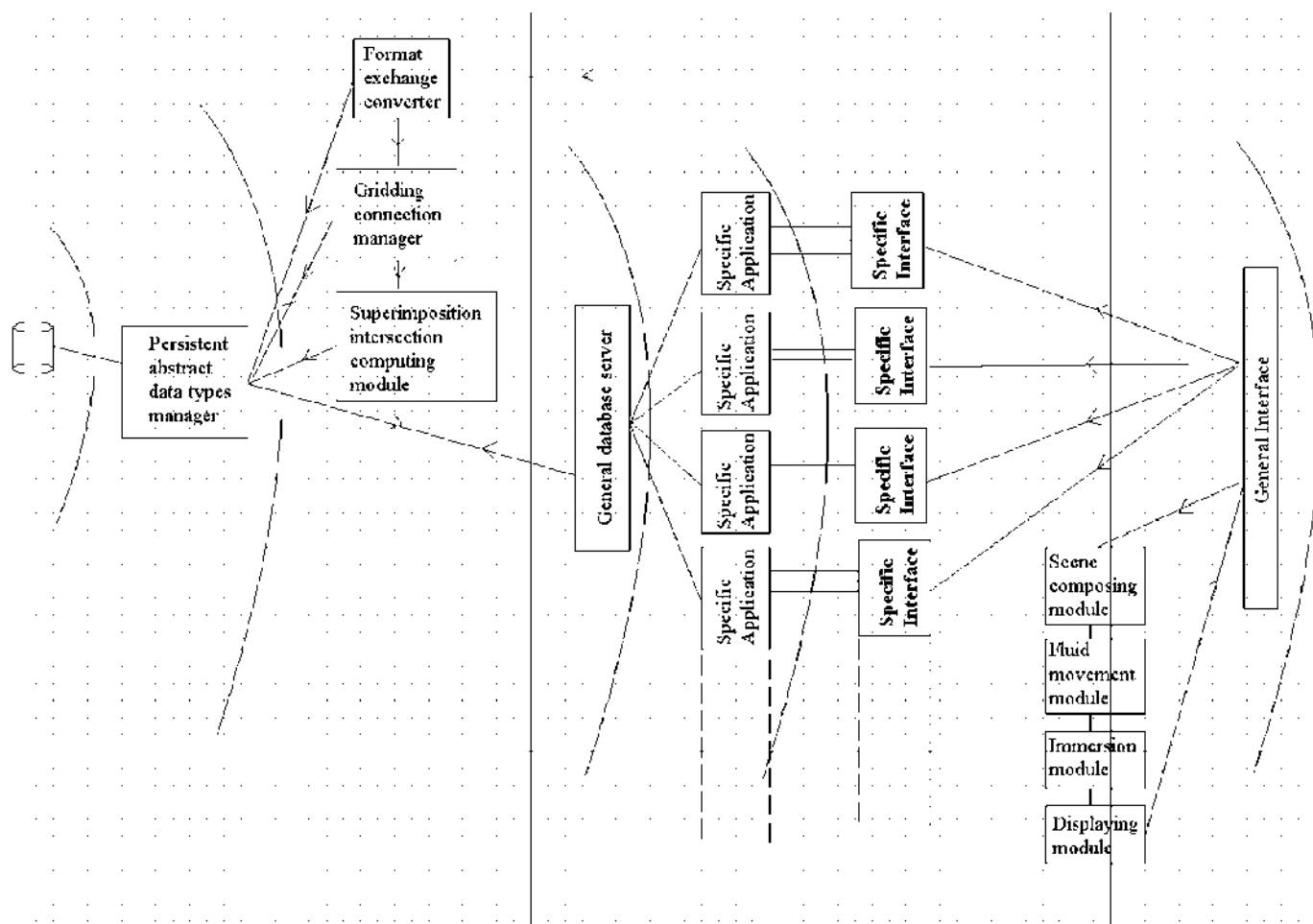


Figure 2. Architecture logicielle sommaire du projet (in Bouillé, Conf. Hong-Kong).



- trianguler le MNT, et plaquer les orthophotos sur les triangles ainsi générés,
- plaquer divers graphes de surface directement sur le MNT,
- composer des graphes souterrains sous le MNT,
- placer des graphes aériens, suspendus par du mobilier stylisé (pylônes), et fabriquer les chaînettes correspondant aux câbles,
- appliquer un maillage auto-adaptatif, au sein de chaque dalle, et relier tous les composants précédents à ce maillage,
- engranger tout ceci dans la base de données,
- gérer les interactions statiques :
 - calculer toutes les intersections et superpositions entre thèmes,
 - les relier au maillage de chaque dalle,
 - enrichir la base de données.
- préparer la gestion de mobiles incorporés dans "l'univers" :
 - écrire les primitives d'un échancier (simulation quasi-parallèle), pour les mobiles,
 - évitement de collision des mobiles,...
 - préparer le graphisme d'un ou plusieurs mobiles dont on assure l'articulation,
 - les associer à un ou des réseaux,
- préparer la gestion d'itinéraires d'immersion :
 - recherche automatique de chemins dans les talwegs,
- formalisation du déplacement d'un véhicule en tout terrain (tenir compte du dévers, etc...),
- et du déplacement asservi sur un graphe, en automatique ou en commandé,
- vers les basses couches, algorithmes de récupération des informations depuis la base de données, selon divers critères,
- en prenant en compte les mailles, les intersections et superpositions, algorithmes de composition de scène, selon une direction et un angle de vision, avec parties visibles, soit en éliminant les parties cachées, soit sans élimination de celles-ci,
- gestion de l'éclairage, de la nébulosité, etc...
- algorithme de déplacement de scène, ou de déplacement dans la scène :
 - en utilisant des matrices homogènes,
 - en passant par des quaternions, ce qui divise par deux le temps de calcul,
- affichage de la scène,
- enchaînement de dalle en dalle lors du déplacement,
- composition de chaque IHM dédiée,
- composition de l'IHM d'accueil,...

Cette liste non exhaustive donne une idée de ce travail, le plus important du projet (quatre semaines environ). A la fin de cette étape, il est encore possible de choisir le langage de programmation, les algorithmes développés étant indépendants de celui-ci.

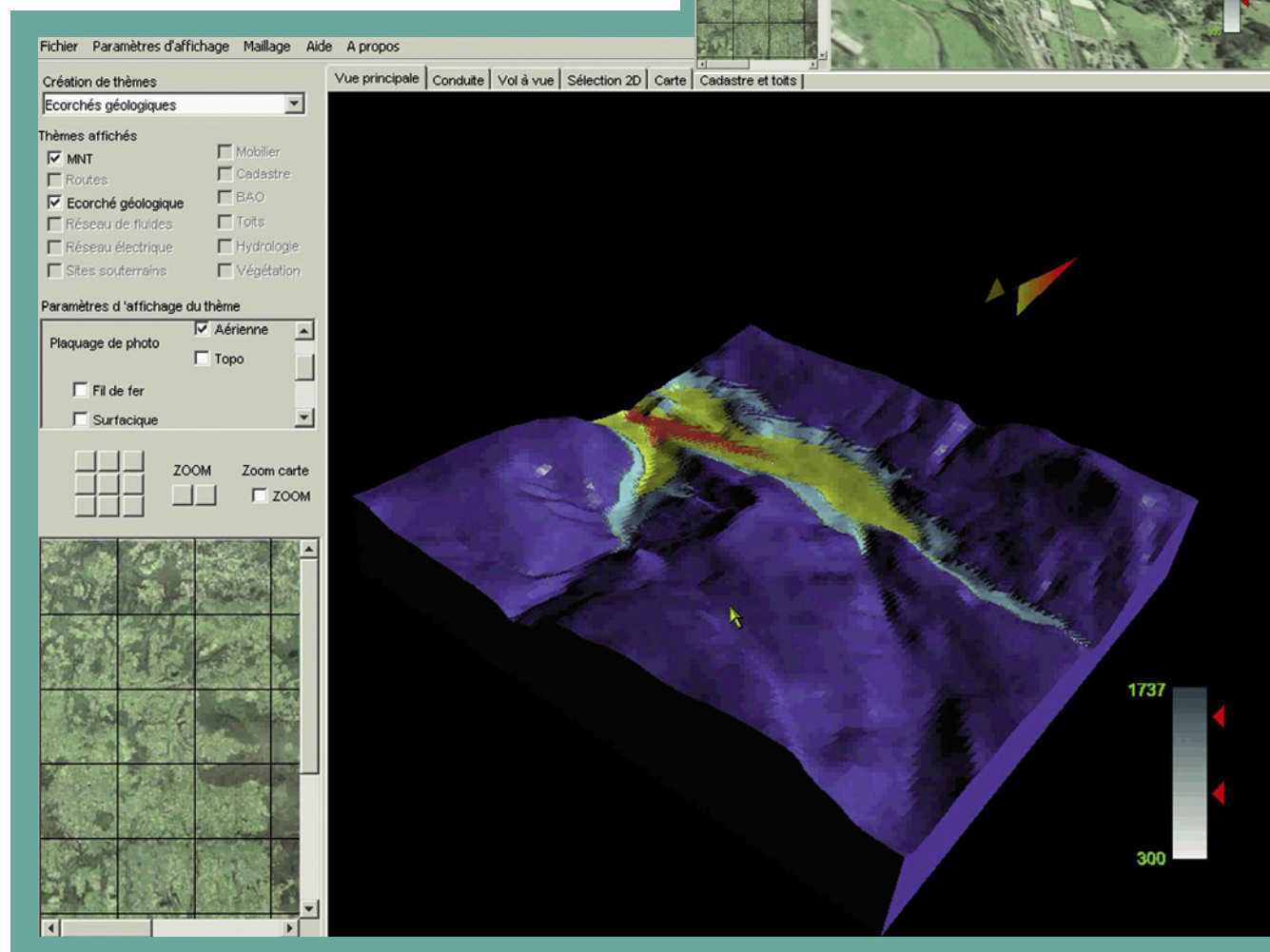
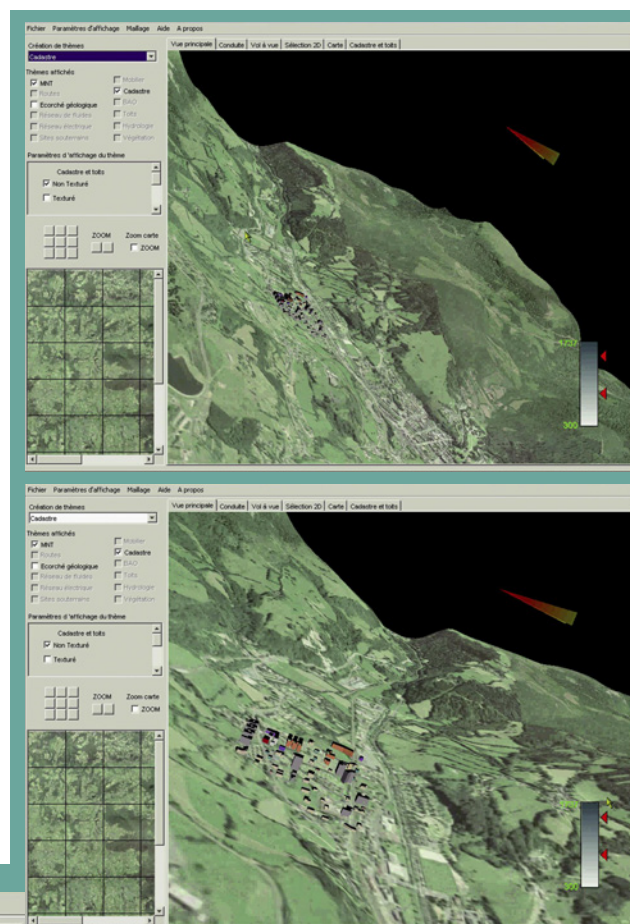
Ces algorithmes doivent pouvoir être enchaînés à l'étape de la programmation sans que l'on découvre alors, beaucoup trop tard, des incompatibilités.

L'étape de programmation

Dans la mesure où tout a été correctement pensé préalablement, cette étape est fort courte. HBDS possède un langage, ADT'81, permettant une manipulation aisée des TAD persistants. Mais le but est de traduire les algorithmes dans un langage couramment utilisé dans le milieu industriel.

L'on pourrait s'en dispenser en utilisant un traducteur automatique d'algorithme, permettant d'obtenir au choix, l'implantation du SIG en C, C++, FORTRAN 95, PASCAL, voire en un excellent langage bien structuré et performant (comme SIMULA), ou tout autre langage, éventuellement peu performant au contraire (mais utilisé alors dans cet entraînement

(6) BNF: Bacchus Naur Form.



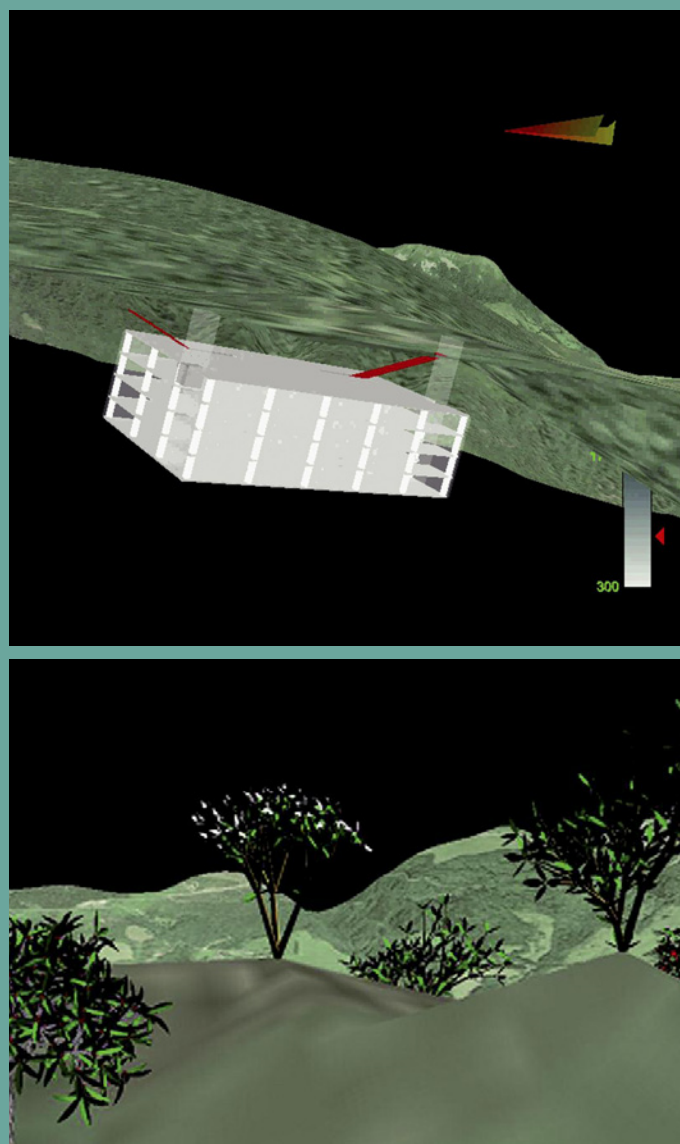
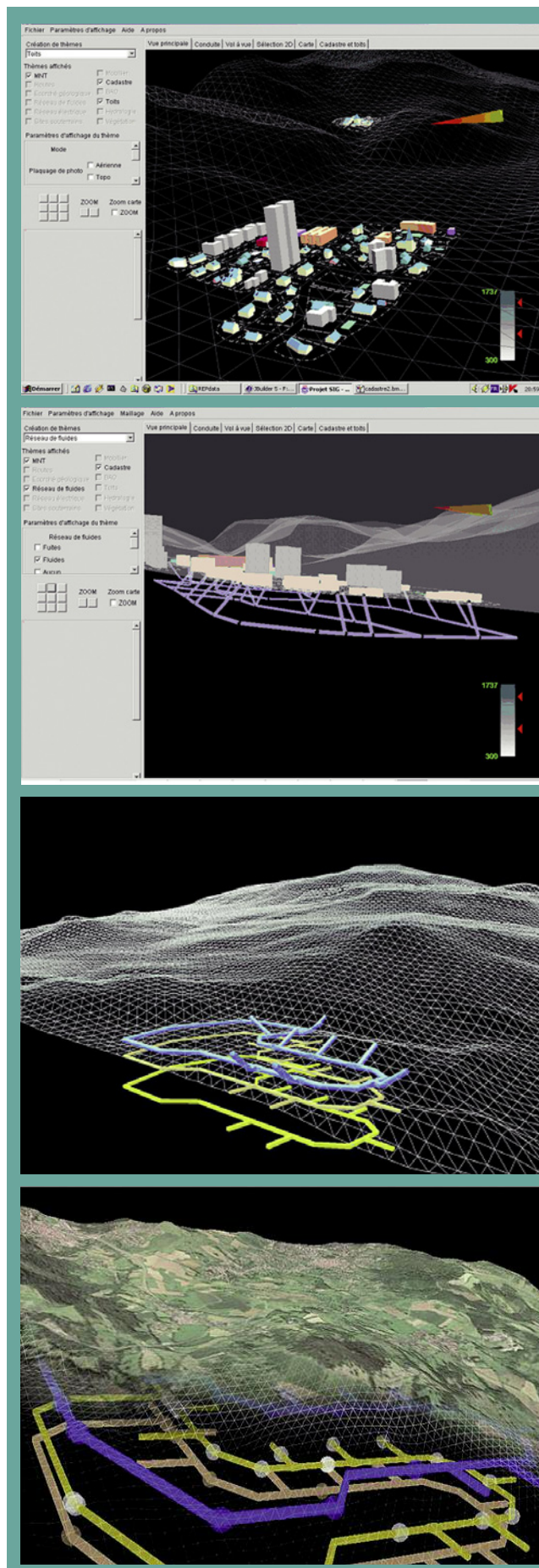


Figure 3. Quelques images des applications : promenade au-dessus du bâti, écorché géologique, bâti et réseau, réseaux souterrains, parkings souterrains, végétation, ... (travaux des Etudiants)

pour préparer les Etudiants à ce langage qui leur est demandé sur le marché du travail...).

C'est en JAVA 3D que le projet est implanté chaque année. Pourquoi ce choix ? Non que ce soit un bon langage ! Il ne repose pas sur une grammaire exprimable par une BNF⁶. Il est lourd ! Il est peu performant, beaucoup trop lent. Qu'importe, le JAVA est demandé, nos Etudiants doivent le connaître, et comme ils ont déjà utilisé Fortran 95 et C++, va pour le JAVA, version 3D évidemment. Il a un avantage non négligeable : il est très utilisé pour les applications sur le réseau.

La traduction des algorithmes en JAVA ne demande que deux ou trois jours d'efforts, et deux jours de test, pour chaque équipe, et ceci peut donc se faire en parallèle. L'interfaçage entre équipes, c'est-à-dire le branchement d'un

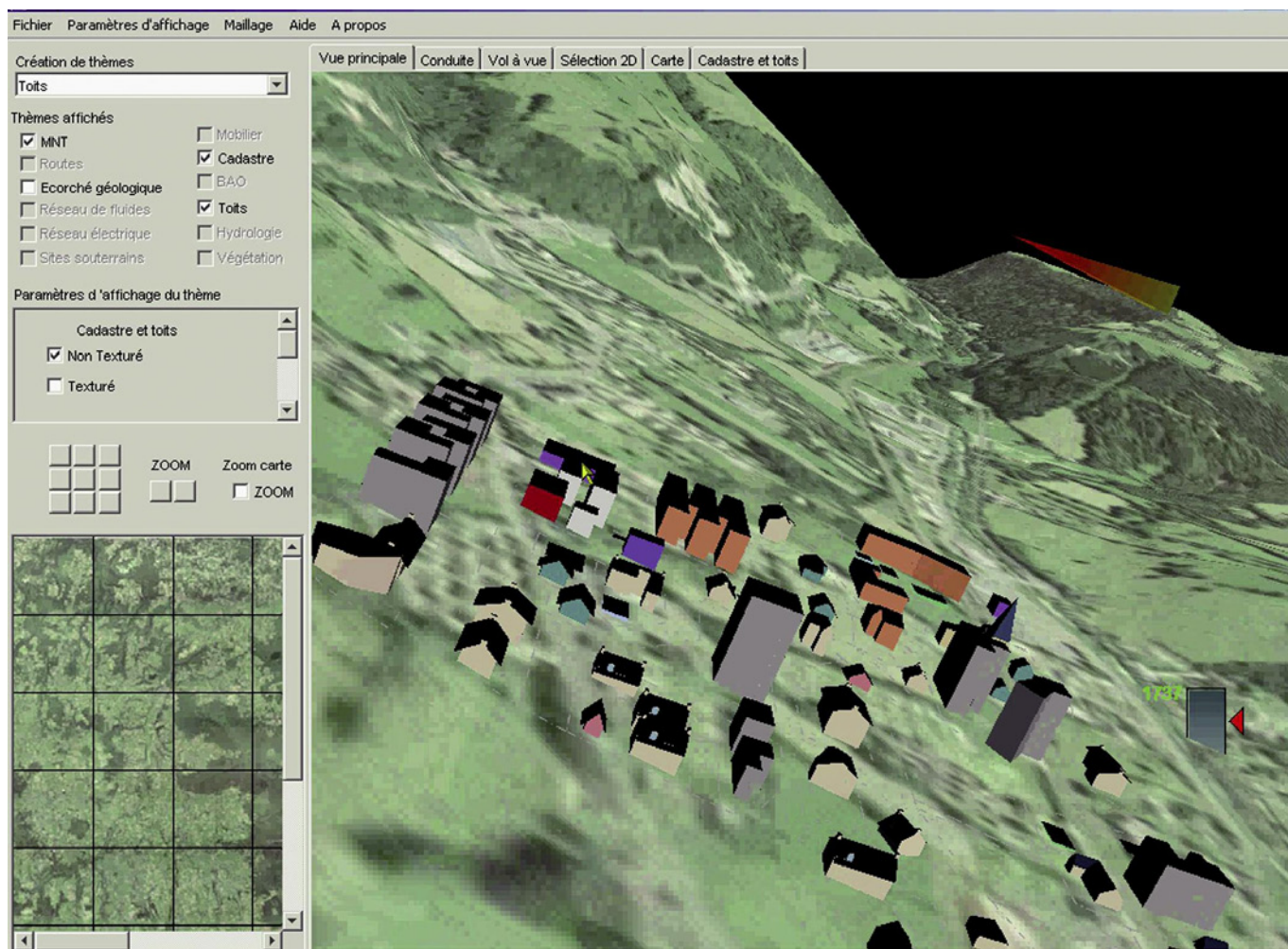


Figure 4. Plongée vers des immeubles pour un rase-motte (travaux des Eudiants).

“branch-group” sur la branche principale, demande deux heures au plus.

On remarquera que le projet pourrait tout aussi bien être retranscrit simultanément vers un tout autre langage, par exemple du C++ associé à OPEN-GL...

Le fonctionnement et les améliorations à apporter

Le “miracle” se produit le dernier jour lorsque l’ensemble fonctionne. L’on peut circuler dans “l’univers”, le plus impressionnant étant le vol tactique, en fond de talweg.

C’est d’ailleurs là que l’on découvre certains imprévus : il y a quelques années, la personne pilotant le vol en rase-motte voulait passer sous les lignes à haute tension, ce qui est logique en vol tactique ; mauvaise manœuvre, nous avons traversé un pylône ; eh bien ! l’avion a continué imperturbablement son vol, car la collision n’avait pas été envisagée dans les algorithmes !!!

Diverses applications ont été greffées : inondation d’une zone, croissance de végétaux dans “l’univers”, propagation d’un incendie (avec contre-feux et changement de direction du

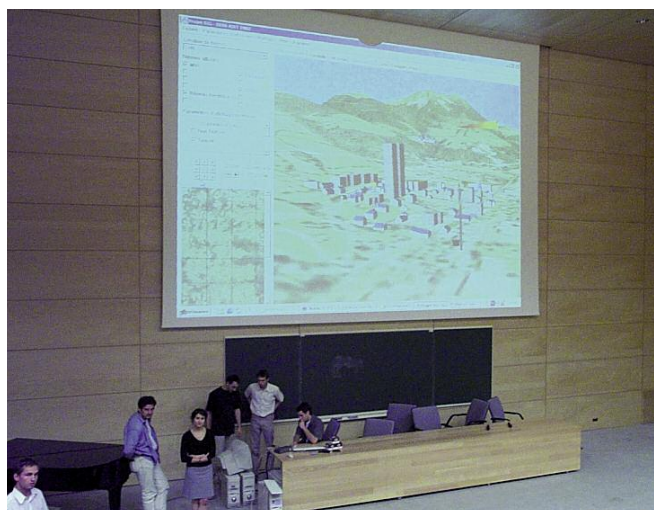


Figure 5. Projection sur grand écran le jour “j” (Photo M. Olivier Bouiri, IGN, avec son aimable autorisation).

vent), fuites de canalisations, construction de parkings souterrains avec ascenseurs en mouvement (figure 3), etc... La figure 4 montre une descente en piqué pour un rase-motte au-dessus d’un groupe de maisons.

Les régions utilisées pour construire concrètement “l’uni- ▶



vers" ont été jusqu'à présent :

- la zone de Chambéry,
- la Chaîne des Puys,

mais évidemment, toute autre région est envisageable.

Le délai de six semaines, accordé pour réaliser ce projet, est un peu court, et il est fréquent que les dernières 72 heures se passent en continu, dans un esprit d'équipe. L'on pourrait faire des choses bien meilleures en accordant in extremis un délai supplémentaire. Ce n'est pas l'objectif !

Il serait aussi possible de réutiliser le produit de l'année antérieure, et d'en assurer de copieuses extensions, faisant progresser le modèle chaque année. Là non plus, ce n'est pas l'objectif !

Le délai court prépare à la vie active, où tout est toujours demandé pour hier, et le stress faisant partie de la vie du Géomaticien, il doit s'acquiescer à l'entraînement... La fierté du produit réalisé est une belle compensation, et la démonstration finale (figure 5) se traduit ensuite par une explosion de joie collective.

Conclusion

Ce projet repose sur des outils de Génie Logiciel développés au LIST depuis fort longtemps et perfectionnés au long des années. Il s'agit d'outils de recherche directement utilisés par des Etudiants en fin d'étude, et non de logiciels commercialisés. L'objectif est essentiellement de les préparer aux SIG de demain, qui sont déjà présents dans divers secteurs. La méthodologie utilisée a pour but de diviser le travail :

- en étapes,
- en équipes,

permettant de gagner ainsi trois choses :

- un temps de développement très court,
- un produit plus modulaire et plus fiable,
- une maîtrise du logiciel permettant d'apprendre à s'interfacer efficacement et d'ajouter aisément de nouvelles extensions sans tout remettre en question.

Dans un contexte dit "orienté-objet", l'on a ainsi monté une architecture comprenant base de données, simulation discrète et continue, visualisation avec immersion, sans aboutir à une "usine à gaz". Il serait intéressant de compléter ceci en intégrant un moteur d'inférence, faisant partie du système SES⁷, permettant de gérer les composants SIG dans un contexte d'intelligence artificielle, ou encore le moteur neuronal fondé sur des TAD développé dès 1993, procurant des facultés d'apprentissage.

La plateforme intégrée est déjà utilisée complètement dans le cadre d'un projet de Génie Logiciel, mais cette dernière tranche n'a pas encore été mise à contribution en totalité dans le projet baptisé "SIG-4D". C'est aisément faisable, mais il faut

(7) SES= Système Expert Structuré, ou Structured Expert System, développé en 1984 autour du noyau d'HBDS.

draît une semaine de plus. Nous sommes convaincus que les futurs SIG-3D incorporeront tous un ou plusieurs moteurs d'inférence, des processus discrets et continus, un moteur neuronal, et des possibilités accrues d'immersion.

La lacune majeure que nous rencontrons actuellement n'est ni conceptuelle, ni logicielle ; elle est matérielle ; c'est l'impossibilité d'obtenir avec des dispositifs à bas prix des facultés ergonomiques de déplacement libre dans le modèle, comme pourrait le faire une chauve-souris ou une libellule. Il reste à attendre un équipement complet (et léger) de réalité virtuelle pour le prix d'une paire de lunettes... ●

NB : Les images de cette communication ne sont pas de l'auteur mais sont extraites des travaux de ses Etudiants lors de la démonstration finale du projet "SIG-4D" (figure 5). Merci à eux pour le travail réalisé et l'investissement fourni.

Contact

Pr. François BOUILLÉ

LIST – Laboratoire d'Informatique des Sciences de la Terre,
Université Pierre et Marie Curie (Paris VI) et ENSG
Francois.Bouille@ensg.eu

Bibliographie

F. Bouillé *Architecture of a collaborative 4-D GIS for student training*, Int.Adv.Workshop on Virtual Environments and Geocollaboration, Hong-Kong, Dec. 2003, 9 p., bib. cum.

F. Bouillé in prep.

Tome 1 : HBDS, le modèle et la méthode, 320 p.

Tome 2 : La formulation simple et rationnelle des algorithmes avec ADL, 250 p.

Tome 3 : Le langage ADT'81 et la manipulation des TAD persistants, 380 p.

ABSTRACT

Key words : design methodology, HBDS structuring, JAVA-3D, simulation, student training, virtual reality, 4D-GIS

The paper presents the specific methodology and tools for design and implementation of a stereoscopic animated 4D-GIS (3D + time), allowing process simulation and providing an immersive environment ; the goal is a six-week training session for students working in cooperating teams, and using an object-oriented platform and virtual reality technics.