COLLANA
3D MODELING & BIM

INFORMATION & 3D MODELING PER IL PATRIMONIO COSTRUITO

A CURA DI TOMMASO EMPLER, ADRIANA CALDARONE,
ELENA D'ANGELO, ALEXANDRA FUSINETTI

dei
TIPOGRAFIA DEL GENIO CIVILE

Dipartimento di Storia
Disegno e Restauro
dell'Architettura

Sapienza
Università di Roma

TIPOGRAFIA DEL GENIO CIVILE

3D Modeling & BIM 2022 - Information & 3D Modeling per il patrimonio costruito

Curatori: Tommaso Empler, Adriana Caldarone, Elena D'Angelo, Alexandra Fusinetti

Collana: 3D Modeling & BIM

Publisher: DEI s.r.l.Tipografia del Genio Civile

ISBN 979-12-5505-080-3

DEI s.r.l. TIPOGRAFIA DEL GENIO CIVILE
Via Cavour 181/A - 00184 Roma
Tel. 06.441.63.71 (r.a.) Fax 06.440.33.07
dei@build.it
www.build.it

* DEI s.r.l. TIPOGRAFIA DEL GENIO CIVILE fa parte di LSWR GROUP

Credit immagine di copertina: G. Amoruso, G. Buratti

KEYNOTE SPEAKER

Alcune riflessioni sul linguaggio di programmazione visiva e il suo apprendimento

Some thoughts on visual programming language and its learning

Michele Calvano
ISPC-CNR

michele.calvano@ispc.cnr.it

La pandemia iniziata nell'anno 2019 a causa del virus SARS-CoV-2 non ha interrotto il *workshop* 3D Modeling & BIM che da allora si svolge in modalità telematica. Quest'anno nell'edizione dedicata alla "*Information e 3d Modeling per il patrimonio costruito*", ho avuto il piacere di partecipare come *keynote speaker* nelle vesti di ricercatore e professionista esperto dei processi di programmazione visuale per l'architettura ed il *design*.

Il mio intervento ha voluto sottolineare quanto oggi *Building Information Modeling* sia un grande contenitore di conoscenza capace di coniugare le diverse discipline che gravitano intorno al vasto tema della modellazione, in particolare per il costruito storico. La mia esperienza nel campo della rappresentazione digitale, mi consente di affermare che in alcuni casi gli strumenti digitali di disegno e di arricchimento informativo non rispondono adeguatamente alle necessità dei diversi domini di conoscenza; tuttavia associando ai *software* BIM strumenti di programmazione informatica, è possibile personalizzare delle procedure per tessere relazioni tra le figure impegnate nella complessa rappresentazione del costruito storico. Da circa quindici anni mi occupo di rappresentazione digitale nell'ambito dell'architettura e del design, prediligendo temi di ricerca legati all'indagine della forma e all'arricchimento informativo attraverso l'uso della programmazione visuale (VPL) per creare prototipi di procedure adatti a gestire modelli 3D. Wikipedia definisce il VPL nel seguente modo:

"... linguaggio che consente la programmazione tramite la manipolazione grafica degli elementi e non tramite sintassi scritta..." (textual programming language o TPL) "…. Un VPL consente di programmare con "espressioni visuali" ma anche all'evenienza di inserire spezzoni di codice … La maggioranza dei VPL è basata sull'idea "boxes and arrows" ovvero le "box" (o i rettangoli le circonferenze ecc...) sono concepiti come funzioni connesse tra di loro da "arrows", le frecce" (Visual Programming Language, 2021).

Nel 2020 sono risultato vincitore di un assegno per attività di ricerca nell'Istituto di Scienze del Patrimonio Culturale del Consiglio Nazionale delle Ricerche (ISPC – CNR). L'istituto

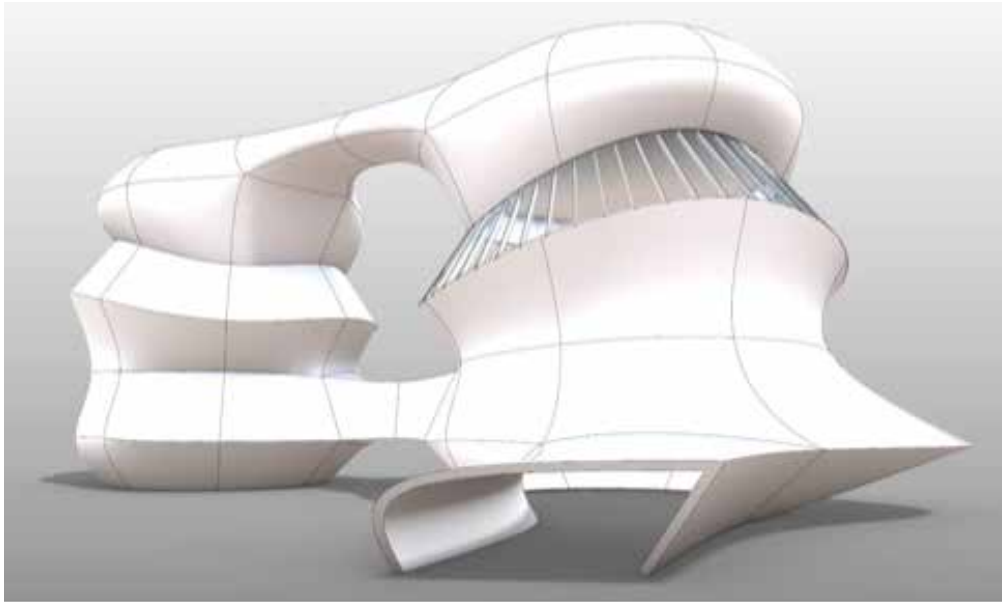


Fig. 1 – Disegno di un'architettura fantastica utilizzando le superfici Sub-D.

Fig 1 – Drawing of a fantastic architecture using Sub-D surfaces.

ospita diversi laboratori tra cui il *Built Heritage innovation Lab* (BHiLab). Il laboratorio è una struttura interdisciplinare che unisce ricercatori con diversa formazione, attivi nell'applicazione di competenze scientifiche ed umanistiche per migliorare la conoscenza e la conservazione del patrimonio costruito; il gruppo è impegnato nell'innovazione di procedure di analisi e di intervento integrate a metodologie di Information and *Communications Technology* (ICT). Nel laboratorio, tra le diverse attività, sto sviluppando processi VPL orientati alla modellazione informata del patrimonio costruito relazionando il *Visual Programming Language* ed il *Building Information Modeling* (BIM).

L'applicazione di originali procedure VPL, mi ha permesso di delineare nel tempo un percorso di ricerca e sperimentazione con cui evidenziare l'utilità del processo di programmazione per la modellazione del paesaggio, lo spazio urbano, l'architettura ed il prodotto. La scelta del comitato scientifico di iniziare i lavori del *workshop* "3D Modeling & BIM 2022" con il mio contributo, esprime il pensiero di una parte della scuola romana del Disegno che riconosce nella programmazione visuale un supporto alla rappresentazione della complessità del paesaggio antropico e naturale italiano.

Il percorso d'apprendimento

Come sono venuto a conoscenza del VPL?

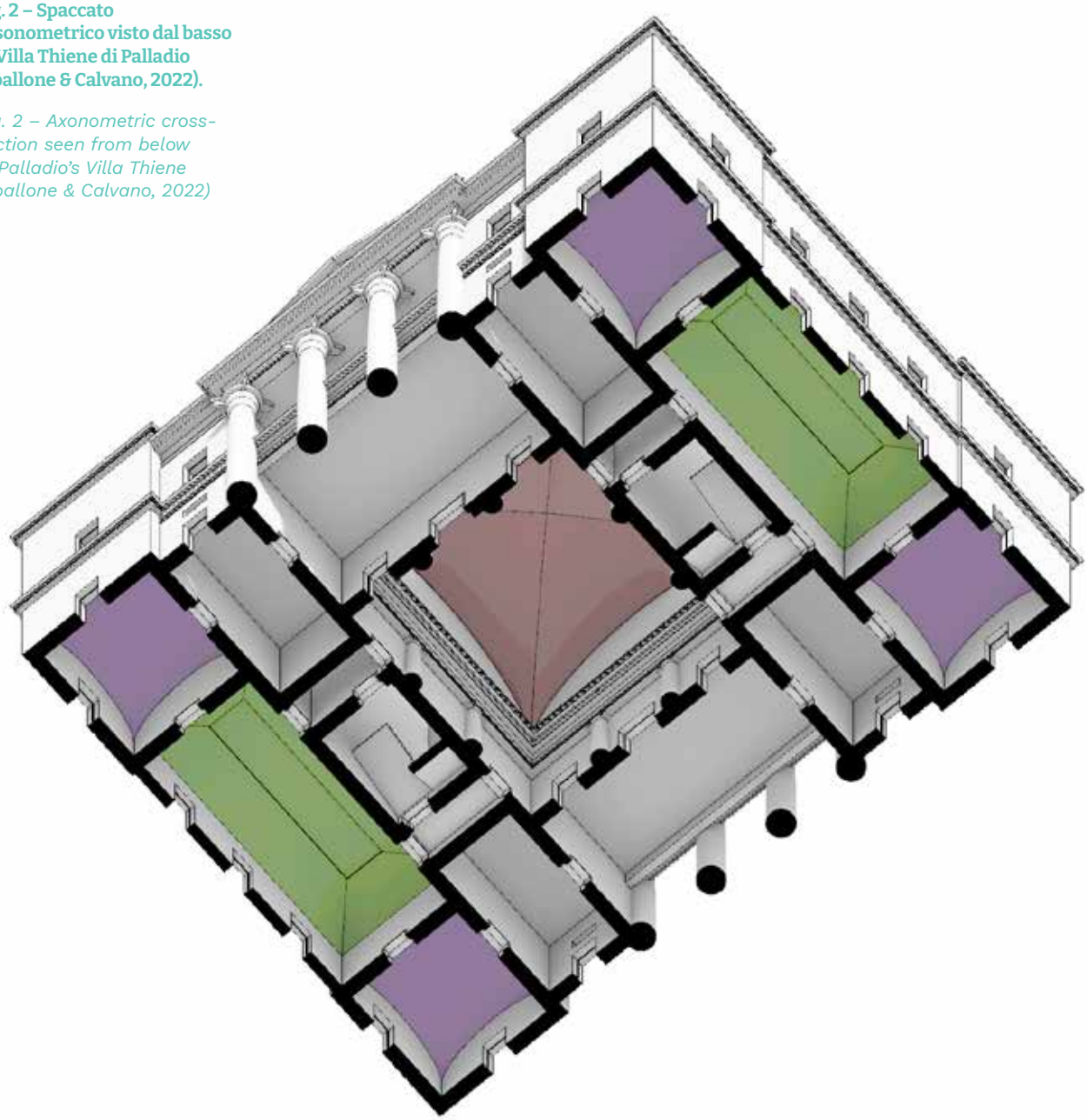
L'apprendimento è iniziato subito dopo il conseguimento della laurea in architettura, parallelamente al percorso di ricerca intrapreso nel dottorato di ricerca in Scienze della Rappresentazione all'Università di Roma Sapienza. Ricordo chia-

The pandemic, that started in the year 2019 due to the SARS-CoV-2 virus, did not interrupt the 3D Modeling & BIM workshop, which has been held telematically ever since. This year, in the edition dedicated to "Information and 3D Modeling for the Built Heritage", I had the pleasure of participating as keynote speaker in the role of researcher and professional expert in visual programming processes for architecture and design.

My speech aimed to emphasise how Building Information Modeling (BIM) today is a great container of knowledge, capable of combining the various disciplines that gravitate around the vast theme of modelling, particularly for historical buildings. My ever-increasing experience in the field of digital representation allows to affirm that, in some cases, digital drawing and information enrichment tools do not adequately respond to the needs of different knowledge domains; however, by associating BIM software with computer programming tools, it is possible to customise procedures to create relationships between the experts involved in the complex representation of histo-

Fig. 2 – Spaccato assonometrico visto dal basso di Villa Thiene di Palladio (Spallone & Calvano, 2022).

Fig. 2 – Axonometric cross-section seen from below of Palladio's Villa Thiene (Spallone & Calvano, 2022)



ramente uno dei primi collegi di dottorato nel 2010, quando un collega dottorando presentò alcuni esercizi di Geometria Descrittiva affrontati digitalmente con un sistema di programmazione visuale[1] ancora in fase di sviluppo.

L'interfaccia del *software* all'epoca era molto acerba, e lo strumento veniva utilizzato creando dei codici complessi dovuti all'inconsapevolezza di una visione principalmente legata alla necessità di esprimere un oggetto anziché all'indagine

rical buildings. For about fifteen years I have been working on digital representation in the field of architecture and design, with a preference for research topics related to the investigation of form and information enrichment through the use of visual programming (VPL) to create prototype procedures suitable for handling 3D models. Wikipedia defines VPL as follows:

"... is any programming language

1 *Il sistema utilizzato era Grasshopper in una delle sue prime relase. Nel 2010 questa applicazione era ancora una plug-in di Rhinoceros, software CAD della McNeel, non ancora perfettamente implementato.*

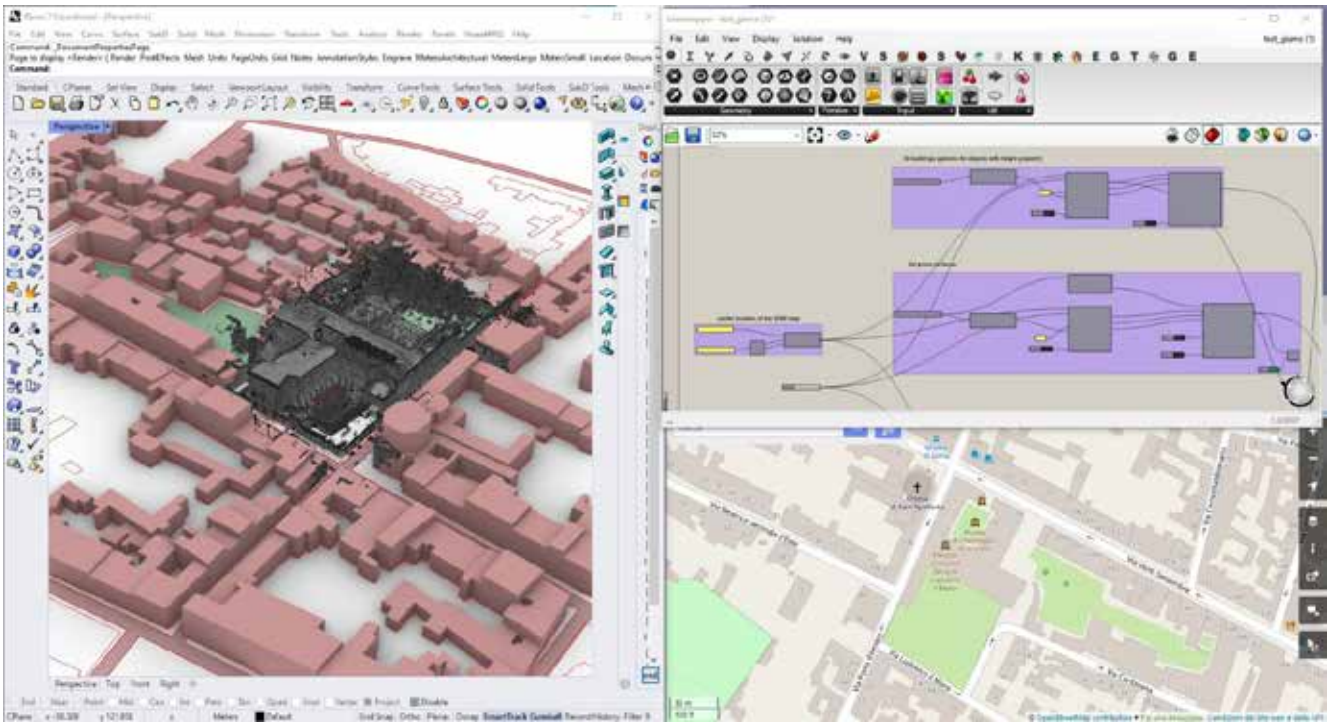


Fig. 3 – Modello 3D ricavato da informazioni Web-GIS unite in un secondo momento alla nuvola di punti generata da rilievo mediante processo SLAM[2]

Fig. 3 – 3D model derived from Web-GIS information later merged with the point cloud generated by surveying using the SLAM[1] process

del processo costruttivo. All'epoca il sistema di programmazione era utilizzato in maniera "muscolare" vista l'assuefazione diffusa per elefantiaci grovigli di connessioni che alla fine partorivano "modellini".

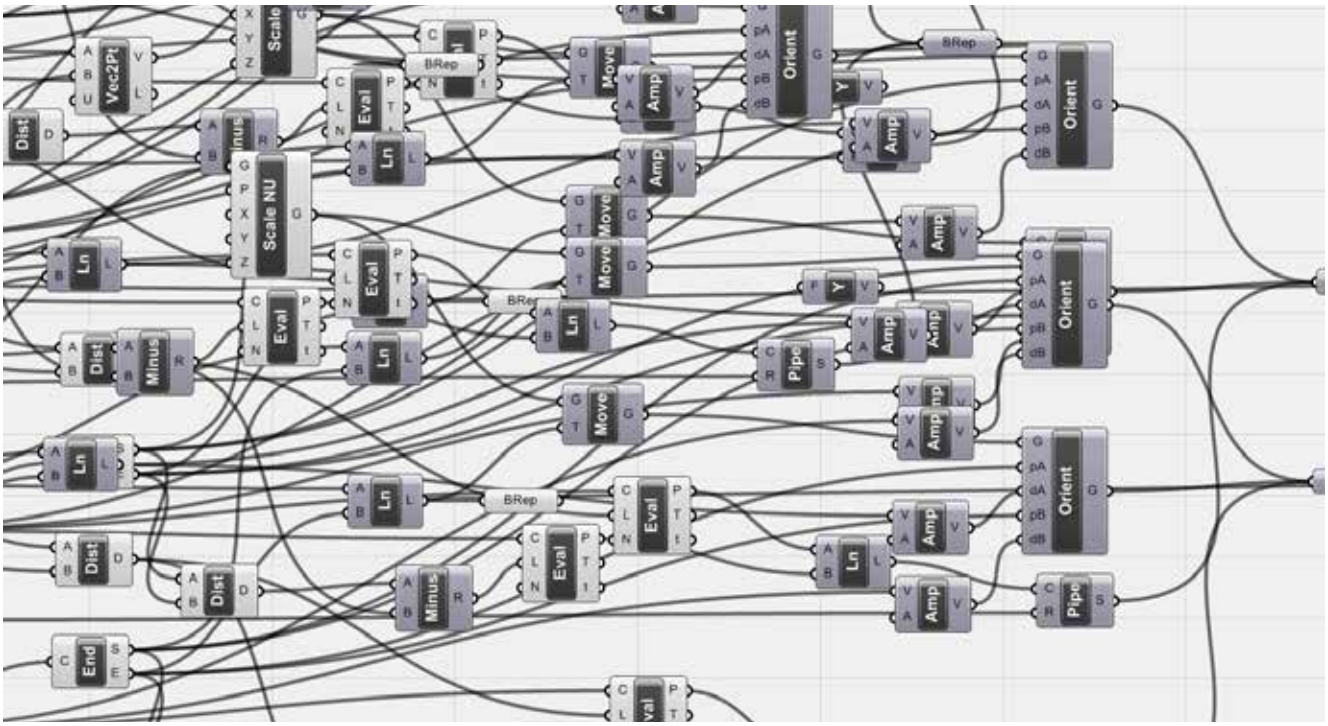
Osservando i risultati di quelle prime sperimentazioni non si poteva certamente individuare il potenziale del linguaggio legato a criteri generativi, costruttivi e responsivi (Calvano & Casale, 2018). Nell'occasione del collegio di dottorato il risultato prevedibile raggiunto dal collega è stato il disappunto di buona parte degli uditori. Malgrado il diffuso scetticismo, quella presentazione ha avuto il merito di farmi conoscere un nuovo strumento di costruzione ed analisi dei modelli. Prima di allora pensavo al disegno digitale come semplice supporto alla rappresentazione mediante elaborati grafici; il disegno era quindi strumento di indagine, comprensione, comunicazione ed ingegnerizzazione del prodotto finale (progetto o rilievo).

L'intervento al collegio di dottorato aveva innescato nella mia mente un processo narrativo che in maniera esplicita rendeva evidenti i processi costruttivi alla base del disegno dove gli *input* si legano agli *output*, diventando essi stessi *input*. Si esprime così un processo evolutivo sempre modificabile nei passi del percorso (Calvano et al., 2020). Negli anni successivi il mio interesse si è spostato dall'estetica del prodotto, al progetto del processo generativo con cui ottenere il prodotto stesso. Questo tipo di pensiero è stato utile ad alimentare l'attività di semantizzazione del disegno digitale

2 Attività svolta nell'ambito del PRIN TECH-START all'interno del BHiLab dell'ISPC-CNR. Soggetto di studio: Palazzo Costabili di Ferrara.

that lets users create programs by manipulating program elements graphically rather than by specifying them textually. A VPL allows programming with visual expressions, spatial arrangements of text and graphic symbols, used either as elements of syntax or secondary notation. For example, many VPLs (known as dataflow or diagrammatic programming) are based on the idea of "boxes and arrows", where boxes or other screen objects are treated as entities, connected by arrows, lines or arcs which represent relations." (Visual Programming Language, 2021).

1 Activities carried out within the framework of the PRIN TECH-START within the BHiLab of the ISPC-CNR. Study subject: Palazzo Costabili in Ferrara.



con cui la scuola romana di rappresentazione stava facendo i conti, impegnata in un importante attività di rinnovamento della geometria descrittiva (Carlevaris et al., 2012).

L'obiettivo del processo di rinnovamento era la rivisitazione degli aspetti non solo metodologici ma anche pedagogici dei metodi di rappresentazione e del loro insegnamento nelle facoltà di architettura avvalendosi delle potenzialità espresse dalla modellazione 3D. Tra gli obiettivi vi era la creazione di una continuità tra il passato ed il futuro delle discipline della rappresentazione da tempo scosse dall'avvento del digitale che fino a quel momento, a mio avviso, stava svolgendo una superficiale azione di *maquillage* dei prodotti (*output*) di disegno. Solo ora, a distanza di anni, posso dire che una completa rivoluzione avviene solo se, all'apprendimento dei metodi di rappresentazione classici e digitali, si unisce l'addestramento al "pensiero algoritmico" unito ad una diretta manipolazione degli applicativi di disegno forniti dal computer; operazione possibile mediante l'utilizzo di linguaggi informatici.

A seguito di queste riflessioni, parallelamente agli studi e alle ricerche proposte dal dottorato ho iniziato un percorso di apprendimento incentrato alla conoscenza del VPL. Mentre il percorso accademico proponeva temi di ricerca avanzati relativi ai diversi ambiti della rappresentazione, con i miei studi affrontavo il processo di apprendimento di un nuovo linguaggio allo stesso modo di un neonato che impara la lingua nativa. Il mio interesse è stato rivolto sin dall'inizio a sistemi grafici di programmazione capaci di supportare le discipline architettoniche. L'apprendimento è iniziato in maniera spon-

Fig. 4 – La scrittura VPL predilige codici efficienti, evitando strutture incomprensibili (Davis, 2010)

Fig. 4 – VPL writing prefers efficient codes, avoiding incomprehensible structures (Davis, 2010)

In 2020, I won a grant for research activities at the Institute of Heritage Science of the National Research Council (ISPC - CNR). The institute hosts several laboratories including the Built Heritage innovation Lab (BHiLab). The lab is an interdisciplinary structure that hosts researchers with different backgrounds, operating in the field of applying scientific and humanistic skills to improve knowledge and conservation of built heritage; the group is focused on the innovation of analysis and intervention procedures integrated with Information and Communications Technology (ICT) methodologies. In the laboratory, among the various activities, I am developing VPL processes oriented to the informed modelling of built heritage by relating Visual Programming Language and Building Information Modeling.

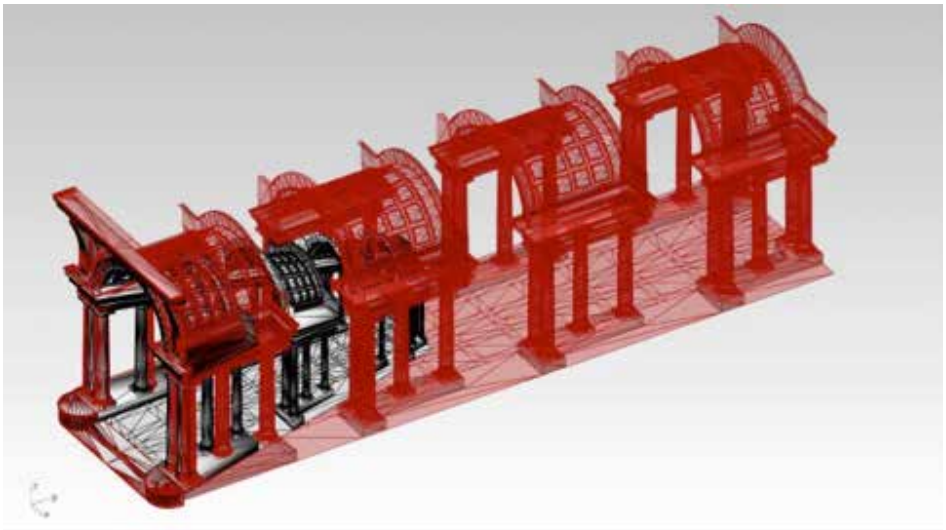


Fig. 5 – Sperimentazioni di prospettiva solida mediante l'utilizzo del Visual Programming Language. Caso studio: Galleria Spada a Roma (Migliari et al., 2014).

Fig. 5 – Experiments in solid perspective using Visual Programming Language. Case study: Spada Gallery in Rome (Migliari et al., 2014).

tanea, senza ricevere, se non saltuariamente, insegnamenti specifici e diretti da parte di esperti, che in quel periodo significava confrontarsi con i pochi professionisti direttamente coinvolti nello sviluppo dei linguaggi VPL di mio interesse. Questi proponevano *workshop* incentrati sulle potenzialità del linguaggio in merito a tematiche specifiche, come potevano essere processi di ottimizzazione della topologia per la generazione delle forme oppure analisi ambientale per il supporto alla progettazione architettonica ed urbana; poco veniva fatto invece per l'apprendimento della sintassi. Dopo aver preso confidenza con la logica e le interfacce del VPL di mio interesse, ho cominciato ad affrontare problematiche geometriche sempre più complesse in modo da approfondire i metodi di scrittura algoritmica e le basi del linguaggio.

Pensiero algoritmico VS pensiero computazionale

Precedentemente ho parlato di "pensiero algoritmico"; con questo termine si intende la capacità di riconoscere aspetti della computazione nel mondo che ci circonda per associarli a tecniche informatiche con il fine di capire e ragionare su sistemi e processi naturali, sociali e artificiali (Wing, 2006). Concetto che si sposa perfettamente con le scienze della rappresentazione, per cui uno degli obiettivi è la discretizzazione del reale attraverso l'utilizzo di processi proiettivi e l'utilizzo di codici grafici condivisi per la costruzione di modelli.

L'avvento dei sistemi CAD porta il processo algoritmico, comunque intrinseco alla geometria descrittiva (Calvano et al., 2020), in ambito digitale; ogni singolo segno è caratterizzato da micro azioni provviste di *input* (dati in ingresso) ed *output* (elemento oggetto del processo). Il pensiero algoritmico assume anche un ruolo pedagogico, permettendo agli stu-

The application of original VPL procedures has allowed me to outline over time a path of research and experimentation that highlights the usefulness of the programming process for the modelling of landscape, urban space, architecture and product. The Scientific Committee's choice to start the activities of the workshop "3D Modeling & BIM 2022" with my contribution expresses the thought of a part of the Roman School of Drawing, that recognises visual programming as a support to the representation of the complexity of the Italian anthropic and natural landscape.

The learning process

How did I learn about VPL? Learning started immediately after graduating in architecture, in parallel with the research path undertaken in the PhD programme in Representation Sciences at Sapienza - Università di Roma. I clearly remember one of the first doctoral colleges in 2010, when a fellow doctoral student presented some exercises in Descriptive Geometry that were tackled digitally with a visual programming system that was still under development[2]. At the time,

2 *The system used was Grasshopper in one of*

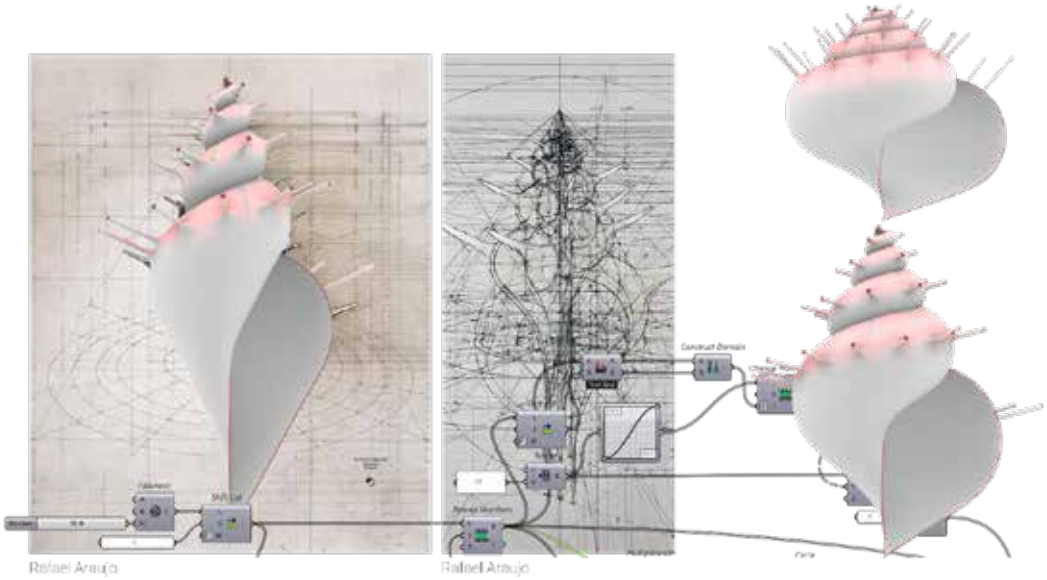


Fig. 6 – Esplicitazione di processi costruttivi in VPL per la modellazione di una conchiglia a partire dai processi algoritmici di Rafael Araujo.

Fig. 6 – Explicitation of construction processes in VPL for modelling a shell from Rafael Araujo's algorithmic processes.

denti di affrontare problemi, di scomporli in parti risolvibili e di elaborare processi digitali per risolverli. Il pensiero algoritmico diventa in questo modo "pensiero computazionale" che si definisce come insieme di processi mentali coinvolti nel formulare e risolvere problemi; la rielaborazione deve avvenire in modo che il processo possa essere rappresentato in una forma eseguibile da una procedura digitale (Wing, 2011).

La soluzione può essere eseguita da un essere umano o da una macchina, o dalle combinazioni di uomini e macchine. Anche il pensiero computazionale si concentra sul processo, non sulla produzione di artefatti o di prove. Questo tipo di attività permette di osservare le rappresentazioni con l'intento di cogliere la genesi costruttiva dei disegni. L'addestramento al pensiero computazionale consente di cogliere il processo algoritmico implicito durante la costruzione dei modelli, per cui le operazioni costruttive sono connesse l'un l'altra senza soluzione di continuità; l'azione si conclude con la digitalizzazione del percorso creando uno strumento risolutivo del problema e, nel migliore dei casi, condivisibile. In ambiente digitale le procedure di costruzione di immagini e modelli non sono univoche, poiché un obiettivo è comunque perseguibile seguendo percorsi differenti (Nerdinger & others, 2005), sottolineando una quota di creatività nell'ideazione del processo e non solo nel prodotto finale, il modello.

Il visual programming language

Al processo algoritmico di riduzione dei progetti complessi, segue la capacità di digitalizzare le azioni mediante approcci informatici che si dividono in due tipi:

- digitalizzazione degli algoritmi tramite linguaggio di programmazione testuale (TPL);

the software interface was very primitive, and the tool was used for creating complex codes, due to the unawareness of a vision mainly focused on expressing an object rather than investigating the construction process. The programming system was used in a 'muscular' manner, resulting in the proliferation of elaborate tangles of connections that, in the end, culminated in 'rough models'.

Looking at the results of those early experiments, it was certainly not possible to identify the potential of language linked to generative, constructive and responsive criteria (Calvano & Casale, 2018). On the occasion of the doctoral college, the predictable result achieved by the colleague was the disappointment of a large part of the audience. Despite the widespread scepticism, that presentation had the merit of introducing me to a new tool for constructing and analysing models. Before that time, I thought of digital drawing as a simple support to representation by means of graphic works; drawing was therefore a tool for investigation, understanding, com-

its first relases. In 2010, this application was still a plug-in for Rhinoceros, CAD software from McNeel, which was not yet fully implemented.

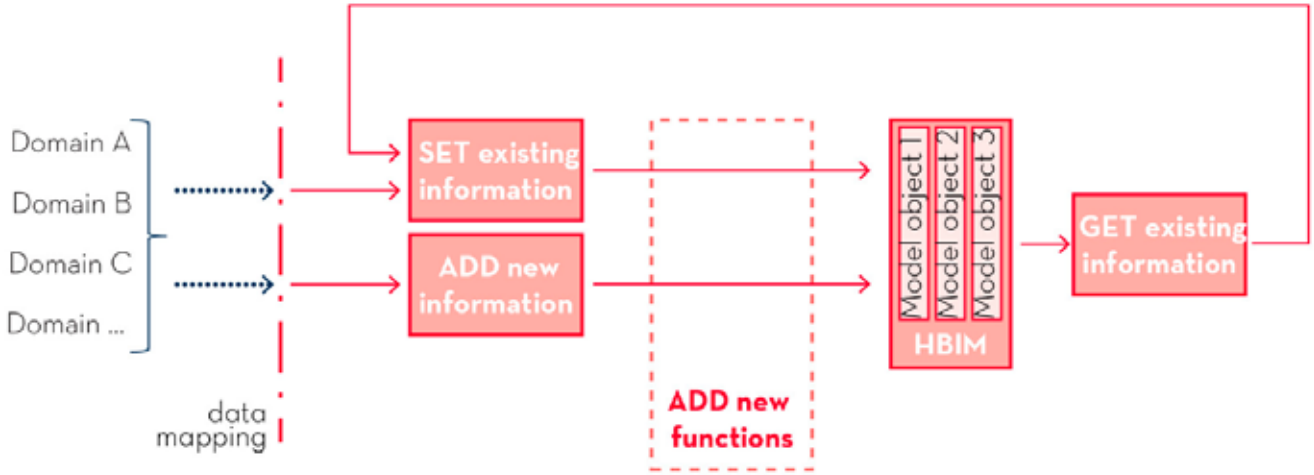


Fig. 7 – Schematizzazione del processo di arricchimento informativo nell'ambito HBIM per mezzo del rapporto VPL-BIM (Calvano et al., 2022).

Fig. 7 – Schematisation of the information enrichment process in HBIM by means of the VPL-BIM connection (Calvano et al., 2022)

- digitalizzazione degli algoritmi tramite linguaggio di programmazione visiva (VPL).

Nel primo caso, l'utente utilizza il TPL per scrivere programmi eseguibili, seguendo le regole del linguaggio scelto (C #, Python, VBScript, ecc.). Questo tipo di programmazione richiede un alto livello di conoscenza informatica (Zabramski et al., 2013), motivo per cui è difficilmente utilizzabile da chi ha seguito un percorso di studi principalmente legato alle arti visuali: le scuole di architettura e *design* non hanno ancora introdotto in maniera curriculare insegnamenti di informatica, se non inseriti autonomamente da alcuni docenti nei propri corsi.

Il VPL rende più accessibile la digitalizzazione degli algoritmi a professionisti che non conoscono le basi di programmazione testuale, ma che invece hanno familiarità con la progettazione visiva. Questo perché il VPL ha una modalità di scrittura che si affida a nodi grafici, facendo leva sulla generale comprensione di strutture a diagramma di flusso, che da sempre risulta essere uno strumento utilizzato per la schematizzazione di processi progettuali. L'interfaccia di programmazione agevolata trova però alcuni limiti nell'accesso alle risorse fornite dalle architetture dei sistemi *hardware*. Questo tipo di linguaggio di programmazione richiede un livello medio di conoscenza informatica.

La programmazione informatica comporta l'applicazione di una serie preimpostata di regole correlate che introducono vincoli formali, dimensionali e dati (parametri); la manipolazione di alcuni dati consente la variazione di modelli in maniera controllata (Caetano et al., 2020). Quando si modella l'architettura storica, l'approccio parametrico mira a combinare vincoli e misure variabili per migliorare la rappresentazione geometrica (Diara & Rinaudo, 2019; Osello et al., 2018) degli elementi che costituiscono gli edifici e a integrare il contenuto informativo attraverso l'ideazione di appropriati parametri (Garagnani, 2013).

munication and engineering of the final product (project or survey).

The talk at the doctoral college triggered a process in my mind that explicitly made evident the constructive workflows underlying the design where inputs link to outputs, becoming inputs themselves. Thus, an evolutionary process is expressed, that is always modifiable at any step of the path (Calvano et al., 2020). In later years, my interest shifted from the aesthetics of the product to the design of the generative process by which to obtain the product itself. This kind of thinking was useful in feeding the digital drawing semantization activity undertaken by the Roman school of representation, engaged in a major renewal of descriptive geometry (Carlevaris et al., 2012).

The goal of the renewal process was to revisit not only methodological but also pedagogical aspects of representation methods and their teaching in the faculties of architecture, by ma-

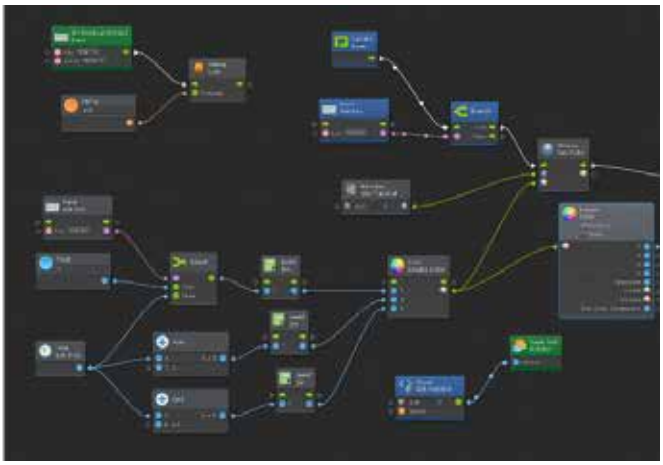


Fig. 8 – TPL vs VPL

Fig. 8 – TPL vs VPL

I linguaggi di programmazione consentono di accedere a librerie provenienti da diverse discipline, per cui i *software* di modellazione che utilizzano TPL e VPL sono capaci di introdurre nel modello operazioni per la costruzione e l'arricchimento informativo, agevolando il lavoro di professionisti e ricercatori interessati all'utilizzo del modello. La manipolazione di codici consente di automatizzare processi complessi fino a creare prototipi di *software* con cui risolvere problemi disciplinari (Calvano et al., 2022). Nel mio caso tale operazione è stata spesso assolta attraverso l'utilizzo del VPL.

I linguaggi visivi più diffusi si basano su interfacce grafiche *user-friendly*, per "disegnare" il processo utilizzando semplici geometrie grafiche: rettangoli (nodi) e curve orientate (connessioni). I nodi identificano la parte minima di codice che genera un'azione. Il nodo è provvisto di *input* e *output* che raccolgono, elaborano e inviano informazioni; il flusso di dati è organizzato utilizzando connessioni rappresentate da entità monodimensionali (frecce, linee, archi, ecc.) (Kuhail et al., 2021). Il VPL è impiegato in molte aree del dominio digitale: educativo, multimediale, videogiochi, automazione, modellazione 3D, ecc. Gli ambienti VPL possono far parte di *software* specializzati (ad esempio, *software* di modellazione 3D, *software* di *rendering*, ecc.) per accedere e implementare le risorse del programma a cui appartengono, oppure possono essere piattaforme autonome che forniscono un'interfaccia più semplice per la programmazione generica; attività ricorrente nel campo dell'istruzione (Tsai, 2019).

I benefici del VPL sono valutabili considerando le caratteristiche che contraddistinguono tutti i linguaggi: sintassi, semantica e pragmatica (Morris, 1938), aggiungendo anche un'ultima caratteristica definita implementazione. Di seguito una breve definizione alle caratteristiche in relazione alla programmazione visuale.

- Sintassi: il VPL ha una sintassi semplificata poiché le relazioni tra segni (nodi) sono delegate a connessioni

king use of the potential expressed by 3D modeling. Among the objectives was the creation of a continuity between the past and the future of representation topics, which had long been shaken by the advent of digital tools, but which, up to that time, in my opinion, were performing a superficial makeup of drawing products (outputs). Only now, years later, I can say that a complete revolution takes place only if one combines training in "algorithmic thinking" with direct manipulation of digital drawing applications; an operation possible through the use of computer languages.

As a result of these considerations, in parallel with the studies and research proposed by the doctoral program, I began a learning path focused on the knowledge of VPL. While the academic syllabus proposed advanced research topics related to different areas of representation, with my studies I was approaching the process of learning a new language in the same way as an infant learning its native language. From the beginning, my interest was directed toward graphical programming systems capable of supporting architectural disciplines. Learning

unidimensionali orientate (curve) che controllano il flusso di informazioni in entrata e in uscita dai nodi.

- Semantica: il VPL consente la disambiguazione semantica, grazie ad un ricco sistema di annotazione di cui i nodi dispongono. In generale, ogni componente grafico è arricchito con informazioni o collegamenti alla documentazione che spiega come funziona il componente.
- Pragmatica: ogni nodo grafico nel linguaggio corrisponde ad un'azione all'interno del programma: quindi, esiste una relazione diretta tra i segni ed i risultati. Diversi nodi collegati attivano una serie di elementi computazionali che influenzano l'efficienza del rapporto tra azione (eseguita dal programmatore) e reazione (risposta del dispositivo programmato).
- Implementazione: i linguaggi di programmazione dovrebbero essere facilmente modificati, anche nel tempo, rispettando le regole semantiche; questa condizione consente l'arricchimento pragmatico del linguaggio e migliora la creatività nell'algoritmo.

L'approfondimento del linguaggio computazionale del VPL e la sperimentazione di processi legati ai metodi di rappresentazione mi ha permesso di coniare un nuovo termine, Disegno Digitale Esplicito:

"Con Disegno Digitale Esplicito (DDE) intendiamo quella modalità di rappresentazione dei modelli per cui l'attenzione del disegnatore è rivolta non solo al risultato formale, ma anche alle procedure che hanno generato l'immagine. L'equilibrio tra procedura e prodotto mette in evidenza nuove modalità creative che si manifestano tanto nel prodotto visuale, quanto nelle procedure che generano il modello. La capacità di costruire relazioni efficienti tra dati e informazioni collegate ad un modello visuale, è uno degli scopi del DDE" (Calvano, 2019).

Conclusioni

A questo punto della trattazione è importante riconoscere nel modello 3D, che sia per l'architettura o il *design*, il processo algoritmico che l'ha generato e renderlo esplicito. Il codice generativo dei modello è una sorta di mappa genetica del prodotto creato, che se modificato in alcuni suoi dati può diventare responsivo, introducendo le caratteristiche cinematiche nel modello.

L'apprendimento del *visual programming language* associato al tema della rappresentazione presuppone la conoscenza dei seguenti ambiti:

- la geometria descrittiva;
- la rappresentazione digitale;
- il pensiero algoritmico.

started spontaneously, without receiving, except occasionally, specific and direct teaching from experts, which at that time meant dealing with the few professionals directly involved in the development of the VPL languages of my interest. These offered workshops focused on the language's potential for specific topics, such as topology optimization processes for shape generation or environmental analysis for architectural and urban design support; little was done, however, for syntax learning. After becoming familiar with the logic and interfaces of the VPL of my interest, I began to tackle increasingly complex geometric problems in order to deepen my understanding of algorithmic writing methods and the basics of the language.

Algorithmic thinking VS computional thinking

Previously I spoke of 'algorithmic thinking', by which I mean the ability to recognise aspects of computation in the world around us in order to associate them with computer techniques, with the aim of understanding and reasoning about natural, social and artificial systems and processes (Wing, 2006). A concept that fits perfectly with the sciences of representation, for which one of the objectives is the discretization of reality through the use of projective processes and the use of shared graphic codes for the construction of models.

The advent of CAD systems brought the algorithmic process, however intrinsic to descriptive geometry (Calvano et al., 2020), into the digital domain; each sign is characterised by micro-actions with 'input' (input data) and output (the element that is the object of the process). Algorithmic thinking also takes on a pedagogical role by allowing students to tackle problems, break them down into sol-

Fig. 9 – Copertina del libro Disegno Digitale Esplicito edito da Aracne.

Fig. 9 –

La geometria descrittiva è da sempre una delle materie chiave nelle facoltà d'architettura e *design*; insegnare oggi questa materia nella maniera delineata da alcuni docenti romani, consente di creare una continuità tra passato e futuro, donando alla disciplina una rinnovata vitalità. La conoscenza delle entità fondamentali del disegno e dei metodi di indagine delle relazioni esistenti tra essi, permette di decostruire i modelli digitali e connettere le entità minime tramite codifica informatica.

Il campo grafico in cui restituire le relazioni non è più il foglio di carta ma è lo schermo del *computer* dove le immagini vengono espresse mediante metodi di rappresentazione digitale (*mesh* e NURBS). La conoscenza degli ingredienti che costituiscono questi metodi di rappresentazione è indispensabile per una efficiente codifica degli algoritmi di disegno ed arricchimento informativo.

La capacità di esplicitazione del pensiero algoritmico è la parte che invece va potenziata nei percorsi di apprendimento

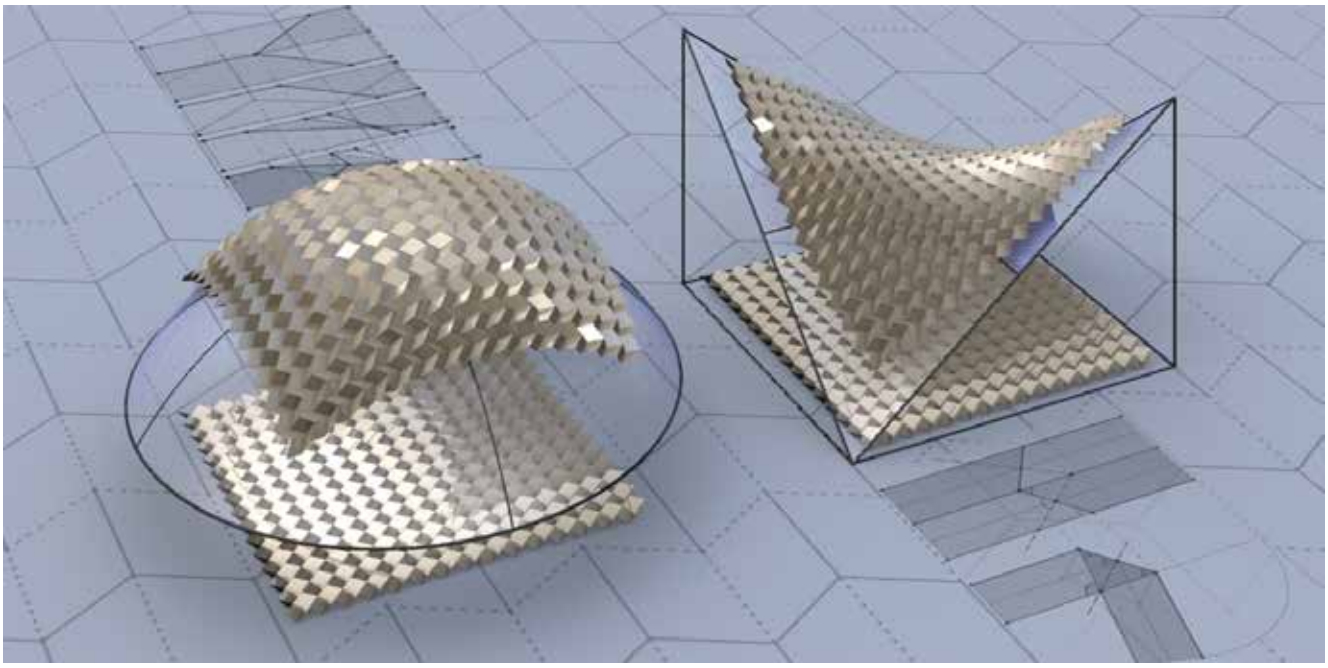
vable parts and process them digitally to solve them. Algorithmic thinking thus becomes 'computational thinking', which is defined as a set of mental processes involved in formulating and solving problems; the reworking must take place in such a way that the process can be represented in an executable form by a digital procedure (Wing, 2011).

The solution can be performed by a human or a machine, or by combinations of humans and machines. Computational thinking also focuses on process, not on the production of artefacts or evidence. This type of activity allows one to observe representations to understand the constructive genesis of designs. Training in computational thinking allows one to catch the implicit algorithmic process during the construction of the models, whereby the constructive operations are seamlessly connected; the action concludes with the digitisation of the path by creating a problem-solving and, at best, sharable tool. In the digital environment, image and model construction procedures are not standardised because an objective can still be achieved by following different paths (Nerdinger & others, 2005), emphasising a level of creativity in the design of the process and not only in the final product, the model.

The Visual Programming Language

The algorithmic process of reducing complex projects is followed by the ability to digitise actions through IT approaches that fall into two types:

- *digitisation of algorithms via*



universitario. Nel fare questo si deve partire dalla consapevolezza che qualunque immagine restituita mediante i metodi canonici di rappresentazione, si genera a seguito di un accumulo di segni costruttivi che definiscono l'immagine finale del modello. Le regole che definiscono il processo di costruzione costituiscono l'algoritmico che il disegnatore segue per ottenere l'immagine finale. Il pensiero algoritmico è quindi insito nei disegni che seguono le regole della geometria descrittiva; questa materia è quindi una palestra per l'apprendimento dei processi codificabili in VPL.

I processi BIM, che oggi sembrano i più appropriati per la descrizione dell'architettura, ricevono un grande apporto dal VPL, che integra i limiti dei BIM *modeler* nella restituzione di istanze architettoniche per il costruito storico e la deficienza informativa verso alcuni domini di conoscenza (Gigliarelli et al., 2016). L'integrazione dei due processi consente di effettuare valutazioni diagnostiche per la conoscenza, l'analisi e la prevenzione dei beni culturali. Introdurre il *workshop* 3D Modeling & BIM 2022 con un approfondimento sul VPL consente a professionisti e ricercatori di comprenderne l'importanza all'interno del panorama architettonico, ambito sempre più multidisciplinare e bisognoso di strumenti capaci di connettere competenze.

Fig. 10 – Superfici Piegate; adattamento di pattern noti a superfici geometriche (Calvano et al., 2019)

Fig. 10 – Folded surfaces; adaptation of known patterns to geometric surfaces (Calvano et al., 2019)

textual programming language (TPL);

- *digitisation of algorithms via visual programming language (VPL).*

In the first case, the user uses TPL to write executable programmes, following the rules of the chosen language (C #, Python, VBScript, etc.). This type of programming requires a high level of computer science knowledge (Zabramski et al., 2013), and is therefore difficult to use by designers who have followed a course of study mainly related to the visual arts: schools of architecture and design still do not integrate computer science into curricula, unless when introduced autonomously by some lecturers in their courses.

VPL makes the digitisation of algorithms more accessible to professionals who do not know the basics of

textual programming but are familiar with visual design. This is because VPL has a writing mode that is based on graphical nodes, leveraging the general understanding of flowchart structures, which has always been a tool used for the schematisation of design processes. However, the facilitated programming interface finds some limitations in accessing the resources provided by hardware system architectures. This type of programming language requires an average level of computer knowledge.

Computer programming involves the application of a predefined set of related rules that introduce formal, dimensional and data (parameter) constraints; the manipulation of certain data allows the variation of models in a controlled manner (Caetano et al., 2020). When modelling historical architecture, the parametric approach aims to combine constraints and variable measures to improve the geometric representation (Diara & Rinaudo, 2019; Osello et al., 2018) of the elements that constitute buildings and to integrate information content through the design of appropriate parameters (Garagnani, 2013).

Programming languages provide access to databases from different disciplines, so modelling software applications using TPL and VPL are able to introduce operations into the model for construction and information enrichment, facilitating the work of practitioners and researchers interested in using the model. Code manipulation makes it possible to automate complex processes to create software prototypes with which to solve disciplinary problems (Calvano et al., 2022). In my case, this was often accomplished through the use of VPL.

The most popular visual languages are based on user-friendly graphical interfaces to 'draw' the process using simple graphic geometries: rectangles (nodes) and oriented curves (connections). Nodes identify the smallest part of code that generates an action. The node has inputs and outputs that collect, process and send information; the data flow is organised using connections represented by one-dimensional entities (arrows, lines, arcs, etc.) (Kuhail et al., 2021). VPL is employed in many areas of the digital domain: educational, multimedia, video games, automation, 3D modelling, etc. VPL environments can be part of specialised software (e.g., 3D modelling software, rendering software, etc.) to access and implement the resources of the programme to which they belong, or they can be stand-alone platforms that provide a simpler interface for generic programming; a recurring activity in the field of education (Tsai, 2019).

The benefits of VPL can be estimated by considering the characteristics that define all languages: syntax, semantics and pragmatics (Morris, 1938), while also adding a final characteristic called implementation. Below is a brief definition of the characteristics in relation to visual programming.

- *Syntax: VPL has a simplified syntax, since the relationships between signs (nodes) are relegated to oriented one-dimensional connections (curves) that control the flow of information in and out of the nodes.*
- *Semantics: VPL allows semantic disambiguation, thanks to a rich annotation system that nodes have at their disposal. In general, each graphical component is enriched with information or links to documentation explaining how the component works.*
- *Pragmatics: each graphic node in the language corresponds to an action within the programme: thus, there is a direct relationship between signs and results. Several connected nodes activate a series of computational elements that affect the efficiency of the relationship between action (performed by the programmer) and reaction (response of the programmed device).*
- *Implementation: programming languages should be easily modified, even over time, respecting semantic rules; this condition allows for pragmatic enrichment of the language and improves creativity in the algorithm.*

Deepening the computational language of VPL and experimenting with processes related to representation methods has allowed me to coin a new term, Explicit Digital Drawing:

"With Explicit Digital Drawing (DDE), we mean a mode of representing models in which the draughtsman's attention is directed not only to the formal result, but also to the procedures

that generated the image. The balance between procedure and product highlights new creative modes that are manifested as much in the visual product as in the procedures that generate the model. The ability to construct efficient relationships between data and information linked to a visual model is one of the aims of the DDE" (Calvano, 2019).

Conclusions

At this point in the discussion, it is important to recognise in the 3D model, whether for architecture or design, the algorithmic process that generated it and to make it explicit. The code generating the model is a kind of genetic map of the created product, which, if modified in some of its data, can become responsive, introducing kinematic characteristics into the model.

Learning the visual programming language associated with the topic of representation presupposes knowledge of the following fields:

- *descriptive geometry;*
- *digital representation;*
- *algorithmic thinking.*

Descriptive geometry has always been one of the key subjects in faculties of architecture and design; teaching this subject today, in the manner outlined by several Roman lecturers, creates continuity between the past and the future, giving the discipline renewed vitality. The knowledge of the fundamental entities of design and the methods of investigating the relationships among them allows for the deconstruction of digital models and the connection of minimal entities through computer coding.

The graphic field in which to represent relationships is no longer the sheet of paper but the computer screen, where images are expressed using digital representation methods (mesh and NURBS). Knowledge of the elements that constitute these methods of representation is necessary for the efficient coding of drawing and information enrichment algorithms.

The ability to explicit algorithmic thinking is the part that needs to be enhanced in the university. In doing so, we must start from the awareness that any image produced using canonical methods of representation is generated as a result of an accumulation of construction signs that define the final image of the model. The rules that define the construction process constitute the algorithmic thinking that the designer follows to obtain the final image. Algorithmic thinking is embedded in drawings that follow the rules of descriptive geometry; this subject is a training ground for learning the processes that can be encoded in VPL.

BIM processes, which today seem to be the most appropriate for the description of architecture, receive a great contribution from VPL, which integrates the limitations of BIM modellers in the restitution of architectural instances for the historic built environment and the information deficiency towards certain knowledge domains (Gigliarelli et al., 2016). The integration of the two processes enables diagnostic assessments for the knowledge, analysis and prevention of cultural heritage. Introducing the 3D Modeling & BIM 2022 workshop with an in-depth study on VPL allows professionals and researchers to understand its importance within the architectural landscape, a field that is increasingly multidisciplinary and in need of tools capable of connecting competencies.

Bibliografia

Caetano, I., Santos, L., & Leitão, A. (2020). Computational design in architecture: Defining parametric, generative, and algorithmic design. Frontiers of Architectural Research, 9(2), 287–300. https://doi.org/10.1016/j.foar.2019.12.008

Calvano, M. (2019). Disegno digitale esplicito. Rappresentazioni responsive dell'architettura e della città. Aracne Editrice.

Calvano, M., & Casale, A. (2018). Represented Models and Typological Algorithms: The Role of Parametric Models for the Design of Product. In Computational Morphologies (pagg. 63–70). Springer. https://doi.org/10.1007/978-3-319-60919-5

Calvano, M., Casale, A., Valenti, G. M., Oliva, A., & Tsiamis, M. (2019). The shape of the folded surfaces. Drawing control and analysis. Francoangeli.

Calvano, M., Lo Turco, M., Giovannini, C. E., & Tomalini, A. (2020). The Narrated Drawing. Explicating algorithms for teaching digital modelling. Connettere | Connecting | Un disegno per annodare e tessere | Drawing for weaving relationships. Reggio Calabria e Messina. 17-18-19 settembre 2020, 196–215. http://ojs.francoangeli.it/_omp/index.php/oa/catalog/download/548/373/3054-1

Calvano, M., Martinelli, L., Calcerano, F., & Gigliarelli, E. (2022). Parametric Processes for the Implementation of HBIM—Visual Programming Language for the Digitisation of the Index of Masonry Quality. ISPRS International Journal of Geo-Information, 11(2), 93. https://doi.org/10.3390/ijgi11020093

Carlevaris, L., De Carlo, L., & Migliari, R. (A c. Di). (2012). Attualità della geometria descrittiva. Gangemi Editore.

Davis, D. (2010, luglio 31). Untangling Grasshopper – Part 1: Design patterns. Daniel Davis. https://www.danieldavis.com/untangling-grasshopper-part-1-design-patterns/

Diara, F., & Rinaudo, F. (2019). From reality to parametric models of cultural heritage assets for HBIM. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W15, 413–419. https://doi.org/10.5194/isprs-archives-XLII-2-W15-413-2019

Garagnani, S. (2013). Building Information Modeling and real world knowledge: A methodological approach to accurate semantic documentation for the built environment. 2013 Digital Heritage International Congress (DigitalHeritage), 489–496. https://doi.org/10.1109/DigitalHeritage.2013.6743788

Gigliarelli, E., Calcerano, F., & Cessari, L. (2016). Implementation Analysis and Design for Energy Efficient Intervention on Heritage Buildings. Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection, 91–103. https://doi.org/10.1007/978-3-319-48496-9

Kuhail, M. A., Farooq, S., Hammad, R., & Bahja, M. (2021). Characterizing Visual Programming Approaches for End-User Developers: A Systematic Review. IEEE Access, 9, 14181–14202. https://doi.org/10.1109/ACCESS.2021.3051043

Migliari, R., Casale, A., & Calvano, M. (2014). Sperimentazioni di architettura parametrica sulla Galleria Spada. In Prospettive architettoniche. Conservazione digitale, divulgazione e studio. Vol. 1 (pagg. 393–397). Sapienza Università Editrice.

Morris, C. W. (1938). Foundations of the Theory of Signs. University of Chicago Press.

Nerdinger, W., & others. (2005). Frei Otto: Complete Works: Lightweight Construction, Natural Design. Birkhäuser Switzerland.

Osello, A., Lucibello, G., & Morgagni, F. (2018). HBIM and Virtual Tools: A New Chance to Preserve Architectural Heritage. Buildings, 8(1), 12. https://doi.org/10.3390/buildings8010012

Spallone, R., & Calvano, M. (2022). Parametric Experiments on Palladio's 5 by 3 Villas. Nexus Network Journal. https://doi.org/10.1007/s00004-022-00592-1

Tsai, C.-Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. Computers in Human Behavior, 95, 224–232. https://doi.org/10.1016/j.chb.2018.11.038

Visual Programming Language. (2021, novembre 15). Wikipedia. https://en.wikipedia.org/wiki/Visual_programming_language

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2011, marzo 6). Research Notebook: Computational Thinking--What and Why? Carnegie Mellon School of Computer Science. https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Zabramski, S., Ivanova, V., Gadima, N., Yang, G., & Leepraphantkul, R. (2013). The effects of GUI on users' creative performance in computerized drawing. Proceedings of the International Conference on Multimedia, Interaction, Design and Innovation - MIDI '13, 1. https://doi.org/10.1145/2500342.2500360