SEMI-AUTOMATIC WORKFLOW FOR AIR-CONDITIONING SYSTEM ZONING AND SIMULATION

Yikun Yang & Yiqun Pan
Tongji University, Shanghai, China

Georg Suter
Vienna University of Technology, Vienna, Austria

ABSTRACT: *Building information modeling shows its potential in the performance driven design, where multiple design solutions are generated and assessed against certain design goals. This paper proposes a workflow for the air-conditioning system design and simulation on thermal zoning level. Thermal zoning plays a pivotal role in the design thinking of engineers, synthesizing load calculation, equipment sizing, and pipe/duct layout. However, it is often done intuitively with its effectiveness and performance unclear at the outset. To make it quantitative, we decompose the zoning process into two levels (control/system) of space aggregation, joining both semantic and numeric characteristics. For the semantic part, space functions are considered through space labeling, accessibility, and adjacency. Regarding the numeric part, spaces are zoned based on their thermal response similarities, using dynamic mode decomposition of the simulated indoor temperature. A two-level hierarchy of duct/pipe network is generated. It connects spaces within each control zone at the second level, and terminal equipment of each zone at the first level, representing typical fan-coil or variable-air-volume systems. For each zoning scheme, the network and configurations are serialized as Modelica scripts for co-simulation with EnergyPlus. The designer can evaluate different zoning schemes in terms of initial cost, energy consumption, and comfort level, based on the simulation result. The entire workflow is implemented in Grasshopper with self-coded plugins.*

KEYWORDS: *BIM, Thermal zoning, Generative design, HVAC system, Performance simulation.*

1. INTRODUCTION

Building Information Modeling (BIM) provides solutions to the design and management problems in the realm of architecture, engineering and construction, as a platform for data and cooperation. The information can be delivered or retrieved by model view definitions (MVD), BIM query languages, or application programming interfaces (API). However, they barely offer model transformation, which involves the insertion, addition of level of detail, or aggregation of objects (Fischer, 1998). In the forward design process, the model transformation expands the room-based model into another space view for analysis (Suter, 2022), such as energy performance or system design.

Zoning is to transform the room-based model (architectural geometry) to a zone-based model, bridging BIM with building energy modeling (BEM), which is also a critical step in the design of an air-conditioning (AC) system. It is more like an "art" for so many factors to consider, such as space function restriction, space load profiles, convenience for ductwork, balance of performance and cost, and even user preference, both semantic and numeric and hard to quantify. The engineers usually solve the zoning problem intuitively by rule of thumb, leading to one solution. How will it affect the system performance or whether there exists an optimum zoning scheme remains unsolved. Back in 2001, Brahme et al. (2001) investigated the generative ducting based on a grid system at the initial stage. Berquist et al. (2017) continued the idea of generative design and experimentally piloted different zoning on several rooms. Bres et al. (2017) examined the zoning effect of the water heating system for residential buildings, with detailed simulation feedback from TRNSYS. With the lower cost of simulation, it is quite possible to automate the system design and simulation at the zoning phase.

In system design, a thermal zone represents the spaces (part or aggregation of rooms) with heating and cooling requirements that are sufficiently similar so that desired conditions can be maintained throughout using a single sensor (denoted as *control unit*). However, in simulation, the thermal zone stands for the spaces that can be lumped together as a single air node (rephrased as a *simulation unit*), where the parameters of air are uniform. The difference is that the simulation unit is scalable, depending on the modeling target (Fig. 1). For example, it should be in line with the control unit when modeling buildings in operation, as recommended by the ASHRAE Standard. When it comes to the building massing, a shoebox model is enough to study how geometric form affects energy performance. Even one thermal zone for a building is acceptable in city-scale simulation and

Referee List (DOI: 10.36253/fup_referee_list)
FUP Best Practice in Scholarly Publishing (DOI 10.36253/fup_best_practice)
Yikun Yang, Yiqun Pan, Georg Suter, *Semi-Automatic Workflow for Air-Conditioning System Zoning and Simulation,* pp. 1049-1060, DOI 10.36253/979-12-215-0289-3.105

planning. In the system design, the control unit is the smallest on the building scale, which considers the load similarity. When the building gets larger, more system units emerge with load diversity considered. For example, on a system zoning level, equipment can be downsized by staggering the load profiles of control units. A proper zoning and routing of distribution network can reduce the power of fan/pump. More energy distribution can be regulated on a larger scale, such as the exhibition center that has multiple plants or power stations.

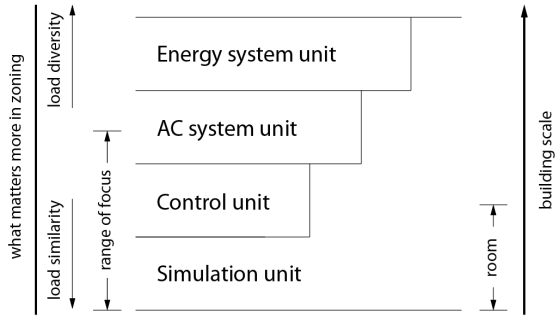

Fig. 1: Different levels of zoning in system design and simulation

Ideally, one should set the control unit exactly the same as the functional space. However, it may have more complex plumbing, ductwork and sensor/equipment installation, increasing the initial cost. There is a trade-off in controlling multiple spaces by a single thermostat since it does cut the cost while causing overheating/cooling. To manifest such an effect, the simulation unit must be smaller than the control unit, or the temperature difference will be evened out during the parameter lumping. Hence, this work takes the functional space as the atomic simulation unit and focuses on the zoning of control unit and system unit.

It is not quite straightforward to evaluate different zoning schemes since it is concerned with the actual system performance and the cost of related equipment configurations. A view model of the distribution network would help with the cost estimation and even the system energy modeling. Bres et al. (2017) pipelined the distribution network generation of the water heating system for residential buildings, by finding the minimum spanning tree (MST) from the potential zone centroids and space boundary vertices. Medjdoub et al. (2018) developed a method for open space fan-coil system layout under specific restrictions. Chen et al. (2022) solved the layout of diffusers and ducts for open spaces with hydraulic balance considered. In related research, the distribution system above the thermal zone level has been rarely visited, especially for non-residential buildings. The problem of distribution network layouts is similar to the design of integrated circuits (Held, 2011) or indoor navigation networks (Fu, 2020).

Following the thread of Bres et al. (2017), this paper details the workflow of thermal zoning and the modeling of air-conditioning distribution systems for office buildings, which can offer multiple zoning schemes in primary design with simulation feedback. After an overview of the methodology, the rest of the paper will be organized into four parts: space view generation, thermal response analysis and zoning, pipe/duct network generation, and model scripting. Each part is exemplified by the same floorplan for reference.

2. METHODOLOGY

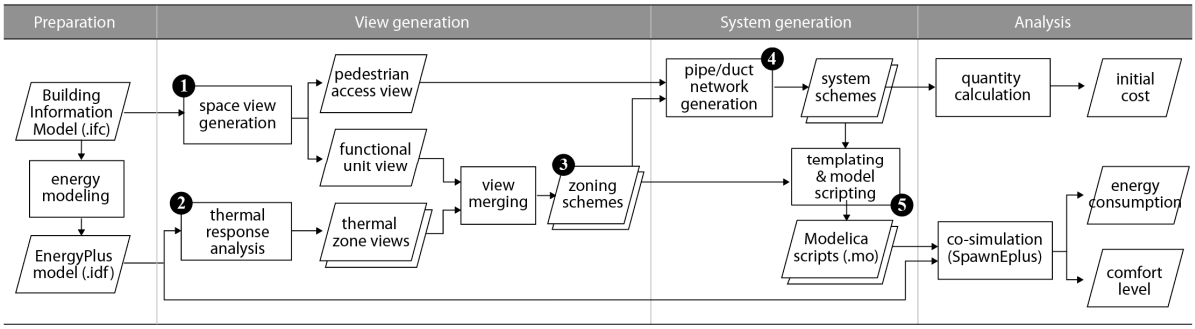

Fig. 2: The semi-automatic workflow of generative thermal zoning toward system simulation

Fig. 2 shows the overall workflow that takes the BIM model (IFC file) as input. In the first preparation stage, the user needs to manually check the integrity of the information model. All spaces must be defined and labeled with

their function correctly. All spaces must have boundary components and the 2nd-level boundary defined. To facilitate the testing, templates of construction type, space schedule and load settings are used. The model conversion from BIM to energy model implements IfcOpenShell (Visschers, 2016) and OpenStudio SDK.

The first step includes previous work by Suter et al. (2013), where a method is described to define space views and transform room-based source space data into corresponding space models. Following certain view definitions, the space layout can be selected, aggregated, or decomposed into functional views that are relevant to schematic design (Suter, 2015). The pedestrian space access network is used to identify circulation areas and shafts for plumbing, and the functional unit view for space groups with similarities.

The second step applies Dynamic Mode Decomposition (DMD) (Schmid, 2022) to group spaces with similar thermal responses, similar to the Koopman Analysis application by Georgescu (2015), only more sampler and cost-effective in computation. By tuning the clustering threshold, multiple thermal zone views are generated. The functional unit view and thermal zone views are overlaid as zoning schemes, satisfying both semantic and physical requirements.

The third step implements the potential network that conceptualizes pipe/duct layout in early design, guiding the system distribution network routing. It does not reflect the actual design so the pipe/duct and the equipment may overlay in the schematic diagram. The network first joins spaces of a thermal zone and then connects each zone to the designated shafts. A JSON file collects the geometry and topology of the generated system, with information of peak load, thermostat inherited from zoning schemes.

The fourth step parses the zoning and system into the simulation model. For system simulation, Modelica is used to reflect the control effect across spaces of a thermal zone, which is the temperature bias caused by sharing a thermostat, while EnergyPlus handles the building physics as co-simulation. The workflow is able to reflect the system performance by simulation outputs (energy consumption and comfort level) and quantity takeoffs (equipment, pipe/duct/junction).

Apart from the space view generation (space modeling system, Suter, 2022), the workflow is implemented on the Grasshopper platform with the help of LadybugTools (Roudsari, 2013) and self-coded components. Modularized components make it easier for testing and visualization. The toolkit[1] is written in C# with Rhino core algorithms and CGAL.

3. FUNCTIONAL UNIT VIEW

Although thermal zoning follows clear yet implicit physical requirements, there are more explicit semantic restrictions to it, such as the fire compartmentation forbidding cross ventilation, or the tenant zone for separate energy management. Additionally, designers must consider space functions to avoid serving bathrooms and offices with the same duct system. The functional unit view can depict such a function isolation.

In previous work (Suter, 2022), a data processing pipeline was proposed for defining space views using space ontologies and layout transformation operations. The generated space view model can help with the space layout analysis by offering insights int space function, accessibility, orientation, daylighting or ventilation. In the first step, room-based source data are extracted from BIM by IFC class filters, such as space geometry and related objects. The second step transforms the data to a source space layout, where spaces are labeled automatically or manually. Labels are assigned by default according to the IFC to space ontologies class mapping. Additional labels are inferred by semantic reasoning. In the third step, the source space layout is transformed into a certain space view, by its defined operation sequence (including filtering, selection, aggregation and update).

The functional unit view uses the space access network to identify the cluster in terms of adjacency and function. The space access network originates from a spatial relation network that builds upon centroid nodes of all layout elements (e.g., spaces and doors), with their spatial relations as edges. Such relations include containment, adjacency, proximity, and partial enclosure. A door adjacent to two spaces indicates an accessible path lies in between, while the isolated node with no accessibility proves to be a shaft. Different levels of depth in the space access network tell how important a space functions as a circulation area. Typically, the main circulation space contains the elements accessing each functional unit, which form a node cut-set that partitions a space access network into multiple components. Spaces are then merged as a functional unit with classification (function label) inferred from the spaces within.

[1] https://www.github.com/ian-quinn/tellinclam

Fig. 3 Sample floorplan [2]

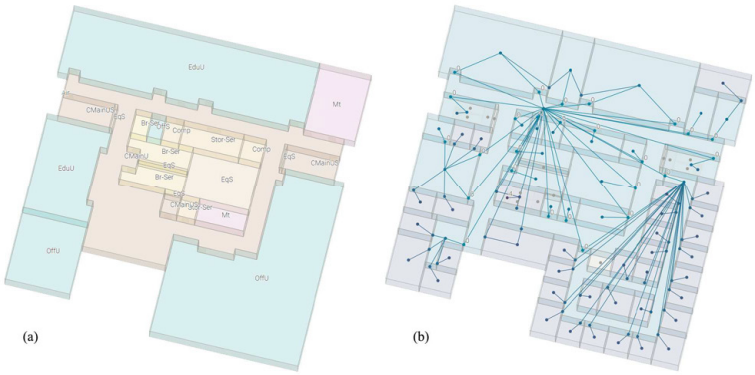

Fig. 4 Generated view of (a) functional unit (b) space access network

Fig. 3 displays the floorplan used for this work, labeled with detail functions. In the example (Fig. 4-a), there are seven major functional units: two marked as educational units (EduU) that include classrooms and affiliated spaces; two marked as office units (OffU) according to their major function; two meeting units (Mt); one circulation unit including the main corridor, stairs and atrium. This view will act as an overlay to the thermal zoning, reflecting function restrictions.

4. THERMAL ZONING VIEW

The control needs for air-conditioning are directly reflected by the thermal responses of all spaces. Taking the whole floorplan as one dynamic system, each space may have different characteristics. The essence of thermal zoning is to apply the same control logic to several spaces sharing similar dynamic characteristics, thus achieving an acceptable control effect with less controller and actuator.

Although many factors are intertwined in this complex dynamic system, such as operation schedule, internal loads, solar radiation, thermal resistance and thermal mass of structure components, the indoor float temperature is one direct externalization of such dynamics. By measuring the time series of temperature waves, we may identify the dynamic characteristics of each space, and aggregate them as thermal zones based on their similarity. Inspired by Georgescu (2015), Dynamic Mode Decomposition (DMD) is used to perform the Koopman Analysis, extracting the dominant modes and their properties based on the free-float temperature data.

The general idea of the Koopman Analysis is to study the time evolution of observables under iteration of a nonlinear system through the Koopman operator U, which is linear but infinite dimensional (Raak, 2016). Considering a dynamic system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k)$ evolving on a manifold M, $\boldsymbol{f}$ is a non-linear map describing how $\boldsymbol{x}$ evolves in discrete time. The operator U acts on the scalar function $g: M \to \mathbb{R}$ (1), which is the selected observable (or the system output), then describes its evolution in a linear, infinite space, along with the system $\boldsymbol{f}$:

$$Ug := g \circ \boldsymbol{f} \qquad \mathrm{Ug}(\boldsymbol{x}_k) = \mathrm{g}\big(\boldsymbol{f}(\boldsymbol{x}_k)\big) = \mathrm{g}(\boldsymbol{x}_{k+1}) \tag{1}$$

Given eigen-decomposition of U (2), we may generally express the vector function $\boldsymbol{g}: M \to \mathbb{R}^n$, in terms of Koopman eigenfunctions ϕ_j and eigenvalues λ_j by (3),

$$U\phi_j(\boldsymbol{x}) = \lambda_j \phi_j(\boldsymbol{x}), \quad j = 1, 2 \ldots \infty \tag{2}$$

$$\boldsymbol{g}(\boldsymbol{x}) = \sum_{j=1}^{\infty} \phi_j(\boldsymbol{x}) \boldsymbol{v}_j \qquad \boldsymbol{g}(\boldsymbol{x}_k) = \sum_{j=1}^{\infty} \lambda_j^k \phi_j(\boldsymbol{x}_0) \boldsymbol{v}_j \tag{3}$$

where $\{\boldsymbol{v}_j\}$ is a set of vector coefficients called Koopman modes of map $\boldsymbol{f}$. The set of eigenvalues $\{\lambda_j\}$ indicates the growth rate and frequency of each mode. The practical idea behind this is to collect a set of data, identify observable $\boldsymbol{g}$ of interest, and then express it in terms of Koopman modes and eigenvalues (Chen, 2012). For example, one may take the time series temperature of spaces as observables (data snapshots $\mathbf{X}(\mathrm{n} \times \mathrm{m})$), and analyze the thermal responses of the complex system by eigenvalues and modes.

[2] https://www.angelo.edu/live/news/12569-no-place-like-home

The DMD algorithm approximates the modes and eigenvalues by a finite data set, with a variant of the Arnoldi method described in Chen et al. (2012) and summarized as (Alg. 1). Assuming the dynamic $\boldsymbol{f}$ is linear with $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$, then eigenvalues of $\boldsymbol{A}$ are also eigenvalues of U. Furthermore, if $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$, then modes $\boldsymbol{v}_j$ are the corresponding eigenvectors of $\boldsymbol{A}$ (Rowley et al., 2009). However, eigen-decomposition of A is hard to solve directly due to its large dimensions. Alg. 1 further approximates A by projecting it onto one Krylov subspace $\mathcal{K}$ expanded by $\boldsymbol{K}$, then calculating the eigenvalues and eigenvectors with the low-rank operator. Note that in real-world non-linear problems, there must be a bias $\boldsymbol{r}$ representing $\boldsymbol{x}_m$ within $\mathcal{K}$, it is critical to find suitable constant vector(s) $\boldsymbol{c}$ to minimize $\boldsymbol{r}$ in least-square sense (Alg. 1 Line 2). Then, $\boldsymbol{C}$ is regarded as an approximation of the action of the Koopman operator on the associated finite dimensional space $\mathcal{K}$ (Raak, 2016). The resulting empirical Ritz values and vectors (Alg. 1 Lines 7, 8) behave in precisely the same manner as the eigenvalues and modes v of U (1). The theoretical deduction can be found in Rowley et al. (2009).

Algorithm 1 DMD (one variant of standard Arnoldi)

Input snapshots of observable $[\boldsymbol{x}_0\ \boldsymbol{x}_1 \cdots\ \boldsymbol{x}_m]$, $\boldsymbol{x}_k \in \mathbb{R}^n$

Output Ritz values $\{\lambda_i\}$, Ritz vectors $\{\boldsymbol{v}_i\}$

1: $\boldsymbol{K} := [\boldsymbol{x}_0\ \cdots\ \boldsymbol{x}_{m-1}]$, $\boldsymbol{c} := [c_0\ \ \ldots\ \ c_{m-1}]^T$

2: $\boldsymbol{x}_m = \boldsymbol{K}\boldsymbol{c} + \boldsymbol{r}, \quad \boldsymbol{r} \perp span(\boldsymbol{x}_0 \ldots \boldsymbol{x}_{m-1})$ ▷ find constant $\boldsymbol{c}$ to construct $\boldsymbol{x}_m$ with minimal residual in least-square sense

3: $\boldsymbol{c} = \boldsymbol{K}^+\boldsymbol{x}_m$ ▷ one solution by observation

4: $\boldsymbol{K}^+ = (\boldsymbol{K}^*\boldsymbol{K})^{-1}\boldsymbol{K}^*$ ▷ when $\boldsymbol{K}$ is not full rank

5: Construct the companion matrix

$$\boldsymbol{C} := \begin{bmatrix} 0 & 0 & \cdots & 0 & c_0 \\ 1 & 0 & & 0 & c_1 \\ 0 & 1 & & 0 & c_2 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & c_{m-1} \end{bmatrix}$$

6: $\boldsymbol{C} = \boldsymbol{T}^{-1}\boldsymbol{\Lambda}\boldsymbol{T}$ ▷ one possible decomposition

7: Output diagonal $\boldsymbol{\Lambda}$ as Ritz values $\{\lambda_1, \ldots, \lambda_m\}$

8: Output $\boldsymbol{V} := \boldsymbol{K}\boldsymbol{T}^{-1}$ as Ritz vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$

There are various approximation methods of $\boldsymbol{A}$ span over the algorithm spectrum, from Koopman analysis (accurate, complex) to DMD (coarse, simple). For example, one can implement a classical Arnoldi algorithm, or by other variants like QR decomposition, SVD decomposition (standard DMD), or Proney-type method (Hankel DMD) (Schmid, 2022). Alg. 1 applies a variant Arnoldi method that takes $\boldsymbol{c} = \boldsymbol{K}^+\boldsymbol{x}_m$ as one solution for Line 2 by observation. It is not unique and the result needs cross-verification.

By simulation, a typical school building may yield year-long (8760 hours) time-series data on the free-float temperature, of hundreds of spaces. The space dimension of each snapshot is far less than its time dimension (n $\ll$ m), where rank deficiency is inevitable. The standard DMD method performs poorly because its SVD process truncates the matrix to n$\times$n with lots of information loss on the time dimension. While, the Arnoldi method gives nice accuracy because it expands the matrix to m$\times$m without truncation. A similar method is applied to the air temperature analysis of a conditioned room (Boskic, 2020), with the data dimension 28$\times$241, and a power system whose data dimension is 7$\times$24~120 (Raak, 2016).

In order to get the free-float temperature and perform the analysis, the following workflow is implemented based on the Grasshopper platform. It includes three steps: 1) build up the energy model by retrieving IFC information; 2) implement the Arnoldi algorithm on the temperature series output by simulation; 3) perform hierarchical clustering and output multiple thermal zoning schemes.

The building floorplan can be manually drawn or transplanted from IFC by IfcOpenShell (Visschers, 2016). The LadybugTools (Roudsary, 2013) helps with the energy modeling based on the geometry and function labeling, by calling OpenStudio SDK. EnergyPlus 9.6.0 performs the year-long simulation without any system, using construction and space function templates from ASHRAE Standard 189.1-200. Because EnergyPlus 9.6.0 does not support thermal zones with multiply-connected region, some corridors are further divided in this case.

Fed with the temperature series, Alg. 1 yields the empirical Ritz vectors and values approximating the Koopman modes $\boldsymbol{v}$ and their eigenvalues λ. If the complex number λ falls near the unit circle, it represents a more steadily evolving mode. Here we name the Growth as $|\lambda|$, the Norm as $|\mathbf{v}|$, and the Frequency as $Im(log(\lambda))/2\pi\Delta t$. Δt stands for the sampling period which, in this case, is 1 hour.

The dominant modes must be selected and cross-validated because the Arnoldi method generates tons of modes (equal to the number of time dimensions). A simple way is to identify the energy-intense frequency bands, then

rank all modes by their growth value in descending order, and look up for those modes with the largest norm. In this case, by Discrete Fourier Analysis, the frequency bands of 1/8760, 1/24, and 1/168 take up over 95% energy of the entire system, which corresponds to the year, day and week period, in line with the actual operation schedule. In this way, the dominant modes are highlighted in red in Table 1.

Table 1: Modes ranked by growth

mode	growth	norm	1/frequency
8237	1.000038	9.185662	9656.458
8238	1.000038	13.93669	-9656.458
8418	1.000029	51.34232	168.4023
8419	1.000029	64.30705	-168.4023
5799	**1.000026**	**249.9451**	**23.92526**
5800	1.000026	267.1617	-23.92526
8239	0.999999	82.70703	∞
5845	0.999960	261.9814	24.0553
5846	0.999960	303.0435	-24.0553
5779	0.999950	150.2885	23.9970

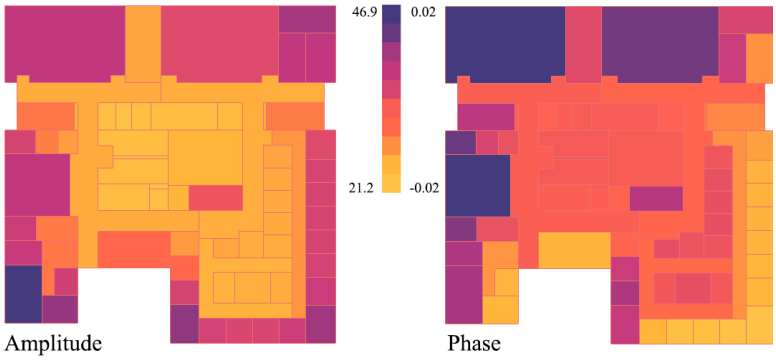

Fig. 6 The distribution of amplitude and phase of mode 5799.

Within the dominant mode, each complex number in $\boldsymbol{v}$ represents the dynamic characteristics of a space. In Fig. 6, each space is rendered by pseudo color based on the calculated amplitude (norm) and argument value as follows. Space aggregation can be identified visually.

$$Amp = \sqrt{Re(\boldsymbol{v}_i)^2 + Im(\boldsymbol{v}_i)^2}, \quad Arg = \arctan\left(\frac{Im(\boldsymbol{v}_i)}{Re(\boldsymbol{v}_i)}\right)$$

To make it quantified, the hierarchical clustering is implemented on the 2-dimensional data space by their Euclidian distance. The hierarchical clustering gradually increases the threshold of cluster distance (e.g., the similarity of dynamic response), and generates multiple layouts (Fig. 7) as the generative zoning process.

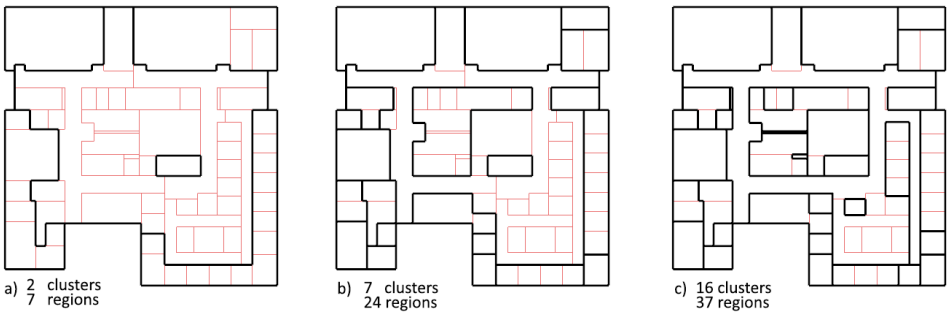

Fig. 7: Different thermal zoning schemes based on different cluster distance.

The final zoning scheme is the overlay of the thermal zoning view and the functional unit view, with all unconditioned spaces ignored. Spaces already grouped in thermal zoning may be partitioned again by the functional unit view.

5. PIPE/DUCT NETWORK VIEW

To better evaluate the effectiveness of zoning schemes, a detailed pipe/duct network view for the distribution system is needed. The designer may grasp a basic idea of system layout and initial cost through the view, even the performance of comfort control and energy consumption via automatic simulation. Similar to the space access network, a pipe/duct network is introduced to describe the path how cooling/heating energy is delivered to each space. Since the zoning process focuses on space aggregation, the terminal ductwork within each space remains unaltered and is consequently not taken into account.

Bres (2017) devised a loop pattern for the water heating system generation, connecting radiators of each zone with pipes around the floorplan perimeter. Due to the buoyancy effect, the radiators are normally located at the baseboard so the water loop should avoid the circulation area. On the contrary, the air-conditioning system

incorporates sizable equipment and ducts, making it more cost-effective to distribute cool/warm air through the ceiling void, from core to perimeter. With such a tree pattern, the pipe/duct network starts from core shafts or mechanical rooms and then spreads outward via circulation areas to terminal spaces.

Pipe and duct require dedicated space. Typically, the ceiling void in circulation areas is allocated for the installation of electric wiring, ductwork or plumbing. These circulation areas can be identified based on the space access graph in section 3 (Fig. 4-b). In graph theory, the degree of a node denotes the number of edges incident to that node. Thus, the inaccessible shafts have a degree of zero while the corridor usually has the largest degree. Other spaces (degree > 2), such as waiting rooms, may serve as circulation areas in their own functional unit. The space view in Fig. 8 can be drawn following the methods in section 3.

Algorithm 2 Generate Skeletons

Input rooms as polygon set $R = \{P_0, P_1, \ldots, P_n\}$. Within set P_0 represents the outer boundary while others inner holes. η_{min}, η_{max}
Output list of edge set $list\{S\}$

Merge any two set R_1 and R_2 if $R_1[0]$ and $R_2[0]$ overlap, then replace $R_1[0]$ and $R_2[0]$ by the polygon of their Boolean union.
Initiate $list\{S\}$
for each remaining R **do**
    Solve Straight Skeleton by R, build graph $\boldsymbol{G}(V, E)$ from generated inner bisectors, output other bisectors as edge set B
    $d \leftarrow 0$
    for each $\boldsymbol{v}$ in $\boldsymbol{V}$ **do**
        $time(\boldsymbol{v}) \leftarrow$ "time" value of the vertex
        $degree(\boldsymbol{v}) \leftarrow$ the degree of vertex
        if $degree(\boldsymbol{v}) = 1$ **then**
            if $time(\boldsymbol{v}) < \eta_{min}$ **then**
                remove $\boldsymbol{v}$ from $\boldsymbol{V}$
            else
                Find edges e_1, e_2 incident to $\boldsymbol{v}$ from B, then add their bisector as edge to G
            if $time(\boldsymbol{v}) > d$ and $time(\boldsymbol{v}) < \eta_{max}$ **then**
                $d \leftarrow time(\boldsymbol{v})$
    Get offset polygon set C by Straight Skeleton algorithm from R by inward depth of d
    Break up all edges in S by intersection points with C
    Remove edges from S that are inside the region formed by C
    Merge edges in C with S, then append S to $list\{S\}$

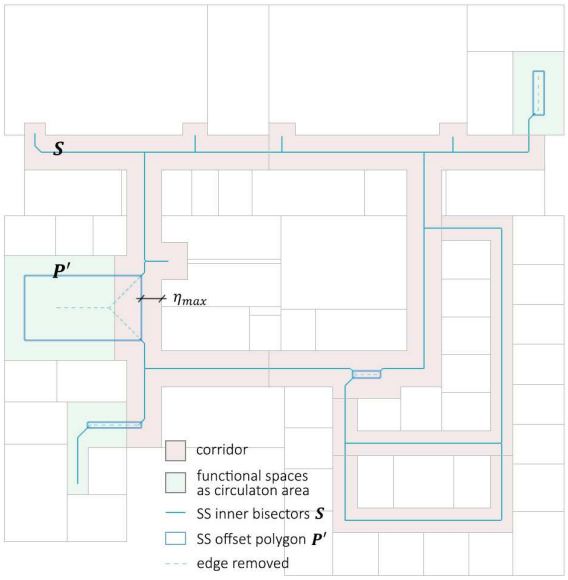

Fig. 8: Circulation areas and the prototype network

Upon the Boolean union of circulation areas, Alg. 2 generates the prototype of potential network by the Straight Skeleton (SS) algorithm. The straight skeleton is defined by continuously moving the polygon edges inwards parallel to themselves at a constant speed. Edges may split in two (split event) or vanish (edge event) due to vertices collision. During this offset process, a set of lines will be traced out by the moving vertices, which is the straight skeleton. It represents the shape well by centerlines, making it suitable for guiding the installation of ductwork. However, for open space such as foyer, the pipe/duct usually walks along the wall (not across the space), which means the offset process should stop at a certain depth. To achieve this, Alg. 2 takes a two-step skeleton generation. Firstly, generate the straight skeleton from the boundary polygon $\boldsymbol{P}$ (Sugihara, 2013), and filter out all interior bisectors as a set $\boldsymbol{S}$. Each vertex in it has a "time" attribute that marks how long it walks during the offset process. Secondly, generate the offset polygon $\boldsymbol{P'}$ by the largest "time" value below the threshold η_{max} (3m, for instance). Thirdly, perform a Boolean subtraction on $\boldsymbol{S}$ using $\boldsymbol{P'}$, and then merge it with $\boldsymbol{P'}$ as the final prototype $\boldsymbol{S}$ (Fig. 8). $2 \times \eta_{max}$ represents the typical width that distinguishes a corridor from a hallway.

In terms of the actual pipe/duct layout, the resulting set of line segments has too many joints, twists and branches. They need further simplification by extension, pruning and alignment. For extension, a vertex generated in an 'edge event' needs to be connected to the mid-point of the edge from which it is collapsed (Fig. 9-b). For pruning, a vertex with too small "time" value below the threshold η_{min} (1m, for instance), as well as the edge incident to it, needs to be pruned (Fig. 9-c). $2 \times \eta_{min}$ represents the minimum width of a typical corridor. Such redundant "branches" are often caused by zig-zag polygon boundaries. For alignment: 1) Considering all prevalent edge directions (edges below length threshold d are ignored), edges in $\boldsymbol{S}$ are grouped by Quality Threshold (QT) clustering with threshold d. The QT clustering takes the maximum cluster diameter as input, finding clusters with guaranteed quality. 2) In each group, an edge can be the alignment baseline only if the total swept area by projecting others to it reaches the minimum. If multiple candidates exist, pick the median. The axis is a line segment that connects all projected vertices along that baseline. 3) The axis will pull the nearby vertices (within the range of d) onto itself while keeping their edge connections. The alignment process aims for the most simplified $\boldsymbol{S}$, by iteratively increasing d until $\boldsymbol{S}$ intersects with any space boundary.

Algorithm 3 Align Skeletons

Input distance threshold d, angle threshold θ, set of edges S
Output set of aligned edges

Initiate $list\langle axis\rangle$
Divide S by Quality Threshold clustering into $\{S_0, \ldots, S_n\}$, taking parameter d and θ measuring the distance and angle between two edges. Edges with length smaller than d are ignored.
for each S' in $\{S_0, \ldots, S_n\}$ **do**
    $box \leftarrow$ outward offset polygon by d
    Group edges if their box overlap, resulting $\{S'_0, \ldots, S'_n\}$
    for each S'' in $\{S'_0, \ldots, S'_n\}$ **do**
        $area_{min} \leftarrow \infty$
        $axis \leftarrow S''[0]$
        for each edge e_i in S'' **do**
            $area \leftarrow 0$
            initiate $list\langle point\rangle$
            for each edge e_j in S'' except for e_i **do**
                $area \leftarrow area$+the area of region swept by e_j projected to e_i
                append projected endpoints to $list\langle point\rangle$
        if $area < area_{min}$ **then**
            $area_{min} \leftarrow area$
            $axis \leftarrow$ new edge spanning all points in $list\langle point\rangle$
        Append $axis$ to $list\langle axis\rangle$
Build graph $\mathbf{G}(V, E)$ from S
for each edge $axis$ in $list\langle axis\rangle$ **do**
    for each vertex $\mathbf{v}$ in $\mathbf{G}(V, E)$ **do**
        if distance between $\mathbf{v}$ and $axis < d$ **then**
            Project $\mathbf{v}$ to $axis$
Output E

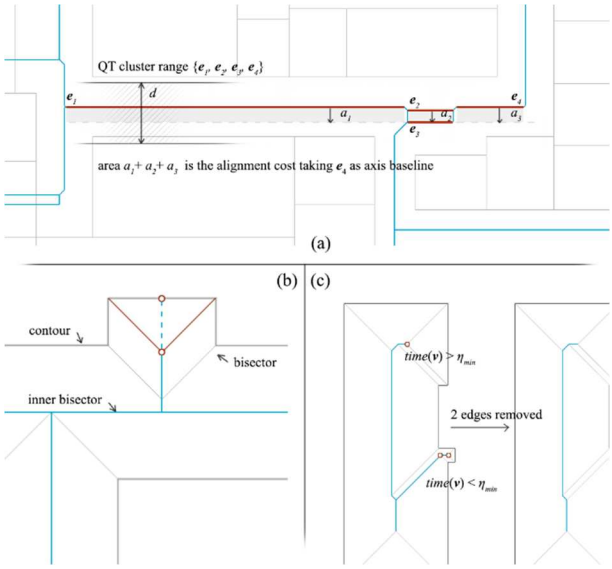

Fig. 9: (a) edge alignment on QT clustering. (b) extrusion of inner bisector with 1-degree vertex. (c) remove vertices with too small "time" value.

In this work, the door locations serve as the entry points to wire in the main network to each space. Such information can be retrieved from BIM (either by IFC or gbXML). If not available (such as a shaft or part of an open space), the centroid of the space region will be used instead. The algorithm finds the Manhattan path between the entry point and the main network by the minimum cost, and then adds them to $\boldsymbol{S}$. Additional penalties will be counted if the path collides with space boundaries. The line segment set $\boldsymbol{S}$ forms a graph $\boldsymbol{G}(V, E)$, serving as the potential network for the detailed layout of pipe/duct system.

According to the hierarchy of zoning outlined in section 1, the terminal nodes (functional spaces) within a thermal zone are connected to a distribution node, representing a variable-air-volume box or air handling unit (AHU). Subsequently, all distribution nodes are connected to the source nodes (shaft or mech rooms). All routes follow the potential network given by an undirected, weighted graph $\boldsymbol{G}(V, E)$. The 2nd-level connection is a Steiner tree problem on terminal nodes $N \subseteq V$ with Steiner points provided as $V \setminus N$. The 1st-level connection can be regarded as the Shortest Path Tree that grows from one source node to all distribution nodes.

Algorithm 4 Get Steiner tree of subset vertices from a graph

Input undirected, weighted graph $\mathbf{G}$, terminal vertices set N
Output Steiner tree graph T

Create the sub graph $\mathbf{H}(V, E)$ by Floyd-Warshall algorithm, making $V \supseteq N$ and E contains all possible path between any $v \in N$
Initiate set V_{del}
Initiate adjacency matrix $A[i, j] \leftarrow e(v_i, v_j)$, $i, j \in |V|$
Initiate adjacency matrix $a[i, j] \leftarrow weight(e(v_i, v_j))$
for k from 1 to $|V|$ **do**
    if $degree(v_k) = 2$ **then**
        $v_i, v_j \leftarrow$ neighbours of v_k
        $A[i, j] \leftarrow A[k, i] \cup A[k, j]$, then $A[k, i], A[k, j] \leftarrow \oslash$
        $a[i, j] \leftarrow a[k, i] + a[k, j]$, then $a[k, i], a[k, j] \leftarrow 0$
        Append v_k to V_{del}
Generate minimum spanning tree $T(V', E')$by Kruskal algorithm based on adjacency matrix $a[i, j]$
for each $e(v_i, v_j)$ in E' **do**
    $e(v_i, v_j) \leftarrow A[i, j]$ ▷ map edges back to H
$V' \leftarrow V' \cup V_{del}$ ▷ bring back relay vertices
Simplify T then output

Algorithm 5 Find centroid of tree

Input undirected, weighted tree graph $\mathbf{T}(V, E)$
Output directed, weighted tree T'

Start form any $v_x \in V$, find the furthest vertex v_0 by Dijkstra algorithm
Start from v_0, find the furthest directed $path = \{v_0, \ldots, v_n\}$ by the same Dijkstra algorithm
Locate the middle point v_m on $path$ by length
$V \cup \{v_m\}$. $E \cup \{(v_{m-1}, v_m), (v_m, v_{m+1})\}$
Traverse T from v_m down to leaf node to build the tree T'

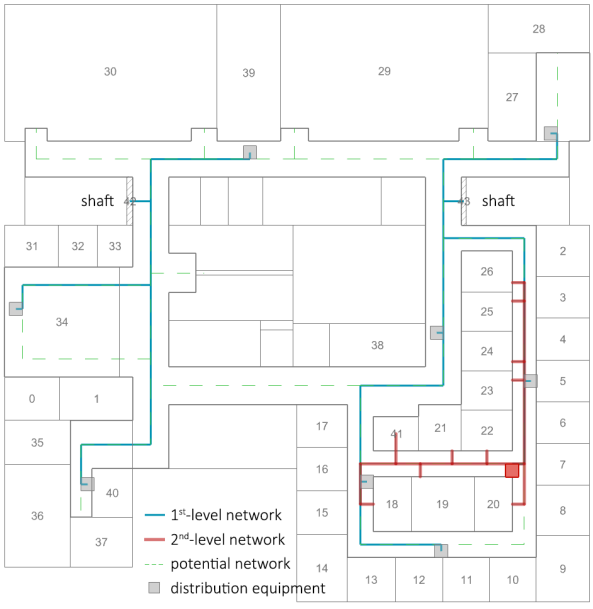

Fig. 10: A sample duct/pipe generation based on the potential network

Alg. 4 describes the steps for solving the 2nd-level connection within a zone: 1) Create the sub-graph $\boldsymbol{H}$ based on $\boldsymbol{G}$, containing the terminals in set N and their possible paths by modified Floyd–Warshall algorithm. 2) Map the graph $\boldsymbol{H}$ to $\boldsymbol{H'}$ by removing relay nodes (degree = 2). 3) Find the MST joining all nodes in the graph $\boldsymbol{H'}$ by Kruskal algorithm, then map it back to $\boldsymbol{H}$. The resulting graph $\boldsymbol{T}$ is the Steiner tree that connects all terminals. To locate the terminal equipment that handles the zone air, Alg. 5 first finds the longest path $\boldsymbol{P}$ by running the Dijkstra furthest path algorithm twice from any node, then adds the mid-point of $\boldsymbol{P}$ as the root node of $\boldsymbol{T}$. The root node minimizes the average path to leaf nodes, implying the minimal cost of distributing cooling/heating air to each space.

All distribution nodes are connected to the source nodes (shaft or mechanical room) along the potential network $\boldsymbol{G}$, in the 1st-level connection. They are clustered by the shortest walk to the nearest source candidate, then get connected by a shortest path tree rooted in that source node (Alg. 4, with Kruskal-MST replaced by Dijkstra shortest-path-tree). It is an edge-weighted breadth-first search that traverses a tree down to every leaf with the same speed.

For simplification, the graph $\boldsymbol{G}$ takes the length of each duct/pipe segment as edge weight. The pressure loss is roughly proportional to the length according to the Darcy-Weisbach equation. Especially in the low-speed ductwork design, a constant pressure loss per unit of duct length is commonly assumed. Moreover, the algorithm ignores many geometric constraints, such as the collision of equipment and ducts in the ceiling void, or the oversized duct that cannot fit in the shaft. Since the generative zoning focuses more on space/system topology than the construction document, it is omitted for our current work.

6. MODEL SCRIPTING

To evaluate the initial cost, energy consumption, and the control effectiveness of zoning schemes at the space level, the co-simulation binding Modelica and EnergyPlus is an ideal choice. *Spawn* (of EnergyPlus) is the latest whole-building energy simulation engine developed by the U.S. Department of Energy, National Labs and industry. It reuses the envelope and daylighting modules of EnergyPlus and couples them with the AC system and control modules from Modelica Buildings Library (Wetter, 2020). This division of tasks optimally leverages EnergyPlus for efficiently solving multi-zone building physics−a task can be time-consuming for Modelica. And, because EnergyPlus takes thermal zone as the basic simulation unit, Modelica is needed for inspecting different behaviors of spaces within a thermal zone.

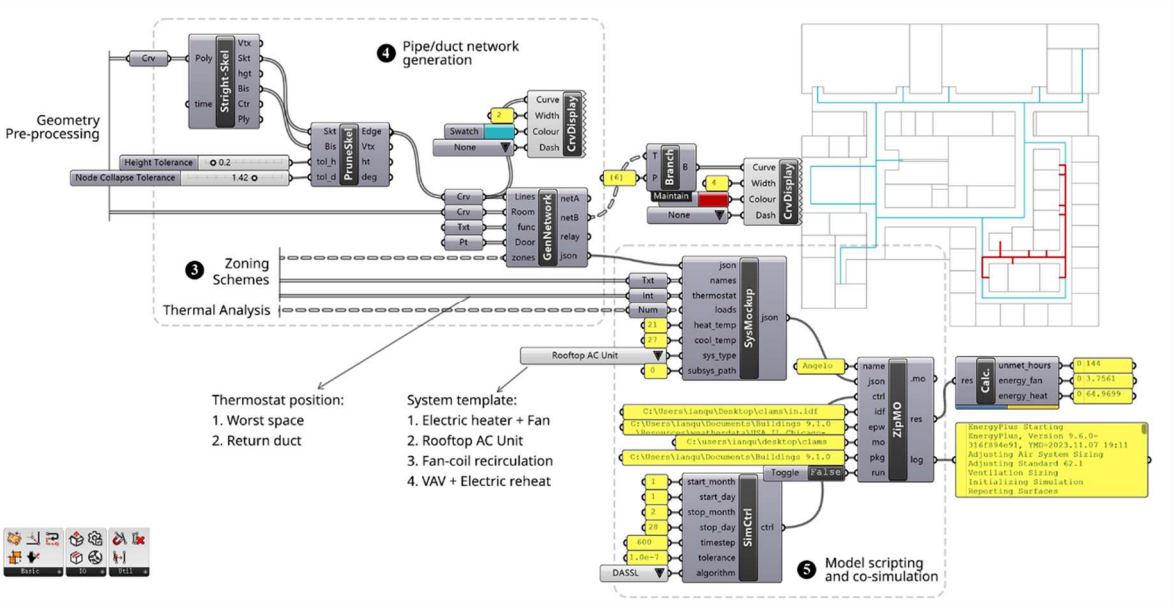

Fig. 11. Workflow of model scripting in developed Grasshopper components.

Based on the typical-day load results from EnergyPlus, the nominal airflow rate is calculated for each space. Then, the algorithm does the equipment and ductwork sizing to meet the nominal airflow. It applies the Equal Friction method to decide the diameter and the pressure loss, which mimics the decision process of engineers.

Following the algorithms in section 2.4, the two-level distribution network, accompanied the functional spaces,

is first serialized in a JSON file (component GenNetwork in Fig. 11). Then, joined with the system and simulation configuration, the JSON is further parsed into the Modelica scripts. The user may customize different systems and control templates in the component SysMockup. When component ZipMO is toggled on, the program will call OpenModelica Shell to perform the simulation and analyze the result.

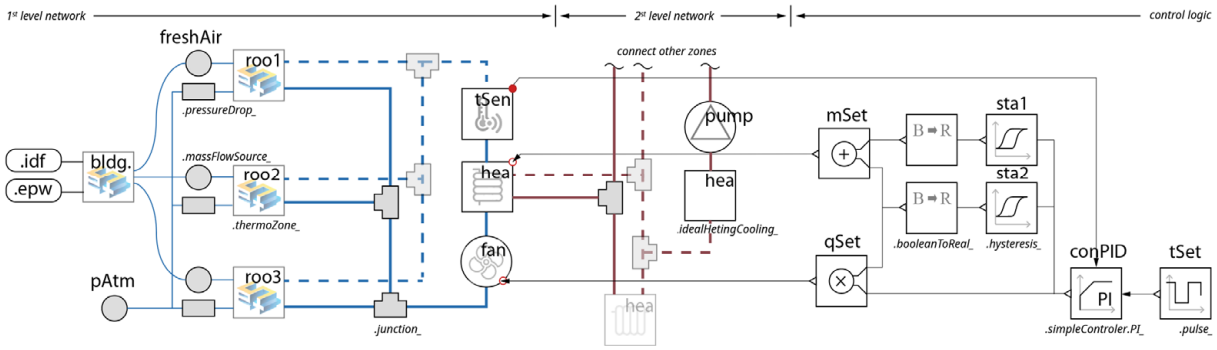

Fig. 12. A sample connection of Modelica model

Fig. 12 gives an example of the generated Modelica model, demonstrated by a recirculation fan-coil system with the water heater and two-speed fan. Each 'thermalZone' component represents a functional space (altogether 43 conditioned spaces), with specific air leakage and fresh air ventilation rate. It exchanges temperature data with the corresponding zone in the EnergyPlus by *Spawn*. The PI controller takes the temperature as input from the sensor mounted on the return duct, and controls the indoor temperature around 21°C during working hours (9:00~18:00). Above the control zone level, a water loop connects all terminal equipment. The fan/pump and heat exchanger are ideal models to give rough estimations. They can be replaced with actual models given detailed parameters, such as performance curves.

Table 2: Simulation results of sample cases

Schemes	Uncomfortable hour (h)	Unmet hour (h)	Fan Power (kWh)	Heat (kWh)	Duct area (m^2)	Pipe length (m)	AHU	Model Equations	Simulation Time (s)
Fig. 4	726	1050	30.82	3317.04	53.60	34.38	9	16857	212
Fig. 7 (a)	1294	1970	35.66	3298.66	78.33	17.54	8	16662	249
Fig. 7 (b)	122	298	33.47	3349.66	24.38	42.67	22	19392	349
Fig. 7 (c)	101	276	34.38	3349.10	17.61	50.75	26	20172	277

The workflow has been tested against 4 zoning schemes for 14 winter days in Chicago. There is a clear trend in Table 2 that the unmet hour decreases along with the increase of control zones. Normally, a control bias of ±1°C is acceptable for comfort conditioning. Hence, the unmet hour accumulates the working hours where the temperature is outside the range of 20~22°C. Similarly, the uncomfortable hour is calculated for temperatures outside the comfort range (20~25°C). In the last case, Fig. 7 (c), 101 uncomfortable hours indicate that one space may get unconditioned for 10 minutes a day on average. However, if 19°C is acceptable, the uncomfortable hour will be zero. Given proper equipment costing spreadsheet, this workflow can offer insights into the advantages and disadvantages of different zoning schemes, in terms of comfort level, initial cost and future energy bills. Such information can facilitate the decision-making process.

7. DISCUSSIONS

This paper introduces a semi-automatic workflow for thermal zoning, the AC distribution system generation, and model scripting. Such a workflow can assist engineers in exploring various zoning schemes and system layouts, taking a step forward toward the generative design driven by simulation. Additionally, it incorporates two unique views into the space view model. One is the thermal response view of spaces (Fig. 6), enabling designers to quickly zone the spaces by color similarity. The other is the potential distribution network view (Fig. 10) that outlines the cost of delivering energy to terminal spaces. However, there are several topics not addressed in this paper, left for future work:

1) The space ontology needs to be enriched, to describe more space relations such as the tenant zone and fire compartmentation. Different tenants must be considered during thermal zoning because their energy

consumption is metered separately. Also, ductwork can be a fire hazard for spreading heat between two compartments.

2) The algorithm for the 1st-level network can be improved, to consider the capacity of terminal equipment and the hydronic balance. It is preferable to evenly distribute the cooling/heating load of zones across multiple risers, ideally making the riser the root node that a tree grows from. If there are outlying spaces in the floorplan with unique load profiles, the algorithm is better to isolate them as distinct system zones (with a direct expansion system for example).

3) More system and control templates are required, for a wide range of system performance comparisons. To a certain degree, the current 2-level network has some universality. Without the 1st-level network, it can be the ductwork of rooftop packaged units. With 1st-level modeled as water pipes and 2nd-level as air ducts, different proportions of water and air can resemble systems from fan-coil (less or no duct) to AHU (more duct). When modeled as air ducts, it suits the constant or variable air-volume system. Each system has specific configurations for model scripting, such as 2-pipe/4-pipe fan-coil, VAV-box reheat, or outdoor air duct.

4) The algorithm lacks the ability to evenly lay out the diffusers and ductwork in an open space. Nevertheless, when the open space has specific function zones allocated, it resembles a multi-room floorplan, with physical partitions replaced by air walls. Under that condition, the pipeline still works given a proper assumption of the airflow rate between spaces (for EnergyPlus modeling in thermal analysis and co-simulation).

8. ACKNOWLEDGEMENT

The authors gratefully acknowledge the financial support from China Scholarship Council in conducting the project.

REFERENCES

Berquist, J., Tessier, A., O'Brien, W., & Attar, R. (2017, May). An investigation of generative design for heating, ventilation, and air-conditioning. *In Proceedings of the symposium on simulation for architecture and urban design.* https://dl.acm.org/doi/10.5555/3289787.3289805

Boskic, L., Brown, C.N., & Mezić, I. (2020). Koopman mode analysis on thermal data for building energy assessment. Advances in Building Energy Research. https://doi.org/10.1080/17512549.2020.1842802

Brahme, R., Mahdavi, A., Lam, K.P., Gupta, S. (2001, August). Complex building performance analysis in early stages of design: A solution based on differential modeling, homology-based mapping, and generative design agents. *In Proceedings of Building Simulation Conference of IBPSA (Vol. 7, p. 661-668).* https://doi.org/10.26868/25222708.2001.0661-668

Bres, A., Judex, F., Suter, G. de Wilde, P. (2017, September). A method for automated generation of HVAC distribution subsystems for building performance simulation. In *Proceedings of Building Simulation Conference of IBPSA (Vol. 15, p. 1548-1557).* https://doi.org/10.26868/25222708.2017.413

Chen, K. K., Tu, J. H., & Rowley, C. W. (2012). Variants of Dynamic Mode Decomposition: boundary condition, Koopman, and Fourier analyses. Journal of Nonlinear Science, 22, 887-915. https://doi.org/10.1007/s00332-012-9130-9

Chen, Z., Guan, H., Yuan, X., Xie, T., Xu, P. (2022). Rule-based generation of HVAC duct routing. Automation in Construction. 139, 104264. https://doi.org/10.1016/j.autcon.2022.104264

Fischer, M., Aalami, F., Akbas, & Akbas, R. (1998). Formalizing product model transformations: case examples and applications. In Artificial Intelligence in Structural Engineering, Information Technology for Design, Collaboration, Maintenance, and Monitoring. 113-132. https://doi.org/10.1007/BFb0030447

Fu, M., Liu, R., Qi, B., & Issa, R. R. (2020). Generating straight skeleton-based navigation networks with Industry Foundation Classes for indoor way-finding. Automation in Construction, 112, 103057. https://doi.org/10.1016/j.autcon.2019.103057

Georgescu, M., & Mezić, I. (2015). Building energy modeling: A systematic approach to zoning and model reduction using Koopman Mode Analysis. Energy and Buildings, 86, 794-802. https://doi.org/10.1016/j.enbuild.2014.10.046

Held, S., Korte, B., Rautenbach, D., & Vygen, J. (2011). Combinatorial optimization in VLSI design. Combinatorial Optimization-Methods and Applications, 31, 33-96.

Medjdoub, B., & Bi, G. (2018). Parametric-based distribution duct routing generation using constraint-based design approach. Automation in Construction, 90, 104-116. https://doi.org/10.1016/j.autcon.2018.02.006

Mezić, I. (2005). Spectral properties of dynamical systems, model reduction and decompositions. Nonlinear Dynamics, 41, 309-325. https://doi.org/10.1007/s11071-005-2824-x

Raak, F., Susuki, Y., Mezic, I., & Hikihara, T. (2016, December). On Koopman and Dynamic Mode Decompositions for application to dynamic data with low spatial dimension. *In Proceedings of IEEE 55th Conference on Decision and Control (CDC) (6485-6491). doi: 10.1109/CDC.2016.7799267*

Roudsari, M.S., & Pak, M. (2013, August). Ladybug: A parametric environmental plugin for Grasshopper to help designers create an environmentally-conscious design. *In Proceedings of Building Simulation Conference of IBPSA (Vol 13. p. 3128-3125). https://doi.org/10.26868/25222708.2013.2499*

Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P., & Henningson, D. S. (2009). Spectral analysis of nonlinear flows. Journal of Fluid Mechanics, 641, 115-127. doi:10.1017/S0022112009992059

Schmid, P. J. (2022). Dynamic Mode Decomposition and its variants. Annual Review of Fluid Mechanics, 54, 225–254. https://doi.org/10.1146/annurev-fluid-030121-015835

Sugihara, K. (2013). Straight Skeleton for automatic generation of 3-D building models with general shaped roofs. *In Proceedings of the 21st International Conference on Computer Graphics, Visualization and Computer Vision.*

Suter, G. (2015, July). Definition of views to generate, visualize, and evaluate multi-view space models of schematic building designs. *In Proceedings of 22nd International Workshop: Intelligent Computing in Engineering, EG-ICE.*

Suter, G. (2022). Modeling multiple space views for schematic building design using space ontologies and layout transformation operations. Automation in Construction, 134, 104041. https://doi.org/10.1016/j.autcon.2021.104041

Visschers, M.G. (2016). BIM based whole-building energy analysis towards and improved interoperability: A conversion from the IFC file format to a validated gbXML file format. Master thesis. Technische Universiteit Eindhoven.

Wetter, M., Benne, K., Gautier, A., Nouidui, T.S., Ramle, A., Roth A., Tummescheit, H., Mentzer, S. & Winther C. (2020, September). Lifting the garage door on Spawn, an open-source BEM-controls engine. *In Proceedings of Building Performance Modeling Conference and SimBuild* (p. 518-525). https://api.semanticscholar.org/CorpusID:221857040