www.cadanda.com
Computer-Aided Design and Applications

Developing Narrative Diagrams for Algorithmic Modeling of Architectural Parametric Design

Chieh-Jen Lin

Tainan University of Technology, t60011@mail.tut.edu.tw

Abstract. From diagrams of buildings to diagrams of algorithms, architects rely on diagrams to bridge abstract intentions to generating of building forms. This paper proposes an approach for generating narrative diagrams which can visually describe what design intentions are reached, and how the diagrams are generated. In order to communicate with a broader public than just AEC professionals, this paper proposes a visual strategy for manipulating and generating narrative diagrams that tell the design stories in parametric architectural design. By providing editable clusters of topological algorithms for recognizing and reasoning spatial relationships among geometric entities, this paper aims to help architects to represent design intentions within the algorithmic process of parametric design.

Keywords: Narrative Diagram, Spatial Language, Design Intention, Algorithmic Modeling, Algorithmic Framework.
DOI: https://doi.org/10.14733/cadaps.2020.447-457

1 INTRODUCTION

Drawings are the traditional media used by architects to predict and communicate their design outcomes, which account for Robin Evans's statement: "*Architects do not make buildings, they make drawings of buildings* [4]." When the audiences are not AEC professionals, however, drawings usually are not the best means of communicating abstract concepts with stakeholders [13]. Video, images, comics, infographics, diagrams, and other visual narratives are often more convincing than drawings even the buildings themselves [13]. With the maturation of 3D technology, 3D visualization has become a major means of remedying the lack of narratives of 2D drawings. But while BIM applications proclaim their progress in project optimization and cost efficiency by deploying 3D visualization, educators have found that 3D visualization is not always a useful communication strategy, and claimed that BIM applications should integrate analog media in order to enhance users' access to the abstract narrative via symbolic language [5].

Architects rely on the graphic language of diagrams to bridge abstract intentions with physical design. Veloso stated there are two kinds of formal system approaches, namely cybernetic and semiotic diagrams, can be used to represent architectural knowledge in a process based on explicit rules [17]. Semiotic diagrams can be used to represent architectural design in the form of a linguistic system. But semiotic diagrams, such as the diagrams of House IV made by Peter Eisenman can usually only represent visual processes of generating forms, rather than the solving processes of design problems. Cybernetic diagrams adopt computational processes and decompose design processes into the computational flows of design information. The parametric diagrams of such algorithmic modeling tools as Grasshopper, Dynamo, and Generative Components all apply the directed graphs of cybernetic diagrams.

New design thinking and strategies have emerged with the popularity of algorithmic modeling tools in recent years. Parametric diagrams of generative algorithms can be easily associated with the rules and dependencies of generating geometries, which led Tedeschi to declare that "*Architects do not make buildings, they make diagrams of buildings* [16]." Parametric diagrams are the critical features of parametric design. According to Oxman, the knowledge of how to manipulate and explore the associative relationships and dependencies of topological geometries is the critical key to parametric design thinking [12]. Unfortunately, in practice, this kind of manipulation and exploration must rely more on algorithmic thinking and scripting skills, than on architectural design knowledge. This is because algorithmic modeling tools are developed to accelerate 3D modeling tasks through the application of algorithms, thus the rules of parametric diagrams and visual algorithms do not necessarily have spatial language, which can be used to directly describe or be associated with geometric and spatial relationships of architectural design [9].

The associative relationships between the architectural design knowledge and the algorithmic processes of generating forms are critical for architectural design at early and conceptual stages. Cognitive research on parametric design suggests that the higher frequency of spatial language used by a designer, the more productive results in the designer's cognitive processes [9], and also indicates that algorithmic modeling tools may not be suitable for tasks involving the use of spatial concepts such as relative positions and spatial relationships. Spatial language is therefore the key to associate architectural design knowledge with algorithmic processes. If the recognition and reasoning mechanisms of spatial language can be introduced into algorithmic modeling tools as algorithmic components or procedures, this should be able to improve the narrative abilities of algorithmic modeling in describing what design intentions have been reached and how design problems are solved.

In addition to the inability to associate algorithms with spatial language, another major obstacle to the application algorithmic modeling in conceptual design is that stakeholders cannot understand those used algorithms. Unlike textual programming languages, the visual language of algorithmic modeling which cannot be easily interpreted themselves by naming parameters, functions, and classes. How to create a visual narrative of design intentions, and how to validate whether design intentions are achieved, remains a technical challenge. This paper consequently proposes an approach for generating narrative diagrams [17], which are similar to semiotic diagrams that can visually describe what design intentions are reached, and how geometric models are generated. Previous studies have proposed an algorithmic framework entitled STG*f*, which implements an algorithmic framework by applying Grasshopper and GhPython plugin as algorithm-aided design tools [10]. By providing editable clusters of topological algorithms for recognizing and reasoning spatial relationships among geometric entities, this paper aims to help architects to represent, develop, and reuse design intentions based on architectural knowledge in the algorithmic process of parametric design.

2 THE APPROACH OF DEVELOPING NARRATIVE DIAGRAMS

Diagrams are a popular means used by famous architects as visual narratives of design stories that bridge their intentions with the design outcomes. Some of the most persuasive examples of diagram use have been produced by the Bjarke Ingels Group (BIG), who has applied serial diagrams, such as those for the VM Houses (Fig. 1a) and the Mountain Dwellings (Fig. 1.b), to tell impressive architecture stories [6]. Different with the diagrams of Peter Eisenman focused on representing his design theories and the logic of formal transformation [3], the narratives of BIG's diagrams are basically like the Japanese comics known as 'manga,' which constitute a widely known medium for narrating stories [15], and BIG therefore terms those diagrams "Archicomic [7]." Apart from their descriptive texts, the narrative capabilities of BIG's diagrams are also based on the gradual transformation of the geometries and introductive symbols, such as the associative colors, lines, and arrows in serial diagrams.

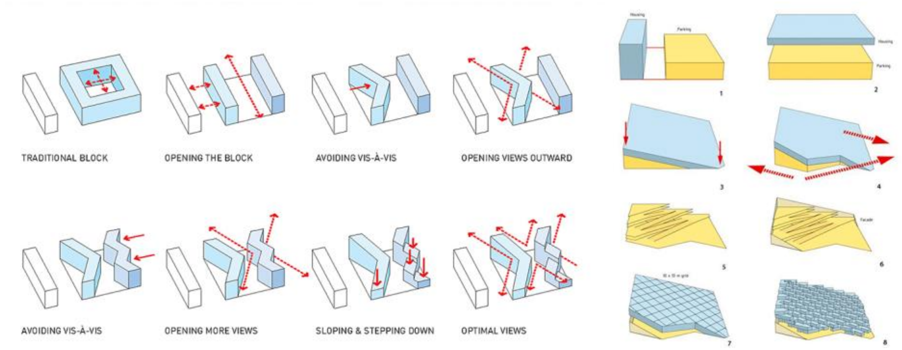

Figure 1: Two narrative diagrams of housing design by BIG: (a) The VM House in Copenhagen (left) [8], and (b) the Mountain Dwelling (right) [1].

In the case of architectural design, there may be completely different stories behind similar building forms, such as the Mobius ring applied by BIG and other architects in different projects. In contrast, architectural design competitions often seem to ask architects to tell the same story through different building forms. When narrating the stories behind the generative forms, it is necessary to visualize not only the generated geometries, but also the input parameters, and generative steps. And even though it may not need to visualize every result of all generative steps, however, designers must at least be able to visualize the critical steps they have selected to narrate their concepts.

While parametric diagrams of algorithmic modeling graphically narrate the generative processes of geometries on the canvas of visual scripts, the resulting narratives cannot always be recognized in the generated 3D visualization of Rhino. Grasshopper's preview function either displays only the results of a single step or overlap all steps of algorithms in the same positions. Overlapping previews cannot distinguish input parameters and generative steps, and the "baked" geometries in Rhino inevitably lose algorithmic information. For generating visual narratives of algorithmic modeling like BIG's serial diagrams, this paper employs the algorithmic framework entitled STG*f* developed in previous studies [10], then proposes an application approach for visualizing and manipulating generative algorithms through manipulating geometries in Rhino as semiotic diagrams.

2.1 Semantic Narratives of Algorithmic Intentions

The STGf framework applies a semantic ontology technique to store and represent chunks of design intentions. Adopting the "Subject-Predicate-Object" triple of semantic ontologies, an algorithmic component in Grasshopper presents a generative or computing process as "Predicate," which usually needs at least one parameter as "Subject," and the generated results constitute an "Object." For example, the "ExtrCrv" component with a 'Base' and a 'Curve' parameter is used to extrude "Base" along "Curve." Therefore, a simple "ExtrCrv" script (Fig.2a) can be represented as the directed graph of a "(B, C)→ExtrCrv" semantic triple (Fig.2b). For attaching more semantic narratives on the directed graph, therefore, a "ExtrCrv" component in Grasshopper can be represented in a semantic narrative in SWRL format as follows (Equation 1):

$$\text{ExtrCrv}(?\text{Base}, ?\text{Curve}) \rightarrow \text{Extrude}(\text{along}(?\text{Base}, ?\text{Curve})) \quad (1)$$

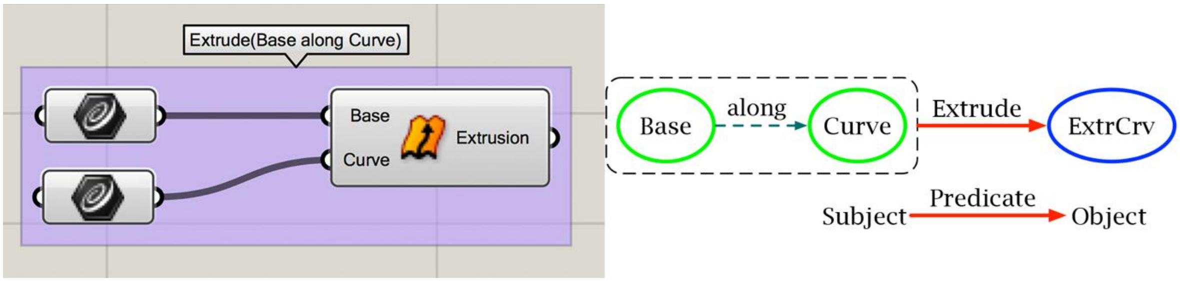

Figure 2: Semantic narratives: (a) The graphic script of an "ExtrCrv" component in Grasshopper (left), and (b) the semantic triple of the "ExtrCrv" ontology (right).

Although this conversion is not necessary in the case of algorithmic scripts, however, the directed graphs of semantic triples can easily be recognized by users and associated the algorithmic scripts with abstract design intentions, and thereby help to visualize semantic narratives. For example, the simple "ExtrCrv" script above may be used to represent the intention of a high-rise building mass, where the "Base" presents the shape of the standard floor, and the "Curve" presents the geometric intention concerning the building forms. To semantically narrate this intention, this script must not only indicate the semantics of input parameters and generated variables, such as "Floor" for the "Base" and "Mass" for the "ExtrCtv," but must also insert more semantic narratives into the triple, such as an "along" conjunction between two parameters, and the "Extrude" predicate between parameters and variables.

2.2 Visual Narratives of Algorithmic Intentions

The comic-like narratives of BIG's diagrams are based on the gradual transformation between illustrations and the introductive symbols connecting those illustrations. To visualize the semantic narratives of algorithmic intentions, the generative processes of algorithms should be displayed as serial diagrams. For example, the previews of the sample "ExtrCrv" script (Fig.2a) are usually displayed as overlapping geometries in Rhino (Fig. 3a). Based on the semantic narratives of the "ExtrCrv" ontology (Equation 1), this sample script should serially display two parameters and the generated variables in a separate manner (Fig. 3b). Consequently, the semantic predicates of "ExtrCrv" ontology will be automatically attached between the associative subjects and objects, such as "along" between the "Floor" and "Raising," and "Extrude" between the "Raising" and the final generated "Mass." This diagram is more able to narrate the semantic relationship between the parameters, the algorithms, and the generated geometries than the preview of Grasshopper.

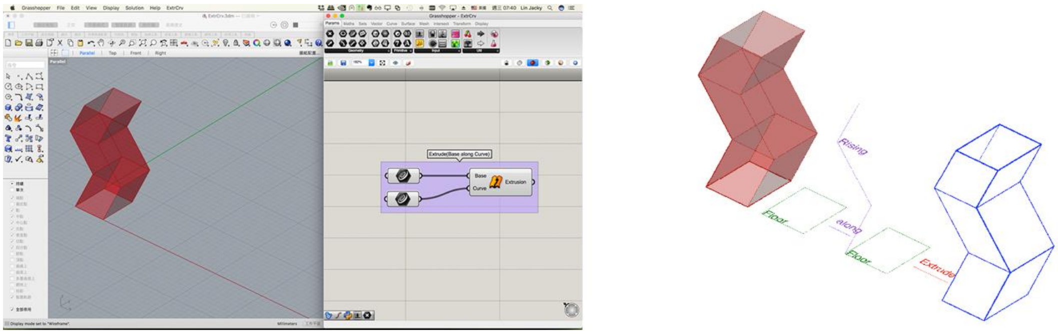

Figure 3: Visualization of semantic narratives: (a) Previews of an "ExtrCrv" graphic script in Rhino (left), and (b) the serial diagrams of the "ExtrCrv" semantic narratives (right) based on the Equation 1.

Clearly, the task of serially visualizing algorithmic steps, like drawing the frames of a manga, may be rather tedious, time and labor intensive. This paper therefore proposes two approaches for semi-automatically assisting the serial visualization process: (1) sample Python scripts and (2) editable clusters of algorithmic components. Python scripts are more powerful, and can more easily generate serial visualizations of this kind, but are more difficult for users to learn and to modify the scripts. Editable clusters of algorithmic components in STGf can provide more textural and graphic introductions for how to generate serial visualizations, and can explain how to modify the visualizations in order to obtain better narratives of design concepts. Due to lack of the version control functions in Grasshopper, however, rewriting a script or modifying algorithmic clusters is often a difficult task for experienced scripters, let alone designers.

2.3 Multiple Narratives of Algorithmic Intentions

Since design competitions often ask architects to tell the same story using different narratives, designers usually would try different ideas for telling a new narrative of a known story. For example, the concepts for CCTV building by OMA attempted to break the idea of building height as a hierarchy symbol, and change the Z-axis extrusion of the building form into a circulating circle [14]. As the concept of "form follows fiction" proposed by Scheeren [14], the different geometric intentions concerning how to change the extruding directions of the building tell the same story in new fictional narratives.

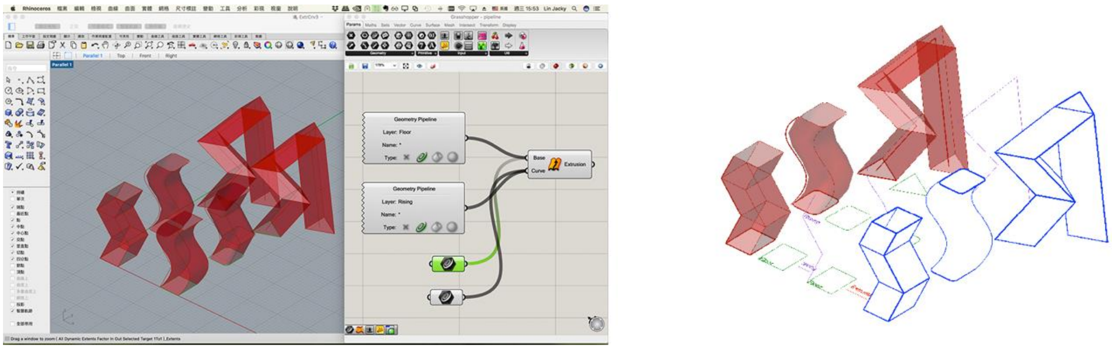

Figure 4: Multiple narratives of the same algorithmic intention: (a) two different approaches for collecting multiple parameters in Grasshopper: automatic or manual (left), and (b) the serial diagrams of multiple narratives of the same “ExtrCrv” script (right).

To test different ideas of new narratives, multiple parameters are useful for the same algorithmic intention. Although most of Grasshopper’s components allow input multiple parameters, however, modifying the collection of parameters sometimes cause unpredictable when a new intention is proposed. To simplify the manipulation multiple parameters of the same algorithmic intention, sample Python scripts and clusters of “Pipeline” components can be provided for designers to enable designers to automatically collecting geometries by layers, names, or geometric features (Fig. 4a), which is easier for designers to manipulate algorithmic intentions than to modify parameter collections directly. Through the visualization of algorithmic steps, designers can simultaneously visually narrate multiple design intentions of the same design story (Fig. 4b).

2.4 Summary of Narrative Techniques

In the previous studies, the generative algorithms developed in the STG*f* framework can visualize whether an input design concept/criterion was satisfied or not. However, it is usually not enough to narrative the whole story of involved design concepts for developing a building project. For example, architects sometimes may choose to violate certain criteria in order to achieve better results, such as to open large windows in the west in order to get a better view regardless of the western exposure problem. As the BIG's diagrams narrating their design stories by a series of gradually changing diagrams to illustrate the design process, a better narrative technique should start with a proposal that satisfies the most basic criteria, then step by step introduce new proposals which are modified by new involved criteria. A serial of narrative diagrams, like a comic of transformed proposals, therefore should be a better narrative technique than a single generated model.

3 INITIAL TESTING AND EVALUATION OF NARRATIVE DIAGRAMS

The basic idea of parametric architecture design is to take design contexts, include the functional requirements, building codes, site's contexts, known design criteria of a building project, as the parameters of generative algorithms. In previous studies, four exams of Taiwan's architect qualification form 2014 to 2017, therefore, had been studied for testing the STG*f* framework in order to understand how to apply an algorithmic framework for helping architects at the conceptual design stages [10, 11]. All of those exams focused on "community-friendly" issues and had strong inner or outer contexts of the given sites. This paper further tests the exam title in 2018, which is "a community sports center," and also provides some contexts around the given site (Fig. 5). Since this exam did not provide the inner contexts within the site, the functional requirements of the project and the features of existing buildings, therefore, became the major parameters for developing generative algorithms.

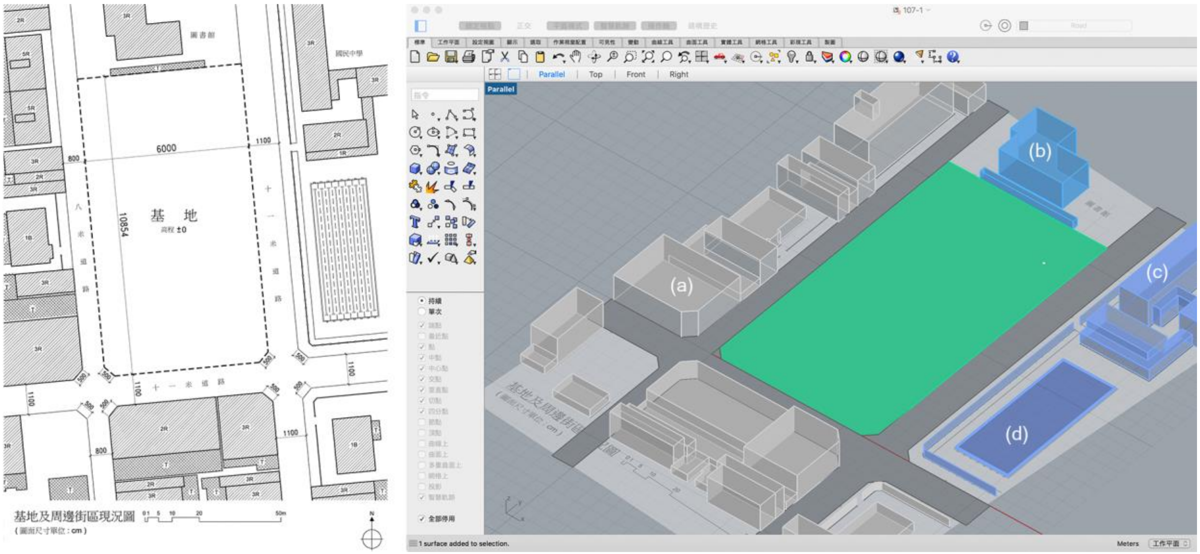

Figure 5: The site's contexts of the architect qualification exam in Taiwan 2018: (a) two or three stories houses with storefronts locate the west and south streets, (b) a library is adjacent to the north of the site, (c) a junior high school locates at the east street, and (d) the school has an 8-ways swimming pool.

3.1 Initial Algorithm of the Building Mass

The initial proposal of schematic design usually stars form the regulations of building codes. For example, the site's area of the exam in 2018 is about 6500 m^2, and the development intensity limit is 40% for the building coverage ratio, and 80% for the floor area ratio. Therefore, the first generative algorithm is to take the shape of the site, those ratio criteria, and an input initial point as parameters (Fig. 6a). By changing the building coverage ratio, such as reducing it from 40% to

20% (Fig. 6b), and moving the position of the initial point, architects can recognize the regulations of the building codes, and possible results of the building mass.

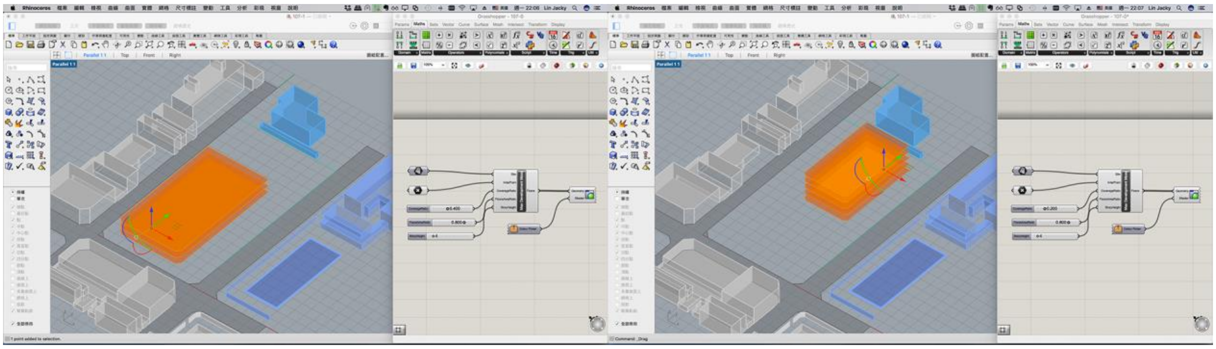

Figure 6: The first algorithm and generated models: (a) the max building coverage ratio and an initial point (left), and (b) the half building coverage ratio and the different position of the initial point (right).

Since there is no specific geometric intention yet, the generative algorithms of Fig 6 are based on three intentions: (1) to scale the shape of the site by the building coverage ratio to generate the preliminary floor area (Equation 2.1), (2) to divide the max floor area ratio by the building coverage in order to get the max number of the building's stories (Equation 2.1), (3) applying "LineArray" component to duplicate the floor along Z-axis according to the specified height and the number of stories to generate the building mass (Equation 2.3). Those semantic narratives can be represented in SWRL format as follows (Equation 2.1~2.3):

$$\text{Site}(?s) \cap \text{BuildingCoverage}(?c) \rightarrow \text{Floor}(\text{Scale}(?s, ?c)) \tag{2.1}$$

$$\text{BuildingCoverage}(?c) \cap \text{MaxFloorAreaRate}(?r) \rightarrow \text{MaxBuildingStories}(?r\ /\ ?c) \tag{2.2}$$

$$\text{LineArray}(\text{Floor}(\text{Scale}(?s, ?c)), ?Z, (?r\ /\ ?c)) \rightarrow \text{MaxBuildingMass}(?s, ?c, ?r) \tag{2.3}$$

3.2 Evolutionary Narratives of the Project's Requirements

The building program of the exam 2018 requires: (a) an indoor standard basketball court, (b) a 400 m^2 weight-training room, (c) a 400 m^2 multi-functional meeting room, (d) four 75m^2 multi-functional aerobics rooms, and (e) a 24-cars and 50-motorcycles parking lot as a statutory community which can became a holiday market for the community. Therefore the next episode of the design story is to arrange those indoor and outdoor spaces for responding the properties of surrounding buildings and the local climate with less rain and long hours of sunshine. A 6-meters modulus is applied to simplify the decision-making of all spatial geometries, and a one-story building mass is generated based on the basic requirements of the building project, and the building coverage ratio of this model is only about 28.1% (Fig.7a). The preliminary intentions of this model are: (1) to place the parking lot connecting the narrower road (Equation 3.1), (2) to place all indoor spaces together as a building mass (Equation 3.2), and (3) to place the building mass connecting the parking lot (Equation 3.3).

$$\text{Parking}(?p) \cap \text{Narrower}(\text{Road}(?x), \text{Road}(?y)) \rightarrow \text{Connecting}(?p, ?x) \tag{3.1}$$

$$\text{Indoor}(?a) \cap \text{Indoor}(?b) \rightarrow \text{Mass}(\text{Connecting}(?a, ?b)) \tag{3.2}$$

Parking(?p) ∩ Mass(Connecting(?a, ?b)) → Connecting(?p, Mass(Connecting(?a, ?b))) (3.3)

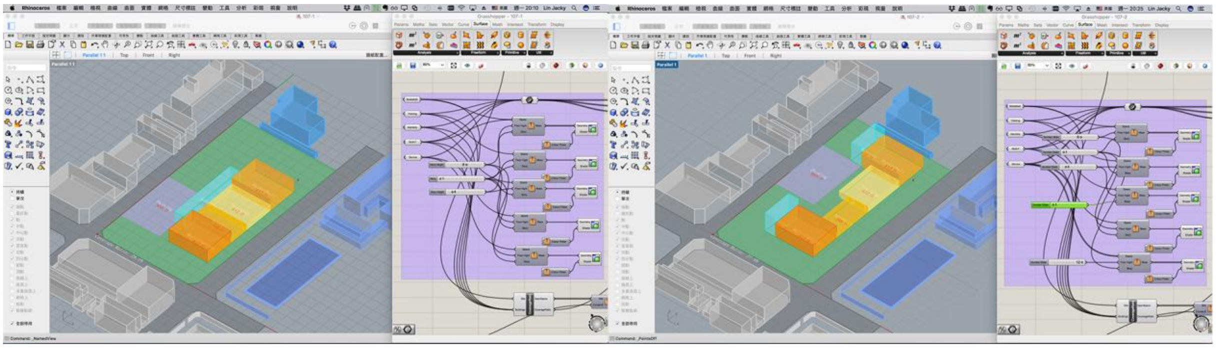

Figure 7: The second and third generated models: (a) a one-story building mass based on the project requirements (left), and (b) the new open space is surrounded by the building mass and close the parking lot for facilitating different usages (right).

It is very important for architects to arrange the building mass in order to shape open space for different activities. Obviously, the open spaces of the model in Figure 7a are only the remaining space after the buildings occupying the site. The indoor spaces lack the connection with the outdoors, and their ventilation and lighting performance is poor. The third model therefore generated by changing the directions of weight-training and aerobics rooms in order to reduce the length of common walls for getting better ventilation and lighting performance (Equation 4.1). Thus, the basketball court is moved to the north and the build mass will disconnect with the parking lot in order to shape one more outdoor space for a holiday market (Equation 4.2). The new outdoor space can provide more possible activities between the community and the sports center.

LengthOfCommonWalls(IndoorSpace(?x)) < LengthOfCommonWalls(IndoorSpace(?y)) → BetterVentilation&Lighting(?x) (4.1)

Disconnect(OutdoorSpace(?p), Mass(?m)) → OutdoorSpaceBetween(?p, ?m) (4.2)

However, since the different shapes and sizes of the weight-training and aerobics rooms, the shape of the new square is not complete. Then the next episode of the design story is: (1) to modify the shape of the weight-training room in order to enlarging the square for better activity (Equation 5.1), and (2) to move the aerobics rooms to the second story for disconnecting the square and getting more privacy (Equation 5.2) (Fig. 8a). However, the directions of weight-training and aerobics rooms will face the west where will occurs serious sun exposure problems. Therefore, the final episode is to attach a corridor which can block the sun shining (Equation 5.3) and connect the basketball court and the multi-functional room (Fig. 8b).

Area(OutdoorSpace(?x)) > Area(OutdoorSpace(?y) → BetterActivity(?x) (5.1)

NotGroundFloor(IndoorSpace(?x)) ∪ Disconnecting(IndoorSpace(?x), AnyOutdoorSpace(?y)) → BetterPrivacy(?x) (5.2)

Connecting(IndoorSpace (?x), IndoorSpace (?y)) ∩ OnTheWest(?y, ?x) → AvoidWestSunshing(?x) (5.3)

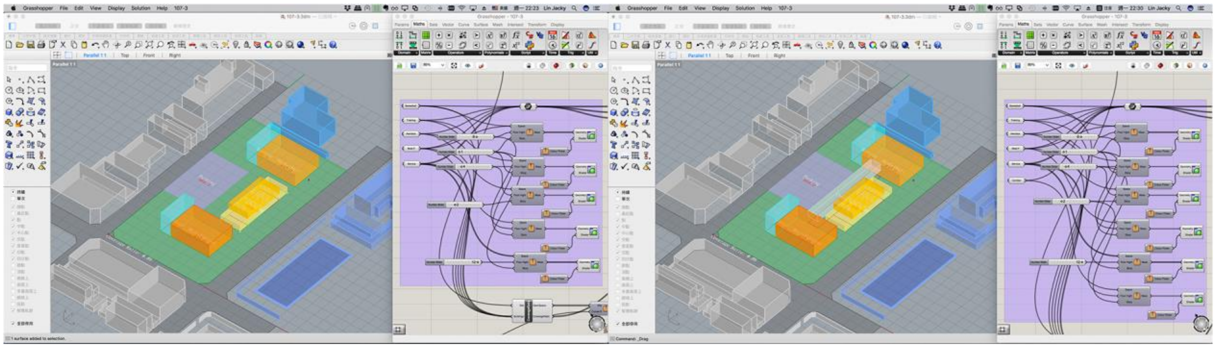

Figure 8: The fourth and final generated models: (a) the aerobics rooms are moved on the weight-training room (left), and (b) a new corridor is attached to block the sun shining and to connect the basketball court and the multi-functional room (right).

3.3 Primary Evaluation of Narrative Diagrams

The above proposes a serial of generated models which are generated by similar algorithms. Each generated model responds to some specific design contexts, such as the project's basic requirements, shaping the outdoor spaces for facilitating activities, vacating passages and connecting spaces, blocking west sun-shining. By attached illustrative texts and symbols, those models can easily to narrate the design story behind the generated proposal. As designers must manually attach textual annotations in the graphic scripts of Grasshopper for explaining the intention and purpose of the specific algorithm, most of the illustrative texts and symbols may need to be inputted manually after the models were generated. By applying the STG*f* framework, it is easy to assign semantic criteria among design objects and visually validate criteria by topological relations. By textual and visual annotations generated by the STG*f* framework, it can help designers to make textual annotations and introductive symbols for narrating their own design stories.

4 DISCUSSION

As a kind of cybernetic diagrams, the parametric diagrams of algorithmic modeling illustrate how the model is generated but usually cannot explain why they were derived. On another hand, the generated model of algorithmic modeling can present the result of designers' concepts, but sometimes may not help designers to communicate their ideas with the audiences. The traditional semiotic diagrams used by architects describe the intention and purpose of a proposal but sometimes cannot help to validate whether the proposal archives the intention or purpose. And the criteria applied by architects cannot always to generative candidate results by algorithms. For example, the limit of the building coverage and the floor area ratio cannot help architects to generate models by the criteria. One of the feasible approaches is to apply the genetic algorithm for optimizing specific multi-criteria [2]. However, the optimization approaches imply the single best solution for the same criteria. As mentioned above, design competitions always ask architects to tell the same stories by using different narratives. Selecting different criteria than only satisfying basic requirements and performances, therefore, become a necessary design strategy for winning a competition. In this situation, the narrative diagrams of design stories rather than performance optimization should have more explanatory power to make a proposal more acceptable for a broader public than just AEC professionals.

5 CONCLUSIONS

Since the parametric diagrams of algorithmic modeling constitute a type of cybernetic diagrams, their narrative ability is naturally inferior to the semiotic diagrams. This paper proposes the approach of applying an algorithmic framework to generate visual narratives of algorithmic intentions, and the framework narrates what design intentions are reached, and how the diagrams are generated. By integrating semantic ontology and applying visualizing algorithms, this paper aims to help architects to associate design intentions with the algorithmic process of parametric design. While educators sometimes claim that "3D visualization is not a design strategy [5]," parametric design and algorithmic modeling is not always good communication strategies. In order to communicate with and convince a broader public than just AEC professionals, this paper proposes a visual strategy for manipulating and generating narrative diagrams that can reveal and tell the design stories embedded in the generative algorithms of parametric architectural design.

ACKNOWLEDGEMENTS

The Ministry of Science and Technology of Taiwan supported this paper under grant number MOST 106-2221-E-165-001-MY2.

ORCID

Chieh-Jen Lin, https://orcid.org/0000-0001-9981-5864

REFERENCES

[1] Basulto, D.: In Progress: Mountain Dwellings / BIG, https://www.archdaily.com/1581/in-progress-mountain-dwellings-big/, Arch Daily.

[2] Chang, M.-C.; Shih, S.-G.: A Hybrid approach of dynamic programming and genetic algorithm for multi-criteria optimization on sustainable architecture design, Computer-Aided Design and Applications, 12(3), 2014, 310-319. http://doi.org/10.1080/16864360.2014.981460

[3] Eisenman, P.; Cole, R.: House IV Project, Falls Village, Connecticut, https://www.moma.org/collection/works/81, The Museum of Modern Art.

[4] Evans, R.: Translations from Drawing to Building and Other Essays, Architectural Association, London, 1997.

[5] Flanagan, R.: BIM's complexity and ambiguity - BIM v. paper architecture, in: Proceedings of the 36th eCAADe Conference - Volume 1, Lodz, Poland, 2018, 265-270

[6] Ingels, B.: Three Warp Speed Architecture Tales, https://www.ted.com/talks/bjarke_ingels_3_warp_speed_architecture_tales, TED Talks.

[7] Ingels, B.: Yes Is More: An Archicomic on Architectural Evolution, Taschen, Köln, 2009.

[8] Iremozn: VM House in Copenhagen, https://iremlandscape.wordpress.com/2012/03/04/vm-house-in-copenhagen/, Iremozn.

[9] Lee, J.H.; Ostwald, M.J.; Gu, N.: The language of design: spatial cognition and spatial language in parametric design, International Journal of Architectural Computing, 14(3), 2016, 277-288. https://doi.org/10.1177/1478077116663350

[10] Lin, C.-J.: The STG-framework: a pattern-based algorithmic framework for developing generative models of parametric architectural design at the conceptual design stage, Computer-Aided Design and Applications, 15(5), 2018, 653-660. https://doi.org/10.1080/16864360.2018.1441231

[11] Lin, C.-J.: Topological vision: applying an algorithmic aramework for developing topological algorithm of architectural concept design, Computer-Aided Design and Applications, 16(3), 2019, 583-592. https://doi.org/10.14733/cadaps.2019.583-592

[12] Oxman, R.: Thinking difference: theories and models of parametric design thinking, Design Studies, 52, 2017, 4-39. https://doi.org/10.1016/j.destud.2017.06.001
[13] Quici, F.: Architecture and visual narrative, Multidisciplinary Digital Publishing Institute Proceedings, 1(9), 2017, 1082. https://doi.org/10.3390/proceedings1091082
[14] Scheeren, O.: Why Great Architecture Should Tell a Story?, https://www.ted.com/talks/ole_scheeren_why_great_architecture_should_tell_a_story, TED Talks.
[15] Schnabel, M.A.; Lo, T.T.; Qu, Y.: Manga architecture – developing architectural narratives, in: Proceedings of the 18th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2013), Singapore, 2013, 771-780.
[16] Tedeschi, A.: AAD Algorithms-Aided Design: Parametric Strategies Using Grasshopper, Le Penseur, Brienza, Italy, 2014.
[17] Veloso, P.L.A.: Cybernetic diagrams: design strategies for an open game, International Journal Of Architectural Computing, 12(4), 2014, 379-397. https://doi.org/10.1260/1478-0771.12.4.379