

École Nationale Supérieure d'Architecture de Nancy  
Université Henri Poincaré (Nancy1)  
Institut National Polytechnique de Lorraine  
École Nationale Supérieure d'Architecture de Strasbourg  
Institut National des Sciences Appliquées de Strasbourg

---

## **Mémoire de master Design Global, spécialité « Modélisation et Simulation des Espaces Bâtis »**

**RÉALISATION DE REPRÉSENTATIONS BI- ET TRIDIMENSIONNELLES  
INTERACTIVES POUR UN SYSTÈME D'INFORMATION ARCHÉOLOGIQUE**



Présenté le **28 Septembre 2007** par **Céline PIACENTILE**

Laboratoire d'accueil :

**Equipe Photogrammétrie  
Architecturale et GEomatique  
INSA Strasbourg  
24, boulevard de la Victoire  
67084 STRASBOURG CEDEX**

Sous la direction de :

**Mlle Elise MEYER  
Doctorante MAP-CRAI**



**En couverture,**

Image de synthèse réalisée par le laboratoire MAP-CRAI à partir des hypothèses de restitution du site d'Echternach

## REMERCIEMENTS

---

---

Je tiens tout d'abord à remercier ma directrice de stage, Mme Elise Meyer, pour m'avoir fait confiance, puis pour m'avoir guidée, encouragée et conseillée tout au long de mon stage. Malgré la rédaction de sa thèse, elle s'est toujours montrée disponible et patiente face aux problèmes que j'ai pu rencontrer.

Mes remerciements vont également à l'ensemble de l'équipe MAP-PAGE pour son accueil et l'ambiance joviale régnant au laboratoire.

De nombreuses discussions avec Emmanuel Alby m'ont permis d'élargir mes connaissances sur la conception de modèles 3D tant au niveau informatique qu'artistique. Je lui suis reconnaissante de l'aide qu'il m'a apporté.

L'outil informatique était nécessaire à ce stage et je remercie Bernard Grandmougin, technicien du laboratoire de géomatique, d'en avoir assuré le bon fonctionnement.

Je souhaite remercier les professeurs de l'École Nationale Supérieure d'Architecture de Nancy sans qui je n'aurais pas pu acquérir les connaissances suffisantes pour aborder ce stage dans de bonnes conditions.

Enfin, je remercie mon père pour ses soutiens moraux et financiers sans lesquels je n'aurais pas pu parvenir à ce niveau d'étude.

# SOMMAIRE

---

---

REMERCIEMENTS.....	3
1. INTRODUCTION.....	6
2. PROBLEMATIQUE.....	8
3. ETAT DE L'ART.....	10
<b>3.1 Définitions</b> .....	10
1.1 Notion de modèle.....	10
1.2 Définitions des Systèmes d'Informations [KOE00].....	10
1.3 Notion d'interactivité.....	11
1.4 Notion de réalité virtuelle.....	11
<b>3.2 Gestion et propagation de données archéologiques sur le Web</b> .....	11
2.1 Utilisation de l'outil multimédia au service de l'archéologie.....	11
2.1.1 Forme privilégiée de visualisation 2D : la carte.....	12
2.1.2 Avantages de la visualisation 3D pour la représentation des données.....	13
2.1.3 Outil de visualisation d'objets 3D : le VRML.....	13
2.2 Intégration de données sur le Web : la question de l'interactivité des données... ..	14
2.2.1 Cartographie interactive.....	15
2.2.2 Modèles 3D interactifs.....	15
4. OUTILS UTILISES POUR LES DEVELOPPEMENTS.....	17
<b>4.1 Identification et discrétisation du modèle selon des lieux et des sous-lieux</b> .....	17
1.1 Etude de la structure d'un fichier SVG du type export « Adobe Illustrator ».....	17
1.1.1 Format SVG.....	17
1.1.2 Structure d'un document SVG (Figures 4.1 et 4.2).....	17
1.2 Etude de la structure d'un fichier VRML du type export « Sketchup ».....	19
1.1.1 Historique succinct de l'évolution du VRML.....	19
1.1.2 Structure d'un fichier VRML.....	19
1.2.1 Noeud « Transform ».....	20
1.3 Discrétisation dans une Base de Données.....	21
1.4 Fonctionnement informatique du système.....	21
<b>4.2 Ancrage et mise en évidence des parties choisies</b> .....	22
2.1 Accessibilité des données : l'ancrage.....	22
2.1.1 Utilisation de technologies de l'environnement Web.....	22
2.1.2 Protocole d'ancrage des plans 2D [MEY07].....	23
2.1.3 Ancrage VRML au moyen d'une URL : le noeud Anchor.....	23
2.2 Mise en évidence des parties choisies au moyen de changement de couleur et de surbrillance.....	24
2.2.1 Utilisation de Javascript pour le repérage dans les plans 2D [MEY07].....	24
2.2.2 Repérage des lieux et sous-lieux du modèle 3D.....	24
2.2.2.1 Possibilités de texturer le modèle 3D.....	24
2.2.2.2 Arêtes définies par les noeuds <i>Shape</i> , <i>IndexedLineSet</i> et <i>Material</i> .....	25
2.2.2.3 Noeud <i>TouchSensor</i> et événements <i>ROUTE</i> .....	27

<b>4.3</b>	<b>Représentation globale de plusieurs plans et modèles</b>	28
3.1	Plan à la volée	28
3.2	Réalisation d'un modèle 3D extérieur et d'une coupe avec un bouton interactif	28
3.2.1	Nœud <i>Switch</i>	28
3.2.2	Bouton interactif HUD	29
3.3	Modèle à la volée	29
<b>5.</b>	<b>PROCESSUS D'AUTOMATISATION DE CREATION DE PLANS ET DE MODELES INTERACTIFS</b>	30
<b>5.1</b>	<b>Plans interactifs</b>	30
1.1	Obtention d'un fichier .svg	30
1.2	Ancrage du fichier .svg : ajout d'interactivité	31
1.3	Plans « à la volée »	33
1.3.1	Création d'une nouvelle base de données s'implémentant automatiquement avec les nouveaux plans créés	33
1.3.2	Création d'un programme reprenant les lieux saisis dans un formulaire par l'utilisateur	35
<b>5.2</b>	<b>Modèles 3D interactifs</b>	36
2.1	Choix du logiciel	36
2.2	Améliorations en fonctions des images de synthèse réalisées par le CRAI	38
2.3	Simplifications du modèle	40
2.3.1	Simplifications géométriques des formes circulaires	40
2.3.2	Simplification de l'apparence géométrique du modèle au moyen de textures	42
2.4	Développements précédemment cités et tests	42
2.4.1	Discrétisation du modèle VRML en fonction du nœud <i>Transform</i>	42
2.4.2	Ancrage du modèle	46
2.4.3	Repérage des entités lieu et sous-lieu	47
2.5	Amélioration des paramètres de navigation dans le VRML reconstruit	50
2.6	Réalisation d'un modèle extérieur avec coupe	51
2.7	Modèles à la volée	51
<b>6.</b>	<b>CONCLUSION GENERALE ET PERSPECTIVES</b>	53
	<b>GLOSSAIRE</b>	55
	<b>TABLE DES ILLUSTRATIONS</b>	59
	<b>BIBLIOGRAPHIE</b>	61
	<b>ANNEXES</b>	66

## 1. INTRODUCTION

---

Ce stage s'est déroulé au sein du laboratoire MAP-PAGE (Photogrammétrie Architecturale GEomatique) situé dans les locaux de l'INSA de Strasbourg.

L'équipe est constituée de quatre enseignants chercheurs, un technicien photogrammètre, un technicien informatique et quatre doctorants. Les thèmes de recherche sont basés sur deux axes principaux : la photogrammétrie et la modélisation tridimensionnelle d'objets architecturaux ou autres.

Ce sujet s'inscrit dans la continuité d'une thèse de doctorat débutée en 2004 par Mme Elise Meyer intitulée : « *Acquisition 3D, documentation et restitution en archéologie : proposition d'un modèle de Système d'Information dédié au patrimoine* » dont la soutenance est prévue en octobre 2007. La finalité de cette thèse se situant dans le deuxième axe de recherche précédemment cité est la création d'un Système d'Information fonctionnant sur le Web permettant aux archéologues de gérer les quantités de données qu'ils produisent, et de pouvoir les représenter.

Un premier site d'expérimentation, un château médiéval situé à Vianden au Luxembourg, a permis d'implémenter le système. Des plans interactifs ont été créés et les modèles 3D existants sur le château ont été optimisés et interfacés.

Il s'agit désormais de tester les outils développés (intégration de nouvelles données) et d'automatiser les processus de création et d'interfaçage des données à intégrer dans le Système d'Information Archéologique réalisé.

### Contexte de l'étude

Au cours des années 1975 et 1976, le Service des Sites et Monuments Nationaux du Luxembourg entreprit des fouilles sur le site de la villa gallo-romaine d'Echternach. Celles-ci fournirent quantités d'informations qui furent ensuite analysées et rassemblées en 1981 dans l'ouvrage de référence traitant de ce site : « *Ausgrabungen in Echternach* » (Metzler et al. [1981]). Puis le site archéologique fut au fil des années quelque peu délaissé.

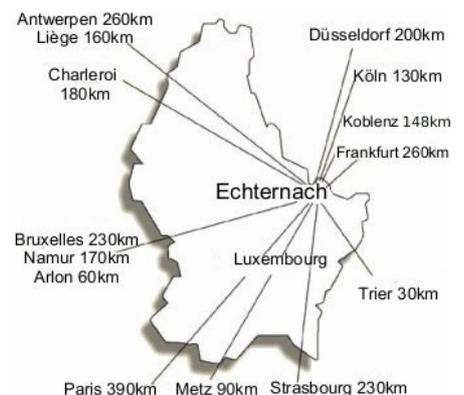


Figure 1.1 : Carte du Luxembourg (webplaza@[2007])

Dans le cadre de l'aménagement du pôle touristique environnant, il fut décidé la construction d'un musée aux abords même des vestiges. L'idée centrale de ce musée est de présenter aux visiteurs une approche de la vie quotidienne dans ce palais-villa à la fin du premier siècle après J-C, à travers des maquettes, des ambiances, des objets-témoins, des présentations des différentes pièces d'habitation et occupations qui s'y déroulaient.

Le Service des Sites et Monuments Nationaux du Luxembourg a confié au MAP-CRAI (Centre de Recherche en Architecture et Ingénierie) la réalisation de tous les documents numériques devant être présentés au public, que ce soit dans le musée ou sur des panneaux *in situ*, permettant aux visiteurs de faire un rapport immédiat entre ce qui subsiste et les hypothèses de reconstitution. Des images de synthèse ont été réalisées à partir de plans 2D et de modèles 3D générés sur AutoCad et 3D Studio.

## Objectifs de l'étude

Le sujet de stage porte précisément sur la transformation des plans 2D et des modèles 3D réalisés par le laboratoire MAP-CRAI en interfaces d'accès aux informations d'une base de données sur Internet. Nous pouvons distinguer deux objectifs principaux :

- le premier est d'étudier les possibilités d'exploitation de dessins 2D ou 3D au format .dwg pour leur utilisation comme interfaces d'accès à des données sur le Web, ce qui implique une transformation des plans en SVG et des modèles en VRML ;
- le second est d'étudier les possibilités d'automatisation des ancrages, tâche fastidieuse à réaliser manuellement pour toutes les représentations. La finalité de l'ancrage réside dans le fait de pouvoir repérer un élément du plan 2D ou du modèle 3D, cliquer sur un élément et obtenir simultanément le plan 2D ou modèle 3D et les informations le concernant stockées dans une base de données.

## Plan de l'étude

Tout d'abord, un état de l'art sera exposé afin de donner des définitions précises de l'objet de notre étude, puis de la positionner face aux travaux existants dans ce domaine.

Par la suite, les différentes étapes du processus d'automatisation et les informations nécessaires à leur compréhension seront relatées dans le deuxième chapitre, ainsi que les différents outils utilisés.

Enfin, nous présenterons les expérimentations concernant la mise en place du processus d'automatisation, en explicitant dans le détail les différentes démarches effectuées pour mener à bien le projet. La présentation du produit final sera appuyée par l'explication des scripts principaux, le script intégral de l'application étant trop volumineux pour être inclus dans le rapport et l'intégralité du script étant la propriété du laboratoire.

Nous conclurons finalement sur l'apport de nos travaux, les améliorations possibles et les perspectives envisageables.

## 2. PROBLEMATIQUE

---

La thèse d'Elise MEYER [MEY07] décrit « *une recherche menée dans le domaine de la conservation du patrimoine* ». Ce terme a pour signification dans son étude « *l'activité qui vise à identifier, à protéger et à faire connaître les éléments estimés de l'environnement bâti, soit les édifices, les structures et les sites créés tout au long de l'évolution d'un pays ou d'une région* ». La finalité de cette thèse est de produire un outil informatique permettant aux archéologues de gérer, de stocker et de diffuser, en les valorisant, les données recueillies lors d'une fouille archéologique.

Les Systèmes de Gestion de Bases de Données (SGBD) offrent des solutions concrètes pour le stockage des différentes sources documentaires rassemblées. De plus, l'utilisation d'Internet dans la perspective de diffuser des informations, et l'avènement de nouvelles solutions *open source* ont orienté la réflexion sur la question de la représentation des sites archéologiques dans un Système d'Information pour l'architecture patrimoniale. De nouvelles opportunités peuvent en effet s'ouvrir avec le développement des technologies du Web tels que la recherche d'informations au travers de différentes interfaces (textuelle, par l'image, en 3D) sur un réseau de ressources plus important.

Le Système d'Information d'Elise Meyer [MEY07] a pour objectif de proposer un système de découverte de sites archéologiques reconstitués numériquement à partir de connaissances spatio-temporelles.

La construction d'interfaces graphiques s'accompagne en effet d'une double problématique de modélisation : modélisation du corpus architectural constituant la géométrie du site (1) et modélisation temporelle de ce corpus (2).

Tout d'abord, l'étude du bâti patrimonial (1).

Un site archéologique est constitué d'un ensemble de lieux et l'enjeu a été de repérer les lieux ayant joué un rôle de pôle. Puis il a été défini des hiérarchies locales, entre lieux et sous-lieux à partir, d'une part, des connaissances préalables accumulées sur le fonctionnement des sociétés antiques et, d'autre part, de l'analyse des données archéologiques recueillies au cours des fouilles. Un sous-lieu est une partie significative d'un lieu, dépendant de celui-ci, ayant une fonction propre et parfois en complémentarité avec les autres sous-lieux du même lieu (voir Glossaire pour le lieu Bains).

Puis, vient l'étude temporelle du site (2).

Les données archéologiques possèdent plusieurs dimensions temporelles dont la plus importante est leur époque d'existence originelle qui permet aux archéologues de reconstituer dans un ordre chronologique la séquence des événements et des processus culturels. Le Système d'Information permet de manipuler ces dimensions temporelles.

Le laboratoire MAP-CRAI avait été mandaté pour la réalisation de documents numériques devant être présentés au public, que ce soit dans le musée ou sur des panneaux *in situ*, permettant aux visiteurs de faire un rapport immédiat entre ce qui subsiste et les hypothèses de reconstitution (cf. Contexte de l'étude dans le chapitre d'**Introduction**).

Dans le cas du site d'Echternach, seule une période significative fut modélisée par le MAP-CRAI (à la différence du site de Vianden pour lequel cinq périodes avaient été représentées). Ainsi, cette dimension temporelle, bien qu'existante au sein du système développé, ne sera pas illustrée dans ce rapport.

Des plans 2D, ainsi que des modèles 3D des divers lieux du site ont été réalisés avec AutoCAD, logiciel de conception pour le dessin industriel ou l'architecture et 3D Studio, logiciel de modélisation et d'animation 3D. Notre travail consiste à optimiser ces plans et ces modèles de manière automatisée et permettre ainsi à l'archéologue d'obtenir rapidement et aisément une représentation interactive des plans 2D et des modèles 3D du site qu'il gère.

De plus, le Système d'Information fait appel à des données multiples. Nous souhaitons avoir une liaison bidirectionnelle entre chacune des données : textuelles, iconographiques, cartographiques, relevés, photographies, etc.

Ainsi, les informations d'un lieu seront accessibles à partir des plans 2D et des modèles 3D du lieu, et il sera également possible d'obtenir les plans ainsi que les modèles à partir des informations textuelles d'un lieu enregistrées dans une base de données. La figure 2.1 illustre les différents liens aller-retour possibles entre les données de différents types.

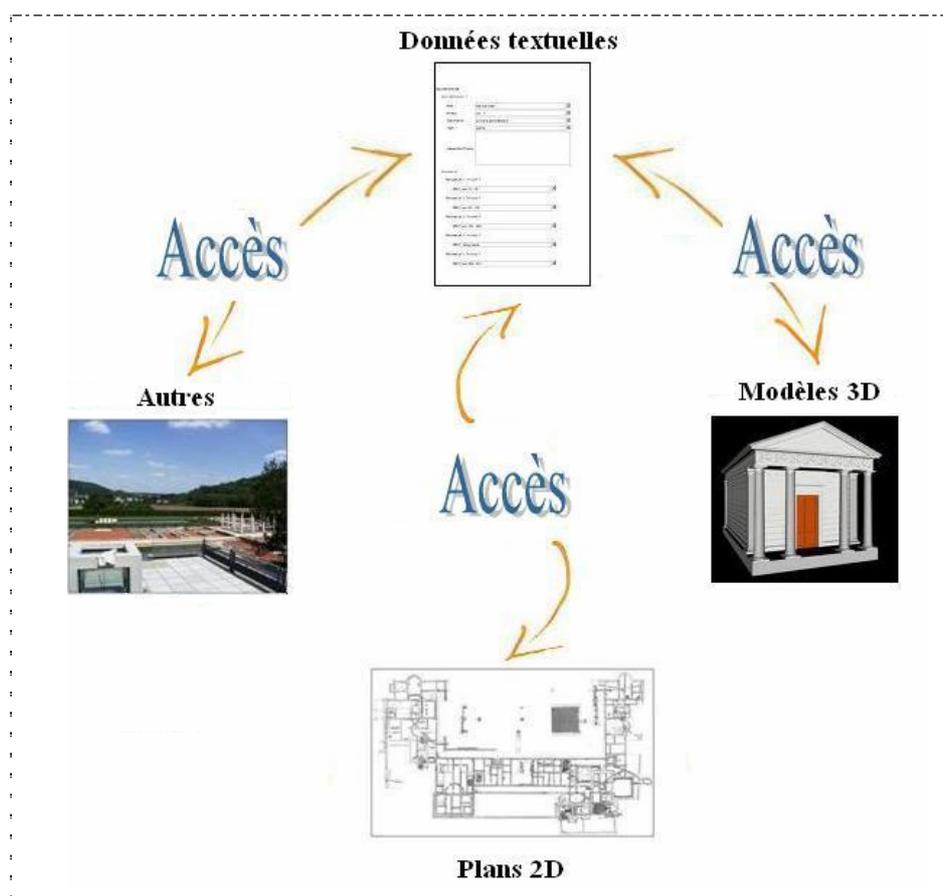


Figure 2.1: Liens aller-retour permettant d'accéder aux diverses données

Ainsi, pour résumer cette problématique, notre travail consistera en l'élaboration de programmes s'implémentant au Système d'Information d'Elise Meyer, basés sur les mêmes principes d'interactivité et permettant l'ancrage automatique de plans 2D et de modèles 3D en relations avec leurs données textuelles.

### 3. ETAT DE L'ART

---

Dans un souci de compréhension, il convient tout d'abord de définir convenablement le sujet, et, à cette fin, nous donnerons les définitions de certains termes utilisés dans le titre de notre mémoire, ainsi que celles du modèle et de la réalité virtuelle.

Puis, pour pouvoir situer cette étude et prendre connaissance des solutions actuelles au problème posé, nous avons analysé divers travaux effectués dans ce domaine, que nous exposerons dans une deuxième partie.

#### 3.1 Définitions

##### 1.1 Notion de modèle

La notion de modèle possède une pluralité de définitions. Nous ne citerons dans ce mémoire que celles liées à notre domaine ;

- 1) « *Ce qui sert ou doit servir d'objet d'imitation pour faire ou reproduire quelque chose [...] Objet de même forme qu'un objet plus grand exécuté en réduction.* » [LEP88]
- 2) « *Objet abstrait, construit selon les procédures dont on estime qu'il constitue une représentation suffisamment fidèle bien que simplifiée de la réalité.* » [LAD91]

Cette notion de simplification est inhérente à l'action de modéliser, mais dans notre cas, il s'agit plutôt d'exprimer les objets qui composent la réalité ou l'imagination de leurs concepteurs sous forme de figures tridimensionnelles. Elle dépend des outils logiciels, des performances du matériel informatique, de la compétence du concepteur mais également du choix de ce qu'il s'agit de modéliser ou non selon les besoins finaux.

##### 1.2 Définitions des Systèmes d'Informations [KOE00]

En 1980, R. Conzet, l'un des pionniers en ce qui concerne les Systèmes d'Informations, donnait la définition suivante :

*« Si les fonctions d'un système se limitent à la saisie, le stockage, le traitement et la représentation de l'information, alors il s'agit d'un système d'informations. Il se compose ainsi de l'ensemble des données et des directives de traitement. Ce système doit permettre à l'utilisateur d'obtenir des informations sous forme compréhensible, dérivées des informations initiales ».*

K. Brassel [1990] définit les Systèmes d'Informations comme des outils polyvalents permettant le traitement et l'analyse informatique de données et d'informations.

Ainsi, un système d'informations va comprendre une chaîne de processus permettant la saisie des données, leur gestion, leur analyse et leur représentation (modèle à quatre composants) [KOE00].

### 1.3 Notion d'interactivité

Citons en premier lieu la définition du petit Larousse [LEP96] afin de comprendre globalement cette notion :

« *Faculté d'échange entre l'utilisateur d'un système informatique et la machine, par l'intermédiaire d'un terminal doté d'un écran de visualisation.* »

Nous pouvons nous demander de quoi est constituée l'interaction de l'utilisateur dans un monde virtuel. Selon Fuchs et al. [FUC01], les tâches exécutées par l'utilisateur se résumeraient ainsi :

- 1) Observer le monde virtuel
- 2) Naviguer dans le monde virtuel
- 3) Agir sur le monde virtuel
- 4) Communiquer avec autrui

Ainsi, notre modèle pourra être considéré comme interactif s'il remplit les conditions précédemment citées.

### 1.4 Notion de réalité virtuelle

« *La réalité virtuelle est une notion issue du monde de l'interactivité et des espaces informatiques en trois dimensions. La réalité virtuelle (ou RV) est un mode d'utilisation et de gestion des mondes en 3D : l'utilisateur évolue en temps réel par le biais d'une interface dans un lieu en 3D. Il peut se déplacer à sa guise sans parcours prédéfini. Cette notion est très complexe et fait appel également à des concepts de perception de l'espace par d'autres sens (touché, ouïe, équilibre, sensation des éléments) que celui de la vision.* » [AUS01].

La réalité virtuelle dans le cadre de ce projet utilise des ressources Internet tant sur le plan des banques de données, que sur celui des plans en deux dimensions et des modèles en trois dimensions.

## **3.2 Gestion et propagation de données archéologiques sur le Web**

### 2.1 Utilisation de l'outil multimédia au service de l'archéologie

L'archéologie a su se servir de l'informatique comme d'un outil dynamique et interactif : pour s'en convaincre, il suffit d'observer la part de plus en plus importante que prend cette application dans les documents à destination du grand public, qu'il s'agisse des produits audiovisuels ou, plus récemment, des productions multimédias.

Les travaux de Pimpaud [1999] sur les applications de l'infographie 3D à l'archéologie en montre les différents apports : « *L'établissement d'une documentation secondaire présente l'avantage de rassembler certaines informations issues de la réalité en les portant sur des supports qui en garantissent la diffusion et la reproduction tout en s'affranchissant des contraintes spatio-temporelles de leur référent.* ».

Outre ces principaux avantages de pérennisation (par l'intermédiaire du stockage et de l'archivage) et de support pour la diffusion des données, l'informatique apporte à l'archéologie des outils dans d'autres domaines tels que l'acquisition, la structuration et la gestion des données.

Les données recueillies sur un site archéologique sont nombreuses et de différents types parmi lesquels nous pouvons recenser de manière non exhaustive les données textuelles, les sources iconographiques, les données cartographiques, les relevés ou les photographies [MEY05].

Après les avoir recueillies et structurées, il faut exploiter ces données et les retranscrire au moyen de documents permettant la diffusion de l'information, notamment par le biais de la visualisation.

L'analyse de grandes quantités de données est en effet grandement facilitée par leur mise en forme visuelle. La visualisation de données « *est une représentation temporaire sur un écran des résultats d'un traitement d'informations* » [LGD02]. Il s'agit d'une technologie utilisée pour reproduire des données par des moyens graphiques visant à la fois à faciliter l'interprétation visuelle des données et à faire des liaisons dynamiques entre les modèles et les données scientifiques associées.

Les techniques de visualisation sont devenues incontournables dans plusieurs domaines tels que la médecine, l'architecture ou l'archéologie. Elles offrent par exemple aux archéologues une possibilité supplémentaire pour valider et vérifier leurs hypothèses de restitution des espaces antiques disparus.

Dans le domaine de l'infographie, nous distinguons trois types essentiels de visualisations, classés du plus simple au plus complexe [AAS03] :

- la visualisation 2D : (Verticale, horizontale), ex. plans 2D ;
- la visualisation 3D : (Verticale, horizontale, profondeur), ex. modèles 3D ;
- la visualisation 4D : (Verticale, horizontale, profondeur, temps), ex. animation.

L'utilisation pertinente d'un type de visualisation, alliée à l'interactivité, permet aux utilisateurs d'appréhender de nombreux problèmes de manière intuitive.

### 2.1.1 Forme privilégiée de visualisation 2D : la carte

Par définition, la cartographie est « *une science qui regroupe l'ensemble des concepts, méthodes, techniques permettant de représenter des surfaces de la Terre et de communiquer de l'information au moyen de cette représentation appelée carte* » [KOE99]. Dans le cas présent, il s'agira d'une représentation spatiale des bâtiments composant le site archéologique.

Parallèlement aux évolutions informatiques, B. McCormick et al. [MCC87] ont étudié l'importance des images, et potentiellement des cartes produites par ordinateur, avant de définir la visualisation scientifique comme « *l'utilisation de la technologie informatique pour créer des affichages visuels dans le but de faciliter la pensée et la résolution de problèmes* ». Pour D. Dibiase [DIB90], cette visualisation scientifique inclut simultanément la compréhension et la synthèse de l'information. La visualisation a pour objectif d'utiliser des représentations graphiques pour aider les scientifiques à représenter les résultats d'analyses complexes, de manière à faciliter leur réflexion. La carte permet, à l'échelle du site, de l'appréhender dans sa globalité, de se repérer ou d'estimer des distances.

A présent, l'ensemble de ces considérations théoriques est rendu possible par la généralisation de l'informatique, et la croissance des possibilités de calcul associées aux machines.

En matière de cartographie, ces possibilités ont ouvert la voie à une cartographie dite dynamique et/ou interactive.

### 2.1.2 Avantages de la visualisation 3D pour la représentation des données

La création de scènes 3D, plus communément appelée maquette virtuelle ou modèle 3D, constitue un outil de représentation graphique de données numériques.

La modélisation 3D est, au sens infographique du terme, le fait « *d'exprimer les objets qui composent la réalité sous forme de figures géométriques tridimensionnelles* » [PIM99].

Elle offre de nombreux avantages :

- Elle donne une représentation théorique ou relevée sous forme numérique d'un site archéologique. Les applications sont nombreuses et dépendent du public concerné. Pour le grand public, la maquette virtuelle permet de restituer de façon réaliste un site historique difficile d'accès ou disparu. Les chercheurs, quant à eux, utilisent cet outil pour valider des hypothèses ou pour essayer de trouver des explications afin de poser de nouvelles problématiques. Il constitue alors une aide à l'interprétation des vestiges.
- D'autre part, les maquettes virtuelles offrent un certain degré d'interactivité qui n'existe pas dans les technologies traditionnelles. Celles-ci s'adressent à la fois aux chercheurs et au grand public qui peuvent ainsi interagir avec les vestiges. L'informatique apporte dans ce cas davantage de souplesse, puisqu'un objet, une fois créé dans la machine, peut être observé sous n'importe quel angle et avec différentes options de rendu.
- Les conséquences des deux points précédemment cités sont une implication plus forte de l'utilisateur : le simple spectateur devient acteur de la scène, manipulant le modèle suivant son intérêt.

### 2.1.3 Outil de visualisation d'objets 3D : le VRML

Les notions de manipulation d'objets et de mondes 3D virtuels au moyen de langages tels que le VRML\* (*Virtual Reality Modeling Language*) ont ainsi émergé (l'apparition de sites Internet tels que <http://www.learningsites.com/> permettent à l'utilisateur de naviguer dans des sites architecturaux divers).

Puis s'est développée l'idée d'offrir à l'utilisateur « *une représentation en temps réel de son activité dans le site, en parallèle au défi d'une interface en trois dimensions* » [GAN01].

L'amélioration du langage VRML procure un certain nombre de moyens par lesquels l'utilisateur peut interagir avec une scène tels que l'utilisation d'animations, complétées par la notion de ROUTE (événements générés par le VRML lui-même, par exemple clic de souris) et enfin des hyperliens qui relient des éléments de la scène à d'autres scènes VRML ou à des documents HTML, tel que l'illustre la figure 3.1.

\* Nous verrons plus en détail ce langage dans le chapitre **Outils utilisés pour les développements**.

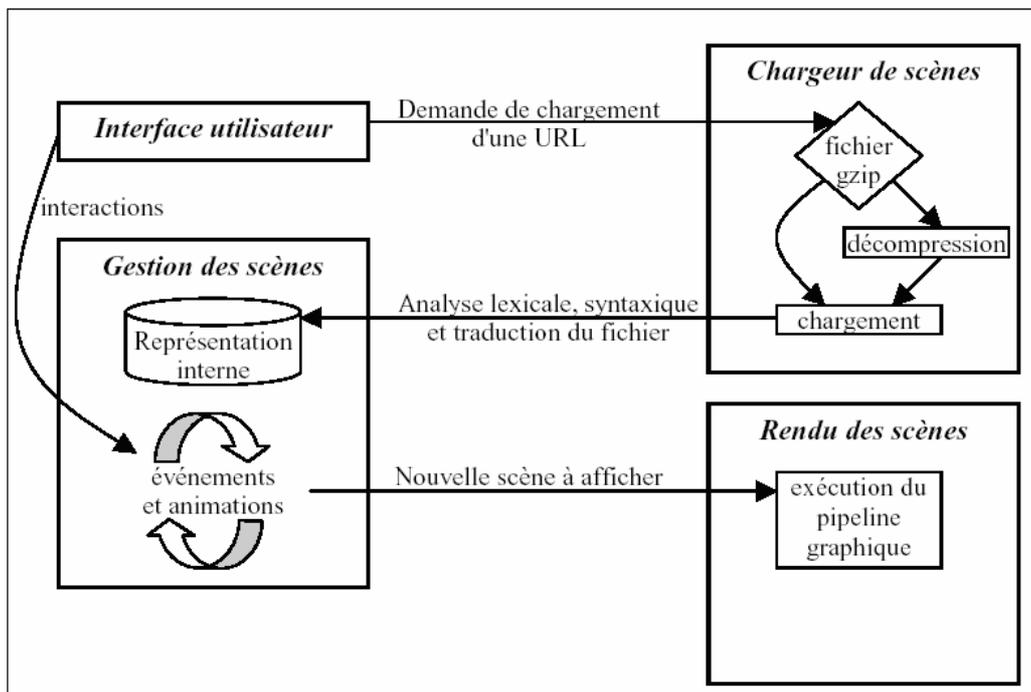


Figure 3.1: Les différents systèmes d'un co-navigateur VRML (Topol [2001])

De nouveaux projets illustrent cette notion que nous pourrions qualifier d'espace virtuel interactif parmi lesquels nous pouvons citer les projets de l'unité mixte de recherche AUSONIUS [AUS01] (<http://www-sira.u-bordeaux3.fr/indexframe.html>). Ainsi, cette équipe de chercheurs a par exemple choisi d'analyser le lieu d'habitation de la petite aristocratie rurale afin de comprendre leur style de vie et utilise pour ce faire des modèles VRML associés aux banques de données. Havemann et al. [HAV06] propose d'intégrer des modèles 3D, soit dans leur intégralité, soit en parties selon le degré de détails que l'utilisateur souhaite visualiser, dans une « librairie » dans laquelle toutes les données seraient interconnectées. D'autres sites tels que <http://www.cvlab.org/> [CVL03], <http://archeus.virtual.club.fr/> [VVA07a], <http://www.temple-de-janus.net/> [JAN ?] permettent à l'utilisateur de visiter virtuellement certains sites archéologiques reconstitués numériquement.

Récemment, de nouveaux outils multimédia sont en cours de développement pour l'archéologie tel que le jeu vidéo [DUF06].

## 2.2 Intégration de données sur le Web : la question de l'interactivité des données

« La pratique de l'archéologie n'est pas destinée à rester la propriété d'un petit nombre d'initiés mais est bel et bien censée bénéficier à l'ensemble de la communauté. » [PIM99]

L'un des outils actuels de communication qui s'est le plus démocratisé ces dernières années est le Web. La problématique désormais ne réside plus dans la question de transmission basique de ces données via des feuilles en HTML mais dans l'apport d'une interface interactive permettant à l'utilisateur de faciliter à la fois sa perception et sa manipulation des données informatiques.

« La représentation interactive des informations a pour objectif de rendre ces mêmes informations accessibles en traduisant visuellement leurs contenus. » [CHA05].

En effet, les données recueillies sur un site sont de natures multiples (textuelles, photographiques, plans...) et si l'écrit reste un outil privilégié pour présenter de manière synthétique les résultats de recherches archéologiques, le rôle de conservation et de visualisation est davantage joué par les autres types de documents tels que les plans 2D et modèles 3D dont nous avons parlé auparavant.

### 2.2.1 Cartographie interactive

Les avancées dans le domaine du Web ont ouvert de nouvelles perspectives dans le secteur de la cartographie interactive et dynamique. Il existe de nombreux travaux de recherche dans ce domaine (Antoniou et al. [ANT04] qui traitent de la conversion des images en fichier XML et SVG) (Adcock et al. [ADC04] qui ont travaillé sur une application Web conçue pour montrer l'information historique de maladie d'animaux par l'intermédiaire de cartes dynamiques) (Josselin et al. [JOS03] dont le Groupe sur la Cartographie Animée et la Représentation des Territoires (GCART) a été mis en place par le CNRS afin de répondre aux questions concernant l'apport de l'animation et des nouvelles technologies de l'information et des communications dans la cartographie) mais surtout une multitude de développements qui utilisent les possibilités offertes par le standard SVG. En quelques années, nous sommes passés d'une cartographie papier à la production de véritables atlas électroniques.

Néanmoins, les projets existants ne sont souvent qu'une publication sur le Web de documents cartographiques simples décrits en SVG. Ce type d'utilisation avec ses possibilités de navigation voire d'animations, convient dans de nombreux contextes. Dans le cas du projet Hypercarte (Grassland et al. [GRA05]) par exemple, dont l'objectif est de mettre au point de nouveaux outils interactifs de production, de représentation et d'interrogation cartographique.

Il faut ajouter à ces standards la mise en ligne des nombreuses bibliothèques Java utilisables pour développer et distribuer des logiciels cartographiques.

### 2.2.2 Modèles 3D interactifs

Ainsi, de nouveaux projets combinent à la fois représentation visuelle de l'objet et textes explicatifs grâce à une association entre plans 2D et/ou modèles 3D et une base de données. Tel est le cas par exemple du projet EROS-3D [2006-2008] (<http://eros3d.ensea.fr/>) qui porte sur l'exploitation d'une base de données comprenant plus de 800 modèles 3D d'œuvres d'art obtenus par un scanner laser. A titre d'exemple encore, le projet FOVEA [FOV03] (Fouille Virtuelle d'Environnement Paléo-anthropologique) accessible sur Internet à l'adresse <http://foveaproject.free.fr/>, qui permet d'intégrer les modélisations du sous-sol, de l'environnement et des fossiles, et de créer un site de fouille "virtuel" permettant à l'utilisateur de choisir sa période de fouille. L'indexation de tous les objets « trouvés » permet aussi de les sélectionner ou de modifier l'affichage en fonction de leurs caractéristiques.

Enfin, voici les projets se rapprochant le plus de l'étude réalisée ici. Comme nous l'avons écrit précédemment, ce PFE s'inscrit dans le cadre d'une thèse de doctorat débutée en 2004 par Elise Meyer [MEY07] et cette bibliographie ne serait pas complète si nous ne citons pas son travail.

D'autres projets en parallèle développent eux aussi des systèmes d'informations avec la même approche et les mêmes technologies à savoir MySQL, PHP et le VRML. Nous pouvons citer le projet CISAR [CIS06] dont le but est de créer un Système d'Information pour l'Archéologie et le Patrimoine basé sur des logiciels *open source* (Heine et al. [HEI06]), ou encore le projet MARVIN (Méthode d'Administration de Réalité Virtuelle pour l'Internet) [MAR07] dont le projet est de créer une méthode afin de générer du VRML dynamique à partir de PHP.

Après avoir défini certaines notions et situé notre étude, nous allons désormais exposer notre travail en deux temps. Tout d'abord, nous présenterons les différents outils informatiques utilisés pour les développements, puis nous présenterons nos résultats et démontrerons ainsi l'apport de ce travail.

## 4. OUTILS UTILISES POUR LES DEVELOPPEMENTS

---

### 4.1 Identification et discrétisation du modèle selon des lieux et des sous-lieux

Sachant qu'AutoCAD est le logiciel utilisé par une majorité de professionnels en CAO/DAO (Conception Assistée par Ordinateur/Dessin Assisté par Ordinateur) et d'archéologues (Eiteljorg II et al. [2002]), nous avons considéré les fichiers .dwg comme données de départ.

L'objectif du projet est d'étudier les possibilités d'exploitation de dessins 2D et 3D au format .dwg pour leur utilisation comme interfaces d'accès à des données sur le Web, ce qui implique une transformation des plans en SVG et des modèles en VRML.

Or AutoCAD ne génère directement ni des fichiers au format SVG (extension .svg), ni des fichiers au format VRML (extension .wrl). Il faut donc passer par un format intermédiaire, puis utiliser un utilitaire pour convertir les fichiers obtenus au format voulu. Nous expliciterons dans le chapitre **Processus d'automatisation de création de plans et de modèles interactifs** pourquoi nos choix se sont portés sur le logiciel « Adobe Illustrator » pour les plans 2D et sur le logiciel de modélisation 3D « Sketchup » pour les modèles.

#### 1.1 Etude de la structure d'un fichier SVG du type export « Adobe Illustrator »

##### 1.1.1 Format SVG

Le SVG (Scalable Vector Graphics) est un langage de description de graphiques en deux dimensions, basé sur le langage XML. SVG est normalisé par le W3C (World Wide Web Consortium), et contrairement au format vectoriel Flash, il s'agit d'un outil *open source*.

En anglais, et pour le monde des images, "*Scalable*" signifie qu'il n'y a pas de contraintes de dimensionnement. Les proportions du dessin sont toujours respectées et la mise à l'échelle se fait sans perte de qualité.

Le langage SVG est intéressant dans la mesure où, par l'intermédiaire de scripts tel que Javascript, il permet la création d'animations et/ou d'événements interactifs (*onmouseover*, *onmouseout*, *onclick*, *onfocus*).

##### 1.1.2 Structure d'un document SVG (Figures 4.1 et 4.2)

L'élément <path> (Figure 4.2) définit le chemin qui borde la forme géométrique que nous souhaitons dessiner. Il peut être fermé ou non. Il s'agit de l'élément de base du langage SVG puisque tout autre élément graphique (*circle*, *rect*, *polygon*, etc) peut se décomposer en un élément path. Chaque lieu et sous-lieu est défini par un élément « path ».

## Structure d'un document SVG

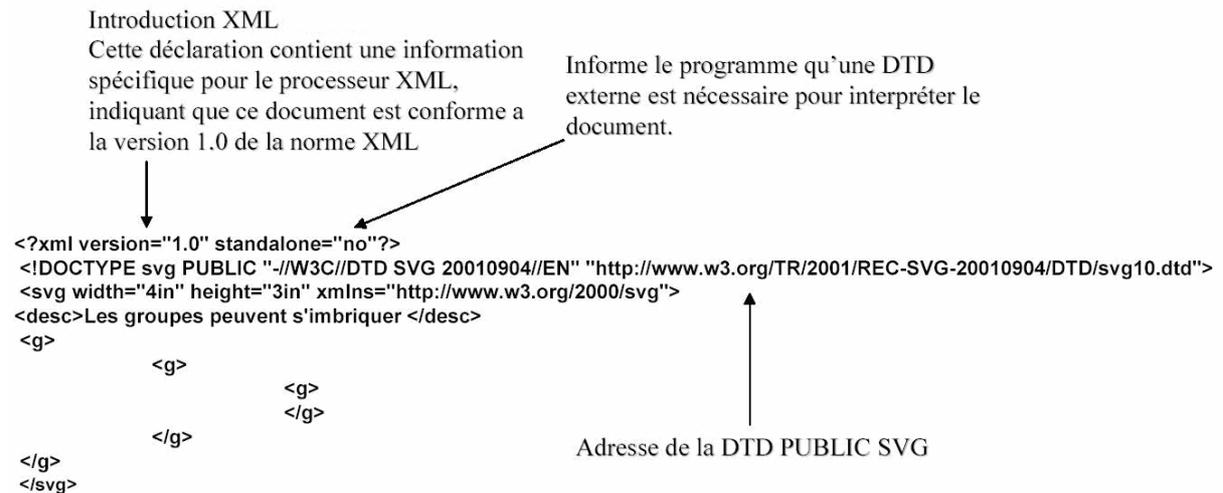
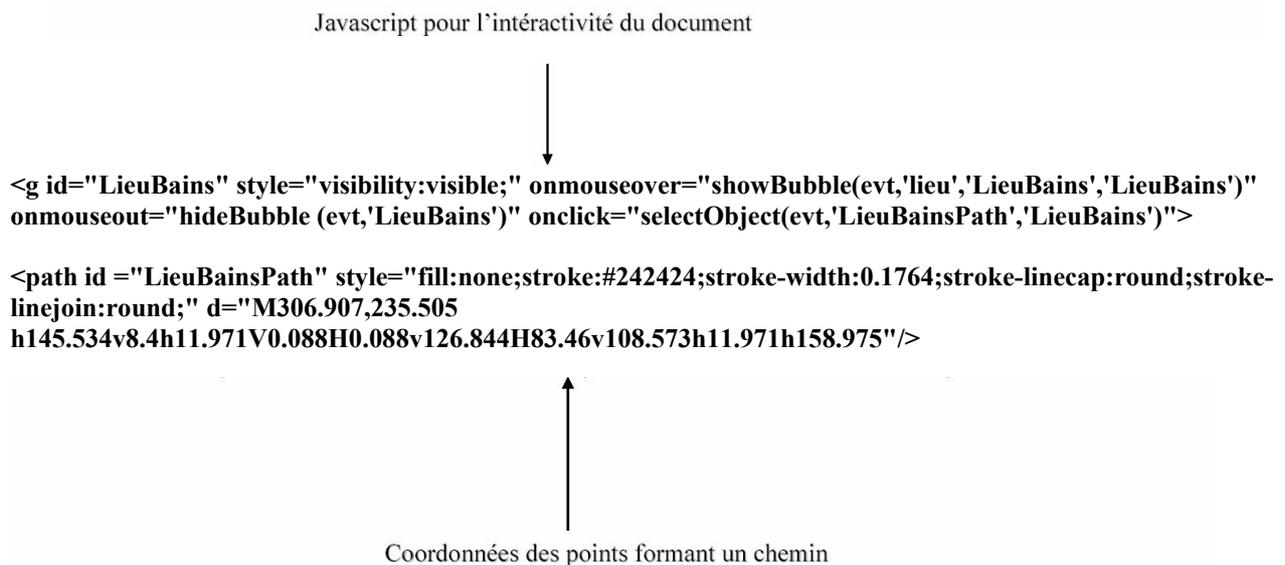


Figure 4.1 : Structure d'un fichier SVG (Ponserre [Date inconnue])

## Structure d'un document SVG



Définitions des données de tracé :

- M x,y désigne le point de commencement du tracé
- L x,y dessine une ligne allant du point courant au point donné (qui devient le nouveau point courant)
- H x dessine une ligne horizontale allant du point courant (xc,yc) au point (x,yc)
- V y dessine une ligne verticale allant du point courant (xc,yc) au point (xc,y)
- Z termine le tracé courant en dessinant automatiquement une ligne droite du point courant au point initial défini dans M

Figure 4.2 : Structure d'un fichier SVG : géométrie d'un lieu repérable

## 1.2 Etude de la structure d'un fichier VRML du type export « Sketchup »

Les ouvrages, revues ou mémoires traitant du langage VRML sont nombreux. Par conséquent, nous avons décidé de ne faire qu'une description succincte de ce langage et de renvoyer le lecteur à ces ouvrages. Nous ne présenterons dans ce rapport que les points importants liés à notre étude.

### 1.1.1 Historique succinct de l'évolution du VRML

La norme VRML 1.0 (*Virtual Reality Modeling Language*), élaborée en 1994, est un format de fichier, qui, associé à un plugin (logiciel traduisant le fichier VRML en une représentation 3D et calculant les images de la scène vues par l'utilisateur), permet de décrire un monde tridimensionnel fixe et de le visualiser interactivement sur Internet.

Trois ans plus tard, une nouvelle version plus puissante (VRML 97 devenu VRML 2.0), autorisant de nombreuses fonctionnalités en sus telle que la possibilité d'animation, est devenue le standard le plus communément adopté.

Aujourd'hui, le développement s'effectue sous l'égide du W3C (World Wide Web Consortium) alors que le langage évolue sous l'appellation X3D (eXtensible 3D). La structure du X3D est calquée sur celle du VRML et de nombreux éléments de syntaxe ont été conservés. Ainsi, passer d'un langage à son successeur est assez aisé (rajout de balises).

Cependant, Elise Meyer [MEY07] a décidé de garder le VRML comme format de description des scènes 3D pour plusieurs raisons :

- Aucun logiciel de modélisation n'exporte en X3D actuellement mis à part Blender, logiciel gratuit mais difficile d'utilisation (cf. tableau 5.1 du chapitre **Processus d'automatisation pour la création de plans et de modèles interactifs**).
- VRML est un langage plus communément employé.
- Pour l'affichage des scènes décrites au format X3D, il existe peu de navigateurs standardisés tels que Cortona ou CosmoPlayer (il en existe un sous environnement Linux appelé FreeWRL).

Ainsi, tous les modèles 3D seront décrits avec le langage VRML 2.0.

### 1.1.2 Structure d'un fichier VRML

Un fichier VRML est constitué d'une entête obligatoire `#VRML V2.0 utf8` destinée à informer le browser sur la version de VRML utilisée, suivi du code VRML.

VRML est un langage basé sur une organisation hiérarchique d'objets (appelés nœuds) ayant un nom et contenant des attributs appelés champs.

Il existe trois types de nœuds (Figure 4.3) :

- ceux qui servent à regrouper (Figure 4.4) ;
- ceux qui contiennent des données (formes 3D, textures, scripts...)
- ceux qui sont attachés à d'autres nœuds et qui ne peuvent être indépendants (*Material* par exemple, qui est un nœud permettant de donner les propriétés matérielles d'un objet, est toujours défini après le nœud *Appearance* et ne peut en être détaché).



### 1.3 Discrétisation dans une Base de Données

Afin de répondre aux besoins d'interopérabilité et de portabilité de l'application, le choix d'utiliser des solutions *open source* s'est rapidement avéré pertinent (raisons économiques et pratiques). Il s'agissait aussi d'un moyen d'évaluer le potentiel des outils libres, dans le contexte de la montée en puissance de l'*open source* dans le domaine de la géomatique, et dans les technologies informatiques en général.

Nous avons installé le pack WAMP 5 (Windows, Apache, MySQL, PHP), plateforme de développement Web sous Windows permettant de développer des sites Web dynamiques à l'aide du serveur Apache2, du langage de script PHP5 et d'une base de données MySQL version 5.



Figure 4.5 : Logo WAMP 5

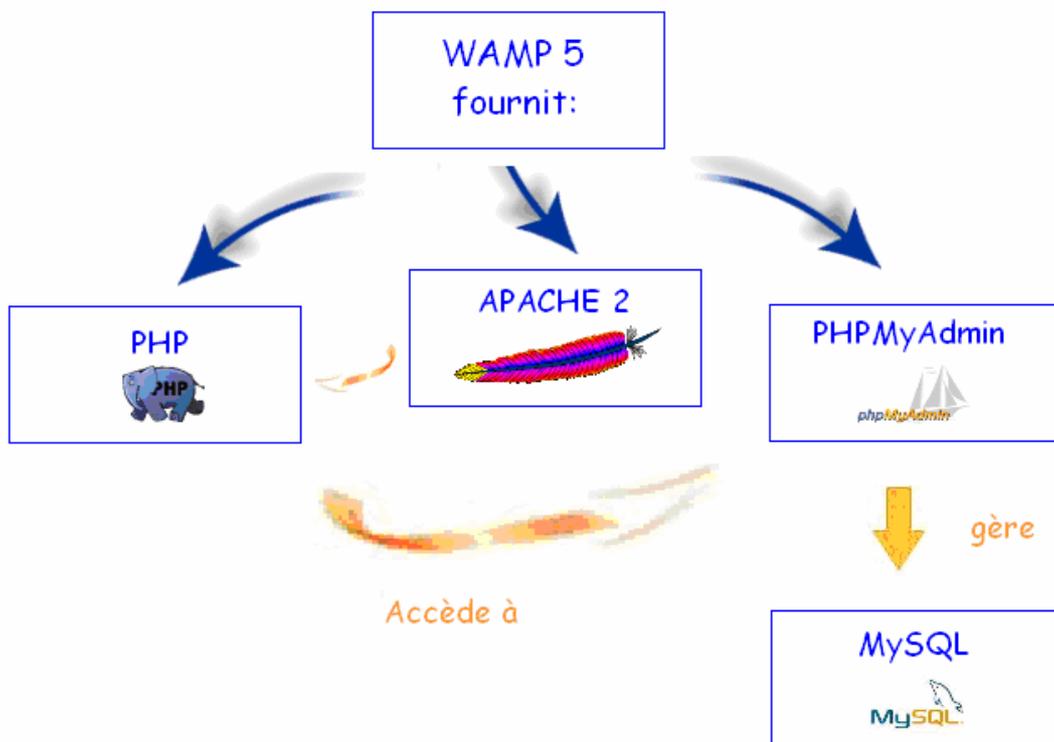


Figure 4.6 : Illustration de WAMP 5 et de ses composants

Nous allons donc élaborer des programmes PHP en interaction avec des bases de données gérées par PHPMyAdmin (outil permettant d'administrer les différentes bases de données) dans lesquelles les plans 2D et modèles 3D vont être discrétisés dans des tables selon leurs structures (Figure 4.6).

### 1.4 Fonctionnement informatique du système

Tel que nous l'avons vu précédemment, le Système d'Information se base sur un serveur PHP. Nos plans 2D et nos modèles 3D seront donc rendus interactifs par l'intermédiaire d'un processus d'ancrage faisant appel à des scripts PHP (en liaison avec des bases de données MySQL), implémentant des Javascripts lors de la reconstruction du fichier SVG et des VRMLScripts et des Javascripts lors de la reconstruction du fichier VRML. La figure 4.7 illustre le fonctionnement informatique du système :

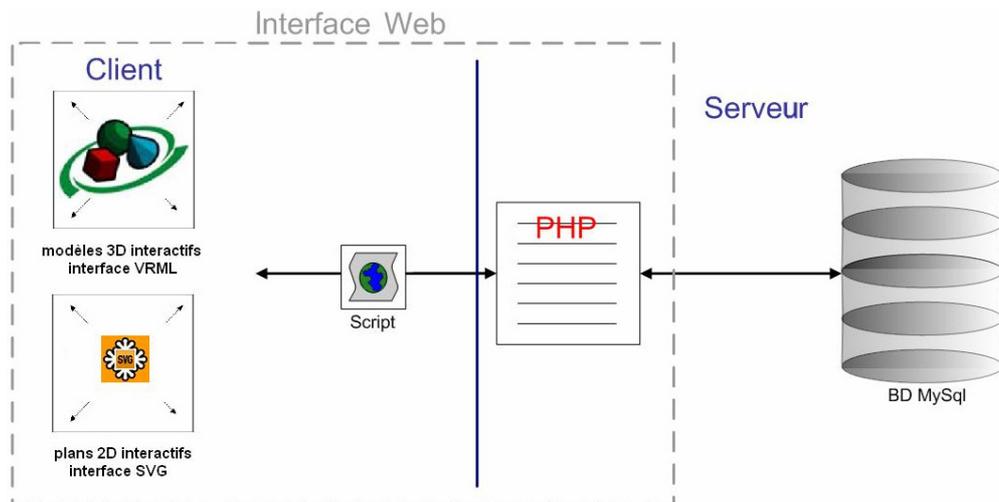


Figure 4.7: Fonctionnement informatique du Système d'Information (Meyer et al. [2006] modifié)

## 4.2 Ancrage et mise en évidence des parties choisies

### 2.1 Accessibilité des données : l'ancrage

#### 2.1.1 Utilisation de technologies de l'environnement Web

Le World Wide Web (littéralement la « toile d'araignée mondiale »), communément appelé le Web, fut inventé en 1990 par une équipe du CERN dirigée par Tim Berners-Lee et Robert Caillau.

Il s'agit d'une application d'Internet dont l'objectif est de permettre au moyen d'un navigateur par un simple clic sur un mot ou un objet (appelés hyperliens ou liens hypertexte), d'accéder à une autre partie du même document, ou à un autre document situé sur un autre ordinateur.

Les normes fondatrices du Web sont :

1. Les *URL* (Uniform Ressource Locator) qui identifient chaque page spécifique sur le Web et qui en facilitent la recherche.
2. Le protocole *HTTP* (HyperText Transfert Protocole) qui, sur le modèle client-serveur, permet de transmettre et de consulter des documents.
3. Le langage *HTML* (HyperText Markup Language) qui permet de lier les données en les balisant.

Nous souhaitons mettre en relation le plan 2D et le modèle 3D avec les informations contenues dans la base de données les concernant. Pour cela, lorsque l'utilisateur cliquera sur une entité géométrique (lieu ou sous-lieu), nous allons indiquer l'adresse URL de la base de données à atteindre sur le réseau, ce que nous appelons l'ancrage.

### 2.1.2 Protocole d'ancrage des plans 2D [MEY07]

Nous souhaitons obtenir un plan interactif de l'ensemble de la villa qui permettra à l'utilisateur de repérer un lieu ou un sous-lieu et d'avoir la possibilité de le cliquer afin d'accéder directement à sa fiche descriptive. Voici le protocole utilisé par Elise Meyer [MEY07] pour rendre les plans 2D du château de Vianden interactifs. Une fois nos fichiers .svg obtenus, nous avons utilisé les mêmes principes d'interactivité que ceux que nous allons décrire.

La première étape consiste à délimiter graphiquement les zones d'intérêt au moyen de polygones jointes placées dans un calque ayant pour nom celui du lieu ou du sous-lieu correspondant, puis d'obtenir un fichier .svg (cf. § 1.1 du chapitre **Processus d'automatisation de création de plans et de modèles interactifs**).

La deuxième étape, après avoir analysé le fichier résultant, est l'insertion d'interactivité dans le graphique. Nous avons donc ajouté des liens sur chacune des zones représentant les lieux et les sous-lieux. Ces liens permettent de cliquer sur la zone, ce clic entraînant l'exécution de la fonction PHP permettant de retrouver toutes les données faisant référence à un lieu ou un sous-lieu choisi, dont il faut écrire l'identifiant comme paramètre du programme PHP à exécuter. En effet, lors de l'exécution d'une fonction définie dans un programme en PHP, celle-ci peut utiliser des paramètres d'entrée indiqués au moment de l'appel du programme PHP après un point d'interrogation et séparés par le terme « & ».

### 2.1.3 Ancrage VRML au moyen d'une URL : le nœud Anchor

Le nœud *Anchor* (Figure 4.8) permet de relier un ou plusieurs objets de la scène à une page HTML ou à un autre fichier VRML. Ainsi, le fait de cliquer sur un lieu ou un sous-lieu déclenchera l'ouverture de la page contenant les informations thématiques relatives au lieu ou au sous-lieu sélectionné.

*Anchor* crée un groupe où:

- tous les enfants (*children*) sont affichés ;
- tous les enfants sont sélectionnables et renvoient à une adresse URL ;
- un champ « *description* » peut être ajouté qui s'affichera lorsque nous déplacerons le curseur sur l'objet ;
- un champ « *parameter* » permet d'ouvrir le lien dans une autre fenêtre.

```
Anchor {  
  url "lien.html"  
  description "lelien"  
  parameter [«target=droite»]  
  children [ . . . ]  
}
```

Figure 4.8 : Description du nœud *Anchor*

## 2.2 Mise en évidence des parties choisies au moyen de changement de couleur et de surbrillance

### 2.2.1 Utilisation de Javascript pour le repérage dans les plans 2D [MEY07]

La dernière étape du protocole d'ancrage d'Elise Meyer [MEY07] est l'ajout d'éléments interactifs supplémentaires. En effet, en l'état, les contours des zones tracées apparaissent en permanence sur l'image et lorsque nous cliquons sur la zone, elle n'est pas mise en évidence, ce qui empêche par la suite de savoir quelle zone a été choisie. De plus, il n'est pas possible d'identifier les noms des lieux définis par les zones en passant sur le graphique par exemple. Pour ajouter de l'interactivité et rendre la navigation dans le graphique SVG plus attractive, Elise Meyer [MEY07] avec l'aide d'Anne Durand, ingénieur de recherche au CNRS, ont écrit des scripts Javascript qui réalisent les fonctions suivantes lors de l'ouverture du graphique SVG dans un navigateur Internet :

- affichage d'une bulle comprenant un titre générique pour tous les graphiques, par exemple « Plan d'Echternach ». Cette bulle disparaît dès que nous passons sur une zone interactive ;
- affichage d'une bulle indiquant le nom du lieu lorsque nous passons sur la zone correspondante. Cette bulle suit les mouvements de la souris et disparaît quand la souris quitte la zone du lieu ;
- modification de l'opacité de la zone : elle devient moins opaque lorsque nous passons dessus avec la souris ;
- mise en évidence de l'objet sélectionné par un changement de couleur lorsque la zone est cliquée. Elle reprend sa couleur initiale lorsque nous cliquons sur une autre zone.

L'apparence de ces bulles est définie par une feuille de style CSS.

Si nous souhaitons définir des sous-lieux dans un lieu, définis graphiquement par l'attribut `id=""`, il nous suffit d'imbriquer les polygones représentant les sous-lieux dans le polygone lieu et de les grouper.

Si tel est le cas, un script a été programmé pour que l'utilisateur ne puisse pas accéder aux sous-lieux sans avoir préalablement cliqué sur le lieu qui les contient. Ainsi, les sous-lieux ne sont accessibles qu'à partir du lieu dans lequel ils sont contenus et il faut d'abord choisir le lieu global pour pouvoir accéder aux sous-lieux sur le graphique.

Un item apparaît lorsque les sous-lieux sont accessibles pour pouvoir revenir en « mode lieu » (ne permettant pas l'accès aux sous-lieux).

### 2.2.2 Repérage des lieux et sous-lieux du modèle 3D

#### 2.2.2.1 Possibilités de texturer le modèle 3D

Les modèles 3D sont des bâtiments représentant des lieux, composés ou non de sous-lieux, sur lesquels nous possédons des informations dans une base de données. Comme nous l'avons vu précédemment, nous accédons à cette base de données au moyen d'URL (cf. § 2.1 du même chapitre). Mais avant d'accéder aux informations, une fonction de repérage du lieu ou du sous-lieu est indispensable. En effet, l'utilisateur, lorsqu'il « navigue » dans le modèle, doit avoir la possibilité de savoir sur quel lieu ou sous-lieu il se situe.

Pour ce faire, Elise Meyer [2007] avait utilisé un changement de couleur des faces dans le premier site d'expérimentation ayant servi à implémenter le système. Lorsque l'utilisateur passe sur le lieu, ses faces changent de couleur spéculaire (définissant la couleur des rayons lumineux réfléchis sur la surface de l'objet) et lorsqu'il clique dessus pour obtenir les informations le concernant, c'est la couleur générale de l'objet qui change.

Cependant, bien que fonctionnel, ce moyen de repérage a pour défaut l'impossibilité de texturer le modèle, ce qui représente une perte d'informations au niveau des propriétés de surface de l'objet.

Les options d'export sont différentes selon les logiciels et il semble que pour le site de Vianden, dont les modèles 3D avaient été obtenus sur le logiciel Maya, il n'était pas possible d'utiliser les arêtes comme objets de représentation. C'est une possibilité d'export qu'offre Sketchup et que nous allons utiliser afin d'avoir la capacité de texturer l'objet :

*« L'objet tridimensionnel, après la modélisation, n'est composé que d'entités géométriques et mathématiques. Fondamentalement virtuelles, elles ne définissent que sa forme et ne permettent pas de rendre compte de ses propriétés de surface. Qu'il soit défini par un volume plein ou par une simple enveloppe, l'objet, lors de la phase de rendu, doit apparaître en fonction d'un certain nombre de caractéristiques visuelles définies par l'utilisateur. L'alternative la plus simple consiste à appliquer au modèle des propriétés de surface spécifiant sa couleur. [...]*

*La principale technique de simulation de matériau consiste en fait à habiller la surface de l'objet avec une image reproduisant une texture, qu'elle provienne de la réalité ou qu'elle ait été créée de toute pièce : la simple numérisation de la surface d'un objet suffit généralement à produire des matériaux virtuels simulant de manière convaincante leurs homologues réels » [PIM99].*

Ainsi, les arêtes effectueront cette fonction de repérage de l'objet grâce aux mêmes procédés que ceux d'Elise Meyer, à savoir surbrillance de l'objet et changement de couleur (cf. §2.4.3 du chapitre **Processus d'automatisation de création de plans et de modèles interactifs**).

#### 2.2.2.2 Arêtes définies par les nœuds *Shape*, *IndexedLineSet* et *Material*

Le nœud *Shape* (Figure 4.9) permet d'introduire un objet graphique 3D dans une scène. Ce nœud est composé de deux champs attendant d'autres nœuds (dépendants) : le champ « *geometry* », paramètre de la géométrie de l'objet, et le champ « *appearance* », paramètre d'attribut visuel de l'objet.

```
• #VRML V2.0 utf8
  Shape {
    appearance Appearance {
      material Material {
        diffuseColor 1 0 0 }
    }
    geometry Sphere { radius 5.0 }
  }
```

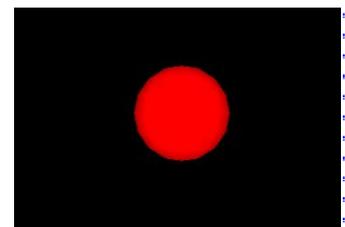


Figure 4.9 : Exemple d'un fichier VRML représentant une sphère rouge [LEN03]

La géométrie d'une arête est définie par le nœud *IndexedLineSet* (Figure 4.10) définissant un ensemble de polygones construites sur des points à partir de deux nœuds dépendants :

- *Coord*, nœud de définition des coordonnées des points
- *CoordIndex*, nœud de définition des polygones

Pour construire cet ensemble de polygones :

- un ensemble de coordonnées est défini ;
- chacune de ces coordonnées sont reliées entre elles en utilisant leurs index ;
- un segment de droite est alors tracé entre chaque paire de sommets ;
- un index de [-1] indique une fin de polygone ;
- la polygone n'est pas automatiquement fermée.

L'apparence d'un objet est défini par le nœud *Material* qui lui-même contient :

- *diffuseColor*, la couleur principale ;
- *emissiveColor*, la couleur brillante (lueur) ;
- *transparency*, la transparence ;
- *shininess*, la brillance ;
- *specularColor*, la couleur de l'ombrage.

Par défaut, le champ utilisé sera la couleur principale. Or, la géométrie *IndexedLineSet* n'est définie que par deux paramètres de couleur : la transparence et la couleur brillante. En effet, la couleur principale n'a pas d'incidence sur la couleur des arêtes et, après que de nombreux essais aient été réalisés, nous nous sommes aperçus que seuls les deux paramètres cités influent. Afin de palier à ce problème, un nœud *Color* permet d'appliquer une couleur différente pour chaque point, ligne ou face.

Cependant, l'automatisation du procédé de repérage ne nous permettait pas l'utilisation de ce nœud (cf. paragraphe suivant). Par conséquent, nous n'avons pu utiliser que les deux paramètres de couleur précédemment cités.

```
Shape {
  appearance DEF COL_ForegroundColor Appearance {
  material Material { diffuseColor 0 0 0
  }
  }
  geometry IndexedLineSet {
  coord Coordinate {
  point [ 50756.454 2381.331 10659.628,
  51535.638 2381.331 11675.081,
  ]
  }
  coordIndex [ 0 1 -1]
  }
}
```

Figure 4.10 : Copie d'écran de la structure *Shape* d'une arête sous VRML

### 2.2.2.3 Nœud *TouchSensor* et événements *ROUTE*

Certains objets ont la possibilité d'émettre et/ou de recevoir des événements (attributs de type *eventIn* et *eventOut*) vers d'autres objets de la scène. Le seul moyen de communiquer avec des composants extérieurs au monde VRML est l'utilisation du nœud *TouchSensor*, permettant de récupérer les événements émis par la souris (Le Gallic [LEG99]).

Les chemins des événements sont décrits par la commande *ROUTE* (Figure 4.11). Cette commande est très importante lorsque nous souhaitons gérer des interactions dans la scène.

Des capteurs récupèrent les événements utilisateurs (*TouchSensor*), puis il y a routage vers des interpolateurs pour créer des animations à l'aide de nœuds *Script*.

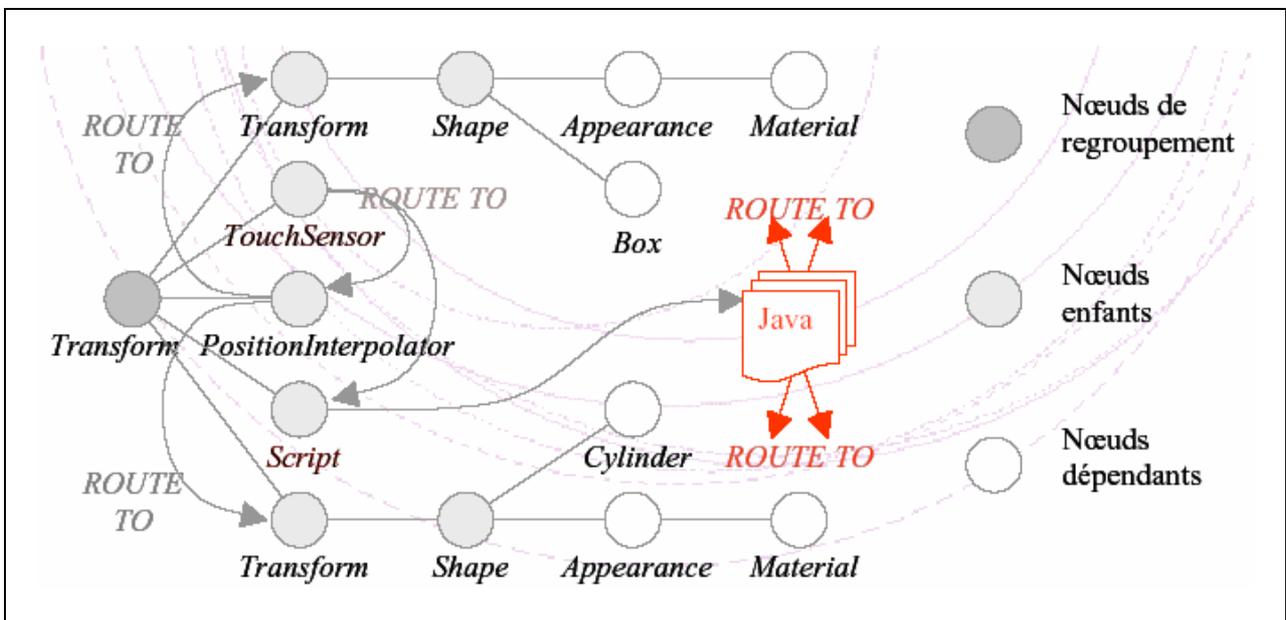


Figure 4.11 : Routage vers des interpolateurs pour créer des animations (Topol [2002])

Quelques règles concernant les *ROUTE* :

- Pour utiliser *ROUTE*, les nœuds doivent être nommés par la commande *DEF*.
- Seuls des événements de même type peuvent être reliés.
- Un événement n'a qu'une source, mais peut avoir plusieurs destinataires (plusieurs *ROUTE* peuvent utiliser la même source).

La première règle ne nous permet pas l'utilisation du nœud *Color* cité précédemment. En effet, le nœud *Color* apparaît dans le champ « *color* » d'une forme géométrique dans certains nœuds dont *IndexedLineSet*. Or, si nous observons la structure d'une *Shape* dont la géométrie est *IndexedLineSet* (Figure 4.12), nous pouvons nous apercevoir que ce nœud n'est pas défini par la commande *DEF*.

```

Shape {
  geometry IndexedLineSet {
    coord Coordinate {
      point [ -2 -2 2, -2 2 2, 2 2 2, 2 -2 2 ]
    }
    coordIndex [ 0 1 -1, 1 2 -1, 2 3 -1, 3 0 ]
    color Color {
      color [
        0 0 1, 0 1 0, 1 1 0, 1 0 0
      ]
    }
  }
}

```

Figure 4.12 : Shape d'une géométrie *IndexedLineSet* définie sans nœud *Appearance* [KDO98]

### 4.3 Représentation globale de plusieurs plans et modèles

#### 3.1 Plan à la volée

Une fois les plans ancrés et dotés d'une fonction de repérage, nous souhaitons avoir la faculté de choisir au moyen d'une liste les plans que nous souhaitons afficher, ceci dans le but d'évaluer l'intégralité du site. L'utilisateur peut ainsi modifier le plan de manière interactive et ne faire apparaître que les éléments qui l'intéressent. Il a ainsi un accès bidirectionnel : soit il accède aux informations via le plan 2D, soit il accède au plan 2D via les informations.

#### 3.2 Réalisation d'un modèle 3D extérieur et d'une coupe avec un bouton interactif

Nous souhaitons que l'utilisateur puisse choisir interactivement entre l'extérieur du lieu et sa coupe. Nous allons créer un « bouton interrupteur » sur lequel nous allons appliquer un nœud *TouchSensor* permettant de passer du modèle à sa coupe à l'aide d'un clic, puis nous allons créer un nœud *Switch* permettant de choisir soit le modèle extérieur, soit sa coupe et enfin des scripts *ROUTE* effectuant l'animation.

Le nœud *TouchSensor* et les événements *ROUTE* ayant été explicités dans le paragraphe précédent, nous allons à présent expliquer le nœud *Switch*.

##### 3.2.1 Nœud *Switch*

Le bloc *Switch* permet de déclarer et regrouper un ensemble d'objets distincts, dont un seul pourra être visible à un moment donné (Figure 4.13).

Les champs du nœud *Switch* sont :

- *choice* : permet de regrouper les objets, dont un seul sera visible à la fois ;
- *whichChoice* : représente l'index du nœud choisi, c'est-à-dire quel élément est actif à un moment donné. Si la valeur est -1, il n'y a aucun nœud choisi.

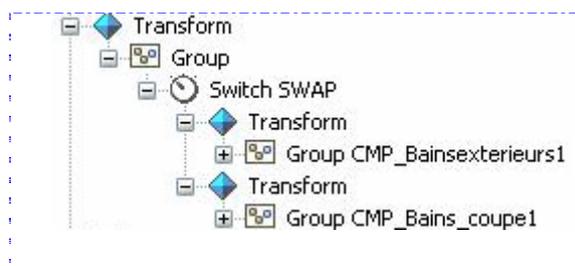


Figure 4.13 : Structure du VRML dont le noeud Switch permet de passer du modèle extérieur à sa coupe

### 3.2.2 Bouton interactif HUD

Le bouton permettant de passer du modèle à sa coupe est constitué d'un HUD (*Head Up Display* dont la traduction est *Visualisation à tête levée*). Les HUD sont une succession de nœuds assurant que le bouton restera à la même position de façon relative au déplacement ou à la rotation de l'utilisateur (Figure 4.14).

Il faut diviser le VRML en deux sections : la première section représente le modèle extérieur et sa coupe, ainsi que le bouton *Switch* (Figure 4.13), la deuxième représente le bouton HUD (sa géométrie).

Dans le cas présent, nous avons employé un script et un *ProximitySensor*.

Le nœud *ProximitySensor* prend en compte les mouvements de l'utilisateur et produit des événements lorsqu'il se déplace, afin de maintenir le bouton HUD placé devant lui. La sonde produit des événements quand l'utilisateur entre, part ou se déplace dans une boîte rectangulaire définie.

Ce nœud a les champs suivants :

- *enabled* indique le statut de la sonde ;
- *center* détermine le centre de la boîte rectangulaire ;
- *size* indique la taille de la boîte.

Le script CODE permettra au modèle extérieur et à la coupe de garder la même position et orientations relatives.

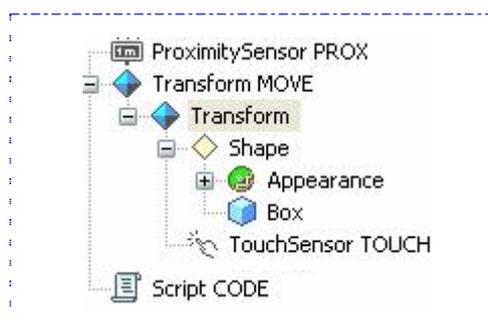


Figure 4.14 : Structure permettant d'obtenir un bouton HUD

### 3.3 Modèle à la volée

De même que pour les plans 2D, une fois les modèles ancrés et dotés d'une fonction de repérage, nous souhaitons avoir la faculté de choisir, au moyen d'une liste, les modèles que nous souhaitons afficher et obtenir ainsi un accès « aller-retour » du modèle aux données.

Après avoir présenté les différents outils utilisés, étape nécessaire à la compréhension de notre étude, nous allons désormais exposer nos développements.

## 5. PROCESSUS D'AUTOMATISATION DE CREATION DE PLANS ET DE MODELES INTERACTIFS

### 5.1 Plans interactifs

#### 1.1 Obtention d'un fichier .svg

Nous avons à disposition des plans 2D réalisés par le MAP-CRAI en format AutoCAD. Pour les raisons précédemment citées (cf. introduction du chapitre **Outils utilisés pour les développements**), nous avons considéré ces données comme données de départ.

AutoCAD ne génère pas de SVG, il faut donc passer par un logiciel intermédiaire pour obtenir un fichier .svg. Nous avons dans un premier temps trouvé un convertisseur *open source* permettant la transformation d'un fichier .dxf en fichier .svg, mais les résultats obtenus ne furent pas satisfaisants. Nous avons alors comparé les différents logiciels d'éditions SVG (cf. Annexe 4). Le SVG est un format vectoriel, il en est de même pour AutoCAD. Or, les logiciels *open source* trouvés convertissaient un fichier raster en .svg. Il nous semblait peu judicieux de convertir un fichier .dwg en format raster pour le convertir à nouveau en format vecteur, malgré la gratuité de toutes ces transformations.

Après comparaison des logiciels nous permettant d'importer du .dwg et d'exporter du .svg (cf. Annexe 4), nous avons choisi Adobe Illustrator, logiciel commercial de création graphique vectorielle, car nous l'avons à disposition à l'INSA et qu'il s'agit d'un logiciel de référence.

La figure 5.1 illustre les différentes étapes d'obtention d'un fichier .svg à ancrer.

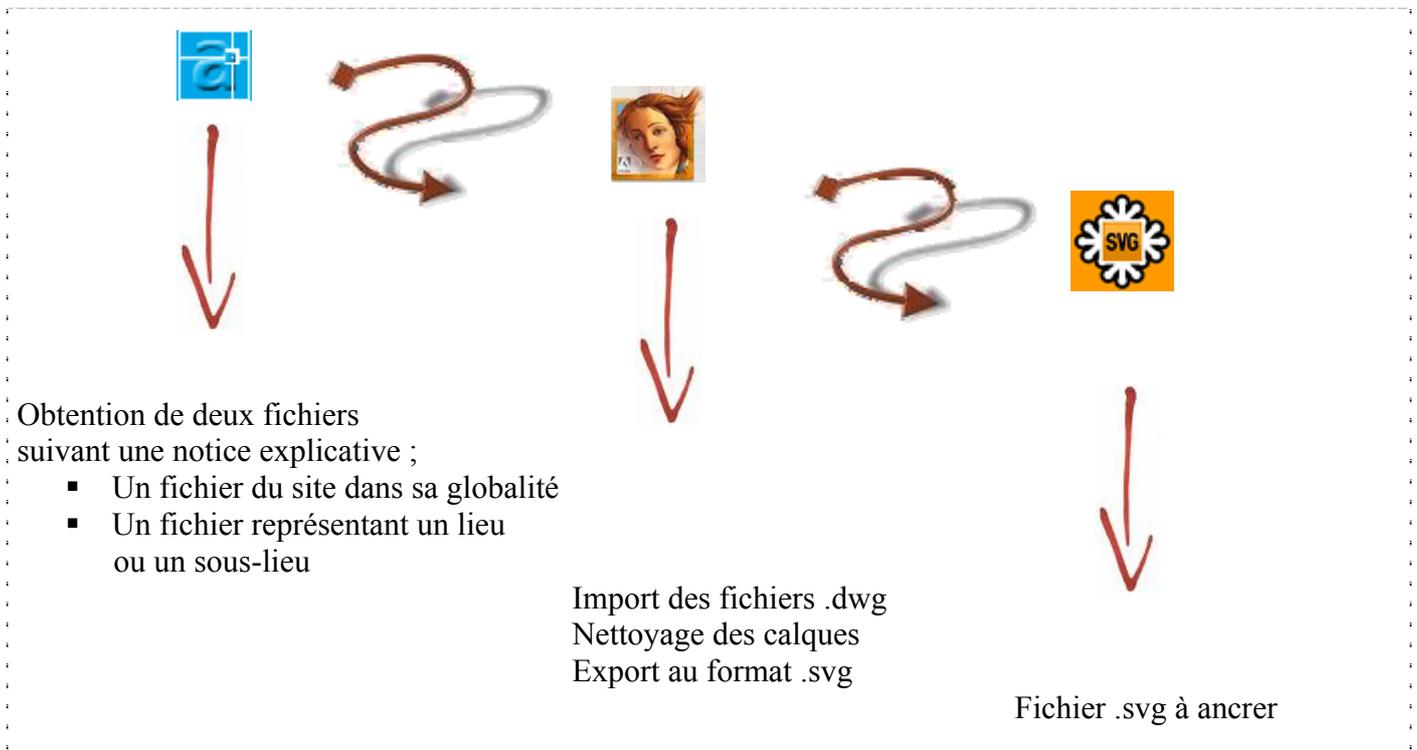


Figure 5.1 : Les différentes étapes d'obtention d'un fichier .svg

## 1.2 Ancrage du fichier .svg : ajout d'interactivité

Geroimenko et Chen [GER05] proposent un tutoriel détaillé permettant de générer des documents SVG interactifs en utilisant Adobe Illustrator, cependant, en le testant par nous même, nous l'avons trouvé assez sommaire. Adobe Illustrator permet en effet d'attacher des scripts en Javascript aux éléments graphiques dessinés, mais ne propose pas de bibliothèque de scripts. Ainsi, l'utilisateur devra soit trouver son script sur Internet, soit le créer lui-même. Ce tutoriel ne permet en fait que d'appliquer les événements sur des éléments graphiques (*onfocus*, *onmouse*, *onclick*...), ce qui est relativement aisé à insérer manuellement dans un fichier .svg. De plus, dans notre cas, il se posait aussi un problème de hiérarchie dans la structure du fichier.

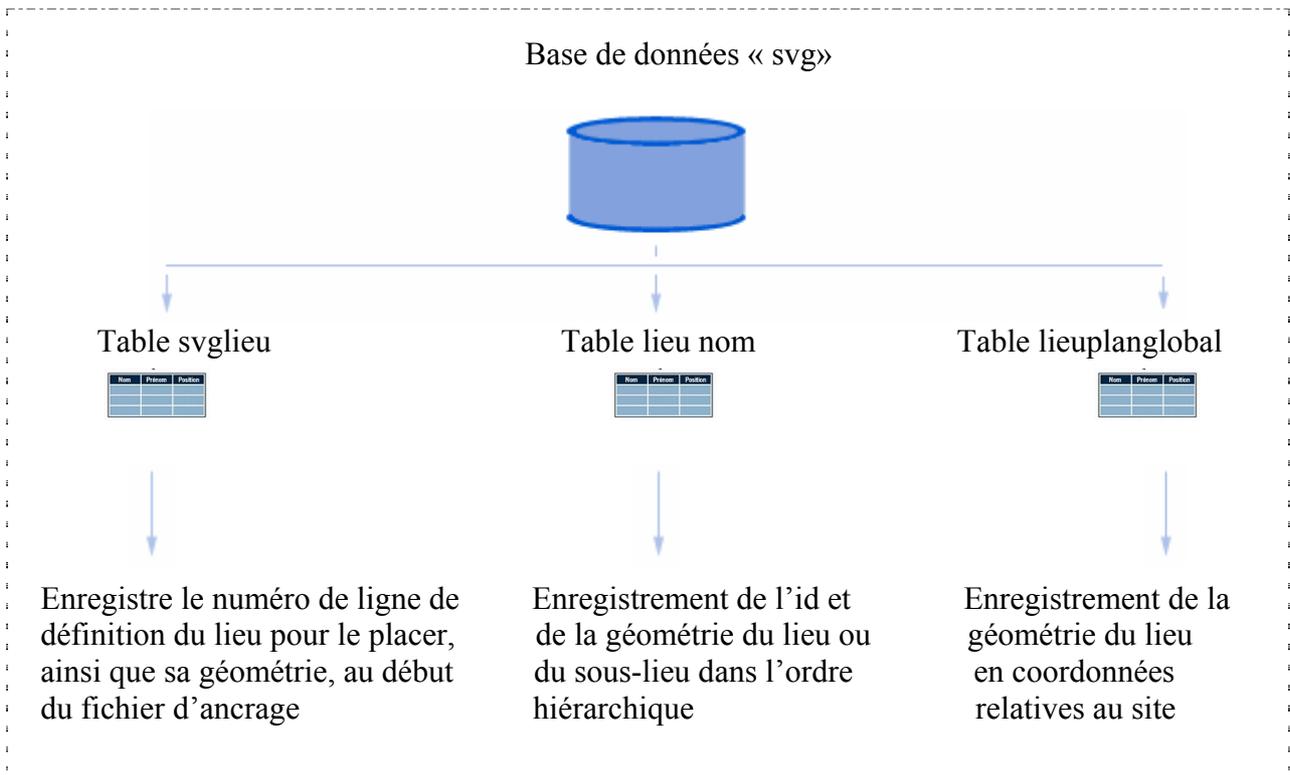
A partir du « protocole d'ancrage » proposé par Elise Meyer [2007] (cf. § 2.1.2 et 2.2.1 du chapitre **Outils utilisés pour les développements**), nous avons réalisé un programme d'automatisation des ancrages.

La structure hiérarchique du fichier SVG est importante, car rappelons le, l'utilisateur ne peut accéder aux sous-lieux sans avoir préalablement cliqué sur le lieu qui les contient. Il faut donc pouvoir discerner le lieu des sous-lieux. Or il n'est pas possible en SVG de grouper chaque sous-lieu sans grouper en même temps leurs géométries, ce qui a pour incidence un seul « path » (cf. §1.1.2 du chapitre **Outils utilisés pour les développements**). Or nous souhaitons que chaque lieu et sous-lieu soit défini par son propre « path » afin de pouvoir effectuer l'ancrage ultérieurement. De même, il ne nous était pas possible de créer une hiérarchie au sein des calques. Nous avons donc dû trouver une solution nous permettant d'identifier le lieu par rapport aux sous-lieux, et de déplacer sa géométrie (qui doit se situer avant les géométries des sous-lieux) au sein du fichier reconstitué. Ainsi, lors de la définition du nom des calques associés aux lieux et aux sous-lieux sous AutoCAD, nous avons l'obligation de nommer le calque du lieu « LieuNomdulieu » afin de pouvoir caractériser une chaîne de caractères propre au lieu. Ainsi, une fois cette chaîne repérée (nous enregistrons le numéro de ligne), nous pouvons reconstruire le fichier.

Les éléments significatifs de la géométrie des lieux et des sous-lieux sont les « path » que nous avons préalablement (avant reconstruction) enregistrés dans une base de données. Or ces « path » ont des coordonnées relatives à la fenêtre d'affichage et une entité se placera automatiquement en haut à gauche de la fenêtre, changeant par la même occasion ses coordonnées initiales. Ainsi, si nous enregistrons le fichier du site dans sa globalité ou un seul lieu du site, nous nous apercevons que les coordonnées du « path » sont différentes.

Dès lors, si nous souhaitons pouvoir afficher plusieurs lieux et sous-lieux, il faut garder les coordonnées du fichier global, et affecter aux lieux et aux sous-lieux le « path » du fichier global (sinon ceux-ci apparaîtront les uns sur les autres en haut à gauche de la fenêtre d'affichage).

Ainsi, la première base de données servant à discrétiser le fichier se nomme « svg » et est composée de trois tables : l'une servant à enregistrer le numéro de ligne du lieu, l'autre à enregistrer les composantes géométriques de chaque lieu et sous-lieu (id et « path ») et la dernière servant à enregistrer le « path » des entités du fichier global (Figure 5.2).



*Figure 5.2 : La base de données "svg"*

Ainsi, après reconstruction du fichier et ajout d'interactivité selon les mêmes procédés que ceux d'Elise Meyer [MEY07] décrits précédemment (cf. §2.1.2 du chapitre **Outils utilisés pour les développements**) un lieu ou sous-lieu sur lequel nous passons sera mis en évidence par un changement d'opacité de la zone, et un objet sur lequel nous cliquons sera d'une part mis en évidence par un changement de couleur et par l'accès interactif à la base de données du lieu ou du sous-lieu concerné dans le Système d'Information tel que le montre la figure 5.3.



```

// Les lignes de code précédées par ce code // représentent des commentaires n'influant pas sur le
programme
//Suppression de la précédente base si elle existe
$query = "DROP DATABASE IF EXISTS `svg`";
mysql_query($query) or die("Erreur dans requête de suppression de la
Base<strong>svg</strong><br>");

// creation de la bdd
if (!mysql_query("CREATE DATABASE $dbname")) die(mysql_error());
echo "success in database creation </strong><br>";

```

## Suppression puis création de la nouvelle base de données "svg"

Ici, il faut créer une base de données s'incrémentant de chacun des modèles afin de pouvoir les stocker dans le but d'en choisir certains et de les afficher simultanément.

Ainsi, dans notre programme d'ancrage, nous avons créé une deuxième base de données appelée « svgvolee » dans laquelle se trouve une table « Lieuxvolee » dans laquelle nous allons récupérer et stocker chacun des fichiers ancrés.

La difficulté réside dans le fait que nous ne souhaitons pas voir apparaître deux fois le même fichier, car cette table va aussi nous permettre de réaliser la liste des plans existants. Ainsi, nous avons dû sélectionner tous les fichiers en enlevant les doublons des résultats de la requête, effacer la table « Lieuxvolee » et la réécrire sans les doublons.

```

$query = "SELECT DISTINCT fichier,fichierplan from Lieuxvolee";
$resqueryvolee = mysql_query($query) or die("Erreur sur la requete 'Lieuxvolee'");
mysql_query("DELETE FROM Lieuxvolee");
while ( $row = mysql_fetch_array( $resqueryvolee)) {
mysql_query("INSERT INTO Lieuxvolee (id,fichier,fichierplan) VALUES ('",
$row['fichier'] . "',' . $row['fichierplan'] . "')");
}

```

## Suppression des fichiers "doublons"

La figure 5.4 illustre la table « Lieuxvolee » générée :

←T→	id	fichier	fichierplan
<input type="checkbox"/>  	10	Bains	<g id="LieuBains" style="visibility:visible;" onmo...
<input type="checkbox"/>  	11	PeristyleSud	<g id="LieuPeristyleSud" style="visibility:visible...
<input type="checkbox"/>  	12	Ailegauche	<g id="LieuAilegauche" style="visibility:visible;"...
<input type="checkbox"/>  	16	Salledapparat	<g id="LieuSalledapparat" style="visibility:visibl...

Figure 5.4 : Table « Lieuxvolee » de la base de données « svgvolee »

### 1.3.2 Création d'un programme reprenant les lieux saisis dans un formulaire par l'utilisateur

Nous avons donc créé un formulaire PHP servant d'interface à l'utilisateur lui permettant de choisir parmi les plans des lieux générés, ceux qu'il souhaite visualiser (Figure 5.5).

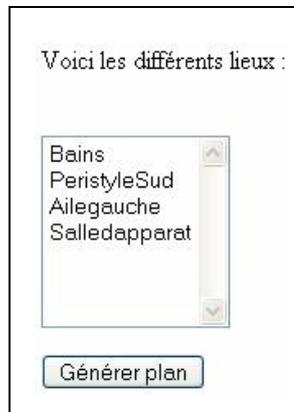


Figure 5.5 : Interface de l'utilisateur s'il souhaite afficher plusieurs lieux ancrés dans un même plan

Le bouton « Générer plan » permet d'exécuter un autre programme qui reconstruit le plan global en sélectionnant dans la base de données précédemment construite, les fichiers SVG des plans des lieux choisis (Figure 5.6).

```
if (is_array($_POST['Lieu'])) // Si des lieux sont sélectionnés
{
foreach ($_POST['Lieu'] as $lieu=>$Value) // Pour chaque lieu
{
$query = "SELECT DISTINCT fichierplan from Lieuxvolee where fichier=
'$Value'"; // Prendre le fichier associé dans la base de données
$resis = mysql_query($query) or die("Erreur sur la requete 'fichierplan'");
$nbmateriaux = mysql_num_rows($resis);
for ($j=0;$j<$nbmateriaux;$j++) {
$fichier = mysql_result($resis,$j);
$pb= str_replace ("\" url \",\"\" url \\",$fichier );
$pb2= str_replace ("$language.\"\"\", \"$language.\"\\\"\",$pb );
$pb3= str_replace ("description \"lieu\", \"description \\\"lieu\",$pb2 );
$pb4= str_replace ("\" \n parameter [\"target=droite\"]\", \"\\\" \n parameter
[\\\"target=droite\\\"]\",$pb3 );
$pb5= str_replace ("lieu (.*)\"\", \"lieu (.*)\\\"\",$pb4 );
$pb6= str_replace ("description \"souslieu\", \"description \\\"souslieu\",$pb5 );
fwrite($fichier9,$pb6. "\n");
//fwrite ($fichier9, " ] \n");
//fwrite ($fichier9, " } \n"); // Ecrire dans le fichier global
}
echo $Value."<br>";
}
}
```

Reconstruction du SVG global en fonction des différents lieux choisis par l'utilisateur

La figure 5.6 représente le plan 2D global généré si nous sélectionnons les quatre plans ancrés existants dans le système.

The screenshot shows the SIDoP web application interface. At the top, there is a navigation bar with links: A PROPOS | PLAN | AIDE ET PARCOURS UTILISATEURS | FAQ (BIENTÔT) | CONTACT | and flags for France, Germany, and the UK. The SIDoP logo is on the left, and a LOGIN button is on the right. The main content area is titled 'ECHTERNACH' and 'Accueil > Sites > echternach'. It features a 'REPRÉSENTATIONS 2D ORIGINALES' section with a prompt: 'Sélectionner le(les) document(s) qui vous intéresse(nt) : Maintenez la touche clavier "ctrl" enfoncée pour sélectionner plusieurs critères.' Below this is a list of documents: Allegauche, Bains, PeristyleSud, and Salle d'apparat. A 'Générer' button is at the bottom of the list. A blue arrow points from the list to the Firefox browser window, which displays the resulting 2D plan. The plan is a yellow floor plan with a grey rectangular area highlighted. The browser window title is 'SVG - Mozilla Firefox'. The status bar at the bottom of the browser says 'Terminé'. Three blue annotations are present: '1. sélection des lieux' points to the document list, '2. génération du plan comprenant les lieux sélectionnés' points to the 'Générer' button, and '3. plan généré "à la volée" par l'utilisateur' points to the 2D plan in the browser.

Figure 5.6 : Plan "à la volée" issu des quatre plans interactifs des lieux créés

## 5.2 Modèles 3D interactifs

### 2.1 Choix du logiciel

Tel que nous l'avons écrit précédemment (cf. introduction du chapitre **Outils utilisés pour les développements**), l'objectif est aussi d'étudier les possibilités d'exploitations de dessins 3D au format .dwg pour leur utilisation comme interface d'accès à des données sur le Web, ce qui implique une transformation des modèles en VRML.

Une des premières étapes de ce projet était le choix du logiciel de modélisation 3D, dont la fonction dans le cas présent est d'importer des fichiers .dwg, retravailler les modèles obtenus et les exporter en .wrl. Dans un souci d'accessibilité, il a fallu comparer les divers logiciels en fonction de deux critères principaux à savoir le prix et les difficultés de compréhension. Les formes des modèles furent elles aussi prises en compte, notamment leur simplicité (pas de géométrie NURBS). Enfin, les modèles seront lourds et nous souhaitons pouvoir les retoucher.

Voici un tableau non exhaustif des divers logiciels que nous avons étudié pour faire notre choix :

	<b>Google Sketchup</b>	<b>Blender</b>	<b>Sketchup Pro version 5</b>	<b>Maya version 8</b>
Prix	Gratuit	Gratuit	± 400 Euros	± 1500 Euros (selon versions)
Points forts	Possibilité de créer des modèles 3D dans Google Earth	Logiciel puissant offrant les mêmes possibilités qu'un logiciel payant	Logiciel très simple d'utilisation	Logiciel complet : possibilités de création de formes complexes de type NURBS, animations, rendu...
Points faibles	Export VRML impossible	Difficultés de compréhension du logiciel	Pas de possibilités de formes complexes de type NURBS.	Le prix fixé pour les entreprises. Il existe en effet une version personnelle gratuite.

*Tableau 5.1 : Tableau récapitulatif des divers logiciels de modélisation comparés*

Plusieurs remarques sont à effectuer sur ce comparatif :

- Tout d'abord, l'objet de ce stage n'était pas de comparer tous les logiciels de modélisation existants, et il s'agit de la raison pour laquelle seuls certains d'entre eux (les plus courants ou ceux dont la licence avait été acquise par le laboratoire) ont été étudiés. Cependant en Annexe 6 figure une liste plus exhaustive d'une comparaison de logiciels 3D effectuée par deux étudiants de l'Ecole des Ingénieurs de la Ville de Paris (Le Gravier & Moignot [LEG07]).
- La plupart des logiciels 3D sont payants et nous ne pouvions donc pas nous les procurer dans le seul but de les comparer.

Considérant tous ces paramètres, nous avons finalement choisi Sketchup version 5 comme logiciel de modélisation 3D. Nous avons importé les données .dxf dans le logiciel tel que l'illustrent les figures 5.7, 5.8 et 5.10. Nous avons par la suite retravaillé le modèle.

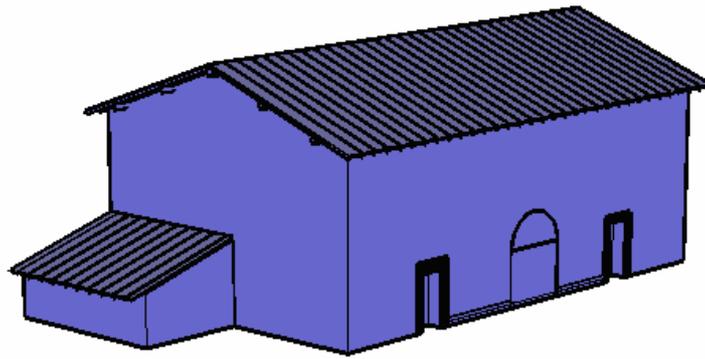
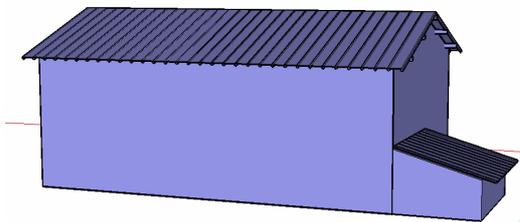


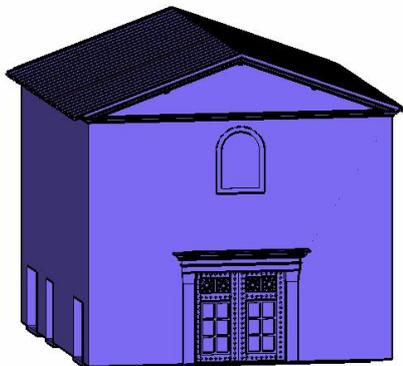
Figure 5.7 : Export du modèle 3D en format .dxf dans le logiciel Sketchup

## 2.2 Améliorations en fonctions des images de synthèse réalisées par le CRAI

Nous avons à disposition des images en format .jpg et .psd (Photoshop) réalisées sous Artlantis, logiciel permettant d'obtenir des rendus d'images de très haute qualité, représentant les différents corps de bâtiments (Figures 5.9 et 5.11). De plus, les images utilisées pour réaliser les textures étaient présentes sous leur forme brute. Il a fallu les assembler et/ou les adapter à la taille de l'objet texturé. Nous nous sommes donc inspirés de ces images pour améliorer géométriquement et « textuellement » nos modèles à l'aide du logiciel Photoshop 7.0 (Figures 5.12 et 5.13).

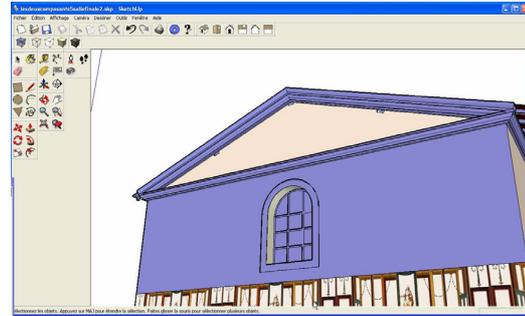
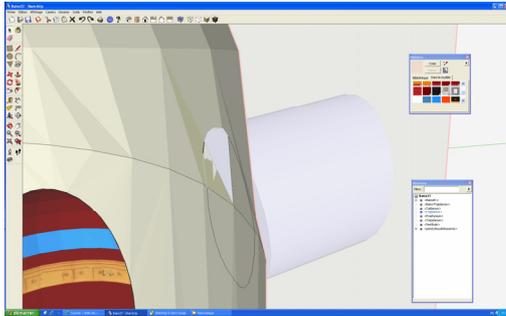


Figures 5.8 et 5.9 : Données de départ du modèle Bains : à gauche le modèle 3D brut et à droite une image de synthèse

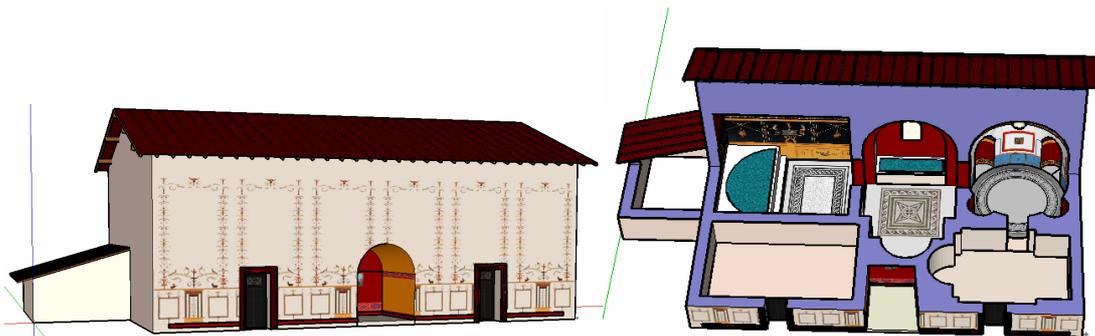


Figures 5.10 et 5.11 : Données de départ du modèle Salle d'apparat : à gauche le modèle 3D brut et à droite une image de synthèse

Possédant des images de synthèse représentant les coupes du bâtiment, il nous a semblé que créer une coupe en 3D afin de représenter l'intérieur des bâtiments était un apport. Il est possible de réaliser automatiquement des coupes sur Sketchup mais pas de les exporter en VRML. Il a donc fallu créer le modèle 3D de la coupe du modèle manuellement pour l'exporter, ce qui ne fut pas aisé, à cause notamment de certaines formes géométriques s'exportant mal depuis AutoCAD.



*Figures 5.12 et 5.13 : Améliorations géométriques des modèles 3D*



*Figures 5.14 et 5.15 : Modèles 3D finaux texturés du lieu Bains*



*Figures 5.16 et 5.17 : Modèles 3D finaux texturés du lieu Salle d'apparat*

## 2.3 Simplifications du modèle

### 2.3.1 Simplifications géométriques des formes circulaires

Les formes circulaires génèrent une multitude de faces (Figure 5.18) ayant pour incidence d'augmenter considérablement la taille du modèle Sketchup mais aussi et surtout, la taille du VRML exporté. Or, si celle-ci est supérieure à 10 Mo, le VRML affiche un message d'erreur et le modèle ne s'ouvre pas, ou très partiellement. Il était donc indispensable de simplifier le modèle tel que l'illustre l'exemple suivant :

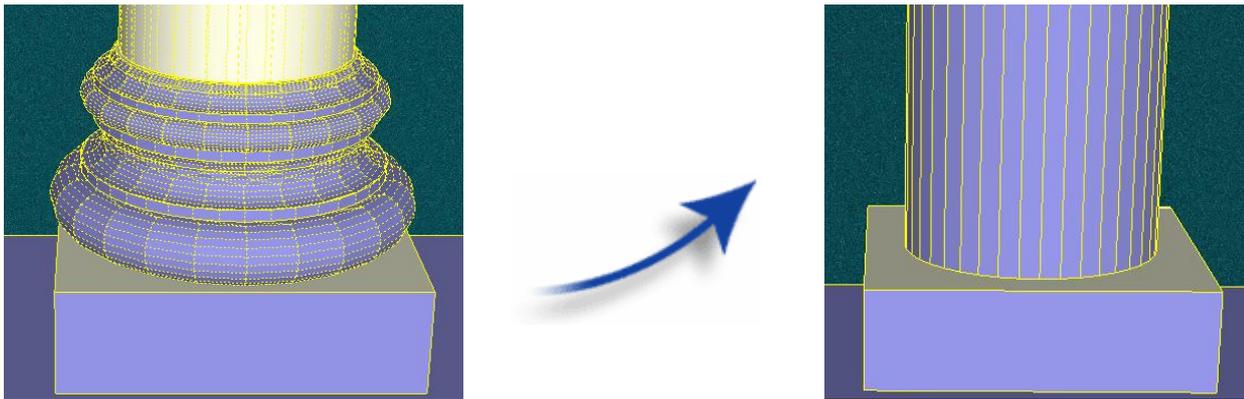


Figure 5.18: Simplification des formes circulaires générant une multitude de faces

Avant simplification de la base et du chapiteau de la colonne, le modèle exporté avait une taille d'approximativement 15 Mo ; celle-ci était de 2 Mo après l'export (Figure 5.18).

Ainsi, certains objets n'ont pas été inclus dans la scène tel que le canthare (du grec kantharos signifiant coupe avec des anses) pourtant présent sur les images de synthèse (Figure 5.11). En effet, en exportant le modèle kantharos réalisé par le MAP-CRAI, nous obtenions un modèle Sketchup de 3 Mo et un modèle VRML de 13 Mo, ayant pour conséquences un modèle incomplet (les anses ne sont pas représentées), l'apparition d'un message d'erreur et la fermeture du modèle (Figure 5.19).

Nous avons conscience que la suppression de certains éléments caractérisant le modèle dans son intégralité, ainsi que la simplification des formes circulaires en formes simplifiées ou polygonales constituent une perte d'information pour l'utilisateur. Cependant, afin de nous justifier, nous renvoyons le lecteur à la définition du modèle de Ladrière [LAD91] (cf. §1.1 du chapitre **Etat de l'Art**).

En marge de notre stage, nous avons cherché des solutions à de tels problèmes.

Nous avons finalement utilisé un principe bien connu en modélisation à savoir l'intégration d'éléments 2D. Une des solutions trouvées est la création d'un nœud *Billboard*. Un *Billboard* est une image 2D qui tourne autour d'un axe défini dans le VRML, perpendiculaire avec l'axe Camera-objet. Cependant, si l'axe de rotation est vertical, les *Billboard* ne peuvent être vus de dessus, raison pour laquelle nous avons décidé de ne pas les intégrer à notre scène.

Pour simuler le kantharos, nous avons utilisé la propriété des images .png (le logiciel Sketchup n'important pas les images .gif) intégrant la transparence pour réaliser une image de notre objet détournée, puis nous avons intercroisé ces images.

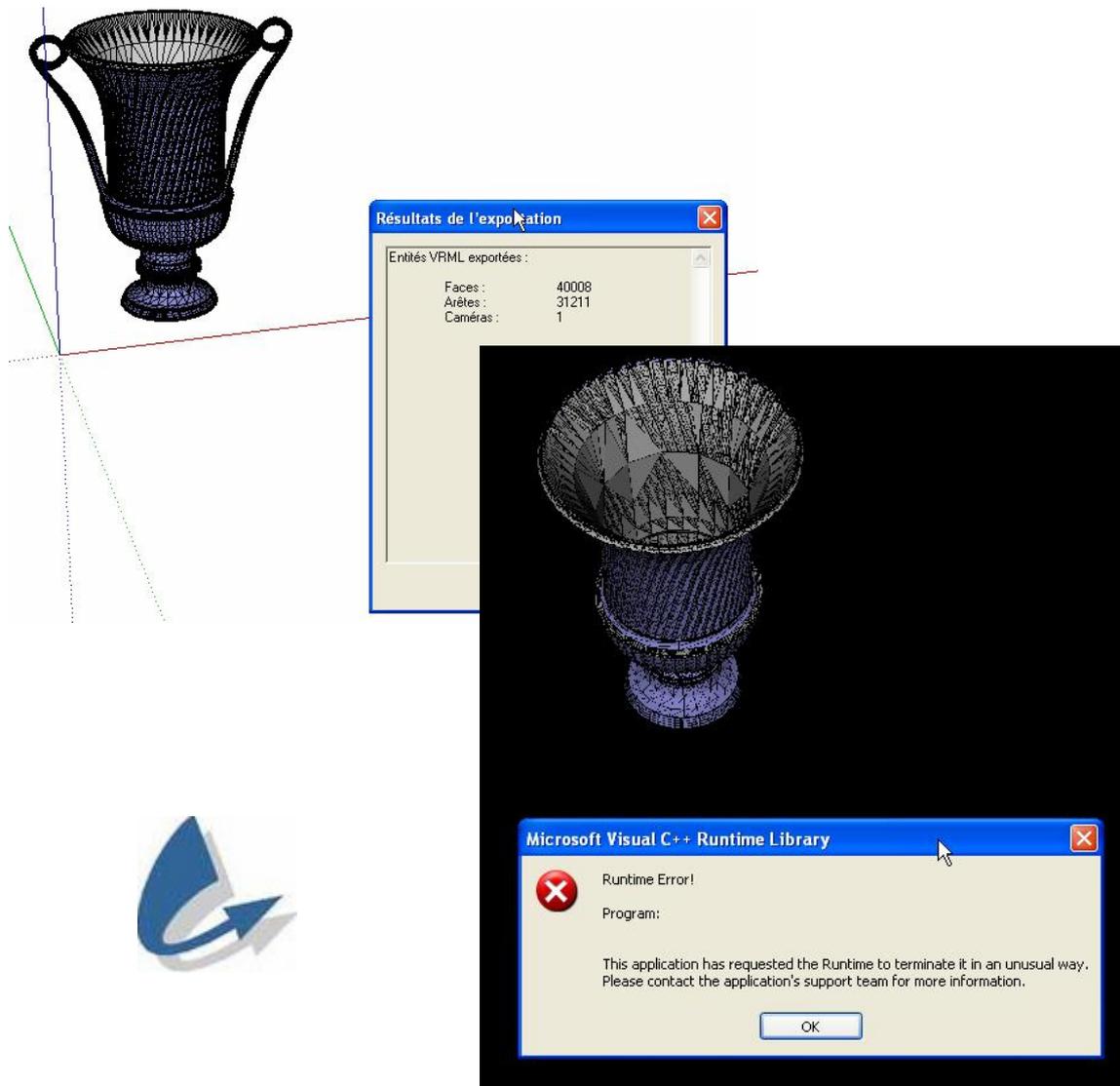
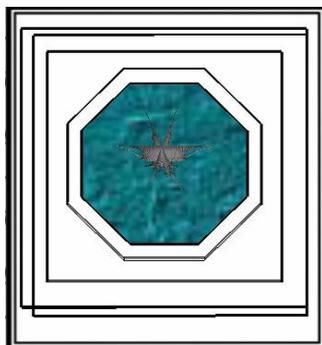


Figure 5.19 : Export du modèle Sketchup kantharos et conséquences d'un fichier VRML trop lourd

Cette solution est peu réaliste en vue de dessus (Figure 5.20) mais est tout à fait acceptable en vue de côté (Figure 5.21).



Figures 5.20 et 5.21 : Représentations du kantharos par images 2D au format .png

### 2.3.2 Simplification de l'apparence géométrique du modèle au moyen de textures

Le modèle Salle d'apparat est constitué de dix-neuf grands pilastres cannelés et dix-neuf petits pilastres cannelés. Ces constructions sont « coûteuses » en terme de faces et d'arêtes. Or elles sont visibles de la coupe, mais que partiellement de l'extérieur. Ainsi, afin de réduire la taille du fichier VRML, nous avons créé à partir de la coupe du modèle Salle d'apparat des images de l'intérieur de la salle, dont nous nous sommes servis comme textures pour le modèle extérieur de la Salle d'apparat (Figures 5.22 et 5.23).



Figures 5.22 et 5.23 : Textures réalisé à partir du modèle de la coupe de la Salle d'apparat

## 2.4 Développements précédemment cités et tests

### 2.4.1 Discrétisation du modèle VRML en fonction du nœud *Transform*

Comme nous l'avons vu précédemment (cf. § 1.2.3 du chapitre **Outils utilisés pour les développements**), nous souhaitons discrétiser le modèle VRML en fonction du nœud *Transform*, lui-même élément séparateur du lieu et des sous-lieux.

La structure hiérarchique du modèle que nous souhaitons obtenir est la suivante :

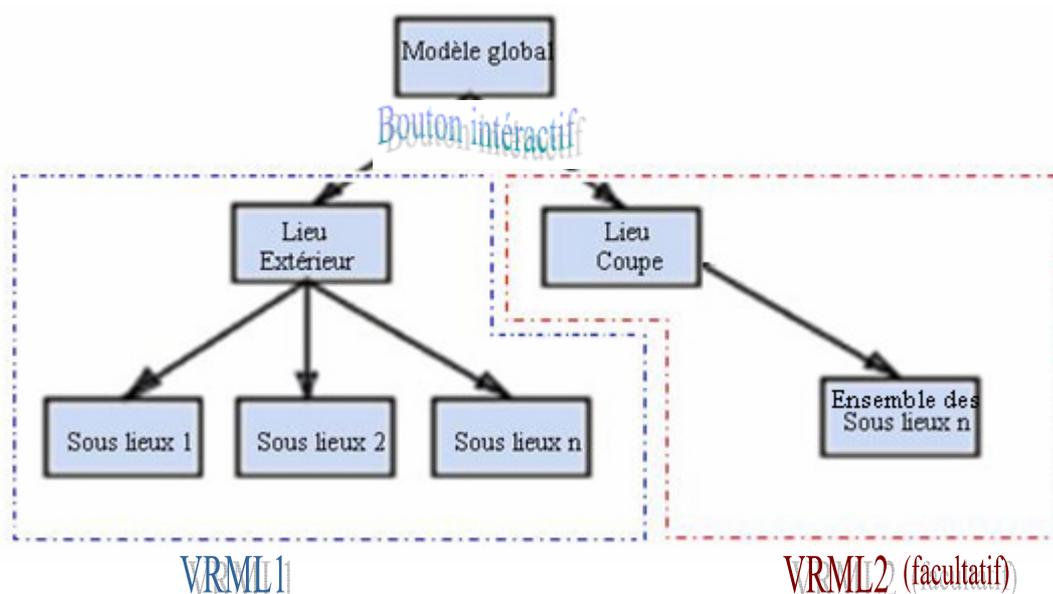


Figure 5.24: Structure hiérarchique du VRML final représentant une entité lieu et sous-lieux

En s'inspirant de programmes créés par Louis-Marie Richard, étudiant en BTS informatique et stagiaire à la Région Alsace qui a collaboré au projet de modélisation tridimensionnelle du Lycée International des Pontonniers de Strasbourg, réalisé durant l'année scolaire 2005-2006 par le laboratoire MAP-PAGE de l'INSA Strasbourg (Alby et al. [ALB07]), une classe *Transform* a été programmée.

class DebutTransform //On appelle classe la structure d'un objet, c'est-à-dire la déclaration de l'ensemble des entités qui composeront un objet. Le développement de classes objet dans certains projets peut donc apporter de nombreux avantages :

- ▶ réutilisation et portabilité facilitée du code
  - ▶ réduction du temps de développement et de débogage sur des projets complexes
  - ▶ allègement du code de certains programmes
- Ici, la classe ne sera pas réutilisée, nous aurions donc pu nous contenter d'une fonction.

```
{
var $estDebutTransform; //Toute variable est précédée par le caractère dollar ($)
var $contenu;
function testerContenu($instance)
{
if (substr(strstr($instance->contenu, "Transform {"),0,11) == "Transform {")
{
$estDebutTransform = 1;
}
else
{
$estDebutTransform = 0;
}
return $estDebutTransform;
}
}
```

## Définition de la classe "Transform"

Lorsque le programme PHP reconnaît la chaîne de caractères « Transform », il enregistre dans une table « structuredeb » le numéro de la ligne. Après avoir reconnu la structure de fin de *Transform*, les numéros de lignes de fin de *Transform* sont enregistrés de la même façon.

```
if ($transform->estDebutTransform)
{
$compteTransform++;
if ($compteTransform > 0)
{
fwrite($fichier5, $line_num_reel." Debut de Transform $compteTransform\n");
$query = "INSERT INTO structuredeb (id,debut,numeroligne) Values
('$line_num_reel','$compteTransform')";
//echo $query2;
$result = mysql_query($query) or die("Erreur sur la requete d'insertion de l'association");
if ($intTransform == 0 and $intGroup == 0 and $compteTransform > 0)
{
}
}
$intTransform = 1;
}
}
```

## Enregistrement du numéro de ligne dans une base de données

Ainsi, nous avons créé une base de données nommée « découpe » constituée de plusieurs tables parmi lesquelles nous pouvons citer « elfixes » (éléments fixes), « matériaux », « structuredeb » et « structurefin », « lieu », « souslieu », elles-mêmes composées de plusieurs champs, tel que l'illustre la figure 5.25.

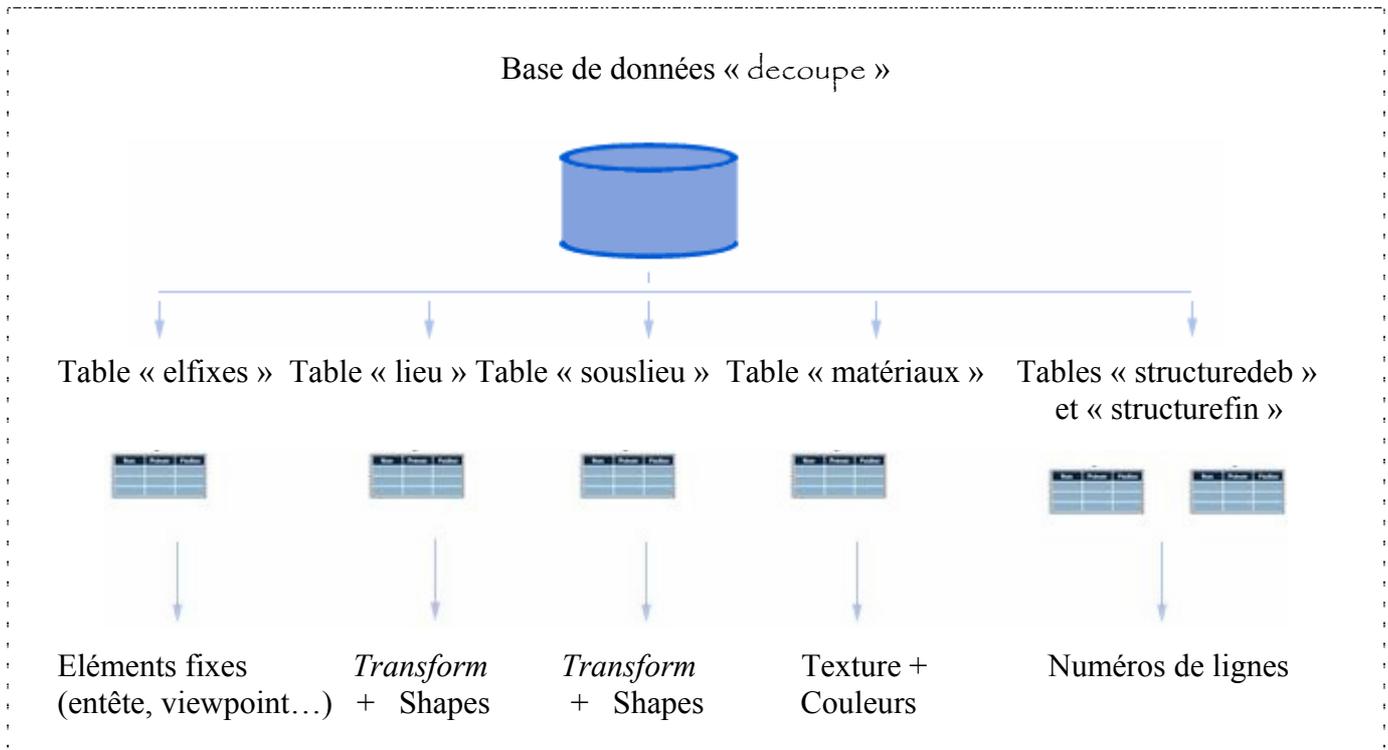


Figure 5.25 : Base de données « découpe » sous PHPMyAdmin

Chacun des fichiers VRML est partitionné à l'aide d'un programme PHP qui place automatiquement les lignes de code dans chacun des champs d'une table selon sa structure. Afin de comprendre comment s'effectue cette discrétisation, nous allons détailler ci-dessous les différentes tables constitutives de la base « découpe ».

La table « Elfixes » est composée des éléments fixes du fichier VRML à savoir l'entête, les viewpoints, c'est-à-dire des caméras de la scène 3D, le début et la fin du fichier global.

La table « matériau » est composée d'un identifiant et de l'apparence générale de la forme. L'ensemble des matériaux sera défini au début de la structure du fichier VRML reconstruit.

Comme nous l'avons écrit précédemment, les tables « structuredeb » et « structurefin » contiennent uniquement les numéros de lignes nous ayant servi à discrétiser notre modèle (Figure 5.26).

	←T→	id	debut	numeroligne
<input type="checkbox"/>		1	Debut de Transform	37916
<input type="checkbox"/>		2	Debut de Transform	54891

Figure 5.26 : Table « structuredeb »

La structure VRML d'un lieu est variable :

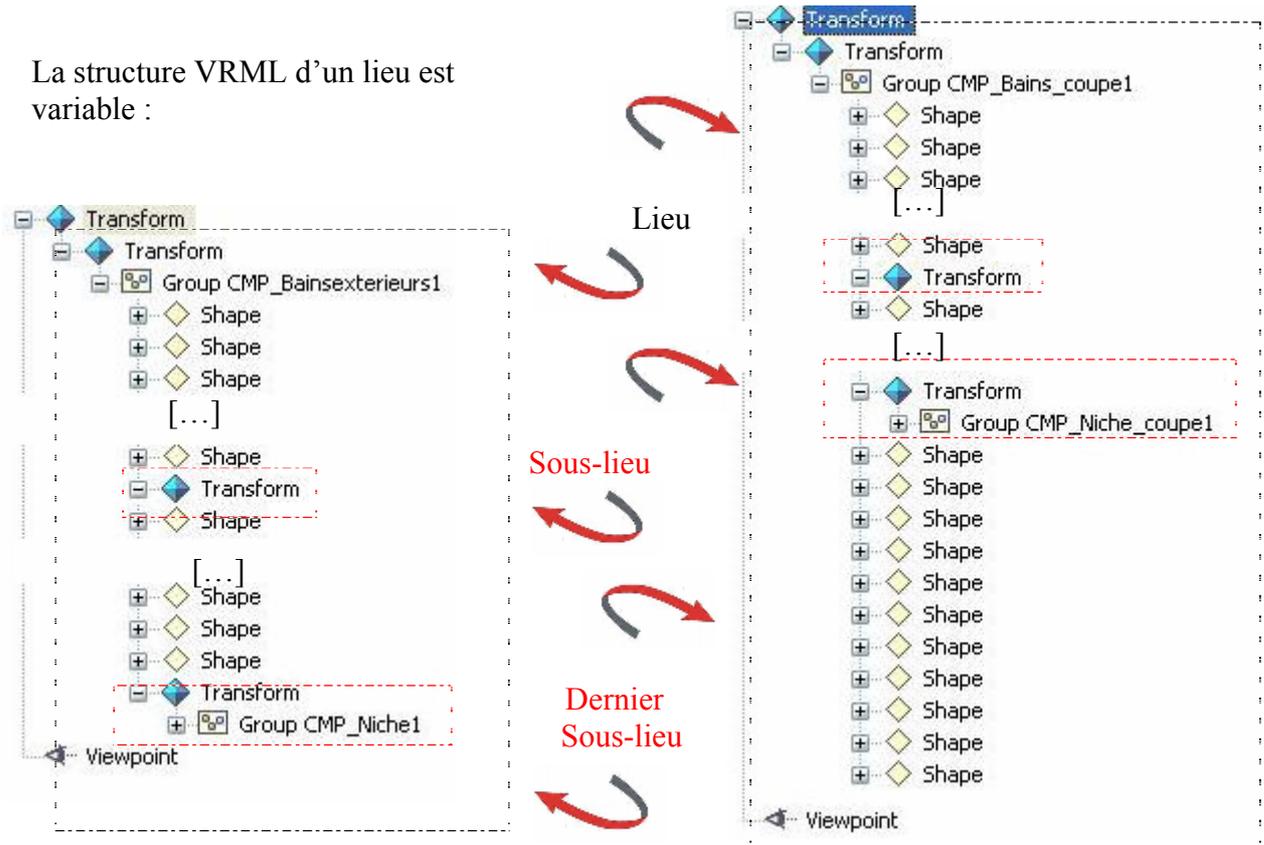


Figure 5.27 : Structures VRML de deux fichiers *Bainsexterieurs* et *Bains\_coupe* (copies d'écran obtenues sur *VrmlPad*, éditeur de texte VRML)

La table « lieu » est composée d'un identifiant (le nom du lieu), d'un début qui correspond à la définition du lieu, d'un « milieudeb » correspondant aux *Shape* comprises entre la définition du lieu et du premier sous-lieu, de « milieufin » correspondant à toutes les *Shape* situées après le dernier sous-lieu, s'il y en a un, et de la fin du fichier (parfois, et ceci est le cas pour « *Bainsexterieurs* » comme le montre la figure 5.27, ce champ reste vide).

La table « souslieu » contient tous les sous-lieux contenus dans le lieu. Elle est composée d'un identifiant (le nom du sous-lieu), de « contenudeb » correspondant à la définition du sous-lieu et de « contenufin » correspondant à toutes les *Shape* contenues dans le sous-lieu.

## 2.4.2 Ancrage du modèle

Une fois cette étape terminée, le fichier est reconstruit grâce au programme PHP qui recherche automatiquement les lignes de codes concernées dans les tables et procède à l'ancrage du lieu et des sous-lieux sous la forme d'une URL qui permettra d'accéder à la page Web requise.

### Ancrage du lieu sous forme d'URL

```
$nomduprojet= ".$nameProject.";
$ssprojet = ".$subProject.";
$paramlangue = ".$language.";
$ecriturephpdeb= "<?php echo \" \" ";
$ecriturephpfin= "\"; \n ?> \n";
$choixlieu="lieu ".$idObjectlieu;
$urllieu = "url
\\\"../..../searchObject.php?nameProject=".$nomduprojet."&subProject=".$ssprojet."&object
Name=".$lieu."&objectId=".$idObjectlieu."&language=".$paramlangue."\\\" \n";
$description = "description \\\"$choixlieu\\\" \n parameter [\\\"target=droite\\\"]";
// Description de l'URL

$query = "SELECT debut from lieu";
$resultat = mysql_query($query) or die("Erreur sur la requete 'debutlieu'");
$deblieu = mysql_result($resultat, 0);
$deblieunew = str_replace("{ ", "{\r\n", $deblieu); // \r est un retour chariot et \n un retour à la
ligne dont le but est de rendre plus lisible le code VRML
$deblieunew2 = str_replace("} ", "\r\n}", $deblieunew); // str_replace remplace une chaîne de
caractère par une autre dans un fichier
$deblieunew3 = str_replace(" geometry", "\r\ngeometry", $deblieunew2);
fwrite($fichier9, $deblieunew3. "\n");
fwrite($fichier9, "Anchor { \n");
fwrite($fichier9, $ecriturephpdeb. $urllieu. $description. $ecriturephpfin);
fwrite($fichier9, "children [ \n");
```

```
Transform {
translation -2400.919 -20472.441 -255.354 scale 1 1 1
children [DEF CMP_Bains_coupe1 Group {
children [
Anchor {
<?php echo " url \"../..../searchObject.php?nameProject=
\".$nameProject."&subProject=".$subProject."&objectName=
lieu&objectId=Bains&language=".$language."\"
description \"lieu Bains\"
parameter [\"target=droite\"] \"
?>
children [
```

Figure 5.28 : Processus d'ancrage : en haut, le programme PHP permettant de reconstruire le VRML et en bas, le code VRML obtenu

Tel que l'illustre la figure 5.28, l'ancrage se situe entre le nœud *Group* et les différentes *Shape* constituant le lieu ou le sous-lieu.

### 2.4.3 Repérage des entités lieu et sous-lieu

Nous souhaitons voir apparaître les lieux et sous-lieux lorsque le curseur passe sur une catégorie d'objet et reconnaître quel objet a été cliqué. Pour ce faire, nous allons utiliser les arêtes comme outil de repérage.

Comme nous l'avons écrit précédemment (cf. § 2.2 du chapitre **Outils utilisés pour les développements**), cette fonction de repérage nécessite la connaissance de plusieurs notions, à savoir la création d'un nœud *Touchsensor*, capteur de l'événement extérieur « clic », le fait que la géométrie *IndexedLineSet* ne soit définie que par deux des cinq paramètres de couleur du nœud *Material*, et l'écriture de scripts Java définissant les fonctions « passer sur un objet », « cliquer un objet » et de « recliquer sur un autre », toutes ces notions ayant été expliquées dans le chapitre précédent.

```
fwrite($fichier9, "DEF JeTouche".$idObjectlieu1." TouchSensor { \n" );  
fwrite($fichier9, "} \n" );
```

#### Définition du nœud "Touchsensor"

Dans le VRML de départ, les arêtes du modèle ont toutes la même « *Appearance* », c'est-à-dire le même « *Material* » définissant leur couleur (Figure 5.29).

```
appearance DEF COL_ForegroundColor Appearance {  
  material Material { diffuseColor 0 0 0  
  }  
}
```

Figure 5.29 : Illustration du nœud *Appearance* des arêtes du VRML obtenu après export

Or, si nous souhaitons pouvoir sélectionner une partie du modèle, il faut qu'elles aient un nom différent pour chaque lieu ou sous-lieu sélectionné. Il faut donc définir un nœud *Material* différent en fonction du lieu ou du sous-lieu auquel les arêtes appartiennent.

De plus, les *IndexedLineSet* n'ont que deux paramètres de couleur compatibles avec le nœud *Material* : la couleur brillante et la transparence. Ainsi, nous avons utilisé ces deux paramètres.

Voici le programme permettant de nommer les arêtes en fonction du lieu ou du sous-lieu auquel elles appartiennent et de changer leur couleur :

```

$query = "SELECT milieudeb from lieu";
$resultatmilieudeb = mysql_query($query) or die("Erreur sur la requete 'entete'");
$milieulieudeb = mysql_result($resultatmilieudeb, 0);
$milieulieudebnew4 = str_replace(" Shape", "\r\nShape", $milieulieudebnew3);
// $milieulieudebnew5 = str_replace("diffuseColor", "emissiveColor", $milieulieudebnew4);
$milieulieudebnew6 = str_replace("material Material { diffuseColor 0 0 0", "material DEF
mat".$idObjectlieu1." Material {\r\n emissiveColor 1 0 0 transparency 0.6",
$milieulieudebnew4); // Nous remplaçons la couleur principale par la couleur émise de manière à
provoquer un changement de couleur des arêtes par l'intermédiaire de scripts Java.
$milieulieudebnew7
= str_replace("COL_ForegroundColor", "COL_ForegroundColor".$idObjectlieu1."", $milieulie
udebnew6); // Nous remplaçons le nom général de l'apparence des arêtes par un nom spécifique
suivant le lieu/ sous-lieu auquel les arêtes appartiennent.
$iCpt = substr_count($milieulieudebnew7, "url");
$milieulieudebnew8 = str_replace("url [\"", "url [\"texture/", $milieulieudebnew7); // Nous
avons amélioré l'import des textures ; en effet, au lieu de se trouver « éparpillées » dans le même
dossier que le VRML, nous avons créé un dossier « texture » les regroupant. Ainsi, texture/ permet de
retrouver l'emplacement de ce dossier.
fwrite($fichier9, $milieulieudebnew8."");

```

## Appellation des arêtes suivant le lieu ou sous-lieu auquel elles appartiennent

Enfin, les chemins des événements sont décrits par la commande *ROUTE*, très importante lorsque nous désirons gérer des interactions dans la scène.

```

fwrite($fichier9, "DEF Passe".$idObjectlieu1." Script { \n");
fwrite($fichier9, "eventOut SFFloat          Couleur".$idObjectlieu1." \n");
fwrite($fichier9, "eventIn  SFFloat          estDessus".$idObjectlieu1." \n");
fwrite($fichier9, "field    SFFloat          CouleurDessus".$idObjectlieu1." 0 \n");
fwrite($fichier9, "field    SFFloat          CouleurPasDessus".$idObjectlieu1." 0.6 \n");
fwrite($fichier9, " url          [ \n");
fwrite($fichier9, " \vrmlscript: \n");
fwrite($fichier9, "function estDessus".$idObjectlieu1."(eventValue) \n");
fwrite($fichier9, " {if (eventValue == true)          Couleur".$idObjectlieu1."          =
CouleurDessus".$idObjectlieu1."; \n");
fwrite($fichier9, "else Couleur".$idObjectlieu1." = CouleurPasDessus".$idObjectlieu1.";}\n");
fwrite($fichier9, "] \n");
fwrite($fichier9, "} \n");
fwrite($fichier9, "ROUTE          JeTouche".$idObjectlieu1.".isOver          TO
Passe".$idObjectlieu1.".estDessus".$idObjectlieu1." \n");
fwrite($fichier9, "ROUTE          Passe".$idObjectlieu1.".Couleur".$idObjectlieu1."          TO
mat".$idObjectlieu1.".transparency \n");

```

## Script Java du changement de couleur lorsque l'utilisateur passe sur un objet

```

fwrite($fichier9, "DEF Click".$idObjectlieu1." Script { \n");
fwrite($fichier9, "eventIn      SFTIME touch \n");
fwrite($fichier9, "field      SFBool  etat  TRUE \n");
fwrite($fichier9, "eventOut  SFColor color \n");
fwrite($fichier9, "field      SFColor active  0 1 0 \n");
fwrite($fichier9, "field      SFColor inactive  1 0 0 \n");
fwrite($fichier9, " url [ \n");
fwrite($fichier9, " \"javascript: function touch(){ \n");
fwrite($fichier9, "etat = !etat ; \n");
fwrite($fichier9, "if (etat){ \n");
fwrite($fichier9, "color = inactive ; \n");
fwrite($fichier9, "color = active ; } \n");
fwrite($fichier9, "  else { \n");
fwrite($fichier9, "color = inactive ; \n");
fwrite($fichier9, "color = active ; } \n");
fwrite($fichier9, "}\" \n");
fwrite($fichier9, "] \n");
fwrite($fichier9, "} \n");
fwrite($fichier9, "ROUTE      JeTouche".$idObjectlieu1.".touchTime      TO
Click".$idObjectlieu1.".touch \n");
fwrite($fichier9, "ROUTE      Click".$idObjectlieu1.".color      TO
mat".$idObjectlieu1.".emissiveColor \n");

```

## Script Java du changement de couleur lorsque l'utilisateur clique sur un objet

Lorsque l'utilisateur a déjà cliqué un élément, le script « Reclick » permet de le désélectionner et de sélectionner le nouvel élément cliqué.

Ainsi, un lieu ou sous-lieu sur lequel nous passons sera mis en évidence par une surbrillance de ses arêtes, et un objet sur lequel nous cliquons sera d'une part mis en évidence par un changement de couleur de ses arêtes et par l'accès interactif à la base de données du lieu ou du sous-lieu concerné dans le Système d'Information tel que le montre la figure 5.30.

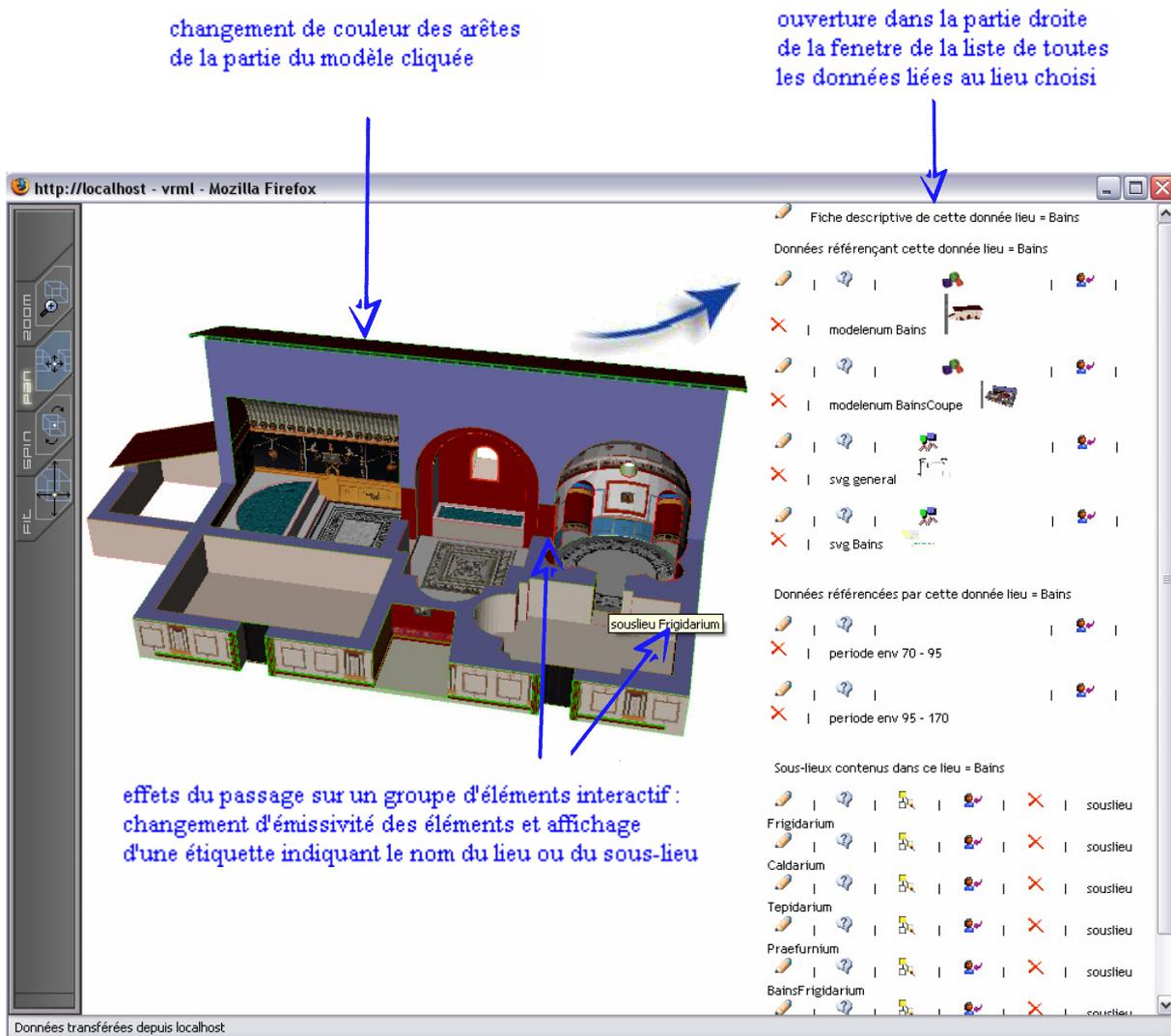


Figure 5.30 : Interface 3D interactive pour l'accès aux données liées aux lieux modélisés

## 2.5 Amélioration des paramètres de navigation dans le VRML reconstruit

Le programme étant fonctionnel, nous avons voulu améliorer le rendu. Les modèles étant lourds, nous avons paramétré des outils de navigation permettant une exploitation efficace du modèle, notamment au niveau de la vitesse de déplacement dans les nouveaux fichiers reconstruits.

```
fwrite ($fichier9,"NavigationInfo { \n");
fwrite ($fichier9,"speed 10000 \n");
fwrite ($fichier9,"type \"WALK \" \n");
fwrite ($fichier9,"} \n");
```

Améliorations des paramètres de navigation faisant suite à l'entête du fichier

## 2.6 Réalisation d'un modèle extérieur avec coupe

Une fois cette étape achevée, nous souhaitons que l'utilisateur puisse choisir interactivement entre l'extérieur du lieu et sa coupe. Cela se fait par l'intermédiaire d'un bouton qui est constamment en interface avec l'utilisateur appelé HUD (Heads Up Display) (cf. § 3.2.2 du chapitre **Outils utilisés pour les développements**).

En effet, un premier bouton à côté du lieu avait été créé mais il se posait des problèmes d'interactivité (lorsque l'utilisateur effectuait une rotation du modèle, le bouton ne lui était plus accessible) (voir l'évolution du bouton sur la figure 5.31).

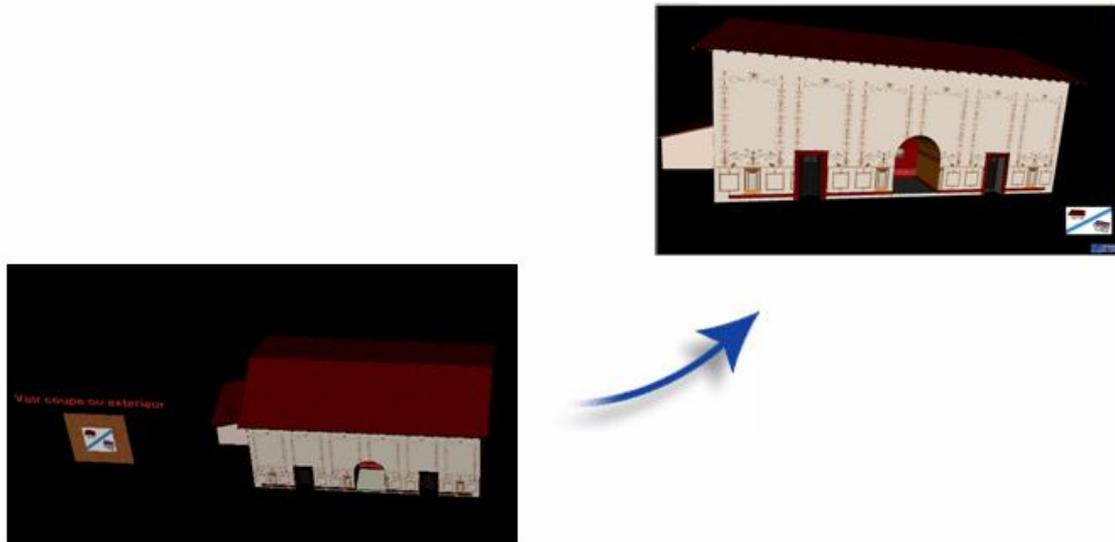


Figure 5.31 : Evolution du bouton permettant de passer de l'extérieur du lieu à sa coupe

## 2.7 Modèles à la volée

Nous avons procédé de façon similaire à celle de la réalisation de plans à la volée, la base de données « découpe » correspondant à la base de données « svg » et la base de données « volee » correspondant à la base de données « svgvolee » dans le cas des plans.

Ainsi, voici la table « Lieu » générée :

←T→	id	fichier	fichiermodele
<input type="checkbox"/>	899	Bainsexterieurs	Shape { appearance DEF COL_ForegroundColor Appear...
<input type="checkbox"/>	900	Bains_coupe	Shape { appearance DEF COL_ForegroundColor Appear...
<input type="checkbox"/>	901	PeristyleSud	Shape { appearance DEF COL_ForegroundColor Appear...
<input type="checkbox"/>	902	Aile1gauche	Shape { appearance DEF COL_ForegroundColor Appear...
<input type="checkbox"/>	903	salleapparat	Shape { appearance DEF COL_ForegroundColor Appear...
<input type="checkbox"/>	904	salledapparat_coupe	Shape { appearance DEF COL_ForegroundColor Appear...

Figure 5.32 : Table « Lieux » de la base de données « volee »

La figure 5.33 représente le modèle 3D global généré si nous sélectionnons les quatre modèles VRML existants dans le système (un des modèles n'a pas encore été texturé).

The screenshot shows the SIDoP website interface. On the left, there is a navigation menu with categories: SITES, ECHTERNACH, and VILANDEN. The main content area is titled 'MODÈLES 3D ORIGINAUX' and contains a section for selecting documents. A list of models is displayed, with 'Ailegauche', 'Bains', 'Bains\_coupe', 'PeristyleSud', 'Salledapparat', and 'Salledapparat\_coupe' selected. A blue arrow points from this list to a 3D model in a Mozilla Firefox browser window. The browser window shows a 3D architectural model of a building with a central courtyard labeled 'souslieu Bassin'. A sidebar on the right lists various data sources and periods.

1. sélection des lieux

2. génération du modèle comprenant les lieux sélectionnés

3. modèle généré "à la volée" par l'utilisateur

Figure 5.33 : Modèle VRML « à la volée » des quatre modèles générés

Nous avons, dans cette partie, présenté nos développements, nous allons désormais conclure sur l'apport de notre travail et sur les perspectives à envisager.

## 6. CONCLUSION GENERALE ET PERSPECTIVES

---

La sauvegarde du patrimoine est une des préoccupations majeures des architectes, archéologues et conservateurs. Dans ce cadre, l'utilisation d'outils informatiques est primordiale. Ils offrent de nouvelles possibilités dans le traitement des données toujours plus nombreuses que les gestionnaires de sites sont amenés à manipuler après les avoir recueillies sur le terrain. Le gain de temps apporté par l'automatisation de certains procédés, ainsi que la facilité d'utilisation de ces programmes sont des apports considérables.

Nous avons démontré l'intérêt de la visualisation lors de l'analyse d'une grande quantité de données et traité les cas des plans 2D et des modèles 3D car ils permettent pour le premier, d'appréhender un site dans sa globalité et pour le deuxième, l'ajout de possibilités de navigation et de visualisation tridimensionnelles.

Ces plans et ces modèles ont subi des transformations. En effet, afin de les utiliser comme interfaces d'accès à des données sur le Web, nous les avons converti, par l'intermédiaire de logiciels (Adobe Illustrator pour les plans et Sketchup pour les modèles), en format SVG pour les plans et VRML pour les modèles. Ainsi les données intégrées dans le Système d'Information sont en conformité avec les standards internationaux (Austin et al. [FEV02]), ce qui facilite l'accessibilité et le partage de ces données.

Nous avons inséré dans le Système d'Information des programmes PHP permettant l'obtention de plans 2D et de modèles 3D interactifs par l'intermédiaire d'un processus d'ancrage automatique. Ce processus permet le repérage des lieux et sous-lieux du site, ainsi que l'accès à leurs données descriptives. L'archéologue pourra, au moyen de notices que nous avons rédigées (cf. Annexes 5 et 7), représenter rapidement et aisément de manière interactive les plans 2D et modèles 3D du site qu'il gère.

Enfin, des interactions bidirectionnelles entre les données textuelles et bi- et tridimensionnelles font de ces représentations du site des interfaces privilégiées d'interrogation.

En attendant le moment hypothétique de la diffusion de telles technologies dans le domaine de l'archéologie, la question de la pérennité de l'information se pose. En effet, un des problèmes auquel nous avons dû faire face et dont nous n'avons pas fait mention dans notre rapport est l'évolution permanente des formats de fichier ainsi que du matériel informatique. Ainsi, bien que le projet de Vianden ait été réalisé seulement deux ans avant celui d'Echternach, nous avons rencontré des problèmes de visualisation, dus aux versions différentes (mise à jour) des outils *open source* utilisés, qui n'étaient parfois pas compatibles avec les programmes précédemment réalisés.

L'utilisation d'Internet dans la perspective de diffuser des informations, et les divers travaux de programmation menés ont orienté la réflexion sur la situation des Systèmes d'Informations dans le contexte de montée en puissance des technologies *open source*, ainsi que sur le potentiel du Web vis-à-vis de la diffusion et du partage des données. L'environnement Internet offre, en effet, actuellement des outils de développement avancés permettant d'allier souplesse, interactivité, performance et universalité.

Ces dernières années, des progrès considérables ont été réalisés par les archéologues dans l'acquisition des données spatiales, leur positionnement de manière précise à l'aide d'outils sophistiqués (GPS) et leur analyse à l'aide d'applications informatiques élaborées : les Systèmes d'Informations Géographiques. Les applications des SIG sont assez récentes (premier ouvrage d'Allen et al. [ALL90]) et vouées à un avenir prometteur dans la discipline.

En plus des informations géométriques et descriptives, ces dernières peuvent ouvrir un potentiel d'analyses, de simulations touchant le domaine de l'urbanisme et de l'aménagement du territoire, sans compter l'apport considérable en connaissances que pourra apporter l'association de ces données graphiques avec des données descriptives et documentaires pour la gestion des bâtiments, tant du point de vue technique que patrimonial.

En effet, les données recueillies sur un chantier de fouilles archéologiques possèdent toutes une composante spatiale : elles ont été retrouvées à un endroit déterminé par des composantes (X, Y, Z). Mais plus encore que leurs positions exactes, ce sont les relations spatiales entre ces entités au sein d'un même site archéologique qui sont les plus importantes pour les archéologues en vue de la reconstitution de processus culturels.

Dans ce cadre, une thèse a été proposée au laboratoire MAP-PAGE intitulée « **Implantation et gestion tridimensionnelle de modèles patrimoniaux et urbains dans un SIG 3D sur Internet** » qui pourrait peut être par la suite se coupler au Système d'Information existant.

## GLOSSAIRE

---

---

**Atrium** : Grande pièce placée derrière le vestibule, commandant toute la distribution et servant de lieu de réunion : son toit est percé en son centre d'une ouverture, le *compluvium*, auquel correspond, dans le sol, un bassin, l'*impluvium*, destiné à recevoir la pluie.

---

### Composants du modèle Bains

**Caldarium** : Salle la plus chaude de l'ensemble des Bains, surchauffée à l'époque impériale. On y accédait après être passé dans le *tépidarium*.

**Preafurnium** : Il fournit la chaleur nécessaire au chauffage de l'édifice thermal. Il constitue le foyer d'alimentation du *caldarium* (salle chaude) et du *tépidarium* (salle tiède).

**Tépidarium** : Salle tiède des thermes, où l'on se nettoyait le corps avant d'entrer dans le *caldarium*.

**Vestiaire** : Pièce dans laquelle le maître de maison et ses convives se déshabillaient et s'enduisaient d'huiles avant de pénétrer dans les salles chaudes. Le vestiaire communique directement avec le *frigidarium* (salle froide).

**Frigidarium** : Dans cette pièce, l'eau et l'air n'étaient pas chauffés. On séjournait dans cette pièce après avoir transpiré dans le *caldarium* (salle chaude) et être repassé par le *tepidium* (salle tiède). Elle comprenait également une piscine froide.

**Vestibule** : En architecture, le vestibule est une pièce par laquelle on passe pour entrer dans un édifice, une maison, et qui sert souvent de passage pour accéder aux autres pièces.

---

**Kantharos** : Désignait dans l'antiquité un vase, une fontaine en forme de coupe du centre de laquelle de l'eau jaillissait. Les canthares s'élevaient notamment au milieu de l'*atrium*.

**Niche** : Renfoncement ou cavité ménagée dans l'épaisseur d'un mur pour y loger une statue, des étagères, des bibelots... Selon sa forme et son plan, une niche est dite *carrée* (plan rectangulaire), *cintrée* ou en *cul de four* (plan courbe, le haut se terminant en trompe), *d'encoignure* (niche prise dans un angle saillant de mur), *feinte* (peinture en trompe l'œil). Une niche est dite *en tabernacle* si elle est ornée de colonnettes et d'un couronnement ouvragé. Une niche *plate* est une niche à fond vertical plat. Une niche *en tour ronde* est une niche ménagée dans le parement intérieur d'un mur courbe. Une *colonne nichée* est une colonne partiellement engagée dans un renfoncement de mur.

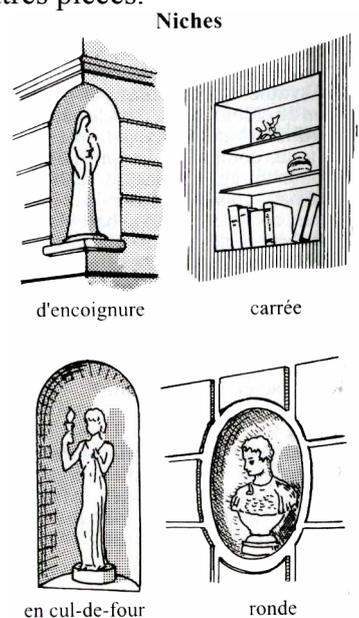


Figure G-1 : Illustration des différentes niches (De Vigan [1992])

**Ordre architectural** : Système cohérent de règles de proportion et d'harmonie qui s'appliquent à un ensemble composé d'un piédestal ou soubassement (base ou plinthe, dé et sa corniche), d'une colonne (base, fût et chapiteau) et d'un entablement (architrave, frise et corniche). Le diamètre de la colonne donne le module, unité de mesure de l'ensemble des proportions de l'ordre considéré.

Nous distinguons cinq ordres principaux, les trois premiers sont grecs ; les ordres composites et toscan ont été ajoutés par les Romains.

- **L'ordre dorique** : il s'agit de l'ordre le plus ancien ; la colonne tronconique repose, au moins à l'origine, directement sur le soubassement général de l'édifice. Le chapiteau est couronné par un large abaque carré, ou tailloir, et l'architrave de l'entablement est lisse, tandis que la frise est ornée d'une alternance de triglyphes et de métopes.
- **L'ordre ionique** : ordre dont la colonne à fût élancé repose sur une base moulurée ; le chapiteau porte d'importantes volutes, caractéristiques de cet ordre, tandis que la frise d'entablement peut être lisse ou sculptées de bas-reliefs.
- **L'ordre corinthien** : variante allégée de l'ordre ionique, caractérisée par une base moins importante, et par son chapiteau à feuille d'acanthé.
- **L'ordre composite** : ordre développé par les romains, à partir d'éléments des trois ordres précédents, nous pouvons considérer que la base est dorique, la colonne est ionique, et le chapiteau est corinthien.
- **L'ordre toscan** : cet ordre romain est un remaniement de l'ordre dorique, il s'en distingue surtout par l'absence de cannelure de son fût, et par la sobriété de son entablement.

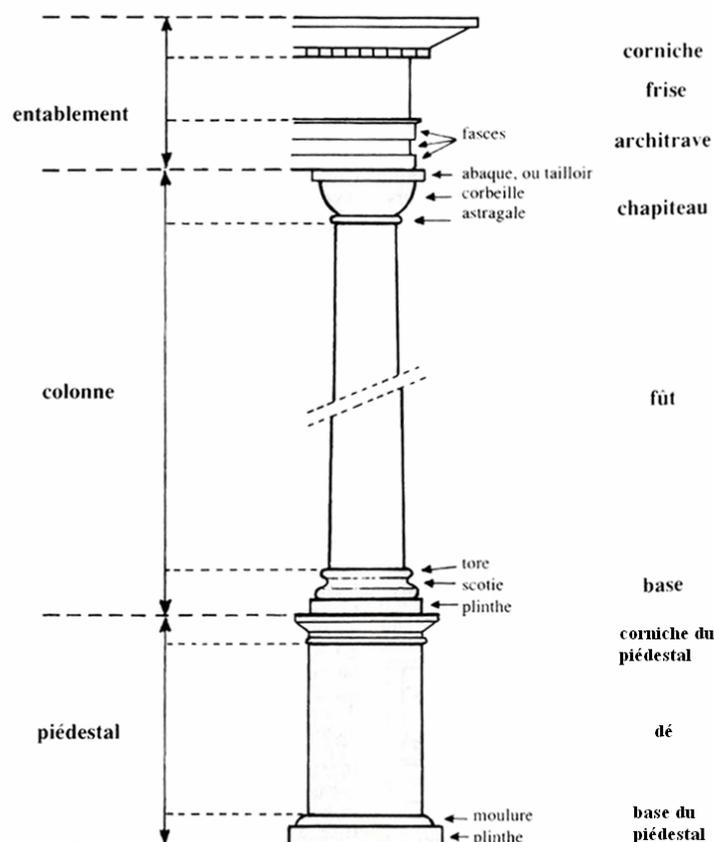


Figure G-2 : Principaux éléments d'un ordre architectural (De Vigan [1992])

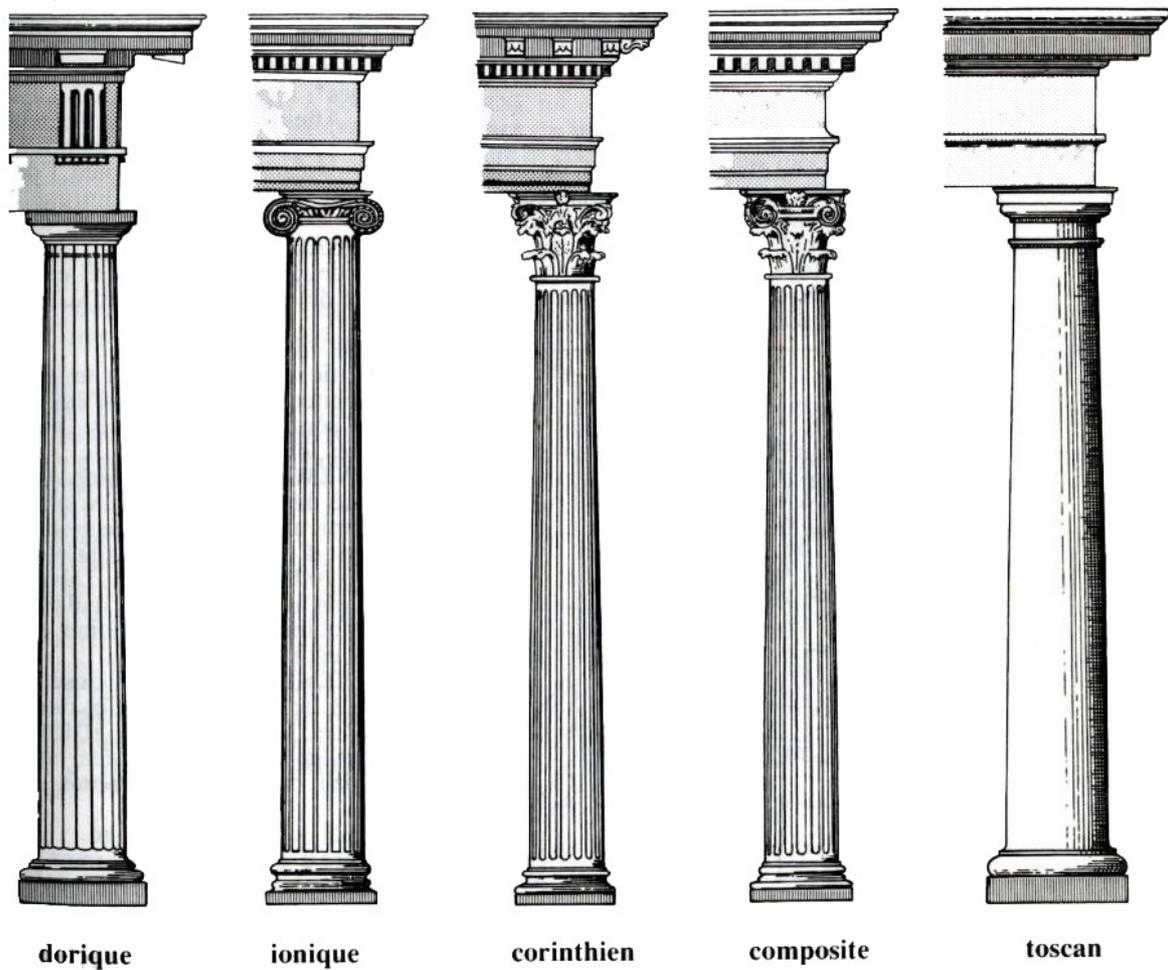
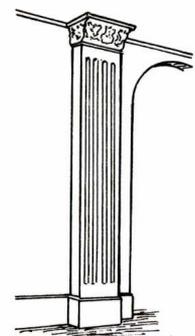


Figure G-3 : Les cinq grands ordres architecturaux grecs et romains (De Vigan [1992])

**Péristyle** : Cour ou jardin entouré de portiques, placé en arrière de l'atrium et distribuant la partie de la maison réservée à la vie privée.

**Pilastre** : membre d'architecture vertical en avant-corps d'un mur, présentant les caractères et l'aspect engagé partiellement saillant.

Dans les ordres architecturaux, son épaisseur en saillie sur le mur est en principe égale au  $\frac{1}{6}$ <sup>ème</sup> de sa largeur. Les proportions modulaires et les ornements des pilastres sont les mêmes que pour les colonnes. Le **pilastre angulaire**, ou cornier, est celui qui cantonne l'angle d'un bâtiment. Le **pilastre ébrasé** est celui qui semble replié dans un angle intérieur. Le **pilastre en gaine** est celui qui se rétrécit de sa base vers son sommet. Le **pilastre lié** est solidaire d'une colonne par sa base et/ou son chapiteau. Le pilastre est dit *ravalé* s'il comporte des moulures, ou des plaques de marbre incrustées en parement ; il est *cannelé* si son parement comporte plusieurs cannelures régulières.



Pilastre cannelé (architecture)

Figure G-4 : Illustration du pilastre cannelé (De Vigan [1992])

**Portique** : 1/ Galerie couverte, en rez-de-chaussée, ouverte sur l'extérieur, et dont la voûte est portée par des colonnes ou des arcades. Les péristyles de l'architecture grecque constituaient des portiques.

2/ Ensemble composé par une poutre et par des poteaux sur lesquels elle prend appui. Le portique multiple est une ossature composée d'un réseau de poutres et de leurs poteaux.

---

## **Informatique**

**CSS** (*Cascading Style Sheet*) : Feuilles de style permettant de modifier l'apparence et la présentation d'un élément d'une page HTML.

**DTD** (*Document Type Definition*) : Grammaire permettant de vérifier la conformité d'un document XML.

## TABLE DES ILLUSTRATIONS

<i>Figure 1.1: Carte du Luxembourg(webplaza@[2007])</i> .....	6
<i>Figure 2.1: Liens aller-retour permettant d'accéder aux diverses données</i> .....	9
<i>Figure 3.1: Les différents systèmes d'un co-navigateur VRML (Topol [2001])</i> .....	14
<i>Figure 4.1 : Structure d'un fichier SVG (Ponserre [Date inconnue])</i> .....	18
<i>Figure 4.2 : Structure d'un fichier SVG : géométrie d'un lieu repérable</i> .....	18
<i>Figure 4.3 : Exemple de graphe VRML [TOP01]</i> .....	20
<i>Figure 4.4 : Hiérarchie du nœud de regroupement Transform [TOP01]</i> .....	20
<i>Figure 4.5 : Logo WAMP 5</i> .....	21
<i>Figure 4.6 : Illustration de WAMP 5 et de ses composants</i> .....	21
<i>Figure 4.7: Fonctionnement informatique du Système d'Information (Meyer et al. [2006] modifié)</i> .....	22
<i>Figure 4.8 : Description du noeud Anchor</i> .....	23
<i>Figure 4.9 : Exemple d'un fichier VRML représentant une sphère rouge [LEN03]</i> .....	25
<i>Figure 4.10 : Copie d'écran de la structure Shape d'une arête sous VRML</i> .....	26
<i>Figure 4.11 : Routage vers des interpolateurs pour créer des animations (Topol [2002])</i> ....	27
<i>Figure 4.12 : Shape d'une géométrie IndexedLineSet définie sans nœud Appearance [KDO98]</i> .....	28
<i>Figure 4.13 : Structure du VRML dont le noeud Switch permet de passer du modèle extérieur à sa coupe</i> .....	29
<i>Figure 4.14 : Structure permettant d'obtenir un bouton HUD</i> .....	29
<i>Figure 5.1 : Les différentes étapes d'obtention d'un fichier .svg</i> .....	30
<i>Figure 5.2 : La base de données "svg"</i> .....	32
<i>Figure 5.3 : Interface 2D interactive pour l'accès aux données liées aux lieux représentés...</i>	33
<i>Figure 5.4 : Table « Lieuxvolee » de la base de données « svgvolee »</i> .....	34
<i>Figure 5.5 : Interface de l'utilisateur s'il souhaite afficher plusieurs lieux ancrés dans un même plan</i> .....	35
<i>Figure 5.6 : Plan "à la volée " issu des quatre plans interactifs des lieux créés</i> .....	36
<i>Figure 5.7 : Export du modèle 3D en format .dxf dans le logiciel Sketchup</i> .....	38
<i>Figures 5.8 et 5.9 : Données de départ du modèle Bains : à gauche le modèle 3D brut et à droite une image de synthèse</i> .....	38
<i>Figures 5.10 et 5.11 : Données de départ du modèle Salle d'apparat : à gauche le modèle 3D brut et à droite une image de synthèse</i> .....	38
<i>Figures 5.12 et 5.13 : Améliorations géométriques des modèles 3D</i> .....	39
<i>Figures 5.14 et 5.15 : Modèles 3D finaux texturés du lieu Bains</i> .....	39
<i>Figures 5.16 et 5.17 : Modèles 3D finaux texturés du lieu Salle d'apparat</i> .....	39
<i>Figure 5.18: Simplification des formes circulaires générant une multitude de faces</i> .....	40
<i>Figure 5.19 : Export du modèle Sketchup kantharos et conséquences d'un fichier VRML trop lourd</i> .....	41
<i>Figures 5.20 et 5.21 : Représentations du kantharos par images 2D au format .png</i> .....	41
<i>Figures 5.22 et 5.23 : Textures réalisés à partir du modèle de la coupe de la Salle d'apparat</i>	42
<i>Figure 5.24: Structure hiérarchique du VRML final représentant une entité lieu et sous-lieux</i> .....	42
<i>Figure 5.25 : Base de données « decoupe » sous PHPMyAdmin</i> .....	44
<i>Figure 5.26 : Table « structuredeb »</i> .....	44
<i>Figure 5.27 : Structures VRML de deux fichiers Bainsexterieurs et Bains_coupe (copies d'écran obtenues sur VrmlPad, éditeur de texte VRML)</i> .....	45
<i>Figure 5.28 : Processus d'ancrage : en haut, le programme PHP permettant de reconstruire le VRML et en bas, le code VRML obtenu</i> .....	46

<i>Figure 5.29 : Illustration du noeud Appearance des arêtes du VRML obtenu après export...</i>	47
<i>Figure 5.30 : Interface 3D interactive pour l'accès aux données liées aux lieux modélisés...</i>	50
<i>Figure 5.31 : Evolution du bouton permettant de passer de l'extérieur du lieu à sa coupe ....</i>	51
<i>Figure 5.32 : Table « Lieux » de la base de données « volee » .....</i>	51
<i>Figure 5.33 : Modèle VRML « à la volée » des quatre modèles générés .....</i>	52
<i>Figure G-1 : Illustration des différentes niches (De Vigan [1992]) .....</i>	55
<i>Figure G-2 : Principaux éléments d'un ordre architectural (De Vigan [1992]) .....</i>	56
<i>Figure G-3 : Les cinq grands ordres architecturaux grecs et romains (De Vigan [1992]) ....</i>	57
<i>Figure G-4 : Illustration du pilastre cannelé (De Vigan [1992]) .....</i>	57

## **TABLE DES TABLEAUX**

---



---

<i>Tableau 5.1 : Tableau récapitulatif des divers logiciels de modélisation comparés .....</i>	37
--	----

## BIBLIOGRAPHIE

---

---

### Ouvrages et revues :

[AAS03] AASHTO (American Association of State Highway and Transportation Officials) Task Force on Environmental Design. - «Visualization in Transportation». - 2003, p4-9/24.  
URL <http://cms.transportation.org/sites/design/docs/VisualizationGuideJuly2003.pdf>

[ALB07] ALBY (Emmanuel), MEYER (Elise), GRUSSENMEYER (Pierre). et RAMPAZZO (Mauro).-« Gestion d'ouvrage architectural : de la saisie photogrammétrique à l'interface 3D d'accès aux données patrimoniales ».- *Revue de l'Association Française de Topographie*, XYZ n°110, ISSN 0290-9057, 1er trimestre 2007, p.45-53.

[ADC04] ADCOCK (Vincent).-« Implementing an integrated SVG application for real time, dynamically generated Internet mapping». - 3rd Annual Conference on Scalable Vector Graphics, Tokyo, Japan · Sept 7-10, 2004.  
URL [http://www.svgopen.org/2004/papers/SVG\\_Open\\_Abstract/](http://www.svgopen.org/2004/papers/SVG_Open_Abstract/)

[ANT07] ANTONI (Jean-Philippe), KLEIN (Olivier), MOISY (Stéphane). – «Cartographie interactive et multimédia : vers une aide à la réflexion géographique». – “*Cybergeo*”, article n° 288, 2007.  
URL <http://www.cybergeo.eu/index2621.html>.

[ANT04] ANTONIOU (Byron), TSOULOS (Lysandro). - «Converting raster images to XML and SVG». - 3<sup>rd</sup> Annual Conference on Scalable Vector Graphics, Tokyo, Japan, 2004  
URL [http://www.svgopen.org/2004/papers/Converting\\_raster\\_images\\_to\\_XML\\_and\\_SVG/](http://www.svgopen.org/2004/papers/Converting_raster_images_to_XML_and_SVG/)

[CHA05] CHAUVIN (Sophie). –« Visualisation heuristiques pour la recherche et l'exploration de données dynamiques : L'art informationnel en tant que révélateur de sens ». - à paraître dans COMMUNICATION et LANGAGES.  
URL [http://www-bu.univ-paris8.fr/web/collections/theses/chauvin\\_sophie.pdf](http://www-bu.univ-paris8.fr/web/collections/theses/chauvin_sophie.pdf)

[DIB90] DIBIASE (David). – «Visualization in the Earth sciences ». - *Bulletin of the college of Earth and Mineral Sciences*, PSU, Vol. 59, No. 2, 1990, p13-18.

[EIT02] EITELJORG (II, Harrison), Center for the Study of Architecture. – «*The CSA CAD Guide for Archaeologists and Architectural Historians*», 2002.  
URL <http://csanet.org/inftech/cadgd/cadgd.html>

[FER02] AUSTIN (Tony), BEACH (Raphael), ALLEN (Kate), BERGSTROM (Aaron), EXON (Sally), FABRI (Marc), FENCOTT (Clive), FERNIE (Kate), GERHARD (Michael), GROUT (Catherine), JEFFREY (Stuart), PRINGLE (Mike), ROBINSON (Damian), RYAN (Nick), TERRAS (Melissa), Learning Sites, McCALL (Anthony), RICHARDS (Julian). – *Creating and Using Virtual Reality: a Guide for the Arts and Humanities [en ligne]*. -, Kate Fernie and Julian Richards (eds.), Arts and Humanities Data Service (AHDS), Univeristy of York, King's Manor, York. Endorsed by the Council for British Archaeology.  
URL [http://vads.ahds.ac.uk/guides/vr\\_guide/index.html](http://vads.ahds.ac.uk/guides/vr_guide/index.html)

**[FUC01]** Le traité de la réalité virtuelle, première édition sous la direction de Philippe Fuchs et la coordination de Guillaume Moreau. - Les Presses de l'Ecole des Mines de Paris, 2001.- 530 pages.

**[GER05]** GEROIMENKO (Vladimir)., CHEN (Chaomei). - Visualizing Information using SVG and X3D: XML-based Technologies for the XML-based Web. - Editions Springer-Verlag, 2005. - 298 pages. - ISBN 1-85233-790-7.

**[GIA01]** GIANNAKIS (Melina). - «Construction et représentation spatiale de l'interactivité numérique par manipulation d'informations 2D et 3D». - Mémoire présenté à la faculté des études supérieures de l'université de Laval pour l'obtention du grade de maître ès sciences, 2001.- p105-131/192.-

URL [http://www.casa.ucl.ac.uk/melina/site\\_work\\_2001/pages/thesis\\_melina.pdf](http://www.casa.ucl.ac.uk/melina/site_work_2001/pages/thesis_melina.pdf)

**[GRA05]** GRASSLAND (Claude)., MARTIN (Hervé)., VINCENT (Jean-Marc), GENSEL (Jérôme)., MATHIAN (Hélène)., OULAHAL (Said)., CUENOT (Olivier)., EDI (Euloge)., LIZZI (Lilliane). - «Le projet Hypercarte: analyse spatiale et cartographie interactive ». -, 2005. - 13 pages.

URL [http://www-id.imag.fr/Laboratoire/Membres/Vincent\\_Jean-Marc/papers/SAGEOHypercarte.pdf](http://www-id.imag.fr/Laboratoire/Membres/Vincent_Jean-Marc/papers/SAGEOHypercarte.pdf)

**[HAV06]** HAVEMANN (Sven)., SETTGAST (Volker)., KROTMAIER (Harald)., FELLNER (Dieter). - «On the integration of 3D Models into Digital Cultural Heritage Libraries». - Project papers from the joint event CIPA/VAST/EG/EuroMed 2006, M. Ioannides, D.Arnold, F.Niccolucci, K.Mania (Editors), Nicosia Cyprus, 30 Octobre-4 Novembre 2006, p161 à170.

**[HEI06]** HEINE (Katja)., BRASSE (Christianne)., et Wulf (Ulrike). - «**WWW**-Based Building Information System for « Domus Severiana » Palace at Palatine in Rome by Open Source Software». - VAST 2006, M. Ioannides, D.Arnold, F.Niccolucci, K.Mania (Editors), Nicosia Cyprus, 30 Octobre-4 Novembre 2006, p75 à 82.

**[JOS03]** JOSSELIN (Didier)., FABRIKANT (Sara). - «Cartographie animée et interactive ». - Revue Internationale de Géomatique, Vol.13, n° 1, Editions Hermès, 2003.

**[LAD91]** LADRIERE (Jean).-« Herméneutique et épistémologie »- Les métamorphoses de la raison herméneutique, Hommage à Paul Ricoeur, acte de colloque de Cerisy-la-Salle, 1/11 aout 1988 / sous la direction de Jean Gresich. Richard Kearney (Ed), Paris : CERF, 1991.-p 107 à 125.-(Collection Passages.)

**[LEG99]** LE GALLIC (Jean-Marc). - « La visualisation des données géographiques- L'approche VRML-Java ». - Mémoire d'ingénieur CNAM, 1999.

**[LEG07]** LE GRAVIER (Gaétan)., MOIGNON (Benjamin). - « La RV/3D appliquée au génie urbain ». -, 2007, p29-30/50.

URL [http://ww2.eivp-paris.fr/chachoua/TER2007/%5BN%B010%5D\\_RapportTERLeGravier-Moignot.pdf](http://ww2.eivp-paris.fr/chachoua/TER2007/%5BN%B010%5D_RapportTERLeGravier-Moignot.pdf)

**[LEP88]** LE PETIT ROBERT. - Dictionnaire de la langue française, dictionnaire le Robert. - Paris, 1988,1211p.

**[LEP96]** LE PETIT LAROUSSE. - Le petit Larousse illustré en couleurs-. Larousse, Paris, 1996, 1784p.

[MAP06] MAP-PAGE (Laboratoire). – «Rapport sur le projet de modélisation tridimensionnelle du lycée international des Pontonniers de Strasbourg ». - Rapport interne du laboratoire, p11/19.

[MCC87] McCORMICK (Bruce)., DEFANTI (THOMAS)., BROWN (Maxine). – « *Visualization in Scientific Computing*, Computer Graphics». - Special issue ACM SIGGRAPH, vol. 21, n°6, 1987.

[MET81] METZLER (Jeannot)., ZIMMER (Johnny)., BAKKER (Lothar). – «Ausgrabungen in Echternach». – Luxembourg : Publication du Ministère des Affaires Culturelles et de la ville d'Echternach, 1981. - 394 p. -

[MEY05] MEYER (Elise). – « Acquisition et imagerie en archéologie. Vers une plate-forme liant informatique et données patrimoniales ».- Rapport interne rendu au MAP-CRAI dans le cadre d'une thèse, 2005.

[MEY06a] MEYER (Elise)., GRUSSENMEYER (Pierre)., Perrin (Jean-Pierre). – « First proposals for a web-based information system in archaeology: storage and interactivity for the preservation and the handling of Cultural Heritage data». - Third Taiwanese-French Conference on Information Technology, Yih-Kuen Tsay et Samuel Cruz-Lara, Nancy, 28-30 Mars, p.195 à 211.

[MEY06b] MEYER (Elise)., GRUSSENMEYER (Pierre)., PERRIN (Jean-Pierre)., DURAND (Anne)., DRAP (Pierre). – «Intra-site Level Cultural Heritage Documentation: Combination of Survey, Modeling and Imagery Data in Web Information System». - Short papers from the joint event, Project papers from the joint event CIPA/VAST/EG/EuroMed 2006, M. Ioannides, D.Arnold, F.Niccolucci, K.Mania (Editors), Nicosia Cyprus, 30 Octobre-4 Novembre 2006, p.129 à 133.

[PAQ06] PAQUET (Eric)., LAHANIER (Christian)., PITZALIS (Denis)., AITKEN (Geneviève)., PETERS (Shawn) et VIKTOR (Herna Lydia). – « Content-based Indexing and Retrieval of Cultural Heritage Data: an Integrated Approach to Documentation with Application to the EROS Database». - Project papers from the joint event CIPA/VAST/EG/EuroMed 2006, M. Ioannides, D.Arnold, F.Niccolucci, K.Mania (Editors), Nicosia Cyprus, 30 Octobre-4 Novembre 2006, p.270 à 276.

[PER95] PEROUSE DE MONTCLOS (Jean-Marie). - «ARCHITECTURE vocabulaire ». – Paris : Imprimerie nationale Editions, ISBN 2-11-080961-2 (Brochés) et 2-11-080962-0 (Reliés), 1995.- p93 - 470/ 622.-

[PET06] PECHLIVANIDIOU-LIAKATA (Anastasia)., MIKROU (Stylianos)., ZEREFOS (Stylianos Christos). – «Digital Preservation and Promotion of the Architectural Heritage of Attica, Project papers from the joint event CIPA/VAST/EG/EuroMed 2006». - M. Ioannides, D.Arnold, F.Niccolucci, K.Mania (Editors), Nicosia Cyprus, 30 Octobre-4 Novembre 2006, p.279 à 283.

[PIM99] PIMPAUD (Alban-Brice). –« *Les applications de l'infographie 3D à l'archéologie* ». - Mémoire de maîtrise d'histoire de l'art et d'archéologie soutenu à l'Université de Paris-X-Nanterre, sous la direction de Mme A.-M. Guimier-Sorbets, 1999.  
URL [http://pimpaud.free.fr/wordpress/wp-content/uploads/2006/05/maitrise\\_pimpaud.pdf](http://pimpaud.free.fr/wordpress/wp-content/uploads/2006/05/maitrise_pimpaud.pdf)

[PON ?] PONSERRE (Sylvain). – « Possibilités d'analyses cartographiques et statistiques avec les outils open source » - Présentation Powerpoint.

URL [http://www.sitg.ch/fichier/commission\\_technique/Outils\\_Open\\_Source.pdf](http://www.sitg.ch/fichier/commission_technique/Outils_Open_Source.pdf)

[TOP01] TOPOL (Alexandre). –« VRML : étude, mise en oeuvre et applications ». - Mémoire d'ingénieur CNAM, 2001.

URL <http://cedric.cnam.fr/PUBLIS/RC263.pdf>

[VIG92] VIGNAN (de, Jean). – DICOBAT dictionnaire général du bâtiment. - Ris-Orangis : Editions Arcature, 1992, ISBN 2-9504805-0-0, p563, p577-579, p624, p664 / 953

#### Documents de formation :

[KOE00] KOEHL (Matthieu). - « Introduction aux SIG ». -, Institut National des Sciences Appliquées de Strasbourg, Filière Topographie, 2000.

[LEN05] LENOIR (Julien). – « VRML 97 ». -, Laboratoire d'Informatique Fondamentale de Lille, pas de filière précisée, 2003.

URL [http://www2.lifl.fr/~lenoir/enseignements/2003-2004/VRML/vrml97\\_2k3\\_2k4.ppt](http://www2.lifl.fr/~lenoir/enseignements/2003-2004/VRML/vrml97_2k3_2k4.ppt)

#### Sites Internet :

[API04] Apia@[2004], VRML le langage de modélisation de la réalité virtuelle

URL <http://apia.u-strasbg.fr/vrml/>

[AUS06] AUSONIUS@ [2006], AUSONIUS - UMR 5607

URL <http://www-sira.u-bordeaux3.fr/indexframe.html>

[CIS ?] CISAR @ Date inconnue

URL <http://www.tu-cottbus.de/cisar/cisar/beschreibung.html>

[CVL03] Cultural VR Lab@ [2003]

URL <http://www.cvrlab.org/>

[ERO05] EROS-3D@ [2005], Intégration multi-modale et recherche dans des bases de données de modèles 3D d'oeuvres d'art

URL <http://eros3d.ensea.fr/>

[FOV03] FOVEA@ [2003], Fouille Virtuelle d'Environnement Paléo-anthropologique

URL <http://foveaproject.free.fr/>

[KDO98] KDO@ [1998], Cours VRML de Base

URL <http://kdo.chez-alice.fr/vrml/sommaire.htm>

[INR07] La villa Antique de Taden @ [2007], Inrap Institut national de recherches archéologiques préventives.

URL <http://www.inrap.fr/taden/Taden.pdf>

[LGD02] Le Grand Dictionnaire@ [2002], Le Grand Dictionnaire : office québécois de la langue française 2002

URL <http://www.granddictionnaire.com>

- [LMO94] L'informatique et l'Archéologie, Beck P., dir., *Le Médiéviste et l'ordinateur*, 29, 1994.N° ISSN 0223-3843. [Online]  
URL <http://lemo.irht.cnrs.fr/29/MO29/HTM>
- [MAR07] MARVIN @ [2007], Méthode d'Administration de Réalité Virtuelle pour l'INternet  
URL <http://users.etu.info.unicaen.fr/~jzerafa/presentationMP/s5-intro/s5-intro.html>
- [DUF06] Pascal Dufresne, « Un nouvel outil pour l'archéologie : le jeu vidéo », *Le Médiéviste et l'ordinateur*, 2006 (Varia) [En ligne]  
URL <http://lemo.irht.cnrs.fr/jeu-video.htm>
- [MAR ?] Sebastien Marnier@ Date inconnue, 3Dnet Animations 2D et 3D pour internet.  
URL <http://grandm.free.fr/index2.html>
- [JAN ?] Visite d'un site archéologique avec base de données : Date et auteur inconnus  
URL <http://www.temple-de-janus.net/>
- [VVA07a] Visites virtuelles pour l'archéologie@ [2007]  
URL <http://archeus.virtual.club.fr/>
- [VVA07b] Visites virtuelles de sites archéologiques @ [2007]  
URL <http://www.learningsites.com/>
- [WEB04] Web3D@ [2004], WEB3D-fr  
URL <http://www.web3d-fr.com/>
- [WEB07] Webplaza@ [2007], Amitié Portugal Luxembourg, asbl- section d'Echternach  
URL [webplaza.pt.lu/aaplecht/PageAAPL1.htm](http://webplaza.pt.lu/aaplecht/PageAAPL1.htm).

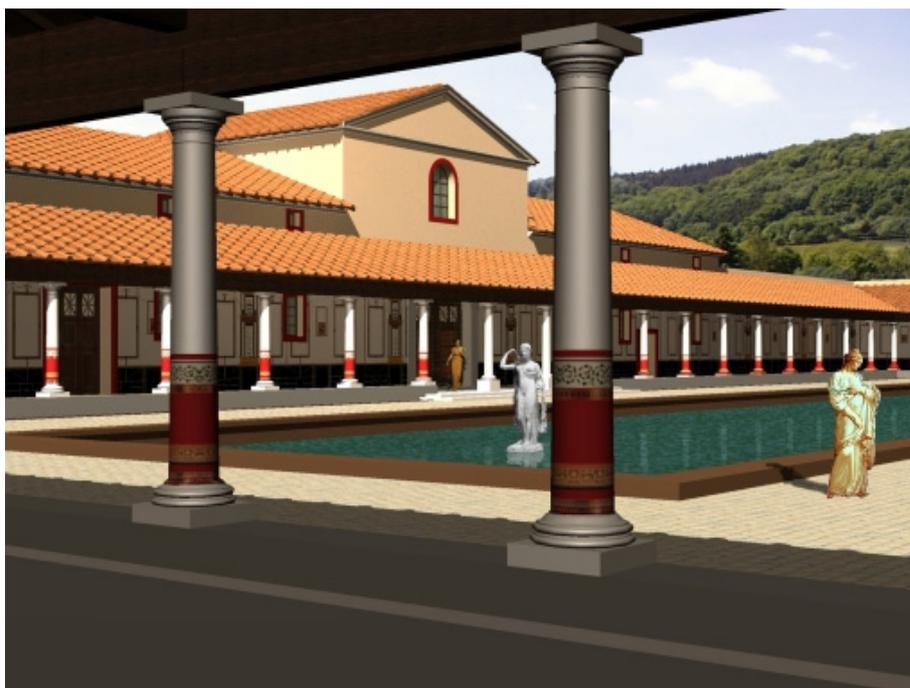
École Nationale Supérieure d'Architecture de Nancy  
Université Henri Poincaré (Nancy1)  
Institut National Polytechnique de Lorraine  
École Nationale Supérieure d'Architecture de Strasbourg  
Institut National des Sciences Appliquées de Strasbourg

---

## **Mémoire de master Design Global, spécialité « Modélisation et Simulation des Espaces Bâtis »**

**RÉALISATION DE REPRÉSENTATIONS BI- ET TRIDIMENSIONNELLES  
INTERACTIVES POUR UN SYSTÈME D'INFORMATION ARCHÉOLOGIQUE**

### **-ANNEXES-**



Présenté le **28 Septembre 2007** par **Céline PIACENTILE**

Laboratoire d'accueil :

**Equipe Photogrammétrie  
Architecturale et GEomatique  
INSA Strasbourg  
24, boulevard de la Victoire  
67084 STRASBOURG CEDEX**

Sous la direction de :

**Mlle Elise MEYER  
Doctorante MAP-CRAI**

**En couverture,**

Image de synthèse réalisée par le laboratoire MAP-CRAI à partir des hypothèses de restitution du site d'Echternach

## TABLE DES ANNEXES

---

---

<b>Annexe 1</b> : Article sur la restitution de la villa romaine d'Echternach .....	69
<b>Annexe 2</b> : Brève introduction à PHP .....	74
<b>Annexe 3</b> : Brève introduction à MySQL.....	79
<b>Annexe 4</b> : Comparaison des différents logiciels d'édition de SVG.....	81
<b>Annexe 5</b> : Notice explicative de la réalisation de plans 2D sous AutoCAD et Adobe Illustrator .....	85
<b>Annexe 6</b> : Comparaison des principaux logiciels de modélisation 3D disponibles sur le marché .....	91
<b>Annexe 7</b> : Notice explicative de la réalisation de modèles 3D sous Sketchup .....	95

## Annexe 1 : Article sur la restitution de la villa romaine d'Echternach

Didier Bur, J-P Perrin

Participants : D. Bur, D. Laroche, J-P. Perrin [CRAI], J. Metzler, J. Zimmer [SSMNL]

### Bibliographie succincte

Ackerman, J. [1997], *La villa, de la Rome antique à Le Corbusier*. Ed. Hazan. EAN 9782850255571, Référence 15823-932, 395 p.

Gros, P. [2001], *L'architecture romaine du début du IIIe siècle av. J.-C. à la fin du Haut-Empire 2, Maisons, palais, villas et tombeaux*. Ed. Picard, Collection Les Manuels d'art et d'archéologie antiques. ISBN 2-7084-0533-0 ; 527 p.

Metzler, J., Zimmer, J. et Bakker, L. [1981], *Ausgrabungen in Echternach*. Publication du Ministère des Affaires Culturelles et de la ville d'Echternach, Luxembourg. 394 p.

Miron, A ., Brück, J ., Frey, M ., Kühnen, K. [1997], *Das Badegebäude der römischen Villa von Borg*. Ed. Kulturstiftung für den Landkreis MerzigWadern.

Petit, J-P ., Schaub, J. [1995], *Bliesbrück-Reinheim, parc archéologique européen*. Ed. Imprimerie Nationale. ISBN 2-87772-298-8, 224 p.

Poisignan, V. [1987], *Implantation et esquisse d'une typologie des villas gallo-romaines en Alsace et en Lorraine*. Cah. Alsac. Arch. Art hist. 30, p103 à 170.

Van Ossel, P. [1985], *Quelques apports récents à l'étude de l'habitat rural gallo-romain dans la région mosane*. Etudes Classiques 53, p79 à 90.

Ward-Perkins, J. [1993], *Architecture romaine*. Ed. Gallimard, Collection Histoire de l'Architecture. ISBN-10 2070150151, ISBN-13 978-2070150151, 205 p.

# Restitution de la villa gallo-romaine d'Echternach

Didier Bur, J-P Perrin

Participants : D. Bur, D. Laroche, J-P. Perrin [CRAI], J. Metzler, J. Zimmer [SSMNL]

## La reconstitution comme composante d'une muséographie

Lorsqu'au cours des années 1975 et 1976, le Service des Sites et Monuments Nationaux du Luxembourg entreprit des fouilles sur le site de la villa gallo-romaine d'Echternach, les différents relevés, stratigraphies, fragments, fournirent quantité d'informations qui furent ensuite analysées et rassemblées en 1981 dans l'ouvrage de référence traitant de ce site : « Ausgrabungen in Echternach » (Fouilles d'Echternach).

Les techniques et matériels informatiques de l'époque étaient loin de permettre une quelconque reconstitution spatiale par ordinateur qui aurait pu illustrer ou étayer les hypothèses volumétriques qui furent alors élaborées. Le site archéologique fut au fil des années quelque peu délaissé, puis dans le cadre de l'aménagement du pôle touristique environnant, une remise en valeur a été décidée. Elle se caractérise principalement par la construction d'un musée aux abords même des vestiges.

L'idée centrale de ce musée est de présenter aux visiteurs une approche de la vie quotidienne dans ce palais-villa à la fin du premier siècle après. J-C : maquettes, ambiances, objets, présentations des différentes pièces d'habitation et occupations qui s'y déroulaient.

Le Service des Sites et Monuments Nationaux du Luxembourg a confié au CRAI la réalisation de tous les documents numériques devant être présentés au public, que ce soit dans le musée ou sur des panneaux in situ, permettant aux visiteurs de faire un rapport immédiat entre ce qui subsiste et les hypothèses de reconstitution.

## Architecture

Comme beaucoup de villas aux alentours et du Nord de l'empire (Abbeville, Estrées-sur-Noye, Borg, Bliesbrück) le plan d'origine de la maison est composé de corps de bâtiments symétriques de part et d'autre d'un axe traversant une cour, ici entourée d'un portique et agrémentée d'un bassin. Les ordres présents à Echternach sont l'ordre ionique, corinthien et surtout toscan, l'ensemble de l'architecture évoquant le traité de Vitruve.

La partie centrale de la villa (118m x 62m) s'organise autour d'une salle d'apparat monumentale flanquée de pièces à vivre, chambres et cuisines. De chaque côté de ce bâtiment principal se trouvent deux cours à péristyle menant pour l'une à un très grand triclinium et pour l'autre à l'ensemble balnéaire (Fig. 1).



Figure A1-1 : Vue aérienne de la villa

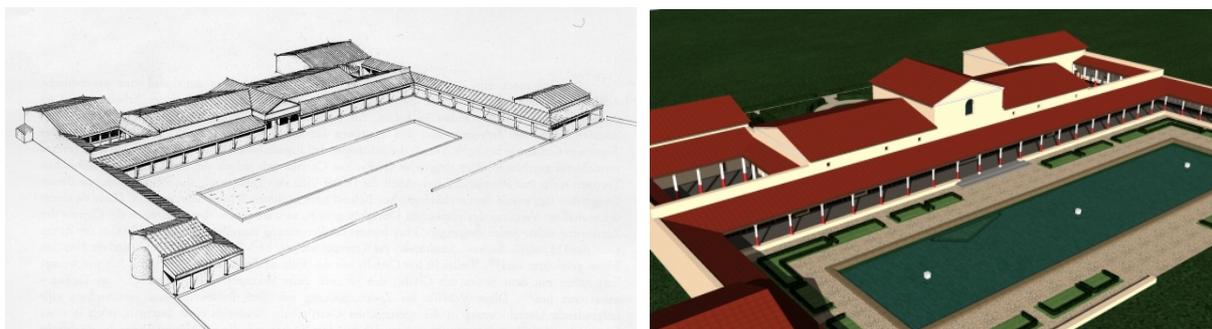
Aux extrémités Nord du portique entourant la cour se trouvent deux petits bâtiments dont l'un à l'Est semble être un triclinium d'été et l'autre à Ouest dont l'usage reste indéterminé. Plus loin ont été localisés d'après des photographies aériennes d'autres bâtiments : logements d'esclaves, granges, écuries, etc. La villa (située sur la voie romaine menant à Trèves) était sans doute destinée à l'élevage de chevaux vendus à l'armée romaine. La construction de la villa a été datée des années 70 ap. JC (soit peu avant la destruction de Pompeï). Nombre de fragments indiquent qu'une partie de la décoration en marbre avait été importée de Rome, ce qui laisse à penser que l'architecture et la décoration étaient fortement influencées par celle de l'Empire. Enfin, le sol de plusieurs pièces était recouvert de mosaïques, de même que les parois des bassins ; la cour et son bassin piscicole (ce dernier fonctionnant toujours) ont été restitués par référence à la villa de Fishbourne (GB).

## Méthode

Outre les hypothèses et propositions de reconstitution existantes, les plans de relevés originaux, les sources principales d'informations et de références se situent bien dans l'architecture et la décoration pompéienne du 1<sup>er</sup> siècle ainsi que dans les villas similaires de la Gaule du Nord (Nennig, Borg, Reinheim...).

Si elle n'est plus originale ni innovante, une reconstitution virtuelle a cependant le mérite de « forcer » son auteur à donner une mesure à toute chose, à compiler et croiser exhaustivement la documentation dont il dispose et ce faisant, de mettre à jour des erreurs, les oublis, qui n'auraient pas été détectés lors d'une reconstitution « traditionnelle » et même de soulever des questionnements restés jusqu'alors inexplorés : nature des matériaux employés, méthodes de mise en œuvre (systèmes constructifs des charpentes, dimensions et proportions des ouvertures...).

Ainsi, la reconstitution originale montrait le portique principal interrompu dans sa partie centrale par un portique corinthien plus haut, supportant la toiture de la salle centrale, bien que le site n'ait pas été fouillé à cet endroit. Lors de la saisie, la mesure d'entraxe des colonnes se trouvait réduite de moitié de part et d'autre de l'ensemble corinthien, alors que s'il était poursuivi de façon continue sur toute la façade, le positionnement des colonnes montrait un ensemble de 21 colonnes plus satisfaisant et c'est donc cette option de restitution qui fut retenue (Fig. 2 et 3).



*Figures A1-2 et A1-3 : la restitution originale de 1981 et celle proposée aujourd'hui*

De même, l'abside du grand triclinium était restituée à l'origine couverte par une coupole. La modélisation de ce volume a montré que la hauteur sous plafond du triclinium aurait alors dépassé les sept mètres, et que le rayon au sol de cette abside surmonté d'une telle coupole créait un volume assez disgracieux dans la cour à péristyle.

Par ailleurs, une toiture en terrasse aurait été plus problématique essentiellement à cause du climat local (le bassin de la cour à portique des bains a par exemple rapidement été transformé en jardin). C'est donc une troisième hypothèse de toiture conique qui fut adoptée. Les fouilles n'ont pas permis de déterminer s'il y avait des ouvertures dans les maçonneries de l'abside ; plusieurs versions ont été simulées et celle retenue montre quelques petites ouvertures comme à la villa de Nennig (Fig 4 et 5).



*Figures A1-4 et A1-5 : Hypothèses de restitution de l'abside du triclinium*

Il est à noter également que les nombreuses rencontres inter-disciplinaires (archéologues-architectes-infographistes) furent le lieu privilégié des réactions et nouveaux questionnements : les images en cours d'élaboration sont un facteur « déclencheur » indéniable pour l'archéologue. Il réagit sur la nature et l'aspect des matériaux, la conformation des éléments architecturaux, les proportions ou encore les procédés constructifs qui sont de facto mieux cernés, la visualisation tridimensionnelle met en évidence des points habituellement laissés dans l'ombre. C'est principalement pour cette raison que la phase de reconstitution proprement dite est plus porteuse d'avancées dans la connaissance archéologique du bâtiment ou du site que les résultats visuels qui en sont tirés par la suite.

La maquette détaillée a été élaborée en superposition au fond de plan des fouilles et d'après les relevés des fragments de bases, chapiteaux, éléments décoratifs, tuiles, etc... Le caractère symétrique et répétitif de l'architecture a conduit à la création d'une bibliothèque 3D constituée pour l'occasion, chacun de ses éléments étant inséré de nombreuses fois dans la maquette globale.

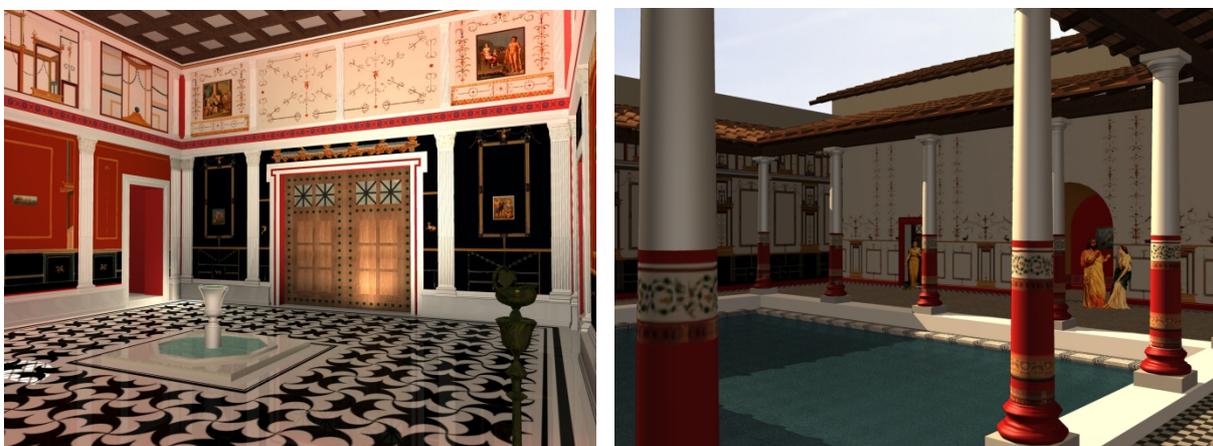
## Résultats

Une première série d'images montrant les différents corps de bâtiments et la manière avec laquelle il composent l'édifice d'origine a été produite (Fig. 6 et 7), se poursuivant avec l'adjonction de bâtiments supplémentaires lors de la deuxième période de construction au début du 2<sup>e</sup> siècle. Certaines de ces images ont été calculées depuis un point de vue virtuel identique à celui d'un observateur situé devant le panneau sur lequel est imprimée l'image, lui permettant d'imaginer la volumétrie du bâtiment en fonction des restes devant lesquels il se trouve.



*Figures A1-6 et A1-7 : Façade sur cour et façade « arrière » du bâtiment central*

Une autre série d'images montrant le « programme décoratif » a été réalisée (Fig. 1 et 9). Les fresques murales utilisées pour ces images sont des images scannées de peintures pompéiennes retravaillées. La quasi-absence de fragments de fresques retrouvés à Echternach autorise seulement à présenter ces décorations comme probables du point de vue de leurs couleurs et de leur géométrie générale.



*Figures A1- 8 et A1-9 : Décoration intérieure de la salle centrale et de la cour à péristyle devant les bains.*

Pour les images détaillées en extérieur, la villa virtuelle devant être incrustée dans son site réel, une prise de vue photographique à 360° et retravaillée pour y faire disparaître tout élément contemporain (ainsi que tous les arbres résineux absents au premier siècle) a été élaborée (Fig. 10 et 11)



*Figures A1-10 et A1-11 : la villa et ses dépendances dans son site*

### **Réalisme de la représentation et vérité archéologique**

Une interrogation récurrente en matière de reconstitution informatique provient paradoxalement du caractère hyper-réaliste des résultats graphiques obtenus. En effet, n'est-ce pas « abuser » le spectateur que de lui donner à voir des images de qualité photographiques dans lesquelles il ne peut distinguer ce qui est de l'ordre de l'hypothèse, de la probabilité ou de la certitude ? Différents moyens de représentation à notre disposition auraient pu être employés : niveaux de transparence, netteté ou flou, fausses couleurs. Cependant l'évocation des ambiances plutôt que celle de vérités archéologiques indiscutables nous a conduit à ne pas mettre en évidence les points sur lesquels cette reconstitution est le résultat d'extrapolations cependant clairement référencées.

L'exemple typique de cette attitude est le portique principal et ses fresques pariétales : la connaissance partielle de son état original aurait voulu que les images le montrent sans fresques peintes puisque peu de fragments colorés ont été retrouvés. Un des objectifs de ces images étant de montrer un aspect souvent méconnu du grand public -l'abondance des décors et des couleurs dans l'architecture romaine, figurer le portique sans ses fresques du quatrième style pompéien lui aurait fait perdre une grande partie de son sens (Fig. 12 et 13).



*Figures A1-12 et A1-13 : Aile ouest du portique de la cour*

Les images doivent de plus être indissociables du propos les accompagnant. Il est impératif d'y joindre un avertissement stipulant que le réalisme quasi-photographique est le vecteur d'une traduction aussi fidèle que possible des ambiances mais que pour autant ces images sont le fruit d'un faisceau d'hypothèses probables plutôt que de la mise en équation de certitudes.

## **Annexe 2 : Brève introduction à PHP**

## 1. Qu'est ce que PHP ?



PHP pour *Pre Hypertext Processor*, est un langage de script exécuté par le serveur Web qui héberge le site (comme les scripts CGI, ASP...) et non par la navigation du visiteur (comme une page HTML, un script écrit en JavaScript ou une applet Java qui s'exécute directement sur l'ordinateur de l'utilisateur). La syntaxe du langage PHP est fortement inspirée de celles du langage C et du Perl.

Ses principaux atouts sont :

- la gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL) ;
- sa richesse fonctionnelle : PHP comporte plus de mille fonctions ;
- la simplicité d'écriture des scripts ;
- la disponibilité sur le Web de nombreux scripts PHP prêts à l'emploi ;
- la possibilité d'inclure le script PHP au sein d'une page HTML ;
- la simplicité de liaison avec des bases de données. De nombreux systèmes de base de données sont supportés, mais le plus utilisé avec le PHP est MySQL, un système de base de données gratuit et disponible sur les plateformes Unix, Linux, et Windows.

Ce langage de programmation permet essentiellement de construire des sites Web dynamiques, particulièrement lorsqu'ils sont reliés à une base de données.

## 2. La relation client-serveur : différences entre HTML et PHP

Lorsque nous consultons une page HTML dite statique, voici les différentes étapes de fonctionnement :

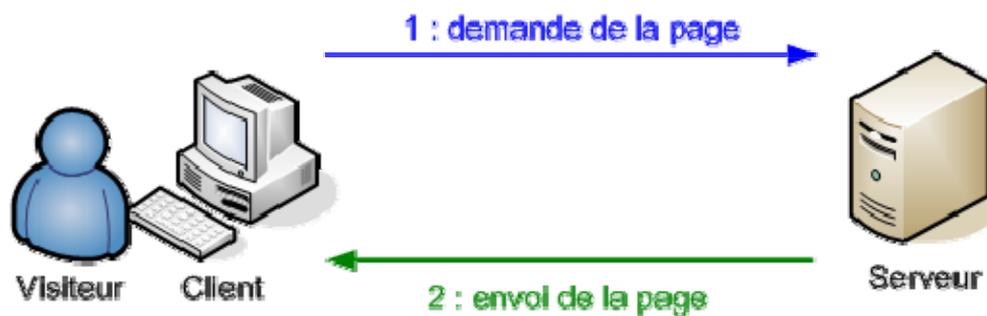


Figure A2-1 : Illustration de la consultation d'une page HTML classique  
(siteduzero@[SDZ07])

1. Le client (visiteur), par l'intermédiaire du navigateur qui envoie l'adresse URL encodée, demande à voir une page Web
2. Le serveur Web (l'ordinateur) qui héberge la page va chercher le fichier dans son disque dur et envoie la page HTML en l'état au navigateur
3. Le navigateur interprète les différents langages se trouvant dans ce fichier (HTML, JavaScript, CSS...) et affiche la page

Ce type de langage, appelé coté-client, est pour des raisons évidentes de sécurité, assez limitatif. Les applications plus complexes seront traitées dans l'espace plus sécurisé qui est le serveur qui héberge le site Web. Les traitements sont alors exécutés côté-serveur et seuls les résultats seront envoyés au navigateur de l'utilisateur.

Lorsque nous consultons une page HTML dite dynamique, voici les différentes étapes de fonctionnement :

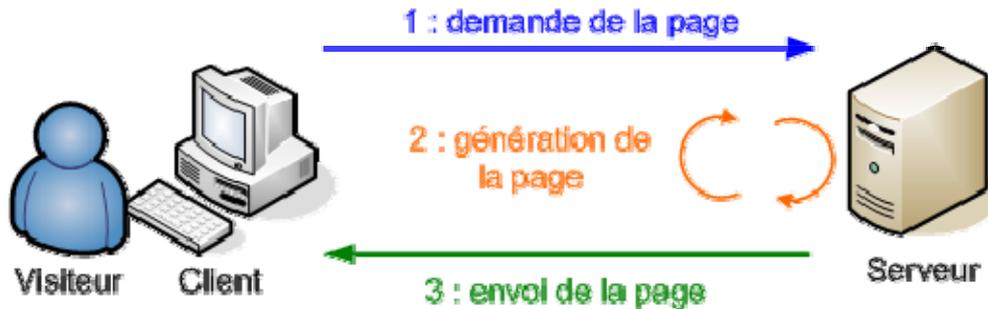


Figure A2-2 : Illustration de la consultation d'une page HTML dynamique (siteduzero@[SDZ07])

1. Le navigateur envoie l'adresse encodée
2. Le serveur Web cherche dans son arborescence si le fichier existe et si celui-ci reconnu comme une application PHP (.php, .php3, .phtml). Si tel est le cas, le serveur Web transmet ce fichier à PHP
3. PHP interprète le fichier, c'est-à-dire qu'il va analyser et exécuter le code PHP. Si ce code contient des requêtes vers une base de données MySQL, PHP envoie la requête. La base de données renvoie alors les informations voulues au script qui peut les exploiter (pour les afficher par exemple)
4. PHP continue d'interpréter la page, puis retourne le fichier dépourvu du code PHP (puisqu'il est exécuté) au serveur WEB
5. Le serveur Web renvoie finalement le fichier au navigateur de l'utilisateur. Ce fichier ne contient plus alors que du HTML.

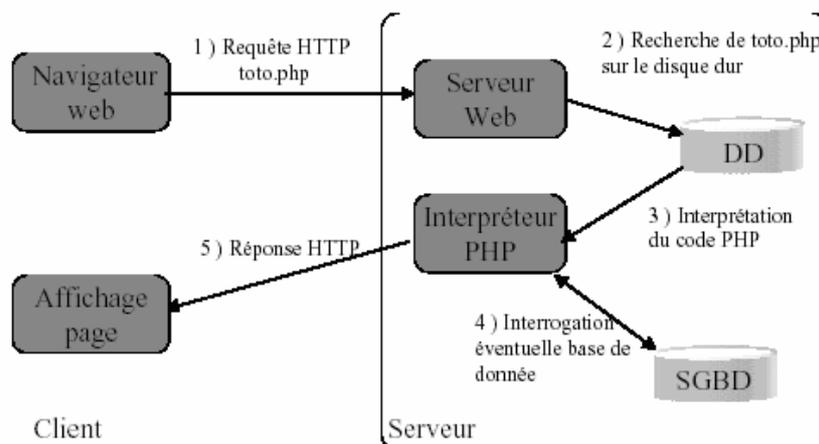


Figure A2-3: Principe de fonctionnement d'un serveur PHP (Brémont [BRE06])

Le code PHP s'exécute côté-serveur, il n'y a ainsi plus aucune trace du code PHP lors de la consultation du code source de la page dans le navigateur PHP, au contraire du JavaScript où le code source reste visible.

La particularité du PHP est que la page générée est destinée à un seul client. Lorsqu'un nouveau client se présente, le serveur génère une nouvelle page HTML. La page générée peut donc être unique et personnalisée à l'utilisateur.

Notons que la génération de la page peut prendre du temps (quelques millisecondes en fonction de la taille de la page) et que le serveur traitant du PHP doit être plus puissant qu'un serveur HTML simple. Si le site rencontre du succès, de nombreux clients peuvent demander une page PHP en même temps.

### **3. Mise en œuvre et déploiement : les problèmes rencontrés**

Pour tester les scripts, il deviendrait très vite pesant de télécharger à chaque fois les fichiers par FTP. C'est pourquoi installer un serveur Web en local sur son ordinateur se révèle indispensable pour tester les scripts en direct. Nous pouvons ainsi programmer en PHP sans avoir besoin d'être connecté à Internet, ce qui peut être intéressant pour les personnes ne disposant pas (encore) de connexions au forfait.

Il existe un outil incontournable pour le PHP, c'est WAMP5 (<http://www.wampserver.com>) (cf. Mémoire de Projet de Fin d'Etudes, §1.3 du Chapitre **Outils utilisés pour les développements**).

Ce programme permet d'installer automatiquement en quelques secondes un environnement de travail complet : soit un serveur Apache 2, PHP 5, MySQL, PHPMyAdmin. Il est ainsi possible de tester localement sous Windows les scripts PHP et les bases de données.

Lorsque WAMP5 est démarré, les serveurs Apache 2 et MySQL sont automatiquement « lancés » (il est même possible de le faire automatiquement au démarrage de Windows). Une petite icône s'installe dans la barre des tâches, à côté de l'horloge, permettant un accès rapide aux fonctions proposées par WAMP5:

- Fichier Log : renvoie aux erreurs générées par Apache et MySQL
- Configuration : donne accès aux différentes configurations de WAMP5
- Administration : gestion des répertoires virtuels, des extensions, de PHPMyadmin
- Web local : ouvre la page <http://localhost/> répertoire racine du site
- Démarrer/Arrêter : démarre/arrête Apache 2 et MySQL
- Quitter : ferme WAMP5

Pour que les pages PHP soient interprétées, il est impératif de placer les fichiers dans le répertoire www (sous le répertoire d'installation de WAMP5). Ce répertoire est configuré par défaut comme répertoire racine du serveur web. Le serveur Apache 2 est configuré pour ouvrir automatiquement un fichier index lors de la saisie de l'adresse '<http://localhost/>' (à condition évidemment que le serveur Apache 2 soit en route). Cette page sert de page d'accueil au web local et permet de vérifier le bon fonctionnement de WAMP5.

Il est conseillé de créer un répertoire par projet dans le répertoire www afin d'avoir une vision plus claire des développements.

Lors de l'installation de PHP, un fichier de configuration nommé « php.ini » sera créé sur le système. Le fichier php.ini est le fichier de configuration de serveur PHP. Ce fichier respecte la structure des fichiers INI bien connus de nombreuses applications Windows. Il s'agit d'un fichier texte ASCII divisé en plusieurs sections, chacune portant un nom et contenant des variables relatives à la section concernée.

Toute modification du fichier php.ini nécessite un redémarrage du serveur afin que la nouvelle configuration soit prise en compte.

Par défaut le fichier php.ini est configuré de sorte à être fonctionnel. Intervenir sur ce fichier peut être intéressant pour rajouter de nouveaux modules ou pour affiner certains paramètres de sécurité mais doit systématiquement faire l'objet d'un backup de sécurité au préalable.

Nous avons changé quelques paramètres afin d'assurer le bon fonctionnement de nos programmes. Tout d'abord, nous avons augmenté le temps maximum d'exécution d'un programme, puis nous avons augmenté la taille de mémoire qu'un script pouvait « consommer ».

```
.....  
; Resource Limits ;  
.....  
max_execution_time = 150      ; Maximum execution time of each script, in seconds  
max_input_time = 100        ; Maximum amount of time each script may spend parsing request data  
memory_limit = 1000M        ; Maximum amount of memory a script may consume (1000MB)
```

Figure A2-4 : Copie d'écran d'une partie du fichier php.ini

Pour toutes les fonctionnalités du langage PHP, celles-ci sont si nombreuses que nous renvoyons le lecteur à quelques sites ;

→ <http://www.siteduzero.com/tuto-3-68-1-premiers-pas-avec-php.html>

Cours sur lequel nous nous sommes inspirés pour écrire cette annexe et qui explique le PHP pour le créateur de site Web dynamique débutant. [SDZ07]

→ <http://www.lehtml.com/php/>

Tutoriel écrit par Luc Van Lancker sur le PHP et MySQL. [LAN04]

→ [http://noria.apf.francophonie.org/IMG/pdf/program\\_php.pdf](http://noria.apf.francophonie.org/IMG/pdf/program_php.pdf)

Formation au langage PHP-MYSQL par Mohamed Ines KEBIR [KEB05]

→ [www.mathrice.org/ecole/support/phpHTML.pdf](http://www.mathrice.org/ecole/support/phpHTML.pdf)

Mémo conçu par Brémont [BRE06] pour permettre une prise en main rapide des langages HTML et PHP.

Il est très incomplet, mais est sans doute une bonne base pour appréhender ces deux langages pour lesquels nous pouvons facilement trouver des compléments d'informations sur les sites web et dans les ouvrages cités en bibliographie.

## **Annexe 3 : Brève introduction à MySQL**



MySQL est une base de données implémentant le langage de requête SQL, langage relationnel très connu. Avec cet outil, il est possible de créer plusieurs bases de données sur un serveur. Une base est composée de tables contenant des enregistrements.

Pour se connecter à une base depuis php, il faut spécifier un nom de serveur, un nom d'utilisateur, un mot de passe et un nom de base.

Les fonctions de connexion :

**mysql\_connect(\$server,\$user,\$password)** : permet de se connecter au serveur \$server en tant qu'utilisateur \$user avec le mot de passe \$password, retourne l'identifiant de connexion si succès, FALSE dans le cas contraire.

**mysql\_select\_db(\$base,\$id)** : permet de choisir la base \$base, retourne TRUE en cas de succès, sinon FALSE.

**mysql\_close(\$id)** : permet de fermer la connexion.

Les identifiants de connexion ne sont pas nécessaires si nous ne nous connectons qu'à une seule base à la fois, ils permettent seulement de lever toute ambiguïté en cas de connexions multiples.

Pour envoyer une requête à une base de donnée, il existe la fonction : **mysql\_query(\$str)** qui prend pour paramètre une chaîne de caractères qui contient la requête écrite en SQL et retourne un identificateur de résultat ou FALSE si échec.

Les requêtes les plus couramment utilisées sont : CREATE (création d'une table), SELECT (sélection), INSERT (insertion), UPDATE (mise à jour des données), DELETE (suppression), ALTER (modification d'une table), etc.

**mysql\_fetch\_row(\$result)** : retourne une ligne de résultat sous la forme d'un tableau. Les éléments du tableau étant les valeurs des attributs de la ligne. Retourne FALSE s'il n'y a plus aucune ligne.

**mysql\_fetch\_array(\$result)** : retourne un tableau associatif. Les clés étant les noms des attributs et leurs valeurs associées leurs valeurs respectives. Retourne FALSE s'il n'y a plus aucune ligne.

**mysql\_fetch\_object(\$result)** : retourne un objet. Les attributs de l'objet correspondent à ceux de la ligne de résultat. Et les valeurs des attributs de l'objet correspondent à ceux de la ligne de résultat. Retourne FALSE s'il n'y a plus aucune ligne.

**mysql\_num\_rows(\$result)** : retourne le nombre de lignes du résultat. Et ainsi permet de remplacer le while par un for.

Penser à bien tester la valeur de retour des fonctions (mysql\_query et les autres) afin de détecter toute erreur et d'éviter de polluer la page avec des *Warnings*.

## **Annexe 4 : Comparaison des différents logiciels d'édition de SVG**

Tableaux comparatifs suivant plusieurs paramètres.

Wikipedia@[WIK07], The Free Encyclopedia.

URL [http://en.wikipedia.org/wiki/Comparison\\_of\\_vector\\_graphics\\_editors](http://en.wikipedia.org/wiki/Comparison_of_vector_graphics_editors)

## Informations générales :

	Creator	First Released	Latest version	List price (USD)	License	Open source
ConceptDraw	CSOdessa	1999	5	\$299, \$79 upgrade	Proprietary	No
CorelDRAW Graphics Suite	Corel [1]	1989	X3 (13)	\$399, \$179 upgrade (includes Photopaint)	Proprietary	No
Adobe Freehand	Adobe Systems	1988	MX (11)	\$399, \$99 upgrade	Proprietary	No
Adobe Illustrator	Adobe Systems	1987	CS3 (13)	\$499, \$169 upgrade	Proprietary	No
Inkscape	Inkscape Team	2003	0.45.1 (03/2007)	Free	GPL	Yes
Intaglio	Purgatory Design	2003	2.5	\$89	Shareware	No
MagicTracer	Elgorithms	2005	1.0	\$49	Demoware	No
OpenOffice.org Draw	The OpenOffice.org community	2000	2.2 (03/2007)	Free	LGPL	Yes
PhotoLine 32	Computerinsel GmbH	1995	13.01	€59	Proprietary	No
Sodipodi	Sodipodi	2000	0.34	Free	GPL	Yes
Synfig	Synfig Team	preview	0.61.05	Free	GPL	Yes
Xara Xtreme	Xara	1995	2.0f	\$79	Proprietary	No
Xara Xtreme LX	Xara	2006	0.7	Free	GPL (except graphic engine)	Yes
Zoner Draw	ZONER Software	??	5	\$49.99 (\$29.99 upgrade)	Proprietary	No
Sketsa	Kiyut	???	4.1	\$49	Proprietary	No
	Creator	Released	Latest version	List price (USD)	License	Open source

## Environnement :

	Windows	Mac OS X	Unix
ConceptDraw	Yes	Yes	No
CorelDRAW Graphics Suite	Yes	Dropped (v.11)	Dropped (v.9)
Freehand	Yes	Yes	No
Adobe Illustrator	Yes	Yes	No
Inkscape	Yes	Yes	Yes
Intaglio	No	Yes	No
MagicTracer	Yes	No	No
OpenOffice.org Draw	Yes	Yes	Yes
PhotoLine 32	Yes	Yes	No
Sodipodi	Yes	No	Yes
Synfig	Yes	Yes	Yes
Xara Xtreme	Yes	No	No
Xara Xtreme LX	No	Yes	Yes
Zoner Draw	Yes	No	No
Sketsa	Yes	Yes	Yes

## Import format vecteur:

Software	AI	CDR	ODG	PS/EPS	PDF	SVG	SWF	DXF
Adobe Illustrator	Yes	Yes	No	Yes	Yes	Yes	No	Yes
ConceptDraw	No	No	No	No	No	No	No	Yes
CorelDraw Graphics Suite	Yes	Yes	No	Yes	Yes	Yes	No	Yes
Inkscape	partial (1)	No	No	partial (2)	No	Yes, native format	No	No
OpenOffice.org Draw	No	No	Yes	Yes	No	partial (2)	No	Yes
PhotoLine 32	No	No	No	No	Yes	No	No	No
Xara Xtreme	Yes (3)	Yes	No	Yes	Yes	partial (4)	No	No
Zoner Draw	Yes	Yes	No	Yes	No	No	No	Yes
Sketsa	No	No	No	No	No	Yes	No	No
Software	AI	CDR	ODG	PS/EPS	PDF	SVG	SWF	DXF

## Export format vecteur:

Software	AI	CDR	ODG	PS/EPS	PDF	SVG	SWF	LaTeX	DXF	TT/T1 fonts
Adobe Illustrator	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	No
ConceptDraw	No	No	?	Yes	Yes	Yes	Yes	Yes	Yes	No
CorelDRAW Graphics Suite	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes
Inkscape	partial (1)	No	Yes	Yes	partial (2)	Yes	No	Yes	No	No
OpenOffice.org Draw	No	No	Yes	Yes	Yes	No	Yes	No	No	No
PhotoLine 32	No	No	No	No	Yes	No	Yes	No	No	No
Xara Xtreme	Yes	No	No	Yes	Yes	Yes	Yes	No	No	No
Zoner Draw	Yes	No	No	Yes	Yes	No	No	No	Yes	No
Sketsa	No	No	No	No	Yes (1)	Yes	No	No	No	No
Software	AI	CDR	ODG	PS/EPS	PDF	SVG	SWF	LaTeX	DXF	TT/T1 fonts

Nous avons entouré les trois logiciels qui acceptent l'import de fichier .dwg et exportent en .svg. Sur les trois logiciels, Adobe Illustrator, ConceptDraw et CorelDRAW Graphics, aucun n'est *open source*. Nous avons finalement utilisé Adobe Illustrator car il s'agissait du seul que nous possédions au laboratoire.

**Annexe 5 : Notice explicative de la réalisation de plans 2D sous  
AutoCAD et Adobe Illustrator**

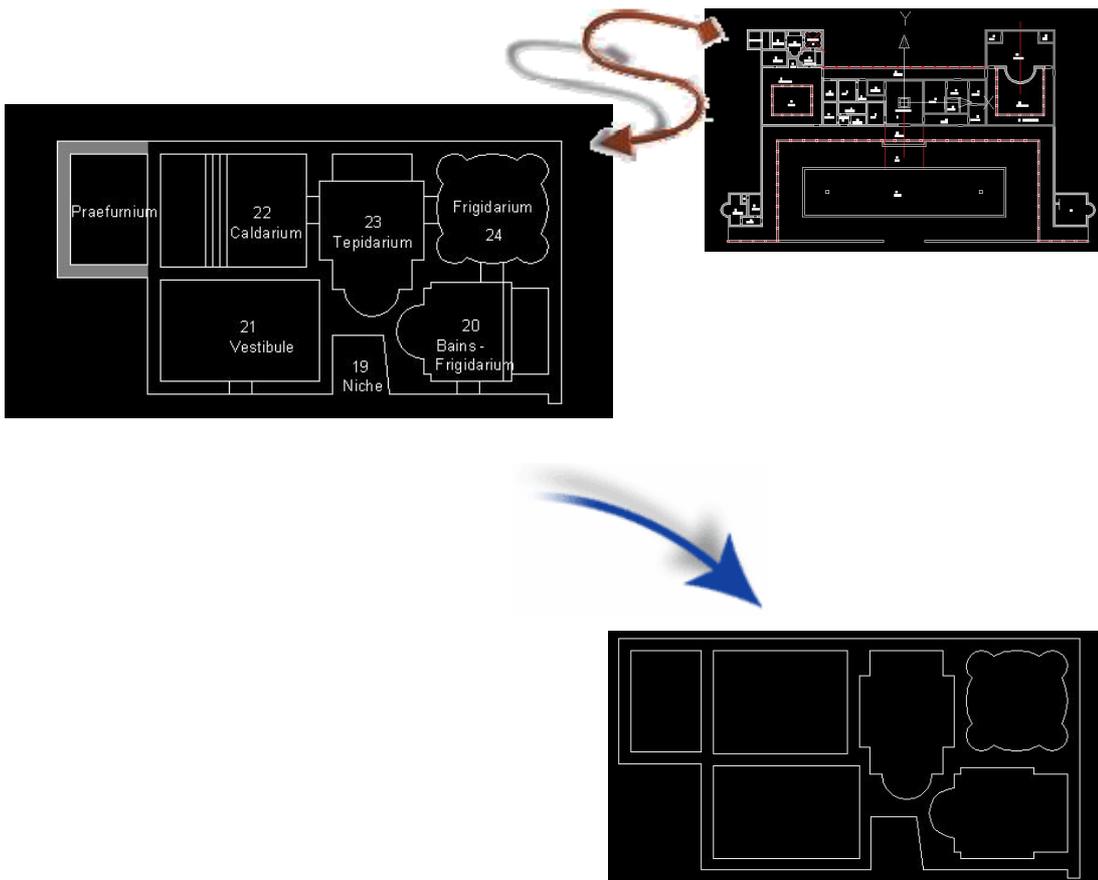


## Notice explicative de la réalisation de plans 2D sous AutoCAD et Adobe Illustrator

Cette notice se décompose en plusieurs étapes que l'utilisateur qui souhaite ancrer automatiquement son plan 2D devra suivre méthodiquement :



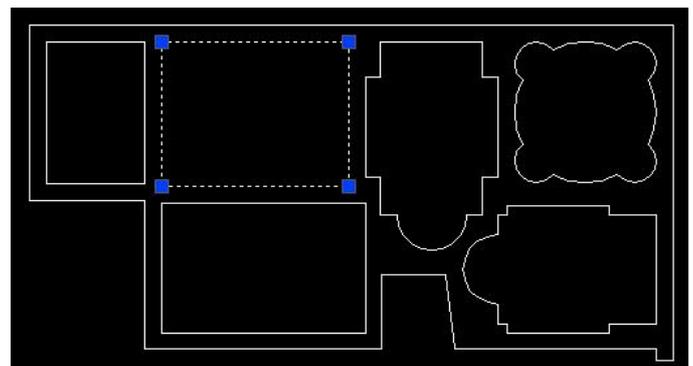
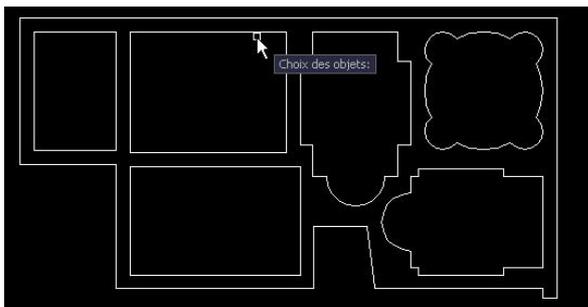
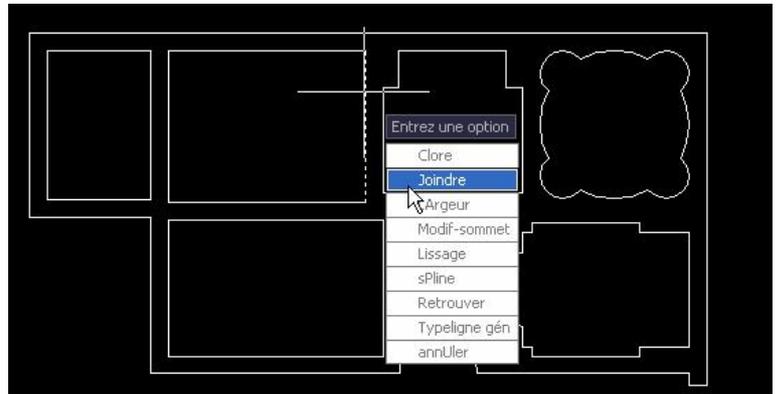
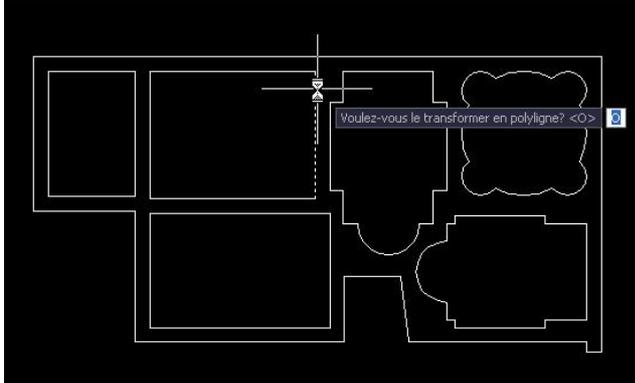
1. Ouvrir avec ACAD le fichier .dwg contenant le plan 2D que vous souhaitez rendre interactif. Une des premières étapes consiste à obtenir un plan global « nettoyé » de l'ensemble du site, puis vous allez par la suite traiter chaque lieu (et ses sous lieux) séparément dans Adobe Illustrator une fois le plan global obtenu. L'étape de nettoyage est ici relatée pour un seul lieu, étape à réitérer pour tous les lieux dans le plan global.



2. Enlever tous les éléments tels que noms des lieux et des sous-lieux, hachures, couloirs. A la fin de cette étape, il ne doit rester que les « murs extérieurs » des lieux et des sous-lieux.

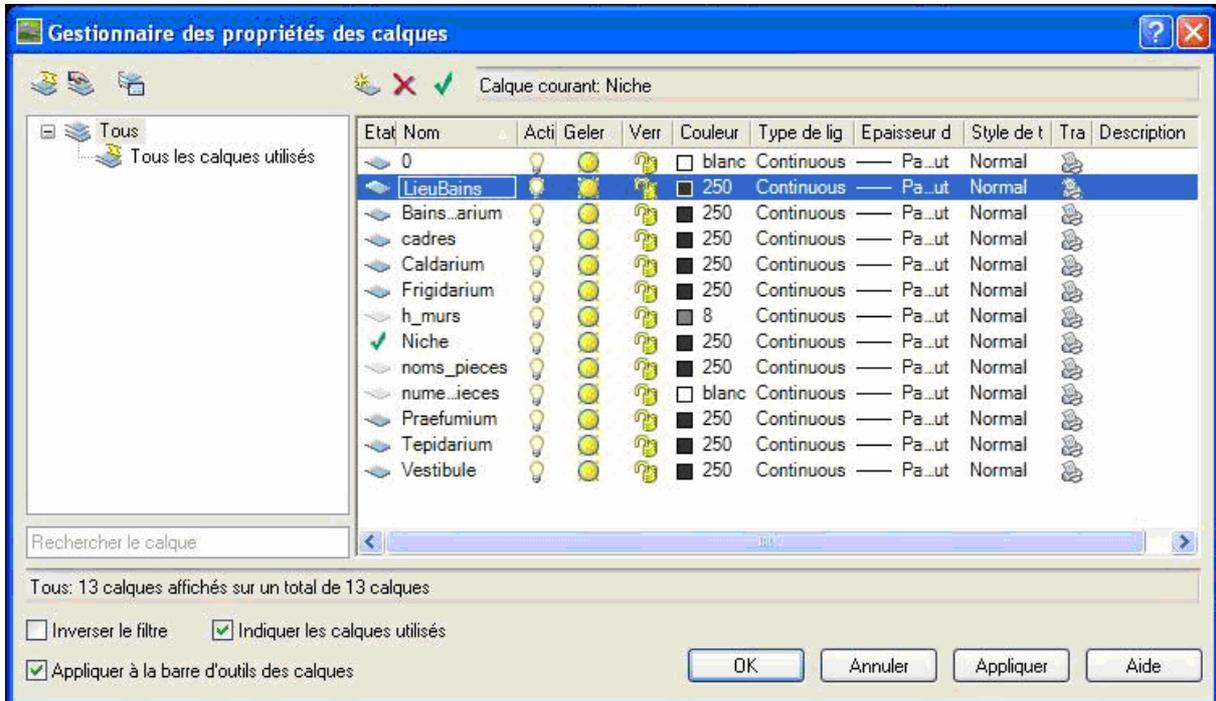


3. Créer des polygones de ces murs extérieurs et les joindre. Il suffit de taper la commande « petit », de sélectionner une ligne, de la transformer en polygone et de joindre toutes les lignes.





- Créer pour chaque sous-lieu le calque correspondant et l'appliquer à la polyligne jointe. Pour le lieu, appeler le calque LieuNomdulieu, par exemple pour le lieu Bains, le calque se nommera LieuBains.



- Vous pouvez choisir d'enregistrer le plan en format .dwg ou en .dxf mais faites attention à la version du logiciel AutoCAD ; la version d'Adobe doit toujours être postérieure à la version du fichier AutoCAD importé.



- Importer le fichier précédemment créé dans Adobe Illustrator en cochant les options Ajuster et Fusionner les calques, puis pour le modèle global, enregistrer le fichier avec les options d'enregistrements citées ci-dessous.





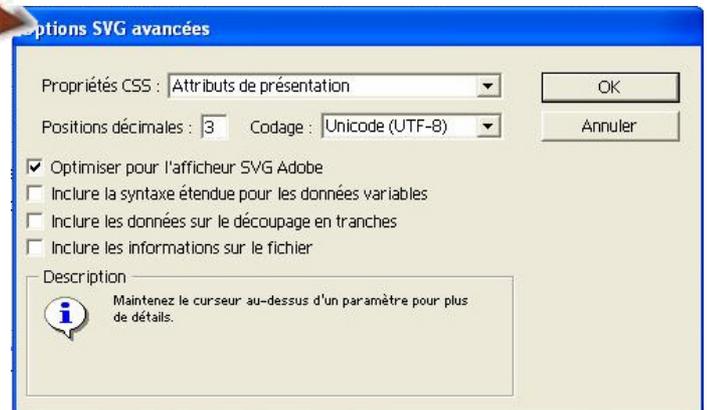
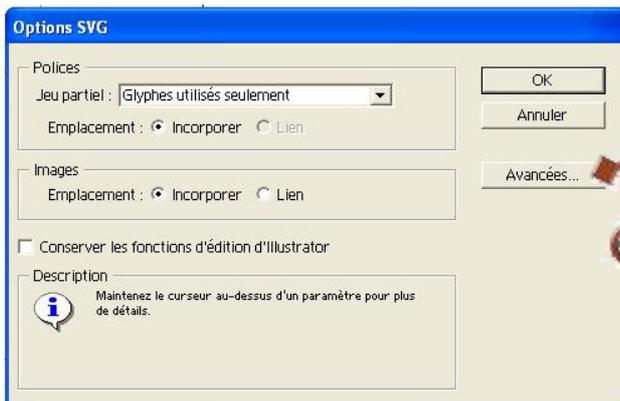
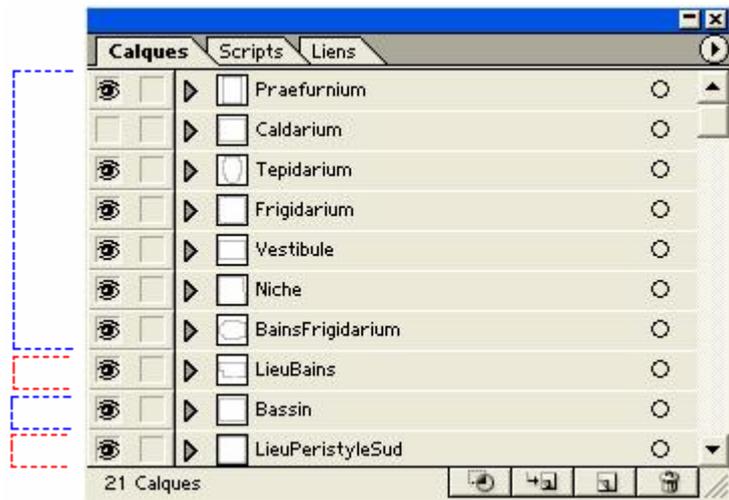
Les calques doivent avoir une certaine structure hiérarchique comme l'illustre la figure suivante, il faut que les calques des sous-lieux précèdent celui du lieu.

Sous-lieux du lieu Bains

Lieu Bains

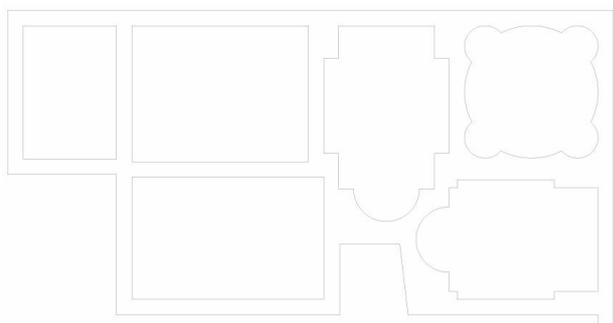
Sous-lieu du lieu PeristyleSud

Lieu PeristyleSud



7. Choisir un lieu (et ses sous-lieux correspondants) à ancrer et supprimer dans Adobe Illustrator les calques qui ne sont ni le lieu, ni les sous-lieux. Puis enregistrer le fichier en .SVG avec les mêmes options que celles du modèle global.

Si toutes les étapes ont été réussies avec succès, il est temps désormais d'ancrer votre plan 2D grâce aux programmes PHP !



## **Annexe 6 : Comparaison des principaux logiciels de modélisation 3D disponibles sur le marché**

Effectué par Gaetan Le Gravier et Benjamin Moignot [LEG07] dans le cadre d'un rapport dénommé « La RV/3D appliquée au génie urbain, *Réalité virtuelle et ambiances urbaines*.  
URL [http://ww2.eivp-paris.fr/chachoua/TER2007/%5BN%B010%5D\\_RapportTERLeGravier-Moignot.pdf](http://ww2.eivp-paris.fr/chachoua/TER2007/%5BN%B010%5D_RapportTERLeGravier-Moignot.pdf)

	<i>coefficients</i>	3DSmax	Blender	Maya	Sketchup	Chaoscope	Milkshape 3D	Terragen	Cinema 4D XL	Xsi	Lightwave
<b>I) Utilisation du logiciel</b>											
<b>Prise en main</b>	5	4	3	3	4 (prise en main très simple)	3	4	3	5	4 (certains outils difficiles à utiliser, mais les bases sont simples)	3
<b>Temps d'apprentissage</b>	4	4 (existe d'excellents tutoriaux en vidéos)	4 (logiciel instinctif)	1 (plusieurs centaines d'heures)	4 (premiers rendus très faciles à obtenir)	2	3	3	4	4 (tutoriaux disponibles sur internet, payant mais bon marchés)	2
<b>Interface</b>	3	3 (barres d'outils très bien agencées)	3 (à première vue, un peu brute)	1 (Interface complexe pour un novice)	5 (on pourrait difficilement faire plus simple)	3 (a priori assez simple, mais pas aussi "intelligente" que pour d'autres)	3	4	4	5 (entièrement personnalisable)	4
<b>Menus et options</b>	3	4 (menus très complets)	3 (de très nombreuses options en barre d'outils)	4 (menus très complets et très bien organisés)	3 (très simples, mais incomplets)	3	3	3	4	5 (personnalisables)	4
<b>Prix de licences</b>	X	4 250 €	Gratuit	environ 10 000 €	469 €	Gratuit	20 €	gratuit	649 €	449 €	836 €
<b>Disponibilité du logiciel</b>	X	en démo (www.presence-pc.com)	téléchargement (www.zdnet.fr)	Versions de démo gratuites (www.journal-du-net.com)	Google sketchup est gratuit (www.sketchup.google.fr)	Téléchargement (www.clubic.com)	Version de démo gratuite (www.zdnet.fr)	téléchargement (www.terragen.eu.org)	en démo (www.clubic.com)	en démo (http://calogatelecharger.com)	Version d'essai gratuite (www.newtek.com)
<b>Note intermédiaire 1</b>		<b>57</b>	<b>49</b>	<b>34</b>	<b>60</b>	<b>41</b>	<b>50</b>	<b>48</b>	<b>65</b>	<b>66</b>	<b>47</b>

	<i>coefficients</i>	3DSmax	Blender	Maya	Sketchup	Chaoscope	Milkshape 3D	Terragen	Cinema 4D XL	Xsi	Lightwave
<b>II) Rendu 3D</b>											
Importation de fichiers (autocad, photoshop, ...)	2	5	5	5	5	0	5	5	3	5	5
Convertisseur d'images	1	5	5	5	0	0	5	3	3	4	4
Capture d'images	2	5	5	5	5	0	5	4	4	3	4
Gestion de l'éclairage	3	5	5	5	5	5	0	3	4	5	5
Textures	3.5	5	5	5	5	5	5	3	5	5	5
Animation 3D	4.5	5	5	5	5	5	5	3	4	4	5
Qualité des rendus	4	4,5	4,5	5 (probablement le meilleur)	3	5 (rendus de superbe qualité dans son domaine)	4	3	4,5	4	5
Puissance de l'ordinateur requise	2	2 (rendus parfois très longs à faire)	3	1 (machine surpuissante)	4	3	3	2	3	2	2
<b>Note intermédiaire 2</b>		<b>102</b>	<b>104</b>	<b>102</b>	<b>95</b>	<b>81</b>	<b>87</b>	<b>70</b>	<b>88.5</b>	<b>90.5</b>	<b>90</b>
<b>III) Utilisation approfondie du logiciel</b>											
Otions d'accrochage	1	5	4	5	5	0	5	3	3	5	4
Utilisation des calques	1	5	5	5	0	0	0	4	4	5	4
Outils de mesure	0.5	5	3	5	5	0	5	3	2	5	5
Outils copier / réseaux	0.5	5	5	5	0	0	5	3	2	5	3
Outils mise en page pour une présentation 2D	2	5	5	5	0	5	0	4	2	4	3
<b>Note intermédiaire 3</b>		<b>25</b>	<b>23</b>	<b>25</b>	<b>7.5</b>	<b>10</b>	<b>10</b>	<b>18</b>	<b>13</b>	<b>23</b>	<b>18</b>
<b>Note finale</b>		<b>184</b>	<b>176</b>	<b>161</b>	<b>162.5</b>	<b>132</b>	<b>147</b>	<b>136</b>	<b>166.5</b>	<b>179,5</b>	<b>155</b>

Nous allons commenter ce résultat car nous ne sommes pas en accord avec certaines notes attribuées notamment au niveau de l'utilisation du logiciel Blender. A notre sens, ce logiciel est plus difficile d'utilisation que ce que les notes font apparaître, c'est pourquoi nous ne l'avons pas choisi comme logiciel intermédiaire AutoCAD - VRML.

**Annexe 7 : Notice explicative de la réalisation de modèles 3D sous Sketchup**

## Notice explicative de la réalisation de modèle 3D sous

Cette notice se décompose en plusieurs étapes que l'utilisateur qui souhaite ancrer automatiquement son modèle devra suivre méthodiquement :

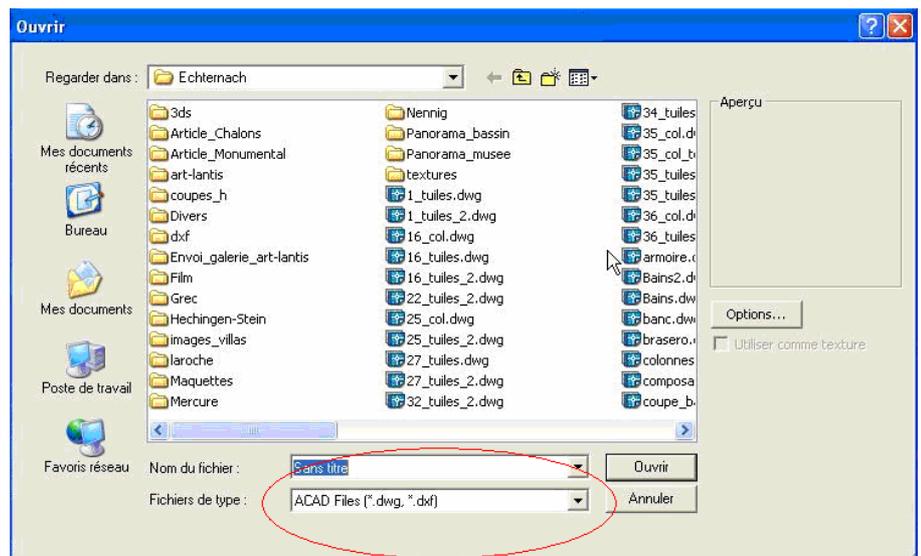
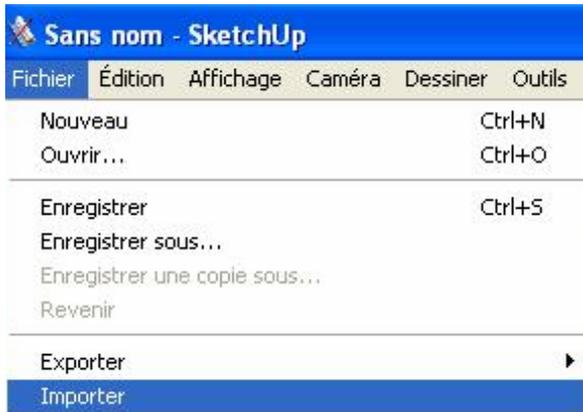


1. Prêter attention à la taille du fichier .dwg que vous souhaitez ancrer. Si le modèle est de taille assez importante (> 10 Mo) et qu'il est possible de diviser le modèle en sous-parties, il est préférable de le partitionner avant de l'importer dans le logiciel Sketchup (le temps d'import peut s'avérer long).

Pour les formes complexes, il peut être utile d'explorer vos polygones avant de les importer dans Sketchup (Voir l'article de Laurent Brixius à l'adresse URL <http://www.arch-image.com/blog/index.php/2007/08/03/113-pourquoi-exploser-vos-polygones-avant-de-les-importer-dans-sketchup>).



2. Démarrer le logiciel de modélisation Sketchup et importer le fichier .dwg :  
Cliquez sur Fichier-> Importer  
Chercher un fichier de type ACAD Files (\*.dwg, \*.dxf)

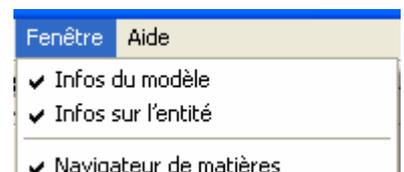


3. Nettoyer le modèle après export (si cela n'a pas été fait précédemment).  
La conversion du fichier .dwg en .skp peut en effet avoir pour conséquences la création de segments (voir 1.).



4. Appliquer les textures que vous souhaitez incorporer dans le modèle :

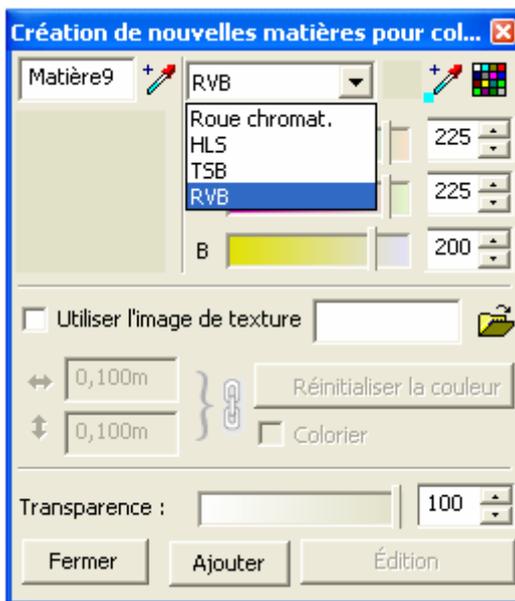
Faire afficher la fenêtre de Navigateur de matières :  
Fenêtre-> Navigateur de matières doit être coché.



La fenêtre suivante apparaît :



5. Appuyer sur le bouton Créer pour faire afficher l'éditeur de matière.



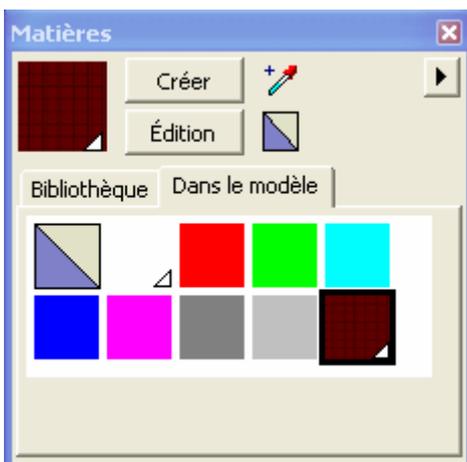
Le nom de la matière s'incrémente automatiquement si aucun nom n'est donné.

Choisir son code de couleur (la roue chromatique vous permet de sélectionner une couleur en fonction de ses caractéristiques de teinte, de saturation et de brillance ; les curseurs RVB vous permettent de sélectionner des couleurs dans la gamme de couleurs RVB (Rouge, Vert, Bleu) ; HLS (Hue Saturation Luminance) vous permet de sélectionner une couleur en fonction de ses caractéristiques de teinte, de saturation et de luminance, et enfin les curseurs TSB vous permettent de choisir des couleurs dans la gamme TSB (Teinte, Saturation, Brillance)).

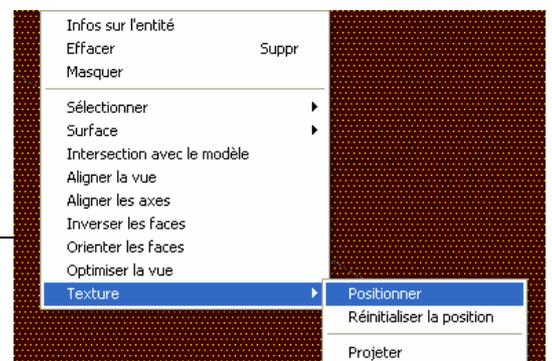
Si vous souhaitez utiliser une texture préexistante, vous l'importez en cochant l'option « Utiliser l'image de texture » et en recherchant l'image dans vos fichiers.

Vous réglez alors la hauteur et la largeur de la texture si elles ne correspondent pas à la face.

Puis vous appuyez sur le bouton « Ajouter », ce qui a pour effet de l'incrémenter au niveau des textures dans le modèle.



Une fois la texture incorporée sur la face et à la bonne taille, vous pouvez la positionner sur la face. Il suffit alors d'effectuer un clic droit sur la face sélectionnée-> Texture->Positionner.



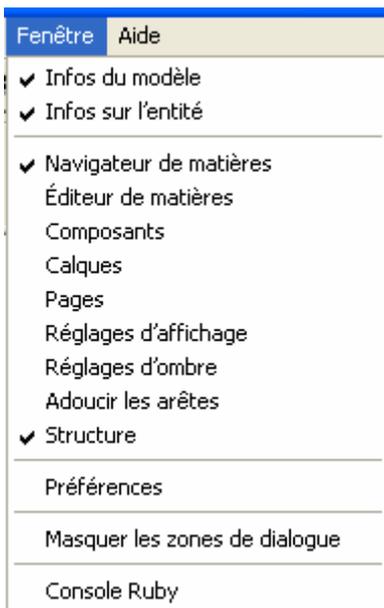


6. Nous allons désormais nous préoccuper de la structure du modèle.

Pour afficher la structure du modèle, cliquer sur Fenêtre-> Structure.

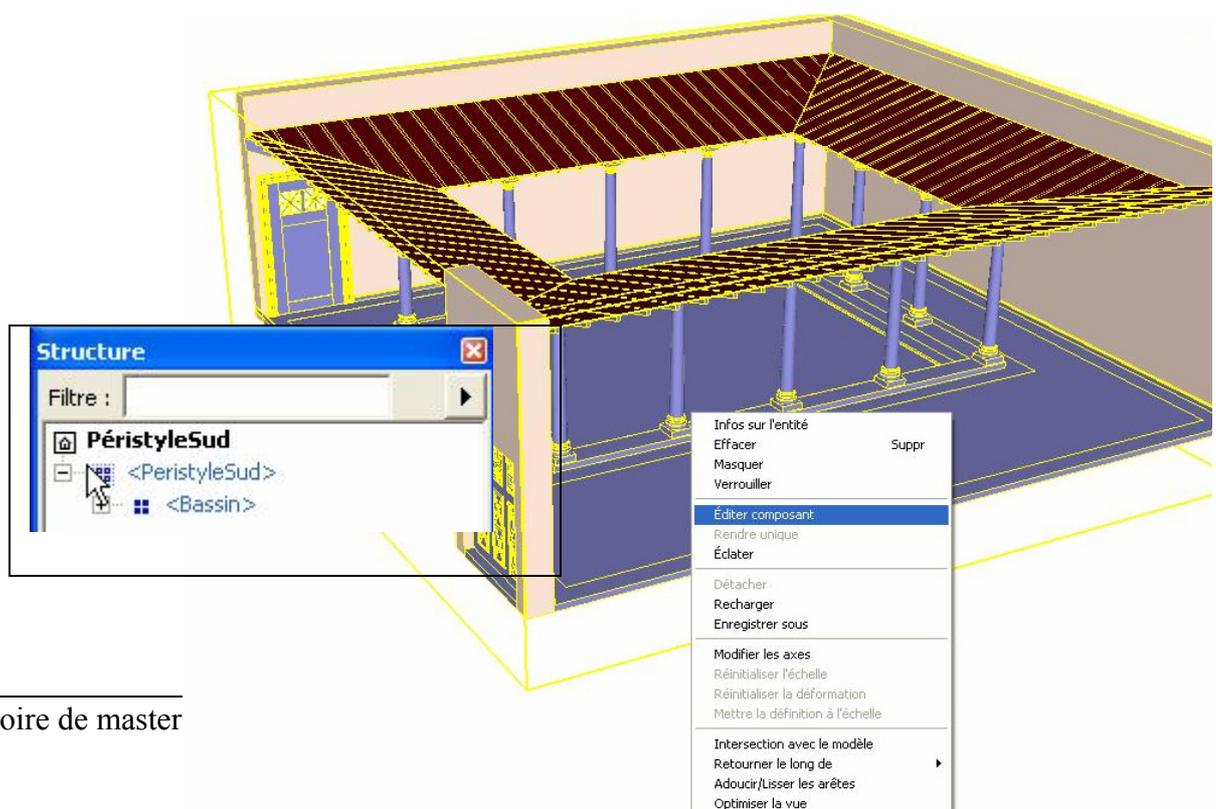
Tout d'abord, sélectionner tout les groupes et/ou composants du modèle et les éclater (clic droit sur tous les groupes sélectionnés et Eclater).

Puis, pour le modèle représentant le lieu dans sa globalité, sélectionner toutes les formes du modèle et créer un composant (clic droit sur le modèle et créer composant), puis le nommer par le nom du lieu.



Puis éditer le composant lieu et former les composants sous-lieux de la même manière que précédemment.

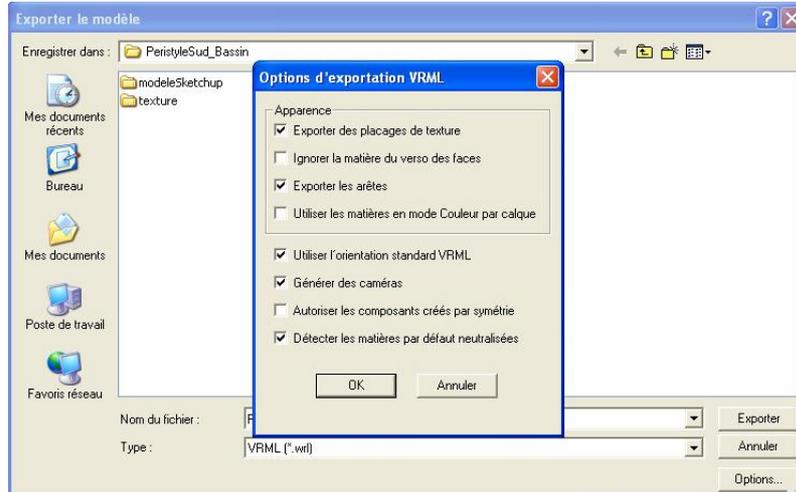
Veillez à la structure hiérarchique du fichier.





## 7. Export du modèle en VRML

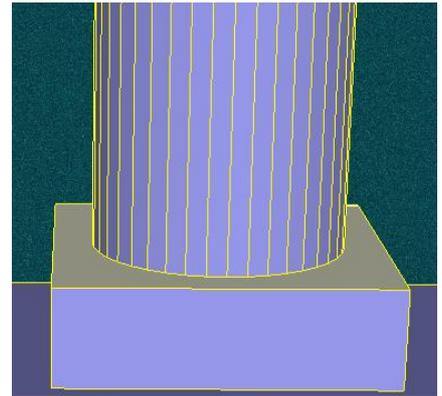
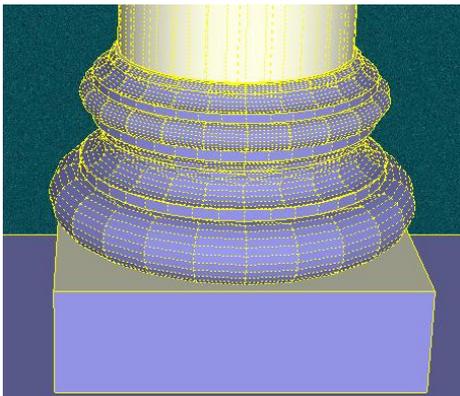
Une fois la structure voulue obtenue (voir la figure précédente ayant pour lieu PeristyleSud et pour sous-lieu Bassin), il est temps d'exporter le modèle. Fichier-> Export-> Modèle 3D



Il est possible de réaliser automatiquement une coupe du modèle sur Sketchup, mais elle ne s'exporte pas sous VRML, il faut donc créer un modèle de la coupe pour l'obtenir.

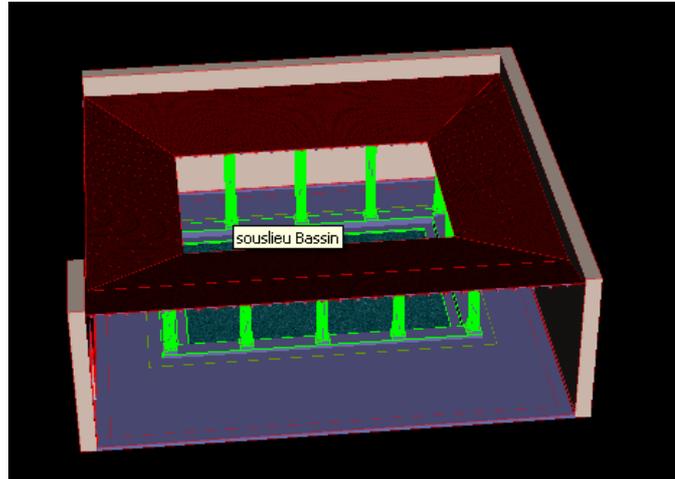
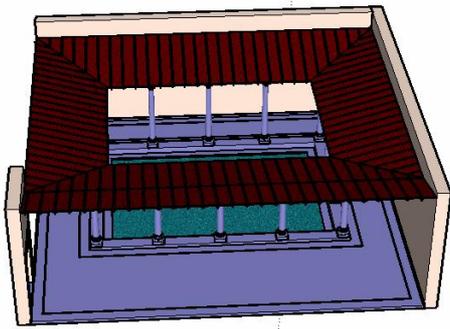


Prêter attention à la taille du VRML. En effet si celle-ci est supérieure à 10 Mo, le VRML affichera un message d'erreur et le modèle ne s'ouvrira pas. Dans ce cas là, il est indispensable de simplifier le modèle tel que l'illustre l'exemple suivant :



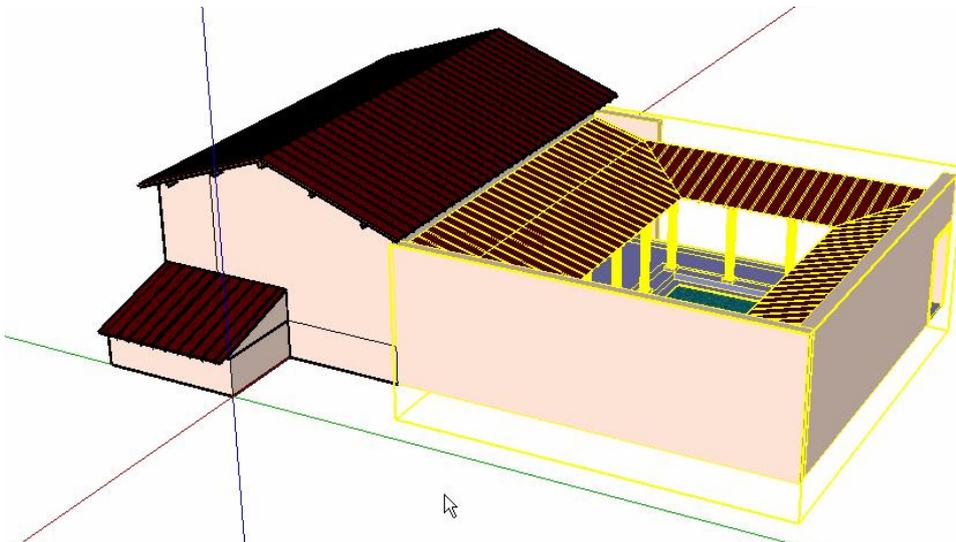
Avant simplification du haut et du bas de la colonne, le modèle exporté avait une taille d'approximativement 15 Mo ; celle-ci était de 2Mo après l'export.

Si toutes les étapes ont été réussies avec succès, il est temps désormais d'ancrer votre modèle grâce aux programmes PHP !



N.B.

Si vous avez divisé le modèle en sous-parties, prenez garde de ne pas les changer de position (ni translation, ni rotation, ni changement d'échelle). En effet, si vous souhaitez faire apparaître plusieurs parties simultanément, celles-ci doivent garder leurs positions d'origine.



Problème d'export du nom des textures : il peut arriver que la texture ne garde pas son nom d'origine si celui-ci est trop long par exemple. Il faut donc mettre dans le fichier « texture » la texture exportée de Sketchup et pas la texture d'origine.

## TABLE DES ILLUSTRATIONS DES ANNEXES

---

---

<i>Figure A1-1 : Vue aérienne de la villa.....</i>	<i>70</i>
<i>Figures A1-2 et A1-3 : la restitution originale de 1981 et celle proposée aujourd’hui .....</i>	<i>71</i>
<i>Figures A1-4 et A1-5 : Hypothèses de restitution de l’abside du triclinium.....</i>	<i>71</i>
<i>Figures A1-6 et A1-7 : Façade sur cour et façade « arrière » du bâtiment central.....</i>	<i>72</i>
<i>Figures A1- 8 et A1-9 : Décoration intérieure de la salle centrale et de la cour à péristyle devant les bains. ....</i>	<i>72</i>
<i>Figures A1-10 et A1-11 : la villa et ses dépendances dans son site .....</i>	<i>73</i>
<i>Figures A1-12 et A1-13 : Aile ouest du portique de la cour .....</i>	<i>73</i>
<i>Figure A2-1 : Illustration de la consultation d’une page HTML classique (siteduzero@[2007]).....</i>	<i>75</i>
<i>Figure A2-2 : Illustration de la consultation d’une page HTML dynamique (siteduzero@[2007]).....</i>	<i>76</i>
<i>Figure A2-3: Principe de fonctionnement d’un serveur PHP (Brémont [BRE06]) .....</i>	<i>76</i>
<i>Figure A2-4 : Copie d’écran d’une partie du fichier php.ini .....</i>	<i>78</i>