



Année Académique 2016-2017
Université de Liège – Faculté des Sciences Appliquées

MÉTHODOLOGIE D'AIDE À LA PROGRAMMATION EN MODÉLISATION ARCHITECTURALE PARAMÉTRIQUE

Jury :

Pierre Leclercq – Promoteur

Aurélie de Boissieu

Xaviera Calixte

Frédéric Delvaux

John Schrayen

Travail de fin d'étude réalisé en vue de l'obtention du grade de master
« Ingénieur civil en architecture » par Vervier Charlotte

Je souhaite, par ces quelques lignes, remercier tous ceux qui ont participé à la réalisation de ce mémoire.

Tout d'abord, ma reconnaissance va au Professeur Pierre Leclercq pour avoir accepté d'être le promoteur de mon travail de fin d'étude et pour m'avoir guidée tout au long de celui-ci.

Ensuite, je tiens également à remercier Mademoiselle Xaviera Calixte et Monsieur John Schrayen pour leurs nombreux conseils et leurs disponibilités.

Ma gratitude va également au Professeur Frédéric Delvaux et au Professeur Anis Gallas pour l'attention portée à mon sujet de mémoire, pour leurs conseils et pour leurs suggestions de lecture.

En outre, je remercie également le Docteur Aurélie de Boissieu, dont la thèse a été l'une la source initiale d'inspiration de ce travail et qui a consentit à faire partie de mon jury.

Je n'oublie pas tous les étudiants qui ont accepté de donner de leur temps pour participer aux phases expérimentales de ce travail et je les remercie.

Enfin, le dernier remerciement mais non le moindre, est adressé à mes proches, famille et amis, qui m'ont soutenue durant ces cinq années d'études universitaires et celles qui les ont précédées.

TABLE DES MATIÈRES

1. INTRODUCTION	5
AVANT – PROPOS	6
1.1. LA MODÉLISATION PARAMÉTRIQUE	7
1.1.1. <i>Apparition des modeleurs paramétriques</i>	7
1.1.2. <i>Concept de modèle paramétrique</i>	8
2. LE DOMAINE DE LA RECHERCHE	12
2.1. MISE EN ÉVIDENCE DE PROBLÉMATIQUES LIÉES À LA MODÉLISATION PARAMÉTRIQUE	13
2.2. PROBLÉMATIQUE	15
2.3. CADRE DE LA RECHERCHE	17
2.3.1. <i>Cadre de la méthode</i>	17
2.3.2. <i>Cadre de l'expérimentation</i>	18
2.4. MÉTHODOLOGIE	19
2.5. DÉFINITION DES CONCEPTS	21
3. MÉTHODES DE CONCEPTION	22
3.1. LA CONCEPTION EN ARCHITECTURE SELON CHRISTOPHER ALEXANDER	24
3.1.1. <i>Méthode de décomposition d'un problème de conception</i>	24
3.1.2. <i>Le langage des Patterns</i>	28
3.2. LES PRINCIPES DU DESIGN ORIENTÉ-OBJET	30
3.2.1. <i>Les origines de la méthode Orientée-Objet</i>	30
3.2.2. <i>Les concepts et principes de la méthode Orientée-Objet</i>	32
3.3. LE DESIGN EN MODÉLISATION PARAMÉTRIQUE D'APRÈS ROBERT WOODBURY	35
3.3.1. <i>Comportement d'utilisation des modeleurs paramétriques</i>	35
3.3.2. <i>La méthode des patterns</i>	37
4. LA MAPMAP	40
4.1. OBJECTIF DE LA MAPMAP	41
4.2. ORIGINES DE LA MAPMAP	42
4.3. PRINCIPES DE LA MAPMAP	44
4.3.1. <i>Stratégie de Division (ou de responsabilité unique)</i>	44
4.3.2. <i>Réflexe de Documentation</i>	47
4.3.3. <i>Principe d'Encapsulation (fonction macro)</i>	49
4.3.4. <i>Dé-foisonnement des Liaisons</i>	50
4.3.5. <i>Description Générique</i>	52
5. LES EXPÉRIMENTATIONS	54
5.1. EXPÉRIMENTATION DE MISE EN ŒUVRE DE LA MAPMAP DURANT LA CONCEPTION	55
5.1.1. <i>Protocole de l'expérience</i>	55
5.1.2. <i>Critères d'analyse</i>	65
5.2. EXPÉRIMENTATION DE COMMUNICATION DE MODÈLES PARAMÉTRIQUES	68
5.2.1. <i>Protocole de l'expérience</i>	68
5.2.2. <i>Critères d'analyse</i>	72
6. RÉSULTATS ET DISCUSSION	74

6.1.	RÉSULTATS DE LA PREMIÈRE EXPÉRIENCE	75
6.1.1.	<i>Analyse des modèles</i>	75
6.1.2.	<i>Séquençage des enregistrements</i>	86
6.2.	RÉSULTATS DE LA SECONDE EXPÉRIENCE	90
6.3.	LIMITES ET PERSPECTIVES DE LA MAPMAP ET SON ÉTUDE	94
6.3.1.	<i>Limites</i>	94
6.3.2.	<i>Perspectives</i>	94
7.	CONCLUSION	97
8.	RÉFÉRENCES	99
8.4.	BIBLIOGRAPHIE	100
8.5.	WEBOGRAPHIE	101
8.6.	LISTE DES FIGURES	102
9.	ANNEXES	104

1. INTRODUCTION

Ce premier chapitre a pour vocation d'introduire le domaine de ma recherche : la modélisation paramétrique. Je commencerai par présenter les raisons qui m'ont motivée à réaliser une étude dans ce domaine. Ensuite, je réaliserai une présentation succincte de la modélisation paramétrique et de mon positionnement vis-à-vis de sa définition.

Avant – propos

Il y a de cela deux ans, durant ma troisième année d'étude en ingénieur civil architecte à l'Université de Liège, j'ai découvert la modélisation paramétrique via une introduction au logiciel Rhinoceros et son extension Grasshopper. Cette formation est dispensée dans le cadre du cours de Modélisation Architecturale Numérique (MAN) par le Professeur Pierre Leclercq qui était alors assisté du Professeur Anis Gallas, ce dernier ayant depuis été remplacé par l'assistant John Schrayen.

Cette approche succincte suffit à m'intéresser à ce mode de modélisation. Au fil des projets que je développais durant les ateliers d'architecture, je m'impatientais d'avoir l'occasion de mettre en œuvre un modèle paramétrique qui me permettrait d'en découvrir plus sur ce logiciel. Chaque nouvelle idée devenait un défi à relever et, au fil des manipulations, je prenais peu à peu conscience des potentialités offertes par la modélisation paramétrique, bien que je n'en aie pas la totale maîtrise. Au-delà des ateliers, ce mémoire m'apparut comme un nouveau défi, une nouvelle occasion de développer mes connaissances du domaine paramétrique, d'un point de vue plus théorique cette fois.

Par la suite, j'ai eu l'occasion de lire le mémoire de mon prédécesseur Raphaël Devaux (2016) qui met en avant quelques problématiques liées au domaine de la modélisation paramétrique et observées dans de grandes agences d'architecture. Or certaines de ses problématiques faisaient écho aux observations que j'avais pu faire précédemment lors de mes propres manipulations : un modèle paramétrique peut vite devenir complexe si bien qu'y effectuer des changements ou le partager avec d'autres utilisateurs nécessite un certain degré d'organisation et de lisibilité au sein du modèle. Je commençais donc à entrevoir un moyen d'apporter ma contribution au domaine de la modélisation paramétrique en cherchant un moyen d'aider les utilisateurs à structurer leur modèle.

Finalement, ayant la chance d'avoir été nommée au poste d'élève-monitrice pour le cours de MAN de cette année 2016 – 2017, cette opportunité vint enrichir mes idées et projets de recherche au vu des potentielles observations effectuelles auprès de débutants en pleine découverte de ce mode de modélisation.

1.1. La modélisation paramétrique

Pour introduire le cadre de ma recherche, je commencerai par un bref exposé de la modélisation paramétrique. De nombreux auteurs s'étant déjà étendus sur le sujet suite au développement croissant de la pratique de cette méthode de modélisation, je choisis de me concentrer sur les thématiques suivantes : l'apparition des logiciels de modélisation paramétrique, la définition du concept de modèle et celle du concept de paramétrisme.

Pour une plus ample exposition de la modélisation paramétrique, je recommande la lecture de la thèse de monsieur Daniel Davis (2013) « Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture » qui m'a beaucoup aidé dans la rédaction de cette introduction.

1.1.1. Apparition des modeleurs paramétriques

Au début des années soixante, Ivan Sutherland créa le logiciel Sketchpad. Bien que son utilisation resta limitée et qu'il ne fut employé que dans les laboratoires Lincoln du MIT, ce logiciel, composé de l'une des premières interfaces graphiques, contribua grandement au développement de l'interaction ordinateur-utilisateur. En effet, l'objectif de Sutherland était de rendre les ordinateurs accessibles à une nouvelle catégorie d'utilisateurs (Sutherland, 2003)¹.

Pour ce faire, le système Sketchpad se compose d'entrées, de sorties et de programmes de calculs. L'utilisateur possède un crayon optique qui lui permet de dessiner et il a accès à un jeu de commandes sous forme de boutons poussoirs qui permettent de modifier le modèle. Le logiciel mémorise les informations concernant la topologie du modèle si bien qu'il est capable de modifier le dessins pour le faire correspondre aux nouvelles conditions que l'utilisateur applique (Sutherland 2003).



Figure 1 - Sketchpad (Sutherland, 2003)

Ainsi, le logiciel Sketchpad est considéré comme étant le premier logiciel de modélisation paramétrique. Par la suite, de nombreux logiciels de modélisation paramétrique ont fait leur

¹ (Sutherland, 2003) fait référence à un rapport technique fondé sur un ouvrage réalisé en Janvier 1963 par l'auteur pour l'obtention du grade de Docteur de Philosophie au Massachusetts Institute of Technology.

apparition : CATIA, SolidWorks, Pro/Engineer, les programmes CAD, Grasshopper, GenerativeComponents, Houdini, etc. (Davis, 2013). Le premier de ces logiciels commercialisé et utilisé en conception paramétrique est Pro/Engineer qui fut mis sur le marché par la Parametric Technology Corporation en 1988 (Weisberg, 2008). Cependant, les premiers logiciels ayant pour vocation d'être utilisés dans les domaines de l'automobile et de l'aéronautique, il fallut attendre quelques années pour que des logiciels de modélisation paramétrique spécifiques à l'architecture fassent leur apparition (de Boissieu, 2010).

1.1.2. Concept de modèle paramétrique

Paramétrisme

Le terme paramétrique a émergé au XIX^{ème} siècle, dans le domaine des mathématiques et des sciences. Il est défini par la 'Concise Encyclopedia of Mathematics' comme « *un ensemble d'équations qui expriment un ensemble de quantités comme les fonctions explicites d'un certain nombre de variables indépendantes, connues comme des 'paramètres'* » (Weisstein, 2003).

Sur base de cette définition, nous pouvons considérer que tout dessin architectural est paramétrique. En effet, un projet d'architecture est toujours lié à des contraintes qui le caractérisent : le budget, le site, la fonction du bâtiment, etc. (Davis, 2013). Ainsi de nombreux auteurs s'accordent à dire que le paramétrisme existe depuis toujours, à travers chaque forme d'architecture. Parmi ces auteurs nous pouvons citer :

« *This thesis begins with the assertion that all design is parametric.* » (Hudson, 2010, p. 18)

« *It must be stated that architectural design is inherently a 'parametric' process, and that the architect has always operated in a 'parametric fashion'.* » (Gerber, 2007, p. 54) cité par (Davis, 2013, p. 23)

Pour illustrer ce concept, prenons l'exemple de la Sagrada Familia d'Antonio Gaudi. En 1883, époque à laquelle la modélisation paramétrique ainsi que les ordinateurs n'existent pas encore, l'architecte espagnol commença le projet de la cathédrale. Pour ce faire, il choisit d'employer un système de proportions simples qu'il appliqua à chaque partie du bâtiment, par exemples : les dimensions d'une partie du temple équivalent un multiple du douzième de la plus grande dimension, ou encore, la hauteur des colonnes exprimée en mètre équivaut au double du nombre de point qui forme la géométrie de cette colonne.

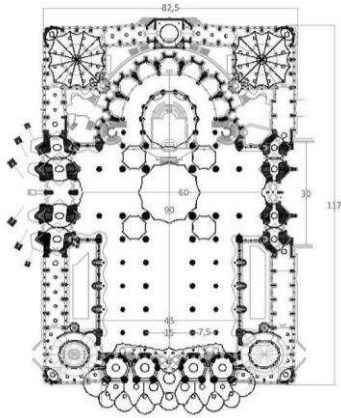


Figure 2 - Plan d'une partie de la Sagrada Família
(www.sagradafamilia.org)

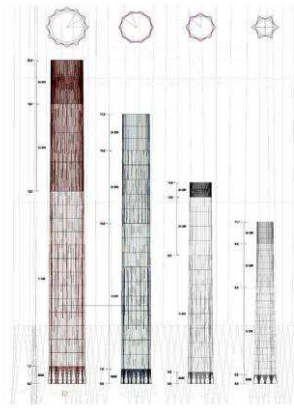


Figure 3 – Schéma de dimensionnement des colonnes de la Sagrada Família de Barcelonne
(www.sagradafamilia.org)

À la mort de Gaudi, le bâtiment n'était pas fini et il est toujours en construction à l'heure actuelle. Or les dessins originaux du bâtiment ont été détruits au début du XX^{ème} siècle. Dans les années nonante, grâce aux règles de composition du projet, Mark Burry et son équipe du SIAL ont pu reconstituer les informations manquantes via la modélisation paramétrique (Burry & Burry, 2006) cité par (de Boissieu, 2010). Ce fut l'un des premiers usages de la modélisation paramétrique en architecture bien que le paramétrisme aie déjà été présent dans la composition du XIX^{ème} siècle.

Considérant que le paramétrisme existe dans toute composition architecturale, il est important de pouvoir distinguer les modèles dis 'paramétriques' des modèles dis 'standards', c'est-à-dire tout modèle ne relevant pas d'une conception paramétrique. En effet, le terme paramétrique est aujourd'hui utilisé pour signifier aussi bien toutes les compositions, que les compositions qui changent ou encore les compositions appartenant au style du paramétrisme (Davis, 2013). En effet, plusieurs auteurs s'accordent à dire que le paramétrisme est synonyme de changement et que c'est cette caractéristique qui permet d'identifier les modèles de type paramétrique.

« Initially, a parametric definition was simply a mathematical formula that required values to be substituted for a few parameters in order to generate variations from within a family of entities. Today it is used to imply that the entity once generated can easily be changed. » (Yessios, 2003, p. 263) cité par (Davis, 2013, p. 24)

« Design is change. Parametric modelling represents change » ... « parametric modelling introduces fundamental change: 'marks', that is, parts of the design, relate and change together in a coordinated way. » (Woodbury, 2010, p. 7 & 11)

Contrairement à ces différentes propositions de définition, Daniel Davis (2013) définit l'architecture paramétrique non pas par ce que le modèle fait, mais par la façon dont il a été conçu, c'est-à-dire par l'expression de relations explicites entre les paramètres de composition rendant chaque modèle paramétrique unique (Davis, 2013). Ainsi, il définit un modèle

paramétrique comme un mathématicien le ferait : « *a set of equations that express a geometric model as explicit functions of a number of parameters* » (Davis, 2013, p. 31), c'est-à-dire « un ensemble d'équations qui exprime un modèle géométrique comme les fonctions explicites d'un certain nombre de paramètres », définition semblable à celle de la 'Concise Encyclopedia of Mathematics' énoncée plus haut (Weisstein, 2003).

Cette position, adoptée par Davis, laisse apparaître que la modélisation paramétrique est divisible en 2 concepts : d'une part, la création d'un modèle paramétrique et, d'autre part, l'exploitation d'un modèle en vue d'explorer diverses possibilités de composition. Cette division se retrouve chez certains architectes, elle est notamment exprimée très clairement dans le travail de Roland Hudson (2010) qui divise toute son analyse de modèle selon ces deux tâches. D'après lui, la première tâche, c'est-à-dire le processus de création, consiste en le développement de la description d'un problème et l'expression d'idées de résolution, tandis que la seconde tâche, l'exploration de la composition de l'espace, consiste à trouver les paramètres appropriés, à générer des alternatives et à évaluer celles-ci.

Bien que je considère que, si ces deux tâches se distinguent, elles n'en restent pas moins liées ; dans le cadre de ce travail, l'attention sera portée sur la première tâche énoncée par Hudson : le processus de construction du modèle paramétrique. C'est pourquoi, nous adopterons la définition du terme 'modèle paramétrique' établie par Daniel Davis (2013) sur base de la définition du terme 'paramétrique' dans un cadre mathématique (Weisstein, 2003).

Finalement, une dernière distinction s'impose entre modélisation paramétrique et algorithmique. Ces deux types de modélisation se distinguent principalement au niveau des types de modeleurs utilisés. Dans le cadre des logiciels paramétriques, tels que Rhinoceros, Digital Project ou CATIA ; l'utilisateur est amené à lier des paramètres et des dimensions à des contraintes géométriques préprogrammées. Par exemple, pour créer un cercle dans Rhinoceros, l'utilisateur utilise la fonction préprogrammée 'cercle' et il lui fournit des paramètres (dont la position du centre, le rayon, ...).

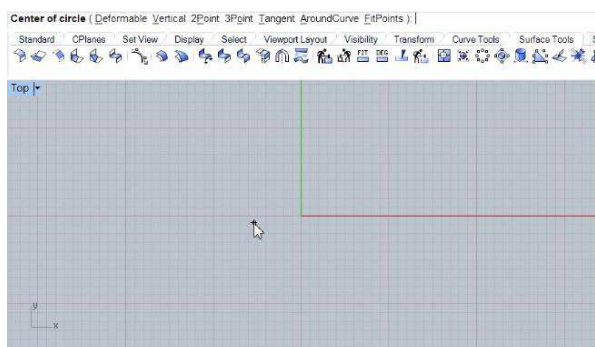


Figure 4 - Création d'un cercle dans Rhinoceros : choix du centre

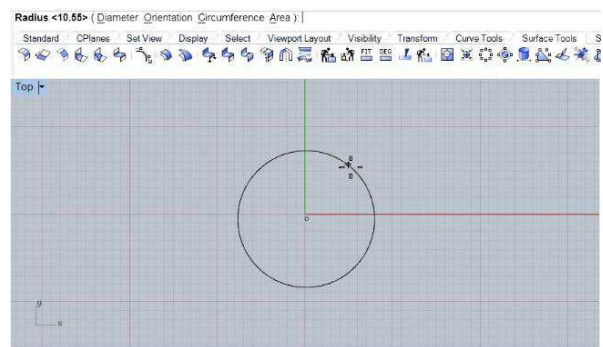


Figure 5 - Création d'un cercle dans Rhinoceros : choix du rayon

Par contre, dans les logiciels algorithmiques, tels que Visual Basic, Generative Components ou Grasshopper, l'utilisateur est amené à utiliser un langage de programmation textuel ou visuel (pour GenerativeComponents et Grasshopper) semblable à un code de programmation informatique qui lui permet de passer outre les limites de l'interface utilisateur (Leach, 2014).

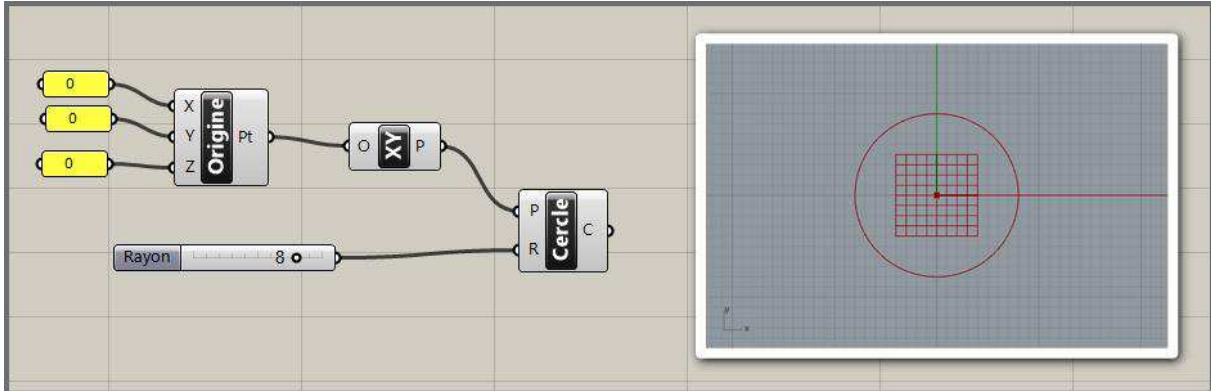


Figure 6 - Création d'un cercle avec Grasshopper par la création d'un code de programmation visuel

Dans la suite de ce travail, bien que l'appellation 'architecture paramétrique' soit conservée, nous nous concentrerons uniquement sur les logiciels de modélisation algorithmique et plus précisément sur ceux possédant une interface de programmation visuelle.

Modèle paramétrique

Le terme 'modèle' sera employé dans le cadre de ce travail pour signifier une 'représentation numérique' d'un élément réel. La création d'un modèle se fait par l'usage d'un outil informatique appelé modeleur.

Il existe, différents type de représentation d'un modèle et les modeleurs paramétriques sont caractérisés par un double système de représentation. D'une part, on retrouve une représentation habituelle en modélisation : la représentation graphique de la géométrie du modèle qui sera appelée ici représentation explicite. D'autre part, la représentation graphique est accompagnée d'une interface de programmation du modèle. Cette représentation informatique, sera appelée représentation symbolique.

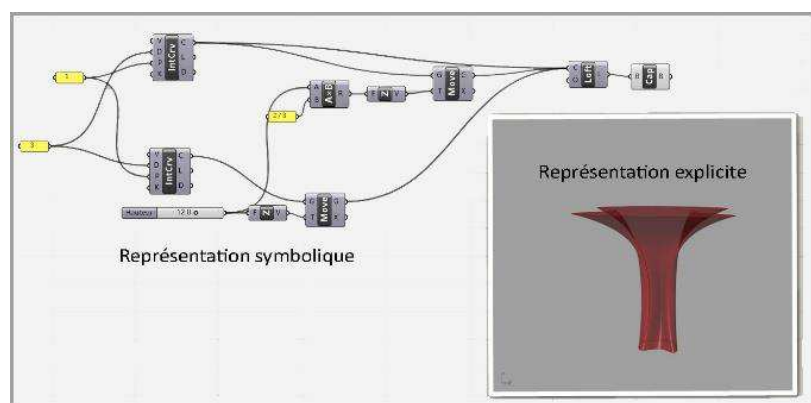


Figure 7 - Représentations symbolique et explicite dans Grasshopper

2. LE DOMAINE DE LA RECHERCHE

Ce second chapitre est consacré à la définition plus précise du cadre de la recherche. Ainsi, les paragraphes suivants permettront d'exposer la thématique qui fera l'objet de ce travail, le cadre dans lequel va se dérouler la recherche et la méthodologie qui a été adoptée pour développer cette étude.

Plan du chapitre :

2.1. MISE EN ÉVIDENCE DE PROBLÉMATIQUES.....	13
2.2. PROBLÉMATIQUE.....	15
2.3. CADRE DE LA RECHERCHE.....	17
2.3.1. <i>Cadre de la méthode</i>	17
2.3.2. <i>Cadre de l'expérimentation</i>	18
2.4. MÉTHODOLOGIE.....	19
2.5. DÉFINITION DES CONCEPTS.....	21

2.1. Mise en évidence de problématiques liées à la modélisation paramétrique

Les outils paramétriques ont de grandes potentialités. Cependant, des problèmes de flexibilités et réutilisabilité des modèles sont récurrents lors de l'utilisation de ce type de modeleurs. De nombreux auteurs ayant mis en avant cette fragilité, peuvent être cités :

« *Frequently I find my models have grown so tangled they can no longer accommodate even the most trivial change. Often I just start again.* » (Davis, 2013, p. 5)

« *To edit the relational graph or remodel completely is also commonplace.* » (J. Burry, 2007, p. 622)

« *[Parametric modelling] may require additional effort, may increase complexity of local design decisions and increases the number of items to which attention must be paid in task completion.* » (Aish & Woodbury, 2005, p. 151)

our exposer les différents facteurs générateurs de la fragilité des modèles paramétriques, Rick Smith a mis en évidence 5 problématiques liées à la modélisation paramétrique (Smith, 2007 ; cité par Davis, 2013, p. 39-45) :

a. « *Les modèles paramétriques requièrent un certain degré d'anticipation* »

Il est nécessaire de planifier certaines caractéristiques d'un modèle avant d'en commencer la construction. Ces caractéristiques (paramètres, contraintes, etc.) seront dépendantes de la manière dont on souhaite exploiter le modèle ensuite.

Contrairement à Hudson (2010) qui conclut sa recherche par l'affirmation que la création d'un modèle paramétrique et son exploitation dans la recherche d'un design sont deux tâches séparées, je pense que les deux restent liées : l'exploitation du modèle est dépendante de la bonne construction de celui-ci dans un souci de réutilisation et communication et identiquement la manière dont un modèle est construit dépend de l'exploitation que l'on souhaite en faire dans un souci de flexibilité et d'adaptabilité.

b. « *Anticiper la flexibilité peut être difficile* »

Il n'est pas évident de définir la manière dont le modèle sera exploité avant même qu'il ne soit conçu. Concevoir, arranger et éditer les dépendances sont les tâches dévolues aux utilisateurs de modeleurs paramétriques d'après Woodbury (2010). L'objectif étant d'obtenir un modèle des plus flexibles pour faciliter son adaptation aux changements dus à toute phase de conception, il convient d'ajouter des paramètres pour chaque aspect de la composition.

c. « Les changements majeurs brisent les modèles paramétriques »

« Après tout le temps et les efforts de programmation de la géométrie, quand tu penses que tu es juste, tu peux encore trouver que tu dois tout recommencer parce que le concept de composition initial a complètement changé. Si le changement est trop important ou la topologie non-familière, il ne reste qu'une option : reconstruire le modèle. » (Rick Smith, 2007; cité par Daniel Davis, 2013, p. 42)

Cet aspect découle directement de la problématique exposée au point (b). Il est difficile d'anticiper la manière dont le modèle va être utilisé et la flexibilité du modèle est limitée par la manière dont on l'a construit et les choix qui ont été effectués lors de sa construction. Lors de changements importants dans les choix de conception initialement posés et utilisés pour la construction du modèle, il est souvent inévitable de devoir recommencer la construction du modèle ou d'une part importante de celui-ci.

d. « Les changements peuvent être difficiles à visualiser »

En raison de l'interdépendance des paramètres et contraintes à travers tout le modèle, il est parfois difficile de visualiser tous les changements qui vont être effectués lors de la moindre petite modification du modèle. En effet, un modèle qui fonctionne très bien peut devenir inutilisable car le designer ne visualise pas où se trouve le problème engendré par le changement effectué. Pour quelqu'un qui n'a pas conçu le modèle en question, il est souvent impossible d'effectuer des changements sans perturber tout le modèle.

e. « Réutiliser et partager des modèles peut être problématique »

Chaque utilisateur d'un modèle paramétrique doit connaître la structure de ce modèle et son fonctionnement de manière approfondie, selon la remarque du point (d), si bien que, souvent, le créateur originel du modèle est le seul qui ait les capacités de travailler avec. En un sens, la personne qui a conçu et construit le modèle devient le propriétaire de celui-ci.

2.2. Problématique

Ces 5 problématiques ont été reprises par Daniel Davis pour introduire ses recherches concernant la flexibilité des modèles paramétriques. Les trois premières problématiques exposées (a-b-c) ci-dessus sont principalement liées à la flexibilité des modèles, ce problème est souvent mis en évidence par les auteurs. Pour ma part, je me concentrerai principalement sur les deux dernières problématiques énoncées ci-dessus soit : d'une part, la visualisation difficile de changements et, d'autre part, la réutilisation et le partage de modèles paramétriques.

Au cours des manipulations du modèleur Grasshopper, le manque de lisibilité des modèles d'esquisse d'étudiant en architecture est récurrent.

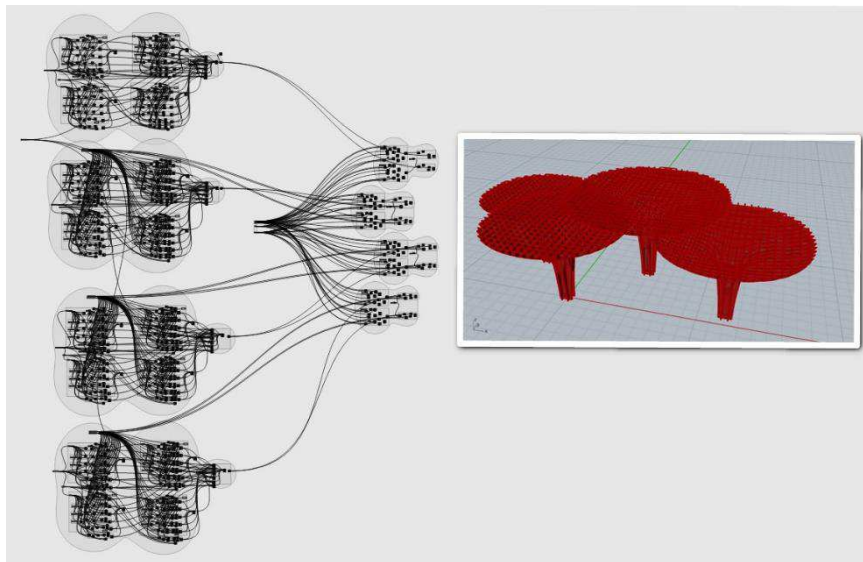


Figure 8 - Modèle paramétrique réalisé sur Grasshopper dans le cadre d'un atelier de conception. On remarque une certaine organisation du code par le positionnement des éléments mais il est rendu illisible par le nombre d'objets et de liens qui en empêchent la compréhension.

Le mémoire de mon prédécesseur Raphaël Devaux (2016) mettait en évidence que ce problème de lisibilité s'observe également au sein d'agence d'architecture lors de la communication et de la transmission des modèles entre différents experts. Ces observations, permettent d'universaliser ce problème de lisibilité et de lui donner une certaine crédibilité.

Finalement, bien que la plupart des personnes concernées par ce sujet développent leurs recherches autour de la problématique du manque de flexibilité des modèles paramétriques, il me semble opportun de consacrer mon travail de fin d'étude à la réflexion d'une solution permettant de simplifier la lisibilité et donc la communication d'un modèle.

Ces observations m'ont ainsi menée à la question de recherche suivante :

« Au vu des problèmes de lisibilité et de communication des modèles, est-il possible d'établir une méthodologie d'aide à la programmation en modélisation architecturale paramétrique ? »

Ainsi, l'objectif de ce travail est, dans un premier temps, l'établissement d'une méthode abordable par tout type d'utilisateur de modèle paramétrique en vue de la conception architecturale, cette méthode prendra le nom de MAPMAP (Méthodologie d'Aide à la Programmation en Modélisation Architecturale Paramétrique). Dans un second temps, la mise en œuvre de cette méthode par des utilisateurs sera étudiée. Enfin, une troisième phase aura pour objectif d'analyser l'efficacité de la méthode proposée en termes de transmission de modèles entre des utilisateurs.

2.3. Cadre de la recherche

À présent que la problématique a été établie, il convient de préciser le domaine de l'étude qui y sera consacré. Pour ce faire, nous commencerons par exposer le cadre de la Méthodologie d'Aide à la Programmation en Modélisation Architecturale Paramétrique (MAPMAP) et, ensuite, nous exposerons comment et pourquoi le domaine d'expérimentation a été établi. Remarquons que l'expérimentation est composée de deux phases qui visent à analyser l'efficacité de la méthodologie : dans un premier temps, du point de vue de sa mise en œuvre et, dans un second temps, du point de vue de sa pertinence dans le cadre de la communication de modèles.

2.3.1. Cadre de la méthode

La méthode qui sera développée dans les chapitres suivants a pour objectif principal de rendre la lecture d'un modèle plus simple pour tout utilisateur. Son but premier n'est donc pas d'améliorer la flexibilité d'un modèle pour rendre celui-ci plus opportuns aux changements dus à une phase de conception architecturale, mais bien d'améliorer sa compréhension, aussi bien pour le concepteur que pour tout autre utilisateur du modèle. L'objectif est de structurer la construction du modèle pour simplifier son adaptabilité et sa communication entre utilisateurs.

Ma volonté est d'établir une méthode adressée aux utilisateurs de modeleurs paramétriques dans le cadre de la conception architecturale. Or, les logiciels destinés à ce type de concepteurs sont généralement composés d'une interface graphique qui facilite la création d'un modèle sans nécessité de connaissance en langage de programmation informatique. En effet, certains reconnaissent que la création d'un modèle paramétrique s'apparente à une forme de programmation :

« *Some people described designing with Pro/ENGINEER to be more similar to programming than to conventional engineering design.* » (Weisberg, 2008, p. 16_12)

C'est pourquoi, la MAPMAP sera établie dans le cadre spécifique de logiciels paramétriques utilisant une interface graphique de programmation visuelle.

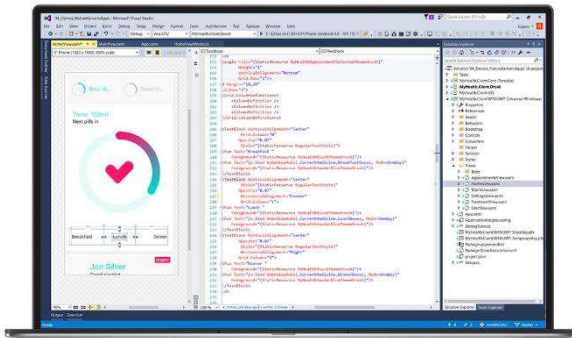


Figure 9 - Interface de programmation textuelle du logiciel VisualBasic (<https://www.visualstudio.com>)

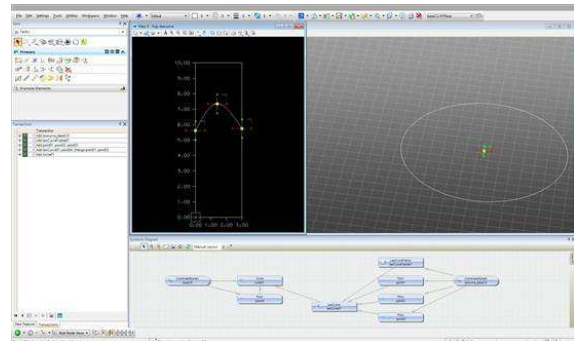


Figure 10 - Interface de programmation visuelle du logiciel GenerativeComponents (<https://communities.bentley.com>)

2.3.2. Cadre de l'expérimentation

Comme dit précédemment, la MAPMAP s'adresse aux concepteurs dans le domaine de l'architecture. Pour réaliser l'étude de la mise en pratique de cette méthode, il a ainsi été décidé que les sujets de l'étude soient des utilisateurs de ce type de logiciel en conception architecturale.

En outre, pour pouvoir analyser les résultats de l'expérience, les sujets doivent présenter une homogénéité au niveau de leur connaissance et de leur expérience avec des logiciels de modélisation paramétrique. C'est dans ce souci d'homogénéité d'échantillonnage, que le choix des sujets s'est porté sur des étudiants ayant reçu la même formation. Il s'agit en l'occurrence des étudiants en ingénieur civil architecte à l'université de Liège.

Enfin, ces étudiants, ayant reçu une formation en modélisation paramétrique avec le modèleur Grasshopper, c'est ce logiciel, composé d'une interface graphique, qui sera au centre de l'étude.

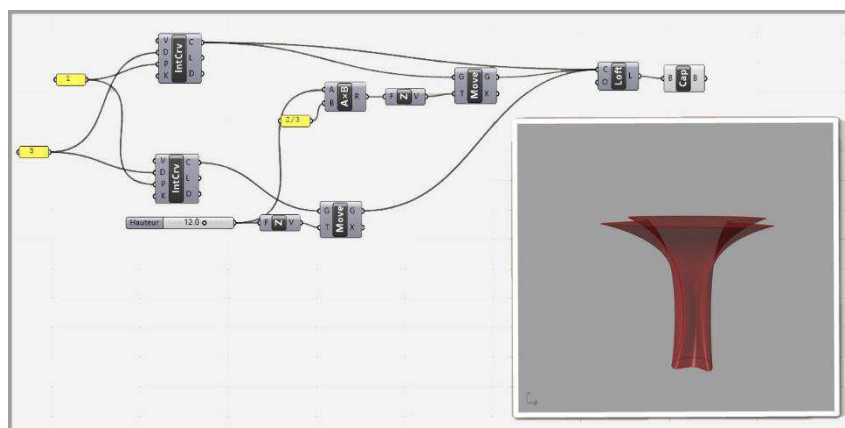


Figure 11 – Interface de programmation visuelle du modèleur Grasshopper, plugin du logiciel Rhinocéros

2.4. Méthodologie

En vue d'étudier la problématique mise en évidence précédemment et suite à l'énonciation de la question de recherche, c'est-à-dire : « Au vu des problèmes de lisibilité et de communication des modèles, est-il possible d'établir une méthodologie d'aide à la programmation en modélisation architecturale paramétrique ? » ; une méthodologie de recherche structurée en quatre étapes est adoptée.

Dans cette étude, nous ne nous focalisons pas sur la conception architecturale mais bien sur la phase de conception d'un modèle paramétrique en vue de participer à la conception architecturale. En conséquence, la première étape, qui fait l'objet du chapitre suivant, consiste à rechercher des méthodes de conception existantes qui ont été développées dans différents domaines : la conception architecturale, la conception d'un programme informatique ou encore la conception d'un modèle paramétrique. L'étude de ces méthodes permet tout d'abord de comprendre pourquoi et comment elles ont été établies. Ensuite, certaines de ces méthodes mettent en évidence la manière dont elles ont évolué selon les besoins réels des utilisateurs. Finalement, ces méthodes présentent des caractéristiques importantes sur lesquelles fonder la MAPMAP.

La deuxième étape a pour objectif d'établir la MAPMAP. Elle est développée sur bases des méthodes étudiées lors de la première étape et sur base de mon expérience en tant qu'étudiante dans le domaine de l'architecture et utilisatrice du logiciel Grasshopper. Alors que les méthodes existantes restent assez génériques, la MAPMAP se veut plus spécifique, si bien qu'elle se consacre au domaine de l'enseignement et au logiciel Grasshopper comme exposé préalablement.

Suite à l'établissement théorique de la MAPMAP, la troisième étape consiste en l'étude de sa mise en pratique à travers la réalisation de deux phases expérimentales :

- a. La première expérience est consacrée au premier objectif de la MAPMAP : rendre un modèle plus lisible. Pour ce faire, nous étudions l'impact de la mise en œuvre de la MAPMAP sur la phase de conception à travers la réalisation de modèles paramétriques par des étudiants.
- b. La seconde expérience est consacrée au second objectif de la MAPMAP : faciliter la communication d'un modèle paramétrique entre utilisateurs. Nous évaluons l'efficacité de communication à travers l'utilisation des modèles réalisés dans la première expérience par des étudiants externes ayant reçu la même formation.

Ces expériences sont réalisées dans un cadre comparatif : avec et sans la MAPMAP. Ainsi les étudiants de la première expérience sont divisés en deux groupes : ceux ayant une approche « naturelle », ou instinctive, et ceux qui auront reçu une formation d'introduction à la MAPMAP. Enfin, avant leur réalisation, ces expériences ont été ajustées sur base d'expériences zéros.

La quatrième et dernière phase de notre étude consiste en l'analyse des résultats des expériences. Cette analyse est réalisée de manière qualitative, et comparative comme annoncé ci-dessus. D'une part, dans le cadre de la première expérience, l'analyse s'effectuera sur base des modèles réalisés par les étudiants et par l'étude des vidéos d'enregistrement de la construction de certains modèles. En effet, deux expériences de construction du modèle auront été filmées de sorte à fournir des informations temporelles sur la mise en œuvre de la MAPMAP. En outre des critères quantitatifs liés à la méthode ainsi que des questionnaires recueillant les avis des étudiants servent à appuyer cette analyse qualitative. D'autre part, la seconde expérience est analysée sur base de mesures temporelles et des avis des étudiants recueillis au sein de questionnaires.

2.5. Définition des concepts

Dans la suite de ce travail, certains termes seront employés couramment pour désigner des parties de modèle dans le cadre de l'utilisation de Grasshopper. Ci-dessous est donc repris un descriptif de quelques-uns pour permettre d'éviter toute confusion sur la manière dont ils sont employés dans ce texte.

Objet

Dans le langage informatique, un 'objet' est une abstraction d'une entité réelle dans le cadre d'un problème ou de sa mise en œuvre dans un domaine virtuel. Il reflète la capacité d'un modèle à conserver des informations et à interagir avec celles-ci (Coad & Yourdon, 1991).

Dans le cadre de ce travail sur la modélisation paramétrique, nous emploierons ce terme 'objet' pour désigner chaque entité constituant un nœud du modèle paramétrique dans l'interface graphique de représentation symbolique de Grasshopper, que cette entité contienne un paramètre, une valeur ou un algorithme.

Paramètre

Rappelons la définition du terme 'paramétrique' selon la Concise Encyclopedia of Mathematics : « *a set of equations that express a set of quantities as explicit functions of a number of independent variables, known as 'parameters'* » (Weisstein, 2003, p. 2150). Nous considérerons donc comme 'paramètre' toute variable indépendante qui constitue une entrée du modèle et qui permet de le définir dans son contexte.

Lien

Dans la conception architecturale, Alexander établit que chaque problème de conception se compose d'exigences et de connections entre ces exigences qui symbolisent leurs dépendances (1971).

De la même manière, lors de la réalisation d'un modèle paramétrique à travers une interface graphique telle que celle de Grasshopper, le modèle est représenté par un graphe d'objets connectés entre eux selon leurs interdépendances. Ce sont ces connections que nous appellerons 'liens' ou 'liaisons' dans la suite de ce travail.

3. MÉTHODES DE CONCEPTION

Comme je l'annonçais dans la présentation de la méthodologie de recherche adoptée pour la réalisation de ce mémoire, la première étape de ce travail est consacrée à la recherche de méthodes existantes sur base desquelles établir ma méthodologie. C'est la présentation de ces méthodes qui fait l'objet de ce chapitre.

Plan du chapitre :

3.1.	LA CONCEPTION EN ARCHITECTURE SELON CHRISTOPHER ALEXANDER.....	24
	3.1.1. <i>Méthode de décomposition d'un problème de conception.....</i>	24
	3.1.2. <i>Le langage des Patterns.....</i>	28
3.2.	LES PRINCIPES DE DESIGN ORIENTÉ-OBJET.....	30
	3.2.1. <i>Les origines du Design Orienté-Objet.....</i>	30
	3.2.2. <i>Les concepts et principes du Design Orienté-Objet.....</i>	32
3.3.	LE DESIGN EN MODÉLISATION PARAMÉTRIQUE D'APRÈS ROBERT WOODBURY....	35
	3.3.1. <i>Comportement d'utilisation des modeleurs paramétrique.....</i>	35
	3.3.2. <i>La méthode des Patterns.....</i>	37

Le travail de conception peut se faire à l'aide de différents outils et notamment les outils de modélisation paramétrique. Remarquons que dans cette étude, nous ne nous focalisons pas sur la phase de conception architecturale mais bien sur la phase de conception d'un modèle paramétrique en vue de participer à la conception architecturale.

Au cours des dernières décennies, des auteurs ont déjà tenté de rationaliser ce travail de conception en établissant des méthodes dans divers domaines. Ici nous nous intéresserons à certaines théories développées dans les domaines suivants :

- a. D'abord nous nous intéresserons au domaine de l'architecture car, le public visé étant de ce domaine, il convient d'établir une méthode proche de celles appliquées dans le cadre de la conception architecturale.
- b. Ensuite, nous aborderons le domaine de la programmation informatique. En effet, de nombreux utilisateurs de modeleurs paramétriques s'accordent à dire que la démarche adoptée dans le cadre de la modélisation paramétrique ressemble à celle développée dans le cadre de la programmation informatique (Weisberg, 2008). La MAPMAP qui sera développée dans la suite de ce travail a donc un intérêt potentiel à s'inspirer des principes qui furent établis par le passé pour répondre à une problématique similaire dans le cadre de la programmation et qui, à l'heure actuelle, ont déjà fait leurs preuves.
- c. Finalement, certains auteurs se sont déjà consacrés au développement d'une méthodologie dans le cadre de la modélisation paramétrique, nous exposerons donc également l'un de ces travaux. En l'occurrence, il s'agit du langage des patterns développé par Robert Woodbury (2010).

Bien que ces domaines d'application varient, nous verrons que les méthodes présentées s'inspirent des précédentes et présentent ainsi des principes communs, principes qui, aux cours des décennies, ont prouvé leur pertinence.

En vue d'établir la MAPMAP, nous nous intéressons donc à certaines de ces théories pour développer une méthode spécifique, abordable par tout un chacun, et basée sur l'utilisation du logiciel Grasshopper. Parmi les méthodes que nous allons exposer ci-dessous, on retrouve la théorie des patterns développée par Christopher Alexander dans le cadre de la conception architecturale, la méthode des « Object Oriented Principles » établie dans le domaine de la programmation informatique et, finalement, la théorie de Robert Woodbury qui s'adresse déjà au domaine de modélisation paramétrique mais de manière assez générique au vu de ce qui sera proposé par la suite.

3.1. La conception en architecture selon Christopher Alexander

Christopher Alexander est professeur en architecture à l'université de Californie Berkeley, et il est le père du mouvement « Pattern Language » en architecture comme en programmation. Ici nous nous intéressons successivement à deux de ses écrits « De la synthèse de la forme, essai » (1971) et « A Pattern Language » (1977).

Dans la suite de ce chapitre, nous observerons qu'il a eu une grande influence sur le domaine de la programmation notamment dans la méthode Orientées-Objets qui sera présentée ensuite, ainsi que dans le domaine de la modélisation paramétrique avec le travail de R. Woodbury présenté dans son ouvrage « Elements of Parametric Design » (2010) et qui sera exposé dans la dernière partie de ce chapitre.

3.1.1. Méthode de décomposition d'un problème de conception

Dans son premier ouvrage, Alexander met en évidence les difficultés de la composition architecturale pour les concepteurs d'aujourd'hui. Pour lui, toute composition a pour objectif de répondre simultanément à différentes exigences reliées entre elles par des interactions.

Alexander n'approche pas la conception comme un problème d'optimisation mais plutôt comme un problème d'harmonie entre la forme et son contexte. L'ensemble de ces deux éléments peut être caractérisé par un certain nombre d'inadaptation et ainsi la solution du problème réside dans la conception d'une forme qui ne fasse pas apparaître ces inadaptations.

Dans sa représentation, Alexander utilise des nœuds qui symbolisent les exigences, les contraintes, que l'objet de la conception doit remplir. En outre, ces nœuds sont reliés par des barres symbolisant les interactions entre ces exigences. Des signes – et + sont utilisés pour caractériser respectivement les interactions conflictuelles et concordantes.

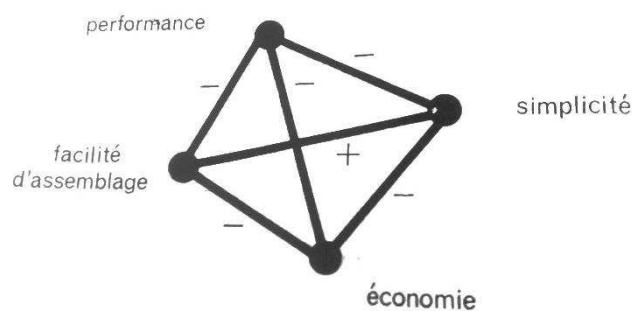


Figure 12 - Problème de composition typique avec des exigences et des interactions entre elles
(Alexander, 1971, p. 2)

À l'heure actuelle, les exigences de composition sont de plus en plus nombreuses si bien que les problèmes engendrés par celles-ci se complexifient et atteignent des niveaux insolubles. Même des problèmes qui peuvent sembler simple au premier abord, peuvent en réalité cacher différents besoins qui deviennent trop complexes pour être saisis intuitivement.

Alexander pose comme hypothèse que la capacité humaine à l'invention est limitée si bien que la difficulté d'un problème que l'homme peut résoudre, est également limitée. Passé un trop grand nombre d'exigences, une décision trop complexe, un concepteur n'est pas capable d'y répondre avec sagesse bien qu'aucun cas de limite absolue n'existe.

L'objectif d'Alexander est alors d'élaborer une méthode de représentation des problèmes qui les rende plus simples à résoudre en réduisant la taille du problème à une échelle abordable par le concepteur.

L'adaptation entre forme et contexte

Notre monde étant hétérogène, ce sont ses irrégularités qui permettent l'existence de la forme et des forces qui la contraignent. D'après Alexander, la forme est la solution du problème de conception tandis que le contexte permet de délimiter ce problème. Le contexte fait donc partie intégrante du problème de conception.

Ainsi, d'après l'auteur, le travail du concepteur a pour objectif de trouver une forme qui réponde au contexte qui la caractérise. Cependant, ce qui fait la difficulté de la tâche, est le discernement de la limite entre le contexte et la forme. Pour qu'une clarté physique puisse être atteinte, Alexander émet l'hypothèse qu'une clarté « programmatic », dans l'esprit et les actions du concepteur, est nécessaire : le concepteur doit pouvoir ramener son problème à ses premières origines fonctionnelles et trouver une manière de les structurer.

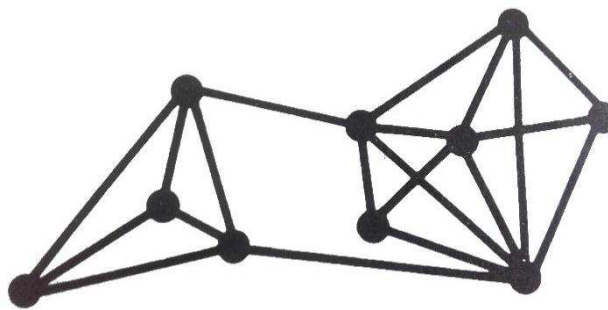


Figure 13 - Représentation d'un système de variables d'inadaptation interconnectées
(Alexander, 1971, p. 35)

Ensuite, Alexander établit qu'aucun système complexe ne peut s'adapter dans un délai raisonnable hormis dans le cas où l'adaptation peut être réalisée sous-système par sous-système, ces sous-systèmes étant suffisamment indépendant les uns vis-à-vis des autres.

Ces exigences reliées entre elles ne peuvent être fixées indépendamment des autres. Hors ces variables sont connectées avec diverses intensités si bien que des sous-systèmes pourront être identifiés dans la structure du problème de conception.

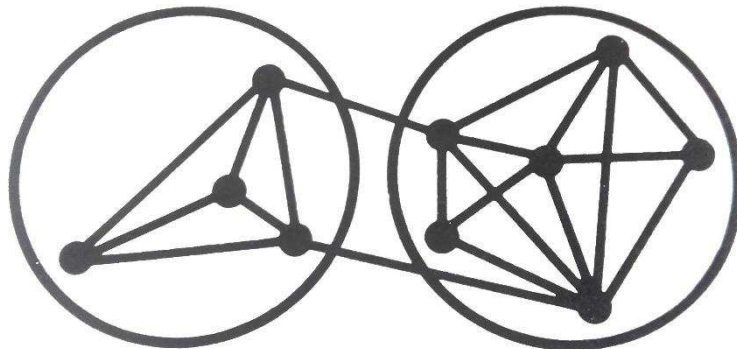


Figure 14 - Décomposition du système en sous-système
(Alexander, 1971, p. 36)

Ainsi l'auteur décrit le processus de conception de la forme « *comme l'action d'une suite de sous-système, tous reliés entre eux, mais d'une manière suffisamment lâche pour pouvoir s'ajuster indépendamment dans un laps de temps « faisable »* » (Alexander, 1971, p. 37).

Les concepts formant les sous-systèmes

Les problèmes de composition actuels sont très complexes et les concepteurs ne peuvent se référer à toutes les inadaptations à chaque fois qu'ils réfléchissent au problème si bien qu'intuitivement, ils ordonnent les éléments en classes auxquelles ils donnent un nom. Comme dit précédemment, il y a des limites au nombre de concept que l'homme peut gérer simultanément. La codification du problème en classes permet de garder une vue global de l'ensemble du problème. Cependant, ces concepts arbitraires ne permettront pas au concepteur de trouver une solution adaptée s'ils ne correspondent pas aux sous-systèmes du système qui eux ne présentent pas ce caractère arbitraire.

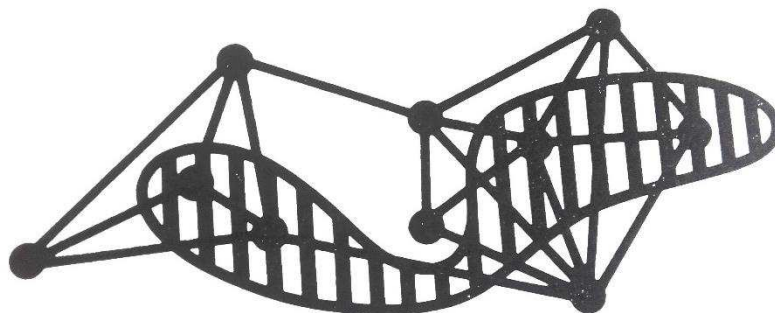


Figure 15 - Décomposition du système selon des concepts arbitraires
(Alexander, 1971, p. 55)

Le programme d'Alexander

Sur base de ses observations, Alexander propose un programme de résolution du problème de conception qui peut être résumé par ces quelques phrases:

« *Le problème est défini par un ensemble d'exigences appelé M. La solution de ce problème sera une forme qui satisfera avec succès toutes ces exigences. [...] Le programme est une hiérarchie des sous-ensembles les plus significatifs de M. Chaque sous-ensemble constitue un sous-problème ayant son intégrité propre.* » (Alexander, 1971, p. 78)

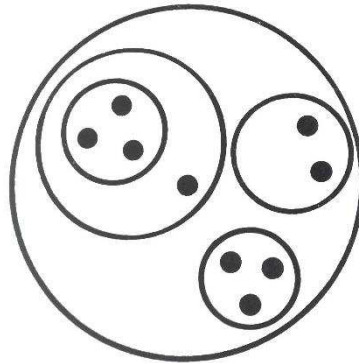
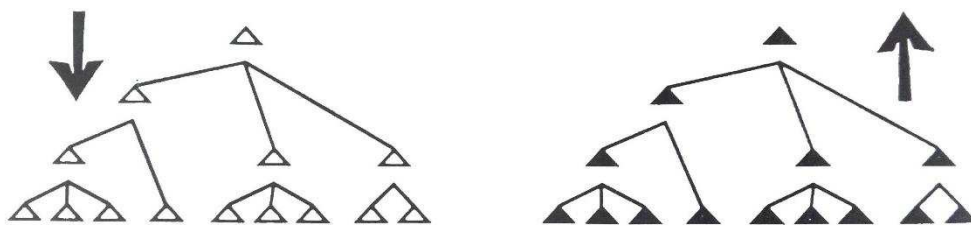


Figure 16 – Ensemble d'exigences M structuré en sous-ensemble
(Alexander, 1971, p.68)

« *Chaque sous-ensemble peut-être transcrit en un diagramme constructif. Et chacun de ces sous-ensembles de M, du fait qu'il contient moins d'exigences que M lui-même, et moins d'interactions entre elles, est plus simple à transcrire en diagramme que M.* » (Alexander, 1971, p. 78)



Programme, consistant en ensembles

Réalisation, consistant en diagrammes

Figure 17 - Diagramme constructif de réalisation du programme
(Alexander, 1971, p. 78)

Dans ses propos, Alexander met en évidence qu'il y a une importante correspondance entre la structure d'un problème et le processus de conception qui répond à ce problème par la création d'une forme physique.

3.1.2. Le langage des Patterns

Suite à la théorie de décomposition d'un problème de conception en sous-problèmes développée par Alexander en 1971, celui-ci propose un langage composé de 253 entités, appelées patterns, qui a pour objectif de rationaliser la conception architecturale. C'est le fonctionnement de ce langage, décrit dans « A Pattern Language » (Alexander, 1977), qui est exposé ci-dessous.

Commençons par expliquer la notion de pattern. Ce terme peut être traduit de l'anglais comme un « *modèle, exemple, échantillon, type, patron (en couture), motif* » (Arnold, 1977, p.154 ; cité par de Boissieu, 2010, p. 101) selon le domaine dans lequel il est employé. Dans le cadre de la conception architecturale, voici comment Alexander le décrit : « *Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.* » (1977, p. X). Chaque pattern décrit un problème qui apparaît fréquemment dans notre environnement et il décrit le noyau d'une solution à ce problème, si bien qu'on peut utiliser cette solution un million de fois sans jamais résoudre le problème de la même manière.

En réponse, à son essai, écrit en 1971, Alexander structure le langage qu'il propose avec des patterns de différentes tailles : on commence avec des patterns très larges, tels que les régions et les villes, pour aller vers des patterns de plus en plus petit, tels que le voisinage, puis le bâtiment, la pièce, et finalement les détails techniques de construction. Ainsi les patterns sont utilisés dans un ordre formant une séquence linéaire basée sur les connections entre les patterns. Aucun pattern n'est une entité isolée, au contraire, chaque pattern est connecté à des patterns d'ordre supérieur, qui viennent avant dans la séquence, pour les compléter de manière plus détaillée et, identiquement, chaque pattern est lui-même complété par des patterns d'ordre inférieur, qui viennent après dans la séquence.

En outre, chaque pattern présente le même format descriptif et est organisé selon deux objectifs : montrer ses liens avec les autres patterns du langage et présenter son problème et sa solution de manière à ce que chacun puisse les juger par lui-même selon son propre contexte et les modifier sans perdre leur essence.

Finalement, le format des patterns est le suivant. D'abord une illustration montre un exemple architectural du pattern et un paragraphe d'introduction expose le contenu du pattern, en expliquant comment il aide à compléter certains patterns d'ordre supérieur. Ensuite, un titre donne l'essence du problème en quelques phrases et le cœur du problème est développé dans les paragraphes suivants qui forment ainsi la partie la plus longue du pattern. Par la suite, un nouveau titre décrit la solution du problème sous forme d'instruction et celui-ci est suivi d'un

diagramme qui indique les composantes principales du pattern. Enfin, le pattern se termine avec un paragraphe explicatif des patterns d'ordre inférieur qui servent à compléter, embellir, ce pattern.

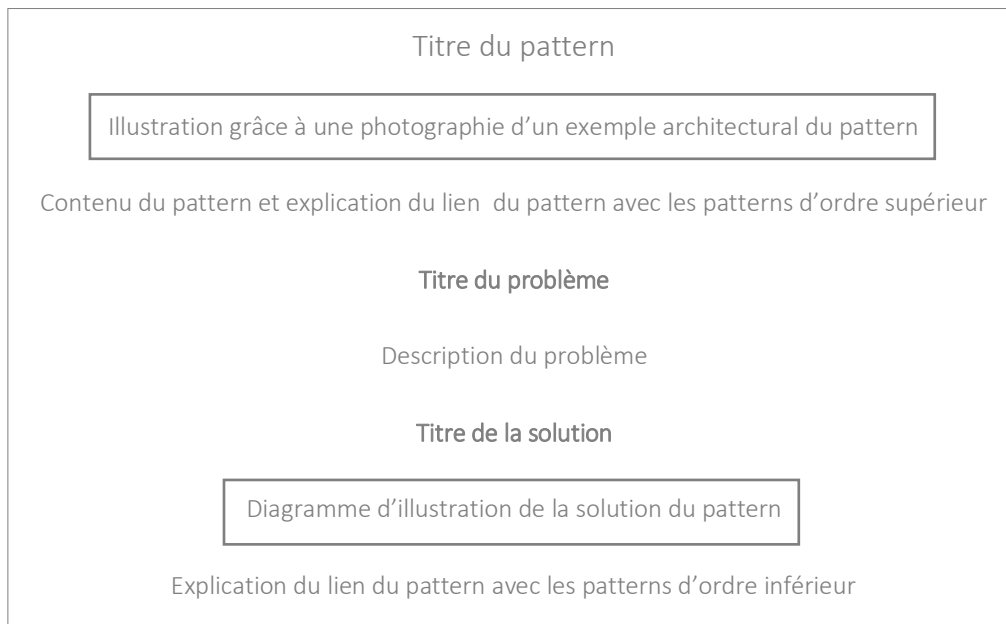


Figure 18 - Schéma de structure d'un pattern

Un exemple concret de pattern se trouve à l'annexe 1. Il s'agit du premier pattern d'Alexander : Independent Region.

3.2. Les Principes du Design Orienté-Objet

En informatique, et plus précisément dans le domaine de la programmation, un nouveau type de langage a commencé à voir le jour dans les années 60 : le langage de programmation orienté-objet. Pour en exposer les principes, nous nous basons principalement sur deux écrits : le premier, « Object-Oriented Design » (Coad & Yourdon, 1991), remonte aux années de l'essor de ce langage ; tandis que le second, « Object-Oriented Analysis and Design » (Dathan & Ramnath, 2010), apporte un point de vue plus récent permettant de retracer l'évolution de ce langage et ces principes.

Avant de parler des langages orientés-objets, il me semble important d'introduire la notion de programmation. D'après Dathan et Ramnath (2010), elle peut se définir comme le processus de création d'un logiciel en traduisant une description d'un langage naturel en une séquence d'opérations exécutables par un ordinateur. Initialement, ce processus de programmation était structuré par deux étapes : premièrement, la reconnaissance des valeurs d'entrées et de sorties du système ; deuxièmement, la décomposition du système en modules fonctionnels. Cependant au début de la programmation, la structure des langages manquait de flexibilité et de lisibilité si bien que les codes existants pouvaient difficilement être transformés ou réutilisés (Dathan & Ramnath, 2010).

Cette problématique de design d'un code en vue de le rendre plus lisible et donc plus facilement adaptable ou réutilisable est identique à la problématique actuelle de la modélisation paramétrique exposée dans ce travail. C'est pourquoi nous nous intéressons ici aux principes de design orientés-objet qui constituent une des solutions d'aide à la programmation informatique pour les concepteurs de programmes.

3.2.1. Les origines de la méthode Orientée-Objet

La méthode Orientée-Objet s'est développée sur base d'un double objectif : d'une part, trouver une meilleure manière d'agencer les modules de code entre eux en permettant d'en cacher les détails et, d'autre part, trouver un moyen flexible de réaliser les assignations (Estival & Cerri, 2005).

Dans les années 90, la méthode Orientée-Objet est en plein essor et devient le moyen le plus populaire d'analyser, designer et développer des logiciels. Dans cette méthode, comme son nom l'indique, l'objet a un rôle central : les éléments d'une situation sont décomposés en objets et en interactions entre ces objets. Le système est une collection d'objets de différents types qui interagissent entre eux (Dathan & Ramnath, 2010). On y retrouve donc des similitudes

avec la conception architecturale du point de vue d'Alexander et avec la structure d'un modèle paramétrique.

Finalement, l'utilisation d'un design Orienté-Objet a pour rôle d'améliorer la productivité, d'augmenter la qualité des programmes et de faciliter leur exploitation. Pour ce faire, le concepteur devra utiliser une syntaxe de représentation uniforme au sein du programme, réutiliser des codes existants et maintenir ses programmes à jour (Coad & Yourdon, 1991).

Historique des langages Orientés-Objets

Le concept d'objet apparaît pour la première fois dans les années 60, lors de la création du langage de programmation Simula 67. En outre le premier langage de programmation Orienté-Objet, Smalltalk, n'apparaîtra qu'en 1980. C'est de ce langage que provient la majeure partie du vocabulaire utilisé dans la méthode Orientée-Objet (Dathan & Ramnath, 2010). Il est intéressant de remarquer qu'Alan Kay et son équipe du Xerox Park, qui ont créé le langage Smalltalk et à travers celui-ci la programmation Orientée-Objet, avaient pour objectif initial de créer un langage informatique naturel et abordable par des amateurs (Small Talk en anglais signifiant 'papotage' soit un langage non formel). Ensuite, les années 80 virent apparaître de nouveaux langages Orientés-Objet et, parmi ceux-ci, il y a notamment le langage Ada, en 1983, qui introduisit les concepts d'héritage, de polymorphisme et de liaisons dynamiques (Dathan & Ramnath, 2010).

Parallèlement à cela, dans les années 70, un concepteur, Bjarne Stroustrup, se lance dans la recherche d'un langage de simulation qui soit plus performant que Simula 67. Son étude le mènera à la création d'un langage Orienté-Objet dérivé du langage C, soit le langage C++ qui fut standardisé en 1997 et qui devint et continue à être l'un des plus célèbres langages de programmation (Dathan & Ramnath, 2010).

Finalement, dans les années 90, on retrouve deux grands événements : d'abord, en 1994, la sortie du livre « Design Patterns » de Gamma et al. (inspiré par le langage des patterns d'Alexander en conception architecturale) qui considère 23 questions de design spécifiques et leur fournit une solution générique via une approche orientée-objet sous forme de patterns ; ensuite, en 1996, la création du langage JAVA qui est un dérivé du C++ plus simple d'utilisation (Dathan & Ramnath, 2010). Ces années 90 furent un tournant dans la méthode Orientée-Objet car on y observe quatre changements d'approche (Coad & Yourdon, 1991) :

- a. Les concepts de l'approche Orientée-Objet sont passés de moyen de codage à des moyens d'analyse et de design.
- b. Les technologies de construction de modèle sont devenues plus puissantes et plus adaptées.

- c. Les modèles sont devenus plus complexes que dans les années précédentes, et des systèmes interactifs simplifie l'interaction avec l'utilisateur.
- d. Les systèmes développés sont plutôt « domain-oriented » : la priorité n'est plus de modéliser des données mais de modéliser les responsabilités et le domaine du problème.

Terminologies

Comme dit précédemment, un vocabulaire spécifique à cette méthode s'est développé avec elle. Pour pouvoir aborder les concepts du langage Orienté-Objet et les principes de programmation qui en ont découlé, nous allons expliquer quelques-uns de ses termes.

Rappelons qu'un **objet** est une abstraction d'une entité réelle dans le cadre d'un problème ou de sa mise en œuvre dans un domaine virtuel. Il reflète la capacité d'un modèle à conserver des informations et à interagir avec celles-ci (Coad & Yourdon, 1991).

Une **classe** est un ensemble d'un ou plusieurs objets descriptibles avec une uniformité de caractéristiques. En outre, ces caractéristiques indiquent comment créer de nouveaux objets dans la classe (Coad & Yourdon, 1991).

Les classes partagent trois types de relation qui sont :

- a. L'**association** : Il s'agit de la relation la plus courante. Elle indique simplement que des objets sont reliés entre eux sans aucun rapport hiérarchique (Dathan & Ramnath, 2010). « *Association. The union or connection of ideas.* » (Webster's, 1977 cité par Coad & Yourdon, 1991, p. 8)
- b. L'**héritage** : Il s'agit d'une relation hiérarchique entre plusieurs classes, basée sur la généralisation et la spécification (Dathan & Ramnath, 2010). L'héritage signifie que les classes spécifiques héritent des caractéristiques des classes génériques dont elles proviennent.
- c. La **généricité** : Il s'agit d'une relation d'héritage plus restreinte car la spécification des classes se fait uniquement au niveau du type de paramètre (Dathan & Ramnath, 2010).

3.2.2. Les concepts et principes de la méthode Orientée-Objet

Concepts

À présent, exposons les sept principaux concepts qui forment la méthode Orientée-Objet et qui ont été développés en vue d'augmenter la qualité des programmes et de faciliter leur exploitation et leur maintenance.

La **modularité** consiste à développer des composants distincts indépendamment et à les intégrer dans un système plus large pour compléter la requête fonctionnelle (Dathan & Ramnath, 2010). C'est composants distincts sont appelés 'modules'.

L'**encapsulation** a pour objectif de cacher certaine partie d'un programme ou certains détails d'un module à des acteurs externes (Dathan & Ramnath, 2010).

« *Encapsulation (Information Hiding). A principle, used when developing an overall program structure, that each component of a program should encapsulate or hide a single design decision... The interface to each module is defined in such a way as to reveal as little as possible about its inner workings.* » (Oxford, 1986 cité par Coad & Yourdon, 1991, p. 7)

La **cohésion** est la mesure de la concentration des responsabilités au sein d'un module. Dans la méthode Orientée-Objet, une grande cohésion est souhaitée (Dathan & Ramnath, 2010). La cohésion est le degré avec lequel les éléments d'un module contribuent à un même but, unique et défini.

Le **couplage** est la mesure de la dépendance des modules entre eux. Dans la méthode Orientée-Objet, un faible couplage est souhaité (Dathan & Ramnath, 2010). En effet, le couplage est le degré d'interdépendance des modules. Il s'agit donc d'une caractéristique importante lorsqu'on évalue un programme car un changement dans un module doit avoir un impact minimum sur les autres (Coad & Yourdon, 1991).

La **modificabilité** est l'aptitude à changer la fonctionnalité d'un module pour permettre au système d'être plus adaptable (Dathan & Ramnath, 2010).

La **testabilité** est la facilité avec laquelle on peut détecter des problèmes dans le programme grâce à la structure du système (Dathan & Ramnath, 2010).

L'**abstraction** est la capacité d'ignorer les aspects non-révélant du but poursuivi afin de se concentrer uniquement sur ceux qui le sont : « *Abstraction. The principle of ignoring those aspects of a subject that are not relevant to the current purpose in order to concentrate more fully on those that are.* » (Oxford, 1986 cité par Coad & Yourdon, 1991, p. 6).

Principes de la méthode Orientée-Objet

Sur base des concepts de la méthode Orientée-Objet, des principes ont été établis pour guider le concepteur lors de la création d'un programme. Ces principes sont répartis en trois catégories : ceux en lien avec les classes d'objets, ceux en lien avec la cohésion d'un module et ceux en lien avec le couplage des modules.

Les cinq premiers principes se rapportent aux classes d'objets :

- a. The Single Responsibility Principle (SRP) : le principe de responsabilité unique est que chaque classe a une seule raison de changer.
- b. The Open-Closed Principle (OCP) : le principe ouvert-fermé dit que chaque classe doit être capable de s'étendre (ouverte) sans être modifiée (fermée).
- c. The Liskov Substitution Principle (LSP) : le principe de substitution de Liskov indique qu'une classe dérivée doit être substituable par sa classe générique.
- d. The Interface Segregation Principle (ISP) : le principe de ségrégation de l'interface consiste à encapsuler certaines parties d'un programme pour ne laisser apparentes que les commandes nécessaires à l'exploitation du programme.
- e. The Dependency Inversion Principle (DIP) : le principe d'inversion de dépendance dit que les classes importantes ne doivent pas dépendre des classes de moindre importance, leurs dépendances doivent être abstraites et non concrètes.

Ensuite, viennent trois principes se rapportant au concept de cohésion :

- a. The Release/Reuse Equivalency Principle (REP) : le principe d'équivalence de sortie/communication indique que seuls les éléments qui sont communiqués par un système peuvent être réutilisés, donc le degré de réutilisation est le degré de communication.
- b. The Common Closure Principle (CCP) : le principe de fermeture commune dit que les classes qui subissent les mêmes changements doivent être groupées ensemble.
- c. The Common Reuse Principle (CRP) : le principe de réutilisation commune consiste à grouper ensemble les classes qui sont utilisées ensemble.

Finalement, les trois derniers principes sont en lien avec le concept de couplage :

- a. The Acyclic Dependencies Principle (ADP) : le principe de dépendances acycliques indique de ne pas créer de cycle au sein d'un graphe de dépendance.
- b. The Stable Dependencies Principle (SDP) : le principe de dépendances stables dit que les dépendances des modules doivent aller dans le sens de la stabilité, soit qu'un module ne peut dépendre que de modules plus stables que lui-même.
- c. The Stable Abstractions Principle (SAP) : le principe des abstractions stables consiste à dire que les modules les plus abstraits sont les plus stables tandis que les modules les plus concrets et donc plus spécifiques sont les plus instables.

3.3. Le design en modélisation paramétrique d'après Robert Woodbury

Robert Woodbury est professeur depuis 1982 et travaille actuellement à la faculté des arts et technologie de l'Université Simon Fraser de Vancouver où il développe des recherches dans les domaines de la conception informatique, l'analyse visuelle, et la programmation ciblée utilisateur.

Son ouvrage principal à ce jour, « Elements of Parametric Design » (2010), se consacre à la conception paramétrique. C'est le contenu de cet ouvrage qui nous intéresse dans la suite de ce sous-chapitre. En effet, Woodbury y a développé une aide à la conception en modélisation paramétrique, sur base des théories d'Alexander dans le domaine de l'architecture.

Au sein de cet ouvrage, nous retiendrons deux chapitres. D'abord, nous nous attacherons au chapitre 3, « How designers use parameters », qui contient les observations de Woodbury et son équipe concernant les comportements des concepteurs de modèles paramétriques (du point de vue de leurs compétences et de leurs stratégies). Ensuite, nous nous attarderons sur le chapitre 8, « Patterns for parametric design » qui expose la méthode des patterns développée par Woodbury et son équipe sur base de leurs observations dans le cadre de la modélisation paramétrique.

3.3.1. Comportement d'utilisation des modeleurs paramétriques

Compétences

D'après Woodbury, les utilisateurs de modeleurs paramétriques font principalement appel à 6 compétences, lors de l'utilisation de ce type de logiciel. Certaines de ces compétences sont semblables à celles utilisées dans d'autres domaines de conception tandis que d'autres sont nouvelles et spécifiques à la modélisation paramétrique. Ces compétences sont les suivantes :

- a. *Conceiving data flow* : au fur et à mesure de l'augmentation de la complexité d'un modèle paramétrique, les dépendances au sein du modèle augmentent en complexité et la visualisation du flux des données devient difficile. L'une des compétences nécessaire au concepteur en modélisation paramétrique est de gérer ce flux de données au sein du modèle.
- b. *Dividing to conquer* : la modélisation paramétrique permet l'utilisation d'une stratégie de « divide-and-conquer », soit de division du modèle en sous-systèmes dont les interdépendances sont limitées, comme le propose Alexander dans le cadre des problèmes de conception architecturale.

- c. *Naming* : les parties d'un modèle sont nommées, il ne s'agit pas d'une loi physique mais d'une bonne pratique chez les concepteurs.
- d. *Thinking with abstraction* : rendre un modèle abstrait permet de l'utiliser dans de nouvelles situations. Lors de la construction d'un modèle paramétrique, il convient de réduire le nombre de paramètres d'entrées aux essentiels et de supprimer les références inutiles pour rendre le modèle plus abstrait.
- e. *Thinking mathematically* : les modèles, et particulièrement les paramétriques, sont composés de relations mathématiques. Les concepteurs ne font pas des mathématiques mais ils les utilisent, il leur est donc nécessaire de comprendre ces mathématiques dont les systèmes paramétriques permettent d'encoder des théorèmes et formulations mathématiques.
- f. *Thinking algorithmically* : la conception d'un modèle paramétrique s'effectue par la création d'un graphe composé de nœuds interconnectés. Ces nœuds sont des algorithmes que l'utilisateur emploie et peut éventuellement modifier, un algorithme étant un ensemble d'instances compréhensibles et exécutables par un ordinateur ayant pour objectif de résoudre un problème.

Stratégies

Woodbury et son équipe ont également observé 9 stratégies particulières et complémentaires aux compétences précitées, qui sont mises en œuvre par les utilisateurs lors de la conception de modèles paramétriques. Ces stratégies sont les suivantes :

- a. *Sketching* : Le dessin fait partie intégrante du travail des designers depuis toujours. Les modèles paramétriques, par leur nature dynamique, permettent de réduire le nombre de dessins manuels. Cependant, leur création étant complexe et coûteuse en temps, il convient de trouver un équilibre entre la conception par dessin manuel et ce que peut lui apporter la modélisation paramétrique.
- b. *Throw code away* : Les concepteurs de modèles paramétriques ont tendance à modéliser ce dont ils ont besoin uniquement si bien que, lors de changements, ils ont tendance à recommencer un nouveau modèle plutôt qu'à réutiliser des modèles ou parties de modèles déjà construits.
- c. *Copy and modify* : Bien que les concepteurs ont cette tendance à jeter leurs modèles et les reconstruire, ils ont également pour habitude de rechercher des modèles existants pour les réadapter à leur propre contexte. En effet, bien qu'ils n'aient pas passer du temps à réaliser un code clair, ils aiment utiliser des modèles clairs et qui fonctionnent, préconçus par d'autres.
- d. *Search for form* : L'une des principales opportunités offerte par la modélisation de type paramétrique est la recherche d'une forme. Ce type de modèle dynamique permet

- d'explorer un éventail de possibilités dont des formes typiques du style paramétrisme, soit une architecture dont la géométrie est à base de surfaces nurbs.
- e. *Use mathematics and computation to understand design* : Il est souvent nécessaire de comprendre les fondements mathématiques des algorithmes pour créer un modèle paramétrique. En outre, de nombreux concepteurs explorent les stratégies de conceptions basées sur les mathématiques.
 - f. *Defer decisions* : La conception d'un projet nécessite généralement un contexte et un certain degré de précision géométrique. Dans la conception via un logiciel paramétrique, il est possible de créer le modèle en créant un réseau de relations sans contexte précis. Il s'agit d'une nouvelle stratégie de conception : reporter ; en introduisant les informations de contextualisation et précision du modèle après l'avoir construit.
 - g. *Make modules* : Lorsque les modèles atteignent des tailles importantes, ils deviennent difficiles à comprendre. Il convient donc de mettre en œuvre une stratégie de création de modules qui a pour objectifs d'encapsuler des parties du modèle relativement indépendantes.
 - h. *Help others* : Certains pratiquants de modélisation paramétrique partagent volontiers leurs modèles. Il s'agit là d'une méthode permettant de s'intéresser à de nouveaux problèmes et d'approcher de nouvelles solutions. Aider les autres est une pratique régulièrement observée dans le cadre de l'utilisation de modeleurs paramétriques.
 - i. *Develop your toolbox* : Les caractéristiques exposées précédemment sont des bases. Woodbury et son équipe considèrent que de nouvelles techniques et stratégies sont à développer grâce aux connaissances accumulées par les utilisateurs et l'expérience des écoles qui investissent dans l'enseignement de ce type d'outils.

Ces observations de compétences et stratégies développées par les concepteurs vont ensuite permettre à Woodbury de développer sa méthode en créant un langage de patterns. Dans la suite de cet écrit, nous nous appuyerons également sur ces observations pour relever les caractéristiques importantes à introduire dans la MAPMAP.

3.3.2. La méthode des patterns

Woodbury et son équipe de l'université Simon Fraser ont passés des années à étudier le comportement d'utilisateurs de modeleurs paramétriques. Sur bases de ces observations, dont une partie a été exposée ci-dessus, ils ont mis au point des patterns spécifiques à la modélisation paramétrique pour aider les concepteurs dans la construction de leurs modèles.

En effet, selon eux : « *Reusable, abstract parts are a keystone for professional practice.* » (Woodbury, 2010, p. 185). À la manière d'Alexander, chaque pattern établi par Woodbury est

une solution générique à un problème et est composé du problème, de sa solution et d'informations contextuelles supplémentaires. L'objectif initial de ces patterns est de simplifier la création d'un modèle en intervenant après la réflexion sur la conception du modèle et avant sa construction par la création d'un graphe composé d'algorithmes. Finalement, ces patterns sont généralement utilisés dans le cadre de l'apprentissage et de l'enseignement des logiciels de modélisation paramétrique.

Woodbury a conçu ces patterns sur base de l'utilisation du logiciel GenerativeComponents, là où nous nous concentrerons plutôt sur le logiciel Grasshopper dans le cadre de ce travail. Mais grâce à leur caractère générique, ces patterns sont pour la plupart applicables à d'autres logiciels que celui à partir duquel ils ont été fondés.

Ainsi, Woodbury et son équipe ont mis au point 13 patterns répartis en 5 catégories : la première catégorie se compose d'un unique pattern CLEAR NAMES ; la seconde catégorie se consacre aux techniques de structuration avec les patterns CONTROLLER, JIG, INCREMENT et REACTOR ; la troisième catégorie permet de spécifier et localiser des objets grâce aux patterns POINT COLLECTION et PLACE HOLDER ; la quatrième catégorie comprend les patterns qui extraient des informations du modèle PROJECTION, REPORTER et SELECTOR ; enfin, la cinquième et dernière catégorie reprend les patterns résiduels, dont les principes sont plus complexes, c'est-à-dire MAPPING, RECURSION et GOAL SEEKER. Le tableau ci-dessous, décrit brièvement le rôle de chacun de ces patterns.

Nom	Rôle
Clear Names	Nommer de manière claire et pertinente les objets
Controller	Contrôler une partie du modèle à travers une autre
JIG	Créer une partie du modèle à part, de manière abstraite et non localisée pour l'introduire ensuite dans le modèle global
Increment	Produire un changement à travers une série de valeurs étroitement liées, il en existe de deux sortes : une variation croissante d'unités ou une variation continue et infinie entre deux limites
Point Collection	Regrouper une collection de points localisés
Place Holder	Déterminer une localisation sur le modèle pour y placer un élément spécifique
Projection	Transformer un objet dans un autre contexte géométrique
Reactor	Faire répondre un objet à la proximité d'un autre (application particulière d'un "controller" dont la propriété de contrôle est la proximité)
Reporter	Extraire les informations d'un modèle et les reporter sous une autre forme

Selector	Sélectionner les éléments d'une collection qui ont des propriétés spécifiques
Mapping	Utiliser une fonction dans un nouveau domaine avec un nouveau rang
Recursion	Répliquer un motif de manière récursive
Goal Seeker	Ajuster les valeurs d'entrée d'une fonction jusqu'à obtenir les valeurs de sortie fixées comme objectif

À la suite d'Alexander, de nombreux auteurs ont exprimés des patterns, chacun à leur manière. Dans le cas de Woodbury, le format des patterns de modélisation paramétrique est le suivant : D'abord, un titre court et explicite décrit le pattern et est suivi par un diagramme représentant le pattern de manière graphique. Ensuite, une phrase décrit l'objectif du pattern tandis qu'un scénario du problème et son contexte explique quand l'utiliser. Les paragraphes suivants expliquent pourquoi utiliser le pattern et comment l'employer pour résoudre un problème donné. Enfin le pattern comprend des « Samples » qui sont des exemples d'application du pattern et un dernier paragraphe énumère les patterns auxquels il est connecté.

Un exemple concret de pattern se trouve à l'annexe 2. Il s'agit du quatrième pattern établi par Woodbury : Increment.

Ces patterns de Woodbury, ont quatre caractéristiques : ils sont explicites, partiales, concentrés sur un unique problème et abstraits. Cette dernière caractéristique d'abstraction est importante, les patterns sont des génériques à adapter à chaque cas d'utilisation. Tandis que les « Samples » sont des exemples de mise en œuvre concrète des patterns.

Enfin, à la manière d'Alexander, Woodbury met en évidence l'importance de « diviser » le modèle. Cette caractéristique fait référence à la seconde compétence mise en évidence par ses recherches et nécessaire pour la réalisation d'un modèle paramétrique : Divide and Conquer. La mise en œuvre des patterns permettra de structurer le modèle en le divisant et inversement.

4. LA MAPMAP

Les trois théories de conception qui viennent d'être présentées vont, à présent, permettre d'établir une Méthodologie d'aide à la Programmation en Modélisation Architecturale Paramétrique (MAPMAP). Ce chapitre est destiné à la présentation de cette méthode à travers le rappel de son objectif, l'exposition de ses origines et la description des principes qui la compose.

Plan du chapitre :

4.1. OBJECTIF DE LA MAPMAP.....	41
4.2. ORIGINES DE LA MAPMAP.....	42
4.3. PRINCIPES DE LA MAPMAP.....	44
4.2.1. <i>Stratégie de Division (ou de responsabilité unique)</i>	44
4.2.2. <i>Réflexe de Documentation</i>	47
4.2.3. <i>Principe d'Encapsulation (fonction macro)</i>	49
4.2.4. <i>Dé-foisonnement des Liaisons</i>	50
4.2.5. <i>Description Générique</i>	52

4.1. Objectif de la MAPMAP

La conception paramétrique à travers une interface visuelle telle que Grasshopper, peut vite entrainer la réalisation d'un modèle complexe dont la lecture est difficile pour le concepteur du modèle ainsi que pour tout autre utilisateur.

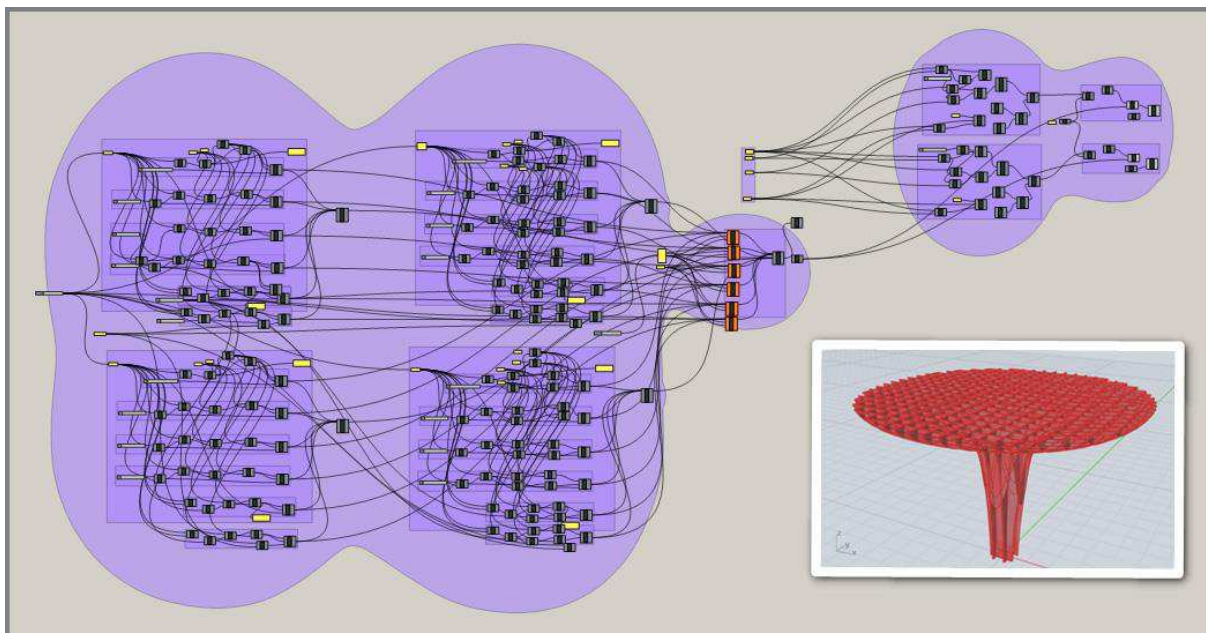


Figure 19 – Exemple de modèle paramétrique construit de manière instinctive.

L'objectif de notre travail est de proposer une méthodologie spécifique basée sur l'utilisation de Grasshopper et utile à tout type d'utilisateurs.

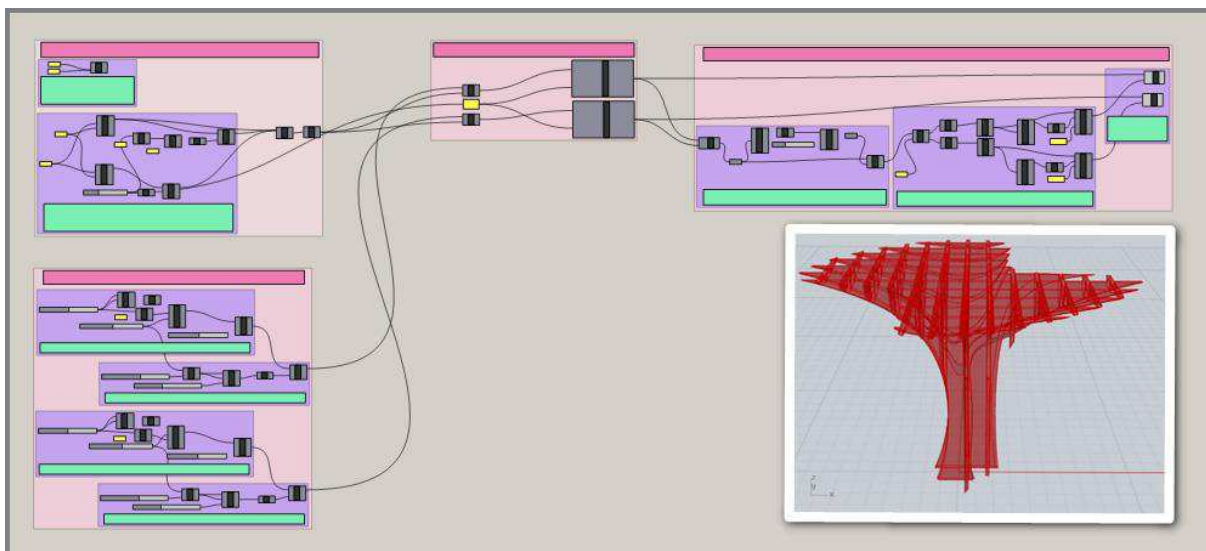


Figure 20 – Exemple de modèle paramétrique construit en appliquant la MAPMAP

4.2. Origines de la MAPMAP

Notre méthode sera établie sur base des trois méthodes qui ont été présentées précédemment, soit : la méthode de conception architecturale d'Alexander, les principes de la méthode de programmation Orientée-Objet et la méthode des patterns de Woodbury.

En ce qui concerne les théories d'Alexander, nous nous concentrerons principalement sur la méthode de décomposition d'un problème en sous-problèmes relativement indépendants. En effet, la seconde partie de la théorie destinée au langage des patterns a déjà été adaptée par Woodbury au domaine de la modélisation paramétrique. Cependant, bien que ce dernier ait fait allusion à une stratégie de 'Divide-and-Conquer' dans ses observations, il n'en suggère pas une application spécifique dans sa méthode des patterns. Cette théorie génère la création d'un premier principe concernant la division du modèle, ce principe étant également renforcé par le concept de modularité de la programmation Orientée-Objet.

Au niveau de la méthode Orientée-Objet, c'est également ce concept de division, de modularité qui domine les principes de programmation. Celui-ci génère notamment les idées de couplage des modules et de cohésion au sein des modules. Ces notions seront renforcées par les observations de Woodbury qui mettent en évidence la nécessité de 'conceiving data flow', c'est-à-dire gérer le flux des données au sein d'un modèle. La réponse à cette nécessité se traduira par la création d'un principe de gestion des liaisons qui sont elles-mêmes la représentation de la transmission des données au sein du modèle.

Le langage Orienté-Objet propose également un principe d'encapsulation qui correspond à l'une des observations de Woodbury qui est la compétence 'Make Modules' (à ne pas confondre avec le principe de modularité correspondant au 'Divide-and-Conquer'). Cette notion sera également traduite à travers l'un des principes de la MAPMAP.

La dernière particularité qui nous intéressera dans la méthode Orientée-Objet est le principe d'abstraction. Woodbury observe que 'thinking with abstraction' est une nécessité dans le cadre de la création d'un modèle paramétrique. Or Alexander émet également ce principe de manière implicite lorsqu'il expose sa théorie car, d'après lui, l'exercice de conception consiste à définir une forme qui réponde aux exigences de son contexte, l'un des problèmes résidant dans la définition de la limite entre la forme et le contexte. La modélisation paramétrique a pour objectif d'augmenter l'abstraction du modèle vis-à-vis de son contexte si bien qu'il convient de le traduire à travers un principe de généralité, d'abstraction, du modèle.

Finalement, en plus des éléments remarquables ci-dessus, une dernière observation de Woodbury 'Naming' sera traduite à travers un principe de documentation au sein de la MAPMAP. Bien qu'elle ne soit pas présente dans les méthodes d'Alexander ou Orientée-Objet, cette

compétence aidant à la réalisation d'un modèle paramétrique a déjà été mise en œuvre dans la méthode de Woodbury à travers son premier pattern 'Clear Names'. Cependant, ce principe est repris au sein de la MAPMAP dans l'objectif de l'étendre aux divers éléments produits par les autres principes de la méthode (division, encapsulation, généricité et gestion des liaisons).

Finalement, la MAPMAP se compose de cinq principes : la Stratégie de Division, le Réflexe de Documentation, le Principe d'Encapsulation, le Dé-foisonnement des Liaisons et la Description Générique.

Remarquons que de nombreuses observations de Woodbury sont à l'origine des principes de la MAPMAP. En effet, dans son ouvrage, Woodbury présentent ces éléments comme étant des compétences aidant les concepteurs à réaliser des modèles paramétriques exploitables. Cependant, sa méthode des patterns ne présente pas de proposition aidant à la mise en œuvre de ces stratégies, hormis pour le principe de 'Naming' traduit dans le pattern 'Clear Names'. C'est pourquoi, la MAPMAP s'alimente des observations effectuées par Woodbury pour apporter, à certaines de ces stratégies, des solutions spécifiques.

4.3. Principes de la MAPMAP

La MAPMAP propose une stratégie pré-opérationnelle, appelée la stratégie de Division ; ainsi que de quatre principes formels que nous nommons : le réflexe de documentation, le principe d'encapsulation, le dé-foisonnement des liaisons et, finalement, la description générique.

Sur inspiration des formats des patterns d'Alexander et Woodbury, ces stratégie et principes vont être présentés en suivant un format identique. Ce format que nous adoptons est le suivant :

- a. Le titre du principe
- b. L'objectif du principe
- c. Les origines du principe
- d. Le contenu du principe
- e. Le moment auquel le principe est mis en œuvre
- f. La manière dont le principe est mis en pratique
- g. L'illustration de la mise en œuvre du principe
- h. (Élément complémentaire)

Les illustrations de la mise en œuvre des principes (f) seront fournies sur base d'un même modèle. Il s'agit d'une esquisse de la volumétrie du 'Gherkin' de Norman Foster, ce modèle constitue le premier exercice de modélisation paramétrique réalisé au cours de Modélisation Architecturale Numérique dispensé à l'Université de Liège dans le cadre de la formation en ingénieur civil architecte.

Les principes vont être présentés suivant l'ordre dans lequel ils viennent d'être énumérés, cet ordre correspondant à une importance décroissante des principes.

4.3.1. Stratégie de Division (ou de responsabilité unique)

Rôle : L'objectif de la stratégie de division est de hiérarchiser le modèle en sous-systèmes relativement indépendants à responsabilité unique au sein du modèle global.

Origines : Cette stratégie pré-opérationnelle est la clé de la méthode et elle a été établie sur base des trois méthodes pré-exposées. Tout d'abord, la stratégie de division est similaire au principe de décomposition d'un problème en sous-problème d'Alexander dans le cadre de la conception architecturale. Ensuite, cette stratégie se retrouve également dans la méthode Orientée-Objet : d'une part, à travers le concept de modularité où le programme est divisé en modules relativement indépendants et, d'autre part, dans le principe de responsabilité unique (Single Responsibility Principle) qui dit que chaque classe a une seule vocation, une seule raison d'être. Enfin, lorsque Woodbury a établi la méthode des patterns pour la modélisation

paramétrique, il a remarqué que, passé un certain degré de complexité, une compétence nécessaire au concepteur est la division du problème en sous-problèmes. À la manière d'Alexander, sa méthode des patterns est une réponse à la stratégie de « divide and conquer ». Au sein des différentes méthodes de conception, la division d'un problème en sous-problème est le principe au centre de la réflexion organisationnelle.

Contenu : Cette stratégie a pour objet de simplifier la phase de modélisation pour le concepteur en divisant le modèle en sous-systèmes plus simples à modéliser que nous appellerons ici 'modules'. Cependant, il convient de différencier ce concept de 'module' avec la stratégie observée par Woodbury et appelée 'make modules' qui consiste en un principe d'encapsulation qui a pour but de cacher des morceaux du modèle pour le rendre plus lisible tandis que nous proposons de créer des modules en mettant en évidence des parties du modèle de sorte à traduire la structure de ce dernier. La stratégie a pour vocation de simplifier la compréhension d'un modèle par un utilisateur externe en traduisant la hiérarchisation dans la représentation symbolique du modèle.

La conception d'un modèle paramétrique de moindre complexité peut toujours être divisée en plusieurs modules ayant chacun un objectif particulier, soit une responsabilité unique. De plus, il est important que le nombre de liens entre ces modules soit limité puisque les liens représentent les dépendances entre les objets et donc entre les sous-systèmes (cette caractéristique des modules se retrouve particulièrement dans le concept de couplage de la méthode Orientée-Objet).

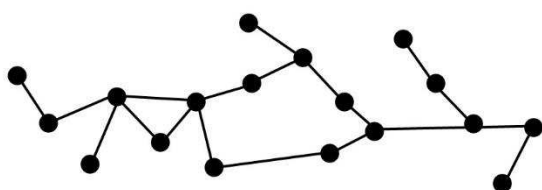


Figure 21 - Graphe d'un modèle composé d'objets et de liens

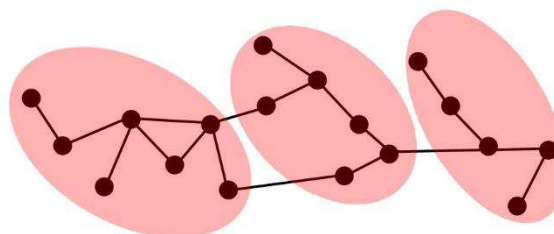


Figure 22 - Graphe d'un modèle divisé en modules

Ensuite, au sein même des modules, une nouvelle division en sous-systèmes, ou modules, d'ordre inférieur peut parfois être réalisée. Le concepteur est donc amené à hiérarchiser son travail en regroupant les fonctions qui ont une même responsabilité, un même objectif. La responsabilité unique d'une partie du modèle peut alors être plus ou moins précise selon le niveau de hiérarchisation dans lequel on se trouve.

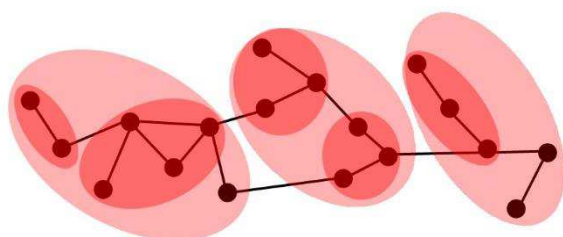


Figure 23 – Graphe d'un modèle divisé en modules primaires et secondaires

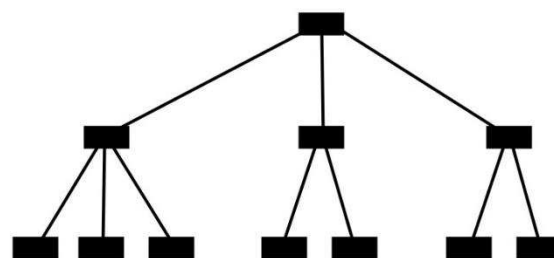


Figure 24 – Hiérarchie des modules primaires et secondaires d'un modèle

Quand : Le principe est à mettre en œuvre au tout début du processus de conception et ensuite tout au long de sa construction. En effet, il s'agit d'une stratégie pré-opérationnelle qui permet de structurer les étapes du travail de modélisation avant de commencer la construction du modèle mais qui devra également être appliquée durant la construction du modèle pour faire transparaître la structure des modules dans la représentation symbolique.

Comment : La première étape de la stratégie de division consiste à avoir une vision d'ensemble du problème de modélisation et de déterminer au sein de celui-ci les différentes étapes de construction du modèle qui formeront la structure de modules primaires. Ensuite, au fur et à mesure de la construction du modèle, il convient de déterminer les éventuels modules d'ordre secondaire au sein de chaque module primaire. Cette subdivision doit se faire en regroupant les objets qui ont une même vocation.

Dans la pratique, ce principe de division peut être mis en place dans Grasshopper par la création de groupes. Formellement, en changeant la couleur des groupes selon le degré de hiérarchisation, on traduit la structure de construction du modèle au sein même de la représentation symbolique.

Illustration : Si l'on reprend le modèle du Gherkin réalisé par les étudiants au cours de MAN, il peut être divisé en trois modules primaires : la formation de la volumétrie, la création d'un membre du châssis, et la réalisation de l'entièreté du châssis. Au sein de ces modules primaires, on retrouve des modules secondaires qui ont été déterminés sur base de leur vocation spécifique au sein du modèle.

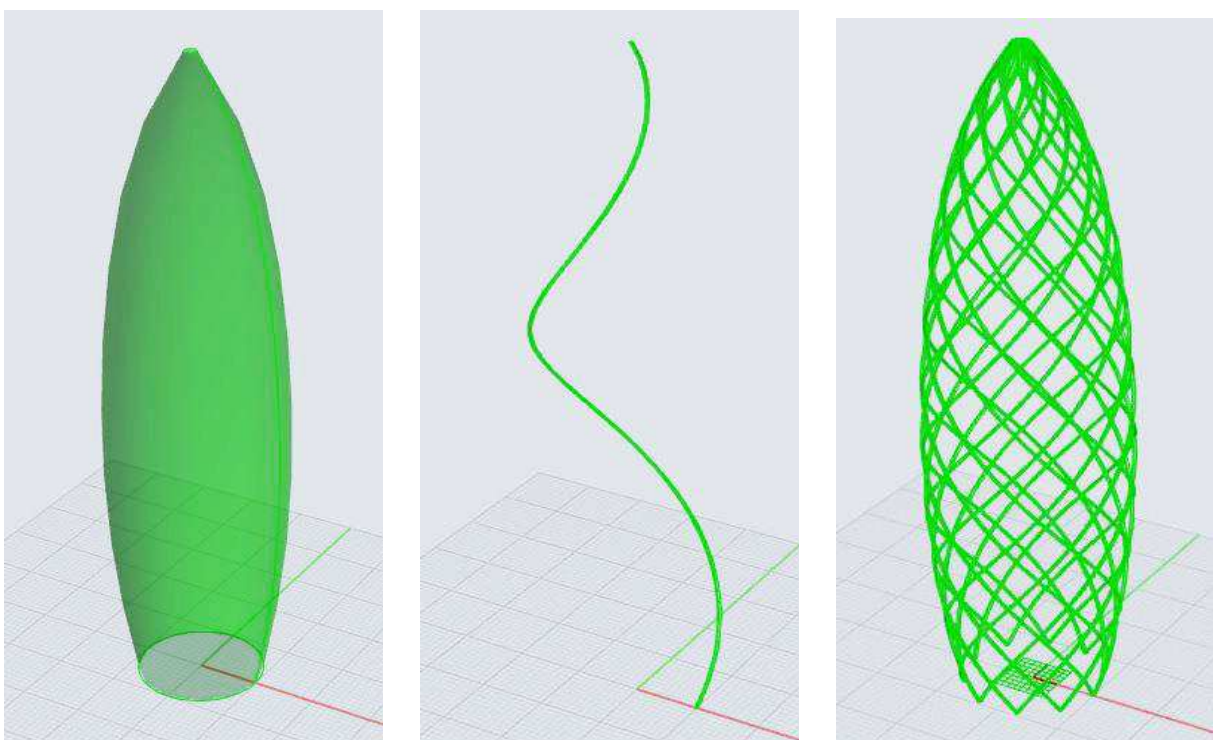


Figure 25 – Décomposition du modèle du Gherkin en trois étapes de construction

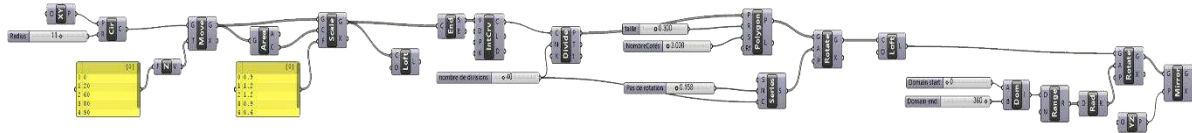


Figure 26 – Exemple de modèle du Gherkin tel que réalisé au cours de MAN

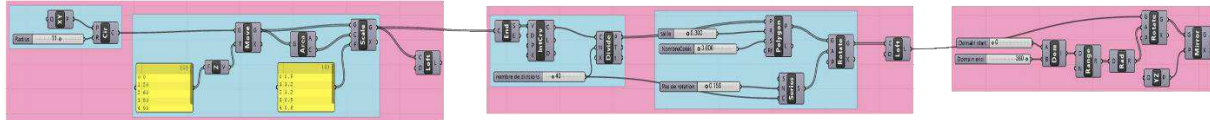


Figure 27 – Exemple de modèle du Gherkin structuré en modules

Complément : Woodbury a créé les patterns de modélisation paramétrique de sorte à ce qu'ils aient un rôle unique au sein d'un modèle paramétrique. Ces patterns se marient parfaitement à une stratégie de division d'un modèle en modules hiérarchisés. Ainsi il convient de compléter la stratégie de division par certains patterns établis par Woodbury. Ces patterns permettront d'une part d'aider le concepteur à structurer son modèle et, d'autre part, une fois la structuration effectuée, ils permettront l'utilisation de 'samples' qui sont des applications spécifiques de chaque pattern.

Cependant, les patterns de Woodbury ont été établis sur base du logiciel GenerativeComponents. Deux d'entre eux [Recursion & Goal Seeker] sont difficilement applicables dans le cadre du logiciel Grasshopper en raison de leur structure itérative. En effet, le logiciel Grasshopper a un fonctionnement séquentiel, c'est-à-dire qu'il exécute les opérations de manière successive mais il ne peut pas réaliser un travail itératif, c'est-à-dire le calcul d'un programme qui fonctionnerait sous-forme de boucle et donc où le résultat d'une opération servirait directement ou indirectement d'entrée à cette opération.

Enfin le premier pattern CLEAR NAMES fait l'objet du principe de documentation, il n'est donc pas repris dans la MAPMAP en tant que pattern complémentaire à la Stratégie de Division. En conclusion, dix patterns ont été repris de la méthode de Woodbury pour alimenter la stratégie de division, il s'agit des suivants : CONTROLLER, JIG, INCREMENT, POINT COLLECTION, PLACE HOLDER, PROJECTION, REACTOR, REPORTER, SELECTOR, MAPPING. (Revoir les objectifs de ces patterns définis par Woodbury décrits dans le paragraphe 3.3.2., page 37.) Une illustration des dix patterns a été réalisée dans le cadre de la première expérience et se trouve à l'annexe 5.

4.3.2. Réflexe de Documentation

Rôle : Le réflexe de documentation consiste à donner des noms et descriptions représentatifs aux différentes entités d'un modèle paramétrique.

Origines : Lors de l'exposition de sa théorie de décomposition d'un problème, Alexander présente la tendance à regrouper les exigences du problème sous forme de classes auxquelles il donne un nom. Woodbury, lors de ses observations, a mis en évidence cette même

compétence de nomination que le concepteur emploie en modélisation paramétrique. Au vu de cette caractéristique, Woodbury propose comme premier pattern le 'CLEAR NAMES' qui consiste à nommer de manière claire et pertinente les objets.

Contenu : Ce principe de documentation consiste donc à renommer de manière précise et explicite les paramètres utilisés dans le modèle à la manière dont le propose le premier pattern de Woodbury. Cependant, il a également pour vocation de donner un titre aux modules déterminés lors de la mise en œuvre de la stratégie de division et de donner une description de leur responsabilité propre au sein du modèle.

Quand : Lors de la structuration du modèle en modules primaires et préalablement à la construction du modèle, il convient de commencer à titrer ces modules. Ensuite, le réflexe de documentation est utilisé en permanence, tout au long de la conception du modèle.

Comment : Pour mettre en œuvre ce principe, quatre opérations clés sont à effectuer durant la modélisation : titrer les modules, (re)nommer les paramètres, décrire les modules secondaires et référencer les patterns utilisés.

En pratique dans Grasshopper, l'objet Params-Input-Panel peut être utilisé pour insérer des titres ou des descriptions. En outre, la typologie de ces panneaux peut être changée de sorte à différencier les panneaux de modules de différents degrés de hiérarchisation et/ou à vocation différentes (titre, description,...).

Enfin, comme le recommande Woodbury pour le pattern CLEAR NAMES, il est important d'employer des noms/titres ayant une signification et d'éviter les noms évasifs tels que: Point_02, Courbe_A, etc.

Illustration : Après la division du modèle du Gherkin en modules lors de l'application de la stratégie de division, les modules primaires (en rose), qui correspondent aux trois étapes de modélisation principale, ont été titré et les vocations des modules secondaires (en bleu), qui sont des subdivisions basées sur le rôle de chaque objet, ont été décrites. Ces titrages et descriptions se sont fait à l'aide de l'outil Params-Input-Panel dont la couleur varie selon le degré de hiérarchisation des modules. Enfin, la pertinence des noms des paramètres d'entrée du modèle a été vérifiée et au besoin le nom a été changé.

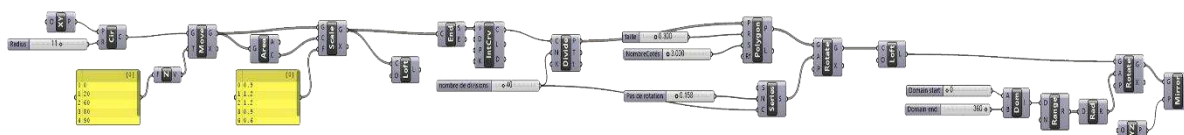


Figure 28 – Exemple de modèle du Gherkin réalisé au cours de MAN

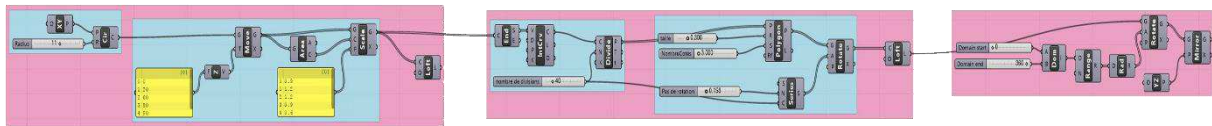


Figure 29 – Exemple de modèle du Gherkin structuré en modules

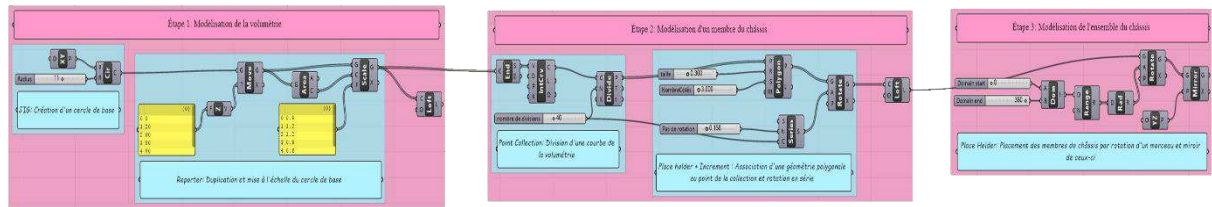


Figure 30 – Exemple de modèle du Gherkin structuré et documenté

4.3.3. Principe d'Encapsulation (fonction macro)

Rôle : Le principe d'encapsulation a pour vocation de simplifier la représentation symbolique d'un modèle paramétrique par la création de « boîtes noires » permettant de cacher des parties du modèle.

Origines : Le principe d'encapsulation provient directement du concept d'encapsulation de la méthode Orientée-Objet. En programmation informatique, ce concept a été traduit par le principe de ségrégation de l'interface, qui consiste à encapsuler certaines parties d'un programme pour ne laisser apparentes que les commandes nécessaires à son exploitation.

Contenu : L'objectif est de simplifier l'interface entre l'ordinateur et l'utilisateur pour faciliter la lecture d'un modèle. Là où la stratégie de division regroupe des objets au sein de modules, l'encapsulation permet de cacher un morceau du modèle au sein d'un nouvel objet qui pourra être ré-exploité dans le modèle comme tout autre objet directement mis à disposition par le modelleur.

Quand : Ce principe est à utiliser durant la modélisation lors de cas particuliers, en l'occurrence lorsqu'un ensemble d'objets, ayant une même vocation, est répété au sein du modèle.

Comment : Dans la pratique, le logiciel Grasshopper permet la création de 'Clusters' qui sont des entités pouvant contenir un ensemble d'objets. Ces entités ont des entrées et des sorties ce qui permet d'en faire des objets du modèle à leur tour.

Au vu du principe de documentation préalablement expliqué, il est important de nommer les clusters, leurs entrées et leurs sorties. De plus, lors de la création d'un cluster, il est intéressant d'ajouter une description de son rôle ou de son contenu pour éviter que sa mise en œuvre empêche de comprendre aisément le fonctionnement du modèle.

Illustration : Dans le cas du modèle du Gherkin, le modèle ne présente pas les caractéristiques d'utilisation d'un cluster. Nous nous projeterons donc dans un contexte particulier imaginé pour l'illustration de ce principe : imaginons que nous souhaitons dupliquer la volumétrie initiale tout en changeant les caractéristiques contextuelles de chaque duplication, nous pouvons créer un cluster de la première étape de modélisation et le dupliquer au sein du modèle comme on dupliquerait un quelconque objet.

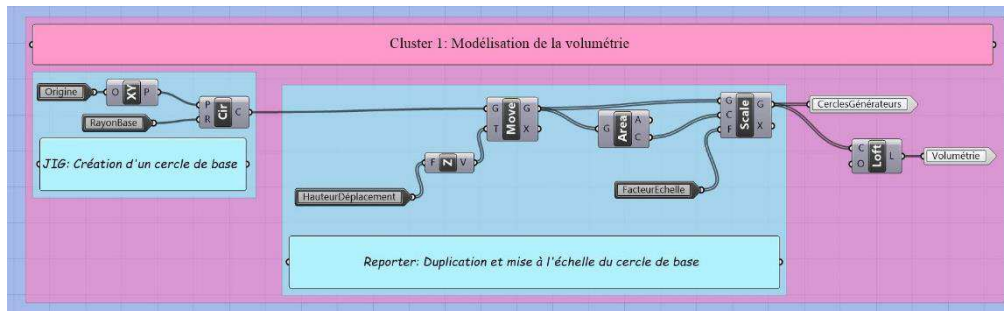


Figure 31 - Cluster contenant la modélisation de la volumétrie dans le cadre du modèle du Gherkin

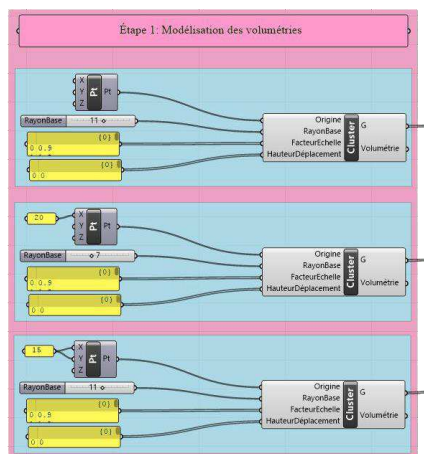


Figure 32 - Duplication du cluster contenant la modélisation de la volumétrie dans le cadre du modèle du Gherkin

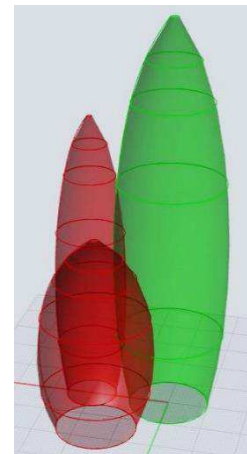


Figure 33 – Représentation explicite de la duplication du cluster de la volumétrie

4.3.4. Dé-fioisonnement des Liaisons

Rôle : Ce principe a pour objectif de rendre un modèle plus lisible en organisant la manière dont les liens relient des objets de différents modules.

Origines : Dans ses observations, Woodbury met en évidence la nécessité de pouvoir contrôler le flux des données au sein d'un modèle à travers la compétence de 'conceiving data flow'. En outre, dans la méthode Orientée-Objet, la notion de liaison des entités se fait également à travers les concepts de cohésion et de couplage. L'objectif étant d'avoir une grande cohésion au sein de chaque module et un faible couplage des modules entre eux. De plus, le principe d'équivalence sortie/communication met également en évidence l'importance des liens reliant des modules : les données qui sortent d'un module sont les données qui peuvent être communiquées aux autres modules via des liens, c'est-à-dire via le couplage des modules.

Contenu : Comme nous l'avons vu dans la méthode Orientée-Objet, pour augmenter la stabilité d'un modèle, il faut le rendre plus abstrait vis-à-vis de son contexte et cela peut se traduire en diminuant le nombre de paramètres d'entrée. Cependant, en diminuant le nombre de paramètres d'entrée, le concepteur risque d'augmenter les liens de dépendances depuis ces quelques paramètres si bien que de nombreuses liaisons vont partir d'une même source vers différentes parties du modèle ce qui augmentera le couplage du modèle et diminuera la lisibilité de celui-ci.

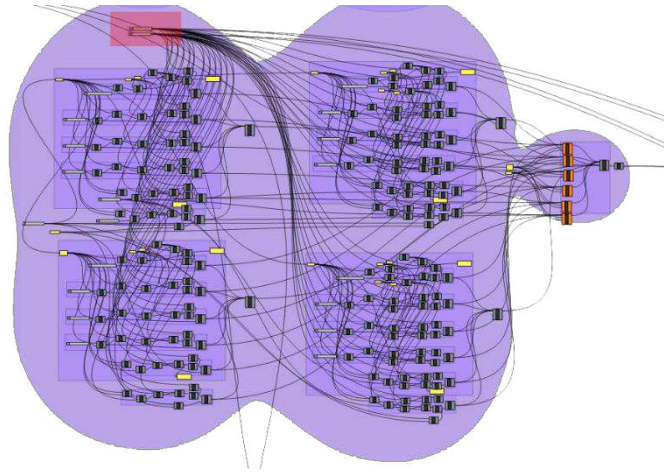


Figure 34 - Exemples de modélisation avec un foisonnement de liens trop important

Le concepteur devra donc commencer par trouver un bon équilibre entre un minimum de couplage et un minimum de paramètres utilisés. Dans le cas où certains paramètres seraient utilisés de nombreuses fois dans différents modules, nous proposons de mettre en place un dé-foisonnement des liaisons entre les modules. Pour ce faire un avatar du paramètre est créé au sein de chaque module nécessitant ce paramètre d'entrée. Ici, nous proposons d'employer le terme avatar pour signifier la création d'un objet dont le rôle est de contenir les valeurs du paramètre qu'il représente.

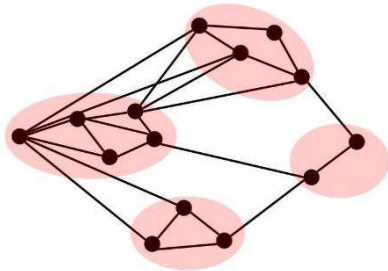


Figure 35 – Graphe d'un modèle avec un important couplage

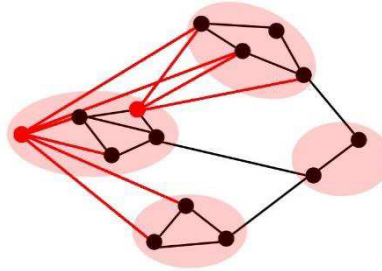


Figure 36 - Mise en évidence du foisonnement de liens provenant de deux paramètres

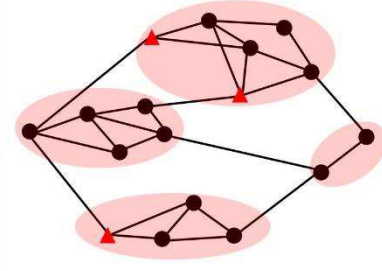


Figure 37 - Dé-foisonnement des liens superflus par la création d'avatars (triangles)

Quand : L'application du principe de dé-foisonnement se fait durant la construction du modèle. À chaque création d'un module primaire, le concepteur doit vérifier les entrées nécessaires au sein de ce module et doit juger si la création d'un avatar est pertinente selon le cas de figure.

Comment : Le dé-foisonnement des liaisons s'effectue par la création d'avatars au sein des différents modules. Ces avatars sont reliés à la valeur source qu'ils représentent et en pratique, dans Grasshopper, un avatar peut être créé à l'aide des objets compris dans l'onglet Params-Geometry, l'objet choisi doit avoir le même type de géométrie que la valeur source (point, courbe, surface, valeur,...).

Illustration : Ce principe ne peut pas être illustré par le modèle, trop simple, du Gherkin. Nous allons donc l'expliquer à travers un autre modèle réalisé par les étudiants dans le cadre de la formation à la modélisation paramétrique du cours de MAN : il s'agit d'un modèle destiné au traitement d'une enveloppe par l'application de cellules Voronoï.

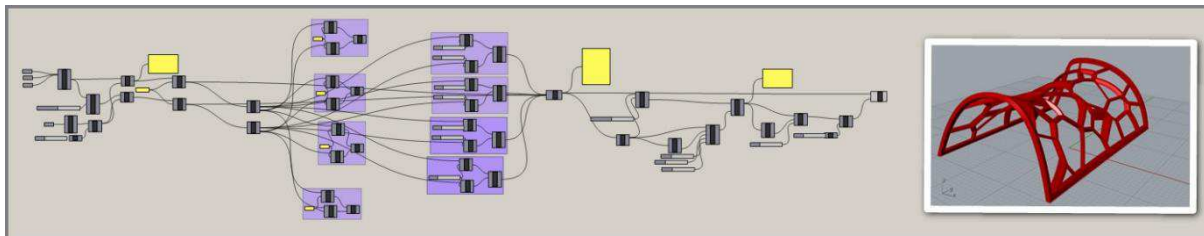


Figure 38 – Exemple de modèle de traitement d'une surface par un Voronoï réalisé au cours de MAN

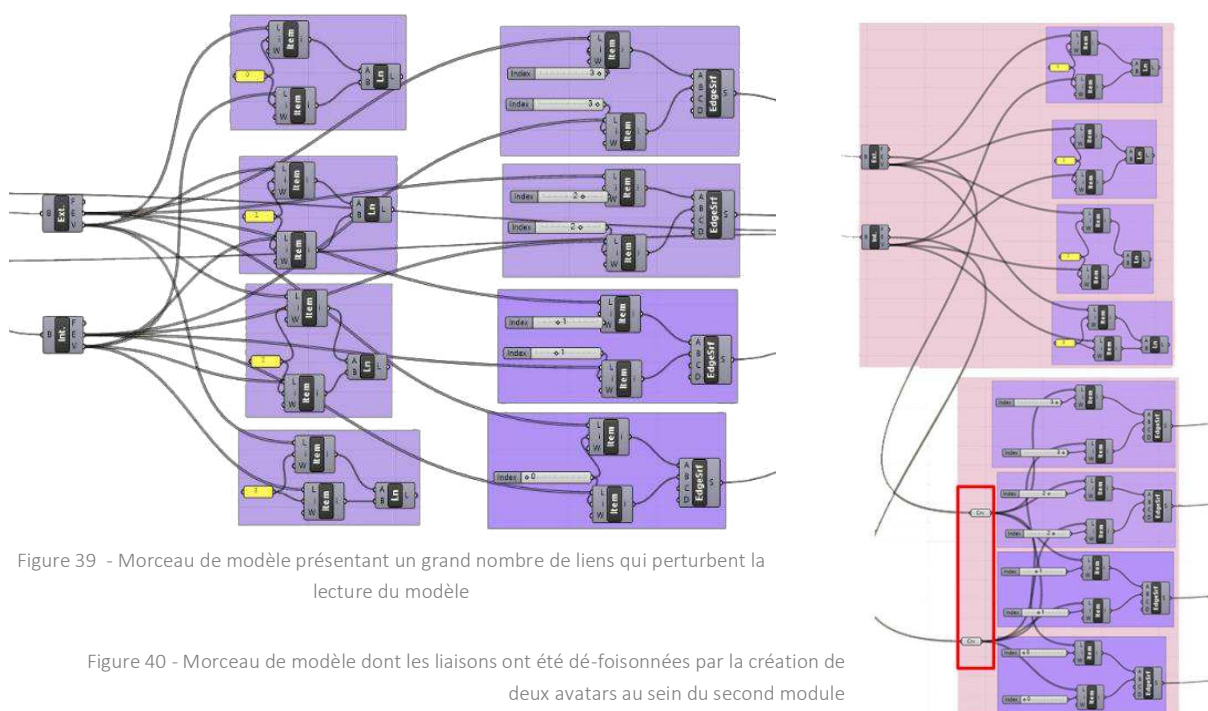


Figure 39 - Morceau de modèle présentant un grand nombre de liens qui perturbent la lecture du modèle

Figure 40 - Morceau de modèle dont les liaisons ont été dé-foisonnées par la création de deux avatars au sein du second module

4.3.5. Description Générique

Rôle : La description générique consiste à prévoir l'évolution du modèle en le rendant plus flexible par substitution de certains objets trop précis. Elle engendre également une meilleure compréhension mathématique et algorithmique du modèle.

Origines : Ce principe trouve ses origines dans la méthode Orientée-Objet : d'une part, à travers les concepts de généricité et d'abstraction et, d'autre part, à travers le principe ouvert-fermé, le principe de substitution de Liskov et le principe d'abstraction stable. En outre, la généricité et l'abstraction sont également des facteurs clés de la mise en place des patterns dans les méthodes d'Alexander et Woodbury. Finalement, les observations de Woodbury énoncent les

principes de « thinking mathematically » et « thinking algorithmically » qui sont des compétences nécessaires au développement d'une plus grande généricité au sein d'un modèle.

Contenu : Bien que la MAPMAP ait pour vocation d'améliorer la compréhension et la communication d'un modèle, ce dernier principe a été établi alors qu'il a pour principal objectif de rendre le modèle plus flexible. En effet, il est impossible de dissocier complètement les deux phases de vie d'un modèle paramétrique : sa conception et son exploitation. C'est pourquoi, en vue d'améliorer la communication d'un modèle à un utilisateur externe, il est important d'avoir déjà mis en place un certain degré d'adaptabilité du modèle.

L'objectif étant d'augmenter le degré d'ouvrabilité du modèle par l'abstraction, la solution proposée est de remplacer des objets trop spécifiques par leur version générique. La mise en œuvre de ce principe s'effectue au cas par cas et dépend de l'imagination, de l'expérience et de la connaissance du logiciel par le concepteur. Le rôle de ce principe est de ne pas figer le modèle en y intégrant des objets susceptibles d'évoluer par la suite.

Quand : Ce principe se met en œuvre tout au long de la construction du modèle, lors de l'intégration de chaque nouvel objet le permettant.

Comment : Ce principe consiste à prévoir l'évolution d'un modèle par la substitution de fonctions précises par leurs fonctions génériques. Son application pratique ne dépend pas du modèleur utilisé et se fera au cas par cas d'après les besoins du modèle et les connaissances du concepteur.

Illustration : Dans le modèle du Gherkin réalisé au cours de MAN, lors de la création du châssis, on utilise un polygone pour en paramétrer le nombre de côtés plutôt que d'imposer une géométrie fixe. En effet, le polygone est la fonction générique de toutes les formes géométriques dont les n côtés ont la même longueur et forment n angles de même amplitude dont la somme vaut $(n-2)*180^\circ$. Cette description pourrait encore évoluer de manière plus générique en remplaçant le polygone par une courbe quelconque.

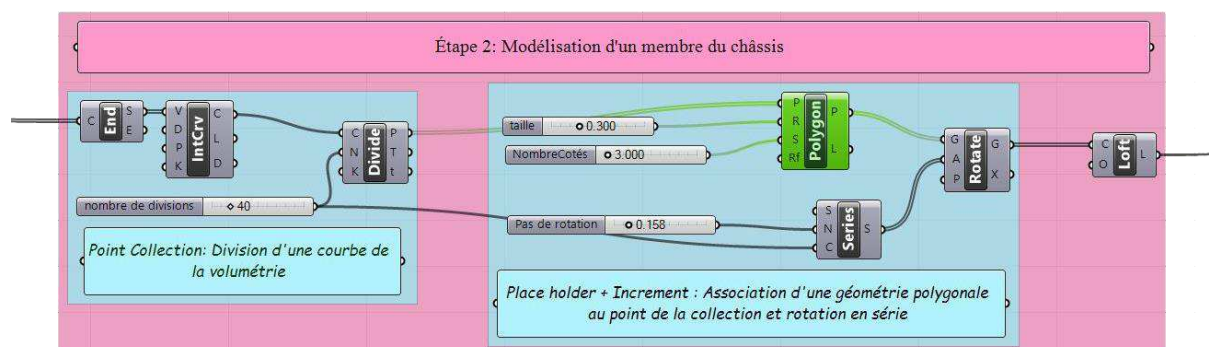


Figure 41 – Exemple d'un morceau de modèle du Gherkin dont l'objectif est la création d'un châssis sur base d'une forme polygonale dont le nombre de côté a été paramétré à 3

5. LES EXPÉRIMENTATIONS

Une fois la MAPMAP établie, il convient de la tester pour pouvoir l'évaluer. Dans cet objectif, deux phases expérimentales ont été réalisées et elles sont décrites au sein de ce chapitre. Pour chacune de ces phases, nous exposerons d'une part pourquoi et comment l'expérience a été construite et, d'autre part, comment son analyse sera réalisée.

Plan de chapitre :

5.1.	EXPÉRIMENTATION DE MISE EN ŒUVRE DE LA MAPMAP DURANT LA CONCEPTION.....	55
	5.1.1. <i>Protocole de l'expérience</i>	55
	5.1.2. <i>Critères d'analyse</i>	65
5.2.	EXPÉRIMENTATION DE COMMUNICATION DE MODÈLES PARAMÉTRIQUES.....	68
	5.2.1. <i>Protocole de l'expérience</i>	68
	5.2.2. <i>Critères d'analyse</i>	72

5.1. Expérimentation de mise en œuvre de la MAPMAP durant la conception

5.1.1. Protocole de l'expérience

Cette première expérimentation est destinée à évaluer le premier objectif de la MAPMAP, soit la mise en œuvre de la méthodologie durant la phase de modélisation pour rendre le modèle plus lisible pour tous. Pour évaluer cette caractéristique, nous étudierons des modèles paramétriques réalisés par les étudiants du cours de MAN 2017, dans le cadre d'un exercice de modélisation sur Rhinoceros-Grasshopper.

Sujets

Les sujets de l'expérience sont des étudiants de 3^{ème} bachelier en ingénieur civil architecte à l'université de Liège. Ces étudiants ont tous reçus la même formation en modélisation paramétrique, dispensée durant le cours de MAN. En cette année 2017, elle était composée de trois travaux pratiques dirigés par l'assistant John Schrayen. Les énoncés de ces exercices de modélisation se trouvent en annexe 3 et les modèles réalisés sont les suivants : le Gherkin de N. Foster, un traitement de surface à l'aide de cellules Voronoï et finalement un traitement de surface modulaire à l'aide d'un maillage.

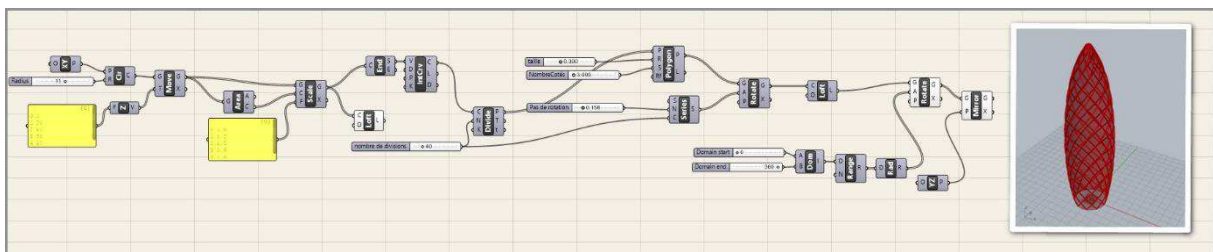


Figure 42 – Exemple de modèle paramétrique 1 réalisé au cours de MAN : le Gherkin de N. Foster

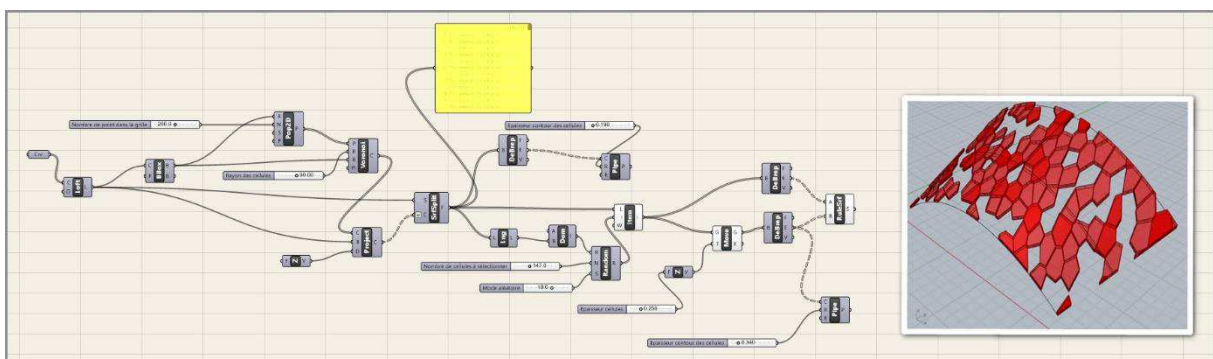


Figure 43 – Exemple de modèle paramétrique 2 réalisé au cours de MAN : traitement de surface par du Voronoï

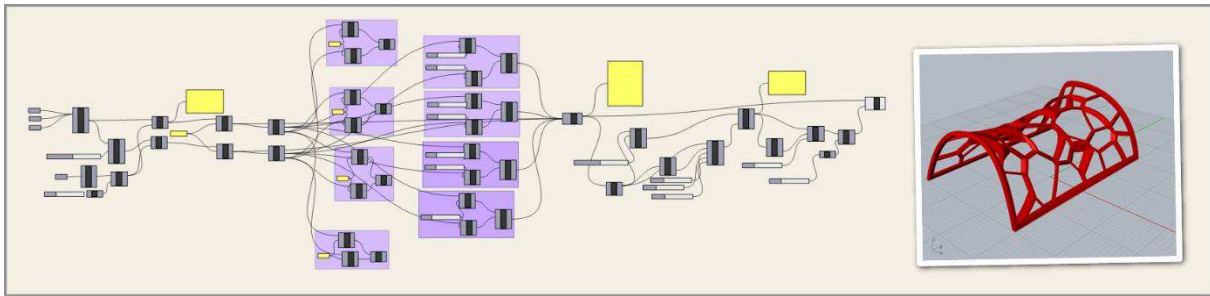


Figure 44 – Exemple de modèle paramétrique 2, seconde version, réalisé au cours de MAN : traitement de surface par du Voronoï (3D)

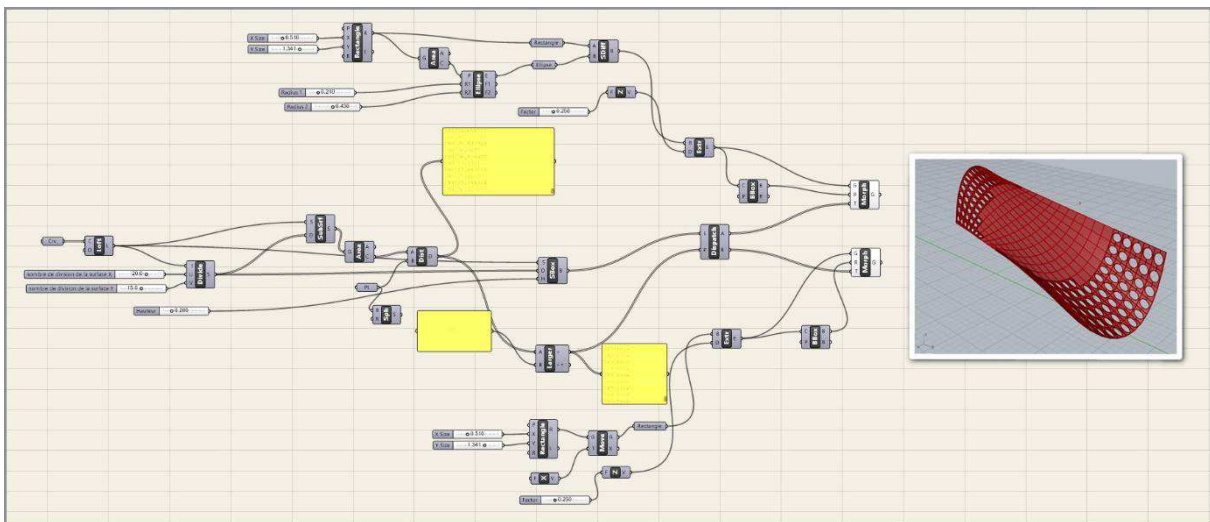


Figure 45 – Exemple de modèle paramétrique 3 réalisé au cours de MAN : traitement de surface modulaire par maillage

Pour pouvoir évaluer la MAPMAP de manière qualitative, il a été décidé de réaliser une analyse comparative. Pour ce faire, la moitié des étudiants soumis à l'expérience ont réalisé l'exercice de manière instinctive sur base de la formation classique dispensée au cours de MAN (groupe 2), tandis que l'autre moitié des étudiants a reçu une formation supplémentaire destinée à leur présenter la MAPMAP (groupe 1). La capsule de cours constituant cette formation complémentaire se trouve en annexe 5. Ensuite, pour la réalisation de l'exercice de modélisation, les étudiants ont été répartis par binômes de travail et, au sein de chacun des deux groupes (avec et sans la MAPMAP), un binôme a été sélectionné comme binôme 'vedette', ce qui signifie que les réalisations de l'expérience par ces deux binômes ont été enregistrées.

Présentation de l'expérience

L'expérience est composée de quatre phases, pour le groupe 1 qui est le groupe 'formé', ou de trois phases, pour le groupe 2 qui est le groupe 'instinctif'. La première étape de l'expérience est la présentation de la MAPMAP aux étudiants du groupe 1. Cette présentation qui dure approximativement 30minutes, se trouve à l'annexe 5.



Figure 46 - Deux premières slides du cours d'introduction à la MAPMAP, présenté au groupe 2

Ensuite, l'énoncé de l'exercice identique pour tous est présenté aux deux groupes séparément. Il s'agit d'un exercice de modélisation inspiré du METROPOL PARASOL de l'architecte Jürgen Hermann Mayer et situé à Séville.



Figure 47 - METROPOL PARASOL de Séville (<http://inhabitat.com>)

Le modèle que les étudiants doivent réaliser a pour but de se concrétiser par une matérialisation via la découpe laser. Ainsi, l'énoncé de l'exercice a été conçu, en accord avec les encadrants du cours, soit le Professeur Pierre Leclercq et l'assistant John Schrayen. Ainsi, le modèle que les étudiants concevront, est composé d'un parasol unique dont la structure est constituée de montants verticaux orthogonaux. Une attention particulière sera apportée aux intersections de ces panneaux en vue du prototypage du modèle.

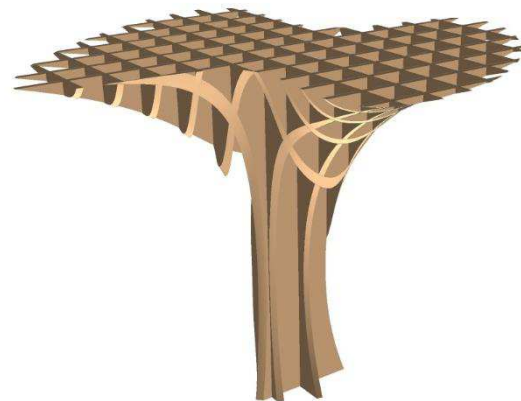


Figure 48 - Modèle à réaliser durant l'expérience.

La troisième étape de l'expérience est la réalisation de l'exercice. Pour ce faire les étudiants disposent de différents documents :

Les étudiants du groupe 2 ont à leur disposition l'énoncé de l'exercice, qui se trouve en annexe 6, et une maquette physique du modèle. Ils peuvent également se servir d'internet et des travaux pratiques réalisés au cours de MAN. Un encadrant est à leur disposition durant toute l'expérience pour répondre à leurs questions, l'objectif étant que les étudiants puissent construire leur modèle sans souffrir d'un manque d'expérience avec le logiciel.



Figure 49 – Première page de l'énoncé distribué aux étudiants

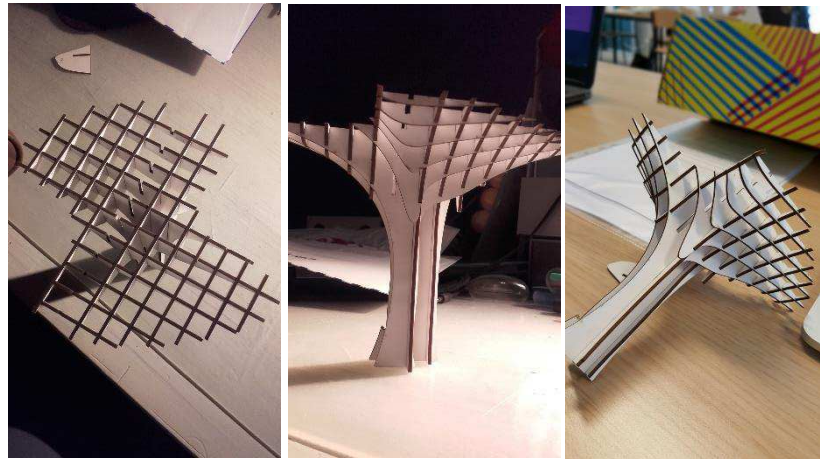


Figure 50 - Modèle physique à disposition des étudiants durant l'expérience

Les étudiants du groupe 1, ont à leur disposition ces mêmes documents et un autre encadrant pour répondre à leurs questions. Les documents relatifs à la MAPMAP et sa mise en œuvre leurs sont également fournis pour construire le modèle : un récapitulatif de la mise en œuvre de la MAPMAP ainsi qu'un tableau récapitulatif des dix patterns de Woodbury repris en complément de la Stratégie de Division de la MAPMAP. Ces documents se trouvent respectivement en annexe 7 et 8.

Enfin, pour la dernière étape de l'expérience, il est demandé à tous les étudiants de compléter un questionnaire. Celui-ci poursuit différents objectifs : pour le groupe 2, il s'agit d'évaluer le niveau de compréhension de l'étudiant vis-à-vis du modèle paramétrique utilisé, soit Grasshopper, ainsi que de prendre connaissance des éventuelles remarques de l'étudiant concernant la réalisation de l'exercice. Pour le groupe 1, les deux premiers objectifs sont identiques à ceux du questionnaire du groupe 2 et ceux-ci sont complétés par deux autres objectifs : évaluer le niveau de compréhension de l'étudiant vis-à-vis de la MAPMAP et prendre connaissance de ses remarques concernant cette méthode. Le contenu des deux questionnaires sera détaillé dans la partie « 4.2.2. Critères d'analyse » et ils sont consultables aux annexes 9 et 10.

Enfin, précisons que les trois ou quatre étapes constituant l'expérience sont réalisées sur une durée de maximum 3heures.

Expérience zéro

Avant de réaliser l'expérience concrète avec les étudiants du cours de MAN, une expérience 'test', appelée expérience zéro, a été réalisée en vue d'améliorer et préciser la mise en place de l'expérience proprement dite. Elle a été réalisée le vendredi 10 mars 2017 dans les studios du LUCID, l'une des unités de recherche de l'ULg.

Cette expérience zéro a été réalisée avec deux étudiants de 2^{ème} master en ingénieur civil architecte de l'université de Liège. Ces étudiants ont suivi la formation du cours de MAN, il y a de ça deux ans, soit durant l'année académique 2014-2015. À l'époque, les travaux pratiques étaient dirigés par monsieur Anis Gallas et la formation à la modélisation paramétrique était constituée de quatre exercices de modélisation. Les trois premiers exercices correspondent aux exercices réalisés au cours de MAN en cette année 2016-2017, qui ont été présentés précédemment aux pages 51 et 52 et dont les énoncés complets se trouvent à l'annexe 3. En outre, le quatrième exercice était la réalisation d'un modèle de pavillon en forme de coquillage qui était destiné à une matérialisation par découpe laser. L'énoncé de cet exercice se trouve à l'annexe 4.

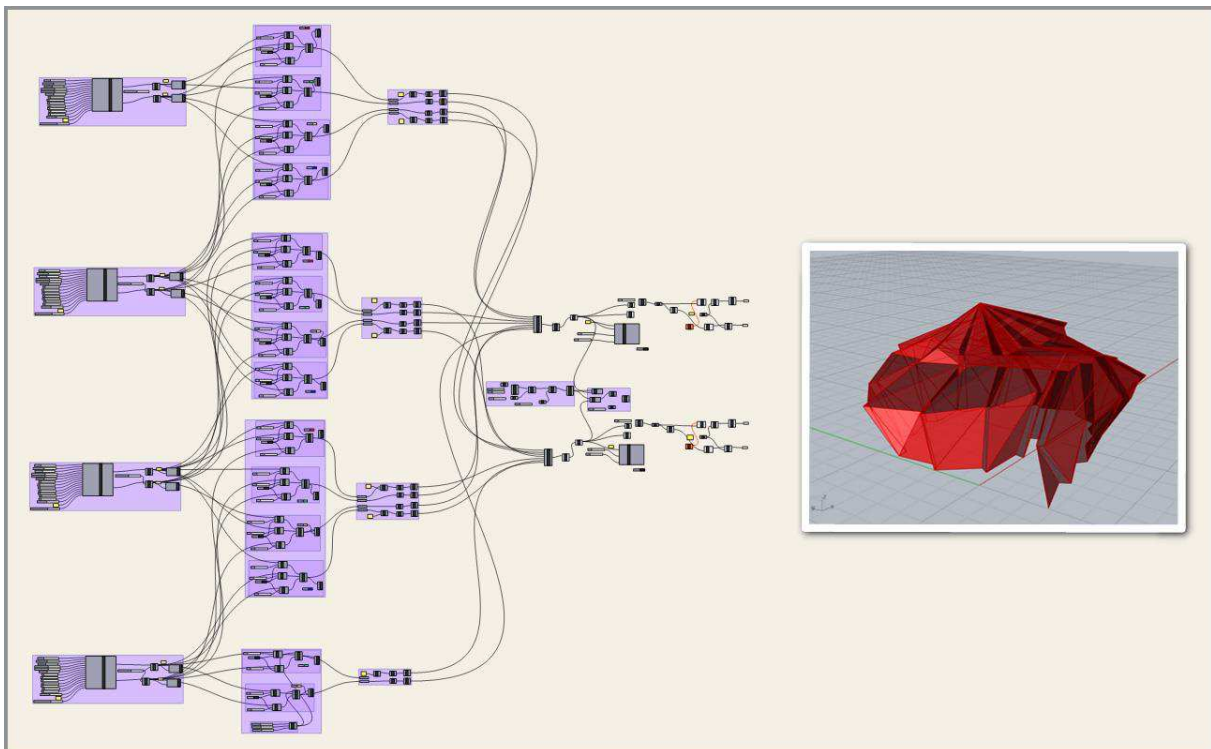


Figure 51 – Exemple de modèle paramétrique 4 réalisé au cours de MAN 2015 : Pavillon en forme de coquillage.

Les deux étudiants de l'expérience zéro devaient présenter un profil semblable aux étudiants du cours de MAN 2017, c'est pourquoi, en plus d'avoir choisi des étudiants qui avaient reçu une formation semblable, des critères supplémentaires ont été ajoutés à la sélection de ces deux sujets : il fallait des étudiants qui aient réutilisé le logiciel Grasshopper dans les deux années qui suivirent leur formation à la modélisation paramétrique du cours de MAN 2015, ce

qui assurait qu'ils se souviennent du fonctionnement du modelleur paramétrique, mais il fallait qu'ils n'aient pas approfondis leur connaissance du logiciel pour rester proche du niveau d'expérience des étudiants de 3^{ème} bachelier.

Pour rentabiliser au maximum l'expérience zéro, les deux étudiants ont été mis dans la situation la plus complexe du point de vue de la mise en œuvre de l'expérience si bien qu'ils occupaient le rôle du binôme vedette du groupe 1 : le binôme mettant en œuvre la MAPMAP et dont la réalisation du modèle était enregistrée.

En ce qui concerne, le déroulement de l'expérience, la formation d'introduction à la MAPMAP est présentée aux étudiants ainsi que l'exercice de modélisation à réaliser comme annoncé précédemment. Ensuite, les sujets sont assis face à un ordinateur de travail, en l'occurrence il s'agit d'un ordinateur PC avec un processeur Intel Core i7 et une carte graphique AMD Radeon HD 8750M et Intel HD Graphics 4600 sur lequel sont installés le logiciel Rhinoceros (version 5) et son extension paramétrique Grasshopper (version 0.9.76.0).



Figure 52 – Binôme de l'expérience zéro construisant le modèle sur Grasshopper à l'aide de la MAPMAP

Durant la réalisation de l'exercice de modélisation, les étudiants sont encadré dans le but de répondre à leurs questions et ils disposent d'un énoncé de l'exercice de modélisation, d'un récapitulatif de la MAPMAP et d'un récapitulatif des dix patterns de Woodbury. Cependant, ils ne disposent pas de la maquette physique du modèle. Celui-ci a été ajouté aux documents par la suite, lors de l'expérience avec les étudiants de 3^{ème} bachelier.

Pour réaliser l'enregistrement de l'expérience, deux dispositifs ont été mis en œuvre. Premièrement, un second écran, lié à l'écran de l'ordinateur de travail, permet à une caméra placée face à ce second écran, d'enregistrer l'expérience d'un point de vue auditif et visuel. Deuxièmement, l'écran de l'ordinateur de travail est enregistré à l'aide du logiciel CamStudio Recorder (version 3.1.0.0).



Figure 53 - Enregistrement caméra d'un écran secondaire

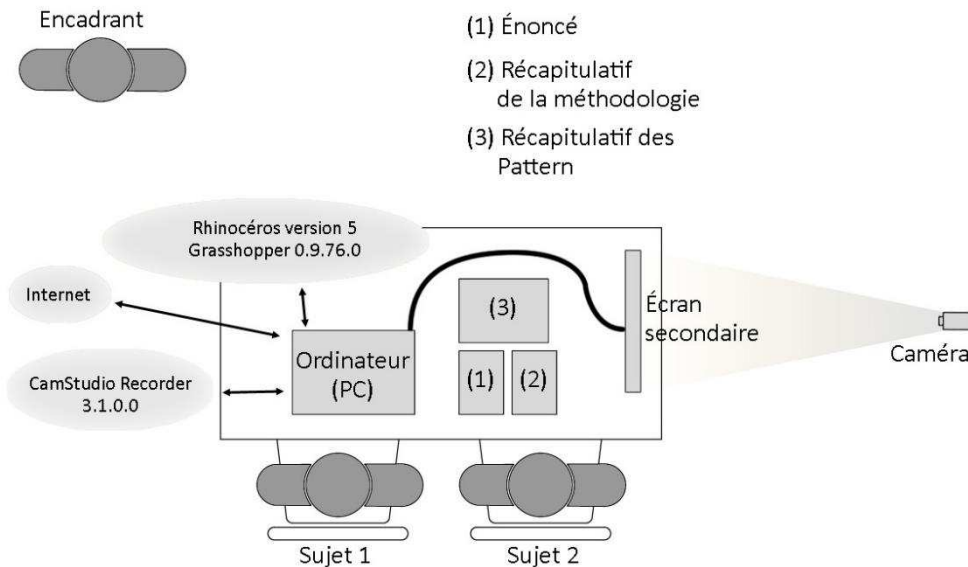


Figure 54 - Disposition de l'expérience zéro

À la fin de l'expérience, les sujets ont complété un questionnaire tel que présenté en page 54 et en annexe 10. Dans le cadre de l'expérience zéro, ce questionnaire a également permis de relever des améliorations à effectuer en vue de la mise en œuvre de l'expérience complète. Cependant, ces modifications étant minimales, les résultats de l'expérience zéro peuvent être considérés et étudiés pour appuyer l'analyse de l'expérience réalisée avec les étudiants du cours de MAN. Les modifications apportées sont les suivantes :

- Dans la capsule de cours servant d'introduction à la MAPMAP (annexe 5), l'explication du principe de dé-foisonnement des liaisons a été précisée. Ce changement a été relevé sur base des commentaires laissés dans les questionnaires.

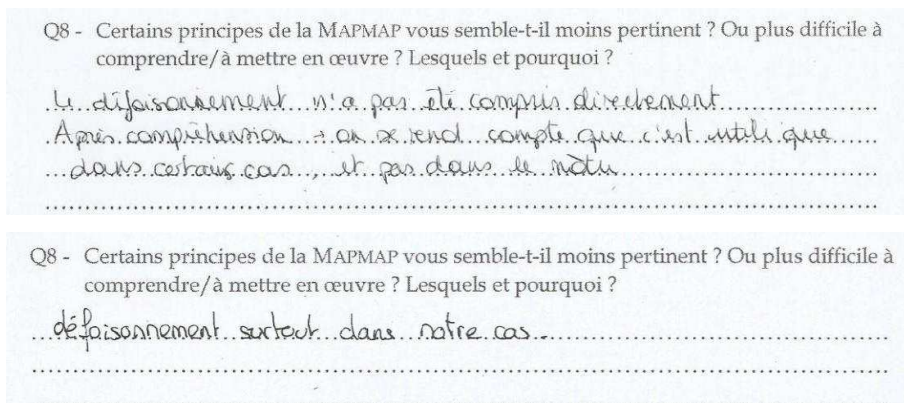


Figure 55 - Réponse à la question 8 du questionnaire complété par les sujets de l'expérience zéro

- L'énoncé de l'exercice de modélisation a été précisé : certains paramètres d'entrée ne possédaient aucune valeur (notamment pour l'épaisseur des montants ainsi que l'intervalle les séparant). Un ordre de grandeur a donc été précisé dans l'énoncé utilisé pour l'expérience concrète.

- Lors de l'enregistrement de l'expérience, la caméra et l'écran secondaire ont recueillis des données de bonne qualité alors que l'enregistrement avec le logiciel CamStudio n'a pas fonctionné au vu du poids des fichiers créés. Ainsi cet outil a été retiré du protocole d'expérimentation.
- Enfin, une maquette physique du modèle a été ajoutée à la liste des documents mis à disposition.

Réalisation de l'expérience

L'expérience concrète s'est ensuite réalisée le mercredi 22 mars 2017, entre 11h et 15h. Les étudiants ayant suivis la formation sur la modélisation paramétrique du cours de MAN en cette année académique 2016-2017, sont au nombre de 18. Ainsi ils ont été répartis en neuf binômes. Les profils des étudiants étant variés, les binômes ont été formés avec un certain degré de mixité. En effet, neuf de ces étudiants suivent la formation classique d'ingénieur architecte proposée par l'ULg, sept sont des étudiants ingénieur français effectuant une année de passerelle vers le master d'ingénieur architecte de l'ULg, une étudiante provient d'un cursus de bachelier en architecture et effectue une passerelle vers le master d'ingénieur architecte et finalement une architecte externe, doctorante, suit la formation en élève libre.

L'assistant John Schrayen a aidé à réaliser le choix des binômes vedettes car celui-ci a une bonne connaissance des étudiants puisqu'il les a déjà encadrés dans différents cours de la formation en ingénieur civil architecte. En effet, le choix a été effectué de la manière suivante : d'une part, les binômes devaient être composés d'étudiants ayant démontrés un minimum de réceptivité vis-à-vis de la modélisation paramétrique au cours des trois premiers travaux pratiques de la formation et, d'autre part, nous recherchions des étudiants de nature expressive en vue de récolter un maximum d'informations auditives dans les enregistrements audio-visuels.

Étant donné que l'expérience zéro n'a pas conduit à des modifications significatives, donc que les résultats peuvent appuyer ceux de l'expérience, et étant donné que les sujets de cette expérience zéro avaient été formés à la mise en œuvre de la MAPMAP, la répartition des neuf binômes consiste en quatre binômes dans le groupe 1 et cinq dans le groupe 2.

Le tableau suivant illustre la répartition des groupes et présente les caractéristiques des ordinateurs sur lesquels ils ont construits leurs modèles.

Caractéristiques des binômes			Caractéristiques des ordinateurs		
Groupe	Binôme	Type de binôme	Système d'exploitation	Processeur	Carte Graphique
1	1	Vedette	Windows	Intel Core i5	NVIDIA Geforce 720M
	2	Classique	Windows	Intel Core i5	NVIDIA Geforce 840M
	3	Classique	Windows	Intel Core i7	NVIDIA Geforce 720M
	4	Classique	Windows	Intel Core i7	AMD Radeon R7 M340 et Intel HD Graphics 250
2	5	Vedette	Windows	Intel Core i7	NVIDIA Geforce GTX 760M
	6	Classique	Windows	Intel Core i7	NVIDIA Geforce GT 720M
	7	Classique	Windows	Intel Core i7	NVIDIA Geforce GTX 860M
	8	Classique	Windows	Intel Core i7	NVIDIA Geforce GTX 960M
	9	Classique	Windows	Intel Core i7	NVIDIA Geforce 920M

En ce qui concerne le déroulement de l'expérience, les sujets sont installés par binômes face à leurs ordinateurs respectifs. Les étudiants utilisent le logiciel Rhinoceros (version 5) et son extension paramétrique Grasshopper (version 0.9.76.0). Selon leur type de groupe, les étudiants disposent des documents présentés précédemment, en page 54.

Pour éviter toute communication concernant la MAPMAP durant l'expérience, les deux groupes étaient séparés en deux lieux avec deux encadrants : le groupe 1, que nous encadrions, a réalisé l'expérience dans les studios du LUCID; tandis que le groupe 2, encadrés par l'assistant John Schrayen, se trouvait dans une salle de classe du bâtiment B52 de l'Université de Liège. En outre les binômes vedettes étaient également séparés des autres binômes pour permettre un enregistrement auditif de bonne qualité.

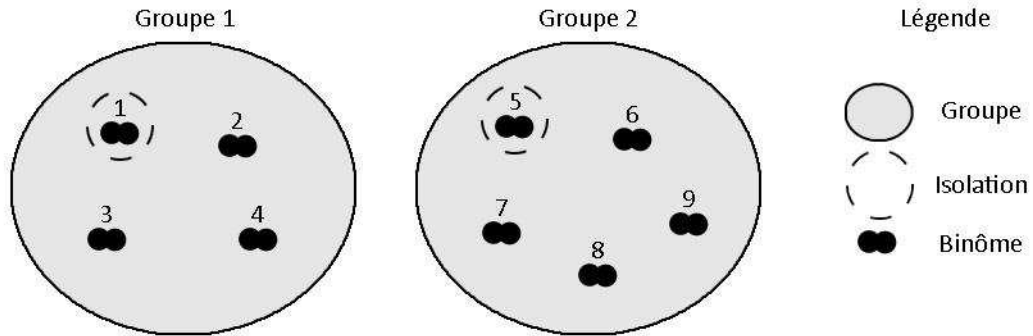


Figure 56 - Schéma de répartition spatiale des binômes



Figure 57 – Groupe 1, binômes classiques

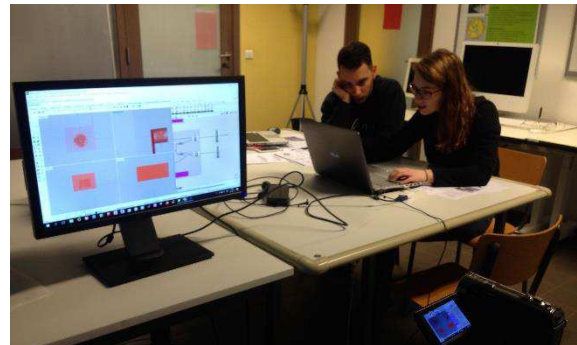


Figure 58 – Groupe 1, binôme vedette



Figure 59 - Groupe 2, binômes classiques

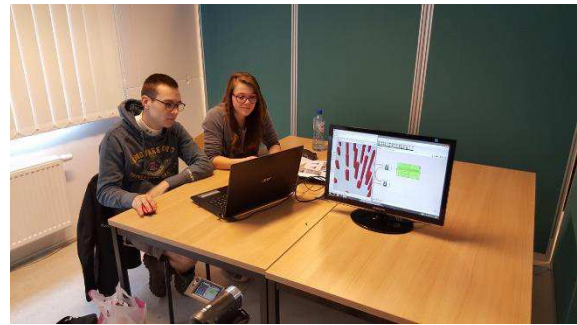


Figure 60 – Groupe 2, binôme vedette

Enfin, les schémas suivants permettent de rappeler le contexte d'enregistrement des deux binômes vedettes :

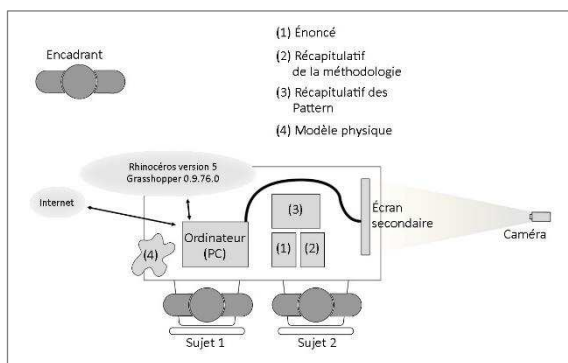


Figure 61 - Contexte d'expérimentation : binôme vedette, groupe 1

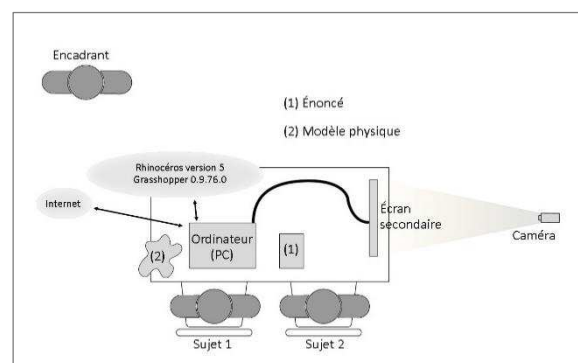


Figure 62 - Contexte d'expérimentation : binôme vedette, groupe 2

5.1.2. Critères d'analyse

En vue d'étudier l'efficacité de la MAPMAP dans le cadre de sa mise en œuvre durant la phase de conception d'un modèle paramétrique, nous entamerons une analyse comparative des groupes 1 et 2, 'formé' et 'instinctif'. Cette analyse sera basée sur différents documents. D'abord, les modèles créés par les différents groupes seront étudiés sur base de critères quantitatifs en lien avec la MAPMAP. Ensuite, les observations recueillies grâce aux modèles seront complétées par un séquençage des enregistrements des expériences des binômes vedettes. Enfin, les questionnaires complétés par les étudiants à la fin de l'expérience permettront de compléter l'analyse des modèles et des enregistrements. Le tableau suivant est un récapitulatif des données récoltées pour chaque binôme, X signifiant que l'on possède un élément de donnée (XX = deux éléments) et O signifiant que cette donnée n'a pas été récoltée

Groupe	Binôme	Modèle	Enregistrement	Questionnaire
1	Expérience zéro	X	X	XX
	1	X	X	XX
	2	X	O	XX
	3	X	O	XX
	4	X	O	XX
2	5	X	X	XX
	6	X	O	XX
	7	X	O	XX
	8	X	O	XX
	9	X	O	XX

Analyse des modèles

Pour effectuer l'analyse des modèles, seize critères majoritairement quantitatifs ont été établis les deux premiers sont d'ordre général, tandis que les suivants sont fondés sur les principes de la MAPMAP.

Tout d'abord, les deux critères d'ordre général sont le nombre d'objets et le nombre de liens au sein du modèle.

Ensuite, quatre critères se rapportent à la Stratégie de Division : le nombre de modules primaires, le nombre de modules secondaires, le nombre de patterns utilisés et reconnus en tant que tels, et le nombre de typologies de groupes utilisées par changement de couleur de sorte à distinguer les modules primaires et secondaires.

Quatre critères également sont établis sur base du Réflexe de Documentation : le nombre de panneaux utilisés pour titrer les modules, le nombre de panneaux utilisés comme description d'un module, le nombre d'objets renommés, et le nombre de typologies de panneaux utilisées pour différencier les titres des notes descriptives et autres. Par panneaux, nous entendons l'utilisation de l'outil 'Params-Input-Panel' suggéré dans la présentation de la mise en œuvre du Réflexe de documentation.

Le Principe d'Encapsulation engendre trois critères d'analyse qui sont : le nombre de clusters créés, le nombre d'objets contenus dans ces clusters et le nombre de répétition des clusters.

Le Dé-foisonnement des Liaisons crée également trois critères : le nombre de paramètres du modèle, le nombre de duplications des paramètres, et le nombre d'avatars de paramètres réalisés.

La Description Générique est plus difficilement évaluable. Elle peut éventuellement l'être sur base du nombre de paramètres nécessaires au modèle. Cependant ce critère permet une simple analyse de l'abstraction du modèle mais pas de sa description générique. En outre ce critère est déjà repris en vue d'évaluer le principe de Dé-foisonnement des liaisons. Finalement l'analyse de ce critère peut se faire à l'aide du nombre d'objets et de liens utilisés au sein du modèle et ces deux critères ont déjà été établis pour évaluer le modèle d'un point de vue global.

Séquençage des enregistrements

Le critère du temps est important dans le cadre de l'évaluation de la pertinence de la MAPMAP. En effet, l'objectif de la MAPMAP est de simplifier la communication d'un modèle et donc de faire gagner du temps aux utilisateurs d'un modèle. Cependant, le concepteur sera réticent à utiliser une méthode qui lui fait perdre du temps dans la construction du modèle. Ce critère de temps sera donc analysé dans cette première expérience ainsi que dans la seconde qui vise la communication du modèle.

Dans l'objectif d'analyser le facteur temps de manière détaillée, les enregistrements des binômes vedettes vont être étudiés à travers un séquençage. Celui-ci reprendra les

caractéristiques suivantes : les étapes principales de conception du modèle, les sous-étapes de conception dont celles liées à la MAPMAP c'est-à-dire les étapes de structuration du modèle, et les événements ponctuels liés à la MAPMAP c'est-à-dire la mise en œuvre ponctuelle d'un principe de la méthode.

Questionnaires

Les derniers éléments mis en œuvre en vue d'évaluer la MAPMAP dans la phase de conception d'un modèle paramétrique sont les deux formats de questionnaires complétés par chaque étudiant en fin d'expérience selon leur groupe. Ces questionnaires permettront de compléter les observations effectuées sur base des modèles et des enregistrements audio-visuels. Les deux modèles de questionnaire relatifs aux deux groupes sont établis dans l'objectif d'apporter ces informations complémentaires, c'est pourquoi ils sont constitué de questions orientées selon les quatre thématiques suivantes : (a) le niveau de réceptivité de l'étudiant vis-à-vis de la modélisation paramétrique en général, (b) le niveau de compréhension de la MAPMAP par l'étudiant (uniquement groupe 1), (c) les difficultés rencontrées lors de la réalisation de l'exercice de modélisation, et (d) l'appréciation de la MAPMAP (uniquement groupe 1). Les questions posées sont les suivantes :

- a. Suite à la formation dispensée durant les séances de TP de MAN, comment considérez-vous votre niveau de compréhension de la modélisation paramétrique et du logiciel Grasshopper ?
- b. Quelle est votre compréhension de la MAPMAP, la méthodologie qui vous a été présentée en début d'expérience ?
- c. Quelles difficultés avez-vous éventuellement rencontré pour réaliser l'exercice ?
- d. D'après vous, quelle est la contribution de la MAPMAP ?
 Quelles difficultés avez-vous rencontré lors de la mise en œuvre la MAPMAP ? Pourquoi?
 Quels avantages identifiez-vous dans l'application de la MAPMAP ?
 Quels inconvénients identifiez-vous dans l'application de la MAPMAP ?
 Certains principes de la MAPMAP vous semblent-ils moins pertinents ? Ou plus difficiles à comprendre/à mettre en œuvre ? Lesquels et pourquoi ?

Les exemplaires des questionnaires fournis aux étudiants sont disponibles aux annexes 9 et 10.

5.2. Expérimentation de communication de modèles paramétriques

5.2.1. Protocole de l'expérience

Le principal objectif de la MAPMAP, c'est-à-dire faciliter la communication d'un modèle paramétrique entre utilisateurs, fait l'objet de cette seconde phase d'expérimentation. Pour ce faire, les modèles construits par les étudiants dans la première expérience vont être exploités par d'autres étudiants qui suivent la formation d'ingénieur civil architecte à l'université de Liège et qui ont déjà suivi le cours de MAN.

Sujets

Les étudiants sélectionnés pour cette seconde expérience sont des étudiants de 2^{ème} master qui ont suivi le cours de MAN durant l'année académique 2014-2015. Comme présentés dans le cadre de l'expérience zéro de la première expérience (page 55), ces étudiants ont été formés à la modélisation paramétrique par monsieur Anis Gallas à travers quatre travaux pratiques de modélisation : une esquisse du Gherkin de N. Foster, un traitement de surface par des cellules Voronoï, un traitement de surface modulaire par maillage et la modélisation d'un pavillon en forme de coquillage.

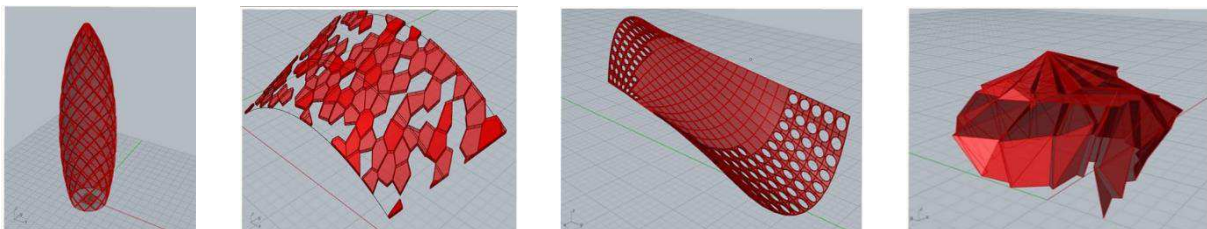


Figure 63 - Quatre modèles paramétriques réalisés dans le cadre du cours de MAN 2015

Les deux étudiants ayant participé à l'expérience zéro de la première expérience n'ont pas participé à cette seconde phase dans le but d'avoir un échantillon de sujet homogène vis-à-vis de leur connaissance du modèle à exploiter.

Contexte de l'expérience

En vue de réaliser cette expérience, deux modèles réalisés par les étudiants de 3^{ème} bachelier ont été sélectionnés, l'un de ces modèles provient d'un binôme du groupe 1 tandis que l'autre provient du groupe 2. Ainsi nous pourrons à nouveau réaliser une analyse comparative des résultats.

L'expérience se déroule en trois phases. Tout d'abord, l'exercice de modélisation réalisé par les étudiants de 3^{ème} bachelier est expliqué au sujet de cette seconde expérience et l'objectif du travail qu'il va devoir réaliser lui est exposé. En l'occurrence, les étudiants vont devoir réaliser un exercice d'exploitation d'un modèle préconçu qui se fait par la mise en condition suivante : « Leurs collègues, les étudiants de 3^{ème} bachelier, ont réalisé des modèles paramétriques qui doivent faire l'objet d'un prototypage. En vue de réaliser ce prototypage, il convient de régler certains paramètres du modèle pour assurer sa bonne confection. Ces paramètres à ajuster sont les suivants : l'épaisseur des montants de la structure qui doit correspondre à l'épaisseur du matériau utilisé lors de la découpe laser, et l'entraxe des montants qui doit être suffisamment faible pour permettre la stabilité du modèle. » Ainsi, les étudiants disposent d'un énoncé, qui se trouve en annexe 11, qui leur explique les deux changements de paramétrage à effectuer au sein du modèle qui leur est fourni.

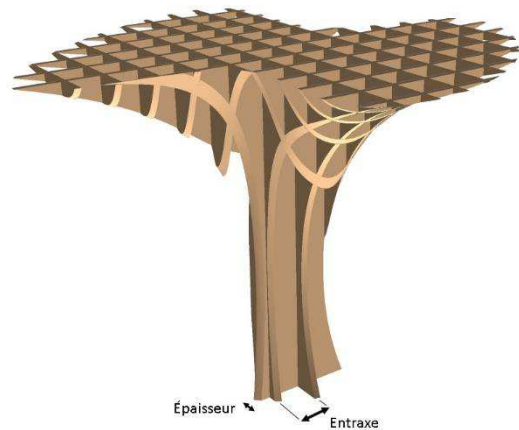


Figure 64 - Modifications à apporter au modèle

Pour la deuxième étape de l'expérience, le sujet dispose de 20 minutes pour prendre connaissance du modèle et effectuer les changements qui lui ont été présentés. Il s'agit donc de changer l'épaisseur des montants pour que celle-ci soit égale à 0,195 unités et de changer l'entraxe des montants pour que celui-ci soit égal à 1 unité. Le choix de limiter l'expérience à 20 minutes d'exploration du modèle, a été établi sur l'hypothèse que si le sujet ne parvient pas à réaliser les deux changements dans ce laps de temps, il n'y parviendra pas en un temps raisonnable, cela s'expliquant sûrement par un manque d'expérience d'utilisation du logiciel.

Enfin, une troisième étape consiste à compléter un questionnaire à la fin de la réalisation de l'exercice. Ce questionnaire, qui sera détaillé lors de la présentation des critères d'analyse et qui est disponible à l'annexe 12, poursuit deux objectifs : d'une part, évaluer la réceptivité et la connaissance de l'étudiant vis-à-vis du modeleur et, d'autre part, prendre connaissance de l'avis de l'étudiant concernant le modèle avec lequel il a dû travailler.

Expérience zéro

Avant de réaliser l'expérience concrète avec les étudiants du cours de MAN, une nouvelle expérience zéro a été réalisée en vue d'évaluer les modèles qui seraient exploités lors de l'expérience concrète. En effet, seuls deux modèles sont sélectionnés pour la réalisation de cette seconde expérience. Ainsi, deux étudiants de 2^{ème} master ont pris part à cette

expérimentation zéro tour à tour, le premier pour sélectionner un modèle du groupe 1 et le second pour sélectionner un modèle du groupe 2.

Concernant le déroulement de l'expérience, le sujet est assis à un bureau face à un ordinateur de travail avec un processeur Intel Core i7 et une carte graphique AMD Radeon HD 8750M et Intel HD Graphics 4600. Sur cet ordinateur est installé le logiciel Rhinoceros (version 5) et son extension paramétrique Grasshopper (version 0.9.76.0). Après avoir exposé le contexte d'expérimentation à l'étudiant, celui-ci dispose des outils suivants pour effectuer l'exercice : un énoncé, une maquette physique du modèle, et un modèle paramétrique préalablement réalisé par un binôme d'étudiants de 3ème année.

Pendant que l'étudiant complète l'exercice, un chronomètre permet d'évaluer le temps nécessaire à l'étudiant pour changer chacun des paramètres et il permet également d'arrêter l'expérience au bout des 20 minutes fixées dans le cas où le sujet n'aurait pas fini.

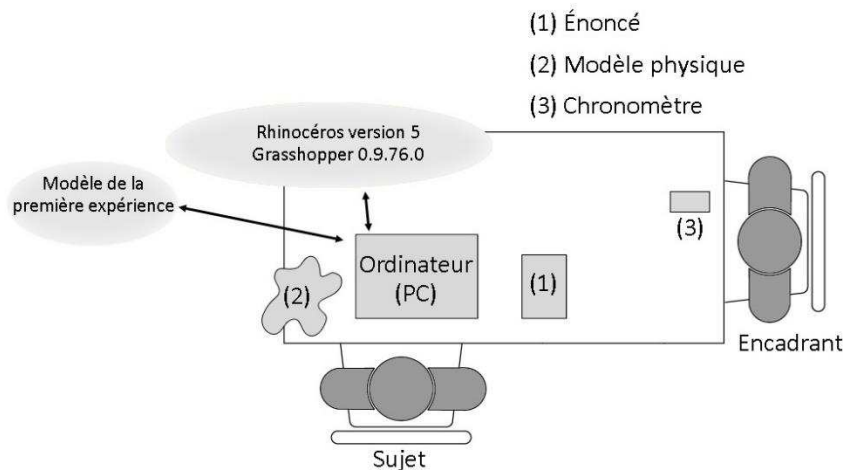


Figure 65 - Disposition de l'expérience zéro servant à valider l'expérience de communication

Au niveau des modèles testés, le premier choix s'était porté sur les modèles réalisés par les binômes vedettes, soient les modèles 1 et 5. Cependant, ces modèles n'étaient pas les plus adéquats notamment parce qu'ils présentaient des particularités qui risquaient de fausser les résultats. Par exemple, dans le cas du modèle 1, qui fut testé avec le premier étudiant de l'expérience zéro, les étudiants de la première expérience ont tenté de mettre en œuvre un cluster dans lequel se trouvait le paramètre d'entraxe des panneaux. Cependant ce cluster n'a pas été mis en œuvre correctement et le risque était qu'aucun sujet de la seconde expérience ne trouve ce paramètre.

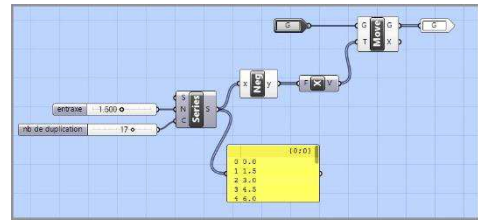
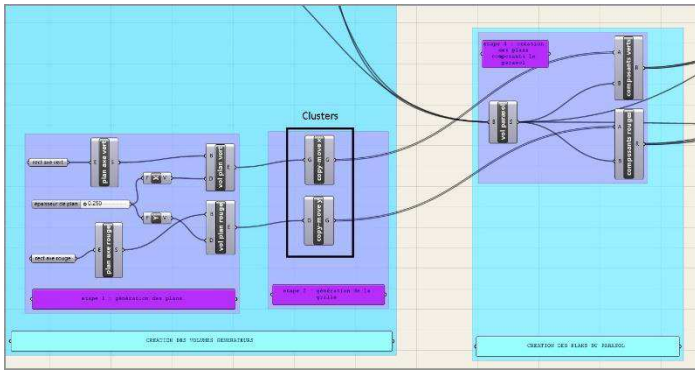


Figure 67 - Contenu du cluster créé dans le modèle 1

Figure 66 - Partie du modèle 1 contenant deux clusters

Au vu de cet échec lors de la première expérience zéro, le choix des modèles a été effectué sur base de l'analyse quantitative des modèles, qui sera exposée dans le chapitre suivant. Les modèles sélectionnés sont alors ceux qui présentent les caractéristiques les plus proches de la moyenne, en vue d'éviter les particularités. Ce furent les modèles 3 et 8 qui furent sélectionnés. Après vérification de la conformité du modèle 3, le modèle 8 fut soumis au second sujet de l'expérience zéro. Cependant, il s'avéra que ce modèle présentait également une particularité non-visible depuis l'analyse quantitative : l'épaisseur des montants avait été introduite via une construction dans le logiciel Rhinoceros si bien que ce paramètre n'apparaissait pas dans la représentation symbolique du modèle, dans Grasshopper.

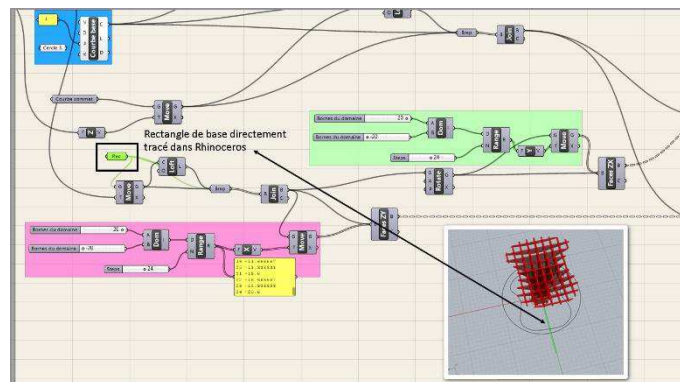


Figure 68 - Morceau du modèle 8 montrant que la base des montants a été directement créée dans Rhinoceros

Finalement, ce fut le modèle 6 qui fut testé et validé pour la suite de l'expérience en tant que modèle réalisé de manière instinctive, sans formation à la mise en œuvre de la MAPMAP.

Déroulé de l'expérience

L'expérience zéro ayant permis de mettre en évidence les failles de l'expérimentation situées au niveau des modèles sélectionnés, l'expérience concrète a pu être mise en œuvre le lundi 24 avril 2017 dans une classe du bâtiment B52 de l'université de Liège.

Pour ce faire, huit étudiants de 2^{ème} master en ingénieur civil architecte à l'université de Liège, ont participé à cette phase expérimentale. Parmi ces huit étudiants, quatre ont réalisé

l'exercice d'exploitation d'un modèle sur base du modèle 3 qui est donc un modèle réalisé par un binôme du groupe 1, soit le groupe formé à la MAPMAP. Par ailleurs, les quatre étudiants restant ont effectué l'exercice sur base du modèle 6 qui a été construit par un binôme du groupe 2, soit le groupe instinctif, non-formé à la MAPMAP.

Toutes ces expérimentations ont été mises en œuvre l'une à la suite de l'autre, ainsi les étudiants ont réalisé l'expérience dans le même contexte et sur le même ordinateur, soit un PC avec un processeur Intel Core i7 et une carte graphique AMD Radeon HD 8750M et Intel HD Graphics 4600. Comme lors de l'expérience zéro, le déroulement de l'expérimentation est le suivant : d'abord une mise en contexte est présentée au sujet par un encadrant, ensuite le sujet a 20 minutes pour effectuer les changements requis. Pour ce faire, il dispose d'un énoncé (disponible à l'annexe 11), d'une maquette physique du modèle et de l'un des deux modèles sélectionnés.



Figure 69 - Réalisation de la seconde expérience

Un chronomètre permet de mesurer la durée de l'exercice et un chronométrage intermédiaire est pris lorsque le premier paramètre a été modifié. À la fin de l'expérience, l'étudiant complète le questionnaire qui lui a été remis (le questionnaire est disponible à l'annexe 12).

5.2.2. Critères d'analyse

Concentrons-nous à présent sur les outils qui serviront l'analyse de l'efficacité de la MAPMAP dans le cadre de la communication de modèle paramétrique entre concepteur et utilisateur externe.

Tout d'abord, comme pour la première expérience, le facteur temps est l'une des caractéristiques primordiales de la méthode. Ce facteur avait déjà une certaine importance dans la phase de conception du modèle. Il est plus important encore au sein de cette seconde expérience puisqu'il traduit l'objectif initial de la MAPMAP qui consiste à faciliter la communication d'un modèle. C'est pour cette raison que ce paramètre de temps est le premier critère de l'analyse de cette expérimentation.

Ensuite, il est important de remarquer que l'échantillon d'étudiants choisis pour cette seconde expérience est plus hétérogène que celui de la première expérience. En effet, ces étudiants ont reçu la formation paramétrique du cours de MAN deux ans au paravant si bien que, durant cette période, certains ont eu l'occasion d'approfondir leurs connaissances de ce type de logiciels, tandis que d'autres ont commencé à en oublier le fonctionnement. Il est important de prendre en compte ces différences de formations au sein de l'analyse des résultats chronométriques. C'est dans cet objectif qu'un questionnaire a été établi.

Ce questionnaire possède une seconde vocation : l'avis du sujet concernant le modèle avec lequel il a dû travailler. Finalement le questionnaire (disponible à l'annexe 12) se compose des quatre questions suivantes :

- Suite à la formation dispensée durant les séances de TP de MAN, comment considérez-vous votre niveau de compréhension de la modélisation paramétrique et du logiciel Grasshopper ?
- Quelles difficultés avez-vous rencontré pour réaliser l'exercice ?
- Trouvez-vous le modèle bien construit ? Pourquoi ?
- Auriez-vous des suggestions pour améliorer la compréhension du modèle ?

La première de ces questions permet d'évaluer le niveau de compétence du sujet vis-à-vis de l'utilisation du modeleur, les troisième et quatrième questions permettent de recueillir l'avis de l'étudiant sur le modèle qu'il a utilisé et finalement la deuxième question permet de recueillir des informations de manière plus générale concernant soit la connaissance du logiciel par le sujet, soit des problèmes de compréhension du modèle, soit des facteurs externes.

En conclusion, l'analyse de la seconde expérience s'effectuera sur base des résultats chronométriques discutés et complétés par les informations recueillies dans les questionnaires.

6. RÉSULTATS ET DISCUSSION

Les expériences ayant été présentées au chapitre précédent, ce chapitre sera consacré à l'analyse de leurs résultats. Grâce à ces résultats, nous mettrons en évidence les limites de la MAPMAP et de l'étude qui y a été consacrée et, sur base de ces limites, nous proposerons différentes perspectives d'évolution de la MAPMAP et de son expérimentation.

Plan du chapitre :

6.1.	RÉSULTATS DE LA PREMIÈRE EXPÉRIENCE.....	75
	6.1.1. <i>Analyse des modèles</i>	75
	6.1.2. <i>Séquençage des enregistrements</i>	86
6.2.	RÉSULTATS DE LA SECONDE EXPÉRIENCE.....	90
6.3.	LIMITES ET PERSPECTIVES DE LA MAPMAP ET SON ÉTUDE.....	94
	6.3.1. <i>Limites</i>	94
	6.3.2. <i>Perspectives</i>	94

6.1. Résultats de la première expérience

Dans la partie « 4.2.2. Critères d'Analyse », nous avons établis trois types d'éléments qui serviront l'analyse de cette première expérience consacrée à la mise en œuvre de la MAPMAP durant la phase de conception d'un modèle. Ces trois éléments vont structurer notre analyse de la manière suivante.

Le premier élément qui sera l'axe principal de nos observations, consiste à réaliser une étude qualitative des différents modèles réalisés par les étudiants, cette étude s'appuiera sur les critères quantitatifs choisis précédemment.

Le deuxième élément, qui sera l'axe secondaire de l'analyse, se focalise sur le facteur temps. Il consiste en une analyse temporelle des enregistrements audio-visuels réalisés dans le cadre de la réalisation de l'expérience par les binômes vedettes.

Enfin, le troisième élément ne formera pas un axe de l'étude à lui seul mais il servira à compléter les observations réalisées au cours des deux axes d'analyse précités. Cet élément a pour but de fournir des informations générales qui pourront compléter et discuter les observations effectuées sur base des modèles et/ou des enregistrements audio-visuels. Ces informations seront récoltées au sein des questionnaires remplis par chaque étudiant à la fin de la construction du modèle paramétrique.

6.1.1. Analyse des modèles

L'analyse des modèles est effectuée suivant la démarche suivante. Tout d'abord, un tableau récapitulatif est établi à partir des critères quantitatifs développés sur base des principes de la MAPMAP. Ce tableau contient des données relatives à chaque modèle, dont le modèle de l'expérience zéro. Ensuite, une analyse particulière de chaque modèle est réalisée pour mettre en évidence leurs caractéristiques de conception. Ces éléments seront relevés et étudiés, puis discutés grâce aux données recueillies au sein des questionnaires.

Pour commencer, voici le tableau des critères d'analyse quantitatifs :

Critères d'analyse		Analyse des dix modèles										Moyenne
Critères relatifs aux binômes	Modèle	0	1	2	3	4	5	6	7	8	9	/
	Groupe	1	1	1	1	1	2	2	2	2	2	/
	Binôme (Vedette ou Classique)	V	V	C	C	C	V	C	C	C	C	/
Critères Globaux	Objets	64	44	54	74	78	49	72	62	69	89	65
	Liens	73	43	49	83	68	63	81	72	84	97	71
Critères relatifs à la Stratégie de Division	Groupes	6	3	4	5	4	0	0	6	6	2	4
	Sous-groupes	12	6	5	8	14	/	/	0	0	3	5
	Patterns reconnus	0	0	2	0	11	/	/	/	/	/	2
	Typologies de groupes	3	2	3	5	3	/	/	1	6	3	3
Critères relatifs au Réflexe de Documentation	Notes de titrage	11	3	11	5	4	0	0	6	0	0	4
	Notes descriptives	6	6	0	0	11	0	0	0	0	0	2
	Objets renommés	15	26	16	19	11	10	5	11	13	1	13
	Typologies de notes	1	2	1	1	2	/	/	1	/	/	1
Critères relatifs au Principe d'encapsulation	Clusters	0	2	0	0	0	0	0	0	0	0	0
	Objets dans les clusters	/	8 & 9	/	/	/	/	/	/	/	/	/
	Duplications des clusters	/	0	/	/	/	/	/	/	/	/	/
Critères relatifs au Dé-foisonnement des liaisons	Paramètres	12	6	13	9	11	7	15	10	9	19	11
	Duplications des paramètres	4	0	1	9	4	0	4	0	5	14	4
	Avatars	0	0	0	0	0	0	0	0	0	0	0

Sur base de ce tableau, chaque modèle peut être analysé de sorte à en relever les spécificités et les informations contenues dans les questionnaires permettront de discuter ces analyses. Les modèles complets sont illustrés à l'annexe 13.

Modèle de l'expérience zéro, groupe 'formé'

Au niveau de ce modèle, qui a été réalisé en appliquant la MAPMAP, on remarque que les bases de la méthodologie ont été appliquées. En effet, de nombreux groupes et sous-groupes ont été formés mais on remarque une erreur dans le suivi de la typologie de ceux-ci, l'un des groupes n'étant pas de la bonne couleur vis-à-vis de la hiérarchie adoptée. En outre des titres et description de fonctionnalité ont été ajoutés tout au long du modèle, cependant aucun changement de typologie n'est observable.

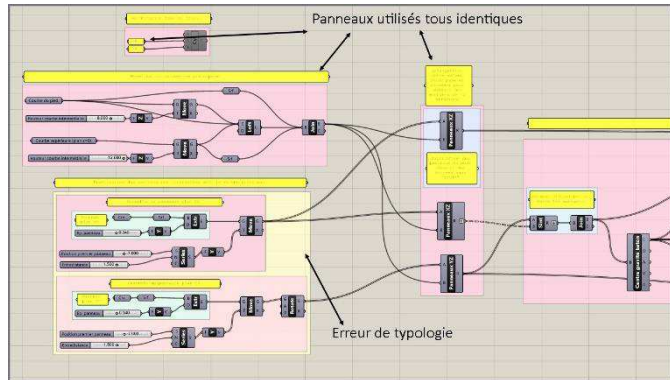


Figure 70 - Modèle 0 - erreurs dans l'application de la MAPMAP

Finalement, remarquons que, lors de l'expérience zéro, l'encadrant ne possédait pas encore une parfaite connaissance des particularités du modèle à concevoir. En effet, l'application de certains objets contenant des algorithmes booléens, nécessitait une mise en œuvre spécifique. Les problèmes engendrés par le placement de ces objets, prirent plus de temps à être résolus que lors de l'expérience concrète où les encadrants étaient avertis des subtilités de modélisation. Nous posons l'hypothèse que la perte de temps dans la création du modèle a pu engendrer un agacement chez les concepteurs qui s'est répercuté sur le temps consacré à la structuration du modèle. Cette hypothèse justifierait l'erreur de typologie de groupe ainsi que le manque de développement du Réflexe de Documentation.

Modèle 1, groupe 'formé'

Ce modèle peut être qualifié de 'presque parfait' en termes d'application de la MAPMAP. Tout d'abord, un nombre raisonnable de groupes et de sous-groupes a été mis en œuvre avec une différenciation au sein des typologies. Ensuite, une note a été ajoutée à chaque groupe et sous-groupe dans le respect de la typologie de ceux-ci. Un très grand nombre d'objets a été renommé et aucun paramètre n'a été dupliqué inutilement. Enfin, le nombre d'objets et de liens est très faible ce qui simplifie le modèle. Cela a été rendu possible par la création d'un cluster lors de la troisième étape de modélisation où la même opération devait être réalisée deux fois. Cependant, dans ce cadre-ci, le cluster n'a pas été parfaitement mis en œuvre :

certaines paramètres d'entrée ne figurent pas dans les entrées du cluster si bien que celui-ci a été modifié lors de sa duplication. Finalement, on obtient deux clusters différents qui remplissent uniquement la tâche de cacher des parties du modèle.

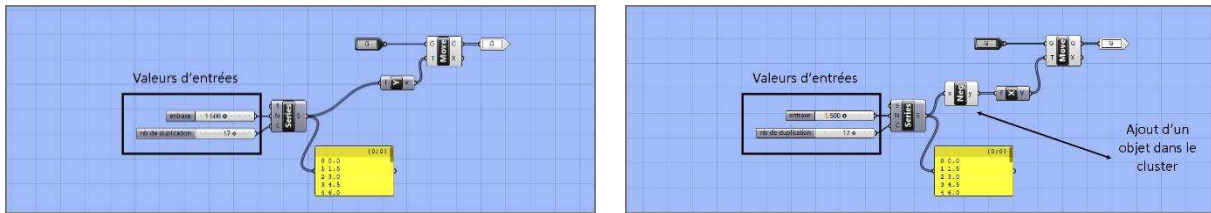


Figure 71 - Modèle 1 - contenu des clusters

L'explication proposée est que les étudiants n'ont jamais eu l'occasion de mettre en place des clusters préalablement dans les modèles réalisés dans le cadre de la formation du cours de MAN, si bien qu'ils n'ont pas parfaitement assimilé la manière dont il faut mettre les clusters en œuvre. En outre, dans cet exercice de modélisation, l'application du principe d'encapsulation n'est pas une nécessité car le modèle ne nécessite pas un grand nombre de répétitions d'une opération.

Modèle 2, groupe 'formé'

Dans ce modèle, la mise en œuvre de la MAPMAP n'est pas aboutie. En effet, le premier degré de division et documentation du modèle est atteint : on retrouve quatre groupes titrés. Cependant, la hiérarchisation du modèle au degré supérieur pourrait encore évoluer : peu de sous-groupe ont été créés et une seule typologie de panneaux est utilisée. En outre, peu d'objets et de liens ont été utilisés et de nombreux objets ont été renommés, ces deux caractéristiques rendent le modèle moins complexe.

Enfin ce binôme a reconnu et appliqué deux patterns qui sont en accord avec la structure de leur modèle. Cependant, on découvre à travers les questionnaires que la mise en œuvre des patterns a été vécue comme une obligation difficile à remplir.

Q5 - Quelles difficultés avez-vous rencontrées lors de la mise en œuvre de la MAPMAP ? Pourquoi ?

- associer les termes aux étapes

Q5 - Quelles difficultés avez-vous rencontrées lors de la mise en œuvre de la MAPMAP ? Pourquoi ?

Repérage des Patterns : difficulté d'identifier les catégories... et d'y associer des fonctions.

Figure 72 – Extraits des questionnaires du binôme 2

Ceci amène à penser que l'essence des patterns, c'est-à-dire répondre à un problème type, n'a pas été assez assimilée car les étudiants n'ont pas eu le temps de s'habituer à cette notion.

Cependant, dans leur effort d'application des patterns, ils ont tout de même compris et appliqué deux de ceux-ci.

Modèle 3, groupe 'formé'

Dans ce modèle, l'essence des principes de la MAPMAP ne semble pas avoir été totalement assimilée non plus. On voit que des efforts de structuration du modèle ont été effectués mais avec quelques maladresses. En effet, de nombreux groupes et sous-groupes ont été utilisés avec une trop grande variété de typologies, si bien qu'elles n'ont pas une signification de hiérarchisation. Ensuite, des titres pertinents ont été ajoutés aux étapes principales de modélisation mais aucune note complémentaire ne documente le modèle. En outre, on remarque tout de même que de nombreux objets ont été renommés. Enfin, ce modèle à la particularité d'utiliser une faible quantité de paramètres d'entrées mais de les avoir dupliqués un grand nombre de fois au sein des mêmes modules primaires, ce qui va à l'encontre du principe de dé-foisonnement des liaisons, qui préconise implicitement de limiter le nombre de paramètre d'entrée et d'utiliser des avatars lorsque cela est nécessaire. Cependant, nous pouvons remarquer que ce principe de conception d'avatars n'était pas nécessaire à cet exercice de modélisation.

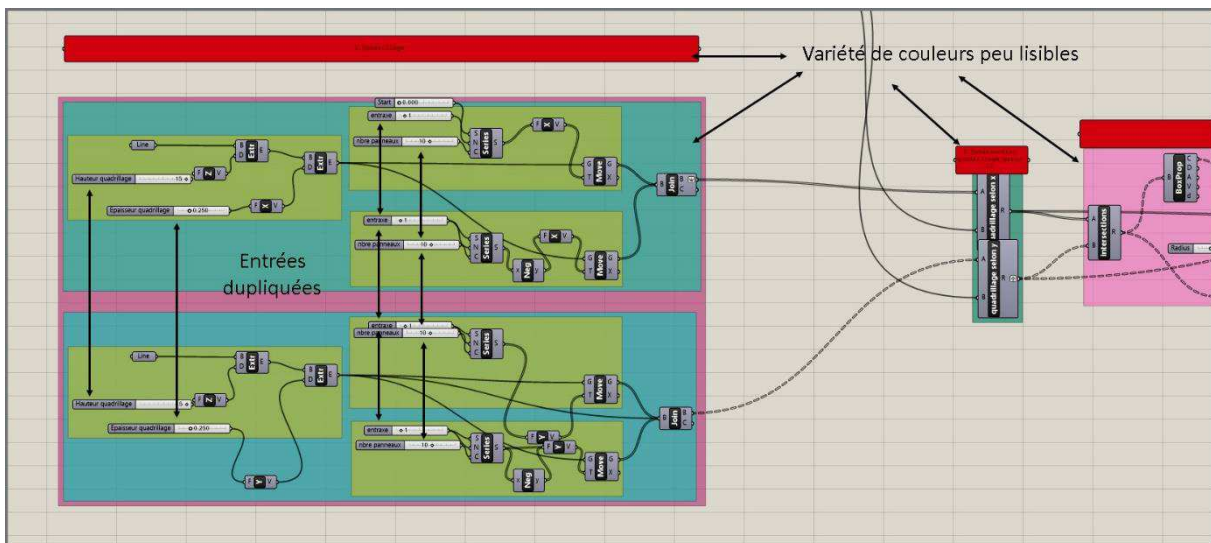


Figure 73 - Modèle 3 - typologies peu compréhensibles et nombreux paramètres dupliqués inutilement

L'hypothèse la plus probable expliquant cette mise en œuvre 'brouillonne' de la MAPMAP, est que les concepteurs ont peut-être présenté moins de patience et plus de perplexité vis-à-vis de l'efficacité de la méthode en comparaison à d'autres binômes du groupe 1. Cependant cette hypothèse n'est pas vérifiable par les remarques faites au sein des questionnaires.

Modèle 4, groupe 'formé'

Ce dernier modèle issu du groupe 1 a été structuré avec un grand nombre de groupes et sous-groupes. Cependant, différentes typologies représentant parfaitement la structure hiérarchique du modèle ont été utilisées. De plus deux typologies de notes permettent de différencier les titres principaux des notes plus précises. La particularité de ce modèle est que onze patterns ont été utilisés, identifiés et mis en évidence tout au long du modèle.

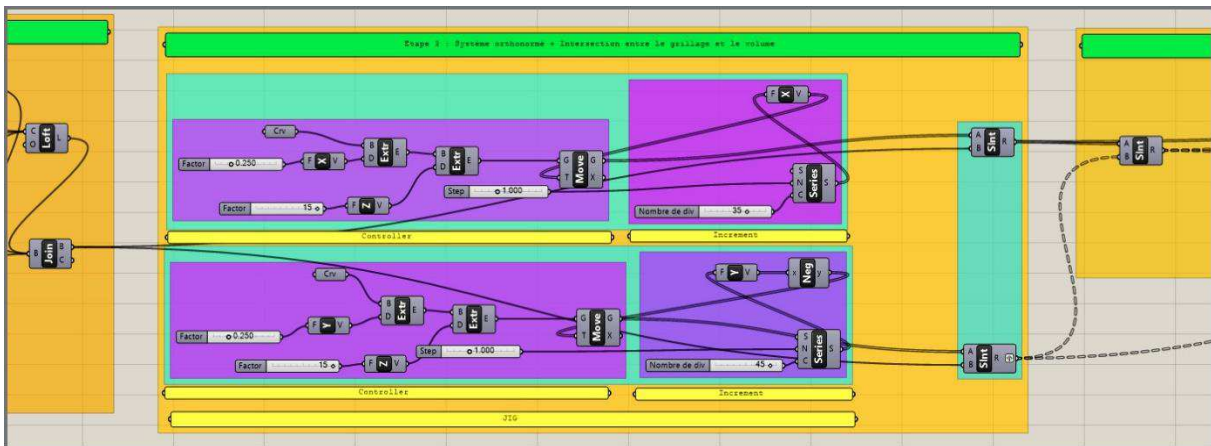


Figure 74 - Modèle 4 - illustration des nombreux groupes et patterns utilisés

Comme dans le cas du binôme 2, les réponses aux questionnaires mettent en évidence la difficulté d'appliquer les patterns car les étudiants n'ont pas eu suffisamment de temps pour les assimiler. Cependant, le binôme ne semble pas avoir perçu les patterns comme une contrainte comme ça a pu être le cas pour le binôme 2.

Modèle 5, groupe 'instinctif'

Ce cinquième modèle a été réalisé par le binôme vedette du groupe 2. On remarque très rapidement que presque aucune caractéristique de la MAPMAP n'y est présente : aucun groupe et aucune note n'ont été appliqués. Cependant quelques objets ont été renommés et aucun paramètre n'a été dupliqué inutilement. En outre, lorsqu'on observe le modèle, on peut deviner une légère structuration des éléments par leur positionnement : les objets identiques sont placés de manière symétrique et tous s'enchaînent de manière chronologique, c'est-à-dire qu'aucun lien ne retourne en arrière.

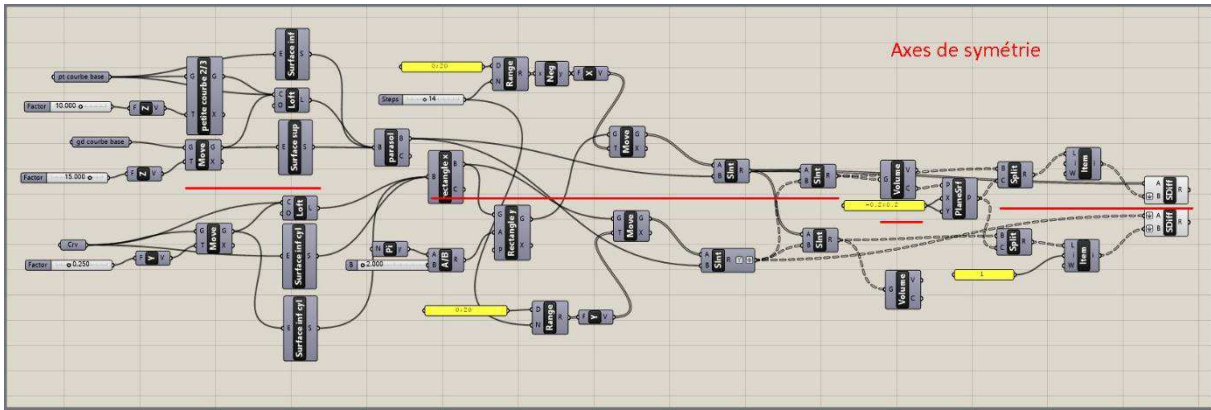


Figure 75 - Modèle 5 - illustration des axes de symétrie de la composition

Enfin, ce modèle est composé de peu d’objets et liens ce qui simplifie sa lecture.

Modèle 6, groupe ‘instinctif’

Comme dans le cas précédent, ce modèle ne présente aucune structure de groupe ni de documentation mais on ressent une structure exprimée à travers le positionnement des objets. À l’opposé du modèle précédent, celui-ci est composé d’un très grands nombre d’objets (dont peu sont renommés), de liens et de paramètres d’entrée.

Une hypothèse explicative pourrait être que ces quantités importantes d’objets et de liens peuvent être à la fois la cause et la conséquence d’un manque de lisibilité de la structure du modèle par les concepteurs. Cette hypothèse est appuyée par deux observations : d’une part, on remarque la présence de panneaux en sortie de certains algorithmes, ce qui peut se traduire comme une volonté du concepteur de mettre en évidence certaines données pour mieux comprendre son modèle ; d’autre part, la disposition des objets a été effectuée de manière peu compacte, ce qui amène la question suivante : le concepteur a-t-il éparpillé les éléments constituant son modèle pour avoir une meilleure visibilité des liens qui les relie ?

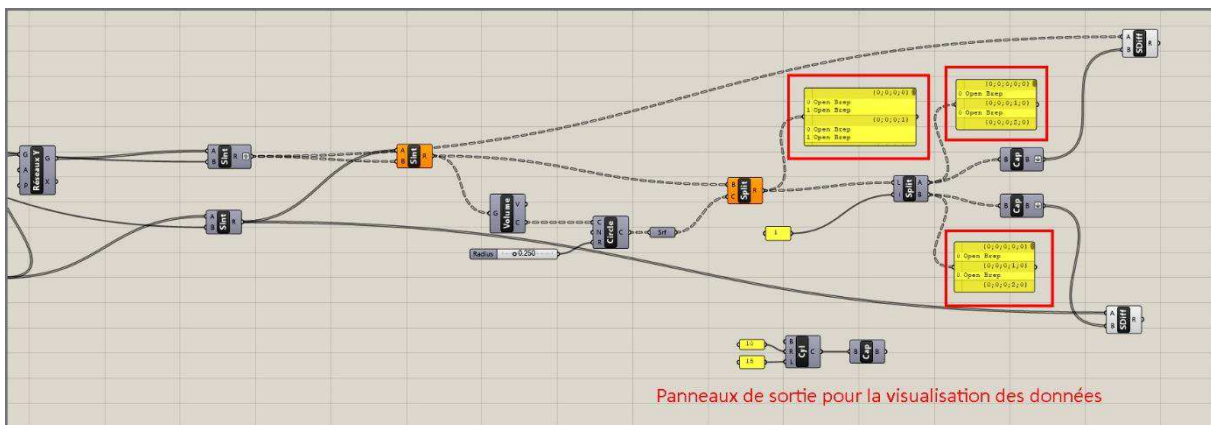


Figure 76 - Modèle 6 - mise en évidence de l'utilisation de panneaux de sortie et de l'étalement des objets

Modèle 7, groupe 'instinctif'

Le modèle 7 a la particularité d'être le modèle du groupe 2 qui a une structure basique faisant écho aux principes de la MAPMAP. En effet, au sein de ce modèle, des étapes de modélisation ont été structurées par la création de groupe et l'insertion de panneaux de titrage. En outre, on remarque que ce modèle est constitué d'une quantité raisonnable d'objets et de liens et qu'aucun paramètre n'a été dupliqué inutilement.

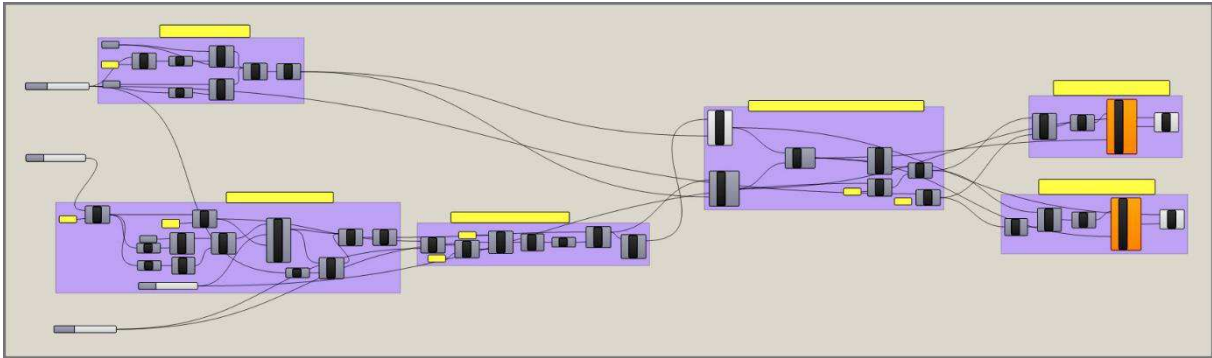


Figure 77 - Modèle 7 - mise en évidence de la structure à travers des groupes et notes de titre

En comparaison aux autres modèles du groupe 2, nous pouvons donc nous poser la question suivante : est-ce la structuration, à l'aide de groupes et de panneaux de documentation, qui a influencé le nombre réduit d'objets et de liens composant le modèle ?

Les questionnaires de ce binôme laissent transparaître la même information : la plus grande difficulté de cet exercice de modélisation était de pouvoir retrouver les objets nécessaires durant la construction. Cette difficulté mise en évidence par ce binôme est très certainement la cause de la création des groupes et de l'ajout des titres.

Q2 - Quelles difficultés avez-vous éventuellement rencontré pour réaliser l'exercice ?
Retrouver les fonctions utiles

Q2 - Quelles difficultés avez-vous éventuellement rencontré pour réaliser l'exercice ?
Retrouver l'implémentation de le nom d'une fonction donnée

Figure 78 - Extraits des questionnaires du binôme 7

Finalement, l'analyse de ce modèle du groupe 2 permet tout particulièrement de mettre en évidence la nécessité et les bénéfices de la MAPMAP.

Modèle 8, groupe 'instinctif'

Au-delà des caractéristique ordinaires du modèle 8 du point de vue du nombre d'objets, de liens, de paramètres, de paramètres dupliqués et d'objets renommés ; ce modèle ne présente

aucune structure liée à la disposition des objets contrairement aux modèles précédents. Cependant, comme dans le modèle 6, des panneaux de visualisation des données ont été insérés en sortie de certains algorithmes et on remarque, une particularité supplémentaire, unique parmi les dix modèles étudiés : six groupes de couleurs différentes ont été utilisés pour contenir certains éléments du modèle.

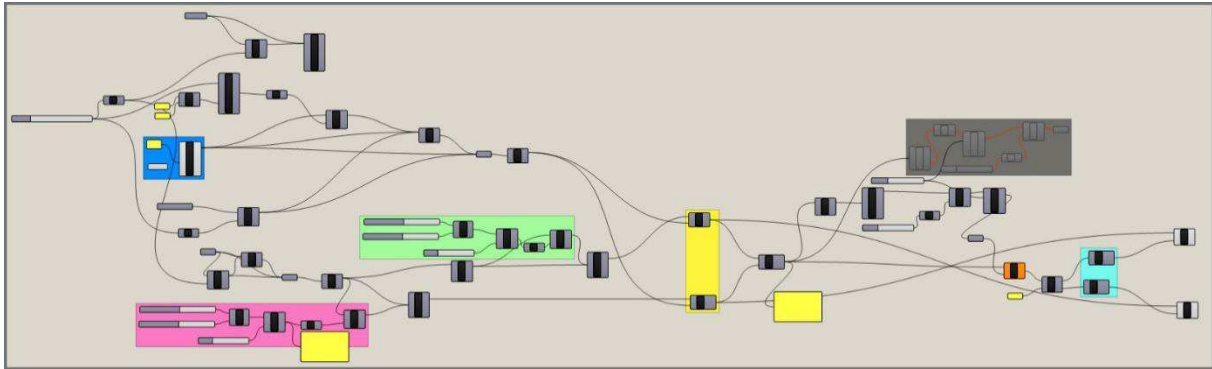


Figure 79 - Modèle 8 - mise en évidence de certains objets par des groupes de couleurs variées

Comme pour le binôme 7, les réponses aux questionnaires mettent en évidence une certaine difficulté pour retrouver les objets voulus dans le modèle déjà construit. On peut donc supposer que ces groupes de couleur ont été placés en vue de mettre en évidence certains éléments clés du modèle pour permettre au concepteur d'y retrouver plus facilement les éléments qui le composent.

Modèle 9, groupe 'instinctif'

Ce dernier modèle est certainement celui dont la communication à un utilisateur externe serait la plus laborieuse. En effet, il est composé d'un grand nombre d'objets et de liens, parmi lesquels on retrouve de nombreux paramètres d'entrée. La plupart d'entre eux ont été dupliqués plutôt que d'augmenter le nombre de liens partant d'un même paramètre. De plus, on ne retrouve aucune structuration du modèle par la position des objets et un seul objet du modèle a été renommé. On remarque tout de même deux caractéristiques qui laissent penser que les concepteurs ont ressenti le besoin de mieux s'y retrouver au sein de leur modèle : d'une part, quelques éléments ont été groupés et, d'autre part, des panneaux de visualisation des données ont été placés en sortie de certains algorithmes.

Finalement, la complexité de ce modèle peut sûrement s'expliquer par le manque de compréhension des principes de modélisation paramétrique en général par les deux concepteurs de ce modèle. En effet, nous découvrons à travers les questionnaires que les deux membres de ce binôme sont les deux étudiants les moins à l'aise avec la modélisation

paramétrique : Leurs réponses correspondent à un niveau 2 alors que les autres étudiants s'autoévaluent à un niveau 3 ou 4.

- Q1 - Suite à la formation dispensée durant les séances de TP de MAN, comment considérez-vous votre niveau de compréhension de la modélisation paramétrique et du logiciel Grasshopper ?
1. Je ne comprends rien et je fais souvent des erreurs
 - ② Je pose beaucoup de question mais j'arrive à suivre
 3. Je construis le modèle au fur et à mesure des explications de l'assistant sans trop de soucis
 4. J'avance plus vite que les explications de l'assistant mais je dois parfois poser des questions
 5. Je fais tout le TP tout seul et pourrait me passer des encadrants
- Q1 - Suite à la formation dispensée durant les séances de TP de MAN, comment considérez-vous votre niveau de compréhension de la modélisation paramétrique et du logiciel Grasshopper ?
1. Je ne comprends rien et je fais souvent des erreurs
 - ② Je pose beaucoup de question mais j'arrive à suivre
 3. Je construis le modèle au fur et à mesure des explications de l'assistant sans trop de soucis
 4. J'avance plus vite que les explications de l'assistant mais je dois parfois poser des questions
 5. Je fais tout le TP tout seul et pourrait me passer des encadrants

Figure 80 - Extraits des questionnaires du binôme 9

Observations d'ordre général

Pour conclure cette analyse des modèles, nous pouvons mettre en évidence des caractéristiques de conception récurrentes dans les modèles et dans le contenu des questionnaires.

En ce qui concerne les modèles réalisés par le groupe 'formé' à la MAPMAP, c'est-à-dire le groupe 1, nous pouvons relever les observations suivantes concernant l'application de cette méthodologie.

- L'utilisation de différentes typologies, au niveau des groupes ou des panneaux de notes, n'est pas optimale que ce soit du point de vue du choix des couleurs ou de la hiérarchisation du modèle.
- Dans le cadre de cet exercice-ci, il n'était pas nécessaire de mettre en œuvre les principes d'encapsulation et de dé-foisonnement des liaisons si bien qu'ils n'ont pas pu être testés et analysés.
- Une remarque récurrente dans les questionnaires est la difficulté de manipuler les objets une fois ceux-ci à l'intérieur des groupes, cependant il s'agit là d'une remarque qui concerne principalement la maniabilité du logiciel Grasshopper et non d'une remarque sur le fond de la méthode.
- Plus d'une fois, il a été relevé que les étudiants avaient des difficultés dans la mise en œuvre des patterns à tel point que, pour certains groupes, les patterns prenaient plutôt

l'apparence d'une contrainte que d'une aide à la modélisation. Sur base de cette expérience, je pense que les patterns de Woodbury nécessitent du temps et de la pratique pour être compris et assimilés par les utilisateurs, si bien que la brève présentation de ces patterns, effectuée dans la formation d'introduction à la MAPMAP, n'était pas suffisante pour permettre aux étudiants de s'en servir dans l'exercice de modélisation qui suivit.

- Enfin les sujets de l'expérience ont généralement cité les mêmes avantages et inconvénients de la MAPMAP. D'une part, ils remarquent généralement que la méthode aide à construire le modèle étape par étape et qu'elle simplifie la relecture si bien que les modifications sont plus facilement effectuables. D'autre part, de nombreux étudiants critiquent le temps 'perdu' à mettre en œuvre la MAPMAP. Cependant, nous verrons par la suite de nos analyses que cette impression ne se justifie pas.

En ce qui concerne les sujets du groupe 2, c'est-à-dire ceux qui ont réalisé l'expérience de manière instinctive, quelques observations générales ressortent également.

- Au sein des modèles, on remarque que chaque groupe a tenté plus ou moins efficacement d'introduire une structure pour organiser son modèle, ou du moins pour s'y retrouver un peu mieux. Parmi ces stratégies que nous avons déjà relevées dans l'analyse de chaque modèle, nous citerons : la création de groupes, l'ajout de titres aux étapes principales, l'organisation des objets par leur positionnement, et l'utilisation de panneaux comme moyen de visualisation du contenu d'un objet algorithmique.
- Dans le contenu des questionnaires, nous avons également relevé à plusieurs reprises la remarque selon laquelle les étudiants ont eu des difficultés à relire leur modèle si bien qu'ils n'arrivaient pas à corriger d'éventuelles erreurs de modélisation ou à retrouver les objets dont ils avaient besoin.

En comparant les résultats des deux groupes, il est intéressant de relever quelques dernières observations.

- Dans les modèles des deux groupes, on remarque souvent la duplication de paramètres de manière inutile. En effet, cela augmente le nombre d'objets et lors de changement le nombre de paramètre à modifier est plus important. Il semble donc que ce caractère implicite du principe de dé-foisonnement des liaisons devrait être mis d'avantage en évidence lors de la présentation de la MAPMAP.
- Dans l'ensemble des modèles également, de nombreux objets ont été renommés (et d'autant plus dans les modèles du groupe 1). Bien que ce soit positif d'après les critères de la MAPMAP, cela soulève des questions qui ne s'étaient jusqu'alors pas posées. En effet, si tous les objets d'un modèle sont renommés, cela ne risque-t-il pas d'en rendre la lecture plus compliquée ? Ne faut-il pas limiter le principe de nomination aux objets de type 'paramètre' uniquement ?

Comme dit précédemment, les sujets du groupe 1 ont critiqué la perte de temps liée à la mise en œuvre de la MAPMAP. De leur côté, les sujets du groupe 2 expriment implicitement avoir perdu du temps lors de la modélisation en raison des difficultés rencontrées à se repérer dans le modèle. Il convient donc de s'interroger sur ce facteur temps et c'est pourquoi nous nous y consacrons entièrement dans la partie suivante de ce travail qui consiste en l'étude des enregistrements audio-visuels des binômes vedettes.

6.1.2. Séquençage des enregistrements

Pour structurer l'analyse des enregistrements audio-visuels de la réalisation de l'expérience par les binômes vedettes des groupes 1 et 2, une analyse 'test' a été réalisée sur base de l'enregistrement de l'expérience zéro. Cette analyse de l'enregistrement a conduit à la mise en place d'un protocole de séquençage composé de trois étapes parallèles. Ainsi l'enregistrement est étudié sous trois aspects : un séquençage global des étapes de modélisation principales, un séquençage détaillé des étapes de construction du modèle, et un relevé des événements ponctuels servant à structurer le modèle, ces événements étant en lien avec la MAPMAP. Sur base de cette structure, l'analyse des enregistrements des binômes 1 et 5 a été réalisée en vue de relever et comparer leurs caractéristiques respectives liées au facteur temps.

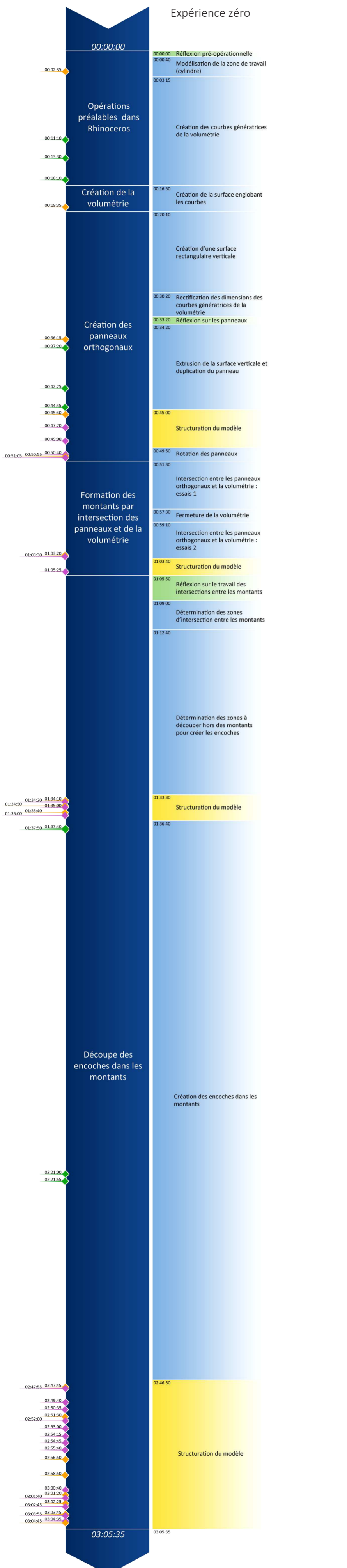


Figure 81 - Analyse temporelle de l'expérience zéro

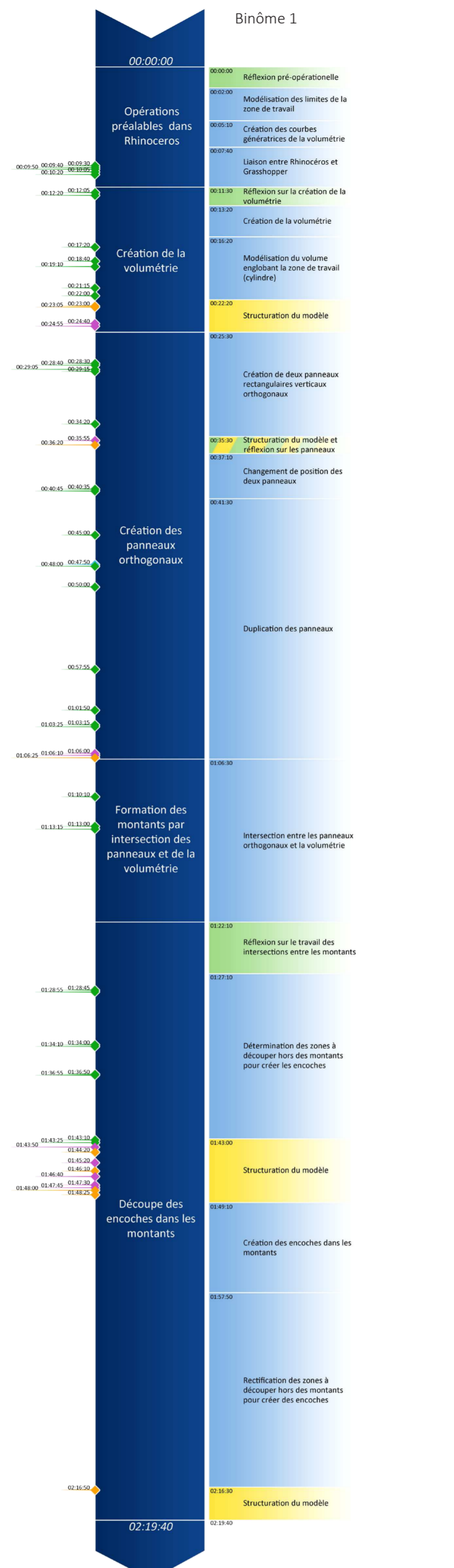


Figure 82 - Analyse temporelle de l'expérience du binôme 1

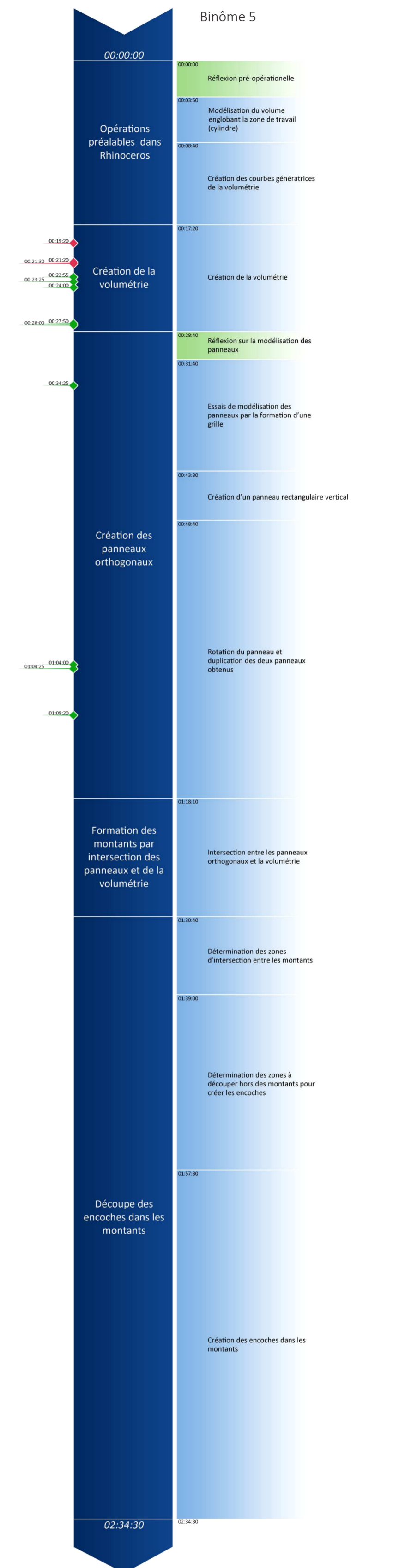


Figure 83 - Analyse temporelle de l'expérience du binôme 5

- Légende**
- hh:mm:ss Temps
 - Étapes principales
 - Étapes de modélisation
 - Étapes de réflexion
 - Étapes de structuration du modèle
 - ◆ Création d'un titre
 - ◆ Création d'un groupe
 - ◆ Nomination d'un objet
 - ◆ Création d'un cluster
 - ◆ Erreur due au manque de nomination

Enregistrement de l'expérience zéro

Si nous analysons, la chronologie de l'expérience zéro, nous remarquons que cinq étapes composent le séquençage global. Par la suite, nous remarquerons que ces cinq étapes sont également identifiables comme structure de la modélisation dans les enregistrements des binômes 1 et 5.

En ce qui concerne, le séquençage des étapes détaillées de construction du modèle, on observe de manière logique que les étapes de réflexion pré-opérationnelle se situent au début des étapes principales tandis que les étapes détaillées de structuration du modèle se situent à la fin des étapes principales. En outre, au niveau de la répartition des événements ponctuels liés à la MAPMAP, les étapes de structuration du modèle sont majoritairement constituées des événements de création de groupes et de documentation du modèle par l'ajout de notes, tandis que les événements de nomination des objets s'effectuent de manière plus équilibrée tout au long de la construction du modèle.

Enfin, nous pouvons remarquer que la dernière étape principale de modélisation, c'est-à-dire l'étape de découpe des encoches dans les montants de la structure, est très longue. Comme nous l'avons exposé lors de l'analyse des modèles, ceci est dû à la difficulté de la mise en œuvre d'opérations booléennes au sein des modèles réalisés sur Grasshopper. Lors de l'expérience concrète, les encadrants étant avertis grâce cette expérience zéro, cette difficulté a été moins marquée, notamment d'un point de vue temporel.

Enregistrement des binômes 1 et 5

Tout d'abord, nous observons que les cinq étapes principales qui permettent de déterminer une structuration dans la construction du modèle, sont identiques pour le modèle de l'expérience zéro et les deux modèles vedettes de l'expérience concrète. Sur base de cette observation, nous pouvons conclure que la MAPMAP est bien une méthodologie qui aide le concepteur à structurer son modèle mais qui ne change pas la manière dont il est pensé.

Ensuite, si l'on observe les caractéristiques du modèle 1 qui a été réalisé par le binôme vedette du groupe 'formé', nous obtenons les mêmes conclusions qu'à travers l'analyse du modèle de l'expérience zéro. En effet, des étapes détaillées de réflexion pré-opérationnelle et de structuration du modèle se trouvent respectivement en début et fin d'étapes principales. À ceci près que l'on constate deux cas particuliers : premièrement, dans la troisième étape principale, on retrouve une étape détaillée à la fois de structuration et de réflexion. Il s'agit d'une démarche d'optimisation du temps consacré à la mise en œuvre de la MAPMAP : l'un des membres du binôme réalisait la structuration du modèle pendant que l'autre membre réfléchissait au déroulement de la suite de la modélisation. Le second cas particulier se situe

dans la cinquième étape principale où l'on observe une étape de structuration du modèle au milieu de l'étape principale. Il s'agit à nouveau d'une optimisation du temps, l'un des membres du groupe s'étant absenté 6 minutes, l'autre membre s'est consacré à l'organisation du modèle en attendant son retour pour continuer l'exercice de modélisation. Finalement, comme pour l'expérience zéro, les étapes de structuration du modèle sont majoritairement composées d'évènement ponctuels de création de groupes et d'ajout de notes tandis que les évènements de nomination des objets ont été réalisés tout au long de la construction du modèle et la création des clusters a été effectuée durant une étape de modélisation.

À présent, concentrons-nous sur l'analyse temporelle de l'enregistrement du binôme 5, c'est-à-dire le binôme vedette du groupe 2 'instinctif'. En comparaison aux deux analyses précédentes, on remarque assez rapidement que peu de phases du travail sont consacrées à la réflexion pré-opérationnelle et qu'aucune n'est consacrée à l'organisation du modèle. Cependant, les deux phases de réflexion pré-opérationnelle sont situées au début des étapes principales de modélisation. Finalement, les éléments ponctuels en lien avec la MAPMAP sont de deux types : des évènements d'erreurs commises par interversion de paramètres d'entrée au début de la modélisation, et, suite à ces erreurs répétées, des évènements de nomination des objets tout au long de la construction du modèle.

Si nous comparons la construction des modèles des deux binômes d'un point de vue temporel, nous remarquons que le binôme 1 (formé) a réalisé son modèle plus rapidement que le binôme 5 (instinctif). Durant, les quatre premières phases de modélisation, les deux binômes ont accordé approximativement le même temps aux étapes de réflexion pré-opérationnelle mais le binôme 1 a consacré du temps à l'organisation du modèle contrairement au binôme 5, hors c'est à la suite de ces deux étapes de structuration du modèle que le binôme 1 prend de l'avance vis-à-vis du binôme deux. Finalement la cinquième étape principale, continuera à creuser l'écart temporel entre les deux binômes. La justification de ses observations est qu'en avançant dans la construction du modèle, celui-ci se complexifiait et la nécessité d'une structure au sein de la représentation symbolique du modèle s'est révélée de plus en plus importante.

En conclusion, malgré les impressions des étudiants du groupe 1, la mise en œuvre de la MAPMAP ne fait pas perdre de temps dans la phase de construction d'un modèle car le temps qu'elle fait gagner, en permettant une relecture plus simple, compense le temps consacré à la réflexion pré-opérationnelle et à l'organisation du modèle.

6.2. Résultats de la seconde expérience

Analysons à présent les résultats de la seconde expérience qui avait pour but l'étude de l'impact de la MAPMAP lors de la communication d'un modèle à un utilisateur externe vis-à-vis de sa conception. Dans le cadre de cette expérience, nous avons moins d'éléments à analyser que dans la précédente, ces éléments ont été présentés au point « 4.3.2. Critères d'analyse », aux pages 68 et 69. Il s'agit, d'une part, de mesures temporelles réalisées durant l'expérience et, d'autre part, de remarques effectuées par les sujets en réponse au questionnaire.

Comme nous l'avons déjà noté, le facteur temporel est important dans l'étude de la pertinence de la MAPMAP. Notre analyse de cette seconde expérience se concentre donc sur ce facteur et les observations effectuées seront complétées et discutées au travers des éléments révélés dans les questionnaires. En outre, contrairement à l'analyse de la première expérience, nous ne tiendrons pas compte des résultats des expériences zéros de cette seconde expérimentation car ceux-ci se réfèrent à d'autres modèles paramétriques.

Pour commencer, un tableau reprenant les différents chronométrages a été dressé, celui-ci contient trois valeurs temporelles : le temps nécessaire au sujet pour changer le premier paramètre, soit l'épaisseur des montants de la structure ; le temps nécessaire pour changer le second paramètre, soit l'entraxe des montants de la structure ; et le temps total.

Information relative au sujet			Chronométrage (mm : ss)		
Sujet	Groupe	Modèle à exploiter	Changement de l'épaisseur	Changement de l'entraxe	Total
1	1	3	<u>07 : 40</u>	02 : 35	10 : 15
2			<u>03 : 52</u>	00 : 23	04 : 15
3			<u>09 : 10</u>	01 : 31	10 : 41
4			<u>06 : 13</u>	00 : 32	06 : 45
5	2	6	<u>03 : 58</u>	<u>08 : 54</u>	12 : 52
6			<u>07 : 45</u>	/	>20min
7			<u>06 : 35</u>	<u>04 : 23</u>	10 : 58
8			<u>14 : 20</u>	01 : 18	15 : 38

Pour analyser ces résultats, commençons par rappeler que l'expérience était limitée à 20 minutes et qu'au-delà de cette durée, l'expérience prenait fin. Nous observons donc que l'une des expérimentations sur le modèle 6, soit le modèle construit sans la MAPMAP, n'a pas abouti.

Ensuite, il convient de préciser que ces modèles sont composés d'opérations booléennes. Ces opérations sont difficiles à calculer pour l'ordinateur, ainsi chaque sujet de l'expérience a été averti de ces complications. Finalement, ces difficultés présentent en réalité un avantage pour notre expérience puisque le sujet ne peut pas trouver le paramètre à changer par tâtonnement, mais qu'il est contraint à comprendre le fonctionnement du modèle par sa représentation symbolique pour trouver les paramètres à faire varier.

Cependant, cet avantage présente également un revers : certains sujets, ayant des difficultés à comprendre le modèle par sa simple représentation symbolique, ont tout de même tenté de faire varier quelques paramètres pour en comprendre le rôle au sein du modèle. En conséquence, certains chronométrages de l'expérience sont composés d'une grande part de temps de calcul de l'ordinateur, en comparaison au peu de temps passé à explorer le modèle. Dans le tableau ci-dessus, les chronométrages empreints de temps de calcul de l'ordinateur ont été souligné. En outre, ces tentatives de changement des valeurs de paramètres ont été plus couramment mises en œuvre par les sujets exploitant le modèle 6 que par ceux exploitant le modèle 3. Ainsi, nous pouvons tout de même conclure que le modèle qui ne fut pas construit à l'aide de la MAPMAP est moins compréhensible pour un utilisateur externe.

Si nous observons, à présent, l'entièreté des valeurs temporelles obtenues au cours de l'expérience, nous remarquons que les chronométrages sont assez variés et disparates. L'une des explications à ce phénomène provient de l'observation précédente, d'après laquelle certains étudiants ont perdu beaucoup de temps en changeant un paramètre qui n'était pas le bon. La seconde explication provient du fait que les étudiants de cette seconde expérience ont des niveaux de compétence en modélisation paramétrique variés. Cette caractéristique se traduit notamment dans les questionnaires soumis aux étudiants à la fin de l'expérience :

Q1 - Suite à la formation dispensée durant les séances de TP de MAN, comment considérez-vous votre niveau de compréhension de la modélisation paramétrique et du logiciel Grasshopper ?

1. Je n'ai jamais rien compris
2. J'arrivais à suivre les TP mais je posais beaucoup de question
3. Je construisais le modèle au fur et à mesure des explications de l'assistant sans soucis mais je n'ai jamais utilisé ce type de logiciel en dehors des TP
4. Depuis les TP j'ai réutilisé quelque fois le logiciel mais je n'ai pas beaucoup approfondi mes connaissances
5. Depuis les TP je me suis intéressé à ce type de logiciel et j'ai développé mes connaissances dans ce domaine

Figure 84 - Extrait du questionnaire du huitième sujet

- Q1 - Suite à la formation dispensée durant les séances de TP de MAN, comment considérez-vous votre niveau de compréhension de la modélisation paramétrique et du logiciel Grasshopper ?
1. Je n'ai jamais rien compris
 2. J'arrivais à suivre les TP mais je posais beaucoup de question
 3. Je construisais le modèle au fur et à mesure des explications de l'assistant sans soucis mais je n'ai jamais utilisé ce type de logiciel en dehors des TP
 4. Depuis les TP j'ai réutilisé quelque fois le logiciel mais je n'ai pas beaucoup approfondi mes connaissances
 5. Depuis les TP je me suis intéressé à ce type de logiciel et j'ai développé mes connaissances dans ce domaine

Figure 85 - Extrait du questionnaire du cinquième sujet

Une dernière remarque d'ordre général et fondée sur l'analyse des résultats temporels, est qu'une subtilité existait au sein des deux modèles : la valeur du paramètre de l'entraxe des montants était déjà identique à celle demandée au sujet lors du paramétrage du modèle si bien qu'elle n'avait pas besoin d'être changée. On remarque grâce au tableau récapitulatif des chronométrages que cette subtilité du modèle a été directement comprise dans le cas du modèle 3 alors que les étudiants exploitant le modèle 6 ont eu plus de difficultés à s'en rendre compte. Cette observation appuie à nouveau l'hypothèse selon laquelle le modèle conçu avec la MAPMAP est plus simple à comprendre et à maîtriser.

Pour terminer cette analyse, résumons les remarques récurrentes formulées par les étudiants. Concernant le modèle 3 réalisé en appliquant la MAPMAP, les sujets mettent en évidence que le modèle est bien ordonné grâce aux groupes, aux couleurs, aux titres et aux noms donnés aux objets. Leurs conseils en vue d'une amélioration de la compréhension du modèle est d'ajouter des titres aux sous-groupes et d'utiliser des noms plus significatifs dans le cas de certains objets renommés. Ces remarques permettent de confirmer la pertinence des principes de la MAPMAP concernant la Stratégie de Division et le Réflexe de Documentation.

Q2 - Quelles difficultés avez-vous rencontré pour réaliser l'exercice ?

Le temps de calcul très long en modifiant un slider

Q3 - Trouvez-vous le modèle bien construit ? Pourquoi ?

Je n'ai pas vraiment regardé la construction du modèle mais directement chercher les sliders avec le nom qui pourrait correspondre à l'épaisseur. Les films de chaque algorithme permet de visualiser rapidement.

Q4 - Auriez-vous des suggestions pour améliorer la compréhension du modèle ?

Si le modèle est destiné à quelqu'un qui doit le fabriquer, utiliser le mot "épaisseur des parois" plutôt que "quadrillage".

Figure 86 - Extrait du questionnaire du troisième sujet

Enfin, concernant le modèle 6, construit de manière instinctive, les étudiants soumis à l'expérience mettent en évidence l'illisibilité de la structure du modèle. D'après ceux-ci le modèle semble bien construit mais il lui manque une organisation traduite visuellement qui permettrait de le comprendre plus rapidement. Leurs conseils, en vue d'améliorer le modèle, sont d'ajouter des groupes, des titres, des notes explicatives, de renommer les paramètres d'entrée et finalement de créer un index reprenant les paramètres d'entrée et leur position au sein du modèle.

Q2 - Quelles difficultés avez-vous rencontré pour réaliser l'exercice ?

Modèle d'écritement linéaire, peu structuré, plein de bulles dans tous les sens, pas d'explications

Q3 - Trouvez-vous le modèle bien construit ? Pourquoi ?

Non, car un "index" pour s'y retrouver, pas d'explications

Q4 - Auriez-vous des suggestions pour améliorer la compréhension du modèle ?

Ajouter des couleurs explicatives, mettre des titres pour chaque étape, mettre des explications pour chaque paramètre et changer avec un "table des matières" des différents paramètres qui dit à quel endroit on se trouve.

Figure 87 - Extrait du questionnaire du septième sujet

6.3. Limites et perspectives de la MAPMAP et son étude

Ce travail de développement de la MAPMAP et de son expérimentation n'est pas une étude figée. L'analyse des résultats des expérimentations démontre que la manière dont la MAPMAP a été évaluée présente certaines limites dues au logiciel, à l'énoncé des expériences et aux sujets des expériences. Cependant, ces résultats ouvrent également de nouveaux horizons : d'une part des perspectives d'évolution de la MAPMAP et d'autre part des perspectives de nouvelles études à mener pour évaluer et améliorer la méthode. Ainsi les paragraphes suivants sont consacrés à la formulation de ces limites et perspectives.

6.3.1. Limites

Le développement de la MAPMAP s'appuie sur le logiciel Grasshopper. Ce point de vue a été choisi dans le but de créer une méthodologie spécifique. Cependant, lors de la réalisation de la première expérience, il a été mis en évidence que la maniabilité du logiciel pouvait limiter la bonne application de la MAPMAP. Par exemple, les groupes du modèleur Grasshopper sont des éléments difficilement modifiables si bien que la mise en œuvre de groupes, en vue d'appliquer la Stratégie de Division, ne doit pas être effectuée trop tôt dans la construction de chaque étape du modèle.

Si nous nous concentrons plus précisément sur la première expérimentation, nous remarquons que celle-ci n'a pas donné le retour escompté. En effet, le modèle qui a fait l'objet de l'exercice de modélisation était adapté au niveau de compétence des sujets de l'expérience. Son degré de complexité étant limité, la création du modèle ne nécessitait pas la mise en œuvre des principes d'Encapsulation et de Dé-foisonnement des Liaisons si bien que la MAPMAP n'a pas pu être entièrement évaluée. En outre, le principe de Description Générique étant plus subjectif, celui-ci n'a pas pu être évalué non plus.

Enfin, les modèles construits lors de la première expérience sont relatifs au niveau de compétence des sujets. En conséquence, le concept de création d'un cluster, qui a tout de même été mis en œuvre par un groupe, ne l'a pas été correctement. De plus, l'application des patterns a été difficilement mise en œuvre car les étudiants n'avaient pas l'expérience nécessaire pour les utiliser tel que Woodbury le préconise dans un cadre éducatif.

6.3.2. Perspectives

Au vu des limites apparues dans cette étude, deux types de perspectives voient le jour : d'une part, des perspectives d'évolution de la MAPMAP et, d'autre part, des perspectives de développement de son étude.

Dans un premier temps, soulignons que la MAPMAP n'est pas une théorie figée. Au contraire, au vu des résultats prometteurs obtenus lors des expériences, il convient de la faire évoluer et de l'enrichir à l'aide des remarques effectuées sur base des résultats des expériences. Voici quelques pistes d'évolution :

- D'abord, le principe de Dé-foisonnement, bien qu'il n'ait pas été mis en œuvre tel quel, mériterait une extension concernant la gestion des paramètres d'entrée. En effet, beaucoup de modèles (des groupes 1 et 2) ont été alimentés par des paramètres d'entrée et leurs duplications. Ces duplications augmentent le nombre d'opérations à effectuer pour changer l'une des valeurs d'entrée. Hors le principe de Dé-foisonnement des liaisons s'applique en vue de rendre le modèle plus lisible lorsque chaque paramètre existe de manière unique au sein du modèle.
- En outre, ce principe de gestion des paramètres d'entrée pourrait entrer en écho avec le principe de Description Générique et s'enrichir du Principe d'Abstraction Stable de la méthode Orientée-Objet. Les modèles réalisés par les sujets de la première expérience sont composés de différentes quantités de paramètres, une explication à ce phénomène est que les binômes ont modélisé plus ou moins d'éléments dans Rhinoceros avant de les intégrer comme entrée dans leur modèle paramétrique. D'après le principe d'abstraction stable, le modèle aurait une plus grande stabilité en réduisant la quantité de paramètres liés au contexte externe, c'est-à-dire liés aux éléments créés dans l'interface de Rhinoceros de manière non-paramétrique.
- Une autre perspective d'évolution de la MAPMAP apparaît au niveau du Réflexe de Documentation. En effet, une remise en question a été émise, suite à l'analyse des modèles, concernant le nombre d'objets à renommer. Il conviendrait de mettre au point d'autres expérimentations pour vérifier qu'un grand nombre d'objets renommés ne met pas la compréhension du modèle en péril. Auquel cas, des limites pourront être fixées pour l'application de ce principe.
- Une idée émise par un sujet de la seconde expérience semble pertinente en tant que potentielle évolution de la MAPMAP. Elle concerne la mise en place d'un index des paramètres dans le cadre de l'application du Réflexe de Documentation. Cet index contiendrait les noms des différents paramètres d'entrée du modèle et leur position au sein du modèle. Ce concept aurait pour objectif de simplifier l'exploitation d'un modèle par un utilisateur externe, il s'agit donc du premier objectif de la MAPMAP.
- Enfin, la méthodologie a été développée dans un cadre spécifique d'application au modeleur Grasshopper. L'une des perspectives d'évolution de la MAPMAP serait d'étendre sa mise en œuvre à d'autres modeleurs paramétriques tels que GenerativeComponents par exemple.

Dans un second temps, remarquons que l'étude que nous avons effectuée de la MAPMAP, a été conditionnée par des questions de temps et de moyens. Cependant, cette étude ayant prouvé l'efficacité de la MAPMAP dans un domaine de recherche restreint, il peut être intéressant

d'envisager de nouvelles études en vue d'analyser son efficacité dans d'autres contextes d'application :

- Une étude pourrait être réalisée sur base d'un exercice de modélisation plus complexe. Cela permettrait, d'une part, d'étudier les principes d'Encapsulation et de Défoisonnement des Liaisons et, d'autre part, une méthode d'analyse du principe de description générique pourrait y être développée.
- Réaliser une étude similaire à celle-ci avec des professionnels, permettrait de situer les limites d'application de la MAPMAP : la méthode est-elle uniquement pertinente dans un cadre éducatif ou l'est-elle également dans un cadre professionnel ?

7. CONCLUSION

À l'origine de cette étude, se trouve la volonté de me consacrer au domaine de la modélisation paramétrique, d'une part, car ce domaine est en pleine émergence dans le milieu architectural et, d'autre part, car cette découverte a éveillé en moi un intérêt particulier.

L'émergence de ce type de modélisation en architecture semble remettre en question les principes de conception habituels et soulève encore de nombreuses questions. Parmi ces dernières, existe notamment la question de la flexibilité d'un modèle paramétrique qui est récurrente chez les auteurs. Cependant, la thématique qui s'imposa à moi pour cette étude est autre. Il s'agit d'une problématique que j'avais moi-même expérimentée au cours de mes créations : les difficultés de lisibilité d'un modèle paramétrique. Finalement, cette problématique fut traduite par la question de recherche suivante.

« Au vu des problèmes de lisibilité et de communication des modèles, est-il possible d'établir une méthodologie d'aide à la programmation en modélisation architecturale paramétrique ? »

En vue de répondre à cette question de recherche, l'étude a évolué à travers six étapes de recherche qui ont été traduites au sein de ce récit par les six premiers chapitres qui le composent. En effet, pour commencer, le premier chapitre a pour vocation d'introduire le thème de la recherche, c'est-à-dire la modélisation paramétrique. Une première partie est consacrée à l'exposition des raisons qui m'ont motivée à réaliser une étude dans ce domaine et une seconde partie se compose d'une présentation succincte de la modélisation paramétrique et de mon positionnement vis-à-vis de sa définition.

Le second chapitre est consacré à la définition plus précise du cadre de la recherche. Ainsi, les paragraphes qui le composent ont permis d'exposer la thématique qui fait l'objet de ce travail, le cadre dans lequel la recherche est effectuée et la méthodologie de recherche qui a été adoptée pour développer cette étude.

L'étape suivante du travail concernait la recherche de méthodes existantes sur base desquelles établir la MAPMAP. Le troisième chapitre fut donc consacré à la présentation de trois méthodes : la méthode de décomposition d'un problème en sous-problème développée par Alexander dans le cadre de la conception architecturale, les principes de design Orientés-Objets exploités dans le cadre de la conception de programmes informatiques et, finalement, la méthode des patterns de Woodbury développée dans le cadre de la conception de modèles paramétriques.

Suite à l'étude de ces trois théories, la Méthodologie d'aide à la Programmation en Modélisation Architecturale Paramétrique (MAPMAP) a pu être établie au sein du quatrième chapitre. Son développement s'est fait autour des axes suivants : le rappel de son objectif, la présentation de ses origines vis-à-vis des trois méthodes pré-exposées et la formulation des cinq principes qui la composent.

Une fois la MAPMAP établie, il convenait de la tester pour pouvoir l'évaluer. Dans cet objectif, deux phases expérimentales ont été réalisées et celles-ci sont décrites au sein du cinquième chapitre. Ces deux expériences concernent : premièrement, la mise en œuvre de la MAPMAP durant la conception d'un modèle paramétrique et, deuxièmement, la communication d'un modèle paramétrique à un utilisateur externe. Chacune de ces expériences a été présentée grâce à la formulation de son objectif, la mise en place du protocole expérimentale et l'établissement des critères qui serviront à effectuer l'analyse de ses résultats.

Le dernier chapitre qui contient la dernière étape de notre étude était destiné à l'analyse qualitative et comparative des résultats des expériences. Cette analyse a été effectuée sur base des critères qui avaient été fixés au cours de l'étape précédente. La discussion des résultats nous conduisit à la formulation des limites et perspectives de la MAPMAP et de son étude.

Grâce aux résultats, nous concluons que la MAPMAP contribue à l'amélioration de la lisibilité d'un modèle. La méthode répond à l'objectif de simplification de la communication d'un modèle paramétrique : un utilisateur externe peut exploiter un modèle qu'il n'a pas conçu plus facilement et plus rapidement, comme nous le prouve la seconde expérience. Remarquons qu'au-delà de cet objectif, la première expérience a prouvé que la MAPMAP contribue à améliorer la lisibilité d'un modèle par son concepteur ce qui réduit également le temps consacré à la construction du modèle paramétrique et diminue les erreurs de modélisation.

Finalement, l'étude de la MAPMAP a été limitée par : la maniabilité du logiciel, le manque de complexité de l'exercice de modélisation constituant la première expérience et le niveau de connaissance de la modélisation paramétrique par les sujets des expériences. Cependant, ces limites permettent d'ouvrir de nouvelles perspectives d'étude comme : étendre les principes de la MAPMAP à d'autres logiciels, mettre en place un exercice plus complexe pour recommencer la première expérience ou encore effectuer celle-ci avec des professionnels. L'analyse des expériences révèle également des perspectives d'évolution de la MAPMAP comme la création d'un nouveau principe concernant la gestion des paramètres d'entrée se traduisant notamment au sein du Réflexe de Documentation par la création d'un index des paramètres ou encore comme la remise en cause de certains aspects du Réflexe de Documentation.

8. RÉFÉRENCES

Plan du chapitre :

8.1. BIBLIOGRAPHIE.....	100
8.2. WEBOGRAPHIE.....	101
8.3. LISTE DES FIGURES.....	102

8.4. Bibliographie

- Aish, R., & Woodbury, R. (2005). Multi-level Interaction in Parametric Design. In A. Butz et al., (éd.) *Smart Graphics* (Vol. 3638, p. 151 - 152). Springer Berlin / Heidelberg. Consulté à l'adresse https://link.springer.com/chapter/10.1007%2F11536482_13 , le 06/05/2017.
- Alexander, C. (1971). *De la Synthèse de la Forme*, essai. Dunod.
- Alexander, C. et al. (1977). *A Pattern Language : Towns, Buildings, Construction*. Oxford University Press, New York.
- de Boissieu, A. (2010). *Modélisation paramétrique en conception architecturale : Caractérisation des opérations cognitives de conception pour une pédagogie*. Ecole Nationale Supérieur d'Architecture de Paris la Vilette, Paris.
- Burry, J. (2007). Mindful Spaces : Computational Geometry and the Conceptual Spaces in which Designers Operate. *International Journal of Architectural Computing* (Vol 5(4), p. 611-624).
- Cerri, A., & Estival, P. (2005). Alan Kay, le roi de l'informatique avec trois couronnes. Consulté à l'adresse : <https://hal.archives-ouvertes.fr/hal-00804149> , le 06/05/2017
- Coad, P., & Yourdon, E. (1991). *Object-Oriented Design*. Yourdon Press Computing Series.
- Dathan, B., & Ramnath, S. (2010). *Object-Oriented Analysis and Design*. Springer Science and Business Media, & Universities Press.
- Davis, D. (2013). *Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture*. School of Architecture and Design, College of Design and Social context, RMIT University.
- Devaux, R. (2016). *Étude du Workflow de Projet d'Architecture Paramétrique : Identification des composantes récurrentes, perturbatrices ou novatrices*. Université de Liège, Liège.
- Hudson, R. (2010). *Strategies for parametric design in architecture : An application of practice led research*. University of Bath, Department of Architecture and Civil Engineering.
- Leach, N. (2014). *Parametrics Explained. Next Generation Building* (Vol 1, p. 33-42). University of Southern California, Harvard GSD.
- Sutherland, I. (2003). *Sketchpad: A man-machine graphical communication system. Technical Report (Nbr 574)*. University of Cambridge, Computer Laboratory.
- Weisberg, D. (2008). *The Engineering Design Revolution : The People, Companies and Computer Systems that Changed Forever the Practice of Engineering*. Chapter 16 : Parametric Technology Corporation. Consulté à l'adresse : <http://www.cadhistory.net./toc.htm> , le 06/05/2017.
- Weisstein, E. (2003). *CRC Concise Encyclopedia of Mathematics*. Second edition. Boca Raton : Chapman & amp, Hall/CRC.
- Woodbury, R. (2010). *Elements of Parametric Design*. Routledge.

8.5. Webographie

[Inhabitat] <http://inhabitat.com/new-photos-of-sevilles-gorgeous-metropol-parasol/> , consulté le 01/03/2017

[PublicSpace] <http://www.publicspace.org/en/works/g315-metropol-parasol>, consulté le 01/03/2017

[SagradaFamilia] <http://www.sagradafamilia.org/index.html?p=155>, consulté le 05/05/2017

[PatternLanguage] <http://www.patternlanguage.com>, consulté le 12/05/2017

[LeMonde] http://www.lemonde.fr/technologies/article/2011/05/27/la-programation-pour-les-non-programmeurs_1528458_651865.html , consulté le 24/05/2017

[Docs.Oracle] <https://docs.oracle.com/javase/tutorial/java/concepts> , consulté le 24/05/2017

[UncleBob] <http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>, consulté le 24/05/2017

[CodeProject] <https://www.codeproject.com/Articles/567768/Object-Oriented-Design-Principles>, consulté le 24/05/2017

[ProgrammeCoursULg] <http://progcours.ulg.ac.be/cocoon/cours/ARCH0017-4.html>, consulté le 29/05/2017

[VisualBasic] <https://www.visualstudio.com/fr/vs/> , consulté le 06/06/2017

[GenerativeComponents]

https://communities.bentley.com/products/products_generativecomponents/w/generative_components_community_wiki/5798 , consulté le 06/06/2017

8.6. Liste des figures

FIGURE 1 - SKETCHPAD (SUTHERLAND, 2003).....	7
FIGURE 2 - PLAN D'UNE PARTIE DE LA SAGRADA FAMILIA (WWW.SAGRADAFAMILIA.ORG).....	9
FIGURE 3 – SCHÉMA DE DIMENSIONNEMENT DES COLONNES DE LA SAGRADA FAMILIA DE BARCELONNE (WWW.SAGRADAFAMILIA.ORG) .	9
FIGURE 4 - CRÉATION D'UN CERCLE DANS RHINOCEROS : CHOIX DU CENTRE.....	10
FIGURE 5 - CRÉATION D'UN CERCLE DANS RHINOCEROS : CHOIX DU RAYON.....	10
FIGURE 6 - CRÉATION D'UN CERCLE AVEC GRASSHOPPER PAR LA CRÉATION D'UN CODE DE PROGRAMMATION VISUEL	11
FIGURE 7 - REPRÉSENTATIONS SYMBOLIQUE ET EXPLICITE DANS GRASSHOPPER	11
FIGURE 8 - MODÈLE PARAMÉTRIQUE RÉALISÉ SUR GRASSHOPPER DANS LE CADRE D'UN ATELIER DE CONCEPTION. ON REMARQUE UNE CERTAINE ORGANISATION DU CODE PAR LE POSITIONNEMENT DES ÉLÉMENTS MAIS IL EST RENDU ILLISIBLE PAR LE NOMBRE D'OBJETS ET DE LIENS QUI EN EMPÊCHENT LA COMPRÉHENSION.	15
FIGURE 9 - INTERFACE DE PROGRAMMATION TEXTUELLE DU LOGICIEL VISUALBASIC (HTTPS://WWW.VISUALSTUDIO.COM)	18
FIGURE 10 - INTERFACE DE PROGRAMMATION VISUELLE DU LOGICIEL GENERATIVECOMPONENTS (HTTPS://COMMUNITIES.BENTLEY.COM)	18
FIGURE 11 – INTERFACE DE PROGRAMMATION VISUELLE DU MODELEUR GRASSHOPPER, PLUGIN DU LOGICIEL RHINOCÉROS	18
FIGURE 12 - PROBLÈME DE COMPOSITION TYPIQUE AVEC DES EXIGENCES ET DES INTERACTIONS ENTRE ELLES (ALEXANDER, 1971, P. 2)	24
FIGURE 13 - REPRÉSENTATION D'UN SYSTÈME DE VARIABLES D'INADAPTATION INTERCONNECTÉES (ALEXANDER, 1971, P. 35).....	25
FIGURE 14 - DÉCOMPOSITION DU SYSTÈME EN SOUS-SYSTÈME (ALEXANDER, 1971, P. 36).....	26
FIGURE 15 - DÉCOMPOSITION DU SYSTÈME SELON DES CONCEPTS ARBITRAIRES (ALEXANDER, 1971, P. 55)	26
FIGURE 16 – ENSEMBLE D'EXIGENCES M STRUCTURÉ EN SOUS-ENSEMBLE (ALEXANDER, 1971, P.68).....	27
FIGURE 17 - DIAGRAMME CONSTRUCTIF DE RÉALISATION DU PROGRAMME (ALEXANDER, 1971, P. 78).....	27
FIGURE 18 - SCHÉMA DE STRUCTURE D'UN PATTERN	29
FIGURE 19 – EXEMPLE DE MODÈLE PARAMÉTRIQUE CONSTRUIT DE MANIÈRE INSTINCTIVE.	41
FIGURE 20 – EXEMPLE DE MODÈLE PARAMÉTRIQUE CONSTRUIT EN APPLIQUANT LA MAPMAP	41
FIGURE 21 - GRAPHE D'UN MODÈLE COMPOSÉ D'OBJETS ET DE LIENS	45
FIGURE 22 - GRAPHE D'UN MODÈLE DIVISÉ EN MODULES.....	45
FIGURE 23 – HIÉRARCHIE DES MODULES PRIMAIRES ET SECONDAIRES D'UN MODÈLE.....	45
FIGURE 24 – GRAPHE D'UN MODÈLE DIVISÉ EN MODULES PRIMAIRES ET SECONDAIRES.....	45
FIGURE 25 – DÉCOMPOSITION DU MODÈLE DU GHERKIN EN TROIS ÉTAPES DE CONSTRUCTION	46
FIGURE 26 – EXEMPLE DE MODÈLE DU GHERKIN TEL QUE RÉALISÉ AU COURS DE MAN	47
FIGURE 27 – EXEMPLE DE MODÈLE DU GHERKIN STRUCTURÉ EN MODULES	47
FIGURE 28 – EXEMPLE DE MODÈLE DU GHERKIN RÉALISÉ AU COURS DE MAN	48
FIGURE 29 – EXEMPLE DE MODÈLE DU GHERKIN STRUCTURÉ EN MODULES	49
FIGURE 30 – EXEMPLE DE MODÈLE DU GHERKIN STRUCTURÉ ET DOCUMENTÉ.....	49
FIGURE 31 - CLUSTER CONTENANT LA MODÉLISATION DE LA VOLUMÉTRIE DANS LE CADRE DU MODÈLE DU GHERKIN.....	50
FIGURE 32 - DUPLICATION DU CLUSTER CONTENANT LA MODÉLISATION DE LA VOLUMÉTRIE DANS LE CADRE DU MODÈLE DU GHERKIN .	50
FIGURE 33 – REPRÉSENTATION EXPLICITE DE LA DUPLICATION DU CLUSTER DE LA VOLUMÉTRIE	50
FIGURE 34 - EXEMPLES DE MODÉLISATION AVEC UN FOISONNEMENT DE LIENS TROP IMPORTANT.....	51
FIGURE 35 – GRAPHE D'UN MODÈLE AVEC UN IMPORTANT COUPLAGE	51
FIGURE 36 - MISE EN ÉVIDENCE DU FOISONNEMENT DE LIENS PROVENANT DE DEUX PARAMÈTRES	51
FIGURE 37 - DÉ-FOISONNEMENT DES LIENS SUPERFLUS PAR LA CRÉATION D'AVATARS (TRIANGLES)	51
FIGURE 38 – EXEMPLE DE MODÈLE DE TRAITEMENT D'UNE SURFACE PAR UN VORONOÏ RÉALISÉ AU COURS DE MAN	52
FIGURE 39 - MORCEAU DE MODÈLE PRÉSENTANT UN GRAND NOMBRE DE LIENS QUI PERTURBENT LA LECTURE DU MODÈLE	52
FIGURE 40 - MORCEAU DE MODÈLE DONT LES LIAISONS ONT ÉTÉ DÉ-FOISONNÉES PAR LA CRÉATION DE DEUX AVATARS AU SEIN DU SECOND MODULE	52
FIGURE 41 – EXEMPLE D'UN MORCEAU DE MODÈLE DU GHERKIN DONT L'OBJECTIF EST LA CRÉATION D'UN CHÂSSIS SUR BASE D'UNE FORME POLYGONALE DONT LE NOMBRE DE CÔTÉ A ÉTÉ PARAMÉTRÉ À 3	53
FIGURE 42 – EXEMPLE DE MODÈLE PARAMÉTRIQUE 1 RÉALISÉ AU COURS DE MAN : LE GHERKIN DE N. FOSTER.....	55
FIGURE 43 – EXEMPLE DE MODÈLE PARAMÉTRIQUE 2 RÉALISÉ AU COURS DE MAN : TRAITEMENT DE SURFACE PAR DU VORONOÏ	55
FIGURE 44 – EXEMPLE DE MODÈLE PARAMÉTRIQUE 2, SECONDE VERSION, RÉALISÉ AU COURS DE MAN : TRAITEMENT DE SURFACE PAR DU VORONOÏ (3D).....	56
FIGURE 45 – EXEMPLE DE MODÈLE PARAMÉTRIQUE 3 RÉALISÉ AU COURS DE MAN : TRAITEMENT DE SURFACE MODULAIRE PAR MAILLAGE	56
FIGURE 46 - DEUX PREMIÈRES SLIDES DU COURS D'INTRODUCTION À LA MAPMAP, PRÉSENTÉ AU GROUPE 2	57

FIGURE 47 - METROPOL PARASOL DE SÉVILLE (HTTP://INHABITAT.COM)	57
FIGURE 48 - MODÈLE À RÉALISER DURANT L'EXPÉRIENCE.	57
FIGURE 49 – PREMIÈRE PAGE DE L'ÉNONCÉ DISTIBUÉ AUX ÉTUDIANTS	58
FIGURE 50 - MODÈLE PHYSIQUE À DISPOSITION DES ÉTUDIANTS DURANT L'EXPÉRIENCE	58
FIGURE 51 – EXEMPLE DE MODÈLE PARAMÉTRIQUE 4 RÉALISÉ AU COURS DE MAN 2015 : PAVILLON EN FORME DE COQUILLAGE.....	59
FIGURE 52 – BINÔME DE L'EXPÉRIENCE ZÉRO CONSTRUISANT LE MODÈLE SUR GRASSHOPPER À L'AIDE DE LA MAPMAP	60
FIGURE 53 - ENREGISTREMENT CAMÉRA D'UN ÉCRAN SECONDAIRE	60
FIGURE 54 - DISPOSITION DE L'EXPÉRIENCE ZÉRO	61
FIGURE 55 - RÉPONSE À LA QUESTION 8 DU QUESTIONNAIRE COMPLÉTÉ PAR LES SUJETS DE L'EXPÉRIENCE ZÉRO.....	61
FIGURE 56 - SCHÉMA DE RÉPARTITION SPATIALE DES BINÔMES	64
FIGURE 57 - GROUPE 1, BINÔMES CLASSIQUES	64
FIGURE 58 – GROUPE 1, BINÔME VEDETTE	64
FIGURE 59 – GROUPE 2, BINÔMES CLASSIQUES.....	64
FIGURE 60 – GROUPE 2, BINÔME VEDETTE	64
FIGURE 61 - CONTEXTE D'EXPÉRIMENTATION : BINÔME VEDETTE, GROUPE 1.....	64
FIGURE 62 - CONTEXTE D'EXPÉRIMENTATION : BINÔME VEDETTE, GROUPE 2.....	64
FIGURE 63 - QUATRE MODÈLES PARAMÉTRIQUES RÉALISÉS DANS LE CADRE DU COURS DE MAN 2015	68
FIGURE 64 - MODIFICATIONS À APPORTER AU MODÈLE	69
FIGURE 65 - DISPOSITION DE L'EXPÉRIENCE ZÉRO SERVANT À VALIDER L'EXPÉRIENCE DE COMMUNICATION	70
FIGURE 66 - PARTIE DU MODÈLE 1 CONTENANT DEUX CLUSTERS.....	71
FIGURE 67 - CONTENU DU CLUSTER CRÉÉ DANS LE MODÈLE 1.....	71
FIGURE 68 - MORCEAU DU MODÈLE 8 MONTRANT QUE LA BASE DES MONTANTS A ÉTÉ DIRECTEMENT CRÉÉE DANS RHINOCEROS	71
FIGURE 69 - RÉALISATION DE LA SECONDE EXPÉRIENCE.....	72
FIGURE 70 - MODÈLE 0 - ERREURS DANS L'APPLICATION DE LA MAPMAP	77
FIGURE 71 - MODÈLE 1 - CONTENU DES CLUSTERS	78
FIGURE 72 – EXTRAITS DES QUESTIONNAIRES DU BINÔME 2	78
FIGURE 73 - MODÈLE 3 - TYPOLOGIES PEU COMPRÉHENSIBLES ET NOMBREUX PARAMÈTRES DUPLIQUÉS INUTILEMENT	79
FIGURE 74 - MODÈLE 4 - ILLUSTRATION DES NOMBREUX GROUPES ET PATTERNS UTILISÉS	80
FIGURE 75 - MODÈLE 5 - ILLUSTRATION DES AXES DE SYMÉTRIE DE LA COMPOSITION	81
FIGURE 76 - MODÈLE 6 - MISE EN ÉVIDENCE DE L'UTILISATION DE PANNEAUX DE SORTIE ET DE L'ÉTALEMENT DES OBJETS	81
FIGURE 77 - MODÈLE 7 - MISE EN ÉVIDENCE DE LA STRUCTURE À TRAVERS DES GROUPES ET NOTES DE TITRE	82
FIGURE 78 - EXTRAITS DES QUESTIONNAIRES DU BINÔME 7.....	82
FIGURE 79 - MODÈLE 8 - MISE EN ÉVIDENCE DE CERTAINS OBJETS PAR DES GROUPES DE COULEURS VARIÉES	83
FIGURE 80 - EXTRAITS DES QUESTIONNAIRES DU BINÔME 9.....	84
FIGURE 81 - ANALYSE TEMPORELLE DE L'EXPÉRIENCE ZÉRO	87
FIGURE 82 - ANALYSE TEMPORELLE DE L'EXPÉRIENCE DU BINÔME 1	87
FIGURE 83 - ANALYSE TEMPORELLE DE L'EXPÉRIENCE DU BINÔME 5	87
FIGURE 84 - EXTRAIT DU QUESTIONNAIRE DU HUITIÈME SUJET	91
FIGURE 85 - EXTRAIT DU QUESTIONNAIRE DU CINQUIÈME SUJET	92
FIGURE 86 - EXTRAIT DU QUESTIONNAIRE DU TROISIÈME SUJET	92
FIGURE 87 - EXTRAIT DU QUESTIONNAIRE DU SEPTIÈME SUJET	93