

Ecole Nationale Supérieure d'Architecture
de Paris La Villette
ENSAPLV

Mémoire dans le cadre du séminaire

Savoir des Activités de Projet Instrumenté

SAPI

Domaine d'étude

Concevoir et Construire l'Architecture

CCA

**Comment la perception des limites techniques dans
l'utilisation des outils de modélisation numérique oriente
la production formelle en architecture**

Présenté par :

LASMI Faycal

Mémoire dirigé par :

Mme. Anne Tüscher-Dokic

Mr. Joaquim Silvestre

Soutenance : 13 Janvier 2026

Ce mémoire est l'aboutissement de nombreux efforts, mais il n'aurait jamais pu voir le jour sans le soutien précieux des personnes qui m'entourent.

Je le dédie avant tout à ma grand-mère, qui nous a quittés le 28 décembre 2025. Sa présence et son souvenir m'accompagnent profondément. Je le dédie également à mes parents, éloignés géographiquement mais toujours proches par leurs encouragements constants et leur confiance inébranlable. Leur soutien, même à distance, m'a donné la force et la détermination d'aller jusqu'au bout.

À mon frère, lui aussi loin, mais toujours présent à travers ses mots et son soutien.

À mes deux tantes, pour leur bienveillance, leur écoute et leur soutien tout au long de ce parcours.

Et enfin à mes amis, pour leur présence, leur patience et les moments partagés durant cette période importante.

REMERCEIMENTS

Ce mémoire est l'aboutissement d'un long travail que je n'aurais pu réaliser seul.

Je tiens à remercier chaleureusement mes encadrants, Joaquim Silvestre et Anne Tüscher, pour leur expertise, leur patience et la justesse de leurs orientations tout au long de cette année. Merci de m'avoir poussé à questionner mes propres outils de conception.

Mes remerciements les plus affectueux vont à ma famille. À mes parents, qui ont tout fait pour ma réussite et qui sont ma source d'inspiration quotidienne. Merci de m'avoir appris à ne jamais abandonner. À ma famille élargie et mes amis, pour votre écoute et votre soutien moral dans les moments de doute.

Enfin, je rends grâce à Dieu de m'avoir accordé la santé et la lucidité nécessaires pour accomplir ce travail, et d'avoir mis sur ma route les bonnes personnes pour m'accompagner.

RÉSUMÉ

La généralisation des outils de modélisation numérique (CAO/BIM) a profondément transformé la phase conceptuelle en architecture, imposant une communication technique entre l'imaginaire de l'architecte et la forme produite. Ce mémoire interroge l'influence de cette relation outil/concepteur sur les possibilités de production formelle, en se focalisant spécifiquement sur le biais de compréhension et le phénomène psychologique d'incompétence apprise (*learned helplessness*).

L'hypothèse centrale postule que les difficultés techniques perçues, amplifiées par l'affordance restrictive des logiciels, créent un « Golfe d'Exécution » (Norman) infranchissable pour l'utilisateur. Face à cet obstacle, le concepteur développe une forme d'auto-censure cognitive, restreignant son exploration formelle à ce qu'il se sent capable de modéliser avec l'outil, au détriment de ses intentions initiales.

Pour vérifier cette hypothèse, cette recherche adopte une méthodologie mixte. D'une part, une enquête quantitative par le moyen d'un questionnaire qui vise à mesurer la perception des contraintes logicielles ressenties par les utilisateurs et leur impact sur le flux créatif. D'autre part, un protocole expérimental de conception comparée (croquis manuel vs modélisation directe) analyse l'évolution morphologique d'un même objet architectural (un escalier). Les résultats visent à quantifier l'écart entre l'intention et l'action (*Intention-Action Gap*), mettant en lumière une tendance à la simplification géométrique et une standardisation des solutions formelles induites par la peur de la complexité technique pendant la phase de modélisation.

Mots-clés : Conception architecturale, Modélisation numérique, Incompétence apprise,, Créativité, Affordance, Intention-Action Gap.

ABSTRACT

The widespread adoption of digital modeling tools (CAD/BIM) has profoundly transformed the conceptual phase in architecture, imposing a technical mediation between the architect's imagination and the produced form. This thesis investigates the influence of this mediation on creativity, focusing specifically on the understanding bias and the psychological phenomenon of learned helplessness.

The central hypothesis posits that perceived technical difficulties, amplified by the restrictive affordance of software interfaces, create a "Gulf of Execution" (Norman) that students find difficult to bridge. Faced with this obstacle, the designer develops a form of cognitive self-censorship, restricting their formal exploration to solutions anticipated as feasible by the tool, often at the expense of their initial intentions.

To test this hypothesis, this research adopts a mixed-methods approach. First, a quantitative survey measures the perception of software limitations and their impact on the creative flow. Second, a comparative design experiment (hand sketching vs. direct modeling) analyzes the morphological evolution of a single architectural object (a staircase). The results aim to quantify the Intention-Action Gap, highlighting a tendency towards geometric simplification and the standardization of formal solutions induced by an apprehension of technical complexity.

Keywords: Architectural Design, Digital Modeling, Learned Helplessness,, Creativity, Affordance, Intention-Action Gap.

Table des matières

1. INTRODUCTION	p.11
2. PROBLÉMATIQUE ET HYPOTHÈSES	p. 11
• 2.1. Problématique	p. 13
• 2.2. Hypothèses de recherche	p. 13
3. MÉTHODOLOGIE (Présentation générale)	p.14
4. ÉTAT DE L'ART	p.15
• 4.1. La conception architecturale	p. 15
• 4.2. La créativité et le numérique	p. 16
• 4.3. Le rôle de l'outil (Biais, Affordance, Intention action gap)	p. 20
5. MÉTHODOLOGIE DE RECHERCHE (Protocole détaillé)	p.23
• 5.1. Enquête quantitative (Questionnaire)	p. 23
• 5.2. Expérimentation (Croquis vs Modélisation)	p. 34
• 5.3. Critères d'analyse et grille d'évaluation	p. 26
6. ANALYSE DES RÉSULTATS	p. 27
• 6.1. Données issues du questionnaire.....	p. 27
• 6.2. Analyse des projets de l'exercice comparatif.....	p. 27
• 6.3. Comparaison des tendances (Maîtrise vs Forme)	p. 28
7. PERSPECTIVES ET OUVERTURES	p. 30
8. CONCLUSION	p. 31
9. ANNEXES	p.35
10. BIBLIOGRAPHIE	p.35

Introduction

L'architecture est une discipline caractérisée par deux grandes étapes. La première repose sur une dimension abstraite, appelée la phase conceptuelle, au cours de laquelle "on pense le projet". La seconde, plus concrète, correspond à la mise en œuvre, ou phase de construction.

C'est cette première étape conceptuelle qui nous intéresse ici. C'est à ce moment que l'architecte commence à réfléchir, à faire des essais, à imaginer différentes formes, en s'appuyant sur les contraintes du site, le programme, mais aussi son expérience. Traditionnellement, l'architecte utilisait des esquisses ou des dessins rapides. Ces outils permettaient d'explorer facilement des idées, car ils sont simples à modifier, à effacer ou à abandonner.

Mais aujourd'hui, avec la généralisation des logiciels de modélisation (comme Rhinoceros, Revit ou SketchUp), cette phase de création a changé. Ces outils sont utilisés de plus en plus tôt, dès la phase embryonnaire du projet. Or, ils fonctionnent avec leurs propres règles : ils demandent une maîtrise des logiques informatiques, une méthode rigoureuse et des connaissances techniques précises.

Cette utilisation n'étant pas neutre. Comme le soulignaient déjà Foster & Partners (2006), les architectes ont tendance à se conformer aux possibilités offertes par les logiciels notamment selon ce que suggèrent les interfaces, se laissant contraindre par des bibliothèques d'éléments prédéterminés. Une question centrale émerge alors : l'architecte adapte-t-il ce qu'il imagine en fonction de ce qu'il sait faire avec le logiciel ? Limite-t-il son champ des possibles pour rester dans ce qu'il maîtrise techniquement ?

Pour répondre à cette interrogation, il est nécessaire d'aborder certains mécanismes psychologiques qui entrent en jeu :

Le premier étant le biais de compréhension, c'est une erreur de jugement dans laquelle l'utilisateur ne juge pas le logiciel sur ce qu'il permet réellement de faire, mais sur ce qu'il pense semble permettre de faire à travers sa propre perception de l'outil et à travers ce que suggère l'interface logicielle. Par exemple si une fonction n'est pas visible ou intuitive (elle peut être visible mais l'icône qui la représente n'est pas très explicite), l'utilisateur suppose souvent qu'elle n'existe pas. Il restreint ainsi son "pensable" à ce qu'il croit être le "possible"

Par la suite l'utilisateur se trouve dans un état de résignation appelé l'incompétence apprise. Face à sa perception de l'outil, l'utilisateur finit par intégrer l'idée qu'il ne parviendra pas à modéliser une forme donnée. Ce n'est pas nécessairement qu'il ne sache pas le faire, mais qu'il renonce à essayer par peur d'incapacité à cause de sa perception et de sa maîtrise ou degré de connaissance de l'outil. Ce qui pousse à une forme d'auto-censure préventive qui limite le champ des possibles.

On peut alors se poser la question suivante : est-ce que l'architecte adapte ce qu'il imagine en fonction de ce qu'il sait faire avec le logiciel ? Est-ce qu'il limite son champ de possibilités pour rester dans ce qu'il maîtrise techniquement ?

PROBLÉMATIQUE :

Ce mémoire a pour objectif de soulever la tension entre liberté d'intention et contrainte induite par la compréhension de modélisation numérique.

Ce biais renvoie à la manière dont un concepteur interprète les possibilités et les limites d'un outil, non pas selon sa maîtrise réelle, mais selon la représentation qu'il s'en fait. Autrement dit, il ne s'agit pas d'une incompétence technique, mais d'une auto-limitation, où l'utilisateur restreint inconsciemment son champ d'exploration formelle à ce qu'il pense possible dans le cadre du logiciel.

Cette influence est d'autant plus complexe qu'elle demeure souvent invisible, voire inconsciente : il est difficile de distinguer si une forme est choisie pour sa pertinence architecturale ou parce qu'elle semble plus aisée à modéliser. Pourtant, comprendre ce mécanisme est essentiel pour saisir l'impact réel de la perception de l'outil sur la créativité et la production formelle en architecture.

On peut alors se poser la question suivante :

Dans quelle mesure l'incompétence apprise, induite par le biais de la compréhension des outils de modélisation, amène-t-elle les concepteurs à abandonner certaines intentions formelles pour ne produire que ce qu'ils perçoivent comme "faisable" ou modélisable dans un logiciel ?

HYPOTHÈSES :

Bien qu'il soit difficile d'identifier avec précision ce qui influence les choix formels d'un concepteur, réduire ce processus à la seule capacité technique de modélisation reviendrait à simplifier à tort la complexité de la conception architecturale. En effet, le projet est influencé par de nombreux facteurs : le bagage théorique, les références culturelles, les contraintes du site, etc.

Notre hypothèse part du principe que l'utilisateur possède les bases nécessaires à la conception. L'enjeu n'est donc pas ici de mesurer sa maîtrise technique, mais d'observer comment l'outil agit comme un filtre sur ses intentions.

Ainsi, l'hypothèse de cette recherche est la suivante :

Le biais de compréhension d'un outil de modélisation numérique conduit le concepteur à restreindre son exploration formelle à des solutions qu'il perçoit comme réalisables dans le cadre fonctionnel de cet outil, réduisant ainsi la diversité et la complexité des formes produites.

La recherche s'appuie sur une démarche qualitative et quantitative, visant à comprendre comment la conscience des limites techniques dans l'usage d'un logiciel de modélisation peut influencer les intentions formelles en architecture. La méthode retenue combine :

Questionnaire en ligne diffusé à un échantillon plus large (étudiants et jeunes architectes), visant à dégager des tendances générales quant à la perception des limites techniques, leur influence sur le processus de conception, et la capacité des concepteurs à en prendre conscience.

Exercice de conception comparée : il sera proposé à un panel d'étudiants. Il consistera à concevoir d'abord une forme architecturale libre sous forme de croquis, puis à tenter de la reproduire à l'aide du logiciel de modélisation qu'ils maîtrisent le mieux. L'objectif est d'analyser les écarts entre l'intention initiale et la réalisation modélisée, afin d'identifier comment la conscience des limites techniques influence, voire oriente, les choix formels dès les premières phases de conception.

ÉTAT DE L'ART

LA CONCEPTION :

Afin de mieux comprendre l'impact des outils numériques post-CAO classique dans la phase conceptuelle d'un projet architectural, il est essentiel de définir, dans un premier temps, ce qu'est la conception dans un sens large et générique, avant de se pencher plus précisément sur la conception architecturale.

La compréhension du processus de conception s'appuie sur plusieurs modèles théoriques majeurs recensés par Visser (2005). Si le modèle historique "Problème-Solution" de Simon (1969) propose une approche linéaire souvent jugée trop mécaniste pour l'architecture, la recherche contemporaine privilégie des approches itératives. C'est notamment le cas du modèle *See-Transform-See* de Schön (1983), qui définit la conception comme une « conversation réflexive avec la situation » basée sur des cycles d'interprétation et d'action, ou encore de la théorie C-K (Hatchuel & Weil), qui modélise la conception comme un dialogue enrichissant entre l'espace des Concepts et celui des Connaissances.

Toutefois, la conception architecturale se distingue des autres domaines par des spécificités propres, telles que la gestion de l'échelle et l'ancrage fort dans un contexte physique et temporel. Selon Safin (2011), elle constitue une activité cognitive et graphique complexe visant une composition originale qui doit satisfaire de multiples contraintes (esthétiques, techniques, culturelles). Contrairement à l'ingénierie qui suit un modèle « prescriptif » et normatif, l'architecture relève d'un modèle « descriptif » (Chupin, 2010) : elle ne suit pas de règles linéaires strictes mais progresse de manière opportuniste par cycles cognitifs. Cette nature incite les architectes à privilégier la recherche de solutions créatives singulières plutôt que la simple résolution fonctionnelle d'un problème.

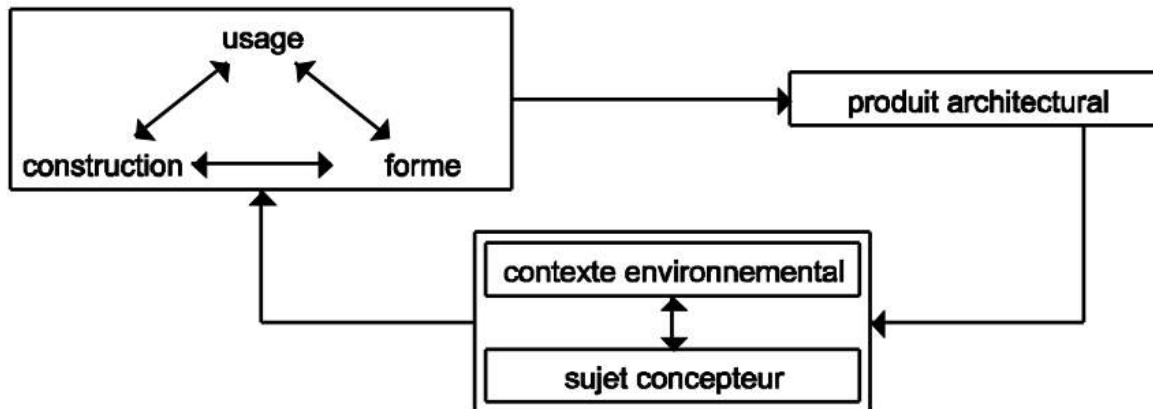


Figure 01 :Modèle du processus de conception par Farel (1991 repris par Claeys, 2013)

Afin de mieux saisir la dimension créative des solutions architecturales, il est utile de définir la créativité, ainsi que la manière dont elle peut être évaluée et qualifiée. La créativité est majoritairement décrite en sciences cognitives comme un processus associatif. Comme le souligne Bruner en citant Henri Poincaré, la pensée créative repose sur la capacité à opérer des combinaisons qui « nous révèlent des parentés insoupçonnées entre des faits connus de longue date, mais que l'on croyait à tort étrangers les uns aux autres ».

Dans cette perspective, l'architecte ne crée pas *ex nihilo*. Il puise dans un répertoire de formes, de contraintes techniques et de références culturelles pour générer une synthèse nouvelle. Runco (2007) précise que cette pensée créative se distingue de la logique rationnelle traditionnelle par sa non-linéarité : dans l'équation cognitive de la création, l'originalité prime temporairement sur la justesse ou la convention.

Les limites du modèle de la "Nouveauté Authentique" face au numérique

Cependant, une tension apparaît lorsque l'on confronte les théories classiques aux outils numériques. Hausman (1976) défend une vision exigeante de la créativité, qu'il associe à une « nouveauté authentique ». Selon lui, un résultat n'est véritablement créatif que s'il est différent, par nature, de toute forme disponible auparavant. Il rejette ainsi les solutions qui ne seraient que des extensions ou des variations de l'existant, privilégiant la notion de « spontanéité ».

Cette définition pose problème dans le cadre de la modélisation paramétrique ou du BIM. En effet, l'outil numérique fonctionne par essence sur des règles, des bibliothèques et des scripts préexistants. Si l'on suit Hausman, l'utilisation d'un logiciel qui contraint nécessairement les possibilités par son code empêcherait toute "vraie" créativité. Cette vision binaire apparaît aujourd'hui réductrice et inadaptée pour analyser le travail des étudiants en architecture.

Pour dépasser cette impasse, il est plus pertinent d'adopter la distinction proposée par Margaret Boden (1998), qui différencie deux types de créativité :

La créativité historique (H-Creativity) : Une invention radicalement nouvelle à l'échelle de l'histoire humaine.

La créativité psychologique (P-Creativity) : Une idée qui est nouvelle *pour l'individu* qui la produit, même si elle a déjà été découverte par d'autres auparavant.

C'est cette P-Creativity qui est au cœur de notre recherche. Lorsqu'un étudiant conçoit un projet, l'enjeu n'est pas qu'il révolutionne l'histoire de l'architecture (H-Creativity), mais qu'il soit capable d'explorer son propre potentiel créatif pour trouver une solution inédite *dans son parcours*.

C'est précisément ici que le biais de compréhension de l'outil intervient : si le logiciel est perçu comme trop complexe ou limitant, il risque d'étouffer cette P-Creativity. L'utilisateur, bloqué par l'interface, ne parvient même pas à produire une forme nouvelle *pour lui-même*, se contentant de reproduire les formes standardisées (cubes, plans) que le logiciel lui tend.

Bien que la définition de Hausman (1976) selon laquelle « chaque manifestation de nouveauté authentique est un signe de créativité authentique » ait marqué les théories de la création, elle repose sur une distinction binaire et problématique entre « vraie » et « fausse » créativité.

Dans le contexte contemporain de la conception architecturale, et plus encore de la conception **numérique**, cette séparation apparaît réductrice.

Dans ce cadre, les pratiques numériques relèvent souvent d'une créativité recombinaire, où la nouveauté émerge du remaniement de contraintes existantes, de scripts ou de logiques logicielles.

Cette approche ne nie pas la créativité, mais la déplace : elle ne réside plus seulement dans la rupture absolue, mais dans la capacité à détourner, réinterpréter ou hybridiser un cadre donné.

Ce mémoire vise à observer si oui ou non la créativité est influencée par l'utilisation des outils de modélisation numérique durant la phase de conception. Dès lors, il apparaît nécessaire de comprendre dans quelle mesure le degré de maîtrise du logiciel détermine le champ des possibles en matière de conception formelle. C'est pour cela qu'il est nécessaire d'aborder la question du niveau de maîtrise qui constitue un enjeu méthodologique majeur dans cette recherche. En effet, la « maîtrise » n'est pas une donnée stable, objective ni directement mesurable. Elle dépend à la fois de l'expérience, de la créativité et du rapport singulier que chaque concepteur entretient avec l'outil.

Comme le souligne Donald Schön (1983) dans sa théorie du *praticien réflexif*, la compétence ne se résume pas à l'application de procédures acquises, mais à la capacité d'entrer dans un dialogue réflexif avec la situation. Appliqué à la modélisation numérique, cela signifie que deux utilisateurs d'un même logiciel peuvent mobiliser des stratégies très différentes : l'un en suivant les règles établies, l'autre en les contournant ou en les réinventant.

De même, Mihaly Csikszentmihalyi (1996), à travers sa notion de *flow*, rappelle que la créativité n'est pas proportionnelle au niveau de maîtrise technique. Un concepteur peut éprouver un haut degré de créativité dans un état d'exploration ou de découverte, précisément parce qu'il se situe dans une zone d'incertitude productive entre compétence et défi.

Enfin, Donald Norman (1988), dans *The Design of Everyday Things*, distingue la maîtrise fonctionnelle d'un outil de sa compréhension interprétative. Autrement dit, savoir exécuter une commande ne signifie pas comprendre les principes sous-jacents du système. Cette distinction est centrale ici : un utilisateur peut "connaître" un logiciel sans pour autant en percevoir les potentialités formelles, ce qui alimente le biais de compréhension étudié dans ce mémoire.

Ainsi, évaluer la maîtrise d'un logiciel ne peut pas se limiter à une mesure de compétence technique (nombre de fonctions connues, temps de modélisation, etc.). Il s'agit plutôt d'observer la manière dont la compréhension juste ou erronée oriente la production formelle. C'est pourquoi la méthodologie adoptée ne cherche pas à mesurer un "niveau de maîtrise" absolu, mais à identifier les effets cognitifs et créatifs de la perception que le concepteur a de l'outil.

et pour mieux comprendre l'impact de ces outils de modélisation il est nécessaire de savoir où et à quel moment ces derniers entrent en jeu pendant la phase de conception, ce qui suggère la présence de différents niveaux d'intégration des outils lors de la conception, comme le soulignent plusieurs chercheurs (Oxman, 2006 ; Rivard & Gero, 1999 ; Salim & Burry, 2010) ils distinguent aujourd'hui trois grandes postures de conception face à l'outil numérique : **concevoir hors, avec ou dans l'outil.**

Concevoir "hors" de l'outil

La conception s'effectue ici de manière autonome : le logiciel n'intervient qu'a posteriori, comme un outil de saisie servant à représenter une idée déjà formée.

Le processus reste ancré dans des méthodes traditionnelles, croquis, maquettes physiques, schémas qui précèdent la modélisation.

Concevoir “avec” l’outil

Cette posture est instrumentée : l’outil devient un partenaire qui accompagne le processus de conception.

Selon Rivard & Gero (1999), le logiciel agit comme un système de feedback, capable de susciter de nouvelles idées par l’expérimentation.

Oxman (2006) parle ici de *Design Knowledge Representation* : le concepteur apprend à “penser à travers” l’outil, qui influence mais ne détermine pas la forme.

Concevoir “dans” l’outil

Dans ce cas, la conception se déroule à l’intérieur même du système numérique, dont la logique structure la pensée du concepteur.

L’outil n’est plus un support, mais un environnement cognitif complet : les formes naissent de la manipulation directe de ses paramètres.

Salim & Burry (2010) montrent que cette immersion entraîne parfois une dépendance aux logiques internes du logiciel, ce qui peut générer un biais de compréhension : l’utilisateur ne conçoit plus que ce que l’outil lui permet ou semble lui permettre.

Ces trois modes de conception ne sont pas exclusifs : la plupart des concepteurs peuvent faire appel à différentes postures entre eux selon les étapes du projet. Toutefois, cette distinction permet d’interroger l’influence réelle de l’outil sur la production formelle.

Le but n’est pas de mesurer une “maîtrise”, mais de comprendre comment la perception de la maîtrise de l’outil agit comme un filtre qui conditionne et restreint les choix formels du concepteur.

Les recherches récentes montrent que les outils numériques ne sont plus de simples supports de représentation, mais participent au processus de conception.

Selon Oxman (2006), nous sommes entrés dans une “première ère numérique du design”, où les outils computationnels modifient les modes de pensée des concepteurs. L’auteure introduit la notion de Digital Design Thinking, qui décrit la manière dont la manipulation directe des paramètres géométriques, des algorithmes et des scripts transforme la relation entre idée et forme.

Dans la même logique, Gero (1990) propose une approche fondée sur la théorie des prototypes de conception, où les représentations mentales évoluent continuellement en fonction des interactions avec l’outil. La conception n’est plus perçue comme une suite linéaire de décisions, mais comme un système adaptatif, réactif aux contraintes et aux affordances du logiciel.

Pour Lawson (2005), dans *How Designers Think*, cette dynamique traduit un changement profond : les outils numériques participent à la construction du raisonnement visuel et spatial du concepteur. L'architecte pense "avec" les représentations qu'il produit ; chaque transformation graphique rétroagit sur la pensée conceptuelle.

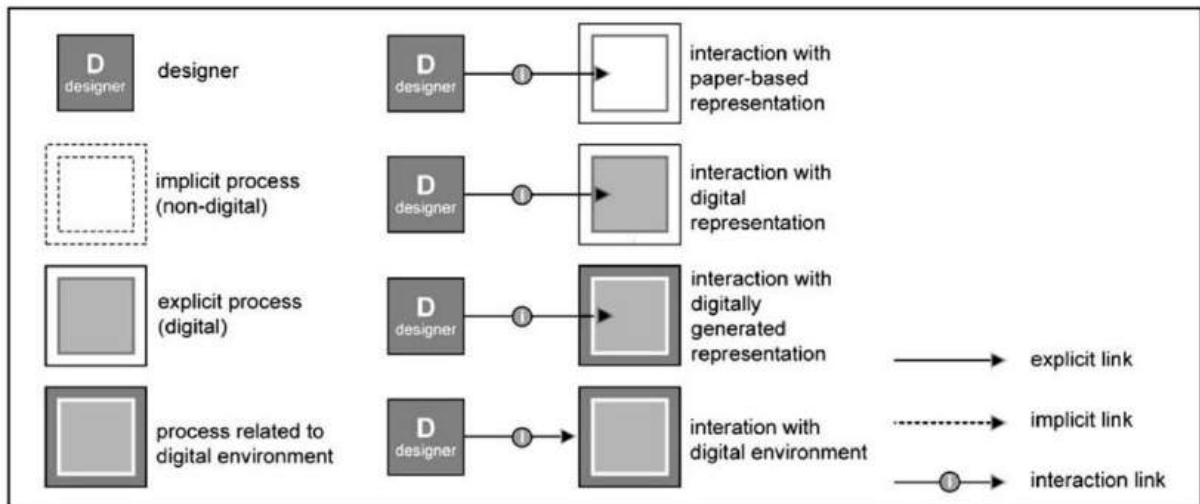


Figure 2. Symboles, limites et liens définis par Oxman (2006)

Kvan (2004), quant à lui, distingue une double héritage de la conception : une composante artistique et intuitive d'un côté, et une composante technique et rationnelle de l'autre. Les environnements numériques réunissent ces deux dimensions, en permettant un va-et-vient constant entre exploration libre et formalisation structurale.

Cela rejoint la distinction entre conception "avec", "dans" ou "hors" de l'outil, évoquée plus haut : la place accordée à l'outil dans le raisonnement influe directement sur la nature de la production formelle.

Ainsi, la littérature contemporaine s'accorde à reconnaître que la conception architecturale numérique ne peut plus être réduite à un transfert de méthodes traditionnelles dans un environnement informatisé. Elle constitue un mode de pensée propre, façonné par l'outil lui-même, et c'est précisément dans cet espace d'interaction entre compréhension, biais et exploration que se situe l'objet de ce mémoire.

Au-delà de la simple maîtrise technique, l'interaction entre le concepteur et l'outil numérique est marquée en grande partie par le concept d'affordance. Introduit initialement par le psychologue J.J. Gibson (1979) dans le cadre de la psychologie écologique, ce terme désigne les possibilités d'action offertes par l'environnement à un individu (par exemple, une surface plane "afforde" la marche).

Selon Donald Norman, le terme *affordance* se rapporte aux propriétés perçues et réelles d'un objet, principalement les caractéristiques fondamentales qui déterminent la manière dont l'objet pourrait être utilisé. Toutefois, dans le contexte des interfaces numériques et de la conception assistée par ordinateur (CAO), c'est cette définition qui s'avère pertinente à ce sujet, qui est l'*affordance perçue* (*perceived affordance*). Contrairement à l'*affordance réelle* (ce que le logiciel peut techniquement faire), l'*affordance perçue* désigne ce que l'utilisateur identifie comme possible à travers les indices visuels de l'interface (icônes, menus, curseurs).

Dans les logiciels de modélisation, cette notion est importante car elle n'est pas neutre vu qu'elle dialogue directement avec l'utilisateur. L'organisation de l'interface créer une *affordance perçue* par l'utilisateur ce qui hiérarchise les possibles : les commandes de modélisation standard (murs droits, dalles planes, fenêtres rectangulaires) bénéficient d'une *affordance forte* (accessibles en un clic, mises en avant visuellement), tandis que les outils de géométrie complexe ou non-standard sont souvent relégués à des niveaux d'interface inférieurs (sous-menus, scripts), présentant une *affordance faible*.

Ce déséquilibre crée un guidage invisible qui aura tendance à suggérer un type de forme (courbe ,rectiligne etc.). Comme le suggèrent Gaver (1991) et plus récemment les critiques du *BIM*, l'outil ne se contente pas d'être un passif réceptacle d'intentions , il suggère activement des solutions par la facilité d'accès à certaines commandes. Le risque pour l'utilisateur est de confondre ce qui est facilement modélisable (*affordance perçue*) avec ce qui est *techniquement réalisable* (*affordance réelle*), restreignant ainsi son vocabulaire formel aux seules "invitations" explicites du logiciel.

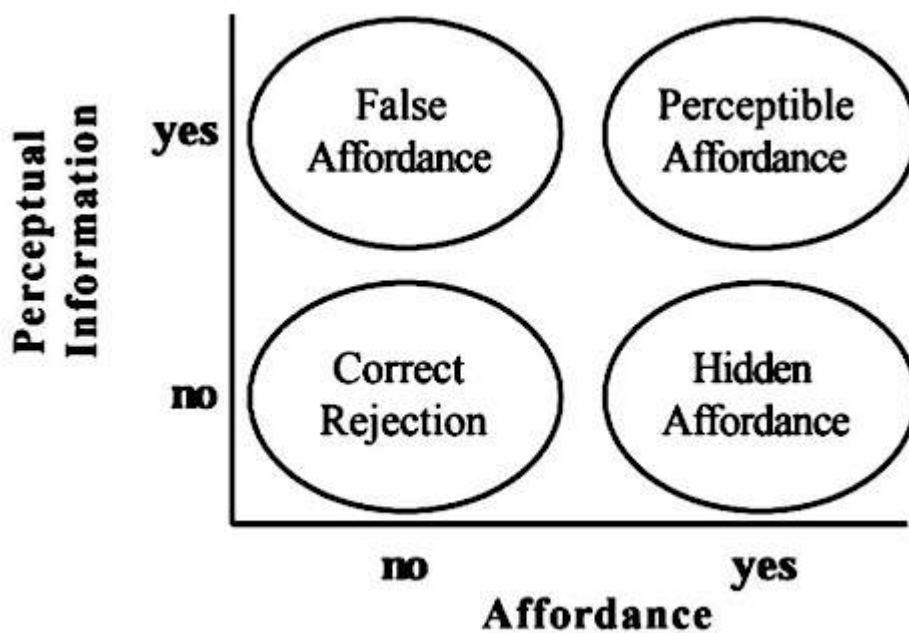


Figure 03 :William W. Gaver. *Technology Affordances*. 1991. Séparer les affordances des informations les rendant disponibles permet de distinguer les rejets corrects des [rejets] perçus, les affordances dissimulées des [affordances] fausses.

après avoir pris connaissance des éléments qui caractérisent l'utilisation des outils de modélisation numériques dans la phase de conception architecturale notamment durant l'exploration formelle, nous constatons que l'utilisateur est en quelque sorte mené et dirigé par l'outil en question ce qui le met face à un phénomène psychologique qui se nomme l'écart Intention-Action ('intention-action gap) ce qui a tendance à séparer la pensée conceptuelle de sa réalisation .

Dans le cadre de cette recherche je vais m'appuyer sur la définition que présente Donald Norman, il le nomme le « Golfe d'Exécution » (*Gulf of Execution*). Ce concept désigne la différence entre les intentions d'un utilisateur et les actions permises par le système pour les réaliser. Plus ce golfe est large, plus l'utilisateur doit fournir un effort cognitif important pour traduire sa pensée en commandes logicielles interprétables par la machine.

Dans le contexte spécifique de la conception architecturale, ce fossé est défini comme la différence observable entre l'ambition formelle initiale de l'architecte (telle qu'exprimée par l'esquisse ou l'image mentale) et sa concrétisation médiatisée par le logiciel.

C'est précisément face à cet écart que se manifeste le mécanisme d'incompétence apprise évoqué plus tôt. Anticipant la difficulté ou l'impossibilité de franchir ce *Golfe d'Exécution*, le concepteur risque d'ajuster ses intentions a priori. Il ne s'agit plus seulement d'un échec technique (ne pas réussir à modéliser), mais d'une censure

préventive : l'architecte réduit la complexité de son intention pour la faire correspondre à sa perception des limites de sa maîtrise de l'outil.

L'outil finit par dicter l'étendue du "pensable" avant même que l'action ne soit tentée.

Un biais psychologique supplémentaire survient dès que l'utilisateur se heurte à ce décalage entre son intention et le résultat observé que les chercheurs appellent **incompétence apprise** (*learned helplessness*), théorisée initialement par Martin Seligman (1967, 1972). Ce phénomène décrit l'état psychologique d'un individu qui, après avoir été exposé de manière répétée à des stimuli incontrôlables, finit par intégrer l'idée que ses actions n'ont aucune incidence sur le résultat. Cette conviction conduit à une passivité durable : le sujet renonce à agir, même lorsque la situation change et que le succès devient objectivement possible.

Transposé au domaine de l'interaction homme-machine et de la modélisation architecturale, ce mécanisme psychologique se manifeste souvent pendant lors de l'utilisation de logiciels avec une courbe d'apprentissage complexe ce qui mène souvent l'utilisateur à commettre plusieurs erreurs techniques (géométries invalides, bug) qui placent l'utilisateur dans une situation d'incontrôlabilité perçue. Face à ces micro-échecs répétés, l'utilisateur ne développe pas seulement une lacune technique, mais une croyance limitante : il apprend l'incapacité de l'outil (ou la sienne propre) à gérer certaines complexités formelles.

Cette résignation cognitive se traduit alors par une stratégie d'évitement. Plutôt que de risquer une nouvelle confrontation avec les limites du logiciel (le Golfe d'exécution), le concepteur restreint son champ exploratoire aux seules solutions dont il maîtrise l'issue certaine. L'incompétence apprise agit ainsi comme un filtre réducteur en amont du processus d'exploration formelle : la forme n'est plus dictée par l'intention architecturale, mais par l'anticipation au préalable de ce que l'outil semble "autoriser".

Protocole expérimental projeté et cadre d'analyse

:

Dans cette partie, je ne présente pas les résultats d'une expérience réalisée, mais la construction rigoureuse d'un dispositif expérimental. Ce protocole projeté a pour vocation de définir les conditions nécessaires à la vérification de notre hypothèse centrale : l'influence du *biais de compréhension* et de *l'incompétence apprise* sur la réduction de l'ambition formelle en architecture.

Le choix du protocole comporte une approche mixte, combinant une méthode qualitative et quantitative, afin d'observer dans quelle mesure ce biais se manifeste, à quel moment il apparaît et comment il modifie par des ajustements *a priori* les intentions formelles.

La méthode quantitative (Questionnaire)

Le questionnaire vise à collecter des données sur la perception des limites techniques, l'influence sur le processus de conception, et la capacité des concepteurs à en prendre conscience. Il est diffusé à un échantillon plus large (étudiants et jeunes architectes) afin d'identifier des tendances déclarées.

Cette évaluation sera réalisée à l'aide d'un mélange de questions fermées et ouvertes :

Questions fermées :

Mesure de la fréquence de l'anticipation ("À quel point anticipez-vous les difficultés de modélisation dès l'esquisse?") et de l'abandon d'idées à cause de l'outil.

Questions ouvertes :

Des questions qualitatives permettront aux répondants de détailler leur expérience (ex. : "Décrivez une idée que vous avez abandonnée ou simplifiée en raison de la complexité logicielle"). L'analyse de contenu de ces réponses fournira des données sur la posture de réflexion du concepteur.

La collecte de données auprès des participants à l'exercice de conception se déploiera en deux temps distincts, encadrant l'expérimentation :

Questionnaire pré-expérimental (Profilage et tendances) :

Il sera diffusé en amont de l'exercice, ce premier volet vise à établir le profil et le niveau de maîtrise technique des participants. Il permet d'identifier leurs habitudes d'usage, leurs comportements face aux logiciels, ainsi que leurs préférences formelles (types de formes privilégiées) avant toute mise en situation.

Questionnaire post-expérimental (Rétroaction et métacognition) :

Il sera diffusé après le déroulement de l'exercice, ce second volet recueille le retour d'expérience immédiat (feedback). Il a pour objectif d'identifier les difficultés techniques rencontrées et les stratégies de contournement mises en œuvre. Le questionnaire aura pour but d'interroger également l'évolution de la perception de l'outil.

La méthode qualitative (exercice de conception)

Afin d'observer la manifestation concrète du biais de l'intention–action gap, il est nécessaire d'effectuer un exercice de conception.

Définition du panel et de l'objet d'étude :

L'échantillon théorique

Le dispositif prévoit un groupe de 30 participants, composé d'étudiants en fin de cycle (Master 1 ou 2) ou de jeunes diplômés. Ce profil est ciblé car il se situe à un point de bascule : les sujets possèdent une culture architecturale suffisante pour formuler des intentions complexes, mais leur maîtrise technique des outils est souvent intermédiaire, ce qui favorise l'apparition du « Golfe d'Exécution » décrit par Norman.

L'objet architectural à concevoir :

Le choix s'est porté sur la conception d'un pavillon ouvert, avec pour seule contrainte programmatique la couverture d'une surface définie, afin d'éviter de perdre du temps dans l'élaboration d'un programme et de se trouver dirigé par celui-ci, notamment par les normes spatiales et architecturales qu'il implique.

Cet objet architectural permet une liberté morphologique qui donnera aux participants plus de liberté d'exploration formelle.

Le choix du logiciel (outil utilisé)

L'expérience impose l'usage d'un logiciel de type BIM dans notre cas Revit, le choix s'est porté vers ce logiciel car premièrement c'est un logiciel couramment utilisé auprès des étudiants et dans les agences et il présente de base une affordabilité visible de certains outils ce qui hiérarchise les possibles en rendant les formes simples (dalles plates) accessibles en un clic, tout en imposant de passer par des procédures ou des outils à faible affordabilité si on souhaite modéliser des formes plus complexes (modélisation in-situ, scripts). C'est cet environnement contraint qui est nécessaire pour vérifier l'hypothèse en question.

Dispositif comparatif :

Pour distinguer ce qui relève de la difficulté intrinsèque de la 3D de ce qui relève spécifiquement de la rigidité de l'interface BIM, les participants sont répartis aléatoirement en trois groupes de 10 personnes.

GROUPE A : Transposition BIM

Phase 1 (Hors-outil) : Conception libre de l'esquisse à la main (crayon, papier, maquette). L'objectif est de fixer l'intention sans filtre numérique.

Phase 2 (Dans-outil) : Modélisation de ce projet sur Revit.

Objectif : Mesurer la perte d'information, la simplification géométrique et les renoncements lors du passage de l'analogique au BIM.

GROUPE B : Transposition Témoin (La variable de contrôle)

Phase 1 (Hors-outil) : Conception libre à la main (identique au Groupe A).

Phase 2 (Dans-outil souple) : Modélisation de ce projet sur un modéleur surfacique intuitif (type Rhinoceros 3D ou Blender).

Objectif : Ce groupe sert de témoin. Si les participants du Groupe B parviennent à modéliser leurs formes complexes alors que ceux du Groupe A échouent ou simplifient, cela confirmera que le blocage provient bien de l'affordabilité spécifique du BIM (et non d'une incapacité générale à modéliser).

GROUPE C : Conception Immersée (Directement dans l'outil)

Phase 1 : conception et modélisation directement dans Revit, sans passage par le croquis préalable.

Objectif : Observer si l'outil censure l'idée dès sa genèse. L'hypothèse est que les formes produites seront structurellement différentes (plus orthogonales, assemblées par composants standards) car l'imaginaire du concepteur sera borné par les commandes immédiatement disponibles (affordance perçue).

Distinctions entre la nature des limites.

Pour analyser correctement l'impact des outils, il est impératif de distinguer trois niveaux de contraintes souvent confondus par les utilisateurs :

Les limites réelles : Ce sont les barrières infranchissables liées au code informatique du logiciel. Par exemple, l'impossibilité pour un logiciel surfacique simple de gérer des données BIM complexes. Ces limites relèvent de l'affordance réelle.

Les limites perçues : C'est ici que se joue le biais de compréhension. L'utilisateur pense qu'une forme est impossible à réaliser parce que l'interface ne la suggère pas explicitement (affordance faible ou cachée). Comme le note Norman, c'est ce décalage qui creuse le « Golfe d'Exécution ».

Les limites auto-imposées: Conséquence directe de l'incompétence apprise. Anticipant un échec ou une difficulté trop grande (le coût cognitif), le concepteur s'interdit d'explorer certaines voies. Il ne s'agit pas d'une impossibilité technique, mais d'une résignation psychologique qui réduit le champ du "pensable" avant même l'action.

Critères d'analyse et grille d'évaluation:

Cette grille vise à quantifier les observations pour mesurer l'impact de l'outil sur la production. Elle s'appuie sur les cadres théoriques de la complexité géométrique des production formelle .

.La complexité géométrique (L'indice de liberté) ce critère évalue la typologie formelle produite. Il permet de vérifier si l'usage de l'outil enferme l'étudiant dans des solutions standards, comme le suggèrent Foster & Partners (2006) concernant les contraintes des bibliothèques prédéterminées.

Orthogonal standard : Géométrie rectiligne basée sur les primitives du logiciel (murs droits, dalles). Correspond à une utilisation passive des "possibilités offertes par les interfaces".

Angulaire / Facetté : Géométrie qui tente de rompre avec l'orthogonalité mais reste limitée par une logique de plans discrets, souvent liée aux méthodes de modélisation "Low Poly" ou simplifiées.

Courbe simple & Forme libre : Géométries nécessitant une maîtrise avancée des paramètres ou de la géométrie algorithmique. C'est ici que l'on mesure l'écart avec le potentiel théorique des outils paramétriques décrit par Turrin et al. (2011) ou Adriaenssens (2016).

Conservation : L'intention spatiale est maintenue .

Adaptation / Simplification : La forme est altérée pour correspondre aux capacités perçues de l'outil.

Abandon : L'idée initiale est rejetée au profit d'une solution par défaut plus simple.

C. L'analyse des opérations logicielles (Approche procédurale) Nous analysons *comment* la forme a été construite, en nous référant à la distinction faite par Salim & Burry (2010) sur l'ouverture du logiciel (*Software Openness*) et la rigidité des paramètres.

Usage de primitives standards : Utilisation des outils automatiques L'étudiant se conforme à la "logique interne du logiciel".

Modélisation In-Situ / Détournement : L'étudiant contourne les outils standards (ex: les volume avec des volumes génériques). Cela démontre une tentative de sortir du cadre imposé par le logiciel pour retrouver une liberté de conception, rejoignant l'idée de "conception hors de l'outil" ou instrumentée.

Scénarios prospectifs et interprétation:

Puisque cette recherche est théorique, nous envisageons trois scénarios de résultats distincts. Chaque scénario offrirait une conclusion différente sur le rôle de l'incompétence apprise.

Scénario A : La Standardisation par l'Outil (Validation de l'hypothèse):

L'observation projetée :

Dans ce scénario, l'analyse comparative révélerait un fossé majeur (Intention-Action Gap) entre les esquisses manuelles et les modèles BIM produits par le Groupe A. Alors que les croquis montreraient des formes fluides, courbes ou topologiques, les modélisations Revit correspondantes se caractériseraient par un retour massif à l'orthogonalité. Parallèlement, le Groupe B (sur modèleur souple type Rhino) parviendrait à conserver une fidélité morphologique bien supérieure, démontrant que la difficulté ne réside pas dans la 3D elle-même, mais bien dans la rigidité de l'interface BIM. Enfin, les projets du Groupe C (conception directe dans Revit) seraient structurellement les plus simples, composés quasi-exclusivement de primitives standards (murs droits, toitures plates).

Interprétation :

Ce scénario validerait pleinement l'hypothèse de l'incompétence apprise. Face à l'interface de Revit, qui hiérarchise les outils en rendant les formes simples accessibles en un clic (affordance forte) et les formes complexes cachées dans des sous-menus (affordance faible), l'étudiant opère une auto-censure préventive. Le mécanisme psychologique à l'œuvre est le suivant : l'étudiant anticipe le coût cognitif exorbitant nécessaire pour modéliser une courbe (création de familles in-situ, géométrie complexe) et préfère inconsciemment modifier son intention architecturale pour la faire entrer dans les "cases" du logiciel. L'outil ne sert plus à représenter le projet, c'est le projet qui se réduit à la capacité perçue de l'outil.

Scénario B : La Transposition Négociée (Nuance de l'hypothèse)

L'observation projetée :

Ici, les résultats seraient plus subtils. L'écart Intention-Action ne se traduirait pas par une simplification simpliste, mais par une mutation de la forme. Les courbes dessinées à la main deviendraient, dans le modèle Revit, des formes facettées ou segmentées. L'étudiant n'abandonnerait pas son idée de complexité, mais il la "traduirait" dans le langage de l'outil : un mur courbe deviendrait une succession de petits murs droits ; une toiture organique deviendrait un assemblage de plans inclinés.

Interprétation :

Ce scénario suggère que l'incompétence apprise n'est pas totale. L'utilisateur ne renonce pas, mais il entre dans une négociation avec le logiciel. Le biais de compréhension reste actif (il pense ne pas pouvoir faire de "vrai" courbe), mais il le compense par une stratégie de détournement des outils standards. Dans ce cas de figure, l'outil agit comme un filtre traducteur. Il ne tue pas la créativité, mais il modifie le vocabulaire architectural : on passe d'une architecture du "trait continu" à une architecture de l'assemblage et du composant. L'affordance de l'outil oriente le style final sans pour autant appauvrir la richesse formelle du projet.

Scénario C : La Potentialisation Numérique (Infirmité de l'hypothèse)

L'observation projetée :

Ce scénario, moins probable chez des étudiants intermédiaires mais théoriquement possible, montrerait des résultats inverses : les projets conçus directement dans l'outil (Groupe C) seraient *plus* complexes et audacieux que ceux conçus à la main. L'analyse morphologique révélerait des structures répétitives complexes, des motifs paramétriques ou des géométries régulières mais sophistiquées, impossibles à dessiner précisément à la main ou trop fastidieuses à esquisser.

Interprétation :

ce scénario se vérifie, cela signifie que l'utilisateur a dépassé le stade de l'incompétence apprise pour atteindre celui du *Digital Design Thinking* (Oxman, 2006). Au lieu de subir les limites de l'outil, il utilise sa puissance de calcul (répétition, array, familles paramétriques) pour générer de la forme. Ici, l'affordance du logiciel devient un moteur créatif : la rigidité du BIM (travailler par objets) pousse

à concevoir des systèmes modulaires intelligents. L'hypothèse de la restriction est alors invalidée : l'outil n'est plus un frein, mais un partenaire cognitif qui étend le champ du "pensable" au-delà des capacités manuelles de dessin de l'architecte.

DISCUSSION GÉNÉRALE :

Au travers des scénarios prospectifs détaillés précédemment, cette recherche théorique met en lumière la tension fondamentale qui s'opère lors de la phase conceptuelle instrumentée : le conflit entre l'ambition formelle (l'intention) et la résistance perçue du logiciel (l'affordance). Si les scénarios varient dans leurs résultats formels (de la simplification radicale à la complexification procédurale), tous convergent vers une conclusion commune :

l'outil de modélisation n'est jamais un réceptacle neutre. Il agit comme un acteur cognitif qui conditionne le champ du possible.

Le "Coût Cognitif" et le rétrécissement du "Golfe d'Exécution" :

L'analyse du Scénario A (simplification massive) permet d'interroger la notion de « Golfe d'Exécution » définie par Donald Norman (1988). Dans le cas d'un logiciel BIM comme Revit, ce golfe n'est pas seulement technique, il est économique (en termes d'effort mental). L'interface du logiciel crée une hiérarchie des coûts cognitifs :

Coût faible : tracer un mur droit, poser une porte standard. (Affordance forte).

Coût élevé : modéliser une double courbure, créer une famille paramétrique in-situ. (Affordance faible).

Face à cette économie de l'effort, le concepteur novice ou intermédiaire, soumis à des contraintes de temps, tendra rationnellement vers la solution au coût cognitif le plus bas. L'incompétence apprise se manifeste ici non pas par une incapacité absolue à faire (l'étudiant pourrait apprendre à faire la courbe), mais par une résignation face à l'effort anticipé. Le logiciel ne dit pas « c'est impossible », il suggère implicitement « c'est trop compliqué pour ce que ça vaut », poussant l'architecte à réviser ses ambitions à la baisse.

L'Affordance comme vecteur de standardisation architecturale:

Les scénarios A et C (conception directe) soulèvent une problématique majeure : la standardisation des typologies architecturales induite par l'outil. Comme le souligne Gibson (1979), l'affordance est une invitation à l'action. Or, l'affordance des logiciels BIM est intrinsèquement liée à une logique constructive standardisée (le mur, la

dalle, la fenêtre). Si les résultats hypothétiques montraient une prédominance des formes orthogonales dans le Groupe C, cela confirmerait que l'outil ne se contente pas d'assister la conception, il la prescrit. En rendant la géométrie orthogonale (la « boîte ») immensément plus accessible que la géométrie topologique (le « blob » ou le pli), le logiciel agit comme un moule normatif. Le biais de compréhension joue ici à plein : l'utilisateur finit par confondre les limites de la bibliothèque du logiciel (ce qui est pré-codé) avec les limites de l'architecture elle-même.

Nuance : De l'incompétence apprise à la "Compétence Négociée" :

Le Scénario B (adaptation/négociation de la forme) permet de nuancer l'hypothèse de l'incompétence apprise totale. Il suggère l'existence d'une zone intermédiaire où le concepteur développe des stratégies de contournement. Dans cette perspective, l'obstacle technique ne bloque pas la créativité, mais la déplace. L'architecte ne dessine plus la forme, il « bricole » l'outil pour s'en approcher (par exemple, en utilisant un outil « toiture » pour faire un mur incliné). Cependant, même dans ce cas de figure plus optimiste, l'influence de l'outil reste déterminante : l'intention initiale est altérée. Elle n'est plus purement issue de l'imaginaire du concepteur, mais devient le résultat d'un compromis diplomatique entre ce que l'architecte veut et ce que l'outil laisse faire.

Synthèse : La nécessité d'une émancipation technique

En définitive, la discussion de ces scénarios hypothétiques permet de réaffirmer que la liberté de conception à l'ère numérique est proportionnelle à la maîtrise technique.

Pour le débutant (Biais fort) : L'outil est une cage. L'incompétence apprise dicte la forme.

Pour l'expert (Biais faible) : L'outil est un levier. La connaissance approfondie (scripting, modélisation avancée) permet de briser les affordances restrictives et de retrouver la fluidité du croquis manuel (Digital Design Thinking, Oxman 2006).

L'enjeu pédagogique n'est donc pas de rejeter le BIM, mais d'enseigner à « penser contre » l'interface, pour éviter que l'architecture produite ne soit que le reflet des lignes de code du logiciel utilisé.

Limites de la recherche

Bien que le protocole expérimental projeté et les scénarios élaborés permettent de structurer théoriquement notre problématique, cette recherche comporte des limites intrinsèques qu'il convient d'exposer. La conscience de ces biais est nécessaire pour nuancer la portée de mes conclusions.

Limite épistémologique : L'absence de validation empirique :

La limite majeure de ce travail réside dans son statut théorique et projectif. Les scénarios de résultats présentés (A, B et C) sont des constructions intellectuelles basées sur une littérature existante et une observation diffuse, mais ils ne sont pas validés par des données de terrain.

Il existe donc un risque de biais de confirmation théorique : j'ai construit des scénarios qui "collent" à nos concepts (incompétence apprise, affordance), alors que la réalité des pratiques étudiantes pourrait révéler des facteurs plus triviaux (manque de temps, désintérêt pour l'exercice, lacunes géométriques fondamentales indépendantes du logiciel). Seule une mise en œuvre réelle du protocole permettrait de trancher entre une causalité logicielle et une causalité pédagogique.

Les conditions de déroulement "Effet de Laboratoire" :

Le protocole proposé place les participants dans une situation artificielle (un exercice court, une consigne imposée, un temps limité). Cette configuration "de laboratoire" diffère considérablement de la réalité de l'agence ou de l'atelier de projet.

Facteur Temps : Dans la réalité, un architecte dispose de temps pour chercher une solution technique, regarder un tutoriel ou contourner un blocage. La contrainte de temps de l'exercice pourrait accentuer artificiellement l'effet de l'incompétence apprise (abandon rapide) là où, en situation réelle, il y aurait eu apprentissage.

Facteur Enjeu : L'absence d'enjeu réel (pas de client, pas de note, pas de construction) peut réduire la motivation à surmonter l'obstacle technique, faussant ainsi la mesure de la persévérance face à l'outil.

La difficulté d'isoler la variable Logiciel :

Malgré l'introduction d'un groupe témoin (Groupe B sur modeleur souple), il reste difficile d'isoler totalement la responsabilité de l'interface Revit. La simplification formelle observée dans nos scénarios pourrait ne pas provenir uniquement de l'interface (affordance), mais d'une lacune de compétence géométrique chez les étudiants. Savoir dessiner une courbe complexe à la main ne signifie pas savoir la rationaliser en 3D. L'écart Intention-Action pourrait donc mesurer autant un déficit de maîtrise constructive numérique qu'un blocage psychologique induit par le logiciel. Le protocole peine à distinguer clairement "Je ne sais pas faire avec cet outil" (Compétence) de "L'outil m'empêche de faire" (Affordance).

Spécificité de l'outil et généralisation :

Enfin, cette recherche se concentre spécifiquement sur le couple Revit / Architecture non-standard. Les conclusions tirées de ces scénarios ne sont pas nécessairement généralisables à l'ensemble de la CAO. D'autres logiciels BIM (comme ArchiCAD ou Allplan) possèdent des interfaces et des logiques d'objets différentes qui pourraient induire des comportements cognitifs distincts.

Perspectives possibles

Plusieurs perspectives de recherche s'ouvrent, notamment à travers une diversification logicielle visant à comparer la rigidité du BIM à la souplesse d'outils surfaciques comme Rhinoceros 3D ou blender , afin de vérifier si l'incompétence apprise est liée à l'usager ou induite par l'interface ou le type de modélisation proposé par le logiciel.

Sur le plan pédagogique, il apparaît essentiel d'enseigner le Digital Design Thinking pour que les étudiants apprennent à « penser à travers » l'outil, tout en les sensibilisant aux mécanismes d'auto-censure qui restreignent leur champ du « pensable ».

Cette recherche met l'accent sur le rôle de l'architecte, qui ne peut plus se contenter d'une maîtrise fonctionnelle des logiciels à savoir exécuter des commandes de base. Pour se détacher du guidage invisible imposé par les affordances logicielles et éviter que l'outil n'enferme la pensée dans des solutions standards , l'architecte doit devenir un praticien capable d'entrer dans un dialogue interprétatif avec l'outil numérique . La maîtrise technique n'est alors plus une compétence accessoire, mais une condition nécessaire pour que le concepteur ne conçoive plus seulement « dans » ou « avec » la logique interne du logiciel, mais qu'il puisse l'orienter selon sa propre intention architecturale.

Par ailleurs, une approche méthodologique élargie permettrait d'évaluer si l'expertise parvient à effacer totalement l'écart intention-action ou si la standardisation des solutions persiste, même chez les professionnels.

Enfin, il apparaît essentiel d'étendre le protocole envisagé à une catégorie d'architectes experts. Cela permettrait d'approfondir la question de l'incompétence apprise : se manifeste-t-elle uniquement chez l'architecte novice, disparaît-elle avec l'expérience, ou persiste-t-elle sous une forme latente chez les professionnels expérimentés ? Il est possible que les experts ne soient pas "plus libres", mais simplement "mieux résignés". Ils auraient intériorisé les contraintes du logiciel au point de ne plus jamais concevoir de formes qui sortent du cadre.

Conclusion

Ce mémoire s'est intéressé sur l'influence que la perception des limites techniques des outils de modélisation numérique exerce sur la production formelle en architecture, en se concentrant sur une étape déterminante du projet : la phase conceptuelle.

À travers l'analyse de plusieurs biais psychologiques, cette recherche cherche à mettre en évidence des mécanismes souvent peu visibles, mais pourtant décisifs, qui influencent la manière dont les architectes et les étudiants conçoivent et traduisent leurs intentions architecturales.

Si notre démarche est restée théorique, la construction du protocole expérimental projeté et l'analyse des scénarios prospectifs ont permis de mettre en évidence un risque majeur : celui d'une architecture du moindre effort cognitif. Il apparaît que face à l'affordance restrictive des logiciels BIM, l'utilisateur tend à réduire son Intention-Action Gap non pas en élevant sa compétence technique, mais en abaissant son ambition formelle. L'anticipation de la difficulté (le coût cognitif de la modélisation complexe) engendre une auto-censure préventive qui standardise le projet avant même sa naissance numérique.

Cependant, cette fatalité n'est pas absolue. Comme l'ont suggéré nos scénarios nuancés, l'outil peut devenir un partenaire de conception ("concevoir avec") dès lors que l'utilisateur dépasse la simple exécution pour entrer dans une négociation créative avec la machine. En définitive, la liberté de l'architecte de demain ne résidera pas dans la puissance de ses logiciels, mais dans sa capacité critique à percevoir les limites qu'ils lui imposent, pour mieux les franchir ou les détourner.

Annexes

LE QUESTIONNAIRE

Questionnaire avant l'exercice

Objectif : Établir le profil technique et les habitudes de conception du participant.

1. Profil et Maîtrise Technique

- Quel est votre niveau de maîtrise déclaré sur le logiciel Revit ? (Débutant / Intermédiaire / Avancé / Expert).
- À quelle fréquence utilisez-vous des outils numériques dès la phase d'esquisse ? (Jamais / Rarement / Souvent / Systématiquement).
- Quels types de formes privilégiez-vous habituellement dans vos projets ? (Orthogonales / Organiques / Paramétriques / Mixtes).

2. Rapport à la Contrainte

- À quel point anticipez-vous les difficultés de modélisation dès la phase de réflexion (esquisse au crayon) ? (Échelle de 1 à 5).
- Avez-vous déjà renoncé à une idée formelle parce que vous ne saviez pas comment la modéliser ? (Oui / Non / Fréquemment).
- Face à un obstacle technique, quelle est votre réaction principale ?:
 - Abandon de l'idée.
 - Simplification de la forme.
 - Contournement du problème (recherche de tutoriel, détournement d'outil).

Questionnaire après l'exercice

Objectif : Recueillir le retour d'expérience immédiat et analyser l'évolution de la perception de l'outil après l'exercice.

1. Analyse des Écarts (Intention vs Réalisation)

- Selon vous, la forme finale modélisée est-elle fidèle à votre intention initiale ? (Échelle de 1 à 5).
- Quels changements majeurs avez-vous opérés entre la version croquis et la version numérique ?.
- Si vous avez simplifié la forme, quelle en est la cause principale ?:
 - Limite intrinsèque du logiciel (ce qu'il ne permet pas de faire).

- Manque de maîtrise personnelle (votre niveau technique).
- Choix conceptuel (la forme simplifiée vous a semblé finalement plus pertinente).

2. Ressenti

- Décrivez une idée ou un détail que vous avez abandonné ou simplifié lors de cet exercice en raison de la complexité logicielle..
- Avez-vous ressenti de la frustration ou un sentiment d'impuissance (incompétence apprise) lors de la phase de modélisation ?.
- Consigne de libération (pour le Groupe 2) : Si vous aviez pu dessiner sans aucune contrainte technique, qu'auriez-vous ajouté ou modifié par rapport à votre modèle Revit ?.

BIBLIOGRAPHIE

1. **Adriaenssens, Sigrid, et al.** (2016). *Advances in Architectural Geometry 2016*. vdf Hochschulverlag AG.
2. **Barrios Hernandez, Carlos Roberto / Hernandez, C. R. B.** (2006). « *Thinking Parametric Design: Introducing Parametric Gaudi* ». *Design Studies*, 27(3), 309-324.
3. **Benliay, Ahmet, et Isa Eren Akbiyik** (2016). *3D Modelling Creating Tool for Landscape Design: Camera*. Wichmann Verlag.
4. **Berlan, S.** (s. d.). « *Mémoire de fin d'études : "La mémoire d'un processus architectural : Interprétation et Extrapolation"* ».
5. **Caetano, Inês, et António Leitão** (2019). « *Integration of an Algorithmic BIM Approach in a Traditional Architecture Studio* ». *Journal of Computational Design and Engineering*, 6(3), 327-336.
6. **Cerny, Tomas, et al.** (2024). « *Is the 3D Model the Way to Go When Presenting Microservice Architecture?* ». *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering Workshops*, 223-27.
7. **Chin, Alexander W. H., et al.** (2024). « *Open-Source Software for Building-Integrated Photovoltaic Tiling for Novelty Architecture* ». *Designs*, 8(6), 118.
8. **Dino, I. G.** (2012). « *Creative design exploration by parametric generative systems in architecture* ». *Metu Journal of the Faculty of Architecture*, 29(1), 207-224.
9. **Gaver, W. W.** (1991). « *Technology affordances* ». *Proceedings of the CHI '91 Conference on Human Factors in Computing Systems*, 79-84.
10. **Gero, J. S.** (1990). « *Design Prototypes: A Knowledge Representation Schema for Design* ». *AI Magazine*, 11(4), 26-36.
11. **Goel, V.** (1995). *Sketches of Thought*. MIT Press.
12. **Janssen, Patrick, et Rudi Stouffs** (2015). « *Types of Parametric Modelling* », 157-66. Daegu, Taiwan.
13. **Keleşoğlu, Muhammet, et Derya Gulec Ozer** (2021). « *A Study on Digital Low Poly Modeling Methods as an Abstraction Tool in Design Processes* ». *Civil Engineering and Architecture*, 9(5), 153-169.
14. **Kvan, T.** (2004). « *The Dual Heritage of Design* ». *Automation in Construction*, 13(4), 403-415.
15. **Lawson, B.** (2005). *How Designers Think: The Design Process Demystified*. Architectural Press.

16. **Lecourtois, Caroline** (s. d.). « Complexité architecturale et assistance informatique ».
17. **Nykänen, Sami** (s. d.). « Blender in Architectural Modeling ».
18. **Olivier, Pauline, et al.** (2019). « Nested Explorative Maps: A new 3D canvas for conceptual design in architecture ». *Computers & Graphics*, 82, 203-13.
19. **Oxman, R.** (2006). « Theory and Design in the First Digital Age ». *Design Studies*, 27(3), 229-265.
20. **Pallasmaa, J.** (2009). *The Thinking Hand: Existential and Embodied Wisdom in Architecture*. John Wiley & Sons.
21. **Robertson, B. F., & Radcliffe, D. F.** (2009). « Impact of CAD tools on creative problem solving in engineering design ». *Computer-Aided Design*, 41(3), 136-146.
22. **Salim, Flora Dilys, et Jane Burry** (2010). « Software Openness: Evaluating Parameters of Parametric Modeling Tools to Support Creativity and Multidisciplinary Design Integration ». *Computational Science and Its Applications - ICCSA 2010*, 6018, 483-97.
23. **Sommer, E., et al.** (2024). « Crafting and Modifying Rhine Castle Models with Parametric Modeling in Blender ». *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2-W4-2024, 405-12.
24. **Spallone, R., et al.** (2019). « 3D Modelling Between Ideation, Geometry, and Surveyed Architecture... ». *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2-W15, 1119-26.
25. **Stals, Adeline** (s. d.). « Pratiques numériques émergentes en conception architecturale dans les bureaux de petite taille ».
26. **Turrin, Michela, et al.** (2011). « Design Explorations of Performance Driven Geometry in Architectural Design Using Parametric Modeling and Genetic Algorithms ». *Advanced Engineering Informatics*, 25(4), 656-75.
27. **Whitehead, H. / Foster + Partners** (2006). « Laws of Form ». *Catalogue of work*.
28. **Gibson, J.J.** (1979) : Concept original de l'affordance dans la psychologie écologique.
29. **Hausman, C.** (1976) : Théorie de la "nouveauauté authentique" en tant que signe de créativité.
30. **Boden, M.** (1998) : Distinction entre créativité historique (H-Creativity) et psychologique (P-Creativity).
31. **Runco, M.** (2007) : Analyse de la pensée créative non-linéaire et de l'originalité cognitive.
32. **Seligman, M.** (1967, 1972) : Théorie de l'incompétence apprise (*learned helplessness*).
33. **Simon, H.** (1969) : Modèle historique linéaire "Problème-Solution".