Functional Decomposition
in
Architecture

Ph.D. Dissertation

Emily Ye Zhang

School of Architecture
Carnegie Mellon University
Pittsburgh, Pennsylvania

Thesis Committee

Professor Ömer Akin (Chair)
Professor Ulrich Flemming
Professor Bruce Maggs
Professor Ardeshir Mahdavi

Abstract

Stacking and blocking (S&B) is an important and complex process in building design. Stacking is the process that decides which spaces should go to which floors in a multi-story building. Blocking is a process to decide which spaces should go to which zones for a given floor. As S&B designs a building on a broad scale, it directs the form and general organization of a building. The S&B process is a complex one that necessitates the consideration of many design requirements. While some requirements are related, each design requirement is fundamentally different from the others and thus cannot be handled in the same way. S&B is also a combinatorial problem, in which the number of possible arrangements of spaces is so large that they cannot all be examined individually. Accordingly, S&B is a challenging and extremely critical aspect of building design.

Computerized S&B exists but there is room for improvement. Currently there are three state-of-the-art S&B programs, but their algorithms are weak, and they can handle only a single design requirement, namely that of functional adjacency. These limitations significantly hamper their ability to assist designers in the S&B process.

This research overcomes some of these limitations by critically examining four pertinent research areas with the objective of developing an interactive design aid.

1) It develops a method for handling multiple design requirements, including functional adjacencies, thermal, acoustic, and daylight requirements.
2) It makes use of graph-partitioning and clustering algorithms to automate the processes of S&B.
3) It adopts effective interface features that allow the user to use the system with ease and flexibility. The interface features include drag and drop, progressive disclosure, and dynamic display of algorithmic processes.
4) It enables an extensive interactivity between the user and the system.

The development of the computer program includes the following areas of research:

- Requirements gathering and modeling through protocol analysis, energy simulation, and problem analysis;
- Software design in OMT [Rumbaugh, 1991] and OOSE [Jacobson, 1992], and implementation in C++;
- Adaptation of the Fiduccia-Mattheyses VLSI graph-partitioning algorithm, a heuristic which finds a solution to the combinatorial problem in linear time;
- Interface design in OOSE use cases and interaction diagrams, and prototyping in ET++.

Acknowledgment

It is a great pleasure to acknowledge the assistance and help provided by those who care about this work.

My sincere appreciation goes to Prof. Ömer Akin, my advisor and thesis committee chairman, for his perspective and unqualified support throughout my work on this thesis, and for his insights in guiding me through major challenges during my research. I would like to thank Prof. Ulrich Flemming for his kind advice, encouragement and insight of important research issues. I especially thank Prof. Ardeshir Mahdavi for his constant guidance and stimulating discussions, without which this work would have been impossible. I am grateful to Prof. Bruce Maggs for his clear explanation of the state-of-the-art algorithms and for his help in developing the algorithms, which are crucial to this research.

I am indebted to others who have contributed to this work in a variety of ways: To Dr. Paul Mathew and Ph.D. fellow student Rohini Brahme for their cooperation in conducting the thermal experiment, to another Ph.D. fellow student Robert Ries for his generous help and support in the thesis and presentation, and to all the members of SEED team for providing me relevant information for this study.

I am very thankful to Prof. Robin Liggett, who has done a marvelous pioneering research in this field. She has helped me without any reservation in answering my questions, explaining her algorithms, and sending me her thesis and software. I hope that she will be as gracious as she always has been to the analysis of her work in this thesis.

As a personal note, I would like to thank Dr. Henry Rowley in his help in answering challenging technical questions, and to Dr. John Dolan in all the insightful ideas about my thesis proposal and thesis.

Summary

Computer support in the building design field is in its infancy. Currently available software programs only assist designers to visualize and represent an existing design or design idea. They do not play a role in decision making. Designers have to rely on their expertise and experience to manually create designs.

As one considers how modern computer technology has been applied to certain design fields, such as VLSI, mechanical design, and structural design, and has improved productivity, the lack of computational support in the building design area has become more obvious.

This thesis aims to bridge the gap in the area of computer-assisted building design, specifically in an important design stage called stacking and blocking (S&B).

Stacking is the process that decides which spaces should go to which floors in a multi-story building. Blocking (or zoning) is a process to decide which rooms should go to which zones for a given floor. Together, the role of S&B in a building design process is vital: It directs the form and general organization of a building. Since it happens early in design, decisions have rather significant impact on later stages of design.

S&B is a complex process in which various design requirements are considered. These requirements include functional adjacencies, and thermal, acoustic, and daylight requirements. Since each dimension of the requirements has a different nature, different requirements may result in potentially conflicting designs. In a hospital, for example, an emergency room may require close proximity and convenient access to an entrance according to its operational (or adjacency) requirements, but according to acoustic requirements, a quiet emergency room should be away from a noisy entrance. For a building over a certain size, there may be many such potentially conflicting requirements that need to be resolved during the design process.

Since S&B is both important and complex, research aimed at automating S&B has been conducted since the late 70s. The most prominent research is known as WinSABA (Windows-based Stacking and Blocking Algorithm) conducted by Professor Robin Liggett at UCLA. WinSABA allocates spaces onto floors and zones according to the adjacency relations between the spaces. It has a graphical user interface that allows the user to interact with the system: For example, when the user selects a room from a list of rooms to be assigned to floors, the system shows what locations are ideal for that room, what locations are prohibited for assignment of that room, etc. Then the user can pick a location and the room is assigned there. WinSABA is pioneering work in

demonstrating that building design problems, when properly formulated, can be assisted by computers. WinSABA also sets a benchmark in computerized S&B by providing algorithmic solutions and a graphical user interface.

But the deficiencies in WinSABA and other state-of-the-art S&B computer programs hamper their usefulness. Because they only consider adjacency requirements, these design systems do not handle other important design parameters such as thermal, acoustic, and daylight requirements. Thus the process of weighing trade-offs amongst different criteria, and exploration of relationships between important performance requirements and design decisions are missing. Their algorithm is time-consuming and does not produce desired output. Since the user has to perform complicated interface operations before he/she is able to make a change in the design, the level of interaction between the user and the system is limited. As design assistance, these programs do not provide flexibility for the user.

This thesis remedies some of the deficiencies in the state-of-the-art and develops an efficient method of handling S&B. Specifically, it makes improvements in four areas:

- It develops an efficient graph-partitioning algorithm for stacking, and a clustering algorithm for zoning.
- It handles multiple design criteria including not only functional adjacencies, but also thermal, acoustic, and daylight requirements that are all crucial to S&B.
- Extensive interactivity allows a user to combine his/her design knowledge with computer's fast computing capability to efficiently create designs.
- It has a user interface that is easy to use. Using interface techniques such as direct manipulation, progressive disclosure, and dynamic display, the interface provides an easy access to the design information in the system and allows the user to conveniently modify a design.

As the program demonstrates, improving automation algorithms and incorporating multiple design criteria both increases the depth of the knowledge in S&B and expands the boundary of the capabilities of state-of-the-art S&B programs.

Table of Contents

List of Figures

Figure D.14 Interaction diagram of viewing FU requirements D-11

List of Tables

1 Motivation and Background

1.1 Background

When designing large, multi-use buildings, architects face the formidable task of intelligently grouping spaces with similar or related functions. The task of grouping spaces is difficult because there are many variables to consider: there are many functional requirements (e.g., adjacency, thermal, and acoustic), and for each of these requirements there are different interrelations between the spaces. For instance, an adjacency requirement may either require spaces to be located right next to each other (strict adjacency), or spaces to be located within a certain distance.

To illustrate the process and its complexity, consider the following example. When designing a university building, architects may have hundreds of spaces to be allocated. With many different ways of grouping those spaces, architects will typically group together all of the office spaces of faculty members within one academic department, group together all of the offices spaces of students within one academic department, etc. They may also group spaces with the same temperature requirements, or segregate spaces with incompatible acoustic requirements. When this is done, there will typically be fewer groups than spaces. The strategy of organizing spaces into groups reduces the number of elements that the architects have to consider, making the task more manageable. Architects also have to deal with a large number of ways that groups can be arranged in the building. They may assign the groups to the same floor, or may stack them one on top of another according to their functional relations or for formal purposes. Thus a rough organizational plan of the building is achieved, and the approximate dimension and shape of the building can be determined. While different architects may have individual approaches to spatially realizing an architectural program, it is plausible to assume that their approaches can benefit from the computational support accomplished in this work.

The process described in this example is typically referred to as a stacking and blocking task. Stacking is the process of grouping spaces into different floors according to certain functional requirements and design constraints such as floor area and areas of spaces. Blocking is the process of dividing the functions on a single floor plan into groups or blocks. The blocks should be architecturally meaningful, for instance, entrance block, reception block. Blocks can be organized hierarchically. Each group of spaces can be a block or a sub-portion of a block. Together the stacking and blocking strategies help decompose a large-scale design problem into more manageable (multiple but smaller) design problems. The next step to blocking is a more refined design stage, i.e., layout design, which is the process of giving spaces exact locations with precise dimensions within a given block.

Stacking and blocking (S&B) is an important phase of the architecture design process. Decisions made in S&B have a significant, if not decisive, impact on the later stages of design. However, while important, it is also very difficult. It is a combinatorial problem. It would be desirable if all the possible ways of arranging spaces to floors and blocks could be considered before arriving at the best arrangement or all the alternatives of good arrangements. When a building size is large, it is extremely difficult, if not impossible, to enumerate all the possibilities. In addition, there are various functional requirements to consider. Each functional requirement is fundamentally different from others, and should be handled in a different way. For example, adjacency requirements are different in nature from thermal requirements. For adjacency requirements, spaces with close geometric distance relations should be grouped. For thermal requirements, spaces with similar thermal requirements should be grouped. Even within a single functional requirement, there are different relations between spaces. For example, acoustic requirements could require spaces to be located together, or to be segregated.

It is hard for a human designer to handle so much information completely and efficiently; it is also time-consuming. While computer technology has been applied to certain areas of design other than building design (e.g., VLSI design) and has increased design productivity [Boyer et al., 1989][Trimberger, 1989][Gelsey, 1992], currently little research has been done on computer-assisted S&B. Therefore this research is motivated to develop effective computer support for S&B.

Because the software is to be developed for architects, their input to the design of the software is essential. An interview with architects has been conducted in order to identify their needs for such a computer program and to obtain requirements on how to design the software. This following section describes this interview and its findings.

The goal for the interview is to obtain the information in four areas below:

- Does S&B play an important role in the building design process?
- Do architects want software of this type?
- What building design requirements are considered in S&B?
- What features in a software tool do architects want?

I visited four AIA[1] Pittsburgh member firms and interviewed one architect in each firm. I interviewed them and recorded important points they made. I asked each architect the same set of questions:

- Is S&B important?
- Is computer support necessary?
- What building design requirements do architects consider in S&B?
- What features in a CAD tool do architects like?
- In what way do architects imagine they would use such a tool?

1. AIA stands for American Institute of Architects.

Besides these questions, I also asked some questions according to the conversational context. For details of the interview, refer to Appendix A for a complete set of questions-and-answers. The following is a summary of the responses:

- While S&B is defined slightly differently by each individual architect, it is a very important phase in architectural design. It defines the relationships between spaces and improves the functionality of a building.
- Computer support is needed. Computers can quickly generate solutions and are more efficient, but architects hesitate to use software for the following reasons: *a)* Computers must violate some requirements in order to get a globally optimal solution. As a result, architects must accept violations. *b)* There are certain criteria other than structural, mechanical, and acoustic requirements that computers cannot handle, such as circulation flow and aesthetic judgments. *c)* Computers are cumbersome to use. For example, in order to draw a stairwell, a user has to enter about 10 numbers, by the time the computer draws this stairwell, architects would have finished drawing the stairwell manually.
- All the interviewees agreed that functional adjacencies are important in both stacking and blocking; mechanical (thermal and plumbing) and acoustic requirements are useful in blocking.
- As concerns features, architects like software that is easy to use and produces precise results without complicated user inputs required.

These findings justify developing computer support for S&B with an exception that some architects are hesitant to use computer tools in designs. This thesis addresses the research issues associated with the three reasons for the hesitation as listed above.

To alleviate hesitation caused by computers violating some design requirements, the software should provide architects with the flexibility to not accept those violations. To achieve this, an interactive design approach is vital. Computers automate major aspects of design requirements and provide a reasonable starting point. The user can then modify the design.

The software should provide an interactive design approach that combines the computer's fast processing capability with the human designer[1]'s aesthetic and visual judgments.

The cumbersome image of computer applications reflects the fact that the current CAD tools in architecture are not easy to use. This raises an important research issue in user interface design. A good system should provide ease and flexibility for the user instead of letting him / her follow some special rules of the system, such as special format and order.

1. A designer, user, and architect refer to the same group of people and will be identified as appropriate in the thesis.

It is clear that the negative response to using computers in architecture firms is due to poor user interface design and limitation in user interaction. Therefore two of the major research objectives are to provide an easy-to-use interface, and to improve user interactivity.

In summary, through this survey, important requirements for software assistance in S&B were identified. They include:

- candidate S&B criteria that include adjacencies, thermal, acoustic, daylight, plumbing, and structural requirements;
- an iterative design approach that allows the user to modify designs devised by the computer;
- an intuitive interface that allows the user to conduct S&B with ease and flexibility.

Because of the importance of S&B and the great potential in developing computerized support in this area, several research institutions and firms have developed S&B software. However, the existing software is inadequate in a number of important aspects, as will be shown next.

1.2 Prior Work

Research on stacking and blocking (S&B) started in the late seventies [Liggett, 1978][Liggett, 1979][Teicholz and Sena, 1986], but significant advances were not made until a decade ago. Past attempts include generating stacking diagrams, allowing the user to directly manipulate interface elements to facilitate the interaction between the user and the system, and using an adjacency matrix to represent functional adjacencies between different departments/activities.

S&B programs are used for two main purposes: facilities management (FM) and building layout design. An FM program manages people, assets, and spaces belonging to an organization throughout the lifetime of a building. This involves managing the proximity issues between functions of a building by grouping activities based on adjacency relations. A building layout program arranges spaces within a building according to their functional requirements. All existing S&B systems have been developed for FM purposes, including WinSABA, a system that can also be used for building layout design.

Since all S&B programs are part of a computer aided facilities management (CAFM) system, their pace of development closely mirrors that of CAFM. This section first gives an overview of state-of-the-art CAFM systems, then reviews the following three state-of-the-art computerized S&B systems: WinSABA[1] (SABA Solutions), OrgTree Stack and Block[2] (Decision Graphics), and WinStack[2] (Peregrine Systems, formerly Innovative Tech). Following the survey is a summary of the strengths and weaknesses of the state-of-the-art in computerized S&B pro-

1. WinSABA was at first developed from academic work by Robin Liggett at UCLA. In 1995, Robin established the company SABA Solutions; and WinSABA has since become a commercial product.
2. OrgTree Stack and Block, and WinStack are both commercial products.

grams, specifying the challenging problems existing S&B programs have solved and which challenging problems remain to be solved.

1.2.1 Overview of CAFM Systems

Most S&B programs are primarily integrated into CAFM systems. CAFM systems assist organizations in the management of their facilities, including people, assets and spaces, in order to achieve maximum economic benefit and organizational efficiency. CAFM systems are important for the overall performance of organizations for the following reasons:

- It is time-consuming to manage the large body of facilities information manually.
- It is difficult to efficiently integrate operations between different departments manually.
- CAFM systems can avoid redundant work by recording the work that has already been assigned or done, whereas in a non-computerized environment, it is not convenient to keep track of the status of a job regarding whether it is done, to whom it is assigned, etc.
- CAFM systems can identify under-utilized or un-utilized spaces so that these spaces can be made use of, whereas it is not as easy to do so manually.
- CAFM systems can become an important strategic decision aid by producing “what if” scenarios, whereas it is error-prone and inefficient to produce such scenarios manually.

CAFM is still in its early stages of development. Since its inception in the 1970s, only the last decade has seen significant advances in this field. The latest generation of CAFM programs integrates CAD drawing tools with the facilities database. This integration includes a bi-directional link between database and CAD tools. CAD tools not only output data contained in a database, but also accept user inputs into the database. Besides integrating CAD tools with the database, CAFM programs also have interfaces with powerful features (such as drag and drop), automated FM functionalities, and sockets for integration with third-party software.

State-of-the-art CAFM programs include Aperture Enterprise Solutions developed by Aperture Technologies, Inc., Archibus/FM10 developed by Archibus, SPAN.FM developed by Innovative Tech, FM:Space developed by FM:Systems, and AutoFM developed by Decision Graphics.

It should be pointed out that a state-of-the-art CAFM system does not necessarily host a state-of-the-art S&B program; and a state-of-the-art S&B program does not necessarily reside in a state-of-the-art CAFM system. WinSABA is such an example: it has advanced S&B functions but it is not state-of-the-art CAFM. On the other hand, OrgTree Stack and Block, and WinStack, are both advanced in their S&B capabilities and within a powerful CAFM environment. The following section reviews these three S&B programs closely.

1.2.2 Survey of Three State-of-the-art S&B Programs

1.2.2.1 WinSABA

SABA (Stacking and Blocking Algorithm) was developed by Robin Liggett in 1979 [Liggett, 1979] and has since become the most widely used algorithm for stacking and blocking. After being equipped with an interactive user interface for Windows, SABA was developed into a commercial product called WinSABA in 1995 [SABA Solutions 1996]. WinSABA uses an adjacency matrix to represent adjacency relationships. The user can assign activities[1] to locations before running the algorithm, and modify a solution after running the algorithm.

SABA was formulated as a "Quadratic Assignment Problem" originally defined by Koopmans and Beckmann in 1957. Since the Quadratic Assignment Problem is a NP-complete problem, Liggett used Graves and Whinston's heuristics to make SABA realizable in polynomial time at the cost of not getting an optimal solution.

SABA handles one-to-one assignment[2], stacking, and "blocking". "Blocking" in WinSABA is actually layout design. It allocates spaces on a floor with physical locations and dimensions. Our notion of a blocking problem, on the other hand, is to let the user divide a floor plan into multiple zones so that a large-scale design problem is divided into more manageable, multiple and smaller design problems. SABA goes beyond this and solves the layout problem, too.

SABA's algorithms need improvement. SABA considers two types of relations: fixed costs and interaction costs. Fixed costs are costs of assigning an activity to a particular location. Interaction costs are travel costs and / or costs of shipping materials. SABA was originally designed to handle one-to-one assignment problems. The stacking and blocking algorithms are adapted from the algorithm for the one-to-one assignment problem.

The one-to-one assignment algorithm decides the best locations for spaces if they are to be allocated to the same number of fixed locations. There are fixed costs associated with each space and each location. There are also interaction costs between different spaces. The goal is to minimize the total costs including both fixed costs and interaction costs.

When dealing with stacking and blocking, SABA divides a single grouping process into a two-stage mapping process which consists of a constructive stage and iterative improvement. The constructive stage is divided into an n-step mapping process, where n is the number of spaces to be allocated. The time order for this process is O(n^3). Also, each space is unnecessarily divided into an integer number of smaller modules and each floor or zone is divided into an integer number of smaller grids whereas the area of a module in a space is equivalent to that of a grid on a floor or zone. In this process, the area of a space is approximated to the sum of

1. Here, an activity refers to a space in which a specific activity is carried out.
2. One-to-one assignment in WinSABA is, for n activities and n locations, to assign each activity to a unique location so that each activity is mapped to only one location, and vice versa. In such an assignment, shapes and dimensions of the locations are not considered.

those of the small modules, and the area of a floor or zone is approximated to the sum of those of the small grids. The process of assigning a space to a floor, for instance, is actually a process of mapping the small modules in the space to the same number of small grids on the floor. This creates a dilemma between precision and efficiency. In order to increase precision of a solution, a space should be divided into as many small modules as possible, which may increase the computing time infinitely. In order to increase efficiency, a space should be divided into as few number of small modules as it can, which will unavoidably result in an imprecise solution. SABA generates an inadequate solution because at each step, when a new space is to be assigned, only the relations between this space and the assigned spaces are considered. The relations between this space and the rest unassigned spaces in the "waiting list" are only partially considered through an "implicit enumeration" method in which an allocation of a space is made when this allocation will most likely achieve an optimal solution against the objective function. This statistical method does not produce an optimal solution when the allocation of the rest unassigned spaces does not follow the "most likely" pattern.

In the iterative improvement stage, a user can modify the design devised in the constructive stage by exchanging pairs of small modules in order to fix the irregular shapes of spaces.

Because of the execution time problem, SABA currently can only handle fewer than 175 spaces in stacking and blocking.

WinSABA's interface is intuitive, interactive and relatively easy to learn. Specifically it has the following three aspects:

- Adjacency relations are categorized into different levels according to their importance, such as "absolute", "essential" and "important". This makes it easy for the user to decide and enter the strength of an adjacency.
- When a new space is to be allocated, the adjacency relations between this space and all the fixed spaces are represented as links with different thicknesses corresponding to different strengths of relations, providing the user an aid in deciding good location(s) for the space.
- A coloring scheme is used extensively. Adjacency links can be represented not only by different thicknesses of lines, but also by different colors of lines with the same thickness. In the interactive mode, when a new space is to be allocated, the preferred, discouraged and prohibited locations are shown in different colors. These intuitively assist the user in deciding good location(s) for the space.

WinSABA has the following five disadvantages:

- First, WinSABA uses a time-consuming algorithm, i.e., SABA, to generate a solution that turns out to be inadequate. SABA is time-consuming because it divides a single grouping process into a multiple-stage mapping process. The time order for this process is $\Omega(n^3)$[1]. SABA's solution is inadequate because at each step when a next space is to be assigned, only the relations between this space and the fixed spaces are considered. The relations between this space and the rest unassigned spaces in the "waiting list" are only partially considered through an "implicit enumeration" method.

- Second, WinSABA does not handle multiple functional requirements. It only deals with adjacency requirements.
- Third, the so-called "blocking", i.e. layout design, result is not feasible. Because of the high penalty for allocating a space across zones, an inter-zone corridor cannot be allocated. A typical layout will look like spaces piled upon one another with irregular shapes. It is not useful as a starting point for layout design, either, since it cannot be appropriately modified in WinSABA. The iterative improvement process focuses on pruning the irregular shape of each space. The basic plan pattern cannot be easily modified.
- Fourth, in WinSABA, "blocking" (i.e., layout) is not a follow-up step to stacking as it should be. After stacking is done, the user cannot select a floor and continue with layout design of that floor. He / she has to set up a data file recording the information of all the spaces assigned to the selected floor and start a new layout project. This difficult process makes the layout portion nearly unusable.
- Lastly, WinSABA's interface has three drawbacks. First, it only supports very limited direct manipulation. The user cannot directly manipulate interface elements in order to perform a certain operation. For instance, if the user wants to move a space from its current floor to another, he / she has to click on it first, and this will pop up the requirements window of the space. The user then has to select "Move" in the window, click on the destination floor, and select "OK" in the window again before this move operation is completed. Second, the interface violates WYSIWYG principle. It does not grey out all the illegal operations. For instance, "Finish Plan" option, which should be activated only before the spaces are assigned, is still activated when all the spaces are assigned, in which case if selected will cause the system to crash. Third, there are some confusing or ambiguous interface elements. For instance, if the user wants to exit the "improving plan" mode, he / she has to select "improve plan" in order to exit from this mode. The meanings of some symbols are unclear. For instance, "s" before a space item (this space is split as discontinuous parts) and "$" before a space item (there are preference costs associated with this space for assignment).

1.2.2.2 OrgTree Stack and Block

OrgTree Stack and Block [Decision Graphics, 1997] creates building stacks (a side elevation view of the building) showing how the spaces are utilized. Furthermore, working relationships and target floors may be set against people or departments using the adjacency matrix. These working relationships can be viewed rapidly using bubble charts which show the rela-

1. "$\Omega(\boldsymbol{n^3})$" is a time order greater than n^3. The amount of additional time will depend on the precision with which the areas of spaces are modeled.

tive proximity of the key relationships. An algorithm can generate stacking according to the relations specified in the adjacency matrix to get the best fit for the building. Some utilities are provided for the user to modify the stacking and blocking result. For instance, "drag and drop" assists a user to move blocks among floors, and "break and glue" either breaks up blocks by head count or glues them together as a single block.

Despite these features, OrgTree Stack and Block has the following disadvantages:

- The system does not provide options of different levels of adjacency relations. It only provides and handles the strict adjacency relation, i.e., two spaces being right next to each other. In cases when two spaces should be allocated within a certain distance, the user has to approximate it either to" adjacency" or to "no relation". This significantly limits the usefulness of the system.
- Blocking is simplified as "departmenting". By default, each department is a single block. Furthermore, blocking can only be performed manually. The user can choose to "break" a department into two or "glue" two departments to join them into one. There is no system assistance in a blocking process except that when splitting a department, the system makes the area calculation on its own by prompting the user for the personnel to be sent to each sub-department and using its database of space requirements for all personnel.
- It does not handle multiple functional requirements. The only functional requirement it handles is adjacencies.

1.2.2.3 WinStack

WinStack is the stacking module in the CAFM program, SPAN.FM. The primary task of WinStack is to produce stacking diagrams. The input to WinStack is provided by two separate yet related modules, Property Portfolio and Space Analysis.

In Property Portfolio, a tree structure represents the real-life organizational structure. Adjacency requirements can be specified between different departments. The leaf level is composed of personnel. Attached to each person is a job status, which is assigned a default space class and type. For each space class and type, there are certain dimensional constraints. Thus organizational requirements are transformed into spatial/dimensional requirements at this point.

In Property Portfolio, the user has to define the spatial structure in which stacking is going to take place, i.e., into which spaces given departments can be allocated. Spatial structure is decomposed into 7 levels, namely: land, owners, buildings, floors, areas, rooms/sublocations, and bins (furniture level). The user has the option to allocate physical locations of any department among buildings and floors (a large department may be split into different buildings and/or floors). This provides the ability to pre-determine decisions regarding the assignment of spaces.

In the Space Analysis module, the user specifies a time, such as month and year, for which the stacking scenario is created. According to the time specified, WinStack can represent past/current space allocation, or forecast future space needs.

Before stacking diagrams are produced, however, the user must also specify floors to be included in the analysis and floor stacking order (the order in which different floors are stacked). Then using the selected departments' dimensional requirements and adjacency relations as defined inside the organizational hierarchy, and the time specified in Space Analysis, WinStack generates "the best fit", i.e., the most efficient setup stacking diagrams (this automatic generation functionality has not yet been implemented).

WinStack provides a graphical floors section interface. The user can re-arrange departments by moving, swapping, splitting and/or joining the departments. The user can also modify adjacency requirements in the organizational hierarchy by modifying the affinity matrix.

WinStack needs improvement in the following areas:

- It does not handle blocking.
- It does not handle multiple functional requirements except adjacency.
- It does not yet automatically find solutions.
- Its interface should clearly indicate which operations are valid and which are not instead of allowing the user to perform an illegal operation before the system warns him/her that it is not a valid operation. For instance, in joining two departments, after a user selects two departments and tries to join them, sometimes a warning dialog window appears telling the user that the two departments he/she has selected cannot be joined because of one of the three listed reasons. A suggested solution is to highlight the departments that can be joined in the same color after the user selects the "Join" command, so that user can select departments with the highlighted color and will not get the warning dialog[1].
- The seven-level spatial decomposition is inappropriate. The first two levels, land and owners, do not fit into the spatial decomposition category. They can be put into the property profile instead.
- Floor stacking order is not necessary. After the user selects floors that he/she wants to include in the analysis, it becomes obvious how the floors should be stacked. For instance, for a twenty story building, if the user selects floor 1, floor 7 and floor 18 to be included in the analysis, it is clear that floor 1 should be on the bottom, floor 7

1. If this suggested method for "joining" departments is taken, in order to keep consistency in the interface design, the other operations should all follow the order of "selecting action before selecting objects". For instance, to swap departments, the user should select "Swap" before he/she selects a pair of departments for swapping; to split a department, the user should select "Split" before he/she selects a department that can be split.

should be in the middle and floor 18 should be on the top. The user does not need to specify this top-down order.

1.2.3 State-of-the-art in Computerized S&B Programs

To date, S&B programs have achieved the following: (1) development of the basic S&B functions and related heuristics to approximate the combinatorial problems, (2) partial graphical user interface development, and (3) representation of adjacency relations in an adjacency matrix.

The existing S&B programs have the following weaknesses:

- Very few existing systems automatically find S&B solutions. Those that do provide an inadequate solution, using a time-consuming algorithm. For instance, WinStack does not have an algorithm to support its stacking process; SABA in WinSABA is time-consuming and eliminates a lot of good solutions by only explicitly considering a subset of adjacency relations at each step of the n-step constructive process.
- None of the systems provides good support for blocking. For instance, OrgTree Stack and Block only assists the user in splitting a department or joining two departments. WinStack does the same. WinSABA, as discussed, goes a step beyond blocking but produces infeasible layouts, and the interface does not support connection from stacking to "blocking" (i.e. layout).
- Existing S&B systems cannot handle multiple functional requirements. They only support adjacency requirements.
- The systems allow only limited user interaction. The iterative process in the normal design practice, i.e., the "back and forth" process of designing, evaluating the design, then going back to modify the input and creating a new design is not supported. WinSABA and WinStack do not allow the user to modify the adjacency / affinity matrix and have the system generate new designs.
- State-of-the-art S&B systems generally have weak user interfaces. In WinSABA not all illegal operations are grayed out, and direct-manipulation is mixed with selections of some table items; in WinStack, a warning dialog is provided when a user has done an illegal operation.

1.3 Research Objectives

This research aims to develop a computational S&B program, called functional decomposition in architecture, hereafter FD, in order to remedy some of the shortcomings in the existing S&B programs:

- This thesis will expand computerized S&B by handling multiple design requirements including functional adjacencies, thermal, acoustic, and daylight requirements.

- This thesis will view S&B as an active participant in layout design. FD will be able to localize a layout design within a single floor or zone so that a large and complex design problem can be "divided and conquered".
- This thesis will enable an automatic link with architectural programming (AP) software so that it can easily get input from such software. The case here is SEED-Pro [Akin et al., 1995]. SEED-Pro is a software system that formulates building requirements in an object-oriented format. By representing design data in the same format, FD can get requirements from SEED-Pro directly.
- Computerized FD will become a seamless transition between AP software and layout design software.

Specifically, there are two goals that this research intends to accomplish. The first is to provide a prototype design system in S&B integrated into the SEED [Flemming and Woodbury, 1995] software environment currently under development at Carnegie Mellon University.

The second goal of the research is, through system design, to gain an increased understanding of the process of S&B, and of the iterative design processes. This will also build a foundation for future research and development.

Both objectives will be accomplished in the context of SEED.

2 Overview of Functional Decomposition in SEED

"Decomposition" (in "functional decomposition") refers to a S&B process, i.e., the process of dividing a large building design problem into smaller design problems comprising floors and zones. "Functional decomposition" refers to the process of decomposing a building into floors and zones according to various functional requirements[1] such as adjacency and thermal. This section provides an overview of FD, which was developed to operate in the SEED environment.

2.1 SEED Environment

The SEED project (**S**oftware **E**nvironment to Support **E**arly Phases in Building **D**esign) is an ongoing multi-disciplinary effort to "provide support, in principle, for the preliminary design of buildings in all aspects that can gain from computer support" [Flemming and Woodbury, 1995]. Its primary emphasis is the generation, exploration and maintenance of design alternatives from each of the disciplines involved in the early design.

SEED features an open-ended modular architecture, where each module provides support for a design activity that takes place in early design stages. SEED contains SEED-Pro Module [Akin et al., 1995], SEED-Layout Module [Flemming and Chien, 1995], and SEED-Config Module [Fenves et al., 1995] [Woodbury and Chang, 1995]. SEED-Pro supports the generation of an architectural program from building requirements expressed by the client and / or in applicable construction standards. SEED-Layout supports the generation of the architectural layout of building elements, such as rooms and corridors. It specifically supports the layout of zones through decomposition hierarchies as described in this thesis. SEED-Config generates three-dimensional configuration and provides support for conceptual structural and enclosure design.

FD was developed within the SEED project. However it can also be used as a stand-alone system. There are different advantages and disadvantages of these two approaches.

FD will benefit from being integrated in the SEED environment in the following respects: FD needs as input many functional requirements, including functional adjacencies, thermal and acoustic requirements, which are all available from SEED-Pro. If FD is integrated with SEED-Layout, designers can use layout designs to visualize or to experiment with the results of FD.

There are also benefits of having FD as a separate system. It will be flexible enough to be used as a tool for architecture design and for FM purposes. Being stand-alone facilitates portability to different machines and platforms.

In its prototype stage, FD was developed as a part of SEED. As the system grows, a stand-alone version can be developed.

1. Functional requirements and design requirements / criteria / constraints are factors considered in the design. They are used interchangeably in this thesis.

The SEED approach is based on the following concepts that are important for this research: specification unit, functional unit, functional unit hierarchy, and design unit. A specification unit defines functional requirements of either a physical space or an organizational entity such as a department. A functional unit represents a set of functional requirements of a physical space. The requirements could be in the form of constraints and criteria regarding the shape, size, etc. of the physical space. A functional unit can be a space or room, a zone, or a floor. As a specification unit is abbreviated as SU, a functional unit is abbreviated as FU. An FU hierarchy is a hierarchical structure used to represent constituent relations between FUs. Each component in an FU hierarchy is an FU. A parent FU spatially contains all its children FUs. This is recursively applied to all FUs. An FU hierarchy is thus able to represent both stacking and blocking decisions in a unified representation. This enables SEED-Layout to move seamlessly between the two processes. A design unit is, in the present context, a physical space with a certain dimension and location on a floor plan. A design unit is abbreviated as DU.

2.2 FD System Architecture

FD deals with multiple functional requirements, including functional adjacency, thermal, acoustic, and daylight requirements. Figure 2.1 shows the FD system architecture. The FD system has three parts: input, the internal engine, and output. The input has a linear set of FUs to be allocated within a building. Each FU in the input contains multiple building design requirements. In the internal engine, the four functional criteria are identified, and represented in computer data structures, before they are combined by the merging engine. The combined requirements are then processed by the decomposition algorithms. An S&B design generated by the decomposition algorithms is represented in the form of an FU hierarchy in the output.

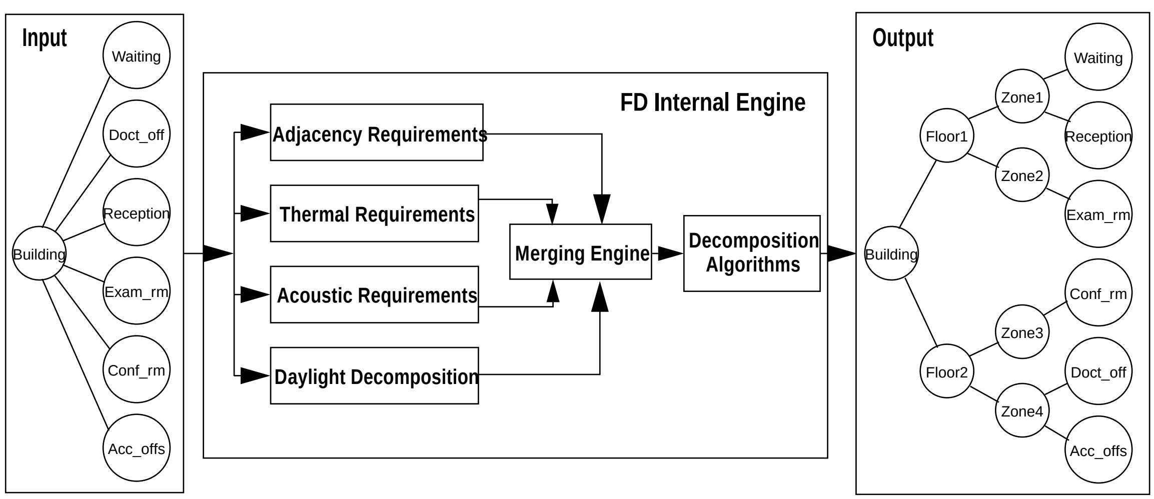

Figure 2.1 FD architecture

The merging engine is a mechanism for combining input design requirements. In order for the merging engine to work, the four aspects of design requirements must be represented in compatible ways. A detailed discussion of how to represent them can be found in the "representing requirements" section for each of the four design criteria.

The four design criteria are relatively independent. This modular structure makes it easy for the system to incorporate additional functional requirements, such as privacy and building structure. They are not currently considered in FD for the following reasons:

Privacy includes acoustic segregation and avoidance of visual disturbance. Acoustic segregation is exactly what decomposition according to acoustic requirements aims to accomplish. Avoidance of visual disturbance is related to building geometry (room shape, windows, doors, etc.) and therefore can only be considered in a more detailed design stage.

Structural requirements may play a role in grouping spaces. For example, grouping spaces with the same column span and structural height will reduce construction cost. This has not been included in order to keep the scope of the work manageable. This will be a future research topic.

2.3 System Features

The system features in FD are *a)* a graph-partitioning algorithm for stacking adapted from state-of-the-art VLSI graph-partitioning algorithms, and a clustering algorithm for blocking, *b)* multiple functional requirements considered in S&B, *c)* a user interface that facilitates user interaction, and *d)* a unique definition for blocking that enables exploration of design alternatives.

2.3.1 Automation of S&B Process

Internally, the four dimensions of design requirements, namely functional adjacencies, thermal, acoustic, and daylight requirements, are represented in a uniform graph format. The S&B algorithms can either partition each graph separately, or partition a unified graph of any combination of the four graphs. These two processes correspond to generation of S&B designs according to a single dimension of design requirements or according to multiple design requirements. The process of partitioning each of these graphs separately is called adjacency decomposition, thermal decomposition, acoustic decomposition, or daylight decomposition, depending on the criterion.

There are two types of constraints that the algorithms deal with, namely functional relations and individual characteristics. In the first case, relations between FUs are a focal point. FUs will be either grouped or separated according to the strength of their relations. There are two types of functional relations, namely functional adjacencies and acoustic segregation. For functional adjacencies, two algorithms, one for stacking and one for blocking, group FUs with close functional adjacencies so that closely related functional areas will be in the same floor or zone, thus enhancing communication between these FUs. For acoustic segregation, the same algorithms can be adapted to separate FUs which are acoustically incompatible so that physical segregation is achieved. This segregation is between FUs with a high level of activity noise and spaces that require a certain level of quiet.

For individual characteristics, such as thermal or daylight requirements, relations can be derived between FUs according to the similarities in these individual characteristics. For thermal requirements, such relations show how similar or different the FUs are in their thermal requirements. Those with similar thermal requirements will be grouped; and vice versa. For daylight requirements, strong relations are derived between FUs with the same orientation constraints. Thus, those with the same orientation constraints will be grouped; and vice versa.

2.3.2 Interactive Processes between User and System

FD allows for extensive interaction between the user and the system. This is based on the following assumptions about the roles of computers and designers in a design context [Klix and Wandke,1986]. Computers should "thoroughly store information" about a design, carry out algorithmic (i.e. programmable) processes as contributions to the design, output information to the designer about the actual state of design, and be able to retrieve input information from the designer for further processing. The designer, on the other hand, when trying to get maximal assistance from the computer, should "based on his idea of the aimed solution, knowledge and experience", provide input to the system whenever the system needs it, judge the quality of a design and modify the design, and decide how to proceed further with the design.

Interactions between designers and systems can be realized in different modes. Specifically, designers may control system processes, or intervene with system decisions. FD adopts both interaction modes in which the user can control the evolution of a design, and either modify the original problem statement or modify the solution generated by the system. Through the interactive approach, FD combines the computer's rapid and mass information storage, processing, and retrieval power, with the human designer's visual ability, judgment, and many capabilities that cannot easily be incorporated into a computer program. The system provides an automatic mechanism for generating S&B solutions. The user provides and modifies problem statements, evaluates the computer-generated solutions and modifies them. This is a cooperative process. The automatic mechanism provides assistance to the user while the user has full control of both the procedure and results.

Figure 2.2 illustrates such an approach. At the beginning the user provides a problem statement. Usually such a problem statement is incomplete. When the system finds out that there is missing information, it will prompt the user for the missing information. The user then has to input more information for a complete problem statement.

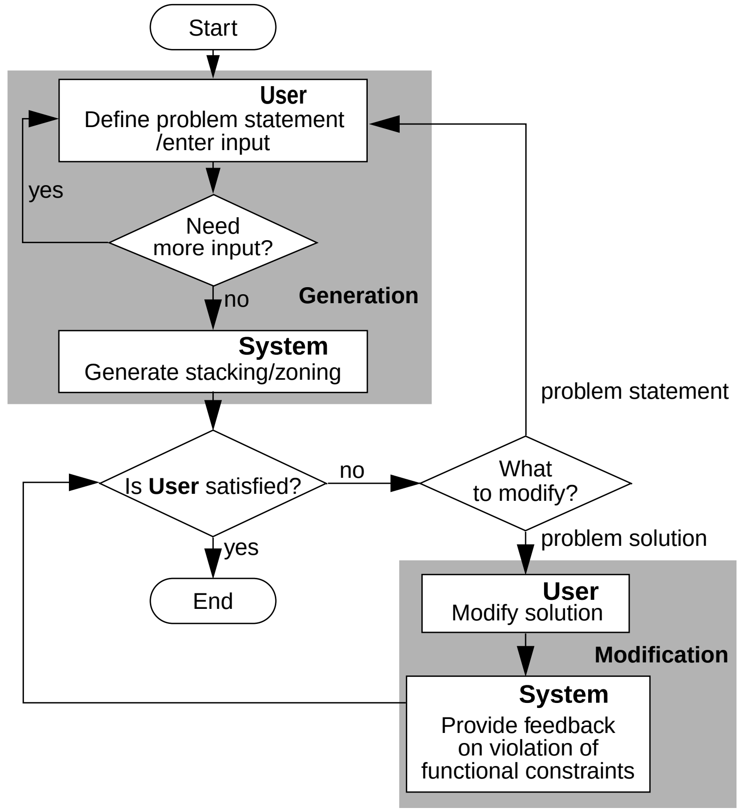

Figure 2.2 Interaction between user and system

When a complete problem statement is obtained, the system will **generate** a grouping (stacking/ blocking) solution. The user will then **evaluate** the solution. If the user is satisfied with the solution, the process is complete. Otherwise the user can either **modify** the problem statement or the solution to his/her satisfaction. If the user modifies the problem statement and requests the system to generate a new solution, the system will **regenerate** a solution based on the modified problem statement. If the user modifies the solution, the system will warn the user of any violations of functional constraints that may be committed. This iterative "generation-evaluation-modification-regeneration" design process continues until the user is completely satisfied with the result.

2.3.3 Zoning in FD: a Different Blocking Approach

In order to support exploration of blocking alternatives, FD has its own definition of blocking, which is called zoning. The need to define zoning arises from the facts that existing S&B programs have improper definitions of blocking. In OrgTree Stack and Block, each department is a zone. In WinSABA, blocking is a layout design within zones having pre-defined locations and shapes. In both programs, the use of zone concepts is inappropriate because:

Although all the FUs in one department may be located within the same zone, there are cases when the FUs belonging to a department are not located within the same zone for certain reasons. Reasons for this include limit of zone size, and consolidation of FUs belonging to different departments so that a single FU can be shared by different departments. Having fixed locations and dimensions of zones is not appropriate either, since this hampers exploration of zoning alternatives.

The proper zoning method, as defined in FD, is a process of grouping FUs on a floor into different architectural zones. The user can specify the constraints of zones, such as zone number and maximum zone size, but the physical locations and dimensions of zones are not defined. Physical locations and dimensions of FUs are not considered, either. Usually FUs with important adjacencies or with similar characteristics are grouped into the same zone. This approach to zoning is important because *a)* it generates different functional areas and thus provides a starting point for layout design, *b)* it supports exploration of design alternatives, and *c)* it groups FUs according to the strengths of their relations instead of simply according to their membership in the same department. From here on when the term “zoning” is used, it refers to this definition, which is intrinsic to FD.

3 Internal Engine

FD's internal engine, as shown in Figure 2.1, consists of three processes: **identifying and representing** the four areas of design requirements, combining these requirements through the **merging engine**, and decomposing a building according to these requirements through the **decomposition algorithms**. In addition to these three processes, a protocol analysis was conducted in order to **observe the design process**, which is important for developing the FD internal engine.

Initially, this section presents the protocol analysis and its findings. It then describes the process of identifying and representing adjacency, thermal, and acoustic requirements. Next, it introduces FD's merging engine. Lastly, it discusses the decomposition algorithms, including a stacking algorithm and a zoning algorithm. For daylight requirements, initial work has been done and can be referenced in Appendix E: Daylight Decomposition.

3.1 Observation of Design Process

In order to observe the process of stacking and zoning, a protocol analysis [Ericsson and Simon, 1993] was conducted. According to Ericsson and Simon, findings of a protocol analysis are valid despite a small number of participating subjects.

Four subjects, who were intended users of the system, participated. One of the subjects is a Ph.D. student in BPD (Building Performance and Diagnostics), one is a practicing architect, and two are mechanical engineers working in the building design industry. The goal of the experiment was to identify important criteria in stacking and zoning, individual operations and sequences of operations that the subjects performed. This section focuses on the sequences of operations that were identified. For important criteria in stacking and zoning, and individual operations, refer to Section 3.2.1.

The subjects were asked to design a two-story hospital building. Rooms/spaces belonging to four different departments were to be allocated within the building. These departments included Entrance/Reception, Urgicare, Pediatrics, and Accounting. The two floors and the rooms/spaces belonging to the departments were represented in the form of small-scale cardboard pieces of fixed dimensions. The subjects were asked to decide the locations for the spaces and put them on the floors. They were also asked to "think aloud". The process was video-taped. In order to keep the identity of the subjects confidential, the job title of each subject is used.

Each subject performed the operations in a different order; but there existed a pattern of a common sequence regardless of particular operations. This common sequence has been identified and is shown as a flow-chart in Figure 3.1.

The flow-chart is a high-level summary of the sequences of operations. It consists of a starting point, a generation process, an evaluation and modification process, and an end point. In this flow-chart, an oval box is a starting point or an end point of design. A rectangular box is an operation under either the "generation" or "modification" category as specified in Section 3.2.1.4. A diamond box can be either a control point or an evaluation point. For instance, the diamond box "Are there other unassigned departments?" is a control point. It makes sure that every department is assigned to a floor location before a next step of action takes place. But the diamond box "Satisfied?" is an evaluation point, at which a subject evaluated the design and made a judgment on it. If he/shes was satisfied a next step of action would take place. If he/she was unhappy with the design, he/she would then modify the design.

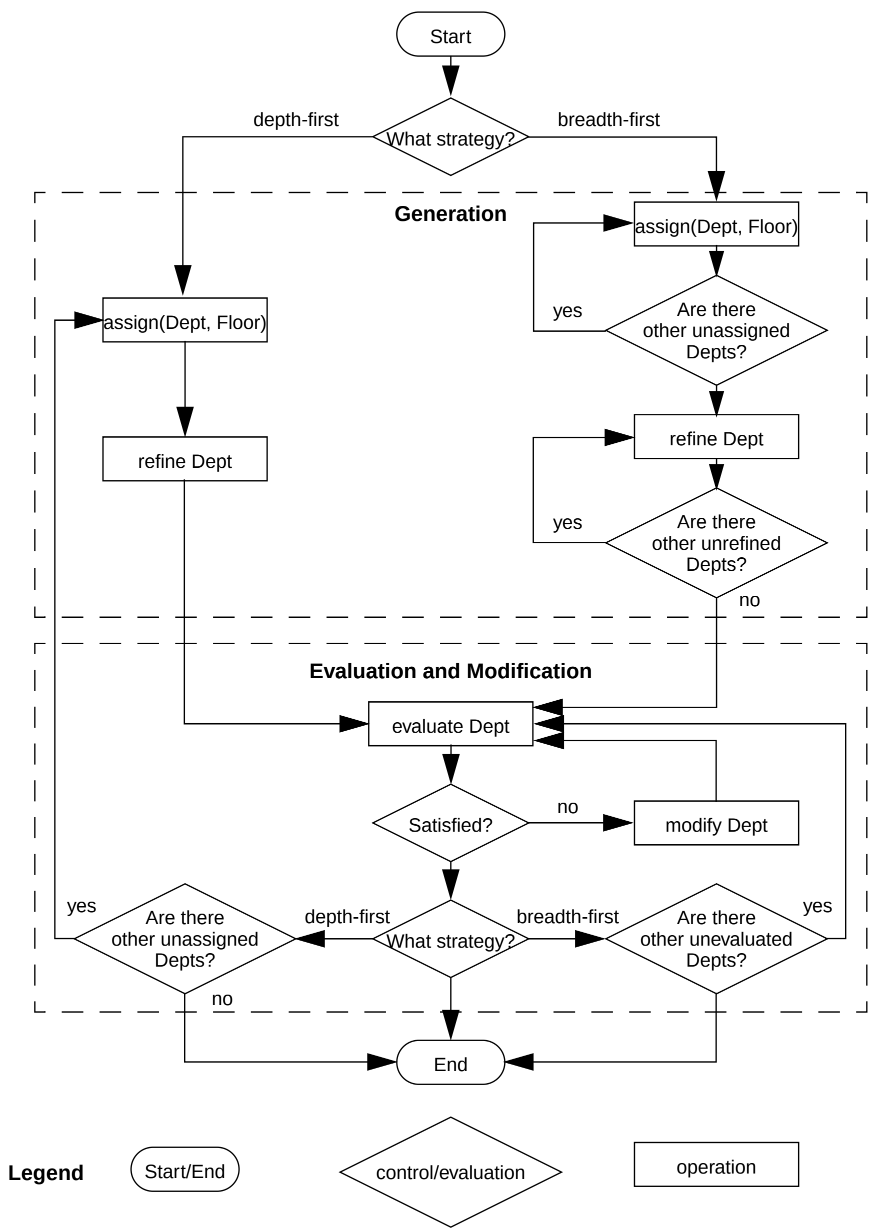

Figure 3.1 Sequence of operations

There exist two general sequences in the flow-chart, namely “depth-first” approach and “breadth-first” approach. A “depth-first” approach, in this experimental context, is completing each department down to the lowest level of detail before moving on to the next department. A “breadth-first” approach, in this experimental context, is processing all the departments at the most general level (i.e., sorting them into aggregate categories) before completing each department at a detailed level (i.e., creating finer grain zones and sub-zones).

The flow-chart illustrates a “generation-evaluation-modification” sequence. This sequence is a basis for the iterative and interactive design process as discussed in Section 2.3.2. FD adopts the “generation-evaluation-modification” sequence, and further expands this idea into an iterative process which is a “generation-evaluation-modification-regeneration” cycle.

In a “depth-first” process, the subjects consider the relations between the entity (department / group / FU) being placed with those that are already placed and those that are unassigned, although they may not explicitly place them at the same time. In this sense, this process resembles the “breadth-first” approach, to a degree. However, in a strict “depth-first” process, grouping for a single floor (assignment of FUs to a floor) and refinement of that floor into zones is completed before a next floor is considered. This process is not efficient. For instance, when a current floor is being designed, some FUs that are assigned to completed floors may need to be grouped to the current floor in order to optimize certain constraints. This move is caused by the nature of “depth-first” approach, which is to consider only one floor at a time without considering the relations among all the floors, so that a global adjustment of assignment of FUs to floors will be unavoidable. Furthermore, the global adjustment will invalidate existing groupings inside floors that are completed. When a significant global adjustment is made, a regeneration of groupings inside related completed floors will be necessary. Thus the “depth-first” process usually involves redundant designs.

On the other hand, a “breadth-first” approach is much more efficient. It takes into consideration the relations between all the entities. A higher-level design is completed before a sub-level design is carried out. This multi-step process guarantees that a large scale problem is reasonably divided before a detailed design for each portion takes place. There is no redundant design in this process. Therefore, FD adopts the “breadth-first” design approach, which is to assign all the FUs to floors before refining each floor into zones, i.e., stacking happens before zoning, which is a refinement of each floor into zones.

The sequences of operations are not only important in forming FD’s general design strategy and the iterative and interactive approach, but also have become the basis for the design of both user interface and algorithms.

As the subjects considered various requirements in designing the hospital, the observed design process as shown in Figure 3.1, is applicable to various design criteria.

3.2 Adjacency Requirements

Through the protocol analysis, not only the design process was obtained, important adjacency requirements were identified as well. These requirements include adjacency criteria and adjacency-related operations. Section 3.2.1 gives a detailed description of these adjacency requirements.

3.2.1 Identifying Requirements

3.2.1.1 Criteria used in stacking

Table 3.5 shows the criteria the subjects used in the stacking process. The first column in the table is a list of major departments that were to be allocated in the building. The second column is a list of floors to which the departments were to be allocated. Column three through column six list different criteria that the subjects used. Specifically, column three, four, five, and six show the criteria used by the Ph.D. Student, the Architect, the Mechanical Engineer1 and the Mechanical Engineer2, respectively. The last row in the table is a summary of the deciding factors that each subject considered in stacking according to the decisions he/she made in allocating the departments.

Table 3.5 Criteria used in stacking

Department	Floor #	**Ph.D. Student in related area**	**Architect**	**Mechanical Engineer1**	**Mechanical Engineer2**
Entrance/ Reception	1	close to the main entrance	close to the main entrance	close to the main entrance	close to the main entrance
Urgicare	1	close to the ambulance entrance	frequent use	same operation hours area as Entrance/ Reception, should be located on the same floor	close to the ambulance entrance; should be most accessible
Pediatrics	2	by default (left-over space)	2nd stage for patients, not for emergency	non 24 hour operation area	by default (left-over space)
Accounting	2	by default (left-over space)	remote relationship with outside	non 24 hour operation area	keep it the hardest place to get to
Deciding factors considered		proximity with existing entrances and area availability	proximity with existing entrances, frequencies of use, public/private nature of the department	proximity with existing entrances; occupancy type: 24 hour area vs. non 24 hour area	proximity with existing entrances; accessibility; area availability

When allocating the departments to floors, all four subjects assigned Entrance / Reception and Urgicare to the first floor, and assigned Pediatrics and Accounting on the second floor. They allocated Entrance / Reception on the first floor for the same reason of the close proximity relation between Entrance / Reception and the main entrance. They assigned Urgicare to the first floor for different reasons, such as it being close to the ambulance entrance, or it being frequently used, etc. They assigned Pediatrics and Urgicare to the second floor for different reasons as well, such as the available area on the second floor, functional nature of the department, etc.

For each subject, the deciding factors considered are different, depending on the background of the subject. For instance, the Mechanical Engineer1 considered occupancy type as a deciding factor in stacking. The Architect considered public / private nature of the department as a deciding factor. However they all considered proximity with existing entrances as a deciding factor. We therefore assume that proximity / adjacency relations are an important factor in stacking.

Note that this protocol analysis only demonstrates the method followed to obtain useful criteria in stacking and zoning. It does not intend to arrive at universal criteria to follow in all kinds of design contexts. In order to achieve this, a larger scale of study is necessary.

3.2.1.2 Criteria used for locating major departments

Table 3.6 shows the criteria the subjects used in allocating the four major departments. The first column is a list of the departments, including Entrance / Reception, Urgicare, Pediatrics and Accounting. Column three through column six record the location each subject chose for each department and the reasons for such an allocation. The last row of the table is a summary of the deciding factors each subject considered in allocating the departments.

As shown in the table, each subject has different reasons for allocating a department to a particular location. For instance, the Ph.D. Student put the Urgicare department in the north location on the first floor because this department should be close to the ambulance entrance. The Architect put Urgicare in the south location on the first floor because it should have an easy access from either of the two existing entrances, i.e., the main entrance and the ambulance entrance.

Table 3.6 Criteria used for locating major departments

Department		Ph.D. Student in related area	Architect	Mechanical Engineer1	Mechanical Engineer2
Entrance/ Reception	location	south	south	south	south
	reasons	close to the main entrance	close to the main entrance	close to the main entrance	close to the main entrance
Urgicare	location	north	south (major part)	north	north
	reasons	close to the ambulance entrance	easy to access from either of the two entrances	by default (left-over space)	close to the ambulance entrance
Pediatrics	location	south	south	south	south
	reasons	public nature of this dept.; close to the elevator	public nature of this dept.; close to the elevator	public nature of this dept.; close to the elevator	public nature of this dept.; close to the elevator
Accounting	location	north	north	north	north
	reasons	by default (leftover space)	private nature of this dept.	private nature of this dept.; away from the elevator	private nature of this dept.; keep it the hardest place to get to
Deciding factors considered		accessibility; proximity with existing entrances; area availability	accessibility; proximity with existing entrances	accessibility; proximity with existing entrances; area availability	accessibility; proximity with existing entrances

As shown in the last row of the table, while each subject had a slightly different set of criteria, they all considered accessibility and proximity with the existing entrances to be crucial factors. Since physical elements affecting accessibility such as doors and paths are not considered in stacking and zoning, accessibility can be considered as being equivalent to proximity or adjacency. Therefore proximity / adjacency appears to be a crucial factor in locating major departments.

3.2.1.3 Criteria used in grouping FUs

Table 3.7 shows the criteria used by the subjects at different levels of groupings, in allocating smaller groups of FUs, in considering vertical relations, and in choosing HVAC (heating, ventilation, and air conditioning) systems. Column one is a list of different levels of grouping tasks performed by the subjects, including

Table 3.7 Criteria used in grouping

	Ph.D. Student in related area	Architect	Mechanical Engineer1	Mechanical Engineer2
Major groupings on each floor	take the existing organization as is by department	take the existing organization as is by department	take the existing organization as is by department; special functional relationships[a]	take the existing organization as is by department
Subgroupings in each major group	similar occupancy; same daylight level and fresh air control	spatial proximity; plumbing functions	plumbing functions; privacy	spatial proximity; same mechanical and electrical requirements
locations of subgroups	priorities of orientations of daylight	put new plumbing functions near existing ones	put new plumbing functions near existing ones	put new plumbing functions near existing ones or elevator shaft; daylight
Vertical relationships considered	staircase; vertical zoning if applicable	no	stack plumbing	stack plumbing and exhaust
HVAC system arrived	one separate system for each floor	one system for the whole building (not sure)	one separate system for each floor	minimum of one separate system for each department
Deciding factors considered	existing organization; occupancy; daylight; fresh air	existing organization; spatial proximity; plumbing	existing organization; spatial relationships; privacy; plumbing	existing organization; spatial proximity; plumbing, exhaust and elevator shaft; daylight

a. Initially the subject wanted to group some accounting offices on the first floor with entrance/reception because he thought that accounting offices would be needed for billing in this area. Later on he realized that there was already a registration/billing area in entrance/reception, thus he gave up the idea of inter-departmental grouping.

major groupings on each floor, subgroupings within major groups, locating subgroups, vertical relations considered, and decision on a HVAC system. Column two through column five present the criteria used by each subject in performing a particular task as listed in column one. The last row in the table is a summary of the deciding factors that each subject used for different levels of groupings.

In general, all the four subjects shared the same criterion in major groupings, i.e., formulating major groupings by department. But for the remaining tasks, the criteria they used were more diverse. For instance, in subgroupings, the Ph.D. Student considered similar occupancy, same daylight level and fresh air control; whereas the Architect considered spatial proximity and plumbing functions.

In summary, the deciding factors the subjects considered include existing organization, spatial proximity, privacy, and mechanical system requirements such as occupancy, daylight, fresh air, plumbing, and exhaust. This shows that a broader spectrum of criteria should be considered in grouping or zoning.

3.2.1.4 Single operations

The single operations the subjects performed are identified through the protocol analysis. They all fall into one of the three general types of operations, namely generation, evaluation, and modification (Table 3.8).

Table 3.8 Single operations

Generation	Sorting	sort (Dept1, Dept2, ..., Deptn)
		getFUs (plumbing)
		getFUs (exhaust)
	Grouping	group (FU1, FU2, ..., FUn)-->Group
		group (FU, Group1)-->Group2
	Allocation	put (FU/Group, Floor)
		assign (Dept, Floor)
		vertical_put (Group2, Floor2, Group1, Floor1)
		buffer_put (FU1/Group1, FU2/Group2, FU3/Group3)
		far_put (FU1/Group1/Dept1, FU2/Group2/Dept2)
		near_put (FU1/Group1, FU2/Group2)
		center_close_put (FU1/Group1, FU2/Group2)
		chain (FU1, FU2, Group/Dept)
	Employment of user's knowl-edge	make_constituent (FU1, FU2)
		make (FU, Floor1, Floor2, ..., Floorn) //vertical FU
		make (FU1, FU/Group/Dept)
		make (separation/connection, Dept1, Dept2)
		make (FU, exterior-wall)
		convert (FU1, newFU2)
		convert (FU1, existingFU2)
		remove (FU/Group)
	Controlling	goto (Dept/Floor)
Evaluation	Evaluation	evaluate (criteria, Group/Dept/Floor)
Modification	Moving	move (FU/Group/Dept)
	Exchanging locations	switch (FU1, FU2)
		switch (Group1, Group2)
		switch (Dept1, Dept2)

Under the generation type, the operations are divided into the following categories: *a)* sorting, *b)* grouping, *c)* allocation of FU / group / department onto floor, *d)* employment of user's knowledge-base, and e) controlling.

Sorting is an action of selecting FUs according to their function(s). Sorting operations include selecting FUs according to their affiliated department, selecting FUs that have plumbing requirements, and selecting FUs that need exhaust. Grouping is an action of either joining multiple FUs into a group, or joining a single FU with a group of FUs. Allocation is to place a FU or a group or a department in a particular location in the building. An allocation usually involves consideration of the relations between the entity to be placed with FUs that are already placed. Thus, besides simple "putting" and "assigning" operations, there are "vertical", "buffer", "far", "near", "center-close", and "chain" operations that allocate an entity in relative spatial relation to FUs that are already allocated. By employing the knowledge base of a designer, he / she can make new FUs out of existing ones, convert existing FUs into new FUs with different functions, or rename an FU or a group of FUs in the building. Controlling is to make a judgment on how to proceed with a design at a point where there are multiple options.

Evaluation is to examine an existing design against certain criteria such as functional adjacencies and daylight, and to consider if any changes in the design are necessary. Modification is to adjust the design in order to improve upon certain criteria. Operations under this category include moving a spatial entity to a new location, and exchanging the locations of two spatial entities.

These operations are the basis of the operations that FD should provide to the user in adjacency decomposition. In other words, FD should allow the user to perform the important operations identified in the protocol analysis. For example, it should allow the user to group FUs, and assign an FU to a floor.

3.2.2 Representing Requirements

There are two types of adjacency requirements: strict adjacency and distance relations. Strict adjacency refers to geometric relations requiring FUs to be located right next to each other. Distance relations refer to geometric relations requiring FUs to be located within a certain distance. In FD, adjacency requirements are represented as a relation between a pair of FUs.

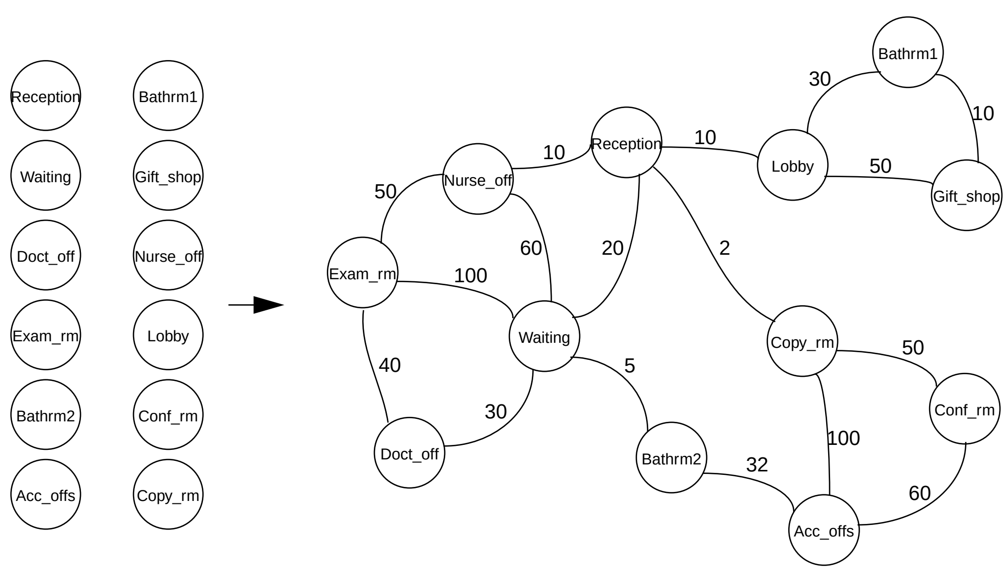

Figure 3.2 Representing adjacency relations in graph

The input of adjacency requirements is a collection of FUs with size constraints for each FU and adjacency constraints between any pair of FUs. The example in Figure 3.2 illustrates the process of formulating adjacency constraints in a graph representation. The nodes represent FUs. The edges represent adjacency relations between two connected FUs. The numbers assigned to the edges show the strengths or weights of the adjacency relations - the larger the weights, the stronger the relations.

The FUs with strong adjacency relations should be grouped into the same floor and / or zone, while the FUs with weak adjacency relations could be separated into different floors and / or zones. More specifically, the goal of decomposing a building according to adjacencies is to minimize the total weight between different floors or zones (sum of weights of edges connecting FUs within different floors or zones), thus increasing the cohesiveness of FUs within the same floor / zone. An objective function for stacking is as follows:

$$bjective = \min\left(\sum_{m=1}^{Nf-1}\sum_{n=m+1}^{Nf}\sum_{FU_i \subset Floor_m,\ FU_j \subset Floor_n,\ m \neq n} Adjacency_{FU_i, FU_j}\right)$$

In the above formula, $Adjacency_{FU_i, FU_j}$ is the strength of the adjacency relation between FU_i and FU_j, and Nf is the number of floors in a building.

Likewise, an objective function of zoning is as follows:

$$jective = \min\left(\sum_{m=1}^{Nz-1}\sum_{n=m+1}^{Nz}\sum_{FU_i \subset Zone_m,\ FU_j \subset Zone_n,\ m \neq n} Adjacency_{FU_i, FU_j}\right)$$

In the above formula, $Adjacency_{FU_i, FU_j}$ is the strength of the adjacency relation between FU_i and FU_j, and Nz is the number of zones on a floor.

3.3 Thermal Requirements

In principle, thermal requirements include energy consumption, environmental impact, and thermal comfort. This work will focus on energy consumption only. Therefore an energy simulation was conducted to identify which thermal parameters are important to stacking and zoning, i.e., grouping by which thermal parameters can reduce energy consumption.

3.3.1 Identifying Requirements

Parametric energy simulation studies were conducted using the building performance modeling software SEMPER in order to identify useful thermal parameters for grouping FUs. A description of SEMPER is in Section 3.3.1.1. Four parameters were tested: space temperature, load schedule, internal load, and air exchange rate.

Toward this end, for each of the four parameters, a single-floor building with a grouped condition was tested against a single-floor building with a mixed condition in order to compare their annual energy use (Figure 3.3). In the grouped condition, FUs within each quadrant shared the same thermal value, whereas in the mixed condition, FUs within each quadrant had different thermal values as indicated by A, B, C, and D.

The idea was to examine if grouping FUs with similar thermal characteristics (e.g., similar desirable air temperatures or similar required minimum air exchange rates) will result in significant differences in the predicted resulting performance, in this case, the annual energy use. In order to compare energy consumptions for the two grouping conditions, the annual total loads of the two grouping conditions were to be obtained in the experiment.

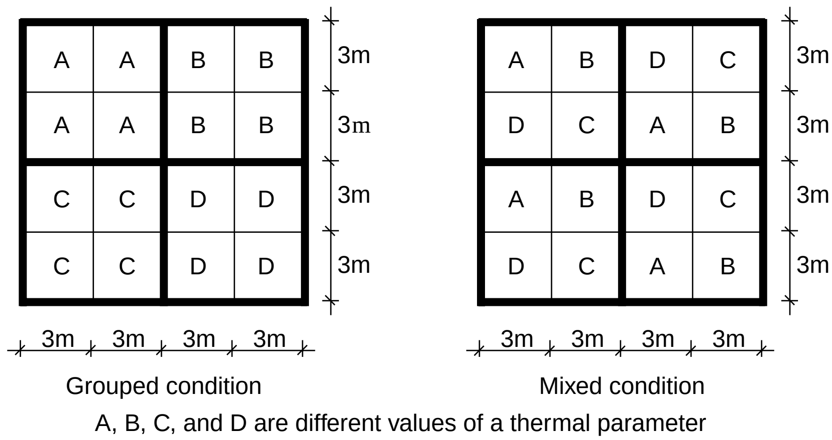

Figure 3.3 Design of thermal experiment

Thermal parameters considered in this study include occupancy schedule, internal heat load (including heat load from equipment, lighting and people), temperature requirements, and air exchange rate requirements. There are other parameters such as external heat load, mean radiant temperature, occupants' activity level, and clothing level. They are excluded from thermal decomposition, because external heat load is related to the locations and orientations of FUs in a building. These cannot be decided in FD since it deals with a phase of design prior to a phase that handles geometric design when FUs are allocated to fixed locations and are given fixed dimensions. The remaining are thermal comfort parameters, which will not be considered in this research.

Two principles were followed in this simulation study. The first principle was to separately test each thermal parameter; i.e., when testing a parameter, the remaining thermal parameters were kept constant so that they would not affect the results for the tested parameter. The second principle was to make all the other factors irrelevant. This was achieved by making each

FU the same size and uniform square shape, designing each group so that each had the same number of FUs, and making locations / orientations irrelevant by not considering solar access.

The analysis of the results of this specific set of simulations, i.e., eight annual energy consumption levels, showed that only grouping based on the temperature values significantly affects the predicted energy use. Temperature may be thus regarded as a useful thermal parameter for grouping FUs.

The following gives a detailed description of the process of this thermal experiment.

3.3.1.1 Tool: SEMPER

SEMPER (**S**imulation **E**nvironment for **M**odeling **Per**formance) [Mahdavi, 1996][Mahdavi et al., 1998] is an active multi-aspect computational tool integrating building performance simulation (e.g., energy, lighting, and acoustics) into computational design systems. It has been developed by the Building Performance and Diagnostics (BPD) research group under the guidance of project director Ardeshir Mahdavi in the School of Architecture at CMU.

In the thermal study, for each tested parameter, two single-floor buildings as shown in Figure 3.3 were input into SEMPER's energy simulation module (NODEM) separately. For each building, the 16 FUs with their dimensions and thermal requirements were entered. Then four HVAC zones were specified with each zone being a quadrant in the floor plan and containing four FUs (Figure 3.4). For instance, with the subscript numbers representing the position of an FU, Zone 1 contained FU1-1, FU1-2, FU2-1, and FU2-2 [Ardeshir et al., 1998].

For each zone, only the set-point[1] temperature and set-point RH (relative humidity) needed to be specified. When testing temperature on the plan in a grouped condition, each zone's set-point temperature was the same as the set-point temperature of the four FUs within that zone. When testing temperature on the plan in a mixed condition, each zone's set-point temperature was 20-22^{o}C, the most restrictive temperature range of the four FUs' within that zone. This guaranteed that the required temperature ranges of all the four FUs', although different from each other, could be met through the temperature setup of the zone they were in. When testing other parameters than temperature, 20-22^{o}C set-point temperature was used for each zone in either grouped condition or mixed condition. In all test cases, a RH range of 40-60% was used for each zone.

When the solar gain option was turned off, a 12 months' energy consumption scenario was generated for each test case. The total of the 12 months' energy consumption was the annual energy consumption used in the result analysis.

---

1. In an HVAC control system, set-point is the controller setting and is the desired value of the controlled variable, e.g., set-point temperature and set-point RH [Haines, 1983].

3.3.1.2 Overall design

A square plan of 16 FUs was used. Each FU was of the size $3m \times 3m$.

The project was designed so that for each tested parameter there were four different values (or ranges) and each value (or range) was shared by exactly four FUs. Default values (or ranges) were used for the other thermal parameters.

Each zone of four FUs were in one quadrant square as shown in Figure 3.3 and Figure 3.4. As for the relative locations of the zones, the closer they were, the smaller the difference in values (or ranges). As it was closer to Zone 2 and Zone 3 than to Zone 4, Zone 1 should have more similarity with Zone 2 and Zone 3 than with Zone 4 in terms of the tested parameter.

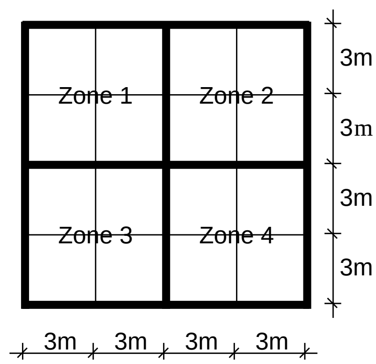

Figure 3.4 HVAC zones in tested building

A thermal value (or range) is expressed by a capital letter with the subscript representing a thermal property. For instance, A_t (20-22° C), B_t (20-26° C), C_t (16-22° C), and D_t (14-28° C) represent different temperature ranges, where the subscript "t" identifies the temperature property. Likewise, A_{RH}(40-60%) is a range of RH and B_i (50W/m^2) is an internal heat value, etc. There is no relation between values (or ranges) as expressed by the same capital letter when they represent different thermal properties. For instance, there is no relation between A_t, A_{RH}, A_i, A_l, and A_a. Similarly, there is no relation between the "B"s, the "C"s, or the "D"s. The default values (or ranges) are A_t (20-22^{o}C) for temperature, A_{RH} (40-60%) for RH, B_i (50W/m^2) for internal heat load, A_l (8:00am-5:00pm) for load schedules, and A_a (0.5h^{-1}) for air exchange rate.

For each parameter, four different values (or ranges) were designed. In each table showing the four values (or ranges), the one value or range in bold and with a larger size represents a default value or range.

3.3.1.3 Temperature

Four temperature ranges were assumed as shown in Figure 3.5. They were: A_t (20-22° C), B_t (20-26° C), C_t (16-22° C), and D_t (14-28° C). A_t was used as default when a thermal parameter other than temperature was tested.

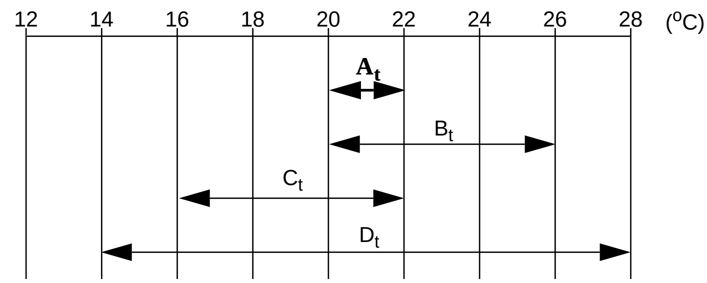

Figure 3.5 Temperature design

3.3.1.4 Relative humidity (RH)

Four RH ranges were assumed as shown in Figure 3.6. They were A_{RH} (40-60%), B_{RH} (30-50%), C_{RH} (50-70%), and D_{RH} (30-70%). A_{RH} was used as default when a thermal parameter other than RH was tested.

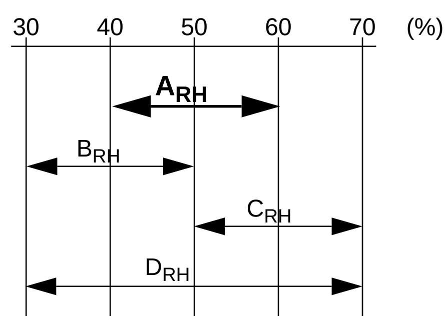

Figure 3.6 RH design

3.3.1.5 Internal heat load

The internal heat load values represented a total of the heat loads generated by equipment, lighting, and people. To illustrate examples of heat loads for a typical office building, lighting wattage (heat generation) is $16 W/m^2$, whereas heat generated by computers is on average between $10 W/m^2$ and $20 W/m^2$, depending on actual computer usage.

Four space internal heat loads were assumed as shown in Table 3.9. They were A_i (30W/m^2), B_i (50W/m^2), C_i (70W/m^2), and D_i (90W/m^2). For the four designed internal heat loads, the heat loads from lighting and people were constant (20W/m^2 and 10W/m^2 respectively); only the heat loads generated by equipment varied, which resulted in different total internal heat loads.

B_i was used as default when a thermal parameter other than internal heat load was tested. For A_i, B_i, C_i and D_i, the heat load generated by equipment were 0W/m^2, 20W/m^2, 40W/m^2, and 60W/m^2, respectively. 20W/m^2 was the default heat load generated by equipment.

Table 3.9 Internal heat load design

type	equip	light	people	sum
A_i	0	20	10	30
B_i	**20**	**20**	**10**	**50**
C_i	40	20	10	70
D_i	60	20	10	90

3.3.1.6 Load schedule

Four load schedules were assumed as shown in Figure 3.7. They were A_l (8am-5pm), B_l (6am-3pm), C_l (11am-8pm), and D_l (5am-8pm). A_l was a typical schedule therefore it was used as default when a thermal parameter other than load schedule was tested.

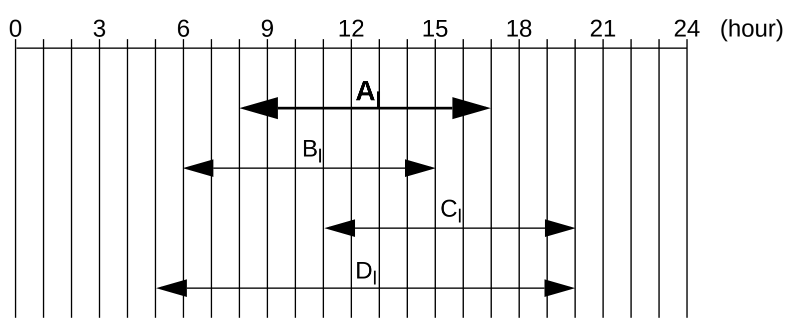

Figure 3.7 Load (occupancy) schedule design

3.3.1.7 Ventilation (air exchange rate)

Four air exchange rates were assumed as shown in Table 3.10. When the subscript "a" was used to identify the air exchange rate property, these four air exchange rates were expressed by symbols A_a ($0.2h^{-1}$), B_a ($0.5h^{-1}$), C_a ($1h^{-1}$), and D_a ($2h^{-1}$). A_a was the default value when a thermal parameter other than air exchange rate was tested.

Table 3.10 Air exchange rate design

A_a	0.2
B_a	**0.5**
C_a	1
D_a	2

3.3.1.8 Data entered and results

This section presents all the input data and simulation results (annual energy loads) in a tabular format. Since four thermal parameters were tested, and for each tested parameter, two grouping conditions, i.e., grouped condition and mixed condition were tested, there are eight tables (Tables 3.11 through 3.18) corresponding to these eight test cases.

In each table, column one lists four zones in a building with each zone having four FUs as listed in column two. Column three through column seven are required low temperature, required high temperature, internal heat load, load schedule (occupancy schedule), and air exchange rate. The last column is the annual energy load obtained through simulation for a corresponding test case.

As shown in Figure 3.4, in each layout, Zone 1 is the 4-FU quadrant in the upper-left position of the building plan. Zone 2 is the 4-FU quadrant in the upper-right position of the building plan. Zone 3 and 4 are the 4-FU quadrants in the lower-left and lower-right positions of the plan, respectively. An FU id is an identity of the FU's location on a building plan. For instance, FU1-2 is the FU in the first row and second column position. FU2-3 is the FU in the second row and third column position. By defining zone and FU id, each table portrays the exact locations of zones and FUs on a corresponding layout.

3.3.1.9 Simulation results of temperature changes

In this test case, all of the FUs share the same set of default values for internal heat load, load schedule (occupancy schedule), and air exchange rate. In a grouped condition, as shown in Table 3.11, FUs within the same zone share the same values of low and high temperatures. FUs within different zones have different low and high temperature values.

Table 3.11 Grouped condition by temperature

Zone	FU id	low temperature (^{o}C)	high temperature (^{o}C)	internal load (W/m^2)	load schedule (hour)	air exchange rate (h^{-1})	annual load (MWh)
1	FU1-1	20	22	50	8-17	0.5	34.2
	FU1-2	20	22	50	8-17	0.5	
	FU2-1	20	22	50	8-17	0.5	
	FU2-2	20	22	50	8-17	0.5	
2	FU1-3	20	26	50	8-17	0.5	
	FU1-4	20	26	50	8-17	0.5	
	FU2-3	20	26	50	8-17	0.5	
	FU2-4	20	26	50	8-17	0.5	
3	FU3-1	16	22	50	8-17	0.5	
	FU3-2	16	22	50	8-17	0.5	
	FU4-1	16	22	50	8-17	0.5	
	FU4-2	16	22	50	8-17	0.5	
4	FU3-3	14	28	50	8-17	0.5	
	FU3-4	14	28	50	8-17	0.5	
	FU4-3	14	28	50	8-17	0.5	
	FU4-4	14	28	50	8-17	0.5	

In contrast to this design, in a mixed condition, FUs within the same zone have different low and high temperatures as shown in Table 3.12.

Table 3.12 Mixed condition by temperature

Zone	FU id	low temperature (^{o}C)	high temperature (^{o}C)	internal load (W/m^2)	load schedule (hour)	air exchange rate (h^{-1})	annual load (MWh)
1	FU1-1	20	22	50	8-17	0.5	44.2
	FU1-2	20	26	50	8-17	0.5	
	FU2-1	14	28	50	8-17	0.5	
	FU2-2	16	22	50	8-17	0.5	
2	FU1-3	14	28	50	8-17	0.5	
	FU1-4	16	22	50	8-17	0.5	
	FU2-3	20	22	50	8-17	0.5	
	FU2-4	20	26	50	8-17	0.5	
3	FU3-1	20	22	50	8-17	0.5	
	FU3-2	20	26	50	8-17	0.5	
	FU4-1	14	28	50	8-17	0.5	
	FU4-2	16	22	50	8-17	0.5	
4	FU3-3	14	28	50	8-17	0.5	
	FU3-4	16	22	50	8-17	0.5	
	FU4-3	20	22	50	8-17	0.5	
	FU4-4	20	26	50	8-17	0.5	

The annual energy load for the grouped condition is 34.2 MWh, whereas the annual energy load for the mixed condition is 44.2 MWh. This indicates that grouping FUs by similar required temperatures can consume less energy annually. Therefore, temperature may be considered as a factor in grouping FUs.

Note that these simulation studies are of explorative nature. They do not intend to arrive at a final conclusion as to what thermal criteria are relevant to stacking and zoning applications. To arrive at such a conclusion, a much larger set of simulations must be preformed based on a multitude of building types, climates, etc.

3.3.1.10 Simulation results of internal load changes

In this test case, all of the FUs share the same set of default values for low and high temperatures, load schedule (occupancy schedule), and air exchange rate. In a grouped condition, as shown in Table 3.13, FUs within the same zone share the same internal load. FUs within different zones have different internal load values.

Table 3.13 Grouped condition by internal loads

Zone	FU id	low temperature (^oC)	high temperature (^oC)	internal load (W/m^2)	load schedule (hour)	air exchange rate (h^{-1})	annual load (MWh)
1	FU1-1	20	22	30	8-17	0.5	55.0
	FU1-2	20	22	30	8-17	0.5	
	FU2-1	20	22	30	8-17	0.5	
	FU2-2	20	22	30	8-17	0.5	
2	FU1-3	20	22	50	8-17	0.5	
	FU1-4	20	22	50	8-17	0.5	
	FU2-3	20	22	50	8-17	0.5	
	FU2-4	20	22	50	8-17	0.5	
3	FU3-1	20	22	70	8-17	0.5	
	FU3-2	20	22	70	8-17	0.5	
	FU4-1	20	22	70	8-17	0.5	
	FU4-2	20	22	70	8-17	0.5	
4	FU3-3	20	22	90	8-17	0.5	
	FU3-4	20	22	90	8-17	0.5	
	FU4-3	20	22	90	8-17	0.5	
	FU4-4	20	22	90	8-17	0.5	

In contrast to this design, in a mixed condition, FUs within the same zone have different internal loads as shown in Table 3.14.

Table 3.14 Mixed condition by internal loads

Zone	FU id	low temperature (^oC)	high temperature (^oC)	internal load (W/m^2)	load schedule (hour)	air exchange rate (h^{-1})	annual load (MWh)
1	FU1-1	20	22	30	8-17	0.5	54.4
	FU1-2	20	22	50	8-17	0.5	
	FU2-1	20	22	90	8-17	0.5	
	FU2-2	20	22	70	8-17	0.5	
2	FU1-3	20	22	90	8-17	0.5	
	FU1-4	20	22	70	8-17	0.5	
	FU2-3	20	22	30	8-17	0.5	
	FU2-4	20	22	50	8-17	0.5	
3	FU3-1	20	22	30	8-17	0.5	
	FU3-2	20	22	50	8-17	0.5	
	FU4-1	20	22	90	8-17	0.5	
	FU4-2	20	22	70	8-17	0.5	
4	FU3-3	20	22	90	8-17	0.5	
	FU3-4	20	22	70	8-17	0.5	
	FU4-3	20	22	30	8-17	0.5	
	FU4-4	20	22	50	8-17	0.5	

The annual energy load (for internal load) for the grouped condition is 55.0 MWh, whereas the annual energy load for the mixed condition is 54.4 MWh. This suggests that grouping FUs by similar required internal loads does not make a significant difference in the annual energy consumption as compared to the mixed condition. Therefore, internal load may not be considered as a factor in grouping FUs.

This is not to say that internal loads are not important design criteria. It simply suggests that they did not appear to be sensitive in the zoning schema changes as indicated in the simulations.

3.3.1.11 Simulation results of load schedule changes

In this test case, all of the FUs share the same set of default values for low and high temperatures, internal loads, and air exchange rate. In a grouped condition, as shown in Table 3.15, FUs within the same zone share the same load schedule. FUs within different zones have different load schedules.

Table 3.15 Grouped condition by schedule

Zone	FU id	low temperature (^oC)	high temperature (^oC)	internal load (W/m^2)	load schedule (hour)	air exchange rate (h^{-1})	annual load (MWh)
1	FU1-1	20	22	50	8-17	0.5	47.9
	FU1-2	20	22	50	8-17	0.5	
	FU2-1	20	22	50	8-17	0.5	
	FU2-2	20	22	50	8-17	0.5	
2	FU1-3	20	22	50	6-15	0.5	
	FU1-4	20	22	50	6-15	0.5	
	FU2-3	20	22	50	6-15	0.5	
	FU2-4	20	22	50	6-15	0.5	
3	FU3-1	20	22	50	11-20	0.5	
	FU3-2	20	22	50	11-20	0.5	
	FU4-1	20	22	50	11-20	0.5	
	FU4-2	20	22	50	11-20	0.5	
4	FU3-3	20	22	50	6-20	0.5	
	FU3-4	20	22	50	6-20	0.5	
	FU4-3	20	22	50	6-20	0.5	
	FU4-4	20	22	50	6-20	0.5	

In contrast to this design, in a mixed condition, FUs within the same zone have different load schedules as shown in Table 3.16.

Table 3.16 Mixed condition by schedule

Zone	FU id	low temperature (^oC)	high temperature (^oC)	internal load (W/m^2)	load schedule (hour)	air exchange rate (h^{-1})	annual load (MWh)
1	FU1-1	20	22	50	8-17	0.5	42.4
	FU1-2	20	22	50	6-15	0.5	
	FU2-1	20	22	50	6-20	0.5	
	FU2-2	20	22	50	11-20	0.5	
2	FU1-3	20	22	50	6-20	0.5	
	FU1-4	20	22	50	11-20	0.5	
	FU2-3	20	22	50	8-17	0.5	
	FU2-4	20	22	50	6-15	0.5	
3	FU3-1	20	22	50	8-17	0.5	
	FU3-2	20	22	50	6-15	0.5	
	FU4-1	20	22	50	6-20	0.5	
	FU4-2	20	22	50	11-20	0.5	
4	FU3-3	20	22	50	6-20	0.5	
	FU3-4	20	22	50	11-20	0.5	
	FU4-3	20	22	50	8-17	0.5	
	FU4-4	20	22	50	6-15	0.5	

The annual energy load for the grouped condition is 47.9 MWh, whereas the annual energy load for the mixed condition is 42.4 MWh. This indicates that grouping FUs by similar required load schedules may consume more energy annually as compared to the mixed condition. Therefore, load schedule may not be used as a criterion in grouping FUs.

3.3.1.12 Simulation results of air exchange rate changes

In this test case, all of the FUs share the same set of default values for low and high temperatures, internal loads, and load schedules. In a grouped condition, as shown in Table 3.17, FUs within the same zone share the same air exchange rate. FUs within different zones have different air exchange rates.

In contrast to this design, in a mixed condition, FUs within the same zone have different air exchange rates as shown in Table 3.18.

Table 3.17 Grouped condition by air exchange rate

Zone	FU id	low temperature (^oC)	high temperature (^oC)	internal load (W/m^2)	load schedule (hour)	air exchange rate (h^{-1})	annual load (MWh)
1	FU1-1	20	22	50	8-17	0.2	48.9
	FU1-2	20	22	50	8-17	0.2	
	FU2-1	20	22	50	8-17	0.2	
	FU2-2	20	22	50	8-17	0.2	
2	FU1-3	20	22	50	8-17	1	
	FU1-4	20	22	50	8-17	1	
	FU2-3	20	22	50	8-17	1	
	FU2-4	20	22	50	8-17	1	
3	FU3-1	20	22	50	8-17	0.5	
	FU3-2	20	22	50	8-17	0.5	
	FU4-1	20	22	50	8-17	0.5	
	FU4-2	20	22	50	8-17	0.5	
4	FU3-3	20	22	50	8-17	2	
	FU3-4	20	22	50	8-17	2	
	FU4-3	20	22	50	8-17	2	
	FU4-4	20	22	50	8-17	2	

Table 3.18 Mixed condition by air exchange rate

Zone	FU id	low temperature (^oC)	high temperature (^oC)	internal load (W/m^2)	load schedule (hour)	air exchange rate (h^{-1})	annual load (MWh)
1	FU1-1	20	22	50	8-17	0.2	47.3
	FU1-2	20	22	50	8-17	1	
	FU2-1	20	22	50	8-17	2	
	FU2-2	20	22	50	8-17	0.5	
2	FU1-3	20	22	50	8-17	2	
	FU1-4	20	22	50	8-17	0.5	
	FU2-3	20	22	50	8-17	0.2	
	FU2-4	20	22	50	8-17	1	
3	FU3-1	20	22	50	8-17	0.2	
	FU3-2	20	22	50	8-17	1	
	FU4-1	20	22	50	8-17	2	
	FU4-2	20	22	50	8-17	0.5	
4	FU3-3	20	22	50	8-17	2	
	FU3-4	20	22	50	8-17	0.5	
	FU4-3	20	22	50	8-17	0.2	
	FU4-4	20	22	50	8-17	1	

The annual energy load (for air exchange rate) for the grouped condition is 48.9 MWh, whereas the annual energy load for the mixed condition is 47.3 MWh. This suggests that grouping FUs by similar required air exchange rates does not consume a significantly different amount of energy as compared to the mixed condition. Therefore, air exchange rate may not be used as a criterion in grouping FUs.

3.3.1.13 Findings of energy simulation

Table 3.19 shows the eight annual energy loads obtained from the thermal experiment. According to Table 3.19, grouping FUs with same temperature requirements can save energy annually by 23%. But grouping FUs with same internal loads, load schedules, and air exchange rates do not save energy. Intuitively this shows that only grouping based on the temperature values significantly affects the predicted energy use. Therefore temperature is used as an example to illustrate a formalization of the thermal decomposition process. Note the illustrative character of these simulation studies. To arrive at a universal judgment as to the relevant thermal criteria for the purposes of decomposition, a much larger set of simulation studies may be necessary.

Table 3.19 Data analysis

Tested parameter	Energy use in mixed condition	Energy use in grouped condition	% of energy saved by using grouped condition
Temperature	44.2	34.2	23
Internal load	54.4	55.0	-1
Schedule	42.4	47.9	-13
Air exchange rate	47.3	48.9	-3

3.3.2 Representing Requirements

As described earlier, a parametric energy simulation study was conducted to identify which thermal parameters were necessary to group FUs for maximizing energy efficiency. Temperature was identified as a necessary criterion. Therefore, thermal decomposition is reduced to temperature decomposition in this work.

In FD, the input of temperature requirements is a collection of FUs with each FU having area and minimum / maximum temperature requirements.

Figure 3.8 shows an example of formulating temperature requirements in a graph representation. The nine nodes represent nine FUs. At first they are positioned in the one-dimensional Euclidean space according to their required average temperatures. The distance between any pair of FUs shows the difference in their required average temperatures. Therefore there is a derived relation, the difference in required average temperatures, between any pair of FUs. The closer two FUs are in the Euclidean space, the stronger the derived relation will be between them.

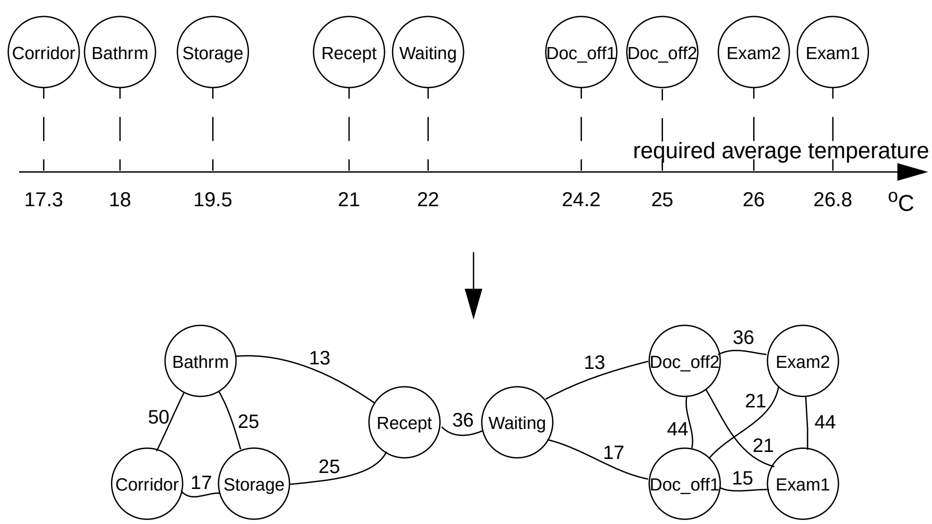

Figure 3.8 Representing temperature requirements in graph

In order to unify temperature requirements with adjacency requirements, a graph is derived to represent temperature requirements. The nodes in the graph are the FUs. The links between nodes show differences in temperatures between connected FUs. The smaller the difference in a temperature, the bigger the corresponding weight, and vice versa. For example, the temperature difference between Corridor and Bathrm (18-17.3=0.7°C) is small as compared to the temperature difference between Waiting and Doc_off2 (25-22=3^{o}C) in the input, therefore the weight between Corridor and Bathrm (50) is greater than the weight between Waiting and Doc_off2 (13) in the graph. Note that there should be edges linking between all possible pairs of FUs. Figure 3.8 only shows some of these edges for illustrative purposes.

Thus the problem is one of grouping FUs into floors and zones according to the average (mean value) temperature requirements. FUs with similar average temperature requirements, therefore greater weights, will be grouped into the same floor or zone, and vice versa, subject to meeting specified floor/zone number and size constraints. The objective function of stacking is as follows:

$$bjective = min\left(\sum_{i=0}^{Nf}\left(\left(\sum_{j=0,\, FU_j \subset Floor_i}^{Nfu_{Floor_i}}(temp_{FU_j} - avgTemp_{Floor_i})\right) / Nfu_{Floor_i}\right.\right.$$

In the above formula, $Floor_i$ is the ith floor in a building, $temp_{FU_j}$ is the required temperature of FU_j, $avgTemp_{Floor}$ is the required average temperature for all FUs on $Floor_i$, Nfu_{Floor_i} is the number of FUs on $Floor_i$, and Nf is the number of floors in the building.

Likewise, the objective function of zoning is as follows:

$$jective = min\left(\sum_{i=0}^{Nz}\left(\left(\sum_{j=0,\, FU_j \subset Zone_i}^{Nfu_{Zone_i}}(temp_{FU_j} - avgTemp_{Zone_i})\right) / Nfu_{Zone}\right.\right.$$

In the above formula, $Zone_i$ is the ith zone on a floor, $temp_{FU_j}$ is the required temperature of FU_j, $avgTemp_{Zone}$ is the required average temperature for all FUs within $Zone_i$, Nfu_{Zone_i} is the number of FUs within $Zone_i$, and Nz is the number of zones on the floor.

3.4 Acoustic Requirements

3.4.1 Identifying Requirements

Problem analysis, also known as content analysis, was carried out in order to obtain important grouping criteria. Problem analysis is to extract important factors through reasoning the relations between different factors under survey.

In order to analyze acoustic conditions under which FUs can be grouped or should be separated, the example of a two-story building was considered. Figure 3.9 is the section of the building. It contains both horizontal and vertical acoustic relations between adjacent FUs [Mahdavi et al., 1998].

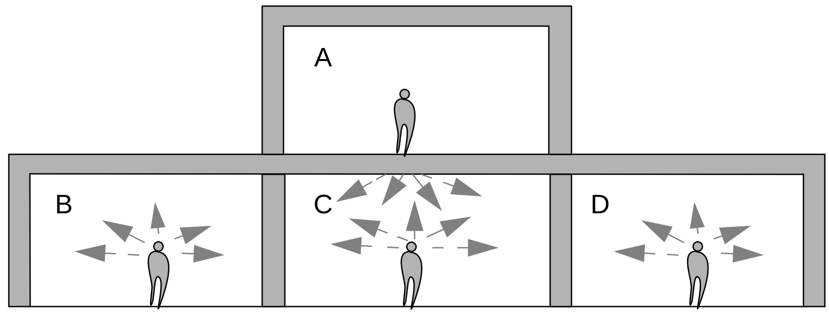

Figure 3.9 Acoustic problem analysis

Three factors, namely activity noise, background noise, and sound transmission class (or insulation class) of partition elements were shown to be significant for acoustic decomposition. Activity noise will be referred to as EEL (expected emission level), and background noise will be referred to as PSL (permissible sound level) later in the thesis.

The objective of acoustic decomposing was identified as avoiding using expensive insulation materials and technology to acoustically decouple adjacent FUs. In order to save construction cost, FUs with compatible background noise and activity noise should be grouped; those with incompatible activity noise and background noise should be separated.

The key issue is to minimize construction cost. Theoretically FUs with any noise levels can be adjacent to each other as long as appropriate noise control technologies are used to reduce the acoustic interference between adjacent FUs. The goal of minimizing construction cost is realized through the arrangement of FUs within a building, either by grouping acoustically compatible ones or by separating ones that interfere with each other.

Different from thermal decomposition where the goal is to minimize operational cost (i.e., energy consumption), the goal of acoustic decomposition is to minimize insulation cost-- a one-time cost. The consideration of insulation cost in acoustic decomposition is a way to meet noise protection. Whereas acoustic requirements can be met through local mitigation (i.e., through partition elements to reduce noise from an adjacent space), thermal requirements cannot be met through such methods of local mitigation. Hence there is no inconsistency between the goals of thermal decomposition and acoustic decomposition.

3.4.2 Representing Requirements

When representing acoustic requirements in an appropriate computational form, some relevant acoustic formulae are used in order to derive acoustic relations between the FUs. These formulae will be set forth in the subsequent sections.

In this field of study, such formulae are represented without citations, because *a)* they are empirically derived, *b)* there are different individual approaches of representing them, and *c)* the nature of the acoustic parameters to be presented in this research are such that they can be estimated in a variety of ways.

3.4.2.1 Important acoustic parameters

For the purpose of this research, three acoustic parameters are important: (1) STC or sound transmission class describing the degree of decoupling between two adjacent FUs for air-borne sound, (2) IIC or impact isolation class describing the degree of decoupling between two vertically adjacent FUs for structure-borne sound, and (3) NC or noise criteria defining the permissible noise level in an FU. Unless otherwise specified, all quantities have units of decibels[1] in sections . through 3.4.2.3.

In order to formulate acoustic requirements, the above parameters are considered in terms of two criteria: (1) EEL or expected emission level, and (2) PSL or permissible sound level. To ensure that the maximum PSL requirements of an FU are met, the required air-borne and structure-borne decoupling level ($STC_{required}$, $IIC_{required}$) between this FU and an adjacent FU can be estimated as follows:

$$STC_{required} = EEL_{airBorne} - PSL + 8$$

and

$$IIC_{required} = EEL_{structureBorne} - PSL + 8$$

Here $EEL_{airBorne}$ refers to the adjacent FU's air-borne noise level, $EEL_{structureBorne}$ refers to the adjacent FU's impact noise level. The constant value of 8 decibel is added to the $STC_{required}$ and $IIC_{required}$ in order to ensure that the PSL value in the receiver FU is not significantly affected by the energy transmitted from the source FU.

As for the format of each of the above-mentioned attributes, FD provides, in this case, two forms of user input, namely discrete categories and/or actual decibel values as shown in Table 20. This will provide flexibility for the user.

1. Decibel (dB) is a convenient dimensionless unit for quantifying physical properties such as sound power, intensity, and pressure. Generally speaking, a decibel level is a logarithmic ratio of a measured physical parameter to a related reference value. The decibel scale is used to describe sound pressure level or sound level [Mahdavi, 1995].

Table 3.20 Discrete attributes of EEL and PSL

Scale Value	EEL	dB	PSL	dB
3	EL	105	EI	75
2	VL	90	VI	65
1	L	75	I	55
0	N	60	N	45
-1	Q	45	S	35
-2	VQ	30	VS	25
-3	EQ	15	ES	15

The discrete attributes for EEL, as shown in Table 3.20, include extremely loud (EL), very loud (VL), loud (L), neutral (N), quiet (Q), very quiet (VQ), and extremely quiet (EQ). The discrete attributes for PSL, also shown in Table 3.20, include extremely insensitive (EI), very insensitive (VI), insensitive (I), neutral (N), sensitive (S), very sensitive (VS), and extremely sensitive (ES). The decibel values represent illustrative sound levels of the corresponding discrete values which must be determined on a case-by-case basis.

Table 21 contains four examples of discrete acoustic requirements by FUs to concretely illustrate the level of loudness / sensitivity of the discrete EEL and PSL values.

Table 3.21 Four FUs' discrete acoustic values (example)

	EEL		PSL
	air-borne	structure-borne	
Kitchen	VL	L	I
Symphony hall	VL	VL	ES
Machine room	EL	EL	EI
Reading room	Q	N	VS

Acoustic decomposition includes horizontal decomposition and vertical decomposition. Horizontal decomposition is formulated as the task of dividing a collection of FUs into floors or zones with minimum construction cost. Vertical decomposition is formulated as the task of specifying required construction cost when a FU is located directly on top of another. In this way, appropriate constraints can be set to guide layout generation toward favorable design solutions.

3.4.2.2 Horizontal decomposition

Horizontal decomposition is realized in two steps, namely required construction-cost calculation and formation of a graph, and partition of the graph. In step one, in order to calculate required construction cost between any pair of FUs, $STC_{required}$ needs to be calculated as an intermediate step. The two $STC_{required}$ values of each pair of FUs will be calculated according to their $EEL_{airBorne}$ and PSL values. Among the two calculated $STC_{required}$s, the greater one is the required decoupling between the two FUs, therefore it will be used a basis in calculating the required construction cost between the two FUs. When the construction costs between all pairs of FUs are calculated, a graph of FUs with weights representing different construction costs will be formulated similar to the one shown in Figure 3.10.

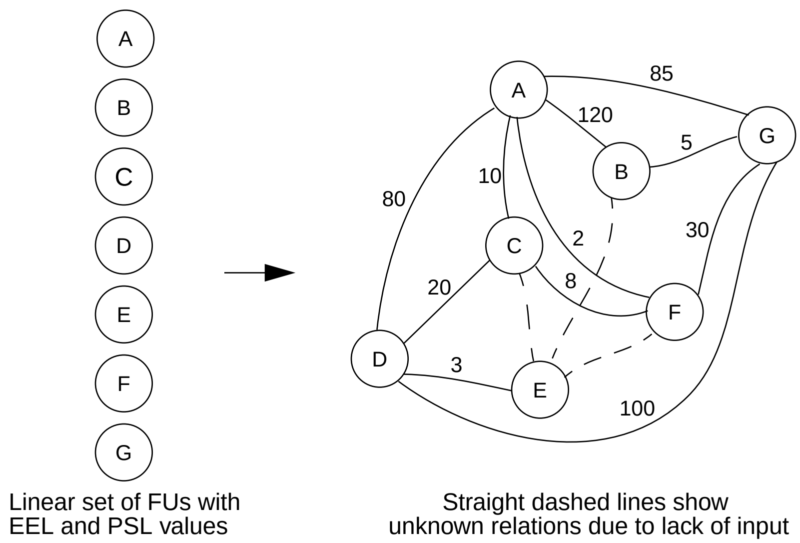

Figure 3.10 Representing acoustic requirements in graph

It is assumed here, for demonstrative purposes, that a required construction cost is dependent on two factors, namely $STC_{required}$ and size of the shared wall between the two FUs. When the height of walls is considered constant, L_{wall}, the length of the shared wall between the two FUs, can be used to estimate the size of the shared wall:

$$wall = min(sqrt(Area_{FU1}), sqrt(Area_{FU2}$$

Obviously, construction cost, expressed in relative cost units, is a complex function of STC. A highly simplified assumption is to correlate cost with material use (expressed in terms of surface density m in kg.m^{-2}) and use an approximate function for the relation between STC and m:

$$TC \cong 32.4\log(m) - 2$$

Assuming χ is a cost coefficient to be determined on a case-by-case basis, the following relation can be derived:

$$ConstructionCost = \chi \times 10^{(STC_{required} + 26)/32.4} \times L_W$$

This equation is used to derive required construction cost due to the partition element between a pair of adjacent FUs.

In step two, a graph-partitioning algorithm is run on this acoustic graph to generate floors or zones. Those requiring greater noise protection costs for decoupling should be separated into different floors or zones in order to minimize noise protection cost; those requiring smaller noise protection costs for decoupling should be grouped into the same floor or zone, since they do not need much decoupling when allocated together. The objective function of stacking is as follows:

$$Objective = min\left(\sum_{k=1}^{Nf} \sum_{i, j, i \neq j}^{FU_i, FU_j \subset Floor_k} Wall_{FU_i, FU_j}\right)$$

In the formula above, $Floor_k$ is the *k*th floor in a building, $Wall_{FU_i, FU_j}$ is the insulation cost of the wall between FU_i and FU_j on $Floor_k$, and Nf is the number of floors in the building.

Likewise, an objective function of zoning is as follows:

$$Objective = min\left(\sum_{k=1}^{Nz} \sum_{i, j, i \neq j}^{FU_i, FU_j \subset Zone_k} Wall_{FU_i, FU_j}\right)$$

In the formula above, $Zone_k$ is the *k*th zone on a floor, $Wall_{FU_i, FU_j}$ is the insulation cost of the wall between FU_i and FU_j in $Zone_k$, and Nz is the number of zones on the floor.

3.4.2.3 Vertical decomposition

Vertical decomposition is to calculate and represent a required construction cost between any pair of FUs when they are vertically adjacent. The processes of vertical decomposition include calculating $IIC_{required}$ and $STC_{required}$ between any pair of FUs if one is to be located on top of the other, and calculating the corresponding required construction cost and representing it using a graph.

To calculate $IIC_{required}$ and $STC_{required}$, assuming FU-A is located on top of FU-B, the following relations can be derived similar to the equations in horizontal decomposition:

$$STC_{required(B \to A)} = EEL_{B, airBorne} - PSL_A + 8$$

$$STC_{required(A \rightarrow B)} = EEL_{A,\ airBorne} - PSL_B + 8,$$

and

$$C_{required} = EEL_{A,\ structureBorne} - PSL_B + \ .$$

Among the two calculated STCs ($STC_{required(B->A)}$ and $STC_{required(A->B)}$), the greater one is the required decoupling between the two FUs.

It is assumed here that a required construction cost is related to two factors, i.e., construction cost resulting from $IIC_{required}$ and construction cost resulting from STC $_{required}$. A simplified approach is already discussed to derive construction cost estimates based on the required STC levels. Again, if matters are substantially simplified, cost may be correlated with material use expressed in terms of surface density *m* in kg.m^{-2}, and an approximate function can be used to establish a relationship between IIC and surface density:

$$IC \cong 35\log(m) - 54$$

The construction cost resulting from $IIC_{required}$ can be then calculated as follows:

$$nstructionCost = \gamma \times 10^{(IIC_{required} + 54)/35} \times Area_{sha}$$

In this equation, $Area_{shared}$ is the shared floor/ceiling area between the two FUs, i.e., the smaller area of the two FUs'; γ is a coefficient that must be determined on a case-by-case basis.

The construction cost resulting from $STC_{required}$, similar to the horizontal decomposition case, is calculated as follows:

$$structionCost = \chi \times 10^{(STC_{required} + 26)/32.4} \times Area_{sha}$$

To unify the construction cost implications of the air-borne and structure-borne decoupling requirements for the vertical decomposition, the following procedure may be followed. First, the minimum necessary surface density is calculated to fulfill both STC and IIC requirements for an FU-dividing partition element. Second, the larger of these two surface density values is selected to modify either the STC or IIC requirements. Third, these modified requirements are used to derive the pair of construction costs. Fourth, the higher of the two construction costs is used in the graph representation. Again, the construction cost is expressed in relative cost units.

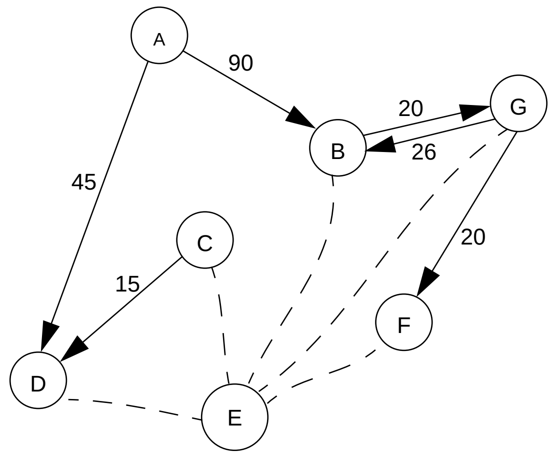

Figure 3.11 Cost graph of vertical placement of FUs

A typical output for vertical decomposition is similar to the one shown in Figure 3.11. It is a graph with weights on edges showing required construction costs in relative units. Specifically, a weight between two adjacent FUs shows the required construction cost for decoupling if the "from" FU is located right on top of the "to" FU. For example, in case of FU-A which is located right on top of FU-B, 90 cost units will be needed. Here, each edge with its assigned cost units is expressed as a constraint (relationship) between two FUs.

In the case of using FD's findings in layout generation, this output would take the form of a set of constraints to be observed when generating layouts for two vertically adjacent floors. In SEED-Layout, all of the vertically adjacent FUs will be checked against the required noise protection cost of the ceiling or floor partitioning. If there is any part of the layout that requires a high noise protection cost, there will be a warning for that portion of the design.

3.5 Merging Engine

Since each design criterion (adjacency, thermal, and acoustic) is represented in a graph, these multiple graphs can be combined into a single graph which comprehensively represents all criteria or requirements.

Figure 3.12 is an example showing how the merging engine works. The user specifies an importance number for each of the adjacency, thermal, and acoustic requirements. These numbers represent the relative importance between the three criteria. In this example, the user specifies 10, 3, and 1 for the importance of adjacency, thermal, and acoustic requirements, respectively. This shows that the user considers adjacency the most important factor in grouping, whereas thermal and acoustic are relatively minor aspects.

Given these importance numbers, each weight on an edge is multiplied by the importance number specified for that graph. Then the three graphs are combined into a single graph in a way that the weights between the same pairs of FUs are added together for a combined weight. The combined weights in the example are 124 between FU1 and FU2, 60 between FU1 and FU3, and 52 between FU2 and FU3.

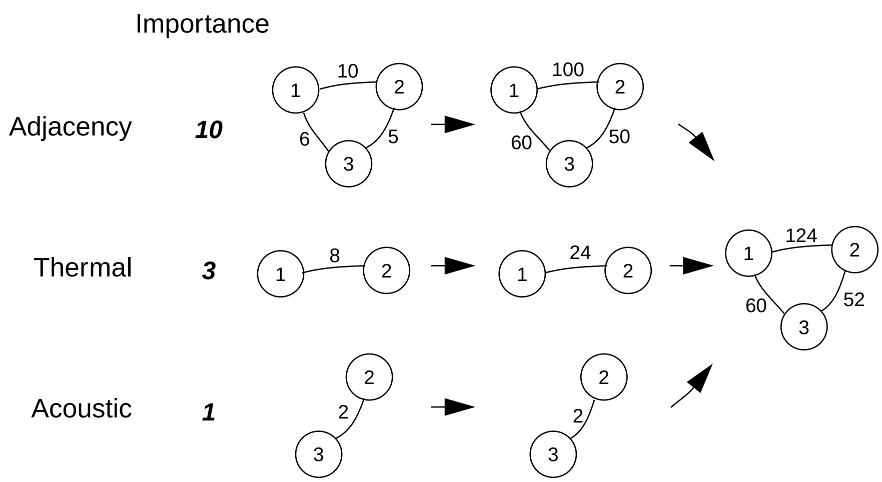

Figure 3.12 Merging engine

After a combined graph is thus derived, it will be partitioned by the decomposition algorithms. Since the algorithms can be applied to either a combined graph representing multiple design requirements, or a graph representing a single dimension of design requirements, partitioning a combined graph is equivalent to combining the outputs of a separately partitioned adjacency graph, thermal graph, and acoustic graph. Therefore decomposition according to the combined requirements is a way to realize the following multiple objectives:

- minimizing total adjacency relations between different floors or zones;
- minimizing energy consumption; and
- minimizing noise protection cost.

3.6 Decomposition Algorithms

This section first discusses the process of stacking and zoning, which is applicable to all decompositions (adjacency, thermal, and acoustic). Then it describes the expected outputs of the stacking and zoning process. Lastly it introduces the decomposition algorithms that can either partition a unified graph, or partition separately the graphs representing adjacency, thermal, and acoustic requirements, respectively.

3.6.1 Process of Stacking and Zoning

The process of decomposing a building includes three separate steps: stacking, horizontal zoning, and vertical zoning. The need for vertical zoning arises from the existence of vertically aligned FUs, such as bathrooms and staircases. Vertically aligned FUs form vertical zones. Each of the three decomposition steps will be discussed at greater detail in the following sections.

3.6.1.1 Stacking

In stacking, FUs are grouped by floor in order to minimize the total weight between different floors. An objective function for stacking is as follows:

$$Objective = min\left(\sum_{m=1}^{Nf-1}\sum_{n=m+1}^{Nf}\sum_{FU_i \subset Floor_m,\ FU_j \subset Floor_n,\ m \neq n} Weight_{FU_i, FU_j}\right)$$

In the objective function above, $Weight_{FU_i, FU_j}$ is the strength of the relation between FU_i and FU_j, and Nf is the number of floors in a building. “Total weight between different floors”, as shown in the formula, is the sum of weights or strengths of relations between all pairs of FUs within different floors. It indicates how well the floors are partitioned.

Note that although this objective function resembles the one used in adjacency decomposition (Section 3.2.2), the two are different. In adjacency decomposition, $Adjacency_{FU_i, FU_j}$ is an adjacency relation between two FUs, whereas in the objective function above, $Weight_{FU_i, FU_j}$ is a unified relation derived from the merging engine. It includes the similarity of temperature requirements and required noise protection, in addition to the adjacency relation between two FUs.

Figure 3.13 is the expected partitioning result of the graph shown in Figure 3.2. In this partitioning result, the total weight between the floors is 10+2+5=17. No other way of grouping the FUs can achieve a smaller total weight between the floors, subject to meeting the pre-defined area constraints of each floor. For this particular example, it is assumed that the number of floors and the area of each floor are fixed.

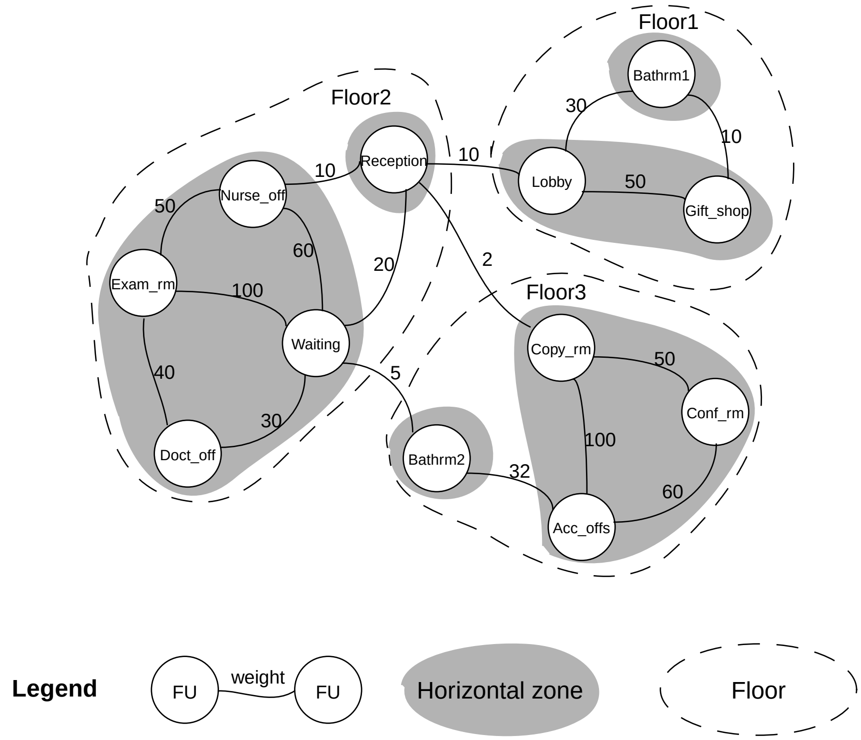

Figure 3.13 FUs with their relations (example)

3.6.1.2 Horizontal zoning

In horizontal zoning, FUs are grouped by zone in order to minimize the total weight between different zones. An objective function for zoning is as follows:

$$\mathit{bjective} = \mathit{min}\left(\sum_{m=1}^{Nz-1}\sum_{n=m+1}^{Nz}\sum_{FU_i \subset Zone_m,\ FU_j \subset Zone_n,\ m \neq n} \mathit{Weight}_{FU_i,\ Fl}\right.$$

In the above objective function, $\mathit{Weight}_{FU_i, FU_j}$ is the strength of the relation between FU_i and FU_j. Nz is the number of floors in a building. “Total weight between different zones”, as shown in the formula, is the sum of weights or strengths of relations between all pairs of FUs within different zones. It indicates how well the zones are partitioned.

Similarly, although this objective function resembles the one used in adjacency decomposition (Section 3.2.2), the two are different. In adjacency decomposition, $\mathit{Adjacency}_{FU_i, FU_j}$ is an adjacency relation between two FUs, whereas in the above objective function, $\mathit{Weight}_{FU_i, FU_j}$ is a unified relation derived from the merging engine. It includes the similarity of temperature requirements and required noise protection, in addition to the adjacency relation between two FUs.

In the example in Figure 3.13, assuming the user wants two zones on Floor1, Lobby and Gift_shop are assigned to the same zone whereas Bathrm1 is a separate zone by itself, because the weight between Lobby and Gift_shop (50) is greater and thus they should be grouped.

There are two optional methods of deciding at what points should FUs be separated into different horizontal zones. The first method is to let the user specify the number of zones and the relative area of each zone. But this is unlikely in the early stage of design. The second method is to allow the user to specify a threshold relation. This applies to the situation when the user knows what levels of relations are relatively more important than others and is able to express this in the form of a threshold weight. FUs with greater weights than the threshold weight should be grouped into the same zone, and those with smaller weights should be separated into different zones.

3.6.1.3 Vertical zoning

For vertical zoning, FUs with vertical relations are organized as vertical zones. Vertical relations include vertical plumbing connections (e.g., bathrooms), vertical traffic connections (e.g., staircases), and vertical spaces (e.g., atria and theaters). For plumbing or traffic connections, a same FU with either a plumbing or traffic function is created for each floor. On each floor, this FU is treated the same way as other FUs regarding its adjacencies, thermal, acoustic, and daylight requirements.

For an atrium or a theater with a multi-story height, a method similar to the one used for a vertical plumbing or vertical traffic function can be employed, such as creating a vertical FU subsuming the space of several floors. This FU is treated the same way as other FUs on each floor regarding its various design constraints. Another way to handle a vertical FU is to consider it as a “negative space”. A “negative space” can be modeled as an “outside FU” in the middle of the building while the adjacency relations between this “outside” FU and other FUs are still valid.

In summary, vertical zoning involves creating the same FU for each floor it occupies and organizing the FUs on continuous floors into a vertical zone. The FUs created for each floor can be treated like other FUs regarding various design constraints and functional requirements.

3.6.1.4 Output

The output of FD is represented in the form of an FU hierarchy. It shows which FUs should be grouped into the same floor / zone, and which FUs should be separated. Figure 3.14 shows the FU hierarchy output of stacking and horizontal zoning for the example in Figure 3.13. The shaded FUs are the original input, i.e., nodes in Figure 3.13. The links represent spatial containment relations; e.g., Floor1 spatially contains Zone1 and Zone2. The input FUs are aggregated into floors, and again into zones, and possibly into sub-zones.

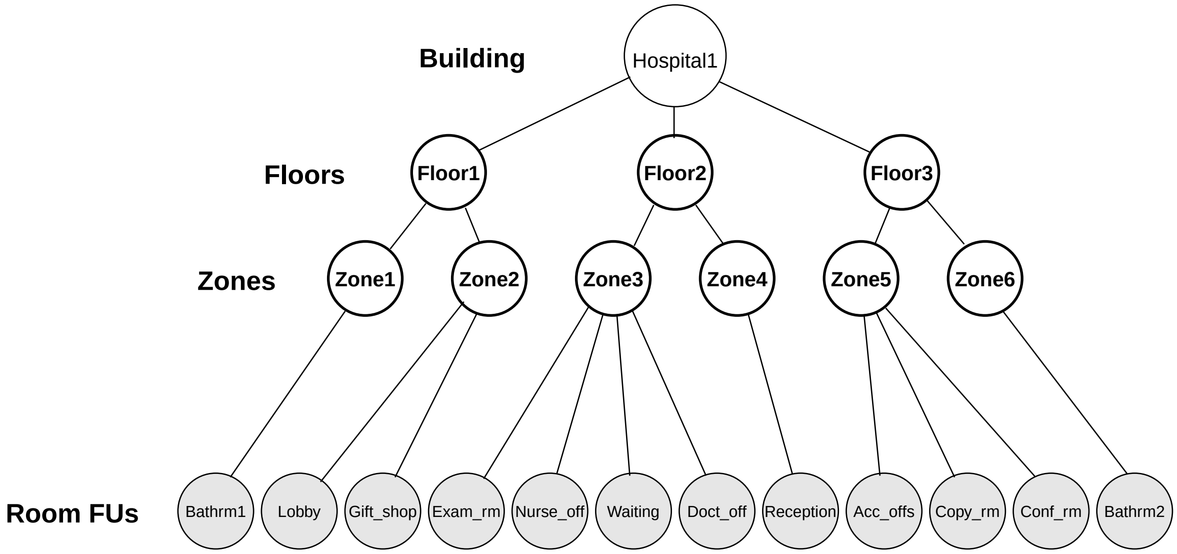

Figure 3.14 Output of adjacency decomposition (example)

While it is used to represent stacking and zoning designs, this hierarchical structure can also represent vertical zones in a building. Figure 3.15 shows the vertical zones for the example in Figure 3.13. The bathrooms on different floors form one vertical zone (V-zone1); and the elevators on different floors form another vertical zone (V-zone2). The vertical zones express the requirement that FUs within each vertical zone be vertically aligned during the stage of layout generation.

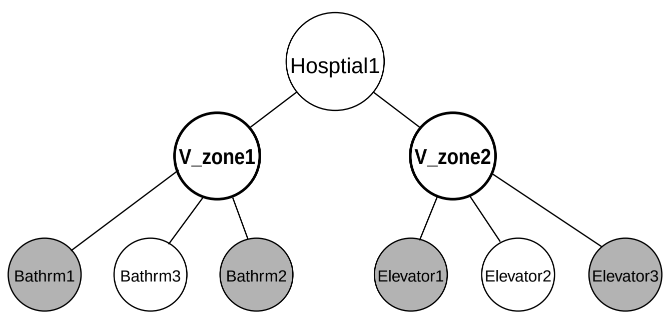

Figure 3.15 Output of vertical zoning (example)

3.6.2 Description of Algorithms

3.6.2.1 Stacking algorithm

The brute force approach in graph-partitioning is to try all the possible ways of partitioning a graph and find the one that will result in the minimum total weight between different groups. This is categorized as an NP-complete problem because the time it takes increases rapidly with problem size; it will be extremely slow when the problem size exceeds a certain value.

In the past, many improvements have been realized to make graph-partitioning problems solvable within polynomial time. A milestone method was achieved through the Fiduccia-Mattheyses graph-partitioning algorithm [Fiduccia and Mattheyses, 1982]. Its contribution is to reduce the running time to linear by "locking" moved nodes and using a special data structure called a bucket list[1] to record expected results of possible moves. This reduces the global optimization problem to one of local optimization.

The original problem Fiduccia and Mattheyses tried to solve was to partition a graph of nodes into two blocks[2] with size constraints. They had a random initial partition and tried to improve this partition by moving a node from one block to the other which would result in the maximum decrease in the total weight between the two blocks[3]. One heuristic they used was to "lock" a node once it was moved so that it would not be moved again. Thus only unlocked nodes were considered for future moves. This method of "locking" moved nodes reduced the computation time from an exponential one to a polynomial one. They also used a bucket list to record the expected decrease in the cut size[4] for each unlocked node. Each time a node was moved, only an update of the gains of the node's neighbors were necessary in the bucket list; calculation of future gains for all the unlocked nodes was not needed. This local update of the bucket list further reduced a polynomial sequences of operations to one that could be solved in linear time.

It becomes obvious that, if adapted to solve the stacking problem, the Fiduccia-Mattheyses algorithm imposes a restriction on the number of floors (only two floors can be handled). In order to adopt this algorithm, improvement has been made so that it can handle any integer number of floors.

The stacking algorithm used in FD is an adaptation of the Fiduccia-Mattheyses graph-partitioning algorithm. The data structure of a graph is shown in Figure 3.16. A graph consists of a set of nodes and a set of edges. Each node is connected with zero or more edges. Each edge connects two nodes. It should be noted that either a node_list or an edge_list is sufficient to represent a graph. The redundant use of both facilitates an efficient retrieval of the edges connected with a node, or the nodes connected with an edge.

In a graph in FD, a node represents an FU, and an edge represents the adjacency/thermal/acoustic relation between the two connected FUs.

---

1. The definition of bucket list will be given later in this section.
2. A block is a structure that contains a set of nodes.
3. A total weight between blocks is called a cut size.
4. Decrease in a cut size is called a gain.

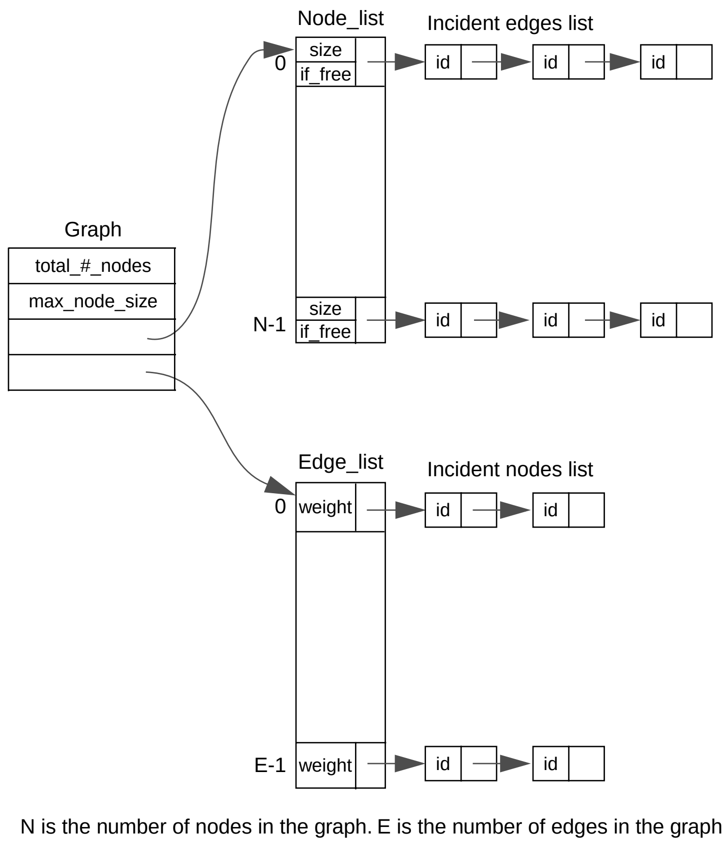

Figure 3.16 Data structure of a graph

The improved algorithm partitions a set of FUs into a pre-defined number of floors with area constraints. Each of the resulting floors contains a set of FUs that satisfy specified floor area requirements. The algorithm is described as follows:

After the user has specified the number of floors, num_floors, and each floor's area, the FUs will be sorted by their total connected weights in descending order. The first num_floors -1 FUs will be assigned to the first num_floors - 1 floors respectively, and the remaining FUs will all be assigned to the last floor. Note that this initial partition normally violates area constraints of the floors. Therefore, the graph-partitioning process will not only minimize the total cut size, but will also readjust the areas of the floors so that the area constraints of the floors can be met.

A bucket array (see Figure 3.17) will be set up, and the future gain of each free[1] FU will be calculated and registered in this bucket array. In Figure 3.17, a block represents a floor. A possible move can be of any unlocked FU from its current floor to any of the other floors. Thus each

1. At the beginning, all the FUs (nodes) are free.

floor has a potential relation to every other floor. This relationship is represented by an array. Each array entry is a bucket list which registers the gains of FUs if they are moved from their current floor (row index) to a destination floor (column index). The maximum possible gain caused by a single move is the maximum total connected weight[1] of an FU within the graph.

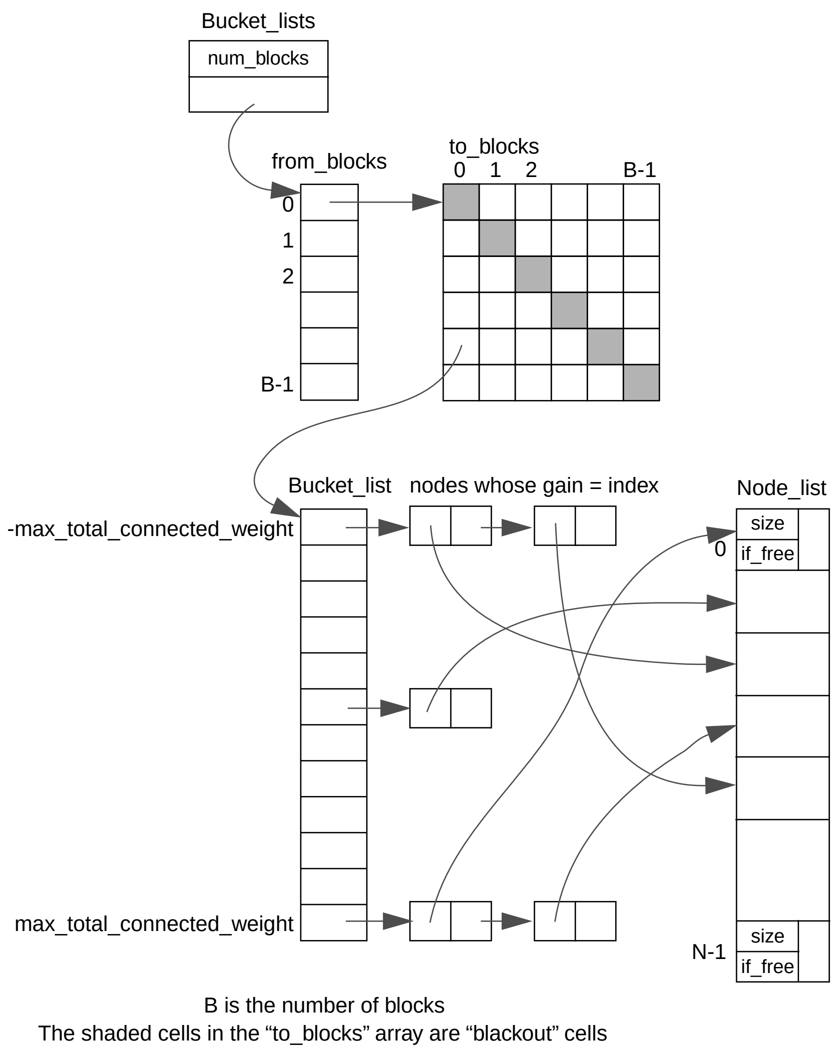

Figure 3.17 Data structure of a bucket array

1. A total connected weight of a FU (node) is the sum of the weights on all edges that are connected with the FU (node).

The move with the maximum gain and more balanced[1] areas between floors will be selected. The corresponding FU will be moved from its current floor to the destination floor as specified in the bucket array, and then locked. The bucket list will be updated accordingly.

Again, the next free FU with the maximum gain and which, if moved, will result in more balanced areas between the floors will be selected, moved, and locked. The bucket list will be updated correspondingly. This process will be repeated until either all of the FUs are locked or no moves will remain that result in more balanced floor area values.

In the above process, an area balance is achieved by selecting moves that will not only decrease the total cut size, but also improve the area balance between floors. Each time a node is moved, the areas of its origin floor and its destination floor are locally adjusted. Thus the optimization process both minimizes the total cut size and improves the area balance in order to generate groupings that meet specified floor area constraints.

Each time an FU moves, a new state[2] will be generated. Figure 3.18 shows the data structure of a state. A state consists of a set of blocks and a set of cut sets. A block represents a floor. Each block contains a list of FUs (nodes). Each cut set contains a set of edges linking nodes within two different blocks.

---

1. A floor's area is the total area of all the FUs assigned to that floor. Balance is the condition that all the floors' areas are the same as their required areas.
2. A state is a partition of FUs (nodes), with each of the FUs assigned to a floor (block).

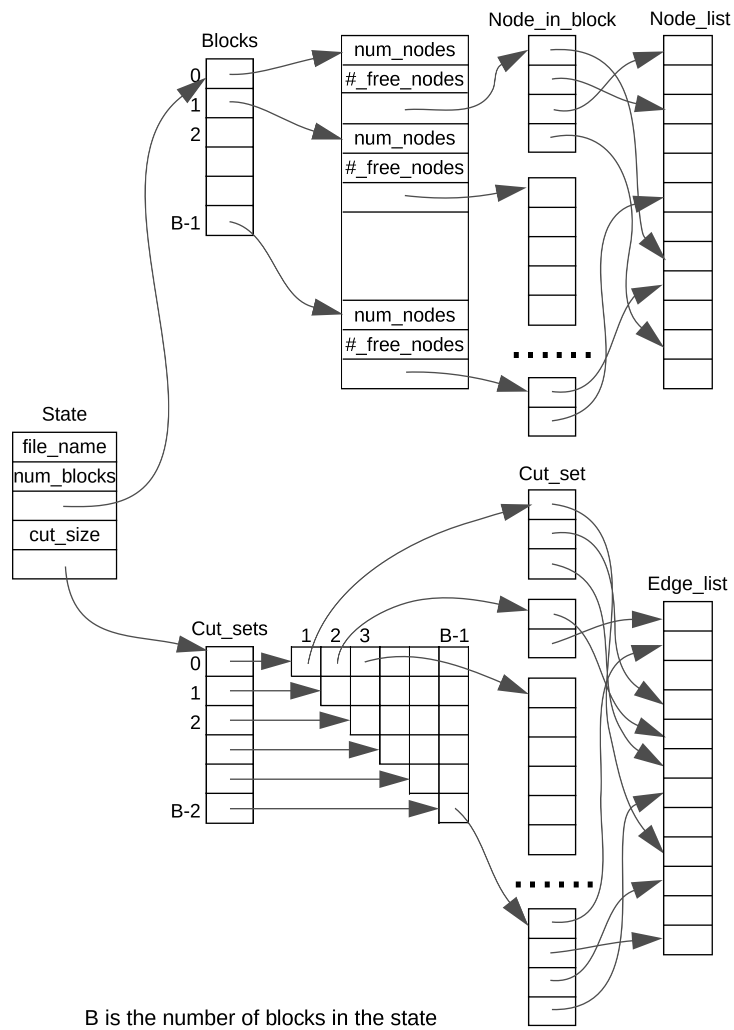

Figure 3.18 Data structure of a state

The results of the stacking algorithm will be a set of states. Those that satisfy the specified floor area requirements are feasible states. An optimal state will be the one which among all the feasible states has the minimum total cut size.

When the stacking algorithm applies to thermal or acoustic decomposition, temperature differences or required construction costs between all possible pairs of FUs are needed. This requires a complete graph to be formulated. When the time it takes to formulate a graph is $N_{fu} \times (N_{fu} - 1)/2$, i.e., $O(N_{fu}^2)$, and the partition of such a graph takes $O(N_{fu}^2)$ time; thus, the time order for this algorithm is $O(N_{fu}^2)$.

The pseudo code for the stacking algorithm is as follows:

```
formulate a graph with node_list and edge_list;
sort node_list into sorted_node_list by total_connected_weight;
total_num_free_nodes = total_num_nodes;
if (there is pre-assignment) {
        pre-assign nodes to specified floors;
        lock the nodes;
        adjust total_num_free_nodes;
}
for (i =  0 to num_floors-1) {
        if (i < floor_num-1 && this floor is empty)
                assign next node in sorted_node_list to this floor;
        if (i == num_floors-1)
                assign remaining nodes in sorted_node_list to this floor;
}
build bucket_array;
//Perform only one pass. This is different from Fiduccia and Mattheyses
//which allows multiple passes over data
while (total_num_free_nodes > 0) {
        get feasible_moves from bucket_array;
        select best_move from feasible_moves;
        if (best_move != NULL) {
                move best_move;
                lock this node;
                total_num_free_nodes--;
                adjust areas of from_floor and to_floor;
                update bucket_array;
                generate a new state;
                //save all feasible states that have been observed to feasible_states_list
                if (the state meets floor area constraints)  //a new feasible state
                        add it to feasible_states_list;
        }
        else
                break;
}
get_optimal_state from feasible_states_list;
```

There are several improvements that can be made to this algorithm to make it more similar to Fiduccia-Mattheyses. They are elaborated as follows:

- In the **initial partition**, instead of assigning all of the FUs to the top floor, either of the following two strategies could be adopted: *a)* assigning FUs to num_floors floors only according to the weights between them, and *b)* assigning FUs to num_floors floors only according to the area constraints. For *a*, the greedy clustering algorithm as described in Section 3.6.2.2 can be used to formulate this initial partition. For *b*, a good heuristic for assigning FUs on the floors would be to put the largest FUs on the floors first. For example, put the largest FU on floor one. If there is sufficient space for the second largest FU, assign it to floor one as well, otherwise on floor two, and so on. Either *a* or *b* can achieve a better balance of areas among the floors, as compared to the method described in Section 3.6.2.1.

- When **selecting a move**, instead of choosing one that strictly improves the balance of floor areas, temporary moves that violate area constraints should be allowed with the possibility that subsequent moves will satisfy the area constraints even better. A penalty is associated with each move that violates the area constraints. Therefore, unless there is a significant decrease in the cut size, moves that violate area constraints will not be considered. Thus, the criterion for selecting a move is to minimize both total cut size and penalty for violation of area constraints:

$$objective = min(totalCutSize + areaPenalt$$

  In the objective function above,

$$otalCutSize = \sum_{m=1}^{Nf-1} \sum_{n=m+1}^{Nf} \sum_{FU_i \subset Floor_m, FU_j \subset Floor_n, m \neq n} Weight_{FU_i, F}$$

$$eaPenalty = c \times \sum_{p=1}^{Nf} abs\left(\left(\sum_{FU_q \subset Floor_p} Area_{FU_q} - reqArea_{Floor_p}\right) / reqArea_{Floo}$$

  Nf is the number of floors in a building, $Floor_p$ is the *p*th floor, FU_q is the *q*th FU on floor *p*, $Area_{FU_q}$ is the area of the *q*th FU, $reqArea_{Floor_p}$ is the required area of the *p*th floor, *c* is a coefficient to be decided on a case-by-case basis, and *abs* is a function of getting the absolute value of a number.

  When all of the FUs are locked, the states that have been generated should be reviewed to determine an optimal state with a minimal total score (sum of total cut size and area penalty). If no state meets the area constraints, the area penalty should be increased and the process repeated. If there is still no state that meets the area constraints, a different initial partition method should be utilized.

- Fiduccia-Mattheyses allows **multiple passes** over data in order to continuously improve the partition result. But multiple passes in the described stacking algorithm is impossible, because each move improves the area balance among the floors. After the first pass, the floor areas are balanced. Therefore, moving an FU will violate the area constraints. As a result, no additional moves are allowed. Only when a different strategy of selecting moves is adopted, as described in the "selecting a move" section above, can multiple passes be realized.

3.6.2.2 Zoning algorithm

The objective function for (horizontal) zoning has been stated in Section 3.6.1.2. Zoning is a different problem from stacking. Although the user may define a maximum or a minimum zone area, there are no strict area constraints for each zone. As such, it is not as strict as stacking where each floor has an area. The major issue in zoning is to group FUs with strong relations together. The stacking algorithm is "expensive" for this purpose in terms of setting up and updating bucket array operations. Therefore, a different algorithm from the stacking algorithm should be used for zoning to focus only on the relations between FUs.

FD zones FUs on each floor based on the so-called "greedy clustering" method. In a graph-partitioning problem, the basic process of greedy clustering is to group nodes into clusters, and gradually merge these clusters until some termination condition is satisfied. When adopted for FD, nodes are used to represent FUs, and clusters are used to represent zones. The criterion of zoning is to group FUs linked by strong weights into the same zones. This process guarantees that FUs linked with strong weights are grouped according to some specified constraints.

Nodes in a graph represent FUs assigned to a floor, and edges represent adjacency / thermal / acoustic relations between the FUs. The greater the weight on an edge, the closer the two linked FUs should be located to each other. The user may specify a required number of zones (<= number of FUs on the floor) and a maximum zone size (>= maximum FU size).

Figure 3.19 illustrates an example of how FD adapts the greedy clustering method in floor zoning. Initially, all edges are sorted in descending order according to their weights. After sorting, the order of the edges is e1, e4, e6, e3, e5, and e2. The initial partition is such that each FU is in a separate zone. Then all of the edges are visited in the sorted order. If two linked FUs are in different zones, their zones will be merged if the required number of zones is not violated. This process terminates when either the number of zones is reduced to the required number, or all of the edges in the graph have been visited.

When e is the total number of edges in a graph, quicksort takes $O(e\log(e))$ time, and traversing the edges and merging the related nodes takes $O(e\log(e))$ time. Thus the time order for this algorithm is $O(e\log(e))$.

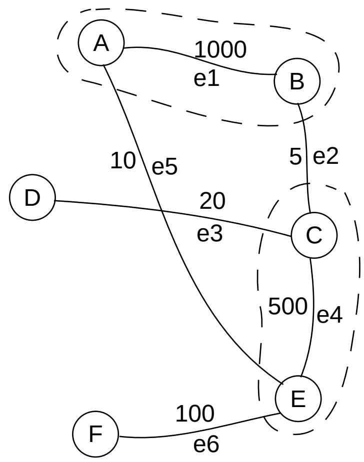

Figure 3.19 Greedy clustering method in floor zoning

The pseudo code for zoning is as follows:

```
//check if the required number of zones n and size max_zone_size are reasonable;
if (max_zone_size < maximum node size)
        exit; //zone size is not reasonable;
else if (the required number of zones n > number of FUs on floor)
        exit; //zone number is not reasonable;
else if (the required number of zones n = number of FUs on floor) {
        assign each FU to a separate zone;
        exit;
}
create a sub-graph for the floor with FU_list and edge_list;
quicksort the edge_list according to weights in descending order;
assign each FU to a separate zone;
while (current number of zones > n, for each sorted edge)  {
        get two linked FUs FU1 and FU2;
        if (FU1 is in zone1 && FU2 is in zone2
        && size(zone1) + size (zone2) <= max_zone_size) {
                merge (zone1, zone2);
        }
}
```

This program was implemented under X-Windows on a Sun Sparc Solaris machine (sun4m, SPARCstation-5). The programming language used was C++ with a CC compiler. The size of the program is 10,948 lines of source code.

4 User Interface

Interface design is especially important in FD. This is because architects' willingness and ability to use computers hinges to a large extent on the ease of use of the CAD software at their disposal (Section 1.1). Despite the intense work devoted to the development of interfaces, interface design in the building design field remains an area that needs more research.

FD interface design facilitates and simplifies the process of stacking and zoning. In order to achieve this end, there are several design guidelines to comply with.

4.1 Interface Design Guidelines

4.1.1 Simplicity

The system is not only designed for experienced users, but also for novice and intermittent users. Cognitive psychology shows that perceived complexity of a system will make the user feel that there is too much information to handle; and it will increase the intimidation factor usually present in the acquisition of a new skill. The FD interface is designed such that there are only a few important elements visible at a time[1].

In order to accommodate the needs of different levels of users, the interface is designed with the progressive disclosure method. Progressive disclosure allows the interface to present the most common choices to the user while initially hiding more complex choices, so that a novice user will feel the system is easy to learn and an experienced user will feel that the system has all the features and power that he / she desires.

4.1.2 Consistency

The FD interface enables the user to bring to bear his / her previously gained knowledge upon new areas without having to learn from scratch. This requires that the interface design be compliant with generic interface standards and the user's expectations. Also within the FD system itself, consistency is maintained at all levels. For example, double-clicking on a rectangular room brings up a dialog window with detailed information on the room; likewise, double-clicking on a tree node brings up a dialog window with information on the FU that the tree node represents.

1. In order to make it look simple, the interface is designed to display a limited number of elements and only show important ones, instead of displaying detailed information. For example, in the floor section presentation, each FU is simply a rectangle. The functional requirements of FUs will not be displayed unless the user queries them.

4.1.3 Flexibility

The system is designed to be flexible. This is achieved by providing different options for customizing the working environment. For instance, some users like intuitive representation of floor section drawing; whereas some other users prefer tree representation so that they can see the grouping results quickly. The system provides both options to cater to different preferences.

4.1.4 Interactivity

The system is designed to actively support interaction between the user and the system. Specifically, the system presents design information for the user, and the user has the freedom of modifying any information that is entered earlier or pre-stored by the system. The system provides automatic support for the user; but the user has full control of the whole process and results.

4.1.5 WYSIWYG

The interface faithfully shows what is going on with the system. All the information is open to the user, including the execution process of the algorithms. In this way the user is well informed of how information is being processed by the system, and how data is being transformed by specific algorithms; the user is thus able to make the right judgement and decision correspondingly.

4.1.6 Feedback

Each time the user does something, there will be immediate feedback which shows the outcome of the user's action. If a process takes a long time to execute, the system will provide a progression indicator showing how much work has been done and how much more time is expected.

4.2 Features

This section describes some of the concrete interface features of FD, including direct manipulation, progressive disclosure, and dynamic display of algorithmic processes. These are illustrations of how the guidelines mentioned above have been implemented.

4.2.1 Direct Manipulation

The user is able to "see-and-point". The user can simply perform an operation on an object without using any menu command. For instance, the user can select an FU on a floor and drag it to another floor. For another example, double-clicking on a room or a tree node brings up a window showing detailed information for the FU that the room or tree node represents.

Direct manipulation is an example of simplicity, consistency, interactivity, WYSIWYG, and feedback guidelines.

4.2.2 Progressive Disclosure

The interface is designed in such a way that it provides a seemingly simple workspace, which is sufficient for a user to perform basic operations such as stacking and zoning. But more complex and advanced functions will be uncovered as the user explores the system. For instance, the design requirements window initially looks like a window containing a few document folders. Each folder, when clicked on, will be unfolded into a window showing detailed design requirements.

Employment of the technique of progressive disclosure helps the user to receive information in a gradual fashion. It also adapts to users with different levels of complexities. Progressive disclosure is an example of simplicity and flexibility guidelines.

4.2.3 Show/hide Options

FD interface allows the user either to see the algorithm results directly, or to view the algorithmic processes inside the system, when the graph-partitioning or clustering algorithm is running. In the second mode, the system displays the algorithmic processes by moving nodes around in the tree view, or moving rooms from one floor to another in the floor view.

Other show / hide options include allowing the user to expand or collapse a portion in the tree view if he / she does or does not want to view that portion, using a progression bar indicating how long an internal process will take, and providing an information bar showing what is going on in the system.

The show / hide options are a reflection of flexibility, feedback, and WYSIWYG guidelines.

4.3 Design

4.3.1 Interface Use Cases

Like most other engineered software system, FD interface consists of four typical design and development stages [Pfleeger, 1991], namely functional specification (use cases design), design specification (specification of object schemata), implementation, and testing.

To date eighteen interface use cases have been developed using the OOSE method [Jacobson et al., 1992]. Each of them corresponds to a utility that FD provides. For a detailed description of these use cases, see interaction diagrams depicted in Appendix D (User Interface Interaction Diagrams). The following is a comprehensive list of all the use cases.

- Start FD
  This use case allows the user to start the FD program.
- New project
  This use case allows the user to create a new project from scratch.

- Load an FU hierarchy

  This use case allows the user to load a SEED-Pro FU hierarchy in the main window.

- Exit FD

  This use case allows the user to terminate the FD program.

- Select criteria

  This use case allows the user to select a subset among the four functional criteria, including functional adjacency, thermal, acoustic, and daylight requirements.

- Edit adjacency matrix

  This use case allows the user to modify the adjacency values in the adjacency matrix.

- Edit acoustic table

  This use case allows the user to modify the EEL and PSL values in the acoustic table.

- Edit temperature table

  This use case allows the user to modify the temperature values of FUs in the temperature table.

- Edit daylight requirements

  This use case allows the user to modify the daylight requirements in the daylight table.

- Edit floors

  This use case allows the user to modify number of floors, each floor's area, and allowed area tolerance for each floor.

- Select tree / floors representation

  This use case allows the user to select either a tree structure or a floors' section drawing to represent a grouping result.

- View / hide process of algorithm

  This use case allows the user to view either the process of an algorithm, or the grouping result generated by the algorithm directly in automatic generation.

- View requirements of a tree node or a room

  This use case allows the user to view detailed requirements for a selected FU.

- Start stacking

  This use case allows the user to run the stacking algorithm.

- Start zoning

  This use case allows the user to run the zoning algorithm.

- Attach a tree node as a child of another tree node

  This use case allows the user to move an FU from its current location to another floor / zone in the tree view.

- Move a room FU to another floor

  This use case allows the user to move a room from its current floor location to another floor in the floor view.

- Collapse / expand a tree node

  This use case allows the user to hide / show the children nodes of a selected tree node.

4.3.2 Interaction Diagrams and Implementation

Apart from the graphical representation of the interface during the design stage, object schemata have also been developed. Interaction diagrams in the OOSE method are utilized to show the use cases in terms of object interaction, etc. In the object schemata, the user, each relevant interface or domain element, and every algorithm is taken as an object. The interaction diagram describes a sequence of actions performed on these objects which completes the execution of a corresponding use case. The detailed interaction diagrams are shown in Appendix D: User interface interaction diagrams.

Based on the interaction diagrams, the interface is implemented in an object-oriented environment. The ET++ framework is used to generate the FD interface. Below are sample interface snapshots that illustrate the “look and feel” of the FD system.

4.3.2.1 Main window

This is the first window that appears on the screen when the user starts the program (Figure 4.1). There are three portions in this window. The top portion is a menu bar containing menu options. The middle portion is the workspace. When a project is loaded, it is displayed in either a tree structure or a floor section form. The lower portion is an information bar showing what is going on in the system.

A principle in FD interface design is to minimize menu operations. The menu bar only contains three basic types of operations listed as follows:

- “FU hierarchy”: This menu contains both input and output functions. As both input and output are represented in the form of an FU hierarchy, the title of this menu is called “FU hierarchy”. Functions on this menu include creating, loading, and saving a project, and exiting the program.
- “Group”: Functions on this menu include editing floor areas, editing design requirements, and starting stacking or zoning algorithms.
- “Preferences”: Functions on this menu include showing / hiding algorithmic operations, and selecting tree view or floor view representations.

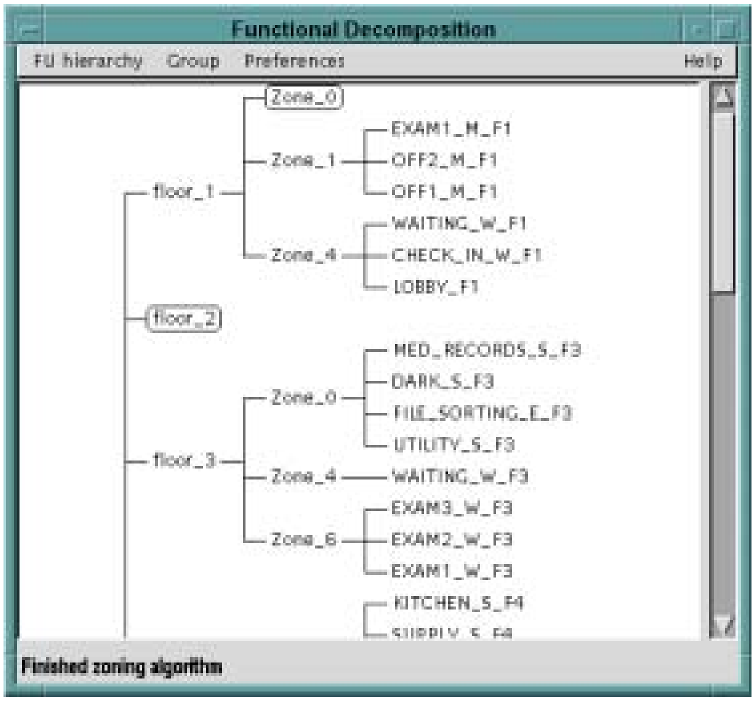

Figure 4.1 Main window

4.3.2.2 Design requirements window

When the user selects “edit design requirements” under the “group” menu, a separate “Design Requirements Window” will be opened (Figure 4.2).

Figure 4.2 Design requirements window

This window contains four document folder-like bars. Each bar, when clicked on, will either bring up a window or expand into one, and display detailed design requirements. For instance, the "Adjacency Properties" bar, when clicked on, will bring up a window showing an adjacency matrix and allowing the user to modify the matrix. For another example, the "Thermal Properties" bar, when clicked on, will bring up a window showing a temperature table and allowing the user to modify the table.

4.3.2.3 Adjacency requirements window

The "Adjacency Properties" bar, when clicked on, will bring up the "Adjacency Requirements Window" showing an adjacency matrix (Figure 4.3). This matrix portrays adjacency relations between all pairs of FUs in a building. When the number of FUs is large, the matrix size will get large and only a portion of it can be displayed. But the user can move the scrollbar from left to right, or up and down to view and modify the whole matrix.

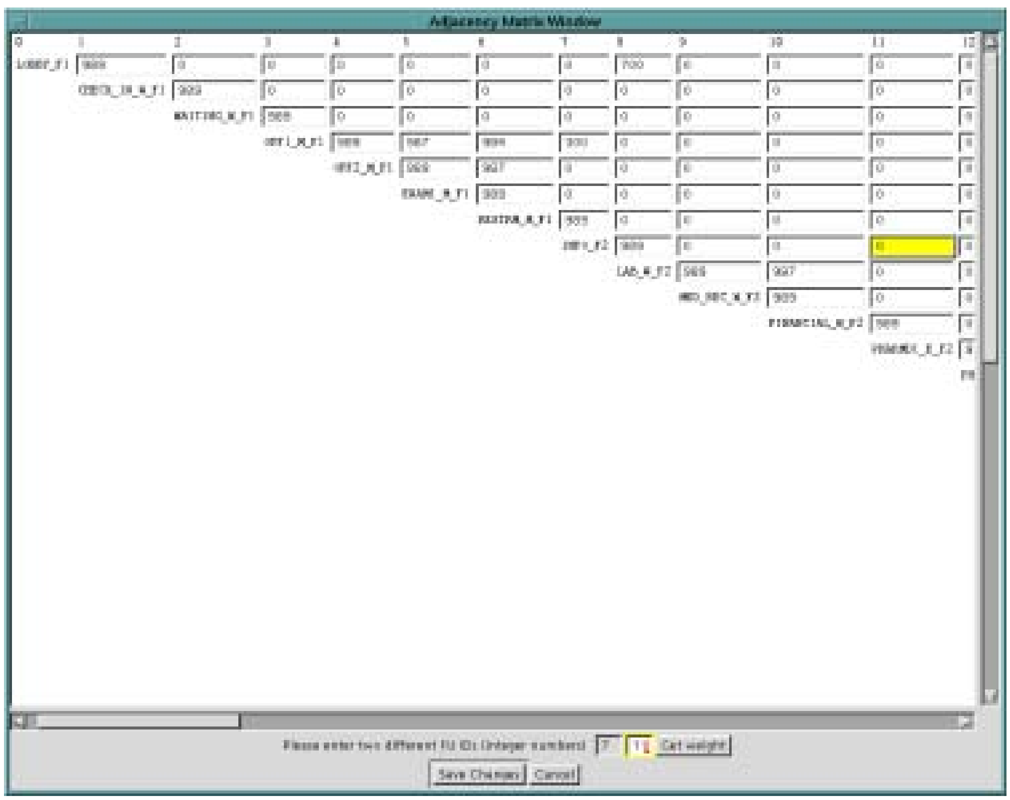

Figure 4.3 Adjacency matrix window

Only the upper-right portion of the window is used to display the matrix. Its diagonally symmetric portion (the lower-left portion) is blank, as the upper-right portion is sufficient to portray the adjacency relations between all pairs of FUs. The ids of FUs are displayed in the first row. The names of FUs are displayed in the diagonal direction. The position of an FU name is determined as such that the row and column numbers for that position are both the same as the id of the FU. For instance, the id of the FU LOBBY_F1 is zero, therefore, the name "LOBBY_F1" is displayed in the position of row zero and column zero.

In order to facilitate the search for a connection between a pair of FUs in a large matrix, searching assistance is provided to highlight a corresponding cell when the user types in two FUs' ids.

4.3.2.4 Temperature requirements window

The "Thermal Properties" bar, when clicked on, will be expanded into a window showing a temperature table (Figure 4.4). This table displays set-point temperatures of all of the FUs in a building. When the number of FUs is large, the table size will get large and only a portion of it can be displayed. But the user can move the scrollbar from left to right, or up and down to view and modify the whole table.

Figure 4.4 Temperature editor window

4.3.2.5 Acoustic requirements window

The "Acoustic Properties" bar, when clicked on, will be expanded into a window showing an acoustic table (Figure 4.5). This table displays EEL and PSL values of all of the FUs in a building. When the number of FUs is large, the table size will get large and only a portion of it can be displayed. But the user can move the scrollbar from left to right, or up and down to view and modify the whole table.

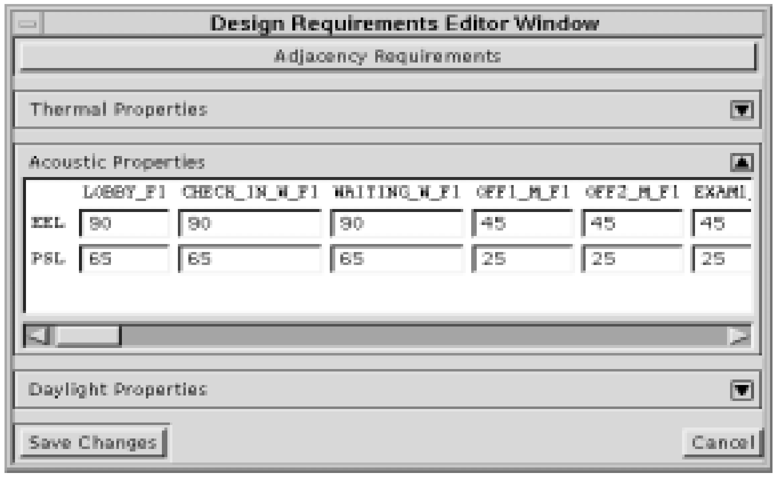

Figure 4.5 Acoustic editor window

4.3.2.6 Tree view

This is first of the two options FD provides in displaying a building organization in the main window (Figure 4.1). The root of the tree is the building. Its direct children nodes are floors. The direct children nodes of the floors are zones; and the direct children nodes of the zones are FUs. Zones may contain, besides FUs, sub-zones.

When a new project is started and the user enters the total number of FUs in the building, the FUs are created using default values provided by the system. A tree view with a bin floor as a child node of the building will be constructed. All of the space FUs are temporarily assigned to the bin floor.

When a project is loaded, the tree view will be constructed according to the requirements already existent in the project files.

When the decomposition algorithm is running and the user chooses to view the algorithmic process, the FUs are seen as moving from one place to another, showing the user each step in the algorithmic process. The final picture will be a full-fledged tree with added floors and zones, or possibly nested zones. The user can then modify the design conveniently by "dragging and dropping" an FU.

When the user wants to focus on a particular portion in the tree view, he / she has the option of collapsing other portion(s) of the tree view by pressing the middle mouse button on the parent node of that portion.

4.3.2.7 Floor view

This is the second of the two options FD provides in displaying a building structure in the main window (Figure 4.6). The graphically displayed floors with FUs on each floor are shown as stacked floors filled with rooms in their proportionate dimensions.

When a new project is started and the user enters the total number of FUs in the building, the floors and FUs are created using default values provided by the system. A floor section view representing a default number of floors with their default areas and an additional bin floor will be constructed. All the FUs will be temporarily assigned to the bin floor for distribution to the other floors.

When an existing project is loaded, the floor section view will be constructed according to the requirements already existent in the project files.

When the stacking algorithm runs, the FUs will be distributed to appropriate floors. When the zoning algorithm runs, zones will be constructed with the FUs on each floor distributed to appropriate zones. When the user selects the "show" option, the FUs are seen as moving from one floor or zone to another when a decomposition algorithm runs. The user can then modify the design by selecting a room and moving it from its current floor or zone to another.

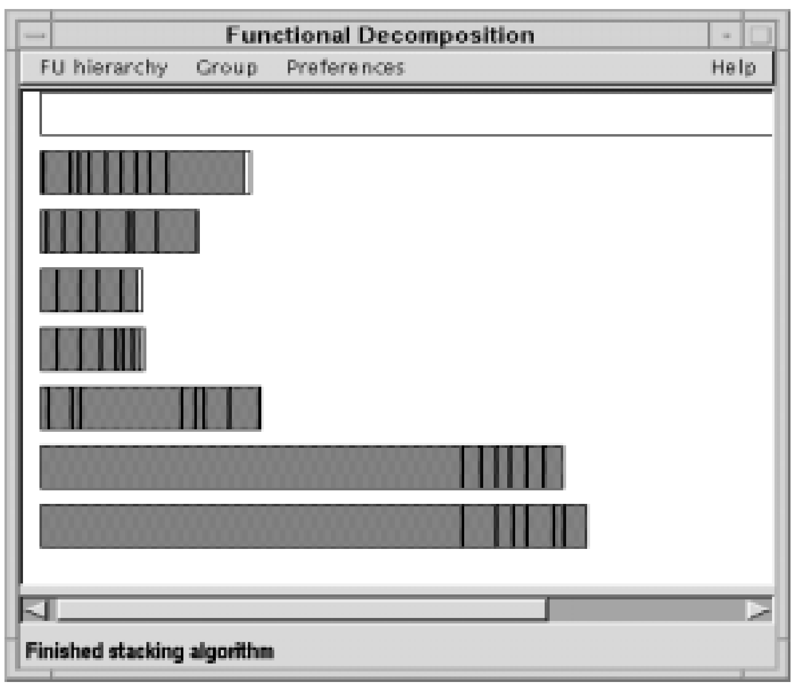

Figure 4.6 Floor view in main window

4.3.2.8 Floor editor window

This window allows the user to enter a particular number of floors, each floor's area, and the allowed area tolerance for each floor (Figure 4.7). If the user changes the number of floors, each floor's area can be adjusted automatically. The system provides two options for the user: either to change the total area of the floors, or to divide the available area among the remaining floors when the user modifies a floor area.

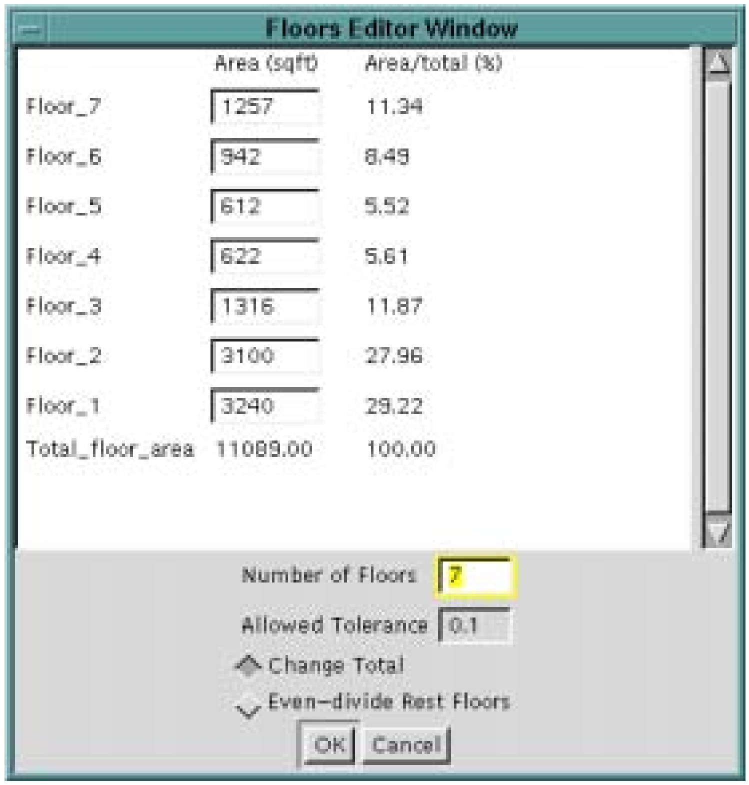

Figure 4.7 Floor editor window

4.3.2.9 FU window

The FU window is opened when the user double-clicks on a tree node FU in the tree view or a room FU in the floor view (Figure 4.8).

The window displays detailed information on an FU, including its id, name, size constraints, adjacent FUs, set-point temperatures, and acoustic and daylight requirements. This information is read-only.

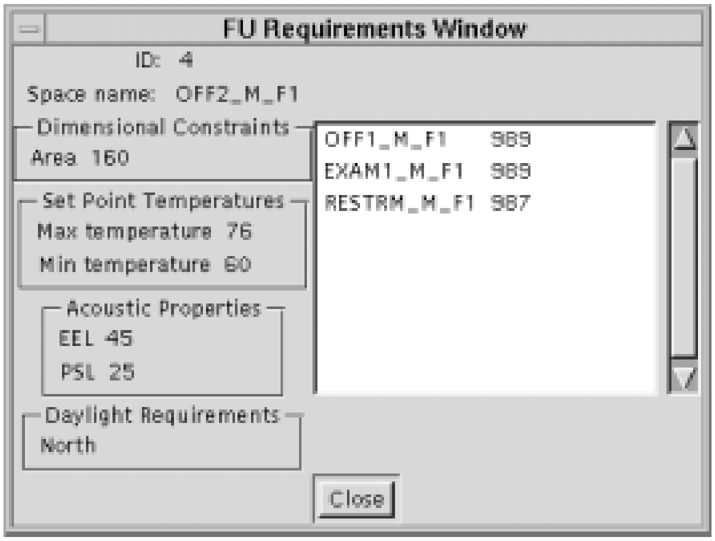

Figure 4.8 FU window

4.3.2.10 Stacking criteria window

When the user selects "start stacking" command on the "group" menu, this window will be opened (Figure 4.9). It allows the user to select a set of design criteria for stacking and define their relative importance by entering weights in the corresponding fields.

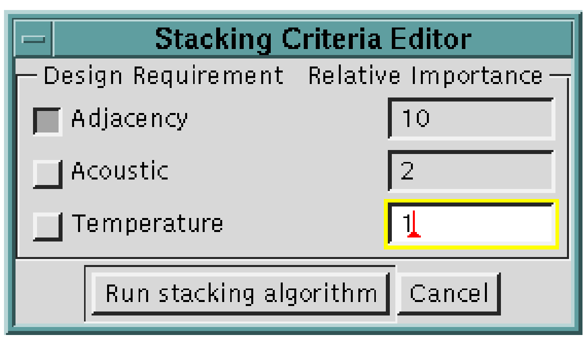

Figure 4.9 Stacking criteria window

4.3.2.11 Zoning criteria window

When the user selects "start zoning" command on the "group" menu, this window will be opened (Figure 4.10). It allows the user to select a set of design criteria for zoning and define their relative importance by entering weights in the corresponding fields. In addition, it allows the user to specify a threshold weight for zoning.

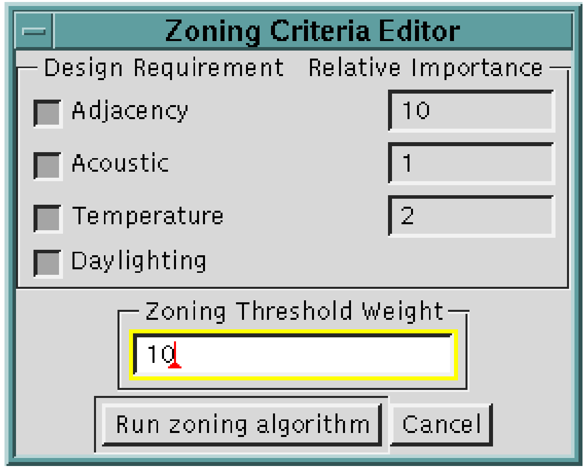

Figure 4.10 Zoning criteria window

5 Algorithm Testing

5.1 Goal

The goal of testing the stacking and zoning algorithms is to evaluate their performance in the stacking and zoning design of real buildings. The selection of test cases, therefore, follows the general principle that they should be typical enough to represent different building types and large enough to adequately test the algorithms.

Specifically four principles are followed in selecting cases:

- The sizes of the buildings should be large enough. A building with over 100 FUs is a good size.
- The buildings should be “tall” enough. They should have a moderate number of floors or more.
- The types of buildings should be diverse and cover a few major building types.
- The buildings should include both existing ones and those that have not yet been designed.

5.2 Procedure

Due to the sizes of the selected projects, it is impossible to enter all the data and requirements manually. The major test procedure is automated. An FU record file contains the size constraints for each FU and required distances between the FUs. A parser reads this file and generates the following:

- a base file with the number of FUs in the building, and the FUs’ names and size constraints;
- an adjacency matrix file that can be used to formulate an adjacency graph.

Besides the FU record file, the user has to create the following input files that contain design requirements. These input files are:

- floor file with the number of floors, and area and allowed area tolerance for each floor;
- temperature file with required high and low temperatures for each FU;
- acoustic file with required EEL and PSL for each FU;
- daylight file showing which FUs need daylight;
- pre-assignment file showing which FUs should be fixed to which floors.

For FUs with special requirements, special methods are used. For instance, for FUs that extend through multiple floors, multiple FUs are created and pre-assigned to the corresponding consecutive floors. Taking a theater as an example, if it is of three-story height, three theater FUs should be created and assigned to floor one, floor two, and floor three, respectively.

The user may choose a subset of design criteria among the four: adjacency, temperature, acoustic, and daylight, and may define their relative importance. The algorithms then run to generate a solution. Before the zoning algorithm runs, however, a threshold weight must be defined so that FUs with stronger relations are grouped and FUs with weaker relations are separated.

For a project with an unknown number of floors and floor areas, the user can experiment with different number of floors and floor areas to generate different stacking and zoning alternatives. It is recommended that a large area tolerance of floors be used in such cases so that FUs with strong relations can be grouped under flexible area constraints.

5.3 Four Projects

The test cases (Table 5.1) are Brooklyn Jail in New York, Kaiser office building in Oakland, California, Falk Clinic in Pittsburgh, and Center for the Arts at CMU in Pittsburgh. They are two high-rises and two multiple-floor buildings. The number of FUs in each building ranges from 86 to 351. The floor numbers range from 4 to 28. The respective buildings represent a jail, an office building, a hospital, and a theater. Therefore the selected cases cover major building types and are of medium to large size.

Table 5.1 Four test cases

Project	Brooklyn Jail	Kaiser high-rise	Falk Clinic	Center for the Arts
Type	Jail	Office	Hospital	Theater
# of Floors	9	28	7	4/5
# of FUs	139	131	351	86
% correctly assigned	90.6	93.1	77.2	NA

5.3.1 Reverse Engineering of Existing Buildings

Three existing buildings are selected to test if the decomposition algorithms can reproduce the existing stacking and zoning designs. For these existing buildings, their floor plans are obtained. The area of each FU is measured. The adjacency relations between FUs are set up relative to their physical distances. The number of floors and each floor's area are the same as the actual figures.

The acoustic and temperature values of the FUs are entered according to their actual values, or to the best knowledge of the author when certain data are not available. Daylight requirements are established for FUs with crucial daylight requirements. Pre-assignment of FUs to floors are entered for FUs that must be located on certain floors.

It is assumed that the stacking and zoning designs of the existing buildings are ideal; therefore given the input of existing adjacency relations between the FUs in a building, the stacking and zoning algorithms, if well designed, should be able to reproduce the existing floors and zones.

For each of the three projects, a stacking result based on a single adjacency criterion is presented. Section 5.3.1.3 gives a detailed analysis of stacking results according to multiple design criteria using the Falk Clinic project as an example.

Since the clustering algorithm can always produce a predicted[1] grouping result by grouping FUs with stronger relations than an adequately specified threshold weight, zoning analysis is omitted from discussion. For a detailed zoning output, see Appendix C: Test Cases.

The following section gives a description for each project and test results. For detailed outputs, also refer to Appendix C: Test Cases.

5.3.1.1 Brooklyn Jail

This is a nine-story building. Adjacency relations are defined:

- combinatorially between all pairs of FUs within the same zone, and
- between a pair of FUs within two adjacent zones.

The second relation is important as it groups adjacent zones on the same floor.

Table 5.2 Stacking analysis of Brooklyn Jail

Floor number		1	2	3	4	5	6	7	8	9	**Total**	Total cut size
Existing design	# of FUs	19	15	15	15	15	15	15	15	15	**139**	0
Solution devised by stacking algorithm	# of FUs matching existing placement	12	15	15	15	15	15	15	15	9	**126**	178
	% of FUs matching existing placement	63.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	60.0	**90.6**	

The stacking result in Table 5.2 shows that the algorithm devises excellent results from floor two up to floor eight. However it does not produce an ideal outcome for both the ground floor and the top floor.

It is speculated that this may be due to the initial partition strategy used in the algorithm. At the beginning, all the FUs are temporarily assigned to the top floor. FUs are then moved from this floor to the lower ones. This process terminates when all the lower floors are filled. Those “leftover” FUs remain on the top floor. In this process, the lower floors have higher priority in grouping. Those FUs that are left out in the grouping process will stay in the top floor.

---

1. An algorithm that produces a “predicted” result occurs when the placements (allocations) match those found in the drawings of buildings that have been reverse-engineered.

5.3.1.2 Kaiser high-rise office building

This is a 28-story office building. Adjacency relations are defined:

- combinatorially between all pairs of FUs within the same zone on floor one with a relatively large number of FUs,
- combinatorially between all pairs of FUs on the same floor from floor 2 to floor 28 with a relatively small number of FUs, and
- between a pair of FUs within two adjacent zones.

Table 5.3 Stacking analysis of Kaiser high-rise

<table>
<tr><td colspan="2">Floor number</td><td>1</td><td>2</td><td>3-27</td><td>28</td><td>Total</td><td>Total cut size</td></tr>
<tr><td>Existing design</td><td># of FUs</td><td>18</td><td>9</td><td>4</td><td>4</td><td>131</td><td>0</td></tr>
<tr><td rowspan="2">Solution devised by stacking algorithm</td><td># of FUs matching existing placement</td><td>12</td><td>9</td><td>4</td><td>1</td><td>122</td><td rowspan="2">270</td></tr>
<tr><td>% of FUs matching existing placement</td><td>66.7</td><td>100.0</td><td>100.0</td><td>25.0</td><td>93.1</td></tr>
</table>

The stacking result in Table 5.3 shows that the algorithm devises excellent results from floor two up to floor 27. However it does not produce an ideal outcome for both the ground floor and the top floor. A speculative reason for this has been given in Section 5.3.1.1.

5.3.1.3 Falk Clinic

This is a seven-story, 351 room (FU) hospital building. On each floor, there are distinct architecture zones toward each direction: north, south, east, and west. Adjacency relations are set up in such a way that FUs within the same zone have adjacency relations. The strength of an adjacency relation is relative to the physical distance between the two FUs. An adjacency relation is also defined between a pair of FUs within two adjacent zones.

The stacking results in Table 5.4 show that the stacking algorithm, although producing excellent results for floor one through floor four (100% accuracy), does not produce results that are as good for floors close to the top (around 65% accuracy). The result for the top floor is especially poor (7.3% accuracy). A speculative reason for this has been given in Section 5.3.1.1.

Table 5.4 Stacking analysis of Falk Clinic according to single criterion

<table>
<tr><th colspan="2">Floor number</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>Total</th><th>Total cut size</th></tr>
<tr><td>Existing design</td><td># of FUs</td><td>45</td><td>39</td><td>53</td><td>52</td><td>61</td><td>60</td><td>41</td><td>351</td><td>0</td></tr>
<tr><td rowspan="2">Solution devised by stacking algorithm</td><td># of FUs matching existing placement</td><td>45</td><td>39</td><td>53</td><td>52</td><td>40</td><td>39</td><td>3</td><td>271</td><td rowspan="2">1381</td></tr>
<tr><td>% of FUs matching existing placement</td><td>100.0</td><td>100.0</td><td>100.0</td><td>100.0</td><td>65.6</td><td>65.0</td><td>7.3</td><td>77.2</td></tr>
</table>

The above provides an analysis of single criterion stacking. Using the Falk Clinic building as an example, stacking results according to multiple criteria are also analyzed in the following section. Table 5.5 shows the stacking results based on adjacency, thermal, and acoustic requirements. The input contains:

- adjacency relations;
- minimum and maximum temperatures of each FU;
- EEL and PSL of each FU;
- relative weight (importance) for each design criterion: 10 for adjacency, 2 for acoustic and 1 for thermal requirements.

In this case of multiple criteria stacking, adjacency weighs more than the other constraints (10 relative to 2 and 1); therefore, it is still a dominant factor and the stacking result is similar to the one based on a single adjacency criterion, as shown in Appendix C. However, in this case, some FUs with similar acoustic or thermal requirements are grouped into the same floor although they do not have adjacency relations. This may result in FU placements that violate adjacency constraints. For instance, all the personnel rooms and utility type of spaces are assigned to floor six since they share the same EEL, PSL, minimum and maximum temperatures, although their adjacency relations may require them to be assigned to the other floors.

Table 5.5 Stacking analysis of Falk Clinic according to multiple criteria

<table>
<tr><th colspan="2">Floor number</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>Total</th><th>Total cut size</th></tr>
<tr><td>Existing design</td><td># of FUs</td><td>45</td><td>39</td><td>53</td><td>52</td><td>61</td><td>60</td><td>41</td><td>351</td><td>21826</td></tr>
<tr><td rowspan="2">Solution devised by stacking algorithm</td><td># of FUs matching existing placement</td><td>44</td><td>23</td><td>43</td><td>16</td><td>37</td><td>18</td><td>3</td><td>184</td><td rowspan="2">34502</td></tr>
<tr><td>% of FUs matching existing placement</td><td>97.8</td><td>59.0</td><td>81.1</td><td>30.8</td><td>60.7</td><td>30.0</td><td>7.3</td><td>52.4</td></tr>
</table>

This shows that grouping based on multiple constraints is an adjustment of single criterion (adjacency) stacking with added constraints.

When the relative weights between different criteria change, the stacking result will change accordingly. A sensitivity analysis illustrates how changes in relative weights affect grouping results. Three types of analysis are conducted that are outlined as follows:

- to compare with adjacency-based stacking, increment the weight of acoustic requirements from one to ten, and examine how changes in acoustic weight affect floor grouping (Table 5.6 and Figure 5.1);
- to compare with adjacency-based stacking, increment the weight of temperature requirements from one to ten, and examine how changes in temperature weight affect floor grouping (Table 5.7 and Figure 5.2);
- to compare with adjacency-based stacking, increment the weights of both acoustic and temperature requirements from one to ten, and examine how changes in acoustic and temperature weights affect floor grouping (Table 5.8 and Figure 5.3).

Table 5.6 shows how many FUs are placed onto floors in violation of adjacency relations when acoustic requirements are taken into consideration, in addition to adjacency relations. A “misplacement” of an FU is a case when acoustic requirements weigh enough to override certain adjacency relations. Figure 5.1 intuitively illustrates the percentage of FUs misplaced in the building due to acoustic considerations.

Table 5.6 Percentage of FU placements violating adjacencies due to acoustic considerations (adjacency_weight = 10, temperature_weight = 0)

Acoustic weight	1	2	3	4	5	6	7	8	9	10
Floor 1	0	0	0	9	15	16	26	25	24	27
Floor 2	0	0	0	26	5	15	21	12	12	21
Floor 3	0	0	10	10	19	21	14	18	18	16
Floor 4	0	0	0	18	20	21	19	49	49	19
Floor 5	0	1	-1	-5	8	7	11	14	14	10
Floor 6	0	1	7	12	26	19	17	16	10	12
Floor 7	0	-1	-1	0	1	4	3	2	3	0
Total number	0	1	15	70	94	103	111	136	130	105
Total %	**0.0**	**0.0**	**4.3**	**19.9**	**26.8**	**29.3**	**31.6**	**38.7**	**37.0**	**29.9**

As shown in Figure 5.1, the effect of acoustic requirements grows with the weight of acoustic requirements when the weight is below 8. It grows most rapidly when the weight is between 3 and 5, relative to 10, the weight of adjacency requirements. Acoustic requirements override adjacency relations most strongly when the weight is 8; at such point 38.7% of FUs are placed in violation of adjacency relations, whereas such a placement favors acoustic relations. After the acoustic weight reaches 8, the impact of acoustic relations decreases as the acoustic weight continues to increase.

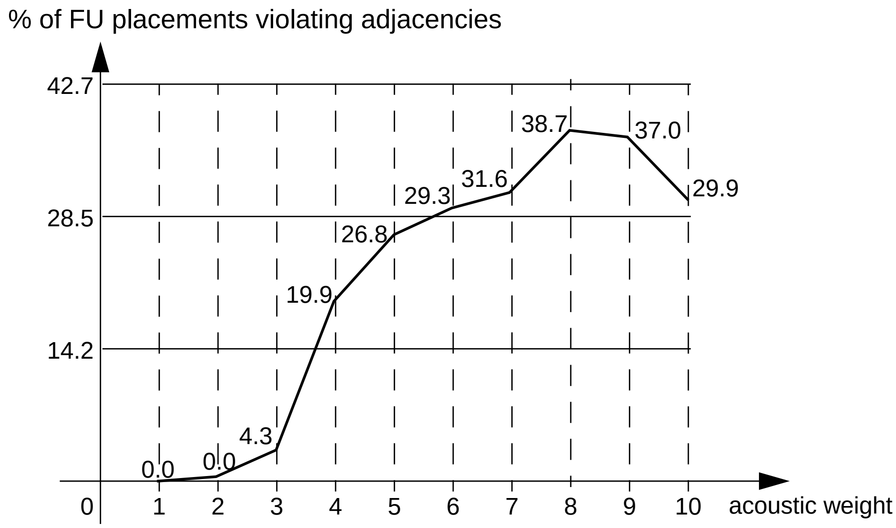

Figure 5.1 Percentage of FU placements violating adjacencies due to acoustic considerations (adjacency_weight = 10, temperature_weight = 0)

Thus the impact of acoustic relations does not always increase with the relative weight of acoustic requirements. This is because when a significant percentage (38.7%) of FUs' placements violate their adjacency relations, the adjacency-based stacking pattern is non-existent, at which point it becomes unimportant whether more or fewer FUs' placements violate their adjacency requirements when the acoustic weight continues to increase.

Table 5.7 Percentage of FU placements violating adjacencies due to temperature considerations (adjacency_weight = 10, acoustic_weight = 0)

Acoustic weight	1	2	3	4	5	6	7	8	9	10
Floor 1	2	25	31	34	47	47				47
Floor 2	18	23	19	20	22	17				21
Floor 3	6	18	33	25	35	37				37
Floor 4	30	40	37	45	45	42				56
Floor 5	10	10	29	29	27	31		No feasible solution due to area tolerance		30
Floor 6	22	19	22	23	19	18				20
Floor 7	-1	-1	-1	-1	-1	-1				-1
Total number	87	134	170	175	194	191				210
Total %	**24.8**	**38.2**	**48.4**	**49.9**	**55.3**	**54.4**				**59.8**

Table 5.7 provides how many FUs are misplaced onto floors in violation of adjacency relations when temperature requirements are taken into consideration, in addition to adjacency relations. A "misplacement" of an FU occurs when temperature requirements weigh enough to

override certain adjacency relations. Figure 5.2 intuitively illustrates the percentage of FUs misplaced in the building due to temperature considerations.

As shown in Figure 5.2, the effect of temperature requirements grows with the weight of temperature requirements when the weight is below 5. It grows most rapidly when the weight is between 1 and 3, relative to 10, the weight of adjacency requirements. After the temperature weight reaches 5, the impact of temperature relations decreases with the temperature weight at a certain point. The reason for this is similar to the explanation of the decrease of impact of acoustic relations when the acoustic weight is over a certain limit. When a significant percentage (55.3%) of FUs' placements violate their adjacency relations, the adjacency-based stacking pattern is non-existent, at which point it becomes unimportant whether more or fewer FUs' placements violate their adjacency requirements when the temperature weight continues to increase.

Quite different from the acoustic analysis in Figure 5.1, temperature requirements override adjacency requirements strongly even when the temperature weight is small. For instance, when the weight is 1, 24.8% of FUs are placed in violation of their adjacency relations due to their temperature requirements. This is because FUs sharing a same required temperature also share a maximum temperature weight. When there are a significant percentage of FUs sharing a same required temperature, a dense graph is produced where there are many edges with this maximum temperature weight. A large number of such edges, although each bearing a small weight, can significantly affect the grouping result.

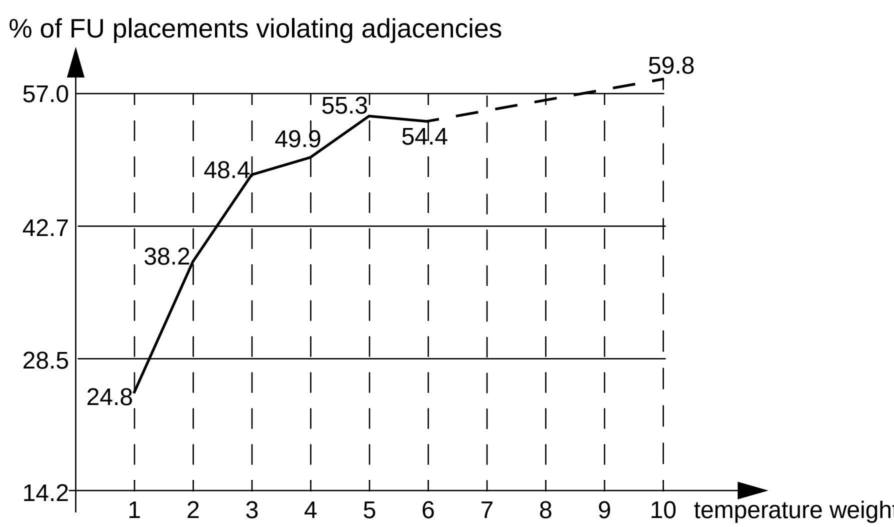

Figure 5.2 Percentage of FU placements violating adjacencies due to temperature considerations (adjacency_weight = 10, acoustic_weight = 0)

On the other hand, in the acoustic case, although many FUs share a same required EEL and PSL, they usually have different dimensional constraints, which result in different required acoustic decouplings between each other. When a few pairs of FUs requiring minimum decoupling have a maximum weight, the remaining FUs have smaller weights between each other (many weights may be rounded to 0). Thus the acoustic graph is a sparse one and affects grouping results less significantly as compared with temperature requirements when their relative weights are the same.

Table 5.8 is an analysis of how acoustic and temperature requirements, when considered jointly with adjacency requirements, affect the grouping result.

As shown in Figure 5.3, the effect of acoustic and temperature requirements grows with the weight of acoustic requirements when the weight is below 6, and between 7 and 9 (when temperature weight is constant). However the impact of acoustic requirements on grouping decreases when the acoustic weight is between 6 and 7, and 9 and 10.

Table 5.8 Percentage of FU placements violating adjacencies due to acoustic and temperature considerations (adjacency_weight = 10, temperature_weight = 1)

Acoustic weight	1	2	3	4	5	6	7	8	9	10
Floor 1	2	13	6	31	30	29	32	31	32	23
Floor 2	18	10	13	17	21	24	7	32	18	16
Floor 3	6	16	26	21	26	27	24	29	45	32
Floor 4	30	13	35	31	36	34	20	27	41	46
Floor 5	10	17	15	21	30	32	15	27	34	16
Floor 6	22	25	19	19	14	14	16	17	17	17
Floor 7	-1	-1	0	3	1	1	5	2	5	6
Total number	87	93	114	143	158	161	119	165	192	156
Total %	**24.8**	**26.5**	**32.5**	**40.7**	**45.0**	**45.9**	**33.9**	**47.0**	**54.7**	**44.4**

This shows that at certain points, the percentage of misplaced FUs does not increase with the weight of design requirements other than adjacency. A closer examination of the data sheds light on a possible explanation for this. When acoustic weight reaches 6, the number of misplaced FUs is 161, comprising 45.9%, almost half of the total number of FUs in the building. When acoustic weight reaches 9, the number of misplaced FUs is 192, comprising 54.7%, over half of the total number of FUs in the building. When there are a significant number (approximately one half) of FU placements that violate adjacencies due to acoustic or thermal considerations, the adjacency-based grouping pattern is almost un-identifiable. Thus, further increasing the acoustic or thermal weight will not always result in additional FUs whose placements violate adjacency relations.

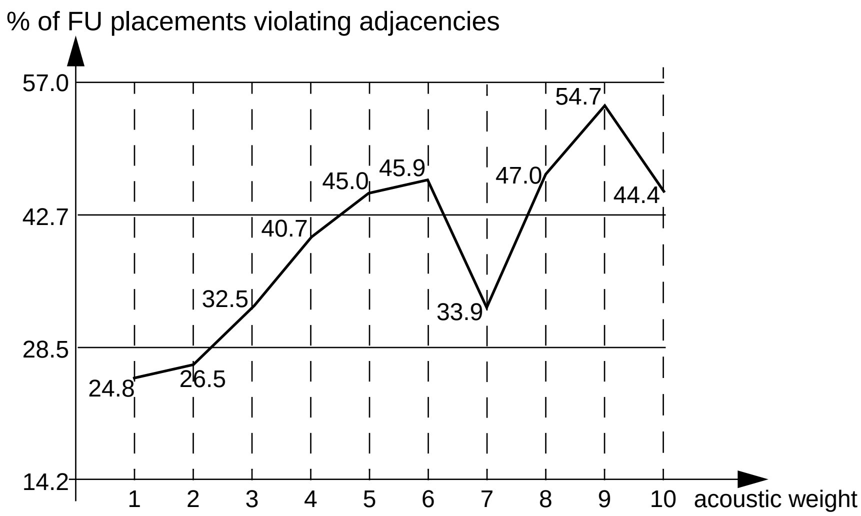

Figure 5.3 Percentage of FU placements violating adjacencies due to acoustic and temperature considerations (adjacency_weight = 10, temperature_weight = 1)

In summary, while it seems unreasonable that all the auxiliary spaces are assigned to the same floor because they share certain acoustic or thermal properties, with smaller relative weights defined for these constraints, a unified stacking result may achieve desired functional relations, saving both construction cost and energy use.

By solving the decomposition problem according to multiple design criteria, the system enables the user to explicate trade-offs among different criteria. The user can experiment with different parameters in order to arrive at a design that is optimal regarding multiple design requirements.

5.3.2 Designing Building from Scratch

An architecture program containing a comprehensive set of design requirements is selected for the Center for the Arts at CMU. This case is used to test the performance of the algorithms in designing new buildings. Alternative stacking and zoning designs devised by the algorithms are reviewed and evaluated by the project architect.

Since this project has an unknown number of floors and floor areas, different number of floors and floor areas have been experimented with to generate different stacking and zoning alternatives. A large area tolerance of floors is used so that FUs with strong relations can be grouped under flexible area constraints.

Different from the other three test cases, the Center for the Arts is not an existing building. The number of floors and floor areas, instead of being decided upon up front, are to be determined by the stacking procedure.

Also special to this case is the fact that it has FUs of multiple-floor height. For instance, the theater is of three-story height; therefore, the same theater FU is created for floor one, floor two and floor three, and is pre-assigned to floor one, floor two, and floor three, respectively. As shown in Appendix C, two alternatives are produced. One is a four-story building including a basement; the other is a five-story building including a basement. These alternatives are examined by and satisfy the project architect.

Designing a building from the beginning is different from reproducing an existing one. Weights between FUs are set relative to the importance of their adjacency relations, instead of according to their physical distances. Number of floors and floor areas are unknown, so that experimenting with different floor configurations helps the designer to achieve viable design alternatives. This is where the role of FD as an “active design assistant” comes into play.

5.4 Comparison with Benchmark: SABA

The performance of the FD stacking algorithm is evaluated against that of SABA, a state-of-the-art benchmark. Three projects are selected as test cases. They are the Brooklyn Jail, Kaiser office building, and Falk Clinic. There are special considerations for SABA:

- Since SABA does not handle zoning, only stacking results are evaluated.
- SABA only handles adjacency requirements, therefore, only adjacency-based stacking results are evaluated.
- Since SABA can only handle a little over 100 FUs (this number varies from project to project), the number of FUs is reduced for SABA for each project.

Table 5.9 Comparison with SABA

Project	Number of FUs	Parameter for comparison	**SABA**	**FD**
Brooklyn	139	Number of FUs can handle	128	139
		Accuracy(%)	**71**	**90.6**
Kaiser	131	Number of FUs can handle	127	131
		Accuracy(%)	**77**	**93.1**
Falk Clinic	351	Number of FUs can handle	109	351
		Accuracy(%)	Not tested	**77.2**

The size of the Brooklyn project (139 FUs) is greater than the one that SABA can handle; therefore, the size of the project is reduced to 128 FUs. As shown in Appendix C, SABA can reproduce 91 / 128 (71%) of the stacking configuration.

For the Kaiser project, SABA cannot handle the 131 FUs; therefore the top floor (the 28th floor) including the FUs on it, is removed from this project. The reduced size is 127 FUs and 27 floors. As shown in Appendix C, SABA can reproduce 98 / 127 (77%) of the stacking configuration for this building.

For the Falk Clinic, SABA cannot handle the 351 FUs. The maximum size SABA can handle is 109 FUs, less than 1/3 of the actual project size. For such a drastically reduced size, the test results would be meaningless. Therefore, this project is not used in testing and evaluation.

As shown in Table 5.9, FD can reproduce existing buildings more precisely than SABA (90.6% to 71%, and 93.1% to 77%, respectively) when SABA is using a reduced building size.

6 Conclusion

6.1 Contributions

With regard to the research objectives as stated in section 1.3, this thesis makes the following contributions pertaining to the development of active assistance for stacking and zoning:

- ***Development of an active design support for stacking and zoning.***

  The development of this computational assistance provides the user with a reasonable starting point in design rather than the user making all the decisions. It enables the user to have the flexibility to handle stacking and zoning according to different constraints and requirements. It allows the user to have extensive interaction with the system at all levels of design. Furthermore, the process of the automation can be shown in an intuitive way, making it both transparent and easily modifiable by the user.

- ***Enhance computerized stacking and zoning by improving the automation process itself.***

  The improvement in the automation process is achieved by modeling building design problems in an efficient computational model with regard to various domain requirements, and adapting state-of-the-art VLSI graph-partitioning algorithms. The Kernighan-Lin and Fiduccia-Mattheyses-based stacking algorithm takes only linear time and generates an excellent partition result.

- ***Increase the range of supported design parameters, including functional adjacencies, thermal, acoustic, and daylight requirements.***

  None of the current "mono-mode" programs is able to handle design requirements other than adjacency, whereas these other design requirements may have an important impact on design. Failing to address several major design criteria makes these programs impractical. By taking into consideration multiple design requirements, FD is flexible enough to handle the primary aspects of design. It facilitates the exploration of relationships between different requirements and design decisions. It also provides help for the user to explore trade-offs in design.

- ***Integration of FD with a comprehensive design environment, SEED.***

  When integrated with other SEED modules, FD can play an active role in assisting layout design generation. It provides alternative solutions of spatial decomposition to the layout module, so that a large and complex layout design can take place within a smaller, less complex, better-defined context, e.g., within a floor or zone.

  The input to FD comes from SEED-Pro in the form of a flat FU Hierarchy. This matches FD's current representation. Additional work needs to be done to translate SEED-Pro's adjacency relations, thermal, acoustic, and daylight requirements into FD's adjacency matrix, thermal, acoustic and daylight tables.

The output of FD is intended to be communicated to SEED-Layout. SEED-Layout is designed to take an FU Hierarchy as input, which is consistent with FD's FU hierarchy representation. More work is necessary to incorporate required construction costs as part of the output for acoustically decoupling vertically adjacent FUs.

- ***Requirements gathering and modeling through protocol analysis, energy simulation, and problem analysis.***

  Different design problems (such as adjacency relations, thermal and acoustic problems) have different natures and should be dealt with differently. In order to solve the problem of adjacency decomposition, a protocol analysis was conducted. For thermal and acoustic decompositions, an energy simulation and problem analysis were carried out, respectively. Thus FD system design requirements have been gathered and further modeled into a computational representation. This, besides demonstrating explorative work in modeling a real world design problem, has also become the basis for the FD system design.

- ***Set up foundation for future research.***

  Through various research methods used to obtain useful domain parameters, knowledge has been gained not only in related domains, but also in understanding the stacking and zoning process itself. This has set up a foundation for either improving upon the existing decomposition components, or incorporating additional design criteria, such as structural requirements. This is possible because the modular system architecture makes it easy to introduce new elements into the system.

6.2 Future Research Issues

During the development of FD, a proof-of-concept approach was taken to demonstrate the major functionalities. In order to make FD a full-fledged system, more research needs to be completed in the following areas:

- As demonstrated in the test results of the Brooklyn Jail, Kaiser high-rise, and Falk Clinic, the stacking and zoning algorithm does not produce the required stacking design for the top floor and in some cases the first floor. A speculative reason for this is that the initial partition strategy does not favor the top floor (see Section 5.3.1.1). While the reason for this remains unclear, more algorithm research is necessary on identifying the real cause and adopting a corresponding method to improve the performance of the algorithm.

- Vertical relations should be fully considered. Such vertical relations include FUs that ought to be assigned to adjacent floors, FUs that should be located with ease of access by stairs or an elevator. Vertical relations, when fully considered, may help to solve the problem of poor top floor grouping. For instance, when FUs are correctly assigned to a lower floor, the top floor can achieve a correct grouping result when its FUs bear adjacency relations with the lower floor. The FUs that ought to be assigned to the top floor will not be assigned to any other floor if their vertical relations are to be met.

- More architectural design requirements should be considered, such as structure, fire egress, and accessibility.
- The relations between different design criteria need to be further explored. Additional research is necessary to explore both the relative weights within a single criterion and relative weights across different criteria when multiple criteria are considered.
- Graphical tools can be incorporated into FD so that designers can tackle aesthetic or formal issues in the stacking and zoning stage.
- Last, but not least, further research needs to be done in human-computer interaction so that the user can use this tool with more ease and flexibility.

7 Bibliography

Akin, Ö., B. Dave, and S. Pithavadian (1987). *Problem Structuring in Architectural Design*, Carnegie Mellon University, Engineering Design Research Center, Pittsburgh, PA.

Akin, Ö. and R. Sen (1995). *Navigation within a structured search space in layout problems.* Environment and Planning B: Planning and Design, vol. 23, p421-442.

A layout generator HeGel-2 is introduced. It generates layout which satisfies hard constraints and optimizes specified soft constraints. A comparison is made between HeGel-2 and several other state-of-the-art layout generators.

Akin, Ö., Sen, R., Donia, M., and Zhang, Y. (1995). *SEED-Pro: computer-assisted architectural programming in SEED*. J. Arch. Engrg., ASCE, 1(4), p153-161.

Anderson, Richard J. and Joao C. Setubal (1992). *On the Parallel Implementation of Goldberg's Maximum Flow,* Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures, New York, NY, p168-177.

The authors describe an efficient parallel implementation of Goldberg's maximum flow algorithm for a shared-memory multiprocessor. The main technical innovation is a method that allows a "global relabeling" heuristic to be executed concurrently with the main algorithm. This heuristic is essential for good performance in practice.

Ardizzone, E., A. Chella and R. Rizzo (1994). *Color Image Segmentation Based on a Neural Gas Network*, Proceedings of the International Conference on Artificial Natural Networks, Berlin, Germany, vol. 2, p1161-1164.

This paper discusses the use of vector quantization algorithms in object extraction from an image, i.e., in separating meaningful data from an irrelevant background context.

Awerbuch, Baruch and David Peleg (1990). *Sparse Partitions*, Proceedings of the 31st Annual Symposium on Foundations of Computer Science, IEEE Comput. Soc. Press, Los Alamitos, CA, vol. 2, p503-513.

This abstract presents a collection of clustering and decomposition techniques enabling the construction of sparse and locality preserving representations for arbitrary networks. Trade-offs between the sparsity of a cover and the sizes of clusters are discussed.

Ayala-Ramirez, Victor and Rene Jaime-Rivas (1995). *3-D Polyhedral Object Recognition Using Fuzzy Indicators*, 1995 IEEE International Conference on Systems, Man and Cybernetics, New York, NY, vol. 2, p1873-1876.

Fuzzy rules give the chances to identify the objects by using the same methods used by human visual system, that is finding similarities between characteristics that are not very precise. This is especially useful when there are many factors (e.g., lights, shades and etc.)that complicate the exact modelization of the scene.

Baron, Robert J. (1981). *Mechanisms of Human Facial Recognition*, Int J. Man-Machine Studies, vol. 15, p137-178.

Several fundamental processes are implicated: encoding of visual images into neural patterns, detection of simple facial features, size standardization, reduction of the neural patterns in dimensionality, and finally correlation of the resulting sequence of patterns with all visual patterns already stored in memory. Neural networks for carrying out these processes are presented.

Baykan, Can (1991). *Formulating Spatial Layout as a Disjunctive Constraint Satisfaction Problem*, Ph. D. thesis, Department of Architecture, Carnegie Mellon University, Pittsburgh, PA.

This thesis presents a method for formulating spatial constraints for layout design. The constraints for each space include atomic constraints and relational constraints.

Boyer, David G., and Robert R. Cordell (1989). *Symbolic Layout for Rapid Full-Custom Prototyping of High-Speed Telecommunications Chips*, Proceedings of the Twenty-Second Annual Hawaii International Conference on Systems Sciences: Architecture Track, vol 1, p92-101.

This paper gives an introduction to Bellcore's MULGA symbolic design system components, including assembling/placement, simulation and verification. It also elaborates why MULGA greatly increases both a single designer's and a CAD programmer's productivity.

Brown, Donald E. and Clarence Louis Pittard (1993). *Classification Trees with Multi-variate Splits*, p475-477.

This paper discusses the disadvantages of some existing algorithms on classifications when applied to multi-variate problems; and recommends an improved version of the algorithm.

CAFM Works, Inc (1990). *CAFM/STACK/BLOCK*.

This report introduces the computer system called CAFM/STACK/BLOCK.

Cagan, Jonathan (1992). *A Shape Annealing Solution to the Constrained Geometric Knapsack Problem*, Computer Aided Design(UK), vol. 26, no. 10, p763-770.

This paper introduces a technique called **shape annealing** for layout or packing of parts, components and configurations. Shape annealing combines the concepts of the stochastic optimization technique of simulated annealing with shape grammars which specify relations between geometric shapes.

Dahlhaus, E., D. S. Johnson, C. H. Papadimitriou, P. D. Seymour and M. Yannakakis (1992). *The Complexity of Multiway Cuts*, Proceedings of the 24th Annual ACM Symposium on the Theory of Computing, p241-251.

In the Multiway Cut problem, an edge-weighted graph is given with the subset of the vertices called terminals; the required output will be to find a minimum weight set of edges that separates each terminal from all the others. A simple approximation algorithm is described which will guarantee a polynomial time when number k of terminals is fixed

Decision Graphics (1997). *OrgTree Stack and Block,* retrieved November 29, 1997 from the World Wide Web: http://www.autofm.co.uk/prod01.htm.

This page gives an introduction of OrgTree Stack and Block computer program.

Ericsson, K Anders and Herbert Simon (1993). *Protocol Analysis: Verbal Reports as Data,* MIT Press, Cambridge, MA.

Fenves, S. J., Rivard, H., and Gomez, N. (1995). *Conceptual structural design in SEED,* J. Arch. Engrg., ASCE, 1(4), p179-186.

Fiduccia, C. M. and R. M. Mattheyses (1982). *A linear Heuristic for Improving Network Partitions,* Proc. ACM/IEEE Design Automation Conf., p175-181.

This paper introduces a linear time graph partitioning algorithm. To deal with cells of various sizes, the algorithm progresses by moving one cell at a time between the blocks of the partition while maintaining a desired balance based on the sizes of the blocks rather than the number of cells per block. Efficient data structures are used to avoid unnecessary searching for the best cell to move and to minimize unnecessary updating of cells affected by each move.

Flemming, Ulrich and Shengfen Chien (1995). *Schematic layout design in SEED environment*, Journal of Architectural Engineering, ASCE, 1(4): 162-169.

Flemming, Ulrich. and Robert Woodbury (1995). *SEED: Overview*, Journal of Architectural Engineering, ASCE, December 1995, vol. 1, no. 4, p147-152.

Gabow, Harold N. (1994). *Efficient Splitting Off Algorithms for Graphs,* Proceedings of the 26th Annual ACM Symposium on the Theory of Computing, New York, NY, p696-705.

Splitting off ia a powerful tool for proving theorems and developing polynomial-time algorithms on graphs, especially for edge-connectivity problems. This paper presents efficient algorithms for splitting off, leading to efficient algorithms for connectivity problems. The algorithms are improvement of the previous algorithms (based on submodular flow) to find k-edge-connected orientation of undirected graph or multigraph.

Gantz, John (1989). *FM: The CAD Connection,* Computer Graphics World, December, p27-29.

This paper discusses the fact that poor integration with CAD may be slowing the facilities management market.

Gelsey, Andrew (1992). *Modeling and Simulation for Automated Yacht Design,* Rutgers University Technical Report, CAP-TR-16.

This technical report gives a detailed description of the development of a computer tool, Design Associate (DA), used in a yacht design organized by the Computer-Aided Productivity project at Rutgers University. It also shows some sample simulation results in the design of the yacht.

Hagen, Lars and Andrew B. Kahng (1992). *New Spectral Methods for Ratio Cut partitioning and Clustering*, IEEE Transactions on Computer-Aided Design, vol. 11, no. 9, p1074-1085.

This paper introduces the second smallest eigenvalue based graph partitioning algorithm, which captures both min-cut and equipartition. Experiments show that this algorithm produces better partitions even without any local improvement than several state-of-the-art algorithms.

Haines, Roger W. (1983). *Control Systems for Heating, Ventilating, and Air Conditioning,* Van Nostrand Reinhold Company Inc., New York.

Huang, J.-H. Dennis and Andrew B. Kahng (199?). *When Clusters Meet Partitions: A New Density Objective for Circuit Decomposition*, p1-17.

This paper presents a new measure of multi-way circuit decomposition, based on a sum of densities objective. Here, the density is the ratio of the number of edges to the number of nodes in the subgraph. The major difference between this measure and ratio cut is that this measure focuses more on the internal view of each cluster while ratio cut focuses more on the external view between the clusters (cuts).

Innovative Tech (1996a). *SPAN.FM User's Manual -Vol. 1 (first printing),* Warminster, PA.

This user's manual gives detailed description of how to use the different functionalities of SPAN.FM.

Innovative Tech (1996?). *SPAN.FM for Windows,* Warminster, PA.

This brochure gives an overview of the design and functionalities of SPAN.FM.

Innovative Tech (1994). *Facilities management Solutions SPAN.FM*, Warminster, PA.

Jacobson, I., M. Christerson, P. Jonsson, and G. Overgaard (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, New York.

This book introduces use case method, an Object-Oriented Software Engineering method (OOSE).

Karger, David R. (1994). *Random Sampling in Cut, Flow, and Network Design Problems*, Proceedings of the 26th Annual ACM Symposium on the Theory of Computing, New York, NY, p648-657.

In this paper, the authors explore random sampling tool for solving undirected graph problems. It is shown that sparse graph, or skeleton, which arises when randomly sampling a graph's edges will accurately approximate the value of all cuts in the original graph, which makes sampling effective for solving cuts in graphs.

Karger, David R. and Clifford Stein (1993). *An O(n exp 2) Algorithm for Minimum cuts*, Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, New York, NY, p757-765.

This paper presents an algorithm for finding minimum cuts in weighted undirected graphs, which breaks the O(mn) "max-flow barrier" for finding minimum cuts.

Kirkpatrik, S., C.D. Gelatt, Jr., and M.P. Vecchi (1983). *Optimization by Simulated Annealing*, Science, 220 (4598), p671-679.

This paper introduces the simulated annealing algorithm, which is a stochastic optimization technique demonstrated to solve continuous or ordered discrete optimization problems of fixed structure.

Klix, Friedhart and Hartmut Wandke (1986). *Man-Computer Interaction Research: MACINTER-I*, Proceedings of the First Network Seminar of The International Union of Psychological Science (IUPsyS) on Man-Computer Interaction Research, Berlin, German Demacratic Republic, October 16-19, 1984, Klix, Friedhart and Hartmut Wandke (Eds.) , Elsevier Science Publishers B.V., The Netherlands.

This book contains a collection of papers on human-machine interaction.

Lang, Kevin and Satish Rao (1993). *Finding Near-Optimal Cuts: An Empirical Evaluation*, Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms, p212-221.

This paper describes the experiments in which the efficiencies of the four initial (local) partitioning algorithms (including the FLOW algorithm) are compared.

Lang, Laura (1989). *Facilities Management*, Computer Graphics World, December, p32-38.

This paper is a research on users' feedback to the commercially available FM programs.

Liggett, Robin S. (1978). *An Exploration of Approximate Solution Strategies for Combinatorial Optimization Problems*, Ph.D. Dissertation, UCLA.

Liggett, Robin S. (1979). *The Quadratic Assignment Problem: an Analysis of Applications and Solution Strategies*, Environment and Planning B, 1980, vol. 7, p141-162.

This paper discusses quadratic assignment problems and heuristics applicable to this class of problems. It also compares the trade-offs between computational efficiency and quality of solutions generated.

Liggett, Robin S. (1981a). *Optimal Space Planning in Practice,* Computer Aided Design, September 1981, vol. 13, no. 5, p277-288.

This paper discusses the Computer Aided Design Group's Space Planning System, a software which produces solutions related to operating efficiency. Specifically it discusses (1) stacking and zone plan optimization problem, (2) block plan optimization problem, and (3) move optimization plan.

Liggett, Robin S. (1981b). *Interactive Graphic Floor Plan Layout Method,* Computer Aided Design, September 1981, vol. 13, no. 5, p289-298.

This paper discusses a floor layout algorithm based on the quadratic assignment problem formulated by Koopmans and Beckmann and a raster-like interface.

Liggett, Robin S. (1992). *Designer-Automated Algorithm Partnership: an Interactive Graphic Approach to Facility Layout*, in Yehuda E. Kalay (ed.), Principles of Computer-Aided Design: Evaluating and Predicting Design Performance, John Wiley & Sons, Inc., New York, p101-123.

This paper gives a detailed introduction of a stacking, blocking, and one-to-one assignment algorithms developed by Prof. Liggett, and graphical user interface built on top of the algorithms.

Mahdavi, Ardeshir (1995). *Fundamentals of Building Physics,* Department of Architecture, Carnegie Mellon University, Pittsburgh, PA.

This document discusses the physics relationships between different parameters in building performance and gives calculations of these parameters.

Mahdavi, Ardeshir (1996). *SEMPER: A New Computational Environment for Simulation-based Building Design Assistance*, Proceedings of the 1996 International Symposium of CIB W67 (Energy and Mass Flows in the Life Cycle of Buildings), Vienna, Austria, p467-472.

This paper introduces the underlying concepts and structure of SEMPER, an active multi-aspect computational tool integrating building performance simulation (energy, lighting, and acoustics) into computational design system.

Mahdavi, A., Mathew, P., Kumar, S., and Wong, N. H. (1997). *Bi-directional Computatinal Design Support in the SEMPER Environment*, Automation in Construction 6 (1997) pp. 353 - 373.

Mahdavi, Ardeshir, Akin, Ömer, and Zhang, Ye (1998). *Formalization of Concurrent Performance Requirements in Building Problem Composition*, Proceedings of the 4th Design and Decision Support Systems in Architecture and Urban Planning Conference, Maastricht, The Netherlands.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953). *Equations of State Calculations by Fast Computing Machines*, Journal of Chemical Physics, 21: p1087-1091.

This paper introduces Monte-Carlo technique which is the basis for simulated annealing algorithm.

Michalski, S. Ryszard and Robert E. Stepp. (1983). *Learning from Observation: Conceptual Clustering*, Machine Learning: An Artificial Intelligence Approach, Michalski, R.S., Carbonell, J. and Mitchell, T. (Eds.), TIOGA Publishing Co., Palo Alto, CA, p331-363.

This paper discusses **conjunctive conceptual clustering**, in which a configuration of objects forms a class only if it is desirable by a concept from a predefined concept class. The presented method arranges objects into a hierarchy of classes closely circumscribed by such conjunctive descriptions.

Mitchell, Tom M. (1996). *Machine Learning*, Tom M. Mitchell and McGraw Hill.

This book covers from the basic concepts, tasks, state-of-the-art algorithms, to the computational theories in machine learning.

Park, James K. and Cynthia A. Phillips (1993). *Finding Minimum-Quotient Cuts in Planar Graphs*, Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, New York, NY, p766-775.

This paper introduces several new algorithms for finding small-quotient cuts in planar graphs.

Pentland, Alex P. (1987). *Recognition by Parts*, Proceedings of the 1st International Conference on Computer Vision, Washington, DC, p612-620.

This paper provides a natural way to recognize an image--by its parts. The representation issues and advantages of recognition by parts are also discussed.

Pfleeger, Shari Lawrence (1991). *Software Engineering: The Production of Quality Software*, Macmillan Publishing Company, New York.

Rajan, E. G., Vinay Kumar, A. G. S. Kiran and Ajitha Chowdhary. (1995). *Neural Automata Based Object Recognition*, 1995 IEEE International Conference on Systems, Man and Cybernetics, New York, NY, vol. 2, p1882-1888.

Neural automata are structural level adaptive systems which show self organization behavior in addition to self learning. The process of neural automata based object recognition system consists of segmentation, skeletonization, iconization and model fitting.

Rao, Satish B. (1992). *Faster Algorithms for Finding Small Edge Cuts in Planar Graphs*, Proceedings of the 24th Annual ACM Symposium on the Theory of Computing, p229-240.

In this paper, the authors present a simple algorithm for finding 1.5 times optimal quotient node or edge cuts in planar graphs. They consider partitioning a planar graph by removing either nodes or edges. In particular, they consider a cut to be either a set of nodes or edges whose removal divides the graph into two pieces.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). *Object-Oriented Modeling and Design*, Englewood Cliffs: Prentice Hall.

This book introduces Object Modeling Technique (OMT), a method to complete the design of an object-oriented software. Three models are introduced. They are Object Model, Functional Model, and Data Flow Model.

SABA Solutions (1996). *WinSABA: A Stacking and Blocking Algorithm for Windows*, retrieved November 29, 1997 from the World Wide Web: http://www.techexpo.com/WWW/saba/saba.html.

This page introduces the WinSABA computer program.

Saran, Huzur and Vijay V. Vazirani (1991). *Finding k-cuts within Twice the Optimal*, Proceedings 32rd Annual Symposium on Foundations of Computer Science, Los Alamitos, CA, p743-750.

This paper introduces two algorithms for the minimum k-cut problem. Each algorithm finds a k-cut having weight within a factor of (2-2/k) of the optimal. One of the algorithms is particularly efficient- it requires a total of only n-1 maximum flow computations for finding a set of near-optimal k-cuts, one for each value of k between 2 and n.

Shor, Peter W. (1991). *How to Pack Better than Best Fit: Tight Bounds for Average-Case On-line Bin Packing*, Proceedings 32rd Annual Symposium on Foundations of Computer Science, Los Alamitos, CA, p752-759.

This paper introduces a O(nlogn) bin packing algorithm for packing items i.i.d. uniform on [0, 1] into bins of size 1 with expected wasted space. This algorithm gives less expected wasted space than that of the Best Fit algorithm. Through analysis of another algorithm which involves putting balls into buckets on-line, it is also shown that the bounds of the introduced algorithm on rightward matching give good bounds for the two-dimensional strip packing problem.

Stein, Benjamin, John S. Reynold, and William J. McGuinness. (1986). *Mechanical and Electrical Equipment for Buildings,* John Wiley & Sons, Inc., New York.

This book gives a detailed technical discussion of major aspects related to building performance, the relationship between mechanical and electrical systems and the buildings they serve, and the role that environmental control systems play in the design of buildings as well as in building performance.

Teicholz, Eric (1994). *CAFM: Making it Work*, Interiors, February, p22-23.

Through a looking back at the evolution of computer-aided facility management, it is revealed how important a well-considered management approach to automating really is.

Teicholz, Eric and Michael Sena (1986). *Forward to the Past*. Computer Graphics World, June, 1986, p15-22.

This paper discusses some technical issues in state-of-the-art stacking and blocking approaches.

Thompson, Kevin and Pat Langley (1991). *Concept Formation in Structured Domains*. Concept Formation: Knowledge and Experience in Unsupervised Learning, D. H. Fisher, M. J. Pazzani and P. Langley (Eds.), Margan Kaufmann Publishers, Inc.

This paper describes **LABYRINTH**, an implemented system that induces concepts from structured objects. **LABYRINTH** is viewed as an approach to **incremental concept formation**. It takes the input which is a sequential presentation of objects and their associated descriptions, and finds clusterings that group these objects into concepts, a summary description for each concept and a hierarchical organization for these objects. A historical background survey of six related systems are also introduced.

Trimberger, Stephen (1989). *Integrated Circuit Productivity Advancements in the 1980s and 1990s*, Proceedings of the Twenty-Second Annual Hawaii International Conference on Systems Sciences: Architecture Track, vol 1, p74-81.

This paper gives an overview of the development of computer-aided VLSI design in the last two decades. The quality of computer generated solutions as compared to experienced and novice users is evaluated. The importance of using automated design systems is also discussed.

Vaidyanathan, Akhileswar Ganesh and James A. Whitcomb. (1995). *Adaptive Image Analysis for Object Recognition, Part I - Entropic Object Location*, 1995 IEEE International Conference on Systems, Man and Cybernetics, New York, NY, vol. 2, p1888-1891.

This paper constructs an automatic and adaptive analysis method in image recognition. The basic idea is that edge (boundary) detection is a pre-condition for recognizing images. Entropy function is used in getting the boundaries between different objects since an entropy function has its self adaptive behavior. After that, recursion helps to refine the information about an object while a spatial object hierarchy helps to avoid redundant scanning.

Weinand, A. and E. Gamma. (1995). *ET++ - a Portable, Homogenous Class Library and Application Framework*, research report, Cupertino, CA: Taligent, Inc.

Whitcomb, James A. and Akhileswar Ganesh Vaidyanathan. (1995). *Adaptive Image Analysis for Object Recognition, Part II - Redundant Object Resolution*, 1995 IEEE International Conference on Systems, Man and Cybernetics, New York, NY, vol. 2, p1892-1898.

Woodbury R., and Teng-Wen Chang. (1995). *Massing and enclosure design with SEED-Config*. J. Arch. Engrg., ASCE, 1(4), p170-178.

Appendix A:
Transcripts of Architects Interview

A.1 David R. Hamburgh

D.R. Hamburgh Associates
5135 Fifth Ave.
Pittsburgh, PA
(412) 621-5511

What functional requirements do architects consider in stacking and zoning?

There are different functional requirements to be considered in stacking and zoning. They include:

- adjacency requirements;
- thermal requirements (HVAC, mechanical);
- acoustic requirements;
- footprint for structural requirements;
- view from inside rooms.

Adjacency requirements are most important since they are related with personnel, and cannot be mechanically reproduced. Initial design process is based on adjacency.

There are other factors that should be considered as well, they include individual comfort, needs of equipment, orientation to the sun, daylighting, relationship to elevators and core facilities, and accessibility.

Is stacking and zoning important?

It is very important. It defines the relationships between spaces. The goal of stacking and zoning is to improve the functionality of a building.

How do architects do stacking and zoning?

- do not do it systematically;
- concurrent with layout design, not a separate step from layout.

Is computer support needed?

Yes. But:

- Computers have to violate some adjacency requirements in order to get globally optimal solution. Architects have to accept violations.

Computers are not practical. For example, a structural design software designed by SEI is supposed to generate core and configuration of a building; but it can only handle symmetrical footprint. This imposes limited use and loses the practicality of the process.

What quantifies accessibility?

Accessibility is generally considered in fire requirements (egress). For example, for an auditorium, several exits may be required with each of them being an access to a courtyard.

Accessibility is a combination of both physical effort and psychological effort. Specifically, it has the following features:

- number of floors in between (physical effort);
- horizontal distance in between (physical effort);
- number of obstructions in between (mental energy).

What features of software do architects like?

Don't like AutoCAD because it is not easy to use. Use PowerCADD3.0 (on MAC) instead.

- PowerCADD has different layers for a drawing. It also has different modifiers (different options) for a drawing element.
- PowerCADD can produce precise drawings, e.g., when a red dot appears you are informed that the line you are drawing is tangent to a circle.
- PowerCADD is easy to learn.

A.2 David C. Brenenborg

Brenenborg Brown Group
4018 Penn Ave.
Pittsburgh, PA
(412) 683-0202

What functional requirements do architects consider in stacking and zoning?

Adjacency requirements is most important. "Adjacency" is the right word. Adjacency does not have to be right next to each other, there are different levels of adjacencies. Distance is a kind of adjacency. Adjacency means two things: (1) work relationship, and (2) utility relationship.

Acoustic and thermal requirements are useful in zoning, not in stacking.

There are also a lot of implicit criteria in decision making. For instance, when you decide if the managers should have the nicest view or the majority should have the nicest view: it's a decision on democracy.

How do architects do stacking and zoning in designing a large-sized building?

Divide and conquer. Locate bigger departments first.

BTW, stacking and zoning is an explicit step before a layout design.

Recommendation of a software and literature

- About 10 years ago, Steelcase in Michigan, a furniture company, either owned or developed a software to automatically allocate departments into different floors according to the importance of adjacencies.

- Charlie Eastman, a professor in CMU long ago, worked on AI. He was a professor in both architecture and CS. He may have some work on this.

On categorical/discrete acoustic values

It's good to provide categories for architects. It's ridiculous to enter numerical values.

Is computer support in stacking and zoning needed?

Don't trust computers. Skeptical about its value. Computer should be a tool like a pencil, a T-square instead of a decision maker.

Computers are not useful, because:

- There are a lot of criteria beyond structural, mechanical, and acoustic requirements. How can computers solve this?

- Architects care about things put in the right place, flow good, and look good. How can computers do this?

- Computers are cumbersome to use. For example, even to draw a stairwell, you have to enter about 10 numbers, by the time computer draws this stairwell, architects should have finished designing two stairwells.

- Computers are not easy to use. People have to enter numbers into a computer in a special format and syntax.

A.3 Jason Fourier

Hayes Large Architects
606 Liberty Ave., 4th floor
Pittsburgh, PA
(412) 391-3086

What functional requirements do architects consider in stacking and zoning?

- Adjacencies are critical. Adjacencies determine circulation.

- Square footage of spaces.

- Acoustic requirements are important in stacking only when you have an equipment room and you want to put it either on the first floor or on the top floor. Otherwise it's not important, e.g., low frequency reverberation (e.g., lecture and talking) is not a big problem.

How much time do architects devote into stacking and zoning?

A typical building design process consists of the following stages:

- Schematic design: it's most important, it determines the mass of a building. At this phase, footprint, mass of a building, stacking, furniture, and equipment are decided; materials are speculated (not decided); adjacencies and square footages are accounted; 60-70% of time is spent in this stage; 15-20% of time spent in this stage is spent in stacking and zoning.
- Design development: at this stage, finishes are decided; 15% time is spent in this stage.
- Construction document: materials are decided.
- Construction administration.

What software do architects like?

Like MicroStation best since it is easy to use. Use AutoCAD and Arris (a drawing tool) as well.

Don't need computer support in stacking and zoning. Architects can do it well.

Recommendation of literature

Bill Hillier is a professor in University of London. He does research on urban planning - deciding where to locate shopping centers and other facilities based on inhabitants' residence and mobile patterns.

A.4 Robert Beckjord

DBA Architects, Inc.
1100 Liberty Ave.
Pittsburgh, PA
(412) 456-1000

What functional requirements do architects consider in stacking and zoning?

- Adjacency, or functional dependency are most important (adjacency matrix and adjacency graph).
- Structural requirements are important in stacking (e.g., spaces with heavy equipment and with smaller spans should be located on lower floors, and vice versa) and zoning (e.g., spaces with the same span should be grouped on the same floor).
- Acoustic requirements may be important.

- Circulation is important for both subjective and objective reasons: (1) subjectively speaking, circulation pattern represents clarity and quality of a design, it's important in way-finding; (2) objectively speaking, minimum safety standards have to be satisfied, fire resistance (egress) requires that there should be two exits from an existing building; corridors should stack one on top of the other in order to save cost.

- HVAC control may be important in deciding perimeter wall.

The goal of stacking and zoning is to satisfy adjacency requirements and get minimum exterior wall (smallest enclosure) in order to (1) achieve energy efficiency, and (2) build less expensively.

Clarification of some concepts

- Conceptual design: It is an experimental stage in the building design. In this stage, architects come up with stacking and zoning, applying to what the structure looks like, what the volume of the building looks like, and what the materials are.

- HVAC = thermal; mechanical = HVAC+plumbing.

How much time do architects devote in stacking and zoning?

There are five stages in building design (the first three stages are design stages; the last two stages are construction stages):

- Schematic design: 15% of time is spent in this stage;

- Design development: 15% of time is spent in this stage;

- Construction drawing: 40% of time is spent in this stage;

- Billing: 5% of time is spent in this stage;

- Construction administration: 25% of time is spent in this stage. 20% of schematic design time is spent on stacking and zoning, i.e., less than 10% of the overall design time is spent on stacking and zoning.

Is computer support for stacking and zoning needed?

In a manual design process, in order to develop design alternatives, architects normally use the first alternative as a basis to come up with other alternatives. They also use the first alternative as a qualifier[1] for other alternatives.

Computers provide alternatives which are optimized results and satisfy all the requirements, and all the alternatives don't have "kinship" relationship and therefore are of equal opportunity.

1. Qualify means evaluating and finding something good.

Computers quickly generate solutions and are more efficient, but this may increase architects' evaluation time because in order to improve a computer-generated design, architects have to modify the input data which cannot be done easily. Thus in order to increase efficiency, computers must be easy to use.

There are two elements in architecture design, namely quantification and qualification. Computers quantify information and architects qualify information.

Appendix B:
Transcript of Protocol Analysis

B.1 Subject: Ph.D. student in related area (BPD)

Date: Nov. 7th, 1995

Q[1]: What criteria shall I use in grouping the things--just according to thermal design or according to thermal zoning with architectural layout?

Q: Is there any criteria as to what spaces should go to the second floor?

Q: Is there any implication of which spaces should be put together?

He puts the same-colored pieces together and reviews the list of spaces in collation to the colored pieces.

S[2]: Urgicare is at the ambulance entrance which means that urgicare should be on the first floor at the ambulance entrance.

Q: From which entrance can I go to the exam rooms?

S: So at the beginning I break them into different chunks so that I can start grouping them. The bathrooms should be near the waiting area.

He starts to put pieces on the first floor plan.

Q: Should new patients go to the reception area first?

S: There has to be a staircase somewhere around the main lobby.

S: The accounting should be pushed onto the second floor. We have to push things up and down because of area requirement.

He arranges pieces in the entrance / reception.

S: We should really take advantage of daylight since it's a long and thin building.

He arranges pieces in urgicare.

S: I can put kitchenette, men's shower, women's shower,... in the existing lounge. The only thing I don't like the staff lounge much is that it does not have much daylight.

Q: Do I have to save the position of the existing elevator? Can I build new staircase so that I can go to the second floor?

Q: We should keep the restaurant, right?

1. Q is an abbreviation of Question by the subject in this whole transcript.
2. S is an abbreviation of Statement by the subject in this whole transcript.

Q: This lobby seems tremendously huge. What activities take place in the lobby? What exactly are contained in the reception area? Will the patients go to the reception area then go to the main lobby, or go to the main lobby first then go to the reception area?

S: What I am going to do is to stretch the main lobby thinner and longer. Then stretch the reception. I will put the gift shop facing the walkway so that people can see things in it.

He places things...

S: Let's for the time being leave it like that.

Q: Nurse station should be close to the ambulance entrance and waiting area. What happens to the dirty room and the clean room?

S: So I leave some space which I probably need.

He reviews the layout on the first floor.

S: In terms of spatial layout, I put entrance / reception, registration / billing and all that kind of stuff near the main entrance.

Q: If a patient comes in, would he go to the doctor's office?

S: OK. Patients usually don't go to doctor's offices. I can try replacing the doctor's offices with the exam rooms. So doctor's offices will have daylight also. I guess it would also be nice if the exam rooms have daylight.

S: OK. Patients usually don't go to doctor's offices. I can try replacing the doctor's offices with the exam rooms. So doctor's offices will have daylight also. I guess it would also be nice if the exam rooms have daylight.

S: Oh from thermal zoning point of view this is going to be problematic. Maybe I should put the doctor's offices back and the exam rooms back. All the exam rooms will be zoned together since they are next to each other. Waiting area, nurse station and... can be a separate zone. All the doctor's offices together become one zone.

S: (Pointing to the available space) We can have a staircase here. So we can have two staircases, one on here and one is in the main lobby.

S: Lets' go up to the next level.

S: So I break a large chunk into smaller chunks. It kinda makes sense to put similar occupancies together from either thermal or performance point of view.

S: I am going to shuffle things (in accounting) around. I think the exam rooms (in pediatrics) should go here so that there are similar occupancies below. It probably makes sense to go from the staircase in the main lobby up to pediatrics. So I will put this stuff (accounting) backward... We need daylight for these people.

He reviews floor two plan.

S: Let's see what happens. Maybe I should put accounting back because this area is less and there needs to be more spaces for the pediatrics.

He puts the accounting back.

S: Doctor's offices definitely should have daylight.

After placing accounting...

S: *The rationale was that: I took the existing organization, major groupings of accounting, pediatrics, urgicare and entrance/reception-- I took them as is because it seems to be the most convenient to me*[1]*. Because I am not very familiar with how exactly spaces in a hospital thing should be arranged, I am assuming the groupings given to us as planned also represent desirable physical groupings in a natural physical/spatial setting. So I tried to maintain those groupings.*

With the groupings of essentially each of those, two groupings are located on the lower floor, and two groupings are on the upper floor. The choice was kinda obvious: the entrance / reception should be towards the south because it should be close to the main entrance; urgicare should be towards the north because it should be close to the ambulance entrance. Up to the second floor, I put pediatrics towards the north because 1). We have exam rooms and doctor's offices on the lower floor; 2). There is more space in the pediatrics area and there is more space towards the north.

It's really nice that we have long plans because basically we have enough daylight so I used double-loaded corridor.

As for the bathrooms, dirty and clean rooms which people don't need to spend much time in, I tried to locate them in the core area.

Doctor's offices are placed along the north because they have nice daylight. Other major exposure is the west side, and of course not particularly desirable from thermal load point of view. But it could be controlled of the solar gain on the west.

I still have to locate where the staircase is going to be, either on the west side or on the east side. Probably it makes sense to put it on the west side so that we have more nice spaces on the east side.

Oh I am not happy with the kitchenette and shower over here. So I can push them back up there. Yeah, I can put them here (in the north wing).

He adjusts the entrance / reception pieces.

---

1. Important points and conclusions are highlighted by italic style.

The staircase is here (pointing to the first floor). The staircase is here (pointing to the second floor).

Now I have daylight for all inhabitable rooms. Oh I just realized a problem here. The existing elevator is here; some disabled has to go through the accounting area. That it not very nice. How can I solve this problem?

Q: Can I build another elevator?

He thinks hard.

Q: We have to use the elevator for patient transportation, right?

He thinks about the existing elevator.

Q: I don't need to keep the existing lounge, right?

S: Then I can move the things over here (along the wall) so that the main lobby can be extended straight to the existing elevator.

He switches to the second floor.

S: So when you come out from the elevator, you hit... first. There is a staircase here for this area. Then you don't need the staircase. This is odd.

Q: How big is the elevator? Will it be able to take all the load that comes in?

S: We do need another stair, right?

He exchanges the positions of pediatrics and accounting.

S: There has to be a staircase anyway because of fire requirements. I can put the staircase here (pointing to a place north of pediatrics).

He puts accounting on the north part of the plan.

S: Now I am done. What I think is: I will keep the existing elevator for moving up and down. And we have one staircase down here for the main lobby. I get rid of the employee lounge. There is going to be another staircase (pointing at the inner corner of the "L" shape). So from both entrances you have a staircase within a short distance to move up to the second floor. The staircase here on the second floor can serve the accounting area. On the first floor there should be another entrance somewhere besides the ambulance entrance.

I put bathrooms for both accounting and pediatrics in one area. From zoning point of view it makes sense. I put all the exam rooms in a row on the west. It's preferable that you put the exam rooms on the west than put doctor's offices on the west because doctor's offices will need nice air and open windows, similarly for the waiting area. For the exam rooms, as I said earlier, they need to be controlled. They can have small windows which are closed.

I[1]: At the beginning you put pediatrics on the north area because you wanted to zone the exam rooms with the exam rooms underneath in the urgicare, and zone the doctor's offices with the doctor's offices underneath. Later on you noticed there was a circulation problem. So you moved them (pediatrics) down. So do you think circulation plays a more important role then thermal zoning?

A[2]: Yes, I think so. To some degree, basic spatial organization is a more important issue.

The other reason is I don't know the occupancy characteristics of these spaces too well, but I presume from thermal zoning point of view all the offices are similar--doctor's offices and accountant's offices. I assume all the exam rooms should be individually controlled depending on how sensitive the equipment in there is. As far as I put them together, there can be one larger air handling system and they have individual control.

I assume each floor will have a different system. It doesn't make sense to zone the exam rooms on the second floor with the exam rooms on the first floor, because there is roof above the exam rooms on the second floor and they will have different loading condition. So I treat each floor differently mechanically with each floor having different zones.

I: When you grouped things together, what major aspects do you consider?

A: *Most importantly, 1). basic chunks. within each group I looked at sub-groups; for example, exam rooms. 2). I looked at it as a thin, long space. So the most efficient way of using the space is to have double-loaded corridor type of situation. Beyond that every inhabitable space, especially for a building like this shape, does have daylight. This is absolutely critical. 3). I tried to put bathrooms, kitchenette together so that they would share the piping and plumbing. I try to put the piping and plumbing together in the core area. Also here I aggregated the bathroom functions.*

I: So another factor you considered is daylight.

A: Yes. Daylight is very critical. I ensure that every continuously inhabitable space has daylight; especially for such a generous space and the shape of the building.

I: For each sub-group within a department, you considered the priorities of different directions--east and west...

A: Yes. The priorities of orientations are north, south, east and west, because east and west have low sun angles which is hard to control. So I put accountant's offices fac-

---

1. I refers to a question/answer/comment by the author.
2. A is an abbreviation for Answer by the subject in this whole transcript.

ing the north which is very nice. As to the pediatrics, I put doctor's offices on the east and exam rooms on the west because they may not need view or can be private or closed. So it's better to put them on the west. For doctor's offices, probably they will use operable windows which are bigger; so it makes sense to put them on the east. Also it's better to have view over the courtyard than over the parking area. Then I assume exam rooms don't have operable windows.

B.2 Subject: architect

Date: Nov. 13th, 1995

S: As I see here, I would put the main lobby near the main entrance. What I am doing now is to try to get related functions together near the main lobby, such as registration / billing. I will put exam rooms next to the registration area.

There is a waiting area in the urgicare. From the waiting area we should have access to the six exam rooms.

Here's a nurse station. We can make the nurse station a float space. From the ambulance entrance, we can go to the nurse station, waiting area and exam rooms.

I can put the doctor's offices adjacent to the exterior wall on the north side.

Q: Can we adjust the position of the door to the existing nursing home?

S: Maybe we can connect the bathrooms to the existing bathrooms.

S: What about the reception area? We can make it a part of the main lobby.

S: The reason I put waiting area and exam rooms in between the two entrances (main entrance and ambulance entrance) is that from either of the entrances, you can come to this area conveniently.

S: I group the kitchenette, locker, shower and bathroom together. From many different points of view, if you group the service areas together, you make the building more efficient by having the plumbing together.

He goes back to the entrance / reception.

S: I make temporary children's room part of the reception area.

S: So a patient comes from the main lobby to the registration / billing first, then to the nurse station and waiting area, then to the exam rooms.

I put the bathrooms here to serve the main lobby, reception area and maybe the restaurant.

S: The nurse station will be a counter; the waiting area will be an open area.

S: I think we need a reception area for the urgicare.

S: Exam rooms have nothing to do with emergency. I don't see any relationship between the exam rooms and the ambulance entrance. If you have an emergency case, you can go to an operation room or a first-aid room. Maybe one of the exam rooms can be changed to an operation room.

He arranges pieces in urgicare.

S: If we have the option of removing the existing bathrooms, we can convert one of them into a staff kitchenette.

S: Now the second floor.

He puts the same colored pieces (pediatrics and accounting) together.

S: I think we must have a lobby here for the elevator.

Q: Is accounting for the patients (pediatrics) or for the building itself as a business function? It has nothing to do with patients (pediatrics)? Then I will put it backward, away from the elevator.

He places pieces in pediatrics.

S: From the elevator lobby, we can go to the registration and billing office. We can expand the waiting area to serve the six exam rooms.

Q: Is the record storage for the filing system?

He places pieces in pediatrics.

S: I group nurse office, billing, record storage and registration together; and group the bathrooms with the waiting area.

S: This has a drawback. When you want to go to the accounting, you have to go through the pediatrics.

He places pieces in accounting.

S: Here I can add a nurse station for the waiting area just to manage the calls from the sick people.

After a patient goes past the nurse office, registration, billing and record storage, he will go to the waiting area and the six exam rooms.

...

Here I put a corridor between the doctor's offices for privacy.

S: Here I can put a door to separate pediatrics and accounting. We should try not to go through pediatrics in order to get to accounting. I will try another approach to solve this problem.

We can keep the original layout, but add a separate side corridor from the elevator lobby to the accounting in order not to go through the pediatrics. It's better for the accounting to have a staircase here so that we can have direct access from the first floor to accounting without the need of a separate corridor.

I: When you grouped the spaces, what major aspects did you consider?

A: *1). The functions of the spaces, how they relate to each other, and how to access from one to another. For example, the main functions like exam rooms, doctor's offices, billing office, and etc.*

2). Economy. I put bathrooms together to share plumbing. We can't put bathrooms everywhere.

3). Open spaces and float spaces. In the waiting area I preferred to have open spaces instead of closed spaces so that we can have control over this area.

4). Privacy. For example, I have a main corridor, I also have a smaller corridor for the doctor's offices to make the doctor's offices private. You can see such arrangements in several other places. Also I have one nurse station and two other accesses. I have to have a space controlled by the nurse station to control the accesses. That's why I made it private.

I: What factors in thermal zoning / thermal design did you consider?

A: *Usually we make the center of the building as a control zone and the exterior areas connected to the elevations of the building as separate zones. But because the nature of this building is too narrow, we'll only have one zone.The heat transfer from the external area to the internal area cannot be applied here because the width of the building is only around 40 feet.*

But for a square-shaped building we can make separate zones because the difference between the inside and outside is too much.

We can sub-divide a zone. For example, we can make the exam rooms in pediatrics a small separate zone because they need higher control for heat and fresh air. Maybe the doctor's offices don't such control.

So according to different functions, we sub-divide a zone. Different functions will determine the thermal zoning, maybe also electric and supplies.

I: Why did you put entrance / reception and urgicare on the first floor; and put pediatrics and accounting on the second floor?

A: *Because the main entrance is here, I put the main lobby near it as a connection to the rest part of the building and the upper level. As for the exam rooms (urgicare), I think most people come here for examination; so I put it on the first floor also.*

I put accounting offices on the second floor because they are not for the outside patients. They have a remote relationship with the outside. As for the exam rooms in pediatrics, I consider them as a second stage for the patients. Urgicare is for emergencies; but pediatrics is not.

To get more function of this building, we can consider adding another elevator or staircase. To improve the design for this "L" shaped building, we should add another elevator to serve the building, so that we don't need this corridor to go past another area.

Q: Do you compare my plans with the layouts generated by computer?

B.3 Subject: mechanical engineer1

Date: Nov. 15th, 1995

Q: First of all, this is an existing building. Alterations will be within the existing building. Are there any additions to the existing building?

S: I will comment as regard to mechanical--HVAC, plumbing and maybe fire protection. Here is a philosophical comment: We try to accommodate whatever the architect would like. We also value any architect's willingness to accept our input in the process of defining functional spaces. Because many times we are ended at design which has no recognition of mechanical systems. This is of course the theme of this evening-- You want my input.

S: Let's start with plumbing. The reason for this is that especially in an existing building, cost of installation and amount of disruption to existing spaces can be minimized if new plumbing functions are situated near existing plumbing functions--waste and stacks.

Q: In your new functions, do you have anything that requires plumbing?

S: I noticed the existing bathrooms on the first floor and second floor, not surprisingly, they are stacked one on top of the other. I noticed also a restaurant which implies a plumbing here. So my first advice is: if you can stay with your functional groupings, if you can situate any bathrooms either in the area near the existing bathrooms or near the restaurant, that would achieve proximity to the existing plumbing.

I just give you my thoughts which keep recurring all these things simultaneous always in my mind, probably recur as the design is refined.

Q: Do you have anything that needs exhaust?

Q: Bathrooms need exhaust, toilet rooms need exhaust. Do you have laboratories that might need exhaust?

S: Usually in cases like this, such as certain regulations for skilled nursing homes, dirty equipment and clean equipment must be accommodated with a ventilation system in that dirty equipment or soiled linen should be given exhaust ventilation and kept under a negative pressure.

S: Also, there are code requirements, the BOCA building code is rapidly becoming universal. So I will invoke requirements of the BOCA code in the discussion.

A requirement of the BOCA code is that any exhaust for foul air, uncleaned air, whether from a toilet complex, or from a soiled linen room, things of that nature, must maintain a certain distance from any fresh air intake for the HVAC system.

I don't know the nature of the existing HVAC system. This is an exiting building. It should have an existing HVAC system.

You give me the latitude of redo the whole building. Other things that are going into my mind which will affect the HVAC and how the HVAC system affects the rooms is as follows:

When you have a building with different functional areas, if these areas have different hours of operations, that must be known, e.g., certain areas might be typical 40 hours a week; other areas might be around-the-clock, e.g., if there is an emergency area, that is an example of 24 hour area... So the area (entrance / reception) is identified as probable around-the-clock operations.

From my point of view if you can group all functions in the building that are around-the-clock operations, that is an advantage, which means you might put them in one HVAC system; you could run them one system around the clock.

For other typical business areas, if you have them in another system, you would not run them 168 hours a week. So in this way you can save operating cost.

Always keep in mind, I consider desirable the functional use in occupancy of the building really does take priority among members of the design thing.

Q: Are all the areas to be conditioned? I assume so. Are you showing me only a two-story building or the two-story of a multi-story building?

Q: Is this restaurant a full-service kitchen, e.g., does it have, this is important, does it have an exhaust hood over a food preparation arrange; do they have grease exhaust? This is important to know because they might have a grease exhaust. My guess is since this restaurant is one story area in the courtyard, that exhaust probably is through the roof of the restaurant and that would influence the concept of the HVAC system. Fresh air intakes are to be avoided in the area.

S: Let me look at the functional list again, entrance / reception, no matter what I will do, you've already told me the main entrance is here. So the blue areas will generally be clustered here. Bathrooms should be near either existing bathrooms or near existing plumbing. There must be some plumbing near here (restaurant).

For the existing bathrooms, you can remove, modify them, but the important thing is that there is plumbing here.

I see locker, bathroom and shower; will they have in common is exhaust and ventilation? Significant of the exhaust is that the exhaust has to leave the building somewhere. I try to honor the aesthetics of the building by having as few penetrations, encroachments upon the building facade. For this low-rise building, I'd like to take the exhaust up through the roof. For that matter, if the areas must be exhausted around lower-levels, the elevations of the lower-level space will necessarily encroach upon the high-level floors.

Q: This is the main entrance for the main building, not for employees, right?

S: You'll need to give me a shaft for ventilation, penetrating the second floor then up to the roof.

S: I can play as an architect as well as an engineer. The bulk of the accounting should be on the second floor. The pediatrics doesn't seem to me an emergency function, so I put it on the second floor. If a child has emergency, he'll go to the urgicare on the first floor.

Now we have designed four general areas; now we should define each individual space within each general area.

I have a strong feeling that this has to be an iterative procedure. I keep changing my mind until I get an optimal learning.

He places the spaces in entrance / reception.

S: I don't want bathrooms too obvious from the main entrance. By placing the new bathrooms in entrance / reception, I'm defining a need for mechanical shaft for ventilation use. But it's so near the existing bathrooms, I might go laterally a short distance (assuming the existing bathrooms are ventilated, exhausted through the roof) and join with that may have the two together as one shaft instead of expanding two shaft spaces. That's a detail. We'll work on that later.

S: Now the urgicare. Maybe we can group the various plumbing functions for the urgicare as a group, put them over here (on top of existing bathrooms).

For the dirty room, I can put it near the bathrooms so that it can share the exhaust shaft. But I don't want to put the dirty room near the kitchenette; well maybe I will anyway.

S: On the second floor, I think I will put something public near the elevator, so the pediatrics should be near the elevator. And the business offices (accounting) should be away from the elevator.

Q: Would it make sense to give one exam room for each doctor's office?

S: Waiting room should be close to the elevator; the bathrooms go with it. It makes more sense functionally to do that then to put them near the existing plumbing.

Q: Do I still have to maintain the corridor to the rest part of the building?

S: I would think you might want doctor's offices between the waiting area and exam rooms for a buffer for overly concerned parents and their children to be treated and examined in the exam rooms.

I see record storage.

Dirty room and clean room should be centered to the exam rooms.

He places pieces in pediatrics.

S: So now I played as an architect. I gave general arrangement for architectural purposes. (looking at pediatrics) So far I see nothing drastically at odds with HVAC, or plumbing function with this arrangement.

S: Here we are with the second floor area--accounting. This is 40 hours per week general business.

Q: Do you have a staircase? You have to.

S: Put the conference room on the corner--give it two windows. Reception area should be close to staircase. Why waste windows on a copy room?

S: I will give a brief check to see if I have ended up in a difficult mechanical design. No. I don't see any.

I: What major criteria did you consider when you laid out the things?

A: *1). Layout could be best serviced by a sensible mechanical system. I tried to cluster the 24 hour areas, around the clock occupancy areas together, and non-around-the-clock areas together. That has been achieved by this plan. Entrance/reception and urgicare, they are 24 hours. They both are on the first floor. Accounting is 40 hours per week. Pediatrics is less than around-the-clock, if more than 40 hours. I have arranged the groups according to that criteria.*

2). Clustering of plumbing functions--bathrooms, kitchenette, as much as possible to avoid having a tree of plumbing pipes, not only minimize first cost, but also ease of maintenance in the years ahead.

I also keep that criteria with some exception of the bathrooms in pediatrics area. I had to stretch the bathrooms for the pediatrics away from the existing plumbing. That's OK, because of the general vicinity of the restaurant.

Other special ventilation needs are the dirty linen rooms. No matter where they are, I can snake a small duct somewhere up to the roof; that will not be a very large duct requirement. It's very important not to let the ventilation be the determinant of the locations of dirty and clean rooms.

The HVAC, I think is rather routine.

I don't see any hot spots, such as clustering of computer rooms. This is a hospital, which under the BOCA code, will be listed under the "I" use groups (institutional sub-groups). It's likely getting into the sprinkler requirements.

I assume certain mechanical utilities make their way into the building and they are located elsewhere other than the area of this project. If that were not correct, some of the major utilities, such as gas, electric, water, just happen to enter this building complex, some of accommodations will have to be made for their entry.

Q: Is this a small part of a larger complex?

S: Utilities entrance locations--check it out!

I: Did you consider any vertical relationship between the two floors?

A: *Yes, when I tried to stack plumbing. I considered it in duct work. I have no idea what existing mechanical system is. I don't know if you have a major mechanical elsewhere with the existing duct work coming into each floor; or I don't know if you are relying on mechanical penthouse on the roof, in which case you'll need to get some duct work shaft spaces from the roof down to the lower floors traversing the second floor. You'll have to find it out for me somewhere.*

As you said, if I was free to redefine everywhere, if you supply a given area with duct work from somewhere in the center of the area then branch outward; the result will be smaller quantities of sheetmetal for the duct work; the result will be less cost mechanically. I can't comment further on that. Thinking vertically is very important. That's something a lot of young architects don't want to do because the media is two-dimensional.

Much of the HVAC system naturally is outside. You want to see the interior finishing and interior furnishings. No one wants to see duct work, pipings, wells and pumps. For buildings of this type the duct work and pipes would be located above the ceilings and what is often overlooked, when ceiling spaces discussed in a preliminary design meeting, typically clear space from ceiling to the floor above, only to be violated when the structural engineer comes by and places a 12 inch beam which could otherwise be a 14 inch beam. Structural concept has to be discussed.

Please all architects remember: I put 99% of my stuff above the ceilings.

I: Does each floor have a separate air conditioning system?

A: *Yes. The reason is different hours of operation. There is no sense of putting in all on one air-conditioning system and having to run two floors worth of air movement. And that fan energy can be added up to a big number in a year. No sense of running a large fan around the clock conditioning the area of the first floor while the second floor is only occupied 30% of the time. For that reason, I would desire them separate.*

S: I can give you more input about HVAC system. Having defined the sizes of the spaces and their functions, I could at this point give a preliminary estimate of duct work sizes, preliminary estimate of mechanical equipment sizes. You, in turn, would translate them into spaces that are required to house the equipment. Or if you don't want to build a structure to protect it, we'll have roof-top equipment.

I could at least give those to you which will achieve general size and weight, so that your structural engineer could begin to conceive the load, not just the live load of the occupancy.

If it were to be on the roof, the size will be important to you because you want to know if the size of the roof-top unit will adversely affect the elevational views of the building. So it could be of some value to you aesthetically.

Also from the preliminary size of the HVAC system, I could make an estimate of preliminary electrical requirements. The electrical engineer could use that to start his preliminary electrical design.

So we interface with each other. Architects are of central kingpin situation. Structural, mechanical and electrical engineers also interface with each other, give each other requirements, and to accommodate each other.

B.4 Subject: mechanical engineer2

Date: Nov. 22nd, 1995

S: (Looking at the list of spaces) I think I will start mechanically with the entrance / reception area. *To me this becomes an obvious thing: the main entrance is here, the main lobby should be close to the main entrance.*

Gift shop should be adjacent to the main lobby. Reception area, registration / billing, temporary children's room, men and women's bathrooms should all be around the main lobby.

S: *Since this is the ambulance entrance, I assume urgicare should go into this general area (north part).*

I: Why do you think that urgicare should go into this area?

A: *If it's urgicare sort of thing, you would want it to be the most accessible part of the building.* You would not want it to be on the second floor: you would not want to go through the main lobby, go through the elevator and get a person up there.

So I would just probably put urgicare there. I don't know if there is enough space to fit it all in, but I would attempt to put it all in there.

S: *Accounting is only for the hospital and for those people to deal with, not for patients. I would try to keep it the hardest place to get to, which should be back in this corner (north part of the second floor).*

I would put pediatrics at this end (south part of the second floor): because you've got the elevator coming up; if they keep coming in with sniffles you don't want them to be close to pregnant women that come in for urgicare.

S: *Mechanically we've got restrooms. I would probably attempt to keep my restrooms grouped somewhere over top of each other so that they could be stacked.* So we don't have a problem of plumbing coming down and squirting all over the place.

In the entrance / reception, I've got an air lock; It would go in the front. I would probably stick the gift shop somewhere in here close to the restaurant and the main lobby. Registration / billing, I would keep it at this end here (close to the existing elevator) because it can cover the pediatrics billing also.

Men and women's bathrooms, probably, it would be nice to put them near the elevator tower because usually the plumbing would be relatively close to the elevator tower. But I don't like them (men and women's bathrooms) in the main lobby. It embarrasses people to open a door in the main lobby and go to a bathroom. It would be nice if they were down in the corridor.

Temporary children's room, I would try to put it somewhere closely observed by the registration / billing.

Q: What is the function of reception?

S: So billing / registration and reception should go together, I think.

S: Mechanically you will need good exhaust out of the restrooms; that's nice because you are close to the elevator shaft which goes all the way up. You can get the duct space all the way to the rooftop with the elevator.

The restaurant has a low roof, so the exhaust can get out easily.

I would try to positively pressurize the main lobby so that when the door is open, the pressure would be out.

I would try to mechanically make sure the reception area be a separate zone, so that they could have their own thermostat. Probably registration / billing might tie in with reception area so that they could collectively be a zone if they are open with each other.

The temporary children's room should be ventilated if there is ductwork changing. You need to worry about that.

S: Urgicare, it's nice for the doctors to get out without having to go through the lobby and the waiting area. I would probably put them here (near the main lobby).

I would put waiting area somewhere up here at the front (toward ambulance entrance).

Q: Do you think people would come in here (main lobby), check in here, and go toward this direction (urgicare)?

S: That would change my plan. I will bring the waiting area down into here (close to the main lobby). I will bring my doctors offices up into this area (against the north side of the wall).

Exam rooms, I would probably make them more like a surgical suite with the rooms grouped in the center and the hallways around the outside. Nurse station, I try to put it here at the end of the exam rooms as a control point.

Clean room and dirty room. I would try to get them back here (near existing bathrooms). Bathrooms in waiting, I would try to get them here (near waiting area).

S: My thought process, mechanically, I try to put this stuff (bathrooms,... in urgicare) back here because I know I've got a lot of exhaust and with the existing bathrooms they could be stacked so that I can get my exhaust through.

The doctors offices, I try to hold them on outside wall to keep them away from the main lobby so that if there is an emergency they can get there by moving through there.

The ambulance entrance would be in close proximity to the exam rooms. I hold the exam rooms in the middle because you don't want windows in the exam rooms.It would be nice to have hallways around the exam area as lounge or waiting space. You get nice windows to look out in the hallways. So it doesn't seem like you are waiting.

The waiting room would be a zone. The exam rooms be a zone; any bacteria would be exhausted out of that area. Things will go in, not out. Extremely important, the dirty room would be exhausted under a negative pressure; the clean room would be under a positive pressure; The kitchenette would be under a slight negative pressure. The doctors offices would be a zone. You would group most of the stuff (dirty room, clean room, locker room...) into a zone.

S: Pediatrics, (going through the list), the elevator would be its entrance. You should have somebody to meet them.

I assume the billing office could be a buffer to separate the waiting for the well and the waiting for the sick.

I probably would do a similar thing to the exam rooms: put them in the center with double corridors. I would put two nurse offices one at the front end and one at the back end as controlling points.

Doctors offices, again, I would put them against the wall out here.

Record storage should be close to the billing office.

S: Let's see accounting. Maybe we may come back to pediatrics afterwards and move things around.

Q: Can I ask for another staircase?

S: I would probably have a staircase somewhere near the existing bathrooms, or close to the exterior wall with a direct door to the outside. From fire safety point of view, it would be nice.

He places pieces in accounting.

S: I would think the offices being on a separate zone. The conference room would be a separate zone by itself. Depending on what system I end up with, I would like each of these to be on a separate zone. They are on the north side of the building, so they don't get too much sun. The main accountant's office, through the western window, will get a little bit of sun in the late afternoon or evening.

I would keep reception area and copy room a separate zone.

Pressure-wise, I would tend to keep copy room under a negative pressure; kitchenette a negative pressure.

I would want to, if I could, get a wall across here between pediatrics and accounting so that pediatrics department would be separated from accounting department.

But there is a problem here: if somebody has a wheelchair and has to get to accounting, because of the location of the elevator he has to move through pediatrics to get to that.

I don't like this (location of accounting), simply because I'd like to have another elevator.

I: If you have the option of adding another elevator, where will you put it?

A: Here (adjacent to the west side of exterior wall near ambulance entrance). Because if you need to get somebody up to pediatrics area directly from the ambulance entrance, the elevator could really serve two purposes--you could do accounting and also back exit for the pediatrics department.

I: Why did you put the same-colored pieces together?

A: I want to keep my areas together. Urgicare is an area for sick people; pediatrics I like to have it as a defined department; accounting I would like to have it as a defined department. Accounting doesn't want pediatrics people to be in his department. *I would like to have each department compartmentalized so that everybody is in his own area.*

HVAC-wise, it makes enormous amount of sense. You have sick people in urgicare, sick people in entrance/reception, sick people in pediatrics, you don't want the sick people to mix in the accounting section. You would try to keep the well people from the sick people.

S: Another input: I am assuming that this is a section of a hospital. You would need to take a child from pediatrics department (or urgicare) and go to the rest part of the existing building without going to the outside.

I want to make sure that they could go to the existing building without going to the outside.

I: What major factors did you consider when you grouped the things?

A: *Taking the exam rooms for an example, it's good to have them close together. It saves travel time for the doctors. They can go from one exam room to another one without wasting time.*

From mechanical and electrical point of view, they need the same conditions, same exhaust and the same temperatures. It's good to group them together.

From the construction cost point of view, you are going to sound-proof these rooms; you are going to white them in a special way; the tighter you can group them together, the easier it is to make the adjacent things not a problem.

I: Did you consider any vertical relationship?

A: *Yes, I did. When I tried to look for mechanical stacking for plumbing and ventilation. I assume this (elevator) is a good place to take air up and to get plumbing up and down. I assume the existing bathrooms also a good place. I tried to stack the restrooms here (in urgicare on the first floor) with the restrooms here (in pediatrics on the second floor).*

Plumbing and exhaust for the mechanical system might be two reasons to think that way.

But that's not absolute. If there were spatial reasons that these would be somewhere else, that would be a problem.

I: What kind of HVAC system did you arrive at according to such a layout?

A: *With the western exposure, this is going to be colder in the morning on the west side. It seems like the building may wrap around somewhere on the east side. You don't have any eastern exposure. It doesn't seem like you are going to get much morning sun. That's why I tried to keep everything on the west side.*

S: The doctors offices could look out. The accountants offices could look out in the north side.

Q: Is there a budget? Do you have an unlimited budget or a moderate budget?

Q: Can we interface with the other things in the existing building? Or should we keep it a separate mechanical system? See, I could use the heat from the restaurant to give myself energy to heat some of these areas.

Q: Is natural gas available? Probably is down in that area.

S: *The least inexpensive way to do is by using roof-top units. I put a roof-top unit upper above here and serve down accounting and urgicare.*

S: *A better way to do this is to use a heat pump loop, which gives us the same ability. We can put a separate unit for each zone, or even a separate unit in each room.*

Once people get here, the light will go on. You will be on air-conditioning mode 80-90% of all the occupied hours. If you are on air-conditioning mode, you need to bring in a lot of outside air.

There will be more people in pediatrics and fewer people in accounting. So pediatrics will have high people load and high lighting load. Accounting will have relatively low people load and so on. This unit is on cooling (pediatrics) and this unit is on heating (accounting). So the BTUs from this unit (pediatrics) could be transferred to this unit (pediatrics). You don't need to go to the gas company or electric company to buy more BTUs.

So first cost of heat pump loop will be higher; but the operating cost will be lower.

So each area will be a separate HVAC system; with a minimum of two systems on each floor, depending on the sub-zones.

I: Daylight factor decides which side you put things, right?

A: Yes. Also I put things adjacent to exterior walls for psychological reasons. But for my exam rooms, we don't need windows. But it would be nice for the waiting area (hallways surrounding the exam rooms) to have windows.

S: *If I were an architect, this (the layouts of the first and the second floors) is what I would do.*

I: Did you consider fresh air?

A: *Yes. Because all the area, particularly the area with negative pressure will need enormous amount of fresh air.* With VVT up on the roof, you can bring in 100% outside air. But with heat pump loop, it's a little harder.

Appendix C:
Test Cases

C.1 Brooklyn Jail

Table C.1 Brooklyn Jail: stacking according to adjacency (State id: 98, Total cut size: 178)

Floor 1			**Assignments matching existing placements**	**Assignments differing from existing placements**
Sum of FUs' area: 23321	Available area: -1	Number of FUs: 18		LCK1_E_F9
				CLL2_E_F9
				CLL1_E_F9
				DAY2_M_F9
				INTW_M_F9
				CTRL_M_F9
			REST_E_F1	
			CELL_E_F1	
			RECV_E_F1	
			WARD_E_F1	
			PUB_E_F1	
			LBBY_M_F1	
			CSLT_M_F1	
			STOR_M_F1	
			VWTG_W_F1	
			LOCK_M_F1	
			YARD_F1	
			DOC_E_F1	
Number of FUs			12	6

Floor 2			**Assignments matching existing placements**	**Assignments differing from existing placements**
Sum of FUs' area: 6456	Available area: 0	Number of FUs: 15	LCK1_W_F2	
			CLN_W_F2	
			UTL_W_F2	
			CLL1_W_F2	
			LCK2_W_F2	
			LCK2_E_F2	
			LCK1_E_F2	
			CLL2_E_F2	
			CLL1_E_F2	
			DAY2_M_F2	
			INTW_M_F2	
			CTRL_M_F2	
			CLL2_W_F2	
			DAY1_M_F2	
			UTL_E_F2	
Number of FUs			15	0

Floor 3			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 6456	Available area: 0	Number of FUs: 15	LCK1_W_F3	
			CLN_W_F3	
			UTL_W_F3	
			CLL1_W_F3	
			LCK2_W_F3	
			LCK2_E_F3	
			LCK1_E_F3	
			CLL2_E_F3	
			CLL1_E_F3	
			DAY2_M_F3	
			INTW_M_F3	
			CTRL_M_F3	
			CLL2_W_F3	
			DAY1_M_F3	
			UTL_E_F3	
Number of FUs			15	0

Floor 4			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 6456	Available area: 0	Number of FUs: 15	LCK1_W_F4	
			CLN_W_F4	
			UTL_W_F4	
			CLL1_W_F4	
			LCK2_W_F4	
			LCK2_E_F4	
			LCK1_E_F4	
			CLL2_E_F4	
			CLL1_E_F4	
			DAY2_M_F4	
			INTW_M_F4	
			CTRL_M_F4	
			CLL2_W_F4	
			DAY1_M_F4	
			UTL_E_F4	
Number of FUs			15	0

Floor 5			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 6456	Available area: 0	Number of FUs: 15	LCK1_W_F5	
			CLN_W_F5	
			UTL_W_F5	
			CLL1_W_F5	
			LCK2_W_F5	
			LCK2_E_F5	
			LCK1_E_F5	
			CLL2_E_F5	
			CLL1_E_F5	
			DAY2_M_F5	
			INTW_M_F5	
			CTRL_M_F5	
			CLL2_W_F5	
			DAY1_M_F5	
			UTL_E_F5	
Number of FUs			15	0

Floor 6			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 6456	Available area: 0	Number of FUs: 15	LCK1_W_F6	
			CLN_W_F6	
			UTL_W_F6	
			CLL1_W_F6	
			LCK2_W_F6	
			LCK2_E_F6	
			LCK1_E_F6	
			CLL2_E_F6	
			CLL1_E_F6	
			DAY2_M_F6	
			INTW_M_F6	
			CTRL_M_F6	
			CLL2_W_F6	
			DAY1_M_F6	
			UTL_E_F6	
Number of FUs			15	0

Floor 7			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 6456	Available area: 0	Number of FUs: 15	LCK1_W_F7	
			CLN_W_F7	
			UTL_W_F7	
			CLL1_W_F7	
			LCK2_W_F7	
			LCK2_E_F7	
			LCK1_E_F7	
			CLL2_E_F7	
			CLL1_E_F7	
			DAY2_M_F7	
			INTW_M_F7	
			CTRL_M_F7	
			CLL2_W_F7	
			DAY1_M_F7	
			UTL_E_F7	
Number of FUs			15	0

Floor 8			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 6456	Available area: 0	Number of FUs: 15	LCK1_W_F8	
			CLN_W_F8	
			UTL_W_F8	
			CLL1_W_F8	
			LCK2_W_F8	
			LCK2_E_F8	
			LCK1_E_F8	
			CLL2_E_F8	
			CLL1_E_F8	
			DAY2_M_F8	
			INTW_M_F8	
			CTRL_M_F8	
			CLL2_W_F8	
			DAY1_M_F8	
			UTL_E_F8	
Number of FUs			15	0

<table>
<tr><th colspan="3">Floor 9</th><th>Assignments matching existing placements</th><th>Assignments differing from existing placements</th></tr>
<tr><td rowspan="16">Sum of FUs' area: 6455</td><td rowspan="16">Available area: 1</td><td rowspan="16">Number of FUs: 16</td><td>LCK2_F9</td><td></td></tr>
<tr><td>LCK1_W_F9</td><td></td></tr>
<tr><td>CLN_W_F9</td><td></td></tr>
<tr><td>UTL_W_F9</td><td></td></tr>
<tr><td>CLL1_W_F9</td><td></td></tr>
<tr><td>LCK2_W_F9</td><td></td></tr>
<tr><td></td><td>SOCL_W_F1</td></tr>
<tr><td></td><td>CAPT_W_F1</td></tr>
<tr><td></td><td>CTRL_W_F1</td></tr>
<tr><td></td><td>VSIT_W_F1</td></tr>
<tr><td></td><td>BTHS_W_F1</td></tr>
<tr><td></td><td>GARD_W_F1</td></tr>
<tr><td></td><td>DPUT_W_F1</td></tr>
<tr><td>CLL2_W_F9</td><td></td></tr>
<tr><td>DAY1_M_F9</td><td></td></tr>
<tr><td>UTL_E_F9</td><td></td></tr>
<tr><td colspan="3">Number of FUs</td><td>9</td><td>7</td></tr>
</table>

C.2 Kaiser Office Building

Table C.2 Kaiser: stacking according to adjacency (State id: 93, Total cut size: 270)

<table>
<tr><th colspan="3">Floor 1</th><th>Assignments matching existing placements</th><th>Assignments differing from existing placements</th></tr>
<tr><td rowspan="15">Sum of FUs' area: 34527</td><td rowspan="15">Available area: 385</td><td rowspan="15">Number of FUs: 15</td><td>RTL1_W_F1</td><td></td></tr>
<tr><td>LBY1_W_F1</td><td></td></tr>
<tr><td>LBY5_W_F1</td><td></td></tr>
<tr><td>BANK_E_F1</td><td></td></tr>
<tr><td>SHP4_E_F1</td><td></td></tr>
<tr><td>SHP3_E_F1</td><td></td></tr>
<tr><td>SHP2_E_F1</td><td></td></tr>
<tr><td>SHP1_E_F1</td><td></td></tr>
<tr><td>STOR_E_F1</td><td></td></tr>
<tr><td></td><td>REST_E_F28</td></tr>
<tr><td></td><td>REST_W_F28</td></tr>
<tr><td></td><td>ELEV_F28</td></tr>
<tr><td>LBY2_W_F1</td><td></td></tr>
<tr><td>LBY_M_F1</td><td></td></tr>
<tr><td>LBY_E_F1</td><td></td></tr>
<tr><td colspan="3">Number of FUs</td><td>12</td><td>3</td></tr>
</table>

<table>
<tr><th colspan="3">Floor 2</th><th>Assignments matching existing placements</th><th>Assignments differing from existing placements</th></tr>
<tr><td rowspan="9">Sum of FUs' area: 52410</td><td rowspan="9">Available area: 0</td><td rowspan="9">Number of FUs: 9</td><td>CAF_E_F2</td><td></td></tr>
<tr><td>SERV_E_F2</td><td></td></tr>
<tr><td>OFF_E_F2</td><td></td></tr>
<tr><td>AUD_M_F2</td><td></td></tr>
<tr><td>OFF2_W_F2</td><td></td></tr>
<tr><td>OFF1_W_F2</td><td></td></tr>
<tr><td>MED_W_F2</td><td></td></tr>
<tr><td>LBY_M_F2</td><td></td></tr>
<tr><td>KIT_E_F2</td><td></td></tr>
<tr><td colspan="3">Number of FUs</td><td>9</td><td>0</td></tr>
</table>

Floor 3			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F3	
			REST_W_F3	
			ELEV_F3	
			OFF_F3	
Number of FUs			4	0

Floor 4			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F4	
			REST_W_F4	
			ELEV_F4	
			OFF_F4	
Number of FUs			4	0

Floor 5			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F5	
			REST_W_F5	
			ELEV_F5	
			OFF_F5	
Number of FUs			4	0

Floor 6			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F6	
			REST_W_F6	
			ELEV_F6	
			OFF_F6	
Number of FUs			4	0

Floor 7			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F7	
			REST_W_F7	
			ELEV_F7	
			OFF_F7	
Number of FUs			4	0

Floor 8			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F8	
			REST_W_F8	
			ELEV_F8	
			OFF_F8	
Number of FUs			4	0

Floor 9			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F9	
			REST_W_F9	
			ELEV_F9	
			OFF_F9	
Number of FUs			4	0

Floor 10			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F10	
			REST_W_F10	
			ELEV_F10	
			OFF_F10	
Number of FUs			4	0

Floor 11			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F11	
			REST_W_F11	
			ELEV_F11	
			OFF_F11	
Number of FUs			4	0

Floor 12			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F12	
			REST_W_F12	
			ELEV_F12	
			OFF_F12	
Number of FUs			4	0

Floor 13			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F13	
			REST_W_F13	
			ELEV_F13	
			OFF_F13	
Number of FUs			4	0

Floor 14			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F14	
			REST_W_F14	
			ELEV_F14	
			OFF_F14	
Number of FUs			4	0

Floor 15			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F15	
			REST_W_F15	
			ELEV_F15	
			OFF_F15	
Number of FUs			4	0

Floor 16			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F16	
			REST_W_F16	
			ELEV_F16	
			OFF_F16	
Number of FUs			4	0

Floor 17			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F17	
			REST_W_F17	
			ELEV_F17	
			OFF_F17	
Number of FUs			4	0

Floor 18			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F18	
			REST_W_F18	
			ELEV_F18	
			OFF_F18	
Number of FUs			4	0

Floor 19			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F19	
			REST_W_F19	
			ELEV_F19	
			OFF_F19	
Number of FUs			4	0

Floor 20			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F20	
			REST_W_F20	
			ELEV_F20	
			OFF_F20	
Number of FUs			4	0

Floor 21			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F21	
			REST_W_F21	
			ELEV_F21	
			OFF_F21	
Number of FUs			4	0

Floor 22			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F22	
			REST_W_F22	
			ELEV_F22	
			OFF_F22	
Number of FUs			4	0

Floor 23			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F23	
			REST_W_F23	
			ELEV_F23	
			OFF_F23	
Number of FUs			4	0

Floor 24			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F24	
			REST_W_F24	
			ELEV_F24	
			OFF_F24	
Number of FUs			4	0

Floor 25			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F25	
			REST_W_F25	
			ELEV_F25	
			OFF_F25	
Number of FUs			4	0

Floor 26			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F26	
			REST_W_F26	
			ELEV_F26	
			OFF_F26	
Number of FUs			4	0

Floor 27			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29035	Available area: 0	Number of FUs: 4	REST_E_F27	
			REST_W_F27	
			ELEV_F27	
			OFF_F27	
Number of FUs			4	0

Floor 28			Assignments matching existing placements	Assignments differing from existing placements
Sum of FUs' area: 29420	Available area: -385	Number of FUs: 7		LBY4_W_F1
				LBY3_W_F1
				RTL5_W_F1
				RTL4_W_F1
				RTL3_W_F1
				RTL2_W_F1
			OFF_F28	
Number of FUs			1	6

C.3 Falk Clinic

Table C.3 Falk Clinic: stacking and zoning according to adjacency (State id: 283, Total cut size: 1381)[1]

Stacking			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Floor 1							
Sum of FUs' area: 7476	Available area: 0	Number of FUs: 45	STAFF_E_F1		WORK_AREA_E_F1	Sum of FUs' area: 1726	Number of FUs: 17
			PERSONNEL2_E_F1		EXAM1_E_F1		
			PERSONNEL1_E_F1		EXAM2_E_F1		
			RESTRM_E_F1		EXAM3_E_F1		
			PATIENT_ASSESS_E_F1		EXAM4_E_F1		
			WAITING_E_F1		EXAM5_E_F1		
			WORK_STATION_E_F1		EXAM6_E_F1		
			EXAM8_E_F1		EXAM7_E_F1		
			EXAM7_E_F1		EXAM8_E_F1		
			EXAM6_E_F1		WORK_STATION_E_F1		
			EXAM5_E_F1		CHECK_IN_E_F1		
			EXAM4_E_F1		WAITING_E_F1		
			EXAM3_E_F1		PATIENT_ASSESS_E_F1		
			EXAM2_E_F1		RESTRM_E_F1		
			EXAM1_E_F1		PERSONNEL1_E_F1		
			WORK_AREA_E_F1		PERSONNEL2_E_F1		
			WAITING_W_F1		STAFF_E_F1		
			CHECK_IN_W_F1		EXAM1_W_F1	Sum of FUs' area: 1820	Number of FUs: 15
			NURSE_OFF_W_F1		EXAM2_W_F1		
			EXAM11_W_F1		EXAM3_W_F1		
			EXAM10_W_F1		EXAM4_W_F1		
			EXAM9_W_F1		EXAM5_W_F1		
			EXAM8_W_F1		EXAM6_W_F1		
			EXAM7_W_F1		EXAM7_W_F1		
			EXAM6_W_F1		EXAM8_W_F1		
			EXAM4_W_F1		EXAM9_W_F1		
			EXAM3_W_F1		EXAM10_W_F1		
			EXAM2_W_F1		EXAM11_W_F1		
			PERSONNEL_W_F1		NURSE_OFF_W_F1		
			EXAM1_W_F1		CHECK_IN_W_F1		
			REGISTR_M_F1		WAITING_W_F1		
			RESEARCH_M_F1		PERSONNEL_W_F1		
			PERSONNEL2_M_F1		RESEARCH_M_F1	Sum of FUs' area: 1430	Number of FUs: 12
			RESTRM_M_F1		OFF1_M_F1		
			LAB_M_F1		SOCIAL_WORKER_M_F1		
			EXAM2_M_F1		OFF2_M_F1		
			EXAM1_M_F1		PERSONNEL1_M_F1		
			PERSONNEL1_M_F1		CONF_M_F1		
			SOCIAL_WORKER_M_F1		EXAM1_M_F1		
			OFF2_M_F1		EXAM2_M_F1		
			OFF1_M_F1		LAB_M_F1		
			LOBBY_F1		RESTRM_M_F1		
			EXAM5_W_F1		PERSONNEL2_M_F1		
			CONF_M_F1		REGISTR_M_F1		
			CHECK_IN_E_F1		LOBBY_F1		
Number of FUs			45	0			

1. Zoning threshold is set to 10, equivalent to an adjacency weight.

Stacking Floor 2			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Sum of FUs' area: 6718	Available area: 0	Number of FUs: 39	OFF2_W_F2		OFF1_W_F2	Sum of FUs' area: 2284	Number of FUs: 22
			OFF3_W_F2		OFF2_W_F2		
			EXAM1_W_F2		OFF3_W_F2		
			EXAM2_W_F2		EXAM1_W_F2		
			EXAM3_W_F2		EXAM2_W_F2		
			EXAM4_W_F2		EXAM3_W_F2		
			EXAM5_W_F2		EXAM4_W_F2		
			EXAM6_W_F2		EXAM5_W_F2		
			EXAM7_W_F2		EXAM6_W_F2		
			EXAM8_W_F2		EXAM7_W_F2		
			CONF_RM_W_F2		EXAM8_W_F2		
			LAB_W_F2		CONF_RM_W_F2		
			CONSULT1_W_F2		RECEPTION_W_F2		
			CONSULT2_W_F2		LAB_W_F2		
			MED_REC_W_F2		CONSULT1_W_F2		
			CHECK_IN_W_F2		CONSULT2_W_F2		
			MEN_W_F2		MED_REC_W_F2		
			WOMEN_W_F2		CHECK_IN_W_F2		
			REGISTR_W_F2		MEN_W_F2		
			OFF1_W_F2		WOMEN_W_F2		
			FINANCIAL_W_F2		REGISTR_W_F2		
			HANDICAP_E_F2		FINANCIAL_W_F2		
			PERSONNEL1_E_F2		HANDICAP_E_F2	Sum of FUs' area: 1934	Number of FUs: 16
			PERSONNEL2_E_F2		PERSONNEL1_E_F2		
			PERSONNEL3_E_F2		PERSONNEL2_E_F2		
			PERSONNEL4_E_F2		PERSONNEL3_E_F2		
			EXAM1_E_F2		PERSONNEL4_E_F2		
			EXAM2_E_F2		EXAM1_E_F2		
			OFF1_E_F2		EXAM2_E_F2		
			OFF2_E_F2		OFF1_E_F2		
			PHARMD2_E_F2		OFF2_E_F2		
			STAFF_E_F2		PHARMD1_E_F2		
			STORAGE_E_F2		PHARMD2_E_F2		
			RECEPT_E_F2		STAFF_E_F2		
			PHARMACY_E_F2		STORAGE_E_F2		
			CASHIER_E_F2		RECEPT_E_F2		
			INFO_F2		PHARMACY_E_F2		
			RECEPTION_W_F2		CASHIER_E_F2		
			PHARMD1_E_F2		INFO_F2		
Number of FUs			39	0			

Stacking			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Floor 3							
Sum of FUs' area: 6295	Available area: -0.000488281	Number of FUs: 53	EXAM1_E_F3		RESTRM1_E_F3	Sum of FUs' area: 1462	Number of FUs: 18
			EXAM2_E_F3		EXAM1_E_F3		
			CT_CONTROL_E_F3		EXAM2_E_F3		
			CT_SCAN_E_F3		CT_CONTROL_E_F3		
			FILE_SORTING_E_F3		CT_SCAN_E_F3		
			RESTRM2_E_F3		FILE_SORTING_E_F3		
			DRESS_E_F3		RESTRM2_E_F3		
			STORAGE1_E_F3		DRESS_E_F3		
			MEN_E_F3		STORAGE1_E_F3		
			WOMEN_E_F3		MEN_E_F3		
			STORAGE2_E_F3		WOMEN_E_F3		
			FREEZER_E_F3		PHLEBOTOMY_E_F3		
			STORAGE3_E_F3		STORAGE2_E_F3		
			STAFF_E_F3		FREEZER_E_F3		
			RESTRM3_E_F3		STORAGE3_E_F3		
			RESTRM1_E_F3		STAFF_E_F3		
			RESTRM4_E_F3		RESTRM3_E_F3		
			STAFF1_S_F3		RESTRM4_E_F3		
			EXAM1_S_F3		STAFF1_S_F3	Sum of FUs' area: 1451	Number of FUs: 16
			EXAM2_S_F3		EXAM1_S_F3		
			STAFF2_S_F3		EXAM2_S_F3		
			RESTRM1_S_F3		STAFF2_S_F3		
			STORAGE1_S_F3		RESTRM1_S_F3		
			READING_S_F3		STORAGE1_S_F3		
			DARK_S_F3		READING_S_F3		
			UTILITY_S_F3		DARK_S_F3		
			STORAGE2_S_F3		UTILITY_S_F3		
			LOCKER1_S_F3		STORAGE2_S_F3		
			LOCKER2_S_F3		MED_RECORDS_S_F3		
			RESTRM2_S_F3		LOCKER1_S_F3		
			SUPPLY_S_F3		LOCKER2_S_F3		
			RESEARCH_S_F3		RESTRM2_S_F3		
			STAFF_W_F3		SUPPLY_S_F3		
			EXAM_A_W_F3		RESEARCH_S_F3		
			EXAM_B_W_F3		STAFF_W_F3	Sum of FUs' area: 1752	Number of FUs: 14
			EXAM_C_W_F3		EXAM_A_W_F3		
			EXAM_D_W_F3		EXAM_B_W_F3		
			EXAM_E_W_F3		EXAM_C_W_F3		
			EXAM_F_W_F3		EXAM_D_W_F3		
			PERSONNEL1_W_F3		EXAM_E_W_F3		
			EXAM1_W_F3		EXAM_F_W_F3		
			EXAM3_W_F3		PERSONNEL1_W_F3		
			PERSONNEL2_W_F3		EXAM1_W_F3		
			RESTROOM_W_F3		EXAM2_W_F3		
			WAITING_W_F3		EXAM3_W_F3		
			VASC_LAB_N_F3		PERSONNEL2_W_F3		
			PERSONNEL1_N_F3		RESTROOM_W_F3		
			REGISTRATION_N_F3		WAITING_W_F3		
			PERSONNEL2_N_F3		PERSONNEL1_N_F3	Sum of FUs' area: 1630	Number of FUs: 5
			EXAM2_W_F3		REGISTRATION_N_F3		
			AMBULATORY_N_F3		AMBULATORY_N_F3		
			PHLEBOTOMY_E_F3		PERSONNEL2_N_F3		
			MED_RECORDS_S_F3		VASC_LAB_N_F3		
Number of FUs			53	0			

Stacking			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Floor 4							
Sum of FUs' area:5782	Available area: 0	Number of FUs: 52	OFF16_S_F4		OFF1_S_F4	Sum of FUs' area: 2029	Number of FUs: 21
			OFF15_S_F4		OFF2_S_F4		
			OFF14_S_F4		OFF3_S_F4		
			OFF13_S_F4		OFF4_S_F4		
			OFF12_S_F4		OFF5_S_F4		
			OFF11_S_F4		OFF6_S_F4		
			OFF10_S_F4		OFF7_S_F4		
			OFF9_S_F4		UTILITY_S_F4		
			KITCHEN_S_F4		SUPPLY_S_F4		
			CHECK_IN_S_F4		WORK_AREA_S_F4		
			OFF8_S_F4		OFF8_S_F4		
			SUPPLY_S_F4		CHECK_IN_S_F4		
			UTILITY_S_F4		KITCHEN_S_F4		
			OFF7_S_F4		OFF9_S_F4		
			OFF6_S_F4		OFF10_S_F4		
			OFF5_S_F4		OFF11_S_F4		
			OFF4_S_F4		OFF12_S_F4		
			OFF3_S_F4		OFF13_S_F4		
			OFF2_S_F4		OFF14_S_F4		
			OFF1_S_F4		OFF15_S_F4		
			EXAM9_E_F4		OFF16_S_F4		
			EXAM8_E_F4		TREATMENT_E_F4	Sum of FUs' area: 1668	Number of FUs: 17
			EXAM7_E_F4		WAITING1_E_F4		
			EXAM6_E_F4		CONF1_E_F4		
			EXAM5_E_F4		CONF2_E_F4		
			EXAM4_E_F4		CONF3_E_F4		
			EXAM3_E_F4		WAITING2_E_F4		
			EXAM2_E_F4		CONSULT1_E_F4		
			EXAM1_E_F4		CONSULT2_E_F4		
			CONSULT2_E_F4		EXAM1_E_F4		
			CONSULT1_E_F4		EXAM2_E_F4		
			WAITING2_E_F4		EXAM3_E_F4		
			CONF3_E_F4		EXAM4_E_F4		
			CONF2_E_F4		EXAM5_E_F4		
			CONF1_E_F4		EXAM6_E_F4		
			TREATMENT_E_F4		EXAM7_E_F4		
			OFF9_W_F4		EXAM8_E_F4		
			OFF8_W_F4		EXAM9_E_F4		
			OFF7_W_F4		OFF1_W_F4	Sum of FUs' area: 1452	Number of FUs: 10
			OFF6_W_F4		OFF2_W_F4		
			OFF5_W_F4		OFF3_W_F4		
			OFF4_W_F4		OFF4_W_F4		
			OFF2_W_F4		OFF5_W_F4		
			OFF1_W_F4		OFF6_W_F4		
			FILE_W_F4		OFF7_W_F4		
			OFF1_N_F4		OFF8_W_F4		
			OFF2_N_F4		OFF9_W_F4		
			RESTRM_N_F4		FILE_W_F4		
			OFF3_W_F4		OFF1_N_F4	Sum area: 633	Number of FUs: 4
			STAFF_N_F4		OFF2_N_F4		
			WAITING1_E_F4		RESTRM_N_F4		
			WORK_AREA_S_F4		STAFF_N_F4		
Number of FUs			52	0			

Stacking Floor 5			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Sum of FUs' area: 6284	Available area: 278	Number of FUs: 59	EXAM1_S_F5		PERSONNEL1_N_F7	Sum of FUs' area: 1658	Number of FUs: 15
				EXAM2_N_F7	OFF_N_F7		
				PERSONNEL4_N_F7	STAFF_N_F7		
				EXAM1_N_F7	RESTRM_N_F7		
				TELEPHONE_N_F7	SECRETARY_N_F7		
				PERSONNEL3_N_F7	PERSONNEL_N_F7		
				WAITING_N_F7	ASSESSMENT_N_F7		
				RECEPT_N_F7	OFF2_N_F7		
				OFF2_N_F7	RECEPT_N_F7		
				ASSESSMENT_N_F7	WAITING_N_F7		
				PERSONNEL_N_F7	PERSONNEL3_N_F7		
				SECRETARY_N_F7	TELEPHONE_N_F7		
				RESTRM_N_F7	EXAM1_N_F7		
				STAFF_N_F7	PERSONNEL4_N_F7		
				OFF_N_F7	EXAM2_N_F7		
				PERSONNEL1_N_F7	LASER_E_F5	Sum of FUs' area: 1680	Number of FUs: 17
			APPOINTMENT_E_F5		PERSONNEL_E_F5		
			MED_RECORDS_E_F5		LAB1_E_F5		
			EXAM9_E_F5		EXAM1_E_F5		
			EXAM8_E_F5		EXAM2_E_F5		
			EXAM7_E_F5		EXAM3_E_F5		
			EXAM6_E_F5		EXAM4_E_F5		
			STAFF_E_F5		PHOTO_E_F5		
			EXAM5_E_F5		EXAM5_E_F5		
			PHOTO_E_F5		STAFF_E_F5		
			EXAM4_E_F5		EXAM6_E_F5		
			EXAM3_E_F5		EXAM7_E_F5		
			EXAM2_E_F5		EXAM8_E_F5		
			EXAM1_E_F5		EXAM9_E_F5		
			LAB1_E_F5		LAB2_E_F5		
			PERSONNEL_E_F5		MED_RECORDS_E_F5		
			LASER_E_F5		APPOINTMENT_E_F5		
				OFF2_W_F7	CONF2_W_F7	Sum area: 380	Number of FUs: 4
				EXAM5_W_F7	EXAM4_W_F7		
				EXAM4_W_F7	EXAM5_W_F7		
				CONF2_W_F7	OFF2_W_F7		
			OFF1_W_F5		OFF1_W_F5	Sum of FUs' area: 1560	Number of FUs: 14
			OFF2_W_F5		OFF2_W_F5		
			OFF3_W_F5		OFF3_W_F5		
			OFF4_W_F5		OFF4_W_F5		
			OFF5_W_F5		OFF5_W_F5		
			PRIVATE1_W_F5		MD_OFF_W_F5		
			PRIVATE2_W_F5		PRIVATE1_W_F5		
			PERSONNEL_W_F5		PRIVATE2_W_F5		
			PRIVATE3_W_F5		PERSONNEL_W_F5		
			OFF6_W_F5		PRIVATE3_W_F5		
			OFF7_W_F5		OFF6_W_F5		
			OFF8_W_F5		OFF7_W_F5		
			WORKSTATION_W_F5		OFF8_W_F5		
			OFF2_N_F5		WORKSTATION_W_F5		
			MEN_N_F5		MEN_N_F5	Sum of FUs' area: 808	Number of FUs: 7
			PERSONNEL2_N_F5		WOMEN_N_F5		
			OFF1_N_F5		PERSONNEL1_N_F5		
			PERSONNEL1_N_F5		CONF_N_F5		
			WOMEN_N_F5		OFF1_N_F5		
			MD_OFF_W_F5		PERSONNEL2_N_F5		
			CONF_N_F5		OFF2_N_F5		
			LAB2_E_F5		EXAM1_S_F5	Area: 198	2
			OFF3_S_F5		OFF3_S_F5		
Number of FUs			40	19			

Stacking			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Floor 6							
Sum of FUs' area: 6561	Available area: -24.9995	Number of FUs: 58		EXAM3_W_F7	NURSE_W_F7	Sum of FUs' area: 1232	Number of FUs: 9
				OFF1_W_F7	WORKAREA_W_F7		
				PERSONNEL_W_F7	EXAM1_W_F7		
				KITCHEN_W_F7	CONF1_W_F7		
				EXAM2_W_F7	EXAM2_W_F7		
				CONF1_W_F7	KITCHEN_W_F7		
				EXAM1_W_F7	PERSONNEL_W_F7		
				WORKAREA_W_F7	OFF1_W_F7		
				NURSE_W_F7	EXAM3_W_F7		
			OFF1_W_F6		OFF1_W_F6	Sum of FUs' area: 1498	Number of FUs: 16
			EXAM1_W_F6		EXAM1_W_F6		
			EXAM2_W_F6		EXAM2_W_F6		
			STAFF1_W_F6		STAFF1_W_F6		
			OFF2_W_F6		OFF2_W_F6		
			EXAM3_W_F6		EXAM3_W_F6		
			EXAM4_W_F6		EXAM4_W_F6		
			RESTRM1_W_F6		RESTRM1_W_F6		
			EXAM5_W_F6		EXAM5_W_F6		
			EXAM6_W_F6		EXAM6_W_F6		
			STAFF2_W_F6		STAFF2_W_F6		
			RESTRM2_W_F6		WAITING_W_F6		
			EXAM7_W_F6		RESTRM2_W_F6		
			EXAM8_W_F6		EXAM7_W_F6		
			EXAM9_W_F6		EXAM8_W_F6		
			STAFF_E_F6		EXAM9_W_F6		
			RESTRM2_E_F6		DRUG_E_F7	Sum of FUs' area: 1249	Number of FUs: 10
			KITCHEN_E_F6		RESEARCH1_E_F7		
			RESTRM1_E_F6		LAB_E_F7		
			EXAM6_E_F6		OFF_E_F7		
			EXAM5_E_F6		RESTRM_E_F7		
			EXAM4_E_F6		TREATMENT_E_F7		
			EXAM3_E_F6		EXAM_E_F7		
			EXAM2_E_F6		RESEARCH2_E_F7		
			MED_RECORDS_E_F6		WORKAREA_E_F7		
			EXAM1_E_F6		PERSONNEL_E_F7		
			OFF_E_F6		WAITING_N_F6	Sum of FUs' area: 916	Number of FUs: 8
			PERSONNEL_E_F6		MEN_N_F6		
				DRUG_E_F7	PERSONNEL1_N_F6		
				RESEARCH1_E_F7	OFF_N_F6		
				LAB_E_F7	CONF_N_F6		
				OFF_E_F7	STAFF_N_F6		
				RESTRM_E_F7	PERSONNEL2_N_F6		
				TREATMENT_E_F7	WOMEN_N_F6		
				EXAM_E_F7	PERSONNEL_E_F6	Sum of FUs' area: 1546	Number of FUs: 14
				RESEARCH2_E_F7	OFF_E_F6		
				WORKAREA_E_F7	EXAM1_E_F6		
				PERSONNEL_E_F7	MED_RECORDS_E_F6		
			WOMEN_N_F6		EXAM2_E_F6		
			WAITING_N_F6		EXAM3_E_F6		
			MEN_N_F6		EXAM4_E_F6		
			PERSONNEL1_N_F6		EXAM5_E_F6		
			OFF_N_F6		EXAM6_E_F6		
			STAFF_N_F6		RESTRM1_E_F6		
			PERSONNEL2_N_F6		KITCHEN_E_F6		
			WAITING_W_F6		RESTRM2_E_F6		
			CONF_N_F6		STAFF_E_F6		
			RECEPTION_E_F6		RECEPTION_E_F6		
			MED_RECORDS_S_F6		MED_RECORDS_S_F6		
Number of FUs			39	19			

Stacking			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Floor 7							
Sum of FUs' area: 5006	Available area: -253	Number of FUs: 45		OFF4_S_F6	OFF1_S_F6	Sum of FUs' area: 2456	Number of FUs: 21
				EXAM13_S_F6	EXAM1_S_F6		
				EXAM1_S_F6	EXAM2_S_F6		
				EXAM12_S_F6	EXAM3_S_F6		
				EXAM2_S_F6	EXAM4_S_F6		
				RECOVERY_S_F6	EXAM5_S_F6		
				EXAM3_S_F6	OFF2_S_F6		
				EXAM4_S_F6	EXAM6_S_F6		
				EXAM11_S_F6	PROCEDURE_S_F6		
				EXAM5_S_F6	WAITING_S_F6		
				OFF3_S_F6	WORKAREA_S_F6		
				OFF2_S_F6	EXAM7_S_F6		
				EXAM10_S_F6	EXAM8_S_F6		
				EXAM6_S_F6	EXAM9_S_F6		
				EXAM9_S_F6	EXAM10_S_F6		
				PROCEDURE_S_F6	OFF3_S_F6		
				EXAM8_S_F6	EXAM11_S_F6		
				WAITING_S_F6	RECOVERY_S_F6		
				EXAM7_S_F6	EXAM12_S_F6		
				WORKAREA_S_F6	EXAM13_S_F6		
				OFF1_S_F6	OFF4_S_F6		
				CONF2_S_F5	EXAM2_S_F5	Sum of FUs' area: 2316	Number of FUs: 21
				PRIVATE2_S_F5	EXAM3_S_F5		
				EXAM2_S_F5	EXAM4_S_F5		
				OFF11_S_F5	EXAM5_S_F5		
				EXAM3_S_F5	OFF1_S_F5		
				OFF10_S_F5	EXAM6_S_F5		
				EXAM4_S_F5	RESIDENCE_S_F5		
				OFF9_S_F5	OFF2_S_F5		
				EXAM5_S_F5	OFF4_S_F5		
				OFF8_S_F5	CONF1_S_F5		
				OFF1_S_F5	PRIVATE1_S_F5		
				OFF7_S_F5	OFF5_S_F5		
				EXAM6_S_F5	OFF6_S_F5		
				RESTRM_S_F5	RESTRM_S_F5		
				RESIDENCE_S_F5	OFF7_S_F5		
				OFF6_S_F5	OFF8_S_F5		
				OFF2_S_F5	OFF9_S_F5		
				OFF5_S_F5	OFF10_S_F5		
				PRIVATE1_S_F5	OFF11_S_F5		
				CONF1_S_F5	PRIVATE2_S_F5		
				OFF4_S_F5	CONF2_S_F5		
			FILE_W_F7		FILE_W_F7		
			SUPPLY_N_F7		SUPPLY_N_F7		
			BLOOD_E_F7		BLOOD_E_F7		
Number of FUs			3	42			

C.3.1 Stacking according to Multiple Criteria

Table C.4 Falk Clinic: stacking and zoning according to multiple criteria (State id: 284, Total cut size: 37789)[1]

Stacking Floor 1			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Sum of FUs' area: 7486	Available area: -10	Number of FUs: 46	EXAM4_E_F1		CHECK_IN_E_F1	Sum of FUs' area: 540	Number of FUs: 5
			EXAM3_E_F1		EXAM1_E_F1		
			EXAM2_E_F1		EXAM2_E_F1		
			EXAM1_E_F1		EXAM3_E_F1		
				SECRETARY_N_F7	EXAM4_E_F1		
				STAFF_N_F7	OFF_N_F7	Sum of FUs' area: 650	Number of FUs: 7
				EXAM2_N_F7	STAFF_N_F7		
				EXAM1_N_F7	SECRETARY_N_F7		
				OFF2_N_F7	EXAM2_N_F7		
				ASSESSMENT_N_F7	OFF2_N_F7		
				OFF_N_F7	ASSESSMENT_N_F7		
				STAFF_E_F6	EXAM1_N_F7		
				EXAM6_E_F6	EXAM2_E_F6	Sum of FUs' area: 546	Number of FUs: 6
				EXAM5_E_F6	EXAM3_E_F6		
				EXAM4_E_F6	EXAM4_E_F6		
				EXAM3_E_F6	EXAM5_E_F6		
				EXAM2_E_F6	EXAM6_E_F6		
			NURSE_OFF_W_F1		STAFF_E_F6		
			EXAM11_W_F1		PERSONNEL_W_F1	Sum of FUs' area: 1820	Number of FUs: 15
			EXAM10_W_F1		EXAM1_W_F1		
			EXAM9_W_F1		EXAM2_W_F1		
			EXAM8_W_F1		EXAM3_W_F1		
			EXAM7_W_F1		EXAM4_W_F1		
			EXAM6_W_F1		CHECK_IN_W_F1		
			EXAM4_W_F1		EXAM5_W_F1		
			EXAM3_W_F1		WAITING_W_F1		
			EXAM2_W_F1		EXAM6_W_F1		
			EXAM1_W_F1		EXAM7_W_F1		
			LAB_M_F1		EXAM8_W_F1		
			EXAM2_M_F1		EXAM9_W_F1		
			EXAM1_M_F1		EXAM10_W_F1		
			OFF2_M_F1		EXAM11_W_F1		
			OFF1_M_F1		NURSE_OFF_W_F1		
			RESEARCH_M_F1		SOCIAL_WORKER_M_F1	Sum of FUs' area: 1430	Number of FUs: 12
			PERSONNEL_W_F1		PERSONNEL1_M_F1		
			WAITING_W_F1		PERSONNEL2_M_F1		
			CHECK_IN_W_F1		REGISTR_M_F1		
			REGISTR_M_F1		RESTRM_M_F1		
			PERSONNEL2_M_F1		RESEARCH_M_F1		
			PERSONNEL1_M_F1		OFF1_M_F1		
			SOCIAL_WORKER_M_F1		OFF2_M_F1		
			RESTRM_M_F1		CONF_M_F1		
			LOBBY_F1		EXAM1_M_F1		
			EXAM5_W_F1		EXAM2_M_F1		
			CONF_M_F1		LAB_M_F1		
			CHECK_IN_E_F1		LOBBY_F1		
Number of FUs			33	13			

1. Multiple criteria for stacking include adjacency, acoustic, and thermal. In zoning, daylight is considered in addition to adjacency, acoustic, and thermal, The relative weights are 10 for adjacency, 2 for acoustic, 1 for thermal, and 10 for daylight. The zoning threshold is set to 10, equivalent to an adjacency weight. With this threshold value, FUs with adjacency relations will be grouped together.

Stacking Floor 2			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Sum of FUs' area: 6854	Available area: -136	Number of FUs: 42	CONF_RM_W_F2		RECEPTION_W_F2	Sum of FUs' area: 1594	Number of FUs: 15
			EXAM8_W_F2		CHECK_IN_W_F2		
			EXAM7_W_F2		MEN_W_F2		
			EXAM6_W_F2		WOMEN_W_F2		
			EXAM5_W_F2		MED_REC_W_F2		
			EXAM4_W_F2		REGISTR_W_F2		
			EXAM3_W_F2		OFF1_W_F2		
			OFF1_W_F2		EXAM3_W_F2		
			FINANCIAL_W_F2		EXAM4_W_F2		
				PATIENT_ASSESS_E_F1	EXAM5_W_F2		
				STAFF_E_F1	EXAM6_W_F2		
				WORK_STATION_E_F1	EXAM7_W_F2		
				EXAM8_E_F1	EXAM8_W_F2		
				EXAM7_E_F1	CONF_RM_W_F2		
				EXAM6_E_F1	FINANCIAL_W_F2		
				EXAM5_E_F1	EXAM5_E_F1	Sum of FUs' area: 580	Number of FUs: 7
				STAFF_N_F6	EXAM6_E_F1		
				OFF_N_F6	EXAM8_E_F1		
				WORKAREA_S_F6	WORK_STATION_E_F1		
			CASHIER_E_F2		PATIENT_ASSESS_E_F1		
			PHARMACY_E_F2		EXAM7_E_F1		
			STAFF_E_F2		STAFF_E_F1		
			PHARMD2_E_F2		OFF_N_F6	192	2
			OFF2_E_F2		STAFF_N_F6		
			OFF1_E_F2		WORKAREA_S_F6		
			EXAM2_E_F2		HANDICAP_E_F2	Sum of FUs' area: 1934	Number of FUs: 16
			EXAM1_E_F2		RECEPT_E_F2		
			MED_REC_W_F2		STORAGE_E_F2		
			REGISTR_W_F2		PERSONNEL4_E_F2		
			MEN_W_F2		PERSONNEL2_E_F2		
			CHECK_IN_W_F2		PERSONNEL1_E_F2		
			WOMEN_W_F2		PERSONNEL3_E_F2		
			PERSONNEL1_E_F2		EXAM1_E_F2		
			HANDICAP_E_F2		EXAM2_E_F2		
			PERSONNEL4_E_F2		OFF1_E_F2		
			PERSONNEL3_E_F2		OFF2_E_F2		
			PERSONNEL2_E_F2		PHARMD1_E_F2		
			STORAGE_E_F2		CASHIER_E_F2		
			RECEPT_E_F2		STAFF_E_F2		
			INFO_F2		PHARMD2_E_F2		
			RECEPTION_W_F2		PHARMACY_E_F2		
			PHARMD1_E_F2		INFO_F2		
Number of FUs			32	10			

Stacking			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Floor 3							
Sum of FUs' area: 6354	Available area: -59.0005	Number of FUs: 44		EXAM1_E_F6	OFF_E_F6	264	2
				OFF_E_F6	EXAM1_E_F6		
				OFF9_W_F4	OFF1_W_F4	Sum of FUs' area: 1020	Number of FUs: 8
				OFF8_W_F4	OFF2_W_F4		
				OFF7_W_F4	OFF4_W_F4		
				OFF6_W_F4	OFF5_W_F4		
				OFF5_W_F4	OFF6_W_F4		
				OFF4_W_F4	OFF7_W_F4		
				OFF2_W_F4	OFF8_W_F4		
				OFF1_W_F4	OFF9_W_F4		
				LAB_E_F7	RESEARCH1_E_F7	Sum of FUs' area: 844	Number of FUs: 6
				RESEARCH2_E_F7	OFF_E_F7		
				EXAM_E_F7	TREATMENT_E_F7		
				TREATMENT_E_F7	EXAM_E_F7		
				OFF_E_F7	LAB_E_F7		
				RESEARCH1_E_F7	RESEARCH2_E_F7		
			STAFF_E_F3		EXAM1_E_F3	Sum of FUs' area: 664	Number of FUs: 8
			DRESS_E_F3		EXAM2_E_F3		
			FILE_SORTING_E_F3		CT_CONTROL_E_F3		
			CT_SCAN_E_F3		CT_SCAN_E_F3		
			CT_CONTROL_E_F3		PHLEBOTOMY_E_F3		
			EXAM2_E_F3		FILE_SORTING_E_F3		
			EXAM1_E_F3		DRESS_E_F3		
			STAFF_W_F3		STAFF_E_F3		
			EXAM3_W_F3		RESTROOM_W_F3	Sum of FUs' area: 1752	Number of FUs: 14
			EXAM1_W_F3		WAITING_W_F3		
			EXAM_F_W_F3		STAFF_W_F3		
			EXAM_E_W_F3		PERSONNEL1_W_F3		
			EXAM_D_W_F3		PERSONNEL2_W_F3		
			EXAM_C_W_F3		EXAM_A_W_F3		
			EXAM_B_W_F3		EXAM_B_W_F3		
			EXAM_A_W_F3		EXAM_C_W_F3		
			VASC_LAB_N_F3		EXAM_D_W_F3		
			PERSONNEL2_W_F3		EXAM_E_W_F3		
			PERSONNEL1_W_F3		EXAM_F_W_F3		
			RESTROOM_W_F3		EXAM1_W_F3		
			WAITING_W_F3		EXAM2_W_F3		
			PERSONNEL1_N_F3		EXAM3_W_F3		
			REGISTRATION_N_F3		AMBULATORY_N_F3	Sum of FUs' area: 1630	Number of FUs: 5
			PERSONNEL2_N_F3		VASC_LAB_N_F3		
			EXAM2_W_F3		REGISTRATION_N_F3		
			AMBULATORY_N_F3		PERSONNEL1_N_F3		
			PHLEBOTOMY_E_F3		PERSONNEL2_N_F3		
			MED_RECORDS_S_F3		MED_RECORDS_S_F3		
Number of FUs			28	16			

Stacking Floor 4	Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Sum of FUs' area: 5339; Available area: 443; Number of FUs: 51	OFF15_S_F4		RESTRM_N_F4	Sum of FUs' area: 3681	Number of FUs: 36
	OFF14_S_F4		OFF1_N_F4		
	OFF13_S_F4		OFF2_N_F4		
	OFF12_S_F4		STAFF_N_F4		
	OFF11_S_F4		WAITING1_E_F4		
	OFF10_S_F4		TREATMENT_E_F4		
	OFF9_S_F4		CONF1_E_F4		
	OFF8_S_F4		CONF2_E_F4		
	OFF7_S_F4		CONF3_E_F4		
	OFF6_S_F4		CONSULT1_E_F4		
	OFF5_S_F4		CONSULT2_E_F4		
	OFF4_S_F4		EXAM1_E_F4		
	OFF3_S_F4		EXAM2_E_F4		
	OFF2_S_F4		EXAM3_E_F4		
	OFF1_S_F4		EXAM4_E_F4		
	EXAM9_E_F4		EXAM5_E_F4		
	EXAM8_E_F4		EXAM6_E_F4		
	EXAM7_E_F4		EXAM7_E_F4		
	EXAM6_E_F4		EXAM8_E_F4		
	EXAM5_E_F4		EXAM9_E_F4		
	EXAM4_E_F4		WORK_AREA_S_F4		
	EXAM3_E_F4		OFF1_S_F4		
	EXAM2_E_F4		OFF2_S_F4		
	EXAM1_E_F4		OFF3_S_F4		
	CONSULT2_E_F4		OFF4_S_F4		
	CONSULT1_E_F4		OFF5_S_F4		
	CONF3_E_F4		OFF6_S_F4		
	CONF2_E_F4		OFF7_S_F4		
	CONF1_E_F4		OFF8_S_F4		
	TREATMENT_E_F4		OFF9_S_F4		
		STAFF2_W_F6	OFF10_S_F4		
		STAFF1_W_F6	OFF11_S_F4		
		EXAM9_W_F6	OFF12_S_F4		
		EXAM8_W_F6	OFF13_S_F4		
		EXAM7_W_F6	OFF14_S_F4		
		EXAM6_W_F6	OFF15_S_F4		
		EXAM5_W_F6	OFF1_W_F6	Sum of FUs' area: 1226	Number of FUs: 13
		EXAM4_W_F6	EXAM1_W_F6		
		EXAM3_W_F6	EXAM2_W_F6		
		OFF2_W_F6	OFF2_W_F6		
		EXAM2_W_F6	EXAM3_W_F6		
		EXAM1_W_F6	EXAM4_W_F6		
		OFF1_W_F6	EXAM5_W_F6		
	OFF2_N_F4		EXAM6_W_F6		
	OFF1_N_F4		STAFF1_W_F6		
	FILE_W_F4		STAFF2_W_F6		
	RESTRM_N_F4		EXAM8_W_F6		
	OFF3_W_F4		EXAM7_W_F6		
	STAFF_N_F4		EXAM9_W_F6		
	WAITING1_E_F4		OFF3_W_F4	432	2
	WORK_AREA_S_F4		FILE_W_F4		
Number of FUs	38	13			

Stacking			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Floor 5							
Sum of FUs' area: 6607	Available area: -45	Number of FUs: 63	EXAM9_E_F5		NURSE_W_F7	Sum of FUs' area: 1364	Number of FUs: 10
			EXAM8_E_F5		EXAM1_W_F7		
			EXAM7_E_F5		CONF1_W_F7		
			EXAM6_E_F5		OFF2_W_F7		
			EXAM5_E_F5		OFF1_W_F7		
			PHOTO_E_F5		EXAM3_W_F7		
			EXAM4_E_F5		CONF2_W_F7		
			EXAM3_E_F5		EXAM4_W_F7		
			EXAM2_E_F5		EXAM2_W_F7		
			EXAM1_E_F5		EXAM5_W_F7		
			STAFF_E_F5		OFF1_W_F5	Sum of FUs' area:1500	Number of FUs: 13
			LAB1_E_F5		OFF2_W_F5		
			LASER_E_F5		OFF3_W_F5		
				OFF2_W_F7	OFF4_W_F5		
				EXAM5_W_F7	OFF5_W_F5		
				EXAM4_W_F7	MD_OFF_W_F5		
				CONF2_W_F7	PRIVATE1_W_F5		
				EXAM3_W_F7	PRIVATE2_W_F5		
				OFF1_W_F7	PRIVATE3_W_F5		
				EXAM2_W_F7	OFF6_W_F5		
				CONF1_W_F7	OFF7_W_F5		
				EXAM1_W_F7	OFF8_W_F5		
				NURSE_W_F7	WORKSTATION_W_F5		
			OFF8_W_F5		EXAM1_S_F3	Sum of FUs' area:821	Number of FUs: 10
			OFF7_W_F5		EXAM2_S_F3		
			OFF6_W_F5		RESEARCH_S_F3		
			PRIVATE3_W_F5		STAFF1_S_F3		
			PRIVATE2_W_F5		STAFF2_S_F3		
			PRIVATE1_W_F5		READING_S_F3		
			OFF5_W_F5		UTILITY_S_F3		
			OFF4_W_F5		LOCKER1_S_F3		
			OFF3_W_F5		DARK_S_F3		
			OFF2_W_F5		LOCKER2_S_F3		
			OFF1_W_F5		OFF2_W_F2	Sum of FUs' area:690	Number of FUs: 7
			WORKSTATION_W_F5		OFF3_W_F2		
				DARK_S_F3	EXAM1_W_F2		
				READING_S_F3	EXAM2_W_F2		
				STAFF1_S_F3	LAB_W_F2		
				LOCKER2_S_F3	CONSULT1_W_F2		
				LOCKER1_S_F3	CONSULT2_W_F2		
				UTILITY_S_F3	OFF3_S_F5	124	2
				STAFF2_S_F3	RESTRM_S_F5		
				RESEARCH_S_F3	MEN_N_F5	Sum of FUs' area:2108	Number of FUs: 21
				EXAM2_S_F3	WOMEN_N_F5		
				EXAM1_S_F3	PERSONNEL1_N_F5		
				LAB_W_F2	PERSONNEL2_N_F5		
				CONSULT2_W_F2	CONF_N_F5		
				CONSULT1_W_F2	OFF1_N_F5		
				EXAM2_W_F2	OFF2_N_F5		
				EXAM1_W_F2	LASER_E_F5		
				OFF3_W_F2	LAB1_E_F5		
				OFF2_W_F2	EXAM1_E_F5		
			RESTRM_S_F5		EXAM2_E_F5		
			OFF2_N_F5		EXAM3_E_F5		
			OFF1_N_F5		EXAM4_E_F5		
			PERSONNEL2_N_F5		PHOTO_E_F5		
			PERSONNEL1_N_F5		EXAM5_E_F5		
			MEN_N_F5		STAFF_E_F5		
			WOMEN_N_F5		EXAM6_E_F5		
			MD_OFF_W_F5		EXAM7_E_F5		
			CONF_N_F5		EXAM8_E_F5		
			LAB2_E_F5		EXAM9_E_F5		
			OFF3_S_F5		LAB2_E_F5		
Number of FUs			36	27			

Stacking			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Floor 6							
Sum of FUs' area: 6792	Available area: -256	Number of FUs: 61	WAITING_S_F6		CHECK_IN_S_F4	289	4
				KITCHEN_S_F4	UTILITY_S_F4		
				SUPPLY_S_F4	SUPPLY_S_F4		
				UTILITY_S_F4	KITCHEN_S_F4		
				CHECK_IN_S_F4	STORAGE1_E_F3	Sum of FUs' area: 798	Number of FUs: 10
				STORAGE3_E_F3	STORAGE2_E_F3		
				STORAGE1_E_F3	FREEZER_E_F3		
				FREEZER_E_F3	STORAGE3_E_F3		
				STORAGE2_E_F3	RESTRM1_E_F3		
				RESTRM1_E_F3	RESTRM2_E_F3		
				RESTRM4_E_F3	MEN_E_F3		
				WOMEN_E_F3	WOMEN_E_F3		
				MEN_E_F3	RESTRM3_E_F3		
				RESTRM2_E_F3	RESTRM4_E_F3		
				RESTRM3_E_F3	WAITING2_E_F4		
				WAITING2_E_F4	MED_RECORDS_E_F5	380	3
				PERSONNEL_E_F5	PERSONNEL_E_F5		
				APPOINTMENT_E_F5	APPOINTMENT_E_F5		
				MED_RECORDS_E_F5	PERSONNEL_W_F5		
				PERSONNEL_W_F5	WORKAREA_W_F7	248	3
				PERSONNEL_W_F7	KITCHEN_W_F7		
				KITCHEN_W_F7	PERSONNEL_W_F7		
				WORKAREA_W_F7	RECEPT_N_F7	Sum of FUs' area: 1008	Number of FUs: 8
				PERSONNEL4_N_F7	RESTRM_N_F7		
				PERSONNEL_N_F7	WAITING_N_F7		
				PERSONNEL1_N_F7	PERSONNEL1_N_F7		
				TELEPHONE_N_F7	TELEPHONE_N_F7		
				PERSONNEL3_N_F7	PERSONNEL_N_F7		
				WAITING_N_F7	PERSONNEL3_N_F7		
				RECEPT_N_F7	PERSONNEL4_N_F7		
				RESTRM_N_F7	WAITING_E_F1	Sum of FUs' area: 606	Number of FUs: 5
				PERSONNEL1_E_F1	RESTRM_E_F1		
				PERSONNEL2_E_F1	WORK_AREA_E_F1		
				WORK_AREA_E_F1	PERSONNEL1_E_F1		
				WAITING_E_F1	PERSONNEL2_E_F1		
				RESTRM_E_F1	STORAGE1_S_F3	Sum of FUs' area: 450	Number of FUs: 5
				SUPPLY_S_F3	STORAGE2_S_F3		
				STORAGE2_S_F3	SUPPLY_S_F3		
				STORAGE1_S_F3	RESTRM1_S_F3		
				RESTRM1_S_F3	RESTRM2_S_F3		
				RESTRM2_S_F3	WAITING_W_F6	272	3
			RESTRM2_W_F6		RESTRM1_W_F6		
			RESTRM1_W_F6		RESTRM2_W_F6		
			PERSONNEL_E_F6		MED_RECORDS_E_F6	Sum of FUs' area: 736	Number of FUs: 6
			KITCHEN_E_F6		PERSONNEL_E_F6		
			MED_RECORDS_E_F6		KITCHEN_E_F6		
			RESTRM2_E_F6		RECEPTION_E_F6		
			RESTRM1_E_F6		RESTRM1_E_F6		
			PERSONNEL1_N_F6		RESTRM2_E_F6		
				DRUG_E_F7	RESTRM_E_F7	405	4
				PERSONNEL_E_F7	PERSONNEL_E_F7		
				WORKAREA_E_F7	DRUG_E_F7		
				RESTRM_E_F7	WORKAREA_E_F7		
			PERSONNEL2_N_F6		CONF_N_F6	Sum of FUs' area: 724	Number of FUs: 6
			WOMEN_N_F6		PERSONNEL1_N_F6		
			WAITING_N_F6		PERSONNEL2_N_F6		
			MEN_N_F6		WAITING_N_F6		
			WAITING_W_F6		MEN_N_F6		
			CONF_N_F6		WOMEN_N_F6		
			RECEPTION_E_F6		WAITING_S_F6	696	2
			MED_RECORDS_S_F6		MED_RECORDS_S_F6		
Number of FUs			17	44			

Stacking			Assignments matching existing placements	Assignments mismatching existing placements	Zoning		
Floor 7							
Sum of FUs' area: 4690	Available area: 63	Number of FUs: 44		OFF16_S_F4	OFF16_S_F4		
				OFF4_S_F6	OFF1_S_F6	Sum of FUs' area: 1826	Number of FUs: 19
				EXAM13_S_F6	EXAM1_S_F6		
				EXAM1_S_F6	EXAM2_S_F6		
				EXAM12_S_F6	EXAM3_S_F6		
				EXAM2_S_F6	EXAM4_S_F6		
				RECOVERY_S_F6	EXAM5_S_F6		
				EXAM3_S_F6	OFF2_S_F6		
				EXAM4_S_F6	EXAM6_S_F6		
				EXAM11_S_F6	PROCEDURE_S_F6		
				OFF3_S_F6	EXAM7_S_F6		
				EXAM10_S_F6	EXAM8_S_F6		
				EXAM9_S_F6	EXAM9_S_F6		
				EXAM8_S_F6	EXAM10_S_F6		
				EXAM7_S_F6	OFF3_S_F6		
				PROCEDURE_S_F6	EXAM11_S_F6		
				EXAM6_S_F6	RECOVERY_S_F6		
				OFF2_S_F6	EXAM12_S_F6		
				EXAM5_S_F6	EXAM13_S_F6		
				OFF1_S_F6	OFF4_S_F6		
				EXAM2_S_F5	EXAM1_S_F5	Sum of FUs' area: 2390	Number of FUs: 21
				PRIVATE2_S_F5	EXAM2_S_F5		
				EXAM3_S_F5	EXAM3_S_F5		
				OFF11_S_F5	EXAM4_S_F5		
				EXAM4_S_F5	EXAM5_S_F5		
				OFF10_S_F5	OFF1_S_F5		
				EXAM5_S_F5	EXAM6_S_F5		
				OFF9_S_F5	RESIDENCE_S_F5		
				OFF1_S_F5	OFF2_S_F5		
				OFF8_S_F5	OFF4_S_F5		
				OFF2_S_F5	CONF1_S_F5		
				RESIDENCE_S_F5	PRIVATE1_S_F5		
				OFF7_S_F5	OFF5_S_F5		
				OFF6_S_F5	OFF6_S_F5		
				OFF5_S_F5	OFF7_S_F5		
				PRIVATE1_S_F5	OFF8_S_F5		
				EXAM6_S_F5	OFF9_S_F5		
				OFF4_S_F5	OFF10_S_F5		
				EXAM1_S_F5	OFF11_S_F5		
				CONF2_S_F5	PRIVATE2_S_F5		
				CONF1_S_F5	CONF2_S_F5		
			FILE_W_F7		FILE_W_F7		
			SUPPLY_N_F7		SUPPLY_N_F7		
			BLOOD_E_F7		BLOOD_E_F7		
Number of FUs			3	41			

C.4 Center for the Arts

C.4.1 Alternative One (four floors)

Table C.5 Center for Arts: stacking according to adjacency and zoning according to multiple criteria (State id: 26, Total cut size: 354)[1]

Stacking				Zoning		
Basement flr			FUs			
1090	0	2	ORCHESTRA_PIT_THRUST	ORCHESTRA_PIT_THRUST	1090	2
			TRAP_ROOM	TRAP_ROOM		

Stacking				Zoning		
Floor 1			FUs			
Sum of FUs' area: 30720	Available area: -1101.66	Number of FUs: 33	STORAGE_REPAIR_REC	JANITOR_CLOSET		
			GALLERY	ELEC_LIGHT_STORAGE	Sum of FUs' area: 25715	Number of FUs: 24
			OFFICE_CONTROL	BOX_OFFICE		
			COAT_ROOM	COAT_ROOM		
			BLDG_SECURITY	BLDG_SECURITY		
			TECH_STUDENT_LOUNGE	FOYER		
			STUDENT_TECH_OFF_C	W_RESTROOM		
			STUDENT_TECH_OFF_B	M_RESTROOM		
			STUDENT_TECH_OFF_A	GALLERY		
			JANITOR_CLOSET	STORAGE_REPAIR_REC		
			PROPERTIES_SHOP	LOBBY_F1		
			ELEC_LIGHT_STORAGE	OFFICE_CONTROL		
			FOYER	LOAD_RECEIVE		
			GREEN_ROOM	TRASH		
			STAGE_STORAGE	SHOP_SUPERVISOR_OFF		
			PROPERTY_ROOM	STUDENT_TECH_OFF_A		
			SHOP_SUPERVISOR_OFF	STUDENT_TECH_OFF_C		
			BACKSTAGE_TOILET	STUDENT_TECH_OFF_B		
			W_DRESSING_ROOM	TECH_STUDENT_LOUNGE		
			M_DRESSING_ROOM	AUDIENCE_CHAMBER_F1		
			WARDROBE	STAGE_STORAGE		
			TV_REHEARSAL_CLASSRM	STAGE_AND_WINGS_F1		
			TV_CONTROL_BOOTH	PROPERTY_ROOM		
			TV_EDITING_BAY	SCENE_SHOP_F1		
			LOBBY_F1	PROPERTIES_SHOP		
			BOX_OFFICE	GREEN_ROOM		
			W_RESTROOM	BACKSTAGE_TOILET		
			M_RESTROOM	W_DRESSING_ROOM	1275	2
			TRASH	M_DRESSING_ROOM		
			AUDIENCE_CHAMBER_F1	WARDROBE		
			STAGE_AND_WINGS_F1	TV_REHEARSAL_CLASSRM	2990	3
			SCENE_SHOP_F1	TV_CONTROL_BOOTH		
			LOAD_RECEIVE	TV_EDITING_BAY		

1. Multiple criteria for zoning include adjacency, acoustic, thermal, and daylight. The relative weights are 10 for adjacency, 2 for acoustic, 1 for thermal, and 10 for daylight.

Stacking				Zoning		
Floor 2			FUs			
Sum of FUs' area: 21625	Available area: 3993.33	Number of FUs: 18	DESIGN_FACULTY_OFF	DESIGN_FACULTY_OFF		
			DESIGN_STUDIO	DESIGN_STUDIO		
			CONTROL_BOOTH	CONTROL_BOOTH		
			COSTUME_GENERAL_STORAGE	COSTUME_GENERAL_STORAGE		
			GRADUATE_DESIGN_STUDIO	GRADUATE_DESIGN_STUDIO		
			COSTUME_DAILY_STORAGE	COSTUME_DAILY_STORAGE		
			ELECTRIC_ROOM	ELECTRIC_ROOM		
			MECHANIC_ROOM	MECHANIC_ROOM		
			SOUND_STUDIO	SOUND_STUDIO		
			FOLLOWSPOT_BOOTH	FOLLOWSPOT_BOOTH		
			FACULTY_VOICE_STUDIO_C	FACULTY_VOICE_STUDIO_C		
			FACULTY_VOICE_STUDIO_B	FACULTY_VOICE_STUDIO_B		
			FACULTY_VOICE_STUDIO_A	FACULTY_VOICE_STUDIO_A		
			DIMMER_ROOM_F2	DIMMER_ROOM_F2		
			LOBBY_F2	LOBBY_F2	15425	4
			AUDIENCE_CHAMBER_F2	AUDIENCE_CHAMBER_F2		
			STAGE_AND_WINGS_F2	STAGE_AND_WINGS_F2		
			SCENE_SHOP_F2	SCENE_SHOP_F2		

Stacking				Zoning		
Floor 3			FUs			
Sum of FUs' area: 24510	Available area: -2891.67	Number of FUs: 33	ARMORY	ACTING_CLASSRM_A		
			ACTING_CLASSRM_A	DANCE_STUDIO	2050	2
			DANCE_FACULTY_OFF	DANCE_FACULTY_OFF		
			MOVEMENT_FACULTY_OFF	FITTING_ROOM	2425	4
			LAUNDRY_DYE_ROOM	COSTUME_SHOP		
			FITTING_ROOM	CRAFT_ROOM		
			CRAFT_ROOM	LAUNDRY_DYE_ROOM		
			ADMIN_ASSISTANTS	DEPT_HEAD_OFF	965	4
			SECRETARY_OFF	SECRETARY_OFF		
			COSTUME_SHOP	DEPT_ASSOC_HEAD_OFF		
			DEPT_ASSOC_HEAD_OFF	ADMIN_ASSISTANTS		
			DEPT_HEAD_OFF	ACTING_CLASSRM_C		
			ACTING_CLASSRM_C	ACTING_CLASSRM_B		
			ACTING_CLASSRM_B	MOVEMENT_FACULTY_OFF	2040	3
			DANCE_STUDIO	ARMORY		
			MOVEMENT_STUDIO	MOVEMENT_STUDIO		
			FACULTY_OFF_L	FACULTY_OFF_A	Sum of FUs' area: 1500	Number of FUs: 12
			FACULTY_OFF_K	FACULTY_OFF_B		
			FACULTY_OFF_J	FACULTY_OFF_C		
			FACULTY_OFF_I	FACULTY_OFF_D		
			FACULTY_OFF_H	FACULTY_OFF_E		
			FACULTY_OFF_G	FACULTY_OFF_F		
			FACULTY_OFF_F	FACULTY_OFF_G		
			FACULTY_OFF_E	FACULTY_OFF_H		
			FACULTY_OFF_D	FACULTY_OFF_I		
			FACULTY_OFF_C	FACULTY_OFF_J		
			FACULTY_OFF_B	FACULTY_OFF_K		
			FACULTY_OFF_A	FACULTY_OFF_L		
			M_REST_CHANGE_RM	M_REST_CHANGE_RM		
			W_REST_CHANGE_RM	W_REST_CHANGE_RM		
			AUDIENCE_CHAMBER_F3	AUDIENCE_CHAMBER_F3	12425	3
			STAGE_AND_WINGS_F3	STAGE_AND_WINGS_F3		
			SCENE_SHOP_F3	SCENE_SHOP_F3		

Table C.6 Center for Arts: stacking and zoning according to multiple criteria
(State id: 19, Total cut size: 1411)[1]

Stacking				Zoning		
Basement flr			FUs			
1090	0	2	ORCHESTRA_PIT_THRUST	ORCHESTRA_PIT_THRUST	1090	2
			TRAP_ROOM	TRAP_ROOM		

Stacking				Zoning		
Floor 1			FUs			
Sum of FUs' area: 29780	Available area: -161.664	Number of FUs: 27	OFFICE_CONTROL	JANITOR_CLOSET		
			GALLERY	GREEN_ROOM		
			STORAGE_REPAIR_REC	BACKSTAGE_TOILET		
			PROPERTIES_SHOP	W_DRESSING_ROOM	1275	2
			JANITOR_CLOSET	M_DRESSING_ROOM		
			ELEC_LIGHT_STORAGE	WARDROBE		
			FOYER	TV_REHEARSAL_CLASSRM	2990	3
			GREEN_ROOM	TV_CONTROL_BOOTH		
			STAGE_STORAGE	TV_EDITING_BAY		
			PROPERTY_ROOM	SHOP_SUPERVISOR_OFF	Sum of FUs' area: 24775	Number of FUs: 18
			SHOP_SUPERVISOR_OFF	ELEC_LIGHT_STORAGE		
			BACKSTAGE_TOILET	PROPERTIES_SHOP		
			W_DRESSING_ROOM	STAGE_STORAGE		
			M_DRESSING_ROOM	SCENE_SHOP_F1		
			WARDROBE	STAGE_AND_WINGS_F1		
			TV_REHEARSAL_CLASSRM	PROPERTY_ROOM		
			TV_CONTROL_BOOTH	BOX_OFFICE		
			TV_EDITING_BAY	AUDIENCE_CHAMBER_F1		
			LOBBY_F1	FOYER		
			BOX_OFFICE	W_RESTROOM		
			W_RESTROOM	M_RESTROOM		
			M_RESTROOM	GALLERY		
			TRASH	STORAGE_REPAIR_REC		
			AUDIENCE_CHAMBER_F1	LOBBY_F1		
			STAGE_AND_WINGS_F1	OFFICE_CONTROL		
			SCENE_SHOP_F1	LOAD_RECEIVE		
			LOAD_RECEIVE	TRASH		

1. Multiple criteria for stacking include adjacency, acoustic, and thermal. In zoning, daylight is considered in addition to adjacency, acoustic, and thermal. The relative weights are 10 for adjacency, 2 for acoustic, 1 for thermal, and 10 for daylight.

Stacking				Zoning		
Floor 2			FUs			
Sum of FUs' area: 22025	Available area: 3593.33	Number of FUs: 17	ACTING_CLASSRM_A	ACTING_CLASSRM_A		
			ARMORY	ARMORY		
			DESIGN_STUDIO	DESIGN_STUDIO		
			GRADUATE_DESIGN_STUDIO	GRADUATE_DESIGN_STUDIO		
			COSTUME_GENERAL_STORAGE	COSTUME_GENERAL_STORAGE		
			COSTUME_DAILY_STORAGE	COSTUME_DAILY_STORAGE		
			FOLLOWSPOT_BOOTH	FOLLOWSPOT_BOOTH		
			CONTROL_BOOTH	CONTROL_BOOTH		
			MECHANIC_ROOM	MECHANIC_ROOM		
			ELECTRIC_ROOM	ELECTRIC_ROOM		
			BLDG_SECURITY	BLDG_SECURITY		
			COAT_ROOM	COAT_ROOM		
			DIMMER_ROOM_F2	DIMMER_ROOM_F2		
			LOBBY_F2	LOBBY_F2	15425	4
			AUDIENCE_CHAMBER_F2	AUDIENCE_CHAMBER_F2		
			STAGE_AND_WINGS_F2	STAGE_AND_WINGS_F2		
			SCENE_SHOP_F2	SCENE_SHOP_F2		

Stacking				Zoning		
Floor 3			FUs			
Sum of FUs' area: 25050	Available area: -3431.67	Number of FUs: 40	FACULTY_VOICE_STUDIO_B	FACULTY_VOICE_STUDIO_B		
			FACULTY_VOICE_STUDIO_A	FACULTY_VOICE_STUDIO_A		
			SOUND_STUDIO	SOUND_STUDIO		
			FACULTY_VOICE_STUDIO_C	FACULTY_VOICE_STUDIO_C		
			DESIGN_FACULTY_OFF	DESIGN_FACULTY_OFF		
			MOVEMENT_FACULTY_OFF	MOVEMENT_STUDIO	1990	2
			FITTING_ROOM	MOVEMENT_FACULTY_OFF		
			DANCE_FACULTY_OFF	DANCE_STUDIO	2050	2
			LAUNDRY_DYE_ROOM	DANCE_FACULTY_OFF		
			ADMIN_ASSISTANTS	FITTING_ROOM	2425	4
			SECRETARY_OFF	COSTUME_SHOP		
			CRAFT_ROOM	CRAFT_ROOM		
			DEPT_ASSOC_HEAD_OFF	LAUNDRY_DYE_ROOM		
			DEPT_HEAD_OFF	DEPT_HEAD_OFF	965	4
			COSTUME_SHOP	SECRETARY_OFF		
			STUDENT_TECH_OFF_A	DEPT_ASSOC_HEAD_OFF		
			TECH_STUDENT_LOUNGE	ADMIN_ASSISTANTS		
			STUDENT_TECH_OFF_C	STUDENT_TECH_OFF_A	600	4
			STUDENT_TECH_OFF_B	STUDENT_TECH_OFF_B		
			ACTING_CLASSRM_C	STUDENT_TECH_OFF_C		
			ACTING_CLASSRM_B	TECH_STUDENT_LOUNGE		
			DANCE_STUDIO	ACTING_CLASSRM_C		
			MOVEMENT_STUDIO	ACTING_CLASSRM_B		
			FACULTY_OFF_L	FACULTY_OFF_A	Sum of FUs' area: 1500	Number of FUs: 12
			FACULTY_OFF_K	FACULTY_OFF_B		
			FACULTY_OFF_J	FACULTY_OFF_C		
			FACULTY_OFF_I	FACULTY_OFF_D		
			FACULTY_OFF_H	FACULTY_OFF_E		
			FACULTY_OFF_G	FACULTY_OFF_F		
			FACULTY_OFF_F	FACULTY_OFF_G		
			FACULTY_OFF_E	FACULTY_OFF_H		
			FACULTY_OFF_D	FACULTY_OFF_I		
			FACULTY_OFF_C	FACULTY_OFF_J		
			FACULTY_OFF_B	FACULTY_OFF_K		
			FACULTY_OFF_A	FACULTY_OFF_L		
			M_REST_CHANGE_RM	M_REST_CHANGE_RM		
			W_REST_CHANGE_RM	W_REST_CHANGE_RM		
			AUDIENCE_CHAMBER_F3	AUDIENCE_CHAMBER_F3	12425	3
			STAGE_AND_WINGS_F3	STAGE_AND_WINGS_F3		
			SCENE_SHOP_F3	SCENE_SHOP_F3		

C.4.2 Alternative Two (five floors)

Table C.7 Center for Arts: stacking according to adjacency and zoning according to multiple criteria (State id: 32, Total cut size: 464)[1]

Stacking				Zoning		
Basement flr			FUs			
1090	0	2	ORCHESTRA_PIT_THRUST	ORCHESTRA_PIT_THRUST	1090	2
			TRAP_ROOM	TRAP_ROOM		

Stacking				Zoning		
Floor 1			FUs			
Sum of FUs' area: 29420	Available area: -801.666	Number of FUs: 32	GALLERY	JANITOR_CLOSET		
			OFFICE_CONTROL	BOX_OFFICE	Sum of FUs' area: 24415	Number of FUs: 23
			COAT_ROOM	COAT_ROOM		
			BLDG_SECURITY	BLDG_SECURITY		
			TECH_STUDENT_LOUNGE	FOYER		
			STUDENT_TECH_OFF_C	W_RESTROOM		
			STUDENT_TECH_OFF_B	M_RESTROOM		
			STUDENT_TECH_OFF_A	GALLERY		
			JANITOR_CLOSET	LOBBY_F1		
			PROPERTIES_SHOP	OFFICE_CONTROL		
			ELEC_LIGHT_STORAGE	LOAD_RECEIVE		
			FOYER	TRASH		
			GREEN_ROOM	SHOP_SUPERVISOR_OFF		
			STAGE_STORAGE	STUDENT_TECH_OFF_A		
			PROPERTY_ROOM	STUDENT_TECH_OFF_B		
			SHOP_SUPERVISOR_OFF	STUDENT_TECH_OFF_C		
			BACKSTAGE_TOILET	TECH_STUDENT_LOUNGE		
			W_DRESSING_ROOM	AUDIENCE_CHAMBER_F1		
			M_DRESSING_ROOM	ELEC_LIGHT_STORAGE		
			WARDROBE	STAGE_STORAGE		
			TV_REHEARSAL_CLASSRM	STAGE_AND_WINGS_F1		
			TV_CONTROL_BOOTH	PROPERTY_ROOM		
			TV_EDITING_BAY	SCENE_SHOP_F1		
			LOBBY_F1	PROPERTIES_SHOP		
			BOX_OFFICE	GREEN_ROOM		
			W_RESTROOM	BACKSTAGE_TOILET		
			M_RESTROOM	W_DRESSING_ROOM	1275	2
			TRASH	M_DRESSING_ROOM		
			AUDIENCE_CHAMBER_F1	WARDROBE		
			STAGE_AND_WINGS_F1	TV_REHEARSAL_CLASSRM	2990	3
			SCENE_SHOP_F1	TV_CONTROL_BOOTH		
			LOAD_RECEIVE	TV_EDITING_BAY		

1. Multiple criteria for zoning include adjacency, acoustic, thermal and daylight. The relative weights are 10 for adjacency, 2 for acoustic, 1 for thermal, and 10 for daylight.

Stacking				Zoning		
Floor 2			FUs			
Sum of FUs' area: 19265	Available area: 1353.33	Number of FUs: 12	COSTUME_SHOP	COSTUME_SHOP	1940	2
			FITTING_ROOM	FITTING_ROOM		
			DANCE_FACULTY_OFF	DANCE_FACULTY_OFF		
			MOVEMENT_FACULTY_OFF	MOVEMENT_FACULTY_OFF		
			ACTING_CLASSRM_A	ACTING_CLASSRM_A		
			ARMORY	ARMORY		
			DESIGN_FACULTY_OFF	DESIGN_FACULTY_OFF		
			DIMMER_ROOM_F2	DIMMER_ROOM_F2		
			LOBBY_F2	LOBBY_F2	15425	4
			AUDIENCE_CHAMBER_F2	AUDIENCE_CHAMBER_F2		
			STAGE_AND_WINGS_F2	STAGE_AND_WINGS_F2		
			SCENE_SHOP_F2	SCENE_SHOP_F2		

Stacking				Zoning		
Floor 3			FUs			
Sum of FUs' area: 19515	Available area: -896.676	Number of FUs: 16	DESIGN_STUDIO	DESIGN_STUDIO		
			STORAGE_REPAIR_REC	MECHANIC_ROOM	1440	2
			CONTROL_BOOTH	STORAGE_REPAIR_REC		
			COSTUME_GENERAL_STORAGE	CONTROL_BOOTH		
			GRADUATE_DESIGN_STUDIO	COSTUME_GENERAL_STORAGE		
			COSTUME_DAILY_STORAGE	GRADUATE_DESIGN_STUDIO		
			ELECTRIC_ROOM	COSTUME_DAILY_STORAGE		
			MECHANIC_ROOM	ELECTRIC_ROOM		
			SOUND_STUDIO	SOUND_STUDIO		
			FOLLOWSPOT_BOOTH	FOLLOWSPOT_BOOTH		
			FACULTY_VOICE_STUDIO_C	FACULTY_VOICE_STUDIO_C		
			FACULTY_VOICE_STUDIO_B	FACULTY_VOICE_STUDIO_B		
			FACULTY_VOICE_STUDIO_A	FACULTY_VOICE_STUDIO_A		
			AUDIENCE_CHAMBER_F3	AUDIENCE_CHAMBER_F3	12425	3
			STAGE_AND_WINGS_F3	STAGE_AND_WINGS_F3		
			SCENE_SHOP_F3	SCENE_SHOP_F3		

Stacking				Zoning		
Floor 4			FUs			
Sum of FUs' area: 8655	Available area: 345.011	Number of FUs: 24	LAUNDRY_DYE_ROOM	CRAFT_ROOM	485	2
			CRAFT_ROOM	LAUNDRY_DYE_ROOM		
			ADMIN_ASSISTANTS	DEPT_HEAD_OFF	965	4
			SECRETARY_OFF	SECRETARY_OFF		
			DEPT_ASSOC_HEAD_OFF	DEPT_ASSOC_HEAD_OFF		
			DEPT_HEAD_OFF	ADMIN_ASSISTANTS		
			ACTING_CLASSRM_C	ACTING_CLASSRM_C		
			ACTING_CLASSRM_B	ACTING_CLASSRM_B		
			DANCE_STUDIO	DANCE_STUDIO		
			MOVEMENT_STUDIO	MOVEMENT_STUDIO		
			FACULTY_OFF_L	FACULTY_OFF_A	Sum of FUs' area: 1500	Number of FUs: 12
			FACULTY_OFF_K	FACULTY_OFF_B		
			FACULTY_OFF_J	FACULTY_OFF_C		
			FACULTY_OFF_I	FACULTY_OFF_D		
			FACULTY_OFF_H	FACULTY_OFF_E		
			FACULTY_OFF_G	FACULTY_OFF_F		
			FACULTY_OFF_F	FACULTY_OFF_G		
			FACULTY_OFF_E	FACULTY_OFF_H		
			FACULTY_OFF_D	FACULTY_OFF_I		
			FACULTY_OFF_C	FACULTY_OFF_J		
			FACULTY_OFF_B	FACULTY_OFF_K		
			FACULTY_OFF_A	FACULTY_OFF_L		
			M_REST_CHANGE_RM	M_REST_CHANGE_RM		
			W_REST_CHANGE_RM	W_REST_CHANGE_RM		

Table C.8 Center for Arts: stacking and zoning according to multiple criteria (State id: 23, Total cut size: 1553)[1]

Stacking				Zoning		
Basement flr			FUs			
1090	0	2	ORCHESTRA_PIT_THRUST	ORCHESTRA_PIT_THRUST	1090	2
			TRAP_ROOM	TRAP_ROOM		

1. Multiple criteria for stacking include adjacency, acoustic, and thermal. In zoning, daylight is considered in addition to adjacency, acoustic, and thermal. The relative weights are 10 for adjacency, 2 for acoustic, 1 for thermal, and 10 for daylight. The zoning threshold is set to 10, equivalent to an adjacency weight.

Stacking				Zoning		
Floor 1			FUs			
Sum of FUs' area: 29280	Available area: -661.666	Number of FUs: 26	GALLERY	JANITOR_CLOSET		
			STORAGE_REPAIR_REC	GREEN_ROOM		
			PROPERTIES_SHOP	BACKSTAGE_TOILET		
			JANITOR_CLOSET	W_DRESSING_ROOM	1275	2
			ELEC_LIGHT_STORAGE	M_DRESSING_ROOM		
			FOYER	WARDROBE		
			GREEN_ROOM	TV_REHEARSAL_CLASSRM	2990	3
			STAGE_STORAGE	TV_CONTROL_BOOTH		
			PROPERTY_ROOM	TV_EDITING_BAY		
			SHOP_SUPERVISOR_OFF	SHOP_SUPERVISOR_OFF	Sum of FUs' area: 24275	Number of FUs: 17
			BACKSTAGE_TOILET	ELEC_LIGHT_STORAGE		
			W_DRESSING_ROOM	AUDIENCE_CHAMBER_F1		
			M_DRESSING_ROOM	STAGE_STORAGE		
			WARDROBE	STAGE_AND_WINGS_F1		
			TV_REHEARSAL_CLASSRM	PROPERTY_ROOM		
			TV_CONTROL_BOOTH	SCENE_SHOP_F1		
			TV_EDITING_BAY	PROPERTIES_SHOP		
			LOBBY_F1	BOX_OFFICE		
			BOX_OFFICE	FOYER		
			W_RESTROOM	W_RESTROOM		
			M_RESTROOM	M_RESTROOM		
			TRASH	GALLERY		
			AUDIENCE_CHAMBER_F1	STORAGE_REPAIR_REC		
			STAGE_AND_WINGS_F1	LOBBY_F1		
			SCENE_SHOP_F1	LOAD_RECEIVE		
			LOAD_RECEIVE	TRASH		

Stacking				Zoning		
Floor 2			FUs			
Sum of FUs' area: 21475	Available area: -856.674	Number of FUs: 16	DESIGN_STUDIO	DESIGN_STUDIO		
			GRADUATE_DESIGN_STUDIO	GRADUATE_DESIGN_STUDIO		
			COSTUME_GENERAL_STORAGE	COSTUME_GENERAL_STORAGE		
			COSTUME_DAILY_STORAGE	COSTUME_DAILY_STORAGE		
			FOLLOWSPOT_BOOTH	FOLLOWSPOT_BOOTH		
			CONTROL_BOOTH	CONTROL_BOOTH		
			OFFICE_CONTROL	OFFICE_CONTROL		
			MECHANIC_ROOM	MECHANIC_ROOM		
			ELECTRIC_ROOM	ELECTRIC_ROOM		
			BLDG_SECURITY	BLDG_SECURITY		
			COAT_ROOM	COAT_ROOM		
			DIMMER_ROOM_F2	DIMMER_ROOM_F2		
			LOBBY_F2	LOBBY_F2	15425	4
			AUDIENCE_CHAMBER_F2	AUDIENCE_CHAMBER_F2		
			STAGE_AND_WINGS_F2	STAGE_AND_WINGS_F2		
			SCENE_SHOP_F2	SCENE_SHOP_F2		

Stacking				Zoning		
Floor 3			FUs			
Sum of FUs' area: 15960	Available area: 2658.32	Number of FUs: 9	COSTUME_SHOP	COSTUME_SHOP	2235	3
			CRAFT_ROOM	CRAFT_ROOM		
			LAUNDRY_DYE_ROOM	LAUNDRY_DYE_ROOM		
			DANCE_FACULTY_OFF	DANCE_FACULTY_OFF		
			ACTING_CLASSRM_A	ACTING_CLASSRM_A		
			ARMORY	ARMORY		
			AUDIENCE_CHAMBER_F3	AUDIENCE_CHAMBER_F3	12425	3
			STAGE_AND_WINGS_F3	STAGE_AND_WINGS_F3		
			SCENE_SHOP_F3	SCENE_SHOP_F3		

Stacking				Zoning		
Floor 4			FUs			
Sum of FUs' area: 10140	Available area: -1139.99	Number of FUs: 33	FACULTY_VOICE_STUDIO_B	FACULTY_VOICE_STUDIO_B		
			FACULTY_VOICE_STUDIO_A	FACULTY_VOICE_STUDIO_A		
			SOUND_STUDIO	SOUND_STUDIO		
			FACULTY_VOICE_STUDIO_C	FACULTY_VOICE_STUDIO_C		
			DESIGN_FACULTY_OFF	DESIGN_FACULTY_OFF		
			MOVEMENT_FACULTY_OFF	MOVEMENT_STUDIO	1990	2
			FITTING_ROOM	MOVEMENT_FACULTY_OFF		
			ADMIN_ASSISTANTS	FITTING_ROOM		
			SECRETARY_OFF	DEPT_HEAD_OFF	965	4
			DEPT_ASSOC_HEAD_OFF	SECRETARY_OFF		
			DEPT_HEAD_OFF	DEPT_ASSOC_HEAD_OFF		
			STUDENT_TECH_OFF_A	ADMIN_ASSISTANTS		
			TECH_STUDENT_LOUNGE	STUDENT_TECH_OFF_A	600	4
			STUDENT_TECH_OFF_C	STUDENT_TECH_OFF_B		
			STUDENT_TECH_OFF_B	STUDENT_TECH_OFF_C		
			ACTING_CLASSRM_C	TECH_STUDENT_LOUNGE		
			ACTING_CLASSRM_B	ACTING_CLASSRM_C		
			DANCE_STUDIO	ACTING_CLASSRM_B		
			MOVEMENT_STUDIO	DANCE_STUDIO		
			FACULTY_OFF_L	FACULTY_OFF_A	Sum of FUs' area: 1500	Number of FUs: 12
			FACULTY_OFF_K	FACULTY_OFF_B		
			FACULTY_OFF_J	FACULTY_OFF_C		
			FACULTY_OFF_I	FACULTY_OFF_D		
			FACULTY_OFF_H	FACULTY_OFF_E		
			FACULTY_OFF_G	FACULTY_OFF_F		
			FACULTY_OFF_F	FACULTY_OFF_G		
			FACULTY_OFF_E	FACULTY_OFF_H		
			FACULTY_OFF_D	FACULTY_OFF_I		
			FACULTY_OFF_C	FACULTY_OFF_J		
			FACULTY_OFF_B	FACULTY_OFF_K		
			FACULTY_OFF_A	FACULTY_OFF_L		
			M_REST_CHANGE_RM	M_REST_CHANGE_RM		
			W_REST_CHANGE_RM	W_REST_CHANGE_RM		

C.5 Test Results of SABA

Table C.9 SABA: stacking according to adjacency (Brooklyn Jail)[1]

	Number of FUs belonging to floor									# of FUs matching existing placement
	1	2	3	4	5	6	7	8	9	
Floor 9									**4**	4
Floor 8							1	**10**		10
Floor 7					2	2	**10**	2		10
Floor 6					3	**13**	4	3		13
Floor 5				5	**10**	1	1			10
Floor 4			5	**9**	1					9
Floor 3		5	**9**	1						9
Floor 2	3	**10**	1			2				10
Floor 1	**16**					1				16
Total	19	15	15	15	16	19	16	15	4	91

1. In testing this project, three FUs are pre-assigned to the first floor, and two FUs are pre-assigned to each of the remaining floors. Bold numbers represent the number of FUs whose assignments match existing placements.

Table C.10 SABA: stacking according to adjacency (Kaiser)[1]

	Number of FUs belonging to floor																											**# of FUs matching existing placement**
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
Floor 27																								1			**2**	2
Floor 26																										**4**		4
Floor 25																									**4**			4
Floor 24																								**4**			2	4
Floor 23																							**4**					4
Floor 22																						**4**						4
Floor 21																					**4**							4
Floor 20																				**4**								4
Floor 19																			**4**									4
Floor 18																		**4**										4
Floor 17																	**4**											4
Floor 16																**4**												4
Floor 15															**4**													4
Floor 14														**4**														4
Floor 13													**4**															4
Floor 12												**4**																4
Floor 11	5									1	**4**																	4
Floor 10	8								1	**3**	1																	3
Floor 9	1							1	**3**																			3
Floor 8							1	**3**																				3
Floor 7						1	**3**																					3
Floor 6					1	**3**																						3
Floor 5				1	**3**																							3
Floor 4			1	**3**																								3
Floor 3		3	**2**						1	1																		2
Floor 2	1	**6**								1																		6
Floor 1	**3**		1							1																		3
Total	18	9	4	4	4	4	4	4	5	7	5	4	4	4	4	4	4	4	4	4	4	4	4	5	4	4	4	98

1. In testing this project, three FUs are pre-assigned to the first and second floors respectively; a single FU is pre-assigned to each of the remaining floors (floors three through 27). Bold numbers represent the number of FUs whose assignments match existing placements.

Appendix D:
User Interface Interaction Diagrams

In an interaction diagram, system objects are expressed by symbols. Figure D.1 provides the definitions of these symbols.

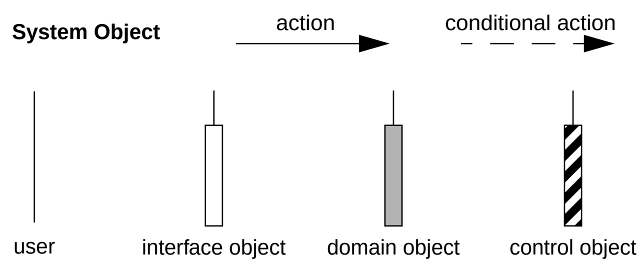

Figure D.1 Legend of interaction diagram

D.1 Start FD

Start the FD program.

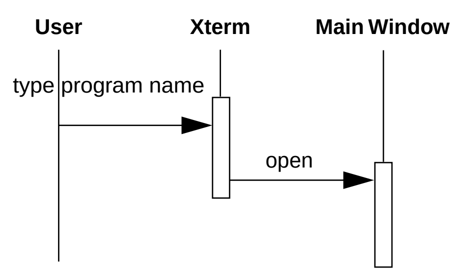

Figure D.2 Interaction diagram of starting FD

D.1.1 Flow of Events

- Associations

  This use case starts when the user types “FD” in the xterm. This will cause the FD program to launch and the Main Window to open, thereby ending the use case.

- Participating Objects

  Main Window

- Special Requirements
- Pre-conditions

- Post-conditions

  The Main Window is opened.

D.2 New project

Start a new FD project from scratch.

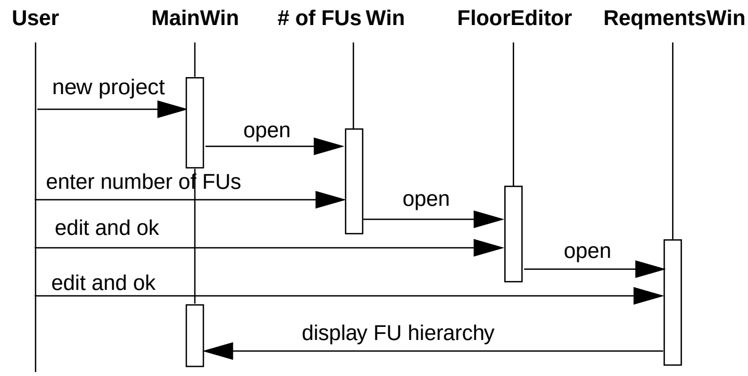

Figure D.3 Interaction diagram of starting new project

D.2.1 Flow of Events

- Associations

  The “new project” operation consists of three steps in which: *a)* the user defines the number of FUs; *b)* the user defines the number of floors and the area of each floor; and *c)* the user defines the necessary design requirements, including adjacency, thermal, acoustic, and daylight. In order to simplify the user’s operations, there are defaults for all of the requirements except for the number of FUs. The user can edit the default values in their corresponding windows.

  This use case starts when the user selects “new project” on the Main Window menu. After the user makes this selection, the system opens a dialogue window which prompts the user for a number of FUs in the building. After the user enters the number of FUs, the dialogue window closes. The Floor Editor Window will then be opened. It prompts the user to edit floor area requirements. When the user finishes editing and selects the “OK” button, the Floor Editor Window closes. Next, the Design Requirements Window will be opened and prompt the user to edit adjacency, thermal, acoustic, and daylight requirements. When the user selects the “OK” button, an FU hierarchy will be constructed and loaded in the Main Window.

- Participating Objects

  Main Window, Number of FUs Window, Floor Editor Window, Design Requirements Window, FU Hierarchy.

- Special Requirements
- Pre-conditions
- Post-conditions

  The FU hierarchy is loaded in the Main Window.

D.3 Load FU hierarchy

Load an existing FD project.

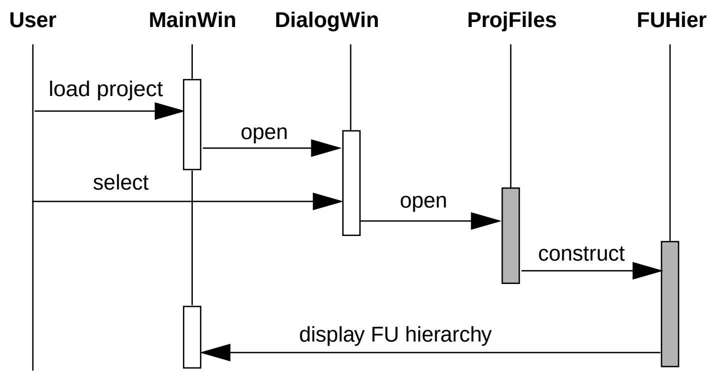

Figure D.4 Interaction diagram of loading a project

D.3.1 Flow of Events

- Associations

  This use case starts when the user selects “load project” on the Main Window menu. After the user makes this selection, the system opens a file dialogue window, which prompts the user to select a project. After the user selects a project, the file dialogue window closes. A flat FU hierarchy will be loaded in the Main Window.

- Participating Objects

  Main Window, File Dialogue Window, project files, FU Hierarchy.

- Special Requirements

  The project files are located within the project directory. They include a base file, an adjacency file, a thermal file, an acoustic file, a daylight file, and a pre-assignment file.

- Pre-conditions
- Post-conditions

  The FU hierarchy is loaded in the Main Window. All of the design requirements are loaded into the system memory (although they do not show in the Main Window).

D.4 Exit FD

Exit the FD program and close all of the opened windows.

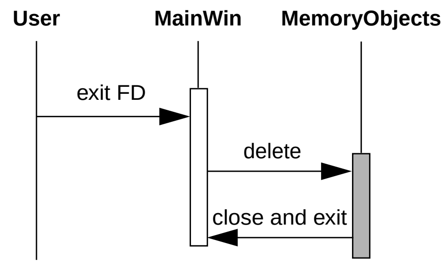

Figure D.5 Interaction diagram of exiting FD

D.4.1 Flow of Events

- Associations

  This use case starts when the user selects “exit FD” on the Main Window menu. After the user makes this selection, the system will *a)* free all of the memory that active objects occupy, *b)* close all of the opened windows, and *c)* exit the FD program. This use case ends when the FD program terminates.

- Participating Objects

  Main Window

- Special Requirements
- Pre-conditions
- Post-conditions

  The active objects are deleted. The Main Window is closed.

D.5 Select criteria

Select stacking or zoning criteria and define their relative weights.

D.5.1 Flow of Events

- Associations

  This use case starts when the user selects either “start stacking” or “start zoning” on the Main Window menu.

  After the user makes this selection, the system will open the Stacking / Zoning Criteria Window, which prompts the user to select a set of criteria. For stacking, the candidate criteria are adjacency, thermal and acoustic. For zoning, the candidate criterion is daylight in addition to adjacency, thermal, and acoustic. This window also prompts the user to define the relative importance between the selected set of criteria. After the user makes the selections, the dialogue window closes, and either the stacking or the zoning algorithm runs.

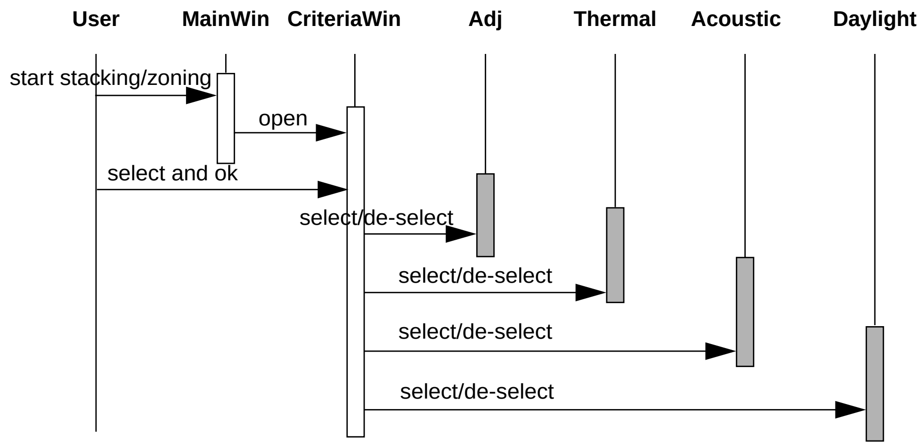

Figure D.6 Interaction diagram of selecting criteria

- Participating Objects

  Main Window, Stacking / zoning Criteria Window, adjacency, thermal, acoustic, and daylight criteria.

- Special Requirements
- Pre-conditions
- Post-conditions

  A set of design criteria are selected with their relative weights defined. The system generates unified requirements based on the selected criteria.

D.6 Edit adjacency matrix

Edit adjacency matrix.

D.6.1 Flow of Events

- Associations

  This use case starts when the user selects "edit requirements" on the Main Window menu. After the user makes this selection, the system opens the Design Requirements Window. When the user selects "adjacency requirements" in this window, the Adjacency Matrix Window will be opened. The user then can edit this matrix. If the user wants to find out the relation between two FUs, he / she has to enter the FUs' ids and the corresponding cell in the matrix window will be highlighted. When the user finishes editing the matrix, he / she can select the "save changes" button, and the adjacency matrix will be saved.

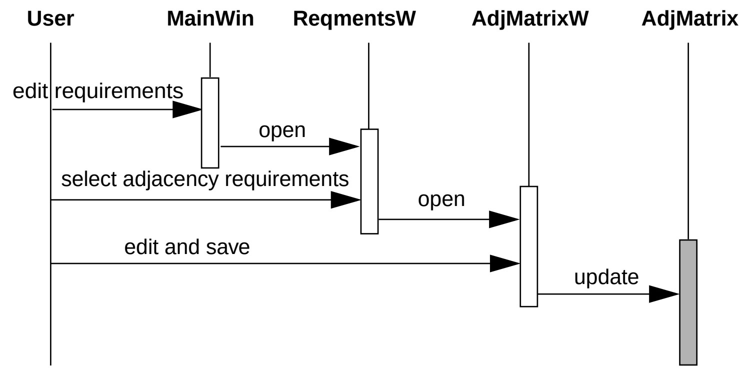

Figure D.7 Interaction diagram of editing adjacency matrix

- Participating Objects

  Main Window, Design Requirements Window, Adjacency Matrix Window, adjacency matrix.

- Special Requirements
- Pre-conditions
- Post-conditions

  The adjacency matrix is saved.

D.7 Edit acoustic table

Edit acoustic table.

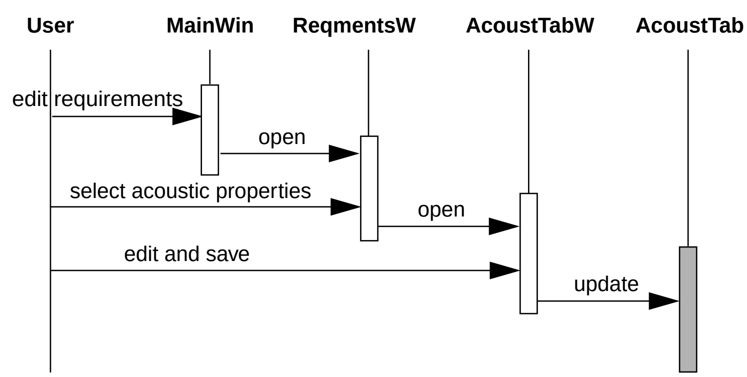

Figure D.8 Interaction diagram of editing acoustic table

D.7.1 Flow of Events

- Associations

  This use case starts when the user selects "edit requirements" on the Main Window menu. After the user makes this selection, the system opens the Design Requirements Window. After the user selects the "acoustic properties" bar in this window, the bar will be expanded into an acoustic table. The user can then edit this table. When the user finishes editing, he/she can select the "save changes" button, and the acoustic table will be saved.

- Participating Objects

  Main Window, Design Requirements Window, Acoustic Table, acoustic table.

- Special Requirements
- Pre-conditions
- Post-conditions

  The acoustic table is saved.

D.8 Edit temperature table

Edit temperature table.

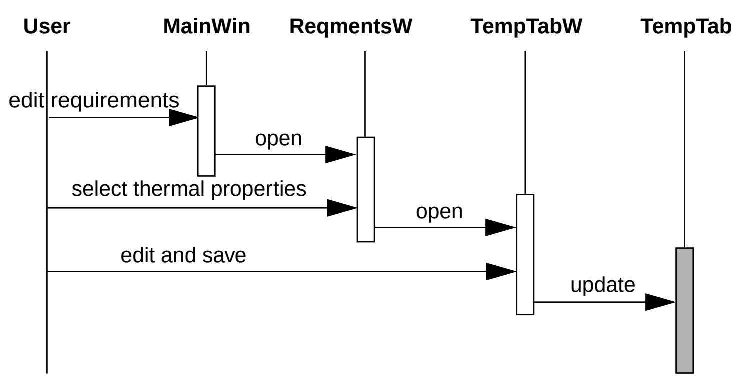

Figure D.9 Interaction diagram of editing temperature table

D.8.1 Flow of Events

- Associations

  This use case starts when the user selects "edit requirements" on the Main Window menu. After the user makes this selection, the system opens the Design Requirements Window. When the user selects the "thermal properties" bar in this window, the bar will be expanded into a temperature table. The user can then edit this table. When the user finishes editing, he / she can select the “save changes” button, and the temperature table will be saved.

- Participating Objects

  Main Window, Design Requirements Window, Temperature Table, temperature table.

- Special Requirements
- Pre-conditions
- Post-conditions

  The temperature table is saved.

D.9 Edit daylight requirements

Edit daylight requirements.

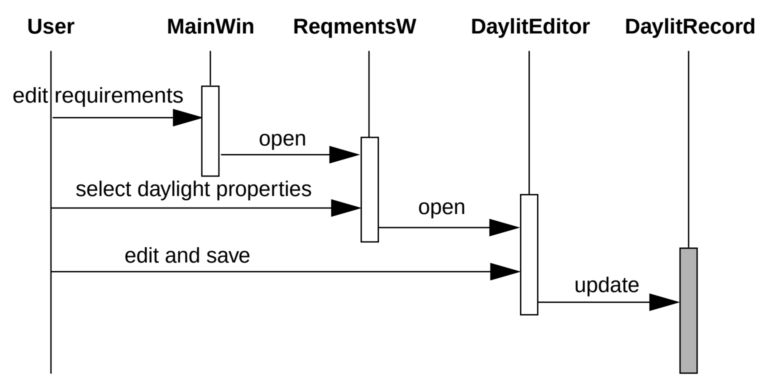

Figure D.10 Interaction diagram of editing daylight requirements

D.9.1 Flow of Events

- Associations

  This use case starts when the user selects “edit requirements” on the Main Window menu. After the user makes this selection, the system opens the Design Requirements Window. When the user selects the “daylight requirements” bar in this window, the bar will be expanded into a daylight editor. The user can then edit FUs’ daylight requirements. When the user finishes editing, he / she can select the “save changes” button, and the daylight requirements will be saved.

- Participating Objects

  Main Window, Design Requirements Window, Daylight Editor, daylight record.

- Special Requirements
- Pre-conditions

• Post-conditions

The daylight record is saved.

D.10 Edit floors

Edit number of floors, each floor's area, and allowed area tolerance level.

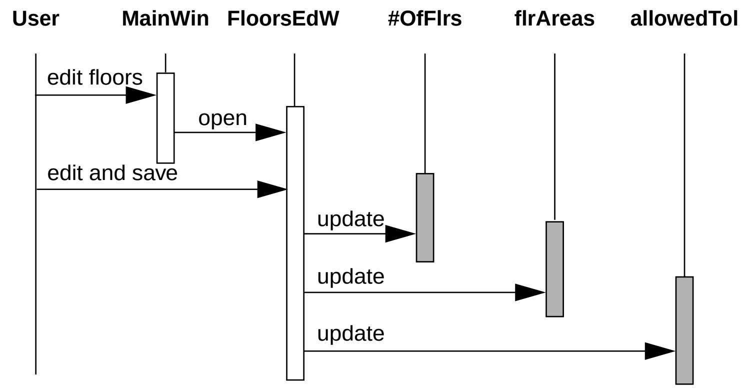

Figure D.11 Interaction diagram of editing floors

D.10.1 Flow of Events

• Associations

This use case starts when the user selects "edit floors" on the Main Window menu. After the user makes this selection, the system will open the Floor Editor Window. The user can then edit the number of floors, each floor's area, and allowed area tolerance. After the user finishes editing and selects "OK", the Floor Editor Window closes and the floors' requirements are updated.

• Participating Objects

Main Window, Floor Editor Window, number of floors, floors' areas, and allowed area tolerance.

• Special Requirements
• Pre-conditions
• Post-conditions

The floors' requirements are updated.

D.11 Select tree/floors representation

Set tree / floor view in the Main Window.

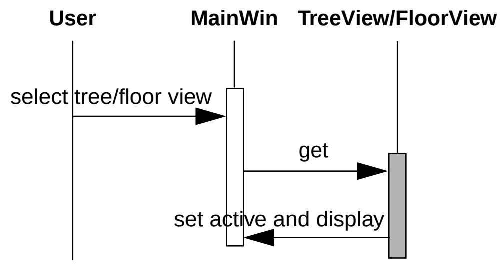

Figure D.12 Interaction diagram of selecting tree/floor view

D.11.1 Flow of Events

- Associations

  This use case starts when the user selects "select tree view" or "select floor view" on the Main Window menu. After the user makes this selection, the system will get the tree view or floor view and set it as active, and display it in the Main Window. This use case ends when the tree view or floor view is displayed.

- Participating Objects

  Main Window, tree view

- Special Requirements
- Pre-conditions
- Post-conditions

  Either tree view or floor view is set as active and displayed in the Main Window.

D.12 Show/hide process of algorithm

Choose to show / hide processes of algorithms in the Main Window.

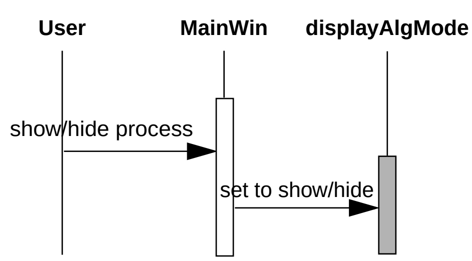

Figure D.13 Interaction diagram of setting viewing/hiding processes of algorithm

D.12.1 Flow of Events

- Associations

  This use case starts when the user selects “show process” or “hide process” on the Main Window menu. After the user makes this selection, the system will set the mode of either displaying or hiding the algorithms’ processes. This use case ends when the mode of displaying the algorithmic processes is set to “show” or “hide”.

- Participating Objects

  Main Window, mode of displaying algorithmic processes.

- Special Requirements
- Pre-conditions
- Post-conditions

  The mode of displaying algorithmic processes is set to “show” or “hide”.

D.13 View Requirements of a tree node or a room FU

View the requirements of a tree node or a room FU.

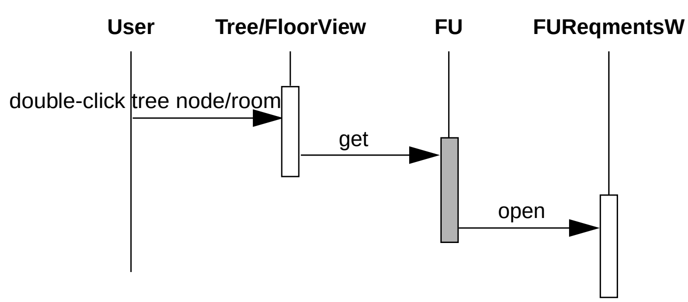

Figure D.14 Interaction diagram of viewing FU requirements

D.13.1 Flow of Events

- Associations

  This use case starts when the user double clicks on a tree node in a tree view or on a room in a floor view in the Main Window. After the user does this, the system will open the FU Requirements Window for the selected FU. This use case ends when the FU Requirements Window is opened.

- Participating Objects

  Tree / floor view, tree node / room, FU, FU Requirements Window.

- Special Requirements
- Pre-conditions

- Post-conditions

  The FU Requirements Window is opened.

D.14 Start stacking

Run the stacking algorithm and generate an FU hierarchy.

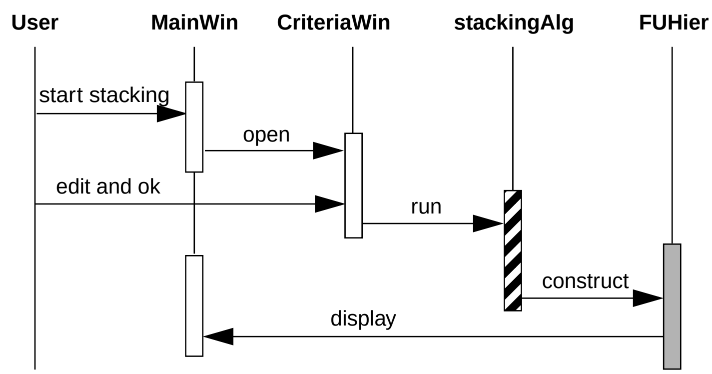

Figure D.15 Interaction diagram of starting stacking

D.14.1 Flow of Events

- Associations

  This use case starts when the user selects “start stacking” on the Main Window menu.

  When the user makes this selection, the system opens the Stacking Criteria Window, which prompts the user to select a set of criteria among the three (adjacency, thermal, and acoustic). It also prompts the user to define the relative importance of the selected criteria. When the user selects the “OK” button, the Stacking Criteria Window closes and the algorithm begins to run. A progress bar indicates the status of the algorithm. In the end, the algorithm determines an optimal solution and constructs an FU hierarchy in the Main Window, in either tree view or floor view, according to the preferred representation. If no feasible solution satisfies the given floor area requirements, a warning dialogue window will be displayed.

- Participating Objects

  Main Window, Stacking Criteria Window, adjacency, thermal, and acoustic criteria, stacking algorithm, FU hierarchy.

- Special Requirements
- Pre-conditions

  Number of floors, floors’ areas are defined.

- Post-conditions

An optimal solution is obtained and a corresponding FU hierarchy is displayed in the Main Window, in either tree view or floor view according to the preferred representation. If no feasible solution satisfies the given floor area requirements, a warning dialogue window will be displayed instead.

D.15 Start zoning

Run the zoning algorithm and generate zones for each floor.

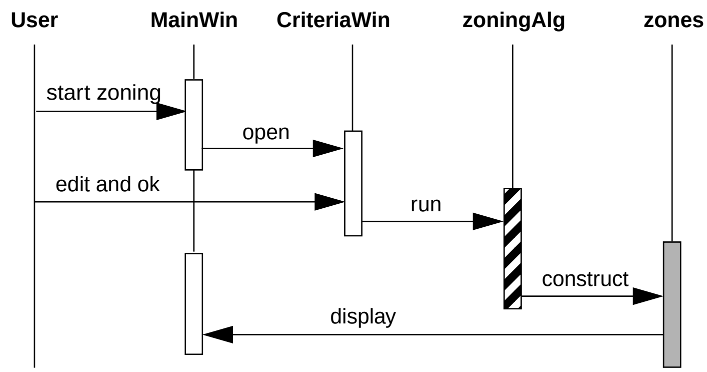

Figure D.16 Interaction diagram of starting zoning

D.15.1 Flow of Events

- Associations

  This use case starts when the user selects “start zoning” on the Main Window menu.

  When the user makes this selection, the system *a)* opens the Zoning Criteria Window, which prompts the user to select a set of criteria among the four, namely adjacency, thermal, acoustic, and daylight requirements, *b)* prompts the user to define the relative importance of the selected criteria, and *c)* prompts the user to define a threshold value. When the user selects the “OK” button, the Zoning Criteria Window closes and the algorithm begins to run. A progress bar indicates the status of the algorithm. In the end, the algorithm finds a zoning solution and constructs zones in the Main Window, in either tree view or floor view according to the preferred representation.

- Participating Objects

  Main Window, Zoning Criteria Window, adjacency, thermal, acoustic, and daylight criteria, zoning algorithm, zones.

- Special Requirements
- Pre-conditions

  Stacking is finished, i.e., FUs have been assigned to floors.

- Post-conditions

The generated zones are displayed in the Main Window, in either tree view or floor view according to the preferred representation.

D.16 Attach a tree node as a child of another tree node

Move a tree node to another tree node and attach it as a child of the destination node.

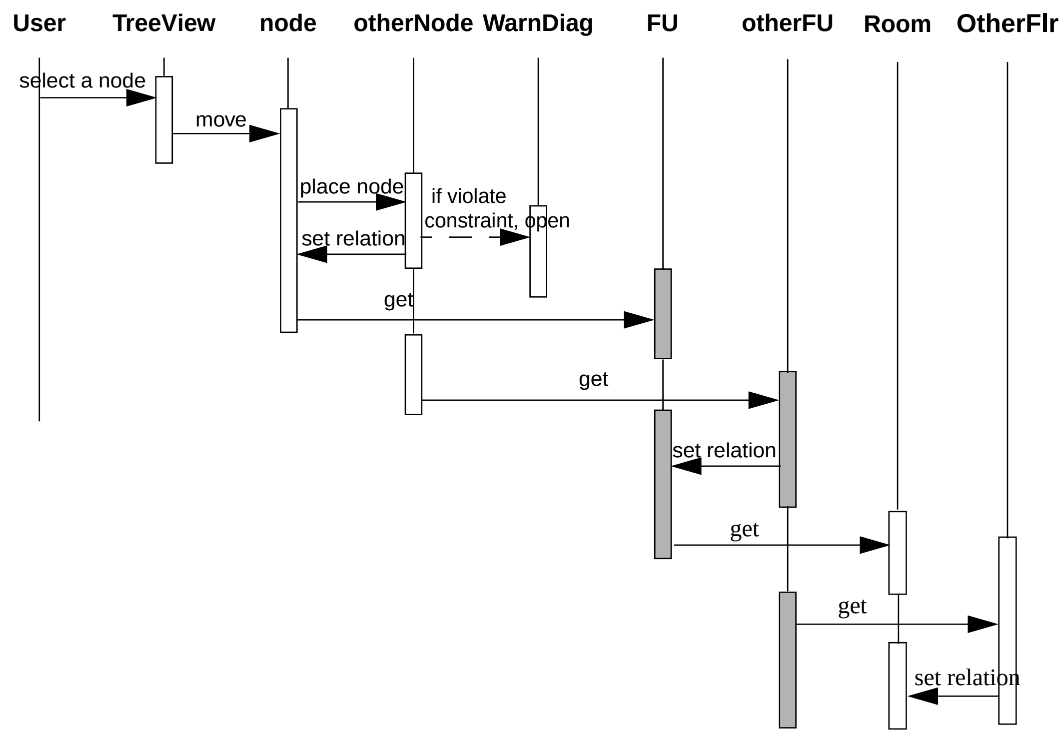

Figure D.17 Interaction diagram of moving a tree node

D.16.1 Flow of Events

- Associations

This use case starts when the user selects a tree node, drags it, and drops it on another tree node in the tree view in the Main Window.

If the destination FU cannot be a parent of the selected FU due to a violation of certain constraints, a warning dialogue window will be opened informing the user of this violation. Otherwise, the selected FU will become a child of the destination FU. The floor view will be updated correspondingly.

- Participating Objects

Tree view, selected node, another node, FU, another FU, warning dialogue window if applicable, room, and another floor.

- Special Requirements

- Pre-conditions
- Post-conditions

  The selected FU becomes a child of the destination FU. The floor view is updated correspondingly.

D.17 Move a room to another floor

Move a room from its current floor location to another floor.

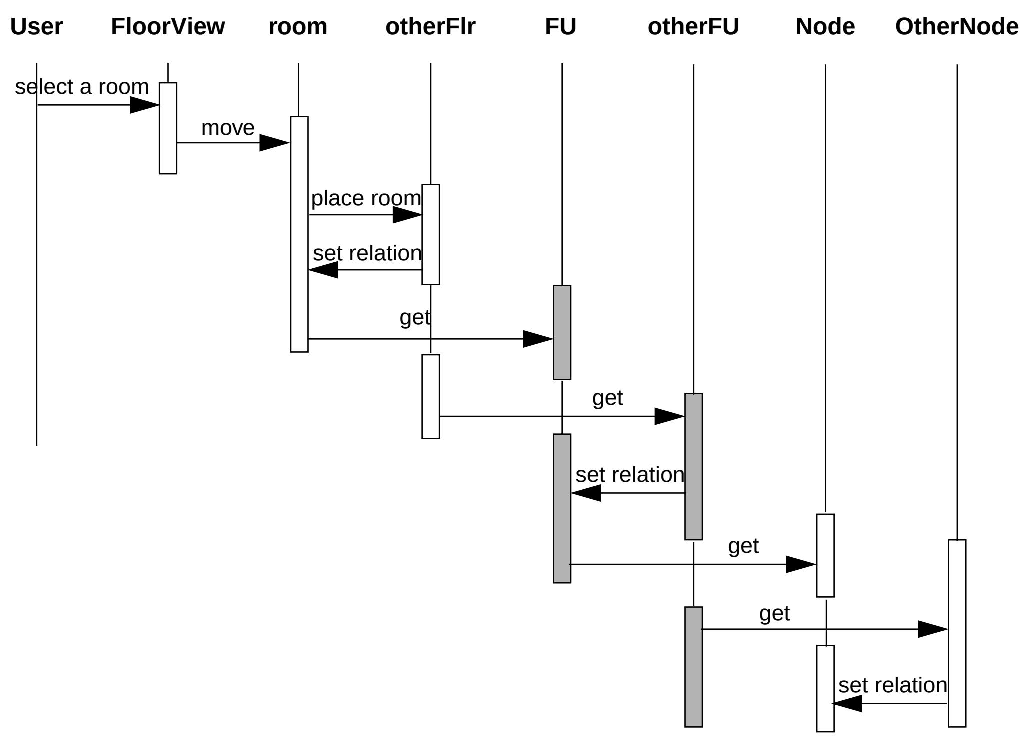

Figure D.18 Interaction diagram of moving a room

D.17.1 Flow of Events

- Associations

  This use case starts when the user selects a room, drags it, and drops it on another floor in the floor view in the Main Window.

  The selected room will be placed on the destination floor. The tree view will be updated correspondingly.

- Participating Objects

  Tree view, selected room, another floor, FU, another FU, tree node, and another tree node (another floor).

- Special Requirements
- Pre-conditions
- Post-conditions

  The selected room is placed on the destination floor. The floor view is updated correspondingly.

D.18 Collapse/expand a tree node

Collapse or expand a tree node.

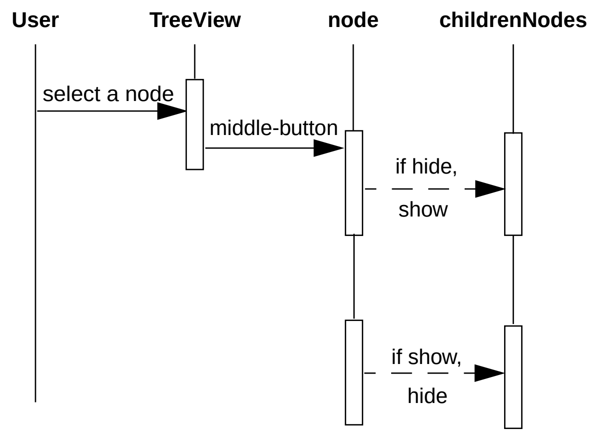

Figure D.19 Interaction diagram of collapsing/expanding a tree node

D.18.1 Flow of Events

- Associations

  This use case starts when the user presses the mouse's middle button to click on a tree node in the tree view in the Main Window. When the user does this, all of the children of the selected node will be hidden if they were originally shown; or they will be shown if they were originally hidden.

- Participating Objects

  Tree view, tree node and its children nodes.

- Special Requirements
- Pre-conditions
- Post-conditions

  The children nodes of a selected tree node are either hidden or shown, depending on their initial state.

Appendix E:
Daylight Decomposition

E.1 Identifying Requirements

Problem analysis, also known as content analysis, was carried out in order to identify the requirements in daylight decomposition. Problem analysis is a method to extract important factors by examining the relations between different factors being surveyed.

In order to analyze daylight conditions under which FUs can be grouped or should be separated, the example of a floor plan in a multi-story building was considered (see Figure E.1). In this plan, there is one internal zone and four daylight zones. Each of the daylight zones face a certain orientation such as north, south, east, and west.

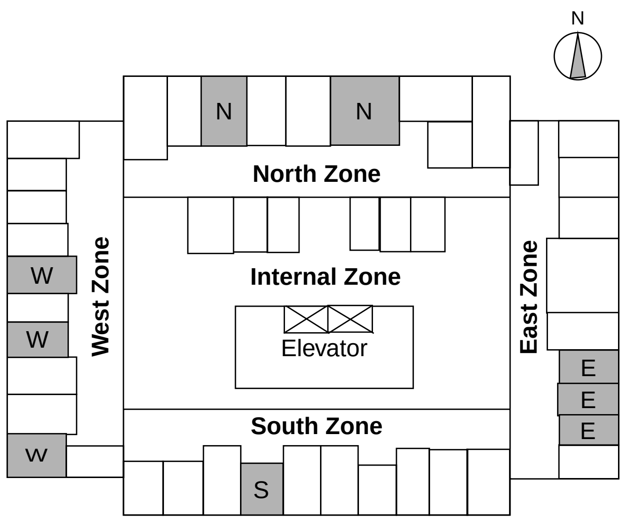

Figure E.1 Sample floor plan with daylight zones

In order to meet FUs' orientation requirements, each FU should be located within a zone that has an external exposure to that orientation. For instance, an FU requiring an eastern exposure should be located within the east zone.

When multiple FUs are assigned to the same daylight zone because of sharing the same orientation requirements, these FUs may not be physically grouped together (see shaded FUs in Figure E.1). Therefore, daylight requirements do not have any grouping implications. This makes daylight zones different from other architectural zones, such as thermal zones or acoustic zones.

E.2 Representing Requirements

The goal of daylight decomposition is to group FUs with the same orientation requirements into the same zone on each floor.

The input of daylight requirements for a set of FUs includes their orientation preferences. For instance, an FU may need daylight from a particular direction, such as north, south, east, or west. Alternatively, it may require an external exposure without specification of a particular orientation.

Daylight is not a factor to consider in stacking because relations between FUs and daylight are horizontal and do not affect the vertical distribution of FUs onto different floors. To handle special cases such as FUs requiring skylight, these FUs can be pre-assigned to the top floor.

Daylight requirements are important in zoning. FUs with the same orientation requirements should be allocated within the same daylight zone so that they can have external exposure to that orientation.

Daylight requirements can be represented in a graph where FUs with the same directional constraints are linked together with strong weights. Figure E.2 shows an example of such a graph. In this figure, each node is an FU. FUs connected by edges share the same required orientation of exposure. For instance, Laser, Photo, CT_scan, and Treatment rooms all need to have an eastern exposure; therefore, a strong relation (e.g., weight 100) is set up between all pairs of these FUs. Likewise, for FUs with generic daylight requirements, such as Reception and Doc_off, a strong weight is set up between all pairs of these FUs.

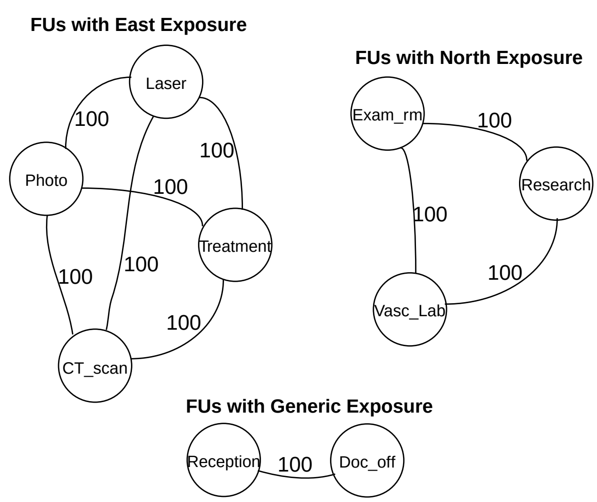

Figure E.2 Representing FUs with daylight requirements in graph

With daylight requirements represented in a graph, the same zoning algorithm can be applied, as described in Section 3.6.2.2.

E.3 Output

By partitioning the graph, FUs sharing the same directional constraints will be grouped into the same zone. As a result, four directional zones are formed, including north zone, south zone, east zone, and west zone. If there are FUs bearing generic daylight requirements, a generic daylight zone will be generated as well.

This zoning result is part of the input to the SEED-Layout module, and the generic daylight zone will be represented in union with the four directions. In other words, a generic zone will be interpreted as having an exposure to any one of the four orientations.