CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Predicting Space Utilization of Buildings through Integrated and Automated Analysis of User Activities and Spaces

By

Tae Wan Kim

CIFE Technical Report #TR214
June 2013

STANFORD UNIVERSITY

If you would like to contact the authors, please write to:

c/o CIFE, Civil and Environmental Engineering Dept.,
Stanford University
The Jerry Yang & Akiko Yamazaki Environment & Energy Building
473 Via Ortega, Room 292, Mail Code: 4020
Stanford, CA 94305-4020

PREDICTING SPACE UTILIZATION OF BUILDINGS THROUGH INTEGRATED AND AUTOMATED ANALYSIS OF USER ACTIVITIES AND SPACES

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF

CIVIL AND ENVIRONMENTAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Tae Wan Kim

May 2013

Abstract

Space-use analysis (SUA) provides architects with predictions about space utilization based on user activity and space information during project development to support their effort of meeting the design intention regarding space-use. Specifically, SUA requires architects to map user activities onto appropriate spaces and then to compute utilization of each space. However, architects currently are limited to manually predicting utilization, and, therefore, SUA is time-consuming when architects update utilization in response to changes in information about user activities and spaces and is inconsistent across architects.

To address these problems, I developed an automated SUA method based on a knowledge-based systems approach. This approach is based on two specific theoretical contributions. First, I developed an ontology for representing user activities for use in SUA. This ontology consists of 10 classes (four concepts and their subclasses) and 24 properties and their facets. Based on this ontology, architects can represent a user activity in a <User>, an <Action>, and two <Spatial requirements> instances. This ontology allows a computer to distinguish between different space-use types that affect SUA, whereas combined existing theories of representing user or construction activities require architects to manually keep track of which space-use type each user activity falls into during SUA. I validated the proposed ontology by representing 28 user activities identified in three case studies and by determining whether or not it can capture all space-use type differentiators, from which 288 space-use types are derived.

Second, I formalized a method for mapping user activities onto appropriate spaces, which consists of “choose spatial requirements,” “find spaces,” and “map the activity onto spaces” steps. This method formalizes knowledge about how to deal with different space-use types that affect SUA during the mapping, whereas existing user presence models lack this knowledge and, therefore, require architects to map user activities onto appropriate spaces manually. I validated this method by measuring the conformity of the activity-space pairs generated by this method with the pairs generated manually by five architects in two cases and by asking the architects to score the acceptability of the method.

Based on these two contributions, I defined the process of the knowledge-based space-use analysis (KSUA) method, which consists of the following four functions: “building the knowledge base,” “mapping user activities onto spaces,” “computing utilization,” and “visualizing the results.” I demonstrated the effectiveness of the KSUA method by developing a prototype system and conducting a trial run on selected areas of the Jerry Yang and Akiko Yamazaki Environment and Energy (Y2E2) Building at Stanford University. The results show that the KSUA method leads to quick updating of utilization with high consistency in SUA.

I also conducted a charrette test to provide quantitative evidence of power and generality of the KSUA method. As a result, six graduate students (novice architects) who participated in the test could update the utilization of each space 6.5 times faster with the KSUA method than the manual SUA. In addition, the standard deviation of predictions across the students with the KSUA method was 68% less than the standard deviation of predictions across the same students with manual SUA, which shows the improved consistency in predictions by the KSUA method. I conducted the test on two cases (one educational building and one office building), each of which has three trials (an initial prediction and two updates), which shows the generality of the KSUA method.

Deployment of the KSUA method in project development should enable architects to achieve high utilization of buildings without over- or under-used spaces. Buildings are more likely to support user activities and the future growth plan of an organization that uses the buildings. Achieving the minimum amount of spaces in a building contributes economic and environmental sustainability because building cost and energy use generally grow in proportion to the size of the building.

Acknowledgments

I worked for a construction management company after finishing my Master's degree in South Korea. My main job was to lead the knowledge management team in the company. Although acquiring the right knowledge and providing the knowledge to the right place at the right time is extremely important in the business of construction management, I felt that the way that knowledge was managed fell short because it was basically building a cabinet system and put all "documents of knowledge" into an appropriate box in the cabinet. I decided to study virtual design and construction (VDC) at Stanford University because VDC works in a different way for the same purpose. VDC formalizes relationships among different product, organization, and process elements of a construction project (right knowledge) so that the elements can "talk" to each other. In that way, if one element changes its properties during the project, other elements that are affected by this element (right place) change their properties immediately (right time). I am very grateful that I could join the Center for Integrated Facility Engineering (CIFE), Stanford University, and contribute the formalization of the relationship between user activities and spaces to VDC.

The key point I do not want to miss is that I could not have made this contribution without many people I have encountered during my PhD years. These people have been the gates of blessing God gave me. I would like to print their names on this paper as well as on my mind so that I can cherish their full-hearted support and, sometimes, even sacrifice and make myself a promise to dedicate my life to being a gate of blessing to those who need my support and sacrifice.

My achievement during the PhD would not have been possible had I not been advised by Professor Martin Fischer, my principal advisor. I am very grateful that I have been able to have him during the whole process of my PhD work. He provided me with great insight about my academic interests, helped me be in the middle of relevant practice by having me conduct an internship in a workplace planning company in Finland, and guided me to move my work forward throughout the writing process. I would also like to mention that, without his advice and encouragement, I would not have been supported by three fellowships and a research grant during my PhD years.

I also truly appreciate the help and support received from my defense committee members. Professor Ram Rajagopal provided me with great knowledge about knowledge representation and reasoning and knowledge-based systems when I had a problem to solve and did not know how to solve the problem. Dr. Calvin Kam encouraged me to meet many industry practitioners and embrace their knowledge about space-use analysis and their concerns in my work. Professor Raymond Levitt, with unmatchable insight on research, pushed me to have a thorough understanding about relevant theories and build my work on those theories.

I also would like to thank CIFE community members for their support during my PhD years. They readily spent their precious time to discuss my work and provided me with sincere and thoughtful feedback. They also helped me interview expert architects and conduct surveys to understand what is going on in reality. Hence, many thanks to Dr. John Kunz, Professor Arto Kiviniemi at the University of Salford, Dr. Jonghoon Kim at DPR Construction, Professor John Haymaker at Georgia

Institute of Technology, and Seunghwan Shim at Hines. I also appreciate Teddie Guenzer, administrative manager at CIFE, for her kind and timely support. I will also never forget help from my world-best colleagues: Amir Kavousian, Anne-Laure Cuvilliez, Dr. Atul Khanzode, Austin Becker, Professor Claudio Mourgues, David Park, Dr. Forest Flager, Forest Peterson, Professor Ju Gao, Jung In Kim, Matt Garr, Mauricio Toledo, Min Song, Natasa Mrazovic, Nelly Garcia-Lopez, Dr. Peggy Yee, Dr. Reid Senescu, Rene Morkos, Robert Graebert, Roshan Mehdizadeh, Sangwoo Cho, Tetsu Kondo, Dr. Ning Dong, Dr. Tobias Maile, Dr. Victor Gane, and Wendy Li.

Beyond the CIFE community, many others also helped me find the motivating cases and elaborate the research: Professor Yinyu Ye from the Department of Management Science and Engineering (Stanford), Dr. Vinay Chaudhri from the Department of Computer Science (Stanford), Claude Reichard, Evelin Sullivan, Mary McDevitt, and Sandra Drake at the Technical Communication Program (Stanford), Professor Hae Young Noh at Carnegie Mellon University, Professor Jung-ho Yu at Kwangwoon University, Jaegon Kim and Juyeon Cho at Seoul National University, Professor Chuck Eastman at Georgia Institute of Techonology, and Dr. Debizet Gilles at Pacte.

Many parts of my work have been done in collaboration with people in the construction industry. Communications with these people, including internship experience, case observations, and validation tests, greatly helped me balance between theory and practice. Although they were really busy, they invited me to their lives with frank opinions and warm smiles. I would like to thank Dr. Ari Pennanen and Markus Mikkola at Haahtela for providing me with an internship opportunity and sharing their knowledge about workplace planning. I would also like to thank Jonghoon Kim, Dr. Sanghyuk Park, Heesuk Chung, and Kyunghwui Park at HanmiGlobal for providing me with an internship opportunity and sharing their knowledge about space programming and space-use analysis. I also appreciate invaluable inputs from Auli Karjalainen at Senate Properties, Tomi Henttinen at Gravicon, Kimon Onuma at Onuma Inc., Rolf Jerving and Ole Kvarsvik at dRofus Inc., Catherine Chan at HDR Architecture, Sandy Tanaka at Devcon Construction, Kyusik Shin at BDP, Yongjoo Kim at Cambridge Seven Associates, and Insoo Kang at DA Group.

Most importantly, I would like to dedicate my dissertation to my family, who has made me who I am. I truly appreciate my father, Joongsoo Kim, my mother, Insook Kwon, and my brother, Taesung Kim, for their love and support. I also feel very lucky and grateful to my father-in-law, Jungho Ahn, my mother-in-law, Okim Yu, and my brother-in-law, Seungjun Ahn, who supported and encouraged my journey to PhD. I thank God every day for allowing me to be with my lovely and devoted wife, Juyeon Ahn. Without her sacrifice and understanding, I would have not even dreamed about finishing this exciting but challenging quest. Finally, I would like to thank my two sons, Austin (Soohyun) and Nathan (Jaehyun), for just being with me and smiling at me. I am refreshed every day by your smiles.

I will now pursue a new quest with the experience and knowledge I have earned during my PhD years. However, I feel very confident that I can get through any difficulties, not because I am invincible, but because I know the gates of blessing God gives me are truly invincible. I hope I can also be one of the gates of blessing for as many people as possible throughout my life.

Table of Contents

List of Tables

List of Illustrations

1. Introduction

Advances in the representation of product (or Building Information Model), organization, and process models in the construction industry, in tandem with advances in the formalization of various performance analysis methods, have facilitated the use of performance-based building, where architects test multiple design options based on the performance criteria in a quick and consistent way and refine the design to best meet their design intention (Becker 2008; Carrara 1994; Fischer 2006). In this context, space-use analysis (SUA) takes into account user activities and spaces to inform architects and other stakeholders about the performance of design options in terms of space-use, i.e., how much each *space* of a building will be occupied by *building users*[1] and *their activities*, in support of the architects' effort of meeting the design intention. SUA is gaining more significance because many companies or public agencies are disposing of or condensing their workspace in response to a challenging economy.

Space-use has three different perspectives: the space perspective, which questions if spaces are highly utilized, the user perspective, which questions if all building users can work as they expect, and the activity perspective, which questions whether a building supports the activities an organization needs to carry out its business. Rarely used space is not satisfactory from the space perspective, while crowded space is not satisfactory from the user and activity perspectives. These two

[1] *Users* and *building users* are interchangeable in this research unless other definitions are provided.

types of spaces are often present in a building at the same time. Therefore, if architects can determine which space will be rarely used and which will be crowded during project development, they can minimize these spaces while at the same time achieving high utilization of them (Sections 1.1 and 1.2).

There are two ways to determine which spaces will be rarely used, crowded, or properly used: a conventional guidelines approach, which chooses and uses historical data as guidelines (Erhan, 2003; Stanford University, 2009) and SUA, which predicts and uses space utilization[2] as a metric for space-use. A guidelines approach can fail to inform architects about project-specific space-use because it takes into account only the fraction of information that is needed for comparison with the guideline. In reality, space-use is driven by building users and their activities in a building. Thus, omitting users or their activities in a guidelines approach can mask the real needs for a particular type of space (Section 1.3). In contrast, SUA can provide project-specific information about space-use by predicting utilization, a direct measure of space-use, based on user activities and spaces. Specifically, SUA requires architects to map user activities onto appropriate spaces and then to compute utilization of each space. However, architects currently are limited to manually predicting utilization, and, therefore, SUA is time-consuming when architects update utilization in response to changes in information about user activities and spaces and is inconsistent across architects (Section 1.4).

---

[2] *Utilization* and *space utilization* are interchangeable in this research unless other definitions are provided.

To address these problems of time-consuming updating utilization during SUA and inconsistency in prediction of utilization, I aimed to develop an automated SUA method with focus on office and educational buildings. I also limited the scope to the programming phase of a project, where user profiles and a space program provide input information to SUA (Section 1.5). Throughout the literature review, I found that a knowledge-based system (Akerkar and Sajja 2010; Brachman and Levesque 2004) provides an inference engine, which supports automated reasoning about project-specific information and project-independent domain knowledge, once this knowledge and information are captured in the knowledge base of the system. Hence, to enable an automated SUA based on a knowledge-based systems approach, I reviewed existing theories concerning representation of user activities and spaces and formalization of knowledge about mapping activities onto spaces and computing utilization (Section 1.6). I discovered two theoretical gaps and defined two research questions to bridge the gaps (Section 1.7):

- What is an ontology for representing user activities for use in SUA?
- What is a method for mapping user activities onto appropriate spaces?

1.1. Three types of spaces in terms of space-use

There are three types of spaces, which differently affect three perspectives of space-use (Figure 1).

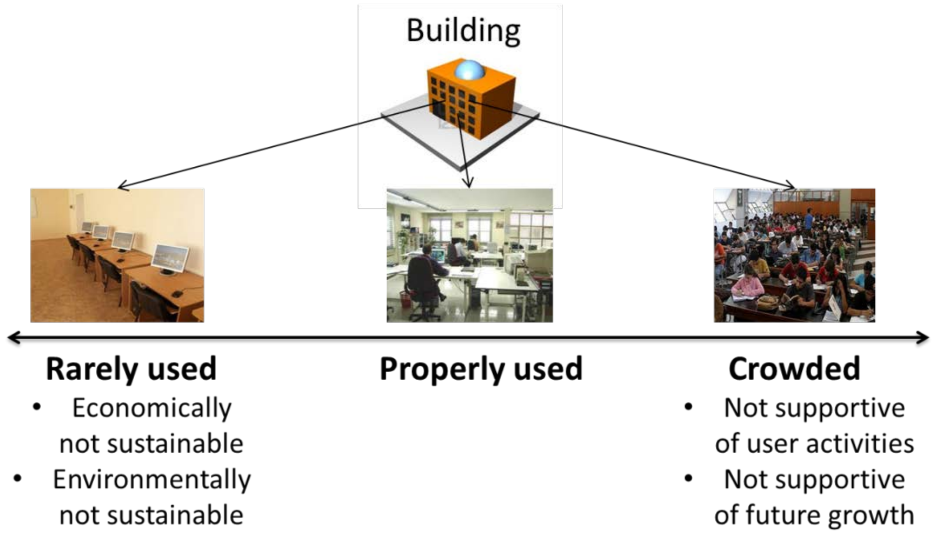

Figure 1. Three types of spaces in terms of space-use: these types of spaces are often present in a building at the same time.

- *Rarely used space* is a space that is occupied by building users less frequently than the design intention specifies. Although this space satisfies the user and activity perspectives, it does not satisfy the space perspective. It is economically not sustainable. For example, according to Stanford University's Space and Furniture Planning Guidelines (2009), an average space cost, including construction costs plus fees, permits, furniture, and equipment, for new building projects at Stanford University is 529 US dollars per square foot. Many single-unit rate cost estimating methods, such as the square foot method and the functional area method (Dell'Isola 2002), postulate that building cost grows in proportion to the floor area of the building. Rarely used spaces are also environmentally not sustainable. For example, average energy use intensities in the US in 2003 were 293 kWh per square meter for office buildings and 262 kWh per square meter for educational buildings (Pérez-

Lombard et al. 2008). Energy use is closely related to the floor area (ECCJ 2010; Sharp 1996). In addition, bigger floor areas lead to more—unnecessary—use of building material (Van den Dobbelsteen and Wilde 2004).

- *Properly used space* is a space that is occupied by building users with a desired frequency, which is specified in the design intention. This space satisfies all three perspectives, representing highly utilized space without harming the user and activity perspectives.
- *Crowded space* is a space that is occupied by building users more frequently than the design intention specifies. Although this space satisfies the space perspective, it does not satisfy the user and activity perspectives. It is not supportive of user activities because it makes users schedule their activities in an inflexible way or form long queues for the activities in the space (Cherry 1999; IFMA 2009). A building is built to fail from day one if it impairs the work productivity of building users by not providing sufficient spaces for their activities (Vischer 2008; Yu et al. 2007). In addition, crowded space does not support future growth of an organization that uses the building (Cherry 1999).

To minimize rarely used space and crowded space, architects have to determine which spaces are rarely used, properly used, or crowded during project development. If a space is rarely used, this space can accommodate more activities or can be decreased to reduce the gross area of the building. If a space is crowded, this space should accommodate fewer activities or should be increased to better support user activities.

1.2. Motivating cases

I examined two building projects in Korea and one project in the US (Whelton 2004) to study how organizations decide on their spaces during project development. These cases show that the space, user, and user activity perspectives are interrelated and often conflict with each other. Architects should determine space-use and the impacts of different design options on the space-use to support clients' and their own decision-making about the design. However, in these cases, architects determined space-use in an intuitive and experience-based way because they did not know how to integrate the three different perspectives.

1.2.1. Case 1: The area of the gym in a construction company

In 2007, a Korean construction company decided to move its headquarters from Seoul to a provincial town. The gross area (6,060 m^2) and the number of floors (4 floors and 1 basement) were determined to maximize the use of the site according to building regulations. An in-house architectural team was in charge of developing the space program and the design for the new building. During the design phase, one of the company's vice-presidents reviewed the design and thought that the area of the gym should be increased for the 200 employees who worked at headquarters (user perspective). The in-house architectural team adjusted the area in accordance with the vice-president's opinion, but when the president reviewed the design, he thought that the gym was too large (space perspective) for the employees' expected exercise activities (activity perspective) and wanted a reduction in area. This incident shows that conflicting opinions can arise when a client organization has multiple decision-makers. In an interview with one member of the in-house architectural team, he recalled that because the team did not have a formal and consistent method of informing the decision-makers (here, the vice-president and the president) about the space-use of the gym, the team “simply followed” the opinion of the most powerful person, in this case, the president. That is, the area of the gym was modified two times even though the decision-makers lacked a solid ground on which to base their decisions.

1.2.2. Case 2: Storage vs. meeting rooms in a publishing company

In 2010, a publishing company in Korea consulted with an architectural firm about the company’s desire to build a new building to provide more space for 20 employees and to provide the president with an art room for her painting activity. The determined gross area was 660 m^2. During the planning phase, the company wanted to increase the area of the storage room to hold an additional 10,000 books (from 20,000 to 30,000 books) (activity perspective). However, because the project had already exceeded its budget, the company had to reduce other spaces to increase the area of the storage. The architect had several options to address this trade-off, including reducing the area of the art room or the number of meeting rooms. However, without an analytical tool that integrates the space (e.g., meeting rooms, the art room, storage for books), user (e.g., employees, editors, the president), and activity (e.g., having a meeting, editing a book) perspectives, the impact of these options on space-use could not be analyzed and compared in detail. The publishing company eventually decided to reduce the number of meeting rooms from three to two without a clear understanding of the impact of the decision.

1.2.3. Case 3: Shared labs vs. independent labs in a university facility

Whelton (2004) describes the Hearst Memorial Mining Building Seismic and Program Improvement Project on the University of California, Berkeley campus. The gross area of the building was 130,000 ft^2 (12,077 m^2). An architectural firm conducted an architectural programming study, which included developing the space program. After developing the space program, the architect found that the project exceeded the original budget determined by the "project planning guide" and re-examined the space program to find appropriate ways to reduce the cost (space perspective). After investigating various options, the architect and client decided to provide shared laboratories instead of separate and independent laboratories. However, these options were analyzed and explained by an intuition-based discussion among the project committee members rather than by a systematic means of integrating the space (e.g., research laboratories, faculty and graduate student offices, classrooms), user (e.g., students, faculty, staff), and activity (e.g., having a class, conducting an experiment) perspectives, and the decision did not consider how shared laboratories might affect students' laboratory activities.

1.3. A guidelines approach

To determine which spaces are rarely used and which are crowded during project development, architects typically use a guidelines approach, or the so-called "choose-and-use" method (Erhan 2003). Architects choose to use a guideline for the proper number or area of a certain type of space, after reviewing relevant surveys, standards, and post-occupancy evaluations of similar buildings. For example, Stanford University's Space and Furniture Planning Guidelines (2009) specify the acceptable area (net assignable square feet) per station for each classroom type. In Western Europe, the proper number of parking spaces is one space per 150 m^2 of office space and one per apartment (Van den Dobbelsteen and Wilde 2004). Architects can then compare each space with the guideline for the space to determine whether the space is rarely used, properly used, or crowded: if the area or the number of a space type is excessive, the space is deemed to be rarely used. If the area or number is insufficient, the space is deemed to be crowded.

1.3.1. Limitations of a guidelines approach

A guidelines approach can fail to inform architects about project-specific space-use because it takes into account only the fraction of information that is needed for comparison with the guideline. For example, information about what user activities will be accommodated by the space, how long these activities will take, and how long the space will be open daily is not used in this approach even though these factors greatly affect space-use.

This section describes the limitations of the guidelines approach using the simplified publishing company case described in Section 1.2.2. The company has 20 employees, and five of them are editors. Let us say the architect chooses to use 0.1 as the proper number of meeting rooms per employee. The architect then can find the number of employees (20) in user profiles and determine that the proper number of meeting rooms is two (Figure 2).

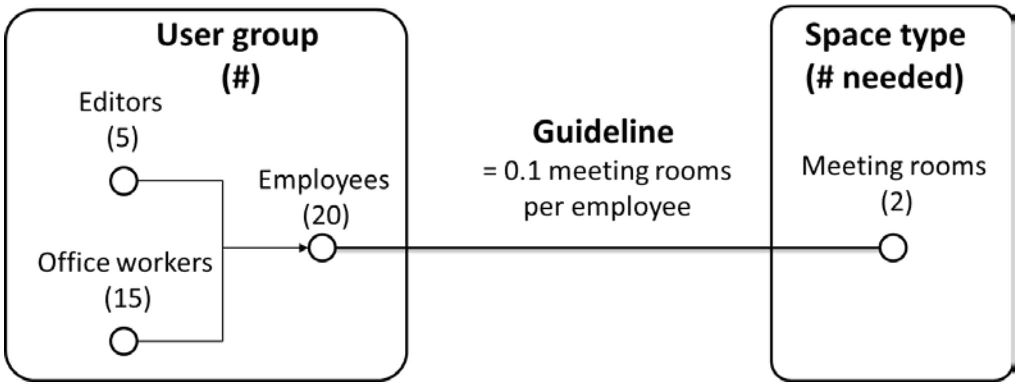

Figure 2. Use of a guideline in the publishing company case: the architect determines the proper number of meeting rooms merely based on the number of employees.

Let us assume that the new building has two meeting rooms (for being properly used) according to the guidelines approach, and each meeting room is open for eight hours daily. According to user profiles of the publishing company, the following two user activities occur in the meeting rooms: book editing and meeting (Figure 3).

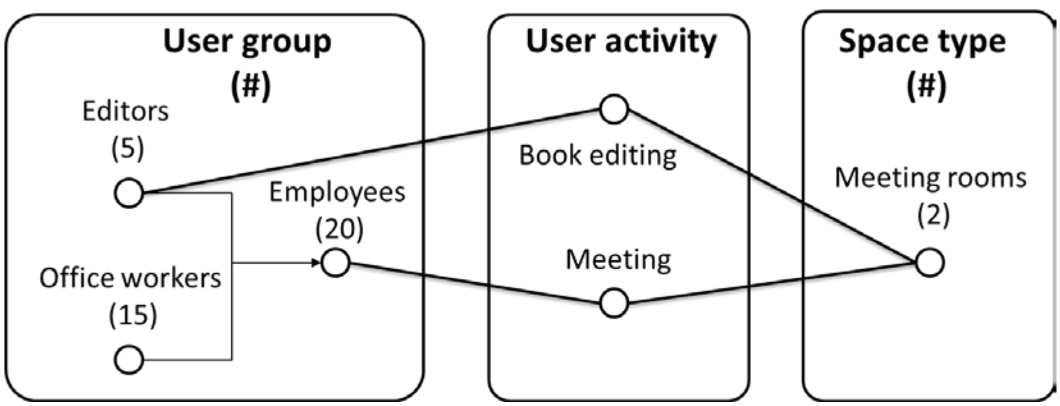

Figure 3. Two user activities occurring in meeting rooms: the space-use of meeting rooms is driven by user activities occurring in the meeting rooms.

- Book editing is performed by five editors. Since editing is individual work, each editor is his or her own group, and the number of groups for this activity is five. Each editor performs one editing task a day (frequency), and each editing task lasts two hours (duration) on average. Therefore, this activity needs to occupy the meeting rooms for 10 hours (= the number of groups × the frequency × the duration). Since there are two meeting rooms in the building, the ten hours are split equally into the two rooms, making each room accommodate this activity for five hours.

- Meeting is performed by 20 employees. Since four employees participate in a meeting on average, the number of groups for this activity is five (= 20 ÷ 4). Each employee joins two meetings a day, and each meeting is one-hour long on average. Therefore, this activity needs to occupy the meeting rooms for 10 hours, and each of the two meeting rooms accommodates this activity for five hours.

In summary, each meeting room needs to accommodate 10 hours of activities (five hours for book editing, five hours for meeting), while it is open for only eight hours daily. The meeting rooms will not be properly used as the guideline suggested. This case example highlights that a guidelines approach can mislead architects about space-use because it does not explicitly take into account user activities by which space-use is driven. Omitting any user activity occurring in a space can mask the real needs for the space.

1.4. Space-use analysis approach

Space utilization is a metric for space-use: 0.50 of space utilization means that there is a fifty-fifty chance the space will be occupied by users when a person opens the door to the space. It can be calculated by dividing total activity loads (or total loads) by the open hours of the space (Pennanen 2004). For example, in the case in Section 1.3.1, since the book editing activity occurs for five hours and the meeting activity occurs for five hours in a meeting room that has eight open hours, the utilization of the meeting room is 1.25 (= 10 ÷ 8). Space-use analysis (SUA) takes project-specific information about user activities and spaces during project development as input and yields predicted utilization of each space as output. SUA requires an architect to follow the two steps to predict utilization: (1) mapping user activities onto appropriate spaces and (2) computing utilization of each space.

To enable architects to understand what a certain range of utilization implies, I categorized four utilization implication levels for non-designated spaces and two utilization implication levels for designated spaces based on the previous work that suggests the implications of utilization (Cherry 1999; Pennanen 2004). Each utilization implication level has a different color-code for enabling the visualization (Table 1).

Architects can determine whether a space will be rarely used, properly used, or crowded in the occupancy phase of a building by comparing the implication level of the predicted utilization with the policy on utilization (Cherry 1999), which is also represented by a utilization implication level. In the frame of performance-based buildings (Becker 2008), the policy on utilization functions as a performance requirement. To ensure that a design option satisfies the requirement, space-use analysis predicts utilization and compares it with the requirement to prevent rarely used or crowded spaces (Figure 4).

Table 1. Utilization implication levels: what does a certain range of utilization imply?

Non-designated spaces			
Range of utilization	Implication	Description	Color-code
utilization <=0.50	No wait (NW)	Activities can be done without waiting.	Green
0.50< utilization <=0.75	Adequate (AD)	Activities may need to be scheduled.	Yellow
0.75< utilization <=1.00	Inconvenient (IC)	Activities need to be relocated.	Red
1.00< utilization	Infeasible (IF)	Activities cannot be physically accommodated.	Gray
Designated spaces			
Range of utilization	Implication	Description	Color-code
utilization <=1.00	No wait (NW)	Activities can be done without waiting.	Green
1.00< utilization	Infeasible (IF)	Activities cannot be physically accommodated.	Gray

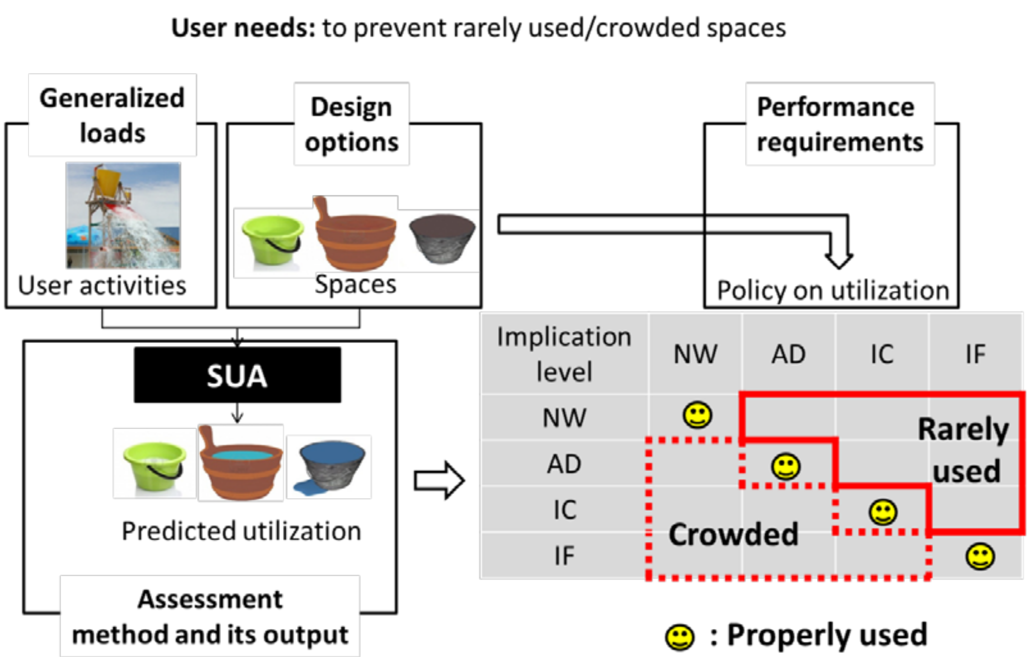

Figure 4. Determining whether a space will be rarely used, properly used, or crowded: space-use analysis (SUA) functions as an assessment method in the frame of performance-based buildings.

1.4.1. Limitations of manual space-use analysis

Despite the project-specificity that SUA brings to the practice of determining space-use in support of performance-based buildings, architects are limited currently to predicting utilization manually based on user activities described in user profiles and a space program or design. As a result, SUA is time-consuming when architects update utilization in response to changes in information about user activities and spaces, and the predicted utilization is inconsistent across architects.

I conducted an experiment where six Stanford graduate students (as novice architects) and six architects (as expert architects, with an average experience of 13.2 years) were requested to predict utilization given the information about four user activities and three space types in the publishing company case described in Section 1.2.2 (Figure 5). They were also asked to update the predicted utilization according to the two changes I made in input information: (1) the number of meeting rooms is decreased from three to two, and the art room, which was originally devoted to the company president's painting activity, can now be used for other activities. (2) In addition to the first change, space preferred by editors is no longer provided for them; this change makes editors perform editing at a workstation instead of a meeting room.

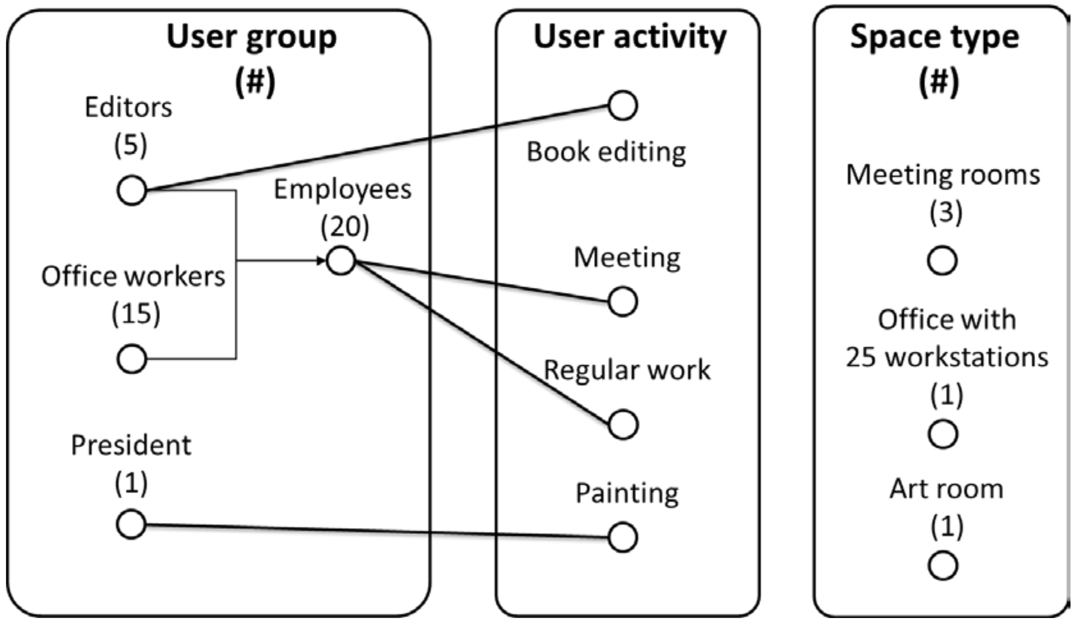

Figure 5. User activities and spaces for the manual space-use analysis experiment: participants performed space-use analysis manually taking into account four user activities and three space types.

As a result, even with only four user activities and three spaces to take into account, students and architects who participated in the experiment spent 6.7 minutes and 4.0 minutes, respectively, to update their prediction according to changes in information about user activities and spaces. More importantly, they spent 28% (for students) to 35% (for architects) of their initial prediction time for each update as their initial prediction times were 24.2 minutes and 11.5 minutes, respectively. Since information about user activities and spaces changes very frequently during project development, reducing this “updating-initial prediction time ratio” is important in space-use analysis to enable architects to respond to these changes quickly.

In addition, given the same information about user activities and spaces, utilization values initially predicted by the six students ranged from 0.29 to 0.82, implying very different situations in terms of space-use. The values updated by the students showed similar variation (0.29 to 1.02 for the first update, 0.27 to 0.70 for the second update). Although architects predicted and updated space utilization with less variance than students, their predictions did also not provide a consistent basis for design decisions: the values initially predicted ranged from 0.23 to 0.46, and the updated values ranged from 0.26 to 0.66 (first update) and from 0.20 to 0.50 (second update). Standard deviations of utilization values were 0.23 for students and 0.13 for architects (Figure 6).

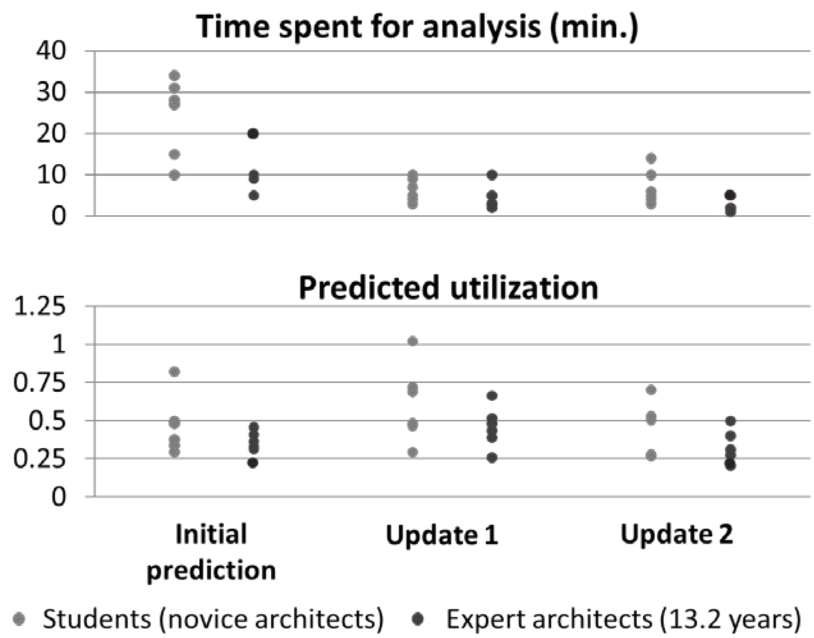

Figure 6. Manual space-use analysis (SUA) experiment: updating SUA manually is time-consuming and is inconsistent across architects.

I found two issues to be the crux of the problem, both of which make manual SUA yield inconsistent results in a time-consuming manner. First, there are various space-use types that affect SUA. For example, in the publishing company case, employees wanted a workstation in the office (i.e., part of a room) for performing regular work, while they wanted a meeting room (i.e., a whole room) for having meetings. Editors preferred a whole room with quiet conditions (i.e., preferred space) for their book editing activity although they could conduct the activity at a workstation in the office (i.e., minimum spatial requirements). The president of the company wanted an art room to be devoted to her painting activity (i.e., designated space). Without computational assistance, it is difficult for architects to keep track of different space-use types and appreciate the impact of these types throughout the SUA.

Second, there are many-to-many relations between user activities and spaces, meaning that a user activity can occur in many spaces, and a space can accommodate many activities at the same time. This complexity creates ($m \times n$) possible pairs of a user activity and a space out of m user activities and n spaces. For example, Pennanen (2004) specifies 95 user activities and 87 spaces for Cygnaeus High School in his dissertation, which creates 8,265 (= 95 × 87) possible activity-space pairs. With this large number of possible activity-space pairs, exploring or optimizing different design options in terms of space-use is often impractical without computational assistance (Flager et al. 2009).

1.5. Research objectives and scope

To address the problems of time-consuming updating during SUA and inconsistency in prediction, I aimed to develop an automated SUA method. This method must (1) predict space utilization, (2) be model-driven to deal with project specificity, (3) distinguish between various space-use types throughout the SUA, and (4) eliminate the human effort of speculating on all the possible activity-space pairs.

I focused only on office and educational buildings because their spaces are determined primarily by user activities, and they have clearer user profiles than other types of buildings. In addition, I limited the scope to the programming phase of a project, where user profiles and a space program provide input information to SUA.

- User profiles refer to a document that describes the user groups, what they do (i.e., their activities), and their functional needs (i.e., spatial requirements) (Cherry 1999). Architects develop this document by identifying two types of usage of a building—assumed and intended—based on the business process diagramming, workshop or interview with clients and users, historical data from similar buildings, trend analysis, etc. Assumed usage is a set of assumptions about the usage of spaces in a building, which function as external factors that architects cannot change during project development (e.g., the expected user groups, their numbers and expected activities); intended usage is the space usage policy that reflects a client's or an architect's intention about how spaces should be used in a building (e.g., keeping a user group from using a space or providing a user group with spaces that satisfy the group's preferences for the group's activities). Intended usage functions as a set of parameters that are changeable during project development. Thus, the automated SUA method must be able to update space utilization in response to changes in intended usage.
- Space program refers to a document that lists the number and the area of each space type. Other features that need to be satisfied by a design, including adjacency, specific equipment, and location, can also be included in text format under the name of "other," "etc.," or "note." Although space location,

geometry, or aesthetics information also affects space-use, I considered only the aforementioned parameters, which are more dominant in SUA.

In the programming phase, architects generally develop the space program based on user profiles, client requirements (e.g., project budget and goals), and regulations. During this development, SUA can help architects determine whether or not the space program they create has the proper numbers of space types, i.e., not too many from the space perspective and not too few from the user and activity perspectives. In addition, clients usually want more spaces at the outset, and the estimated cost can easily exceed the project budget (Peňa and Parshall 2001) as shown in Case 2 (Section 1.2.2) and Case 3 (Section 1.2.3). In this case, SUA enables architects to test different options in terms of space-use so they can reduce spaces with minimal impacts on the user and activity perspectives.

1.6. Overview of points of departure

My intuition was to develop a SUA method that is built based on a knowledge-based systems approach, which I call the knowledge-based space-use analysis (KSUA) method. A knowledge-based system "uses one or more experts' knowledge to process information and to solve a problem in a specific domain" (Akerkar and Sajja 2010). Thus, once project-specific information about user activities and spaces as well as project-independent domain knowledge about SUA are captured in the knowledge base of the system, the inference engine of the system automatically reasons about the information and knowledge to generate activity-space pairs and compute utilization of each space. I also used the theory of workplace planning (Pennanen 2004) because it provides formalized knowledge about computing utilization given activity-space pairs.

In addition, since a space program is in a tabular format, architects can relatively easily represent and capture it in the knowledge base of the KSUA system. However, I found the following two gaps in theories:

First, existing user activity and construction activity representation models (Akinci et al. 2002a; Darwiche et al. 1988; Pennanen 2004) do not represent user activities in a way that distinguishes between different space-use types. For example, these representation models do not represent the case where preferred space and minimum spatial requirements of a user activity are different, and therefore architects have to keep track of this information and speculate on the impact of this information on an ad-hoc basis.

Second, although existing user presence models (Ioannidis et al. 2012; Shen et al. 2012; Tabak 2008) represent both the concept of user activity and the concept of space and relate these two concepts explicitly, these models require an architect to manually map the activities onto the spaces. This is because different space-use types and their impact on the task of mapping have not been formalized and integrated into a single method.

1.7. Research questions and tasks

To enable an automated SUA based on a knowledge-based systems approach, I defined the following two research questions to fill the gaps in existing theories:

- *Q1: What is an ontology for representing user activities for use in SUA?*

  An ontology is formal when all identified user activities in case studies can be represented using the ontology. In addition, a developed ontology is comprehensive when all identified space-use types can be distinguished using this ontology.

- *Q2: What is a method for mapping user activities onto spaces to generate activity-space pairs?*

  A developed method has sufficient conformity for mapping when activity-space pairs generated by this method conform to the pairs generated by expert architects. In addition, a developed method is acceptable when expert architects agree that the method is acceptable from their perspectives.

I conducted four research tasks to answer the two research questions and validate the power of the developed KSUA method. First, I identified and defined space-use types that affect SUA based on literature review, case observations, interviews with architects, and an expert survey (Chapter 3). Second, I developed an ontology for representing user activities, which distinguishes between different space-use types and answers the first research question. I validated this ontology by representing user activities identified in three case studies and by determining whether or not it can capture all space-use types (Chapter 4). Third, I developed a method for mapping user activities onto spaces, which applies different rules to different space-use types and answers the second research question. I validated this method by making expert architects conduct the mapping given information about user activities and spaces and by measuring the conformity and the acceptability of the method (Chapter 5). Last, I developed a prototype based on the findings and validated the KSUA method quantitatively using the prototype. I first applied the prototype to selected areas of the Jerry Yang and Akiko Yamazaki Environment and Energy (Y2E2) Building, Stanford

University, to test the functionality. I used frame logic, or F-Logic (Angele et al. 2009; Kifer 2005), to represent project-specific information about the selected areas and domain knowledge about SUA for the prototype (Chapter 6). Then, using the prototype, I conducted a charrette test (Clayton et al. 1998), where participants predicted and updated utilization given user profiles and a space program without the prototype I developed (i.e., the conventional (manual) process) and with the prototype (i.e., the innovative (automated) process). I measured consistency across the participants and speed of updating utilization for comparison (Chapter 7). Figure 7 depicts these research tasks.

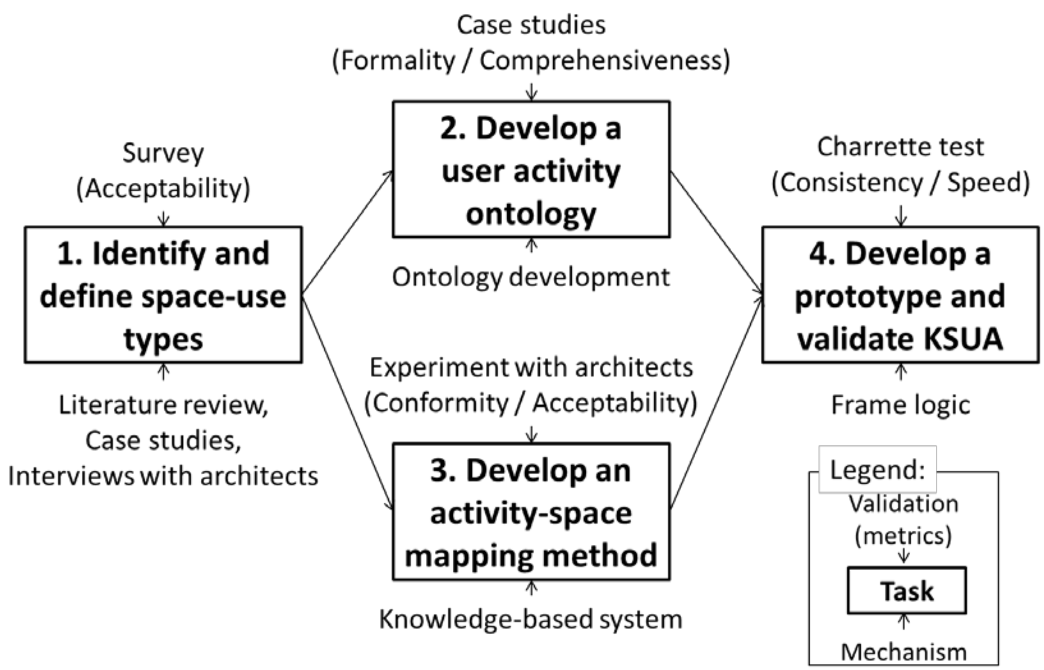

Figure 7. Research tasks: step 2 answers the first research question, and step 3 answers the second research question.

2. Points of departure

I reviewed prior work concerning SUA to examine existing research efforts and space utilization theory and knowledge-based system on which I build the KSUA method. Since the application of knowledge-based systems requires representation of user activities and spaces as well as formalization of knowledge about mapping user activities onto spaces and computing utilization, I also reviewed theories and models related to the representation and the formalization.

2.1. Prior work

The prior work, which provides useful theoretical concepts, consists of the following five domains: (1) architectural programming, (2) post-occupancy evaluation, (3) operations research for space assignment, (4) computer reservation systems, and (5) space planning systems. Although the importance of SUA has been recognized widely (Gibson 2000; Pendlebury 1990), previous research efforts provide only limited knowledge for automating SUA.

Architectural programing (Cherry 1999; Peňa and Parshall 2001; Yu et al. 2008) provides limited formalization of the analysis for the following two reasons: (1) this method does not specialize in SUA, but serves the broader purpose of accommodating a client's demand or an architect's intention. Therefore, although generally accepted steps for this method exist, it is difficult to formalize detailed procedures for each step. Thus, this method heavily depends on an architect's experience and expertise. (2) Although user activities and spaces are interrelated and must be taken into account simultaneously, the relationship between these two is not formalized in this method. Consequently, user activities are often gathered and analyzed on an ad hoc basis, which makes SUA inconsistent and time-consuming.

Post-occupancy evaluation (Preiser et al. 1988; Whyte and Gann 2001; Zimmerman and Martin 2001) provides knowledge for establishing a guideline for various building types, which is discussed in Section 1.3. However, it does not adequately incorporate project specificity into SUA, meaning that detailed properties of spaces and user activities are ignored, and therefore utilization is not tracked and updated when this information changes.

Operations research (Gajjar and Adil 2011; Ritzman et al. 1979) focuses on optimizing the decision variables such as space types and numbers rather than representing the space-use of current design options in support of architects' decision-making. This approach finds the best solution (values for decision variables) when there is a clear set of goal variables (e.g., minimizing the project cost) and constraints (e.g., the maximum number of a space type) that are numerically defined. However, since many factors affect architects' decision-making, and some of which, such as clients' intention and politics among different user groups, are difficult to define numerically, finding decision variables that optimize utilization is less meaningful. Rather, SUA must inform architects about the impact of their decisions on utilization so they can make sound decisions based on multiple performance metrics including utilization.

Computer reservation systems, also known as global distribution systems, are "computer systems that travel agents use to book airline seats, rental cars, hotel rooms, and other travel reservations and services" (Emmer et al. 1993). These systems are useful in the occupancy phase of a building when space utilization is between 0.5 and 0.75, which implies that activities may need to be scheduled, as shown in Table 1 in Section 1.4. These systems assist travel agents in finding space types (e.g., an aisle seat in an airplane or a non-smoking room with two beds in a hotel) that satisfy spatial requirements of users. However, they do not explicitly link users' activities to space types and therefore are not able to respond to changes in the activities automatically. A more systematic and explicit way of handling user activities in the programming phase of a building project would allow architects to keep track of user activities expected in the building and to determine whether or not these activities are supported by appropriate space types. In addition, when space utilization is less than 0.5, these systems are of little use because spaces are empty during most of the open hours. When space utilization is larger than 0.75, these systems are also of little use because users cannot find appropriate time slots for performing their activities.

Space planning tools, such as dRofus (http://www.drofus.no/) and Onuma Planning System (http://onuma.com/products/OnumaPlanningSystem.php/), help architects plan spaces based on design requirements and keep track of the information about spaces throughout the building projects. Onuma Planning System visualizes the utilization level of each space using color-codes when architects input the occupancy and the capacity data of the space. However, these tools do not predict and update utilization by taking into account user activities and spaces simultaneously.

2.2. Space utilization theory

Space utilization theory provides a measure for determining which spaces are rarely used or crowded. Space utilization has been recognized as an important factor in both construction space planning and building space planning. The construction phase is shorter than the occupancy phase, and, for planning, field workers on a construction site are deemed to be resources rather than autonomous agents. Therefore, space utilization in construction space planning is generally targeted at maximum utilization that does not violate constraints, and research in this field has often been directed at finding and visualizing conflicts between construction spaces and other resources (Akinci et al. 2002b; Chau et al. 2005; Guo 2002; Thabet and Beliveau 1994).

In terms of building space planning, Cherry (1999) introduces space utilization as a formula for predicting the number of spaces needed for classes in educational buildings. According to Cherry, 100% (or 1.0) utilization implies that the utilization is unacceptable to users due to scheduling inflexibility and long queues for activities in the space. In contrast, 0% (or 0.0) utilization implies that the utilization is unacceptable to clients due to building costs. Space utilization is similar to the manufacturing industry’s capacity utilization, which is “a ratio of the actual output to a sustainable maximum output, i.e., capacity” (Corrado and Mattey 1997). The biggest difference is that capacity utilization is targeted at the point where marginal costs are equal to average costs in manufacturing (Corrado and Mattey 1997). However, it is difficult to define and measure either costs in buildings. Therefore, Cherry (1999) emphasizes the need for a policy on utilization to use the utilization as a measure of the space-use.

Pennanen (2004) defines the utilization as a ratio of total activity loads in a space to open hours of the space to apply it to other types of buildings, such as office buildings and hospitals. Based on his work experience, he argues that if the utilization of a space is less than 0.50, activities can be conducted without waiting. If the utilization is larger than 0.50, activities may need to be scheduled. In addition, if the utilization is larger than 0.80, there seems to be a shortage of space (Pennanen 2004). Thus, space utilization in building space planning is targeted to meet a policy on utilization, and research in this field has often been directed at predicting space utilization and comparing it to the policy on utilization before the building is in its occupancy phase (Cherry 1999; Pennanen 2004). Based on these research efforts, I further defined what a certain range of utilization implies. (See the utilization implication levels shown in Table 1 in Section 1.4.)

2.3. Knowledge-based systems

The biggest advantage of a knowledge-based system in terms of SUA is that it can automatically infer new facts based on existing information and knowledge once they are represented in facts and rules (Akerkar and Sajja 2010; Brachman and Levesque 2004; Chinowsky 1991). Facts refer to “a set of raw observations, symbols, or statements” (Akerkar and Sajja 2010), represented in formal languages so that a knowledge-based system can interpret them for itself. Rules, also known as production rules, refer to “a two-part structure comprising an antecedent set of conditions and a consequent set of actions” (Brachman and Levesque 2004); the general form of rules is IF <conditions> AND <conditions> THEN <actions>. An inference engine of the system generally uses the following two rules to process project-specific information: *modus ponens*—“when A is known to be true, and a rule states “if A, then B,” it is valid to conclude that B is true”—and *modus tollens*—“when B is known to be false, and a rule states “if A, then B,” it is valid to conclude that A is false” (Akerkar and Sajja 2010).

This characteristic enables architects to predict space utilization consistently and quickly. Specifically, if a knowledge-based system has facts about activities, facts about spaces, and rules about mapping user activities onto spaces in its knowledge base, this system can generate facts about activity-space pairs consistently and quickly. If the system has rules about computing utilization in its knowledge base, it can then generate facts about space utilization consistently and quickly, given the facts about activities, spaces, and activity-space pairs (Figure 8).

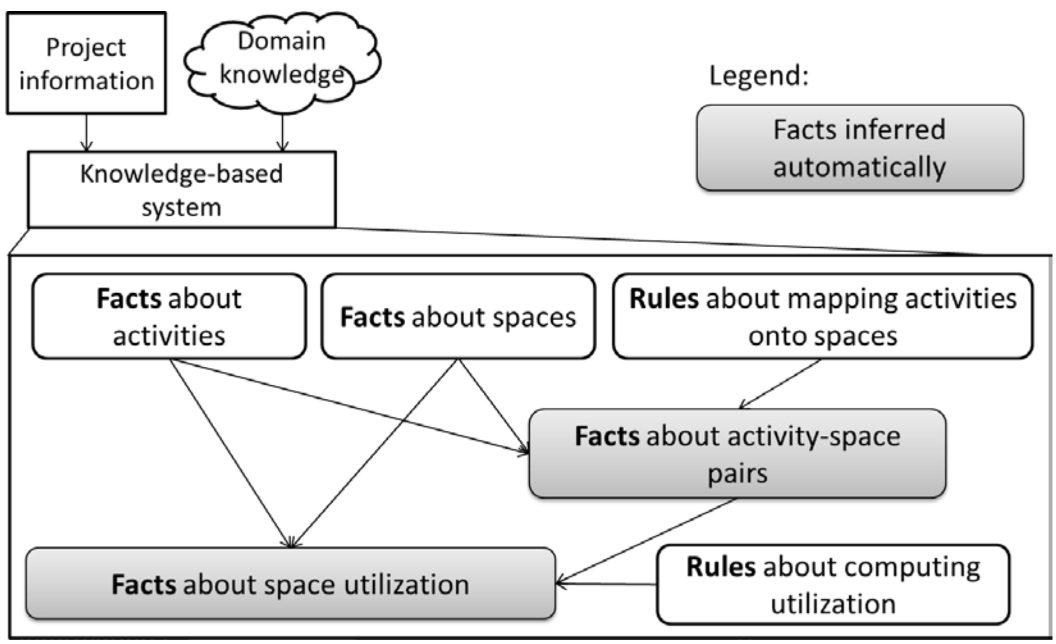

Figure 8. Application of a knowledge-based system for SUA: the knowledge-based system infers new facts based on information and knowledge that are represented in facts and rules.

Spatial computing for design is defined as “a body of work that is concerned with the use of formal methods in knowledge representation and reasoning in general, and terminological and spatial representation and reasoning in specific, for solving problems in modeling and validation in the domain of spatial design" (Bhatt and Freksa 2010). Based on the definition, Bhatt and Freksa (2010) propose an iterative refinement model in spatial design that is assisted by knowledge-based systems. According to their model, a spatial design should be abstracted and represented in a computer-interpretable form so that knowledge systems can reason about the design to produce valuable information that was originally barely noticed. The information is then used to provide design feedback and its visualization within the conventional design workflow. I utilize this model specifically for the development of an automated SUA that assists with the iterative refinement of the design according to space utilization. To do so, the spatial design and its representation should be integrated with user activity information because space utilization is affected not only by the spatial design itself (e.g., by increasing the number of spaces) but also by the intended usage of buildings (e.g., by limiting a user group's usage of a space).

2.4. Representation of space information

A space program is a document that lists the number and the area of each space type. Other features that need to be satisfied by a design, including adjacency, specific equipment, and location, can also be included in a text format under the name of "other," "etc.," or "note." Since the space program is in a tabular format, architects can relatively easily represent and capture it in the knowledge base of the KSUA system. For example, a simple relation schema (Ullman and Widom 2007) represents the space information as follows:

- Space program (space type, area, number, other features),

  where the underlined attribute represents the primary key to the relation (space program)

A Building Information Model (BIM) provides a computational representation of a design, and, consequently, space information that is stored in BIM as properties (e.g., space area, space type, open hours) can seamlessly feed into the KSUA method. BIM standards, such as General Services Administration (GSA)'s BIM Guide for Spatial Program Validation (GSA 2007) and Industry Foundation Classes (IFC) (http://www.buildingsmart.org/standards/ifc), specify how each space type and its properties should be represented for use in computer-assisted design process, and the KSUA method also needs similar specifications of space information that serve the purpose of the method. In addition, a declarative spatial reasoning framework (CLP(QS)) that allows qualitative spatial reasoning to interface with declarative programming languages (Bhatt et al. 2011) and a three-level formalization (conceptual, qualitative, and quantitative) for design artifacts and specification of design requirements based on the formalization (Bhatt et al. 2012) can allow architects to formalize space information as a high-level abstraction and can allow computers to reason about the space information on a quantitative level that is connected to BIM. However, for the sake of the simplicity, in this research, I formalize space information only as a high-level abstraction.

Some efforts have classified "other features" of spaces so a computer can recognize whether or not they are being satisfied by a design or an as-built model during project development (Building Smart Alliance 2013; Kiviniemi 2005).

However, I do not use these classifications and instead use the collective term “other features,” because these classifications do not help the development of the automated SUA method. In addition, it is easy to apply a set of classifications to the developed method once the classifications to be applied are chosen; the only thing to watch out for is that the same classifications must be used to describe user activities in user profiles and features of spaces in the space program.

2.5. Representation of user activity information

It is difficult to represent user activities and their spatial requirements and capture information about these in the knowledge base because user profiles usually describe this information in a text format. I investigated existing theories that represent construction activities or user activities for use in computer systems and found that an ontology that represents user activities for use in SUA is still missing.

Many researchers have represented construction activities for various purposes, such as planning (Aalami et al. 1998; Darwiche et al. 1988), time-space conflict analysis (Akinci et al. 2002a), cost estimation (Staub-french et al. 2003), and field instruction generation (Mourgues et al. 2012). Darwiche et al. (1988) represent activities as a tuple of <Objects>, <Actions>, and <Resources> on which other representations have been built to support different purposes (Table 2).

Table 2. Construction activity representations: the <OAR> tuple by Darwiche et al. (1988) provides a foundation for many ontologies for representing construction activities.

Author	Purpose	Representation	Example
Darwiche et al. (1988)	Representation of construction activity for planning	<Objects> <Actions> <Resources>	<Wall> <Paint> <Paint, Ladder, Painter>
Aalami et al. (1998)	Representation of construction method for automated planning	<Components> <Actions> <Resources> <Sequences>	<Concrete Block> <Cure> <none> <Pour>
Akinci et al. (2002a)	Representation of work spaces for time-space conflict analysis	<Components> <Actions> <Resources> <Spaces>	<Exterior Windows> <Place> <Crew W-1, Window panels> <Outside the windows requiring 2.5m width, 3m length and 2.5m height>
Staub-French et al. (2003)	Representation of construction activity for cost estimation	<Feature> <Objects> <Actions> <Resources>	<Wall-Bean Intersection> <Apply> <Fire Caulking> <Labor, Caulk>
Mourgues et al. (2012)	Representation of construction activities for generating field instructions	<Action> <Resource> <Object> <Work area>	<Pour> <Concrete> <Walls> <Garage, 4th floor>

I found two representations important in representing user activities in SUA. First, Darwiche et al. (1988) differentiate between the (construction) activity concept and the action concept. This differentiation is beneficial to SUA because architects can define an action and use the action to define multiple user activities. Second, Akinci et al. (2002a) define the “required work space” concept to specify spatial requirements for performing a construction activity in a computer-interpretable form. Since this concept is represented in a generic or project-independent way, a computer should find project-specific spaces that are required to perform the activity based on information about the activity and the design. The benefit of this approach is that spaces required by construction activities keep being updated whenever there are changes in information about these activities and the design, which is a good fit for SUA.

In terms of user activity representations, some researchers have modeled user behavior in buildings to simulate users' movement such as herding and separation in emergency (Pan et al. 2007) or in normal situations (Dijkstra and Timmermans 2002; Yan and Kalay 2006). However, these simulation models only partially represent user activities for use in SUA. Two research efforts have modeled user activities to predict space utilization. Tabak (2008) combines user and space information to simulate users' occupancy in a building. He divides activities into skeleton activities (i.e., activities that are formed in a sequence) and intermediate activities (i.e., physiological or social activities). Activity properties include frequency, duration, priority, location, and building. Pennanen (2004) models user activities to compute the utilization of each space automatically. The properties he considers include activity driver, load (i.e., hours that an activity demands from spaces), and group size. These models provide not only background knowledge of user activities in buildings, but also a set of properties of user activities that I can use in this research.

However, combined existing theories still do not provide an appropriate set of subclasses and properties of spatial requirements that are applicable to SUA. Akinci et al. (2002a) divide required work space (i.e., spatial requirements of construction activities) into four subclasses—hazard space, crew space, equipment space, and protected space—which cannot be directly used in representing spatial requirements of user activities. In addition, combined existing theories cannot capture all space-use types that need to be differentiated throughout the SUA. For example, these theories do not represent the case where preferred space and minimum spatial requirements of a user activity are different, and therefore architects have to keep track of this information and speculate on its impact on an ad-hoc basis.

2.6. Workplace planning

Workplace planning, one of the advanced theories in terms of using space utilization in building space planning, offers formalized knowledge about computing utilization given activity-space pairs (Pennanen 2004). Architects can compute utilization in the following steps:

- Step 1: compute *event quantity* for each activity, which refers to the number of groups for the activity; it is calculated by dividing the number of users by the size that the activity requires to have, i.e., group size.

  Event quantity = (the number of users × the ratio of the activity) ÷ the group size of the activity

- Step 2: compute *activity load* for each activity, which refers to hours that an activity demands from spaces.

  Load = event quantity of the activity × the frequency of the activity × the duration of the activity

- Step 3: compute *load per space* for each activity by dividing the activity load by the number of the space (type) that accommodates the activity.
- Step 4: compute *total loads* (or total activity loads) of a space by summing up all the loads per space of activities that occupy the space.

- Step 5: compute *utilization* of a space by dividing the total loads of the space by the open hours of the space.

Thus, I use workplace planning theory to formalize knowledge about computing utilization in KSUA.

2.7. User presence modeling

Many researchers have modeled user presence information, i.e., when or how long users are present in each space, for visualizing occupancy of building users (Akbas et al. 2007), analyzing space utilization (Tabak 2008; Pennanen 2004), or simulating energy performance (Bourgeois et al. 2006; Ioannidis et al. 2012). Diversity profiles represent user presence in a space in numbers between zero (0) and one. They are widely used (Bourgeois et al. 2006; Mahdavi and Pröglhöf 2009; Dong and Andrews 2009) because of their simplicity in representing user presence in a space. However, they do not include user activities and therefore do not allow architects to update user presence systematically according to changes in information about user activities.

To overcome the limitation of diversity profiles, many user presence models (Ioannidis et al. 2012; Pennanen 2004; Shen et al. 2012; Tabak 2008; Yan and Kalay 2006; Zimmermann 2007) represent both user activity information and space information and relate these two types of information explicitly. However, although some of the literature identifies certain space-use types that affect the generation of activity-space pairs, such as an activity that requires preferred space (Cherry 1999) or a designated space (Pennanen 2004), existing user presence models require architects to map user activities onto spaces manually because different space-use types and their impact on the mapping have not been formalized and integrated into a single method. For example, Tabak (2008) requires architects to name a facility and a space for each activity. Ioannidis et al. (2012) extend Tabak's work to include more organizational information, such as unit of enterprise (organization the user works for) and role of actor (user), but they still require architects to fill the property "activityHasSpaceType," which has space gbXML as its range. To support the development of KSUA, knowledge about finding appropriate spaces for an activity and filling the property of the activity that describes where this activity will occur must be formalized and represented in rules.

2.8. Conclusions from points of departure

Prior work concerning SUA includes architectural programming, post-occupancy evaluation, operations research for space assignment, computer reservation systems, and space planning systems. Although the importance of SUA has been recognized widely, previous research efforts provide only limited knowledge for automating SUA (Section 2.1). Space utilization theory provides a measure for determining which spaces are rarely used or crowded, and it also provides knowledge about what a certain range of utilization implies (Section 2.2). Based on this background theory, I found that a knowledge-based system is useful to automate SUA because it provides an inference engine that automatically reasons about project-specific information about user activities and spaces, as well as domain knowledge about SUA (Section 2.3).

In terms of representing project-specific information, the space program is in a tabular format, and architects can relatively easily represent and capture it in the knowledge base of a knowledge-based system (Section 2.4). However, it is difficult to represent user activities and their spatial requirements and capture information about these in the knowledge base (Section 2.5) because of the following theoretical gap (Gap 1):

Gap 1: existing user activity and construction activity representation models do not distinguish between different space-use types that affect SUA. Therefore, architects have to keep track of different space-use types and speculate on the impact of these types on an ad-hoc basis.

In terms of formalizing domain knowledge about SUA, given activity-space pairs, workplace planning offers formalized knowledge about computing utilization (Section 2.6). However, current user presence models do not include the knowledge about mapping user activities onto appropriate spaces (Section 2.7) because of the following theoretical gap (Gap 2):

Gap 2: different space-use types and their impact on the mapping have not been formalized and integrated into a single method. Therefore, architects have to map user activities manually onto appropriate spaces and update the mapping throughout the SUA.

These two gaps must be addressed to enable the KSUA method (Figure 9).

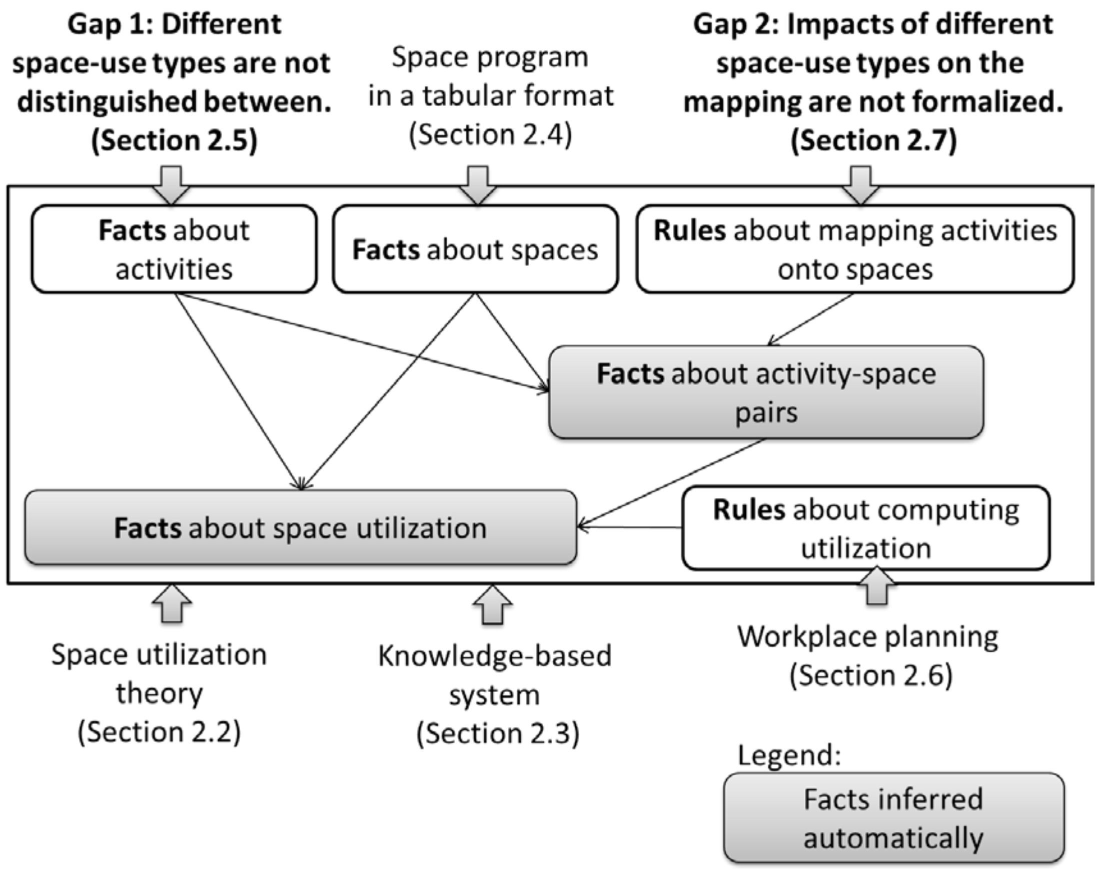

Figure 9. Summary of points of departure: I found two theoretical gaps that hindered the development of automated space-use analysis based on a knowledge-based systems approach.

3. Space-use types that affect space-use analysis

Since both research questions (Section 1.7) are closely related to space-use types that affect SUA, it is important to define space-use types that need to be taken into account in the KSUA method. This chapter describes how I derived 288 space-use types. Without computational assistance, architects have to keep track of the space-use type information of each user activity and speculate on the impact of this information when mapping user activities onto spaces and computing utilization throughout the SUA, which is often impractical.

3.1. Characteristics of user activities

The characteristics of user activities describe the contexts that inform how space-use types need to be treated during SUA. I first identified five characteristics of user activities in a descriptive format based on literature review, case observations, and interviews with architects. I then surveyed 14 subject matter experts working in three construction companies in the USA who have more than four years of experience in SUA (Appendix B). Their average experience in the construction industry was 11.0 years; their average experience in SUA was 8.5 years. I asked them to indicate their level of agreement on each of the characteristics as follows—1: strongly disagree, 2: disagree, 3: neither agree nor disagree, 4: agree, 5: strongly agree—to measure acceptability. I also asked them to name other characteristics that must be taken into account when analyzing space-use. As a result, all of the following characteristics (“C”) were accepted because they scored more than 4.0 on a Likert scale of one to five. No additional characteristics other than the five were identified by the survey.

- C1: Some users have more stringent spatial requirements for their activities than the minimum requirements support (Cherry 1999). This characteristic scored 4.0 over 5.0. One expert stated in the survey that this is closely related to the organizational culture: an organization with strong hierarchical structure requires allocation of bigger and quieter space for people with important responsibilities, while an organization with flat structure provides every employee equal space regardless of the importance of the employee's responsibilities.
- C2: Some activities require a designated space, e.g., a professor's office (Pennanen 2004). This characteristic scored 4.5 over 5.0. If a space is designated, the space cannot be used by other users even if the space is vacant.
- C3: Some activities require occupying the whole room (e.g., a meeting activity requires a whole conference room), while others need part of a room (e.g., regular work of an employee requires occupation of only one workstation in the office area). This characteristic scored 4.5 over 5.0.
- C4: Some activities are conducted in a specifically named space while others are conducted in any space providing certain features. Unlike a designated space, a specifically named space can be used by many users when the space is vacant. This characteristic scored 4.5 over 5.0. Experts named many features of spaces that may be required by an activity, including multimedia settings, ventilation, ability to control lighting, indirect light, and location.
- C5: Some activities are atypical activities, which are not conducted on a regular basis, e.g., commencement. This characteristic scored 4.6 over 5.0.

Atypical activities are not taken into account in utilization computation, although they are predictable events and must be accommodated by the design (Cherry 1999).

3.2. Space-use types

Based on the identified characteristics of user activities in Section 3.1, I defined six space-use type differentiators (SUTDs); a type of a user activity can be explained as a set of choices for each SUTD. There are three SUTDs that a user activity must go through:

- SUTD 1 differentiates between *atypical activities* and *typical activities*. Atypical activities are not conducted on a regular basis and therefore do not affect the computation of utilization. In contrast, typical activities affect both mapping of user activities onto spaces and computing utilization.
- SUTD 2 differentiates between *important users* and *regular users*. If users of an activity are deemed to be important, architects must provide them with a preferred space, i.e., space that satisfies more stringent requirements. If users are regular, architects can provide them with space with the minimum requirements.
- SUTD 3 differentiates between *the situation where the space preferred for performing an activity is not identified* and th*e situation where the preferred space and the minimum requirements are different*. In this research, “preferences” refer to the preferred space, and “constraints” refer to the minimum requirements.

The following three SUTDs are applied to either constraints or preferences:

- SUTD 4 differentiates between *the situation where an activity requires a designated space* and *the situation where an activity does not require a designated space*.
- SUTD 5 differentiates between *the situation where an activity requires occupying a whole room* and *the situation where an activity requires occupying part of a room*. In this research, this "part of a room" is referred to as equipment. Both the whole room and the equipment are space that users can occupy.
- SUTD 6 differentiates between *the situation where an activity specifies the type of a space it needs* and *the situation where an activity specifies certain features that need to be satisfied by a space*.

In summary, a space-use type of an activity can be explained as a set of choices at each SUTD. Since three SUTDs (4-6) are attached to constraints or preferences, not directly to user activities, the number of space-use types is 32 ($=2^2 \times 2^3$) for the situation where preferences are not identified and therefore are equal to constraints, and is 256 ($=2^2 \times 2^3 \times 2^3$) for the situation where preferences differ from constraints. All of these 288 (=32 + 256) space-use types must be distinguished and treated differently by the proposed KSUA method (Figure 10).

These SUTDs bring together the space, user, and activity perspectives by supporting architects to decide on the space-use type of an activity taking into account all of the three perspectives. For example, changes in atypical activities affect the user and activity perspectives but do not affect the space perspective because they are not used in utilization computation. From the important users' perspective, spaces must satisfy their preferences to accommodate their activities. From the regular users' perspective, spaces must satisfy their constraints to accommodate their activities. Changes in space-use type of an activity by architects can affect the user and space perspectives by changing information about where the activity will occur in the occupancy phase.

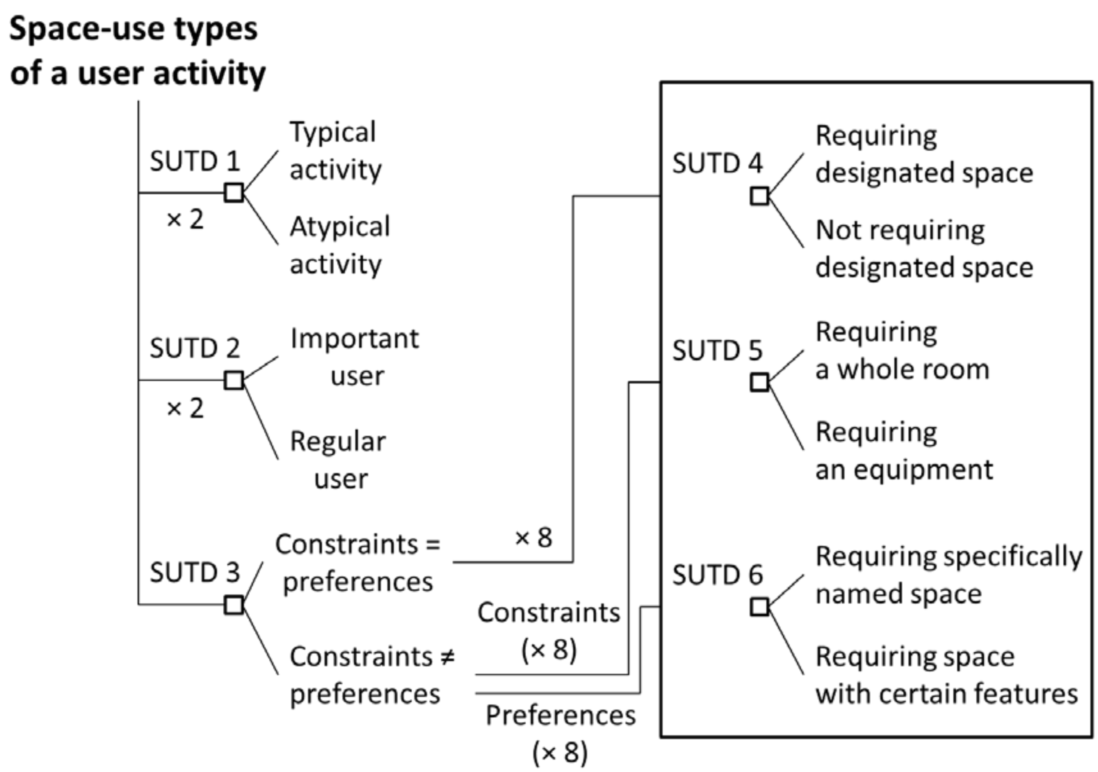

Figure 10. Space-use types: Each user activity must be characterized with SUTD 1 to 3, and either constraints or preferences must be categorized with SUTD 4 to 6. There are 288 space-use types that architects must distinguish and treat differently during space-use analysis.

4. An ontology for representing user activities in space-use analysis

This chapter describes the ontology I developed to provide a common vocabulary for architects to represent the information about user activities. Such an ontology enables architects to gather, store, and reuse user activity information, and to feed it systematically and seamlessly to a computer system that conducts SUA.

According to the ontology development methodology suggested by Noy and McGuinness (2001), I defined 10 classes and their properties that are necessary for distinguishing between different space-use types, which are described in Chapter 3. I further defined facets of each property to describe the number of the values the property can have (i.e., cardinality), type of values it can have (i.e., value type), and allowed values for the value type (i.e., range). I conducted three case studies to determine whether or not the developed ontology can represent user activities for use in KSUA. Two of the three were educational buildings and one was an office building. As a result, 100% of 28 user activities in three cases were successfully represented using the developed ontology. In addition, the ontology is comprehensive enough to represent all of the 288 space-use types because it has been developed to capture all of the six SUTDs.

4.1. Concepts for space-use analysis

Based on the space-use types described in Chapter 3 and the combination of existing theories that represent user activities or construction activities described in Section 2.5, I defined the concepts for SUA and the ontological relationships among the concepts.

The concepts include spaces, equipment, users, user activities, actions, and spatial requirements.

- *Space* is defined as a physical entity that accommodates a user activity, e.g., a conference room. *Equipment* is another physical entity that accommodates user activities (e.g., a workstation, a computer), but it represents part of a room while *a space* represents a whole room. A *space* can have multiple pieces of *equipment*, in which case the whole space is not occupied by activities. Therefore, spaces are grouped into two subclasses: *occupiable space* that has no equipment and is occupied by activities and *non-occupiable space* that is not occupied and has pieces of equipment that are occupied by activities. *Occupiable spaces* and *equipment* can be designated by user activities while *non-occupiable spaces* are not allowed to be designated.
- *User* is defined as a subject of a user activity, e.g., students, employees. In SUA, *user* and *user group* are interchangeable because SUA does not consider individual users and their personal needs, e.g., Tom works well with Jane, so he wants to study near her.
- *User activity* is defined as an *action* of users that requires occupying spaces. Therefore, a user activity is defined not only by its action, but also by users and requirements of the activity.
- *Spatial requirements* are defined as properties of a space that an activity requires for occupying the space.

The ontological relationships I defined in Figure 11 answer the questions of how to describe user activities, how to relate user activities to spaces or equipment, and how to compute space utilization based on the product (space and equipment) and the organization (user, user activity, action, spatial requirements) information.

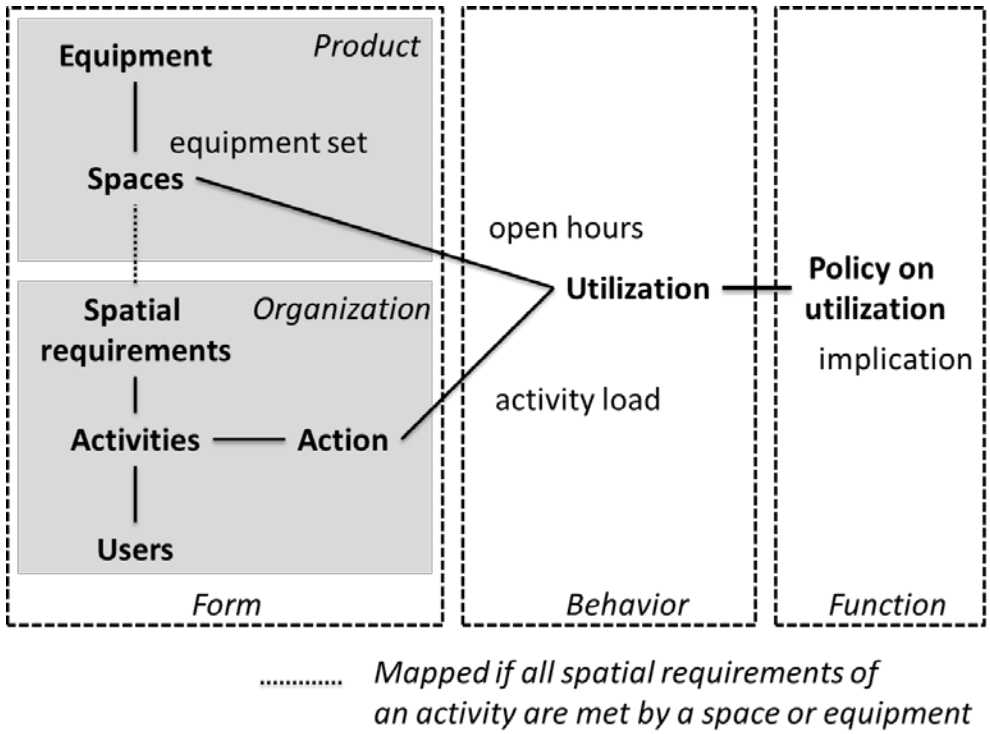

Figure 11. The ontological relationships among the concepts for space-use analysis: architects can represent user activities as a <User><Action><Spatial requirements> tuple, and a user activity is mapped onto spaces that satisfy all spatial requirements of the activity.

I suggest a tuple of <User>, <Action>, and <Spatial requirements> (i.e., <UAS> tuple) as a representation of user activities for automated SUA. Examples of user activities from one case study I analyzed are (1) <Employees><Have a meeting><In a meeting room that is larger than 15m^2>, (2) <Editors><Edit a book><In any room with quiet conditions>, and (3) <A company president><Paints as her hobby><In an art room>. User activities are accommodated by spaces or equipment that satisfies spatial requirements of the activities. The concept of *equipment set* is defined to differentiate the same type of equipment in different spaces. Space utilization is computed in light of the activity-space pairs based on workplace planning (Section 2.6) and compared with the policy on utilization to determine whether the space will be rarely used, properly used, or crowded (Section 1.4).

4.2. An ontology for representing user activities

This section describes an ontology I developed to represent user activities for use in KSUA. Figure 12 depicts the ontology using a Unified Modeling Language (UML) class diagram, widely used in documenting multiple concepts and their relationships (Clements et al. 2011). I divided concepts into subclasses (i.e., specialization) or introduced properties to accommodate the six SUTDs established in Section 3.2. I also defined other properties that are necessary for mapping user activities onto spaces or computing utilization.

I used frame logic, or F-Logic (Angele et al. 2009), one of the knowledge representation and reasoning languages, to define the ontology in a computer-interpretable form (Section 4.2) and to test it in three case studies (Section 4.3). F-Logic provides the following basic syntax to define an ontology and create instances (i.e., project-specific information about user activities) based on the ontology:

- Specialization:

  subclass of concept :: concept.

- Introduction of properties:

  concept [property {cardinality} *=> value type or other concept].

- Instantiation:

instance name : concept [property -> value or other instance].

Unless stated otherwise, the default cardinality of a property is {0:*}, which means that the property can have any number of values or instances (Angele et al. 2009).

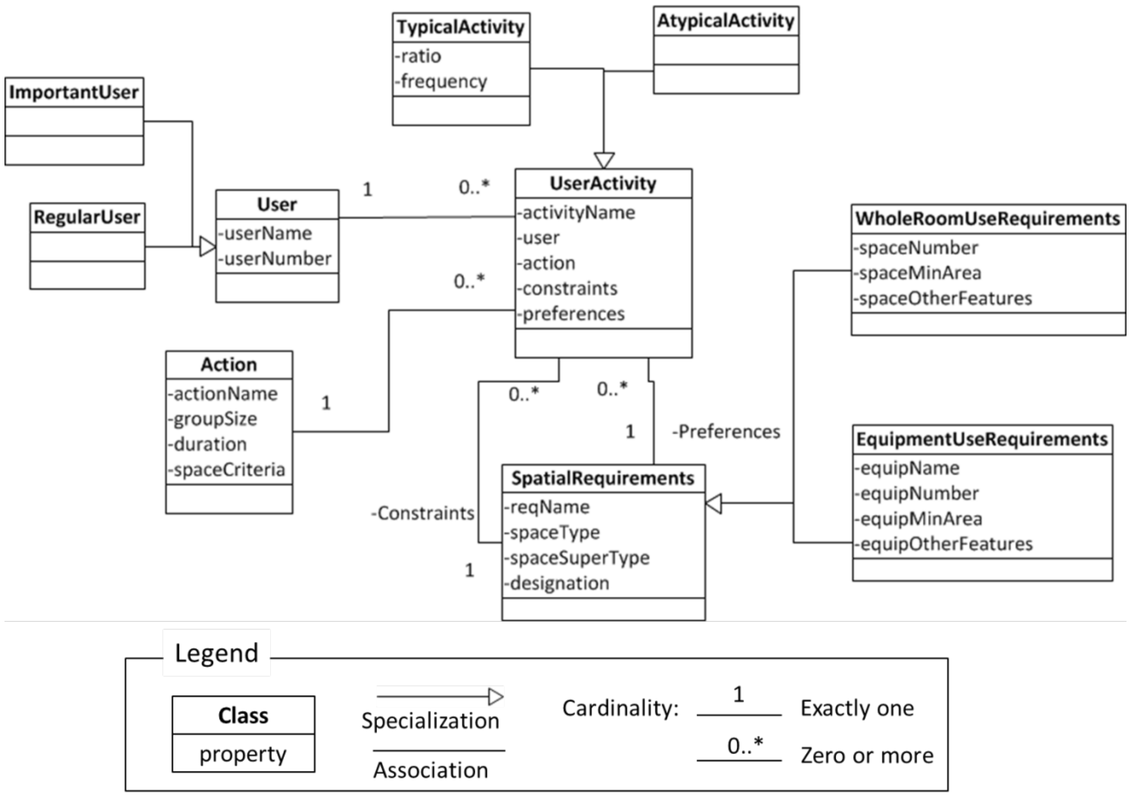

Figure 12. An ontology for representing user activities in SUA: architects can represent a user activity as <User>, <Action>, and two <Spatial requirements>, i.e., constraints and preferences.

4.2.1. User activity

A user activity is represented by a tuple of <User>, <Action>, and two <Spatial requirements> instances, i.e., preferences and constraints. There are two subclasses of user activities regarding SUA: (1) <Typical activity>, which is conducted on a regular basis and therefore needs to be taken into account in computing utilization, and (2) <Atypical activity>, which does not affect space utilization but needs to be included in

SUA so that an architect can ensure that the design also accommodates this activity. Therefore, typical user activities have additional properties that enable KSUA to compute and evaluate utilization. Table 1 shows the properties and their facets that this ontology formalizes to represent a user activity. Common properties of <User activity> are

- *Activity name*: The name of a user activity, e.g., student meeting, class activity. Each name must be unique to hold integrity constraints (Ullman and Widom 2007).
- *User*: The main subject (or driver) of a user activity. This property must have exactly one instance of the <User> concept, which is described in detail in Section 4.2.2.
- *Action*: The description of what is being performed by a user activity. This property must have exactly one instance of the <Action> concept, which is described in detail in Section 4.2.3.
- *Constraints*: The minimum spatial requirements of an activity. An activity by "regular users" is conducted in the spaces that satisfy all constraints of the activity. This property must have exactly one instance of either the <Whole room use requirements> or the <Equipment use requirements> concept, which is a subclass of the <Spatial requirements> concept. Detailed information about spatial requirements is described in Section 4.2.4.
- *Preferences*: Spatial requirements for better performance of an activity. An activity by "important users" is conducted in the spaces that satisfy all the preferences of the activity. Like the *constraints* property, this property must

have exactly one instance of either the <Whole room use requirements> or the <Equipment use requirements> concept, which is a subclass of the <Spatial requirements> concept.

The properties of the <Typical activity> concept are

- *Ratio*: The property that represents what percentage of users is involved in an activity. This property has a value between 0 (none of the users are involved) and 1 (all of the users are involved). Since this property is related to utilization computation, only the <Typical activity> concept contains this property.
- *Frequency*: The property that represents how many times a user is involved in an activity per day. This property is also only for the <Typical activity> concept, and it has a positive float-type value.

Table 3. Facets of properties of the <User activity> concept.

Property	Cardinality	Value type	Range
Common properties			
Activity name	{1:1}	String	N/A
User	{1:1}	Instance	User
Action	{1:1}	Instance	Action
Constraints	{1:1}	Instance	Spatial requirements
Preferences	{1:1}	Instance	Spatial requirements
Properties of typical activity			
Ratio	{1:1}	Float	0 to 1
Frequency	{1:1}	Float	Positive

Given these definitions, I developed the following ontology for representing user activities in F-Logic:

Event :: Thing.

Activity :: Event [user {1:1} *=> User, action {1:1} *=> Action, constraints {1:1} *=> SpatialReq, preferences {1:1} *=> SpatialReq].

TypicalActivity :: Activity [ratio *=> float, frequency *=> float].

AtypicalActivity :: Activity.

4.2.2. User

User is the subject of a user activity. In this research, user and user group are interchangeable because KSUA does not consider individual users and their personal needs. This concept has two subclasses: (1) <Regular user>, which requires satisfying the *constraints* of the users' activities, and (2) <Important user>, which requires satisfying the *preferences* of the users' activities. Table 4 shows the properties and their facets that this ontology formalizes to represent a user group. These properties are

- *User name*: The name of a user group, e.g., student, employee. Each name must be unique to hold integrity constraints (Ullman and Widom 2007).
- *User number*: The number of a user group. This property has exactly one positive integer as its value.

Table 4. Facets of properties of the <User> concept.

Property	Cardinality	Value type	Range
Common properties			
User name	{1:1}	String	N/A
User number	{1:1}	Integer	Positive

Given these definitions, I developed the following ontology for representing users in F-Logic:

Entity :: Thing.

User :: Entity [number {1:1} *=> integer].

ImportantUser :: User.

RegularUser :: User.

4.2.3. Action

Action describes what is being performed by a user activity. I assume that a user activity has only one action and has no workflow because it is difficult to represent all user activities as workflows. Action has the following properties that are needed in KSUA (Table 5):

- *Action name*: The name of an action, e.g., meeting, experiment. Each name must be unique to hold integrity constraints (Ullman and Widom 2007).
- *Group size*: The property that represents the number of users that an action requires to have per occurrence. This property has a positive integer-type value.
- *Duration*: The property that represents how long an action continues per occurrence. This property has a positive float-type value.
- *Space criteria*: The property that represents how much space an action takes per user, e.g., 17 ft^2 (1.6 m^2) per student for a small class (Stanford University 2009). This property is used in KSUA to compute space-use area and develop an activity-loaded space diagram, which visualizes the space-use of a room by multiple activities (Chapter 6).

Table 5. Facets of properties of the <Action> concept.

Property	Cardinality	Value type	Range
Action name	{1:1}	String	N/A
Group size	{1:1}	Integer	Positive
Duration	{1:1}	Float	Positive
Space criteria	{1:1}	Float	Positive

Given these definitions, I developed the following ontology for representing actions in F-Logic:

Action :: Event [groupSize {1:1} *=> integer, duration {1:1} *=> float, spaceCriteria {1:1} *=> float].

4.2.4. Spatial requirements

Spatial requirements are conditions of a space that a user activity requires for occupying the space. Since some activities require occupying a whole room, while others need only part of a room, spatial requirements have the following two subclasses: (1) the <Whole room use requirements> class, which characterizes the required conditions of a whole room, and (2) the <Equipment use requirements> class, which characterizes the required conditions of part of a room, which I call "equipment" in this ontology. Thus, all instantiation of spatial requirements must take one of these forms to be used in KSUA. Table 6 shows the properties and their facets that this ontology formalizes to represent spatial requirements. The common properties of <Spatial requirements> are

- *Requirements name*: The name of a set of requirements. Since an instance of the <Spatial requirements> concept contains multiple required conditions, such as space type and space size, this name is usually meaningless and often takes a form of identification numbers, e.g., constraint1, constraint2.
- *Space type*: The specific type of a space that an activity occupies, e.g., large conference room, small conference room, and class room. This property is

introduced to represent activities that are conducted in a specifically named space, and it has zero or more instances of the <Space type> concept.[3]

- *Space super type*: The property that represents the required super type of a space, e.g., conference room. This property has zero or more instances of the <Space super type> concept, which is assigned to each space.
- *Designation*: This property represents whether or not an activity requires occupying a designated space. This property must have either "True" or "False" as its value.

The properties of the <Whole room use requirements> concept are

- *Space number*: The number of the space type that an activity requires occupying. This property has zero or one positive integer-type values.
- *Space minimum area*: The minimum area of a space that an activity requires. If not specified, this property has the value of the space-use area, which is computed by multiplying the values of *group size* and *space criteria* of <Action> of an activity.
- *Space other features*: This property represents required features of a space, e.g., multimedia settings, indirect light, and location. This property has zero or more instances of the <Features> concept. The <Features> concept can use the design requirements interface developed by Kiviniemi (2005) to automatically compare required space features, which are represented by this ontology, with

---

[3] Kiviniemi (2005) calls space type "space program instance (SPI)" to distinguish this type, which is used in a requirements model, from space instance (e.g., meeting room A204, A205), which is used in a design model.

features fulfilled by the design, which are represented in a BIM. However, I do not describe the use of the interface in further detail and focus instead on the ontology for representing user activities for use in KSUA.

The properties of the <Equipment use requirements> concept are

- *Equipment name*: The specific name of a piece of equipment that an activity occupies, e.g., a workstation, a computer. This property has zero or more instances of the <Equipment> concept. Since the <Equipment user requirements> concept is a specialization of the <Spatial requirements> concept, an instance of <Equipment use requirements> can have both *space type/super type* and *equipment name* at the same time, e.g., a computer in a conference room-type space.
- *Equipment number*: The number of equipment instances that an activity requires. This property has zero or one positive integer-type values.
- *Equipment minimum area*: The minimum area of an equipment instance that an activity requires occupying. If not specified, this property has the value of the space-use area, which is computed by multiplying the values of *group size* and *space criteria* of <Action> of an activity.
- *Equipment other features*: This property represents required features of an equipment, e.g., a specification of a computer. This property has zero or more instances of the <Features> concept.

Table 6. Facets of properties of the <Spatial requirements> concept.

Property	Cardinality	Value type	Range
Common properties			
Requirements name	{1:1}	String	N/A
Space type	{0:*}	Instance	Space type
Space super type	{0:*}	Instance	Space super type
Designation	{1:1}	Boolean	True, False
Properties of whole room use requirements			
Space number	{0:1}	Integer	Positive
Space minimum area	{0:1}	Float	Positive
Space other features	{0:*}	Instance	Features
Properties of equipment use requirements			
Equipment name	{0:*}	Instance	Equipment
Equipment number	{0:1}	Integer	Positive
Equipment minimum area	{0:1}	Float	Positive
Equipment other features	{0:*}	Instance	Features

Given these definitions, I developed the following ontology for representing actions in F-Logic:

SpatialReq :: Entity [spaceType *=> SpaceType, superType *=> SpaceSuperType, designation {1:1} *=> boolean].

WholeRoomUseReq :: SpatialReq [number {0:1} *=> integer, minArea {0:1} *=> float, otherFeatures *=> Features].

EquipmentUseReq :: SpatialReq [equipment *=> Equipment, number {0:1} *=> integer, minArea {0:1} *=> float, otherFeatures *=> Features].

4.3. Validation of the developed ontology

The ontology must be formal and comprehensive for use in SUA, as described in Section 1.7. Note that my goal was not to create the best ontology but to create an ontology that is formal enough to represent user activities for use in SUA without ad hoc input and comprehensive enough to distinguish all space-use types identified in Chapter 3. Therefore, I tested the ontology I described in Section 4.2 to determine whether or not it is formal and comprehensive.

4.3.1. Formality of the developed ontology

To validate the formality, I conducted three case studies in which I identified user activities and attempted to represent them using the ontology. The three cases are the Jerry Yang and Akiko Yamazaki Environmental and Energy (Y2E2) Building located at Stanford University, United States of America (Appendix D1), Cygnaeus High School located in Jyväskylä, Finland (Appendix D2), and X Construction Company located in Seoul, South Korea[4] (Appendix D3). For the first case, I gathered and defined user and space information in selected areas of the Y2E2 Building based on observation, hourly measurements of space-use, interviews with users, the BIM of the building, and Stanford University's Space and Furniture Planning Guide (2009). The number of students was scaled down proportionate to their enrollment in the Department of Civil and Environmental Engineering, which is located in and uses the building. I represented 13 user activities using the proposed ontology. The second case is described by Pennanen (2004). It is the Cygnaeus High School renovation project in Finland, which he uses to demonstrate the effectiveness of workplace planning. This case is also described in the dissertation of Whelton (2004). Based on these studies, I identified and represented five user activities using the developed ontology. For the

[4] The name of the company cannot be mentioned here because of confidentiality constraints.

third case, I examined the design of the office for X construction management company in Korea and interviewed three employees of the company to identify user activities. For small and large meeting rooms, office area, rest area, and restrooms, I identified and represented 10 user activities of four different user groups. Table 7 shows the number of user groups and the number of user activities for each case.

Table 7. Three case studies for validating the developed ontology: I identified 28 user activities in three cases and represented them using the developed ontology.

	Y2E2	Cygnaeus	X Construction	Total
The number of user groups	5	4	4	13
The number of user activities	13	5	10	28

As a result, 100% of 28 user activities in three cases were successfully represented using the developed ontology, showing the formality of the proposed ontology. I could classify these 28 user activities with 11 of the 288 space-use types I identified in Section 3.2. Two activities were atypical activities. Constraints and preferences were the same in six activities. Four activities needed a designated space. Here are some examples of user activities:

- In the Y2E2 Building, 70% of 122 graduate students (users) take classes once a week, and each class has 20 students and lasts two hours per occurrence on average (action). This activity requires a space that is at least 37 m^2, and a screen for a computer is preferred (spatial requirements). The representation is as follows:

  gradsHavingClass:TypicalActivity [user -> grads, action -> haveClass, ratio -> 0.7, frequency -> 0.2, constraints -> cons1, preferences -> pref1].

grads:RegularUser [number -> 122].

haveClass:Action [groupSize -> 20, duration -> 2.0, spaceCriteria -> 1.7].

cons1:WholeRoomUseReq [minArea -> 37, designation -> False].

pref1:WholeRoomUseReq [minArea -> 37, otherFeatures -> screenProvided, designation -> False].

- In Cygnaeus High School, 70 teachers (users) perform regular work (action) at a workstation in the teachers' office. Teachers have their own workstations for this activity. The size of the workstation needs to be larger than 3 m^2, preferably larger than 4.5 m^2 (spatial requirements). The representation is as follows:

teacherWorking:TypicalActivity [user -> teachers, action -> doRegularWork, ratio -> 1.0, frequency -> 1.0, constraints -> cons2, preferences -> pref2].

teachers:RegularUser [number -> 70].

doRegularWork:Action [groupSize -> 1, duration -> 1.45, spaceCriteria -> 2.7].

cons2:EquipmentUseReq [spaceType -> teacherOffice, equipment -> workstation, minArea -> 3.0, designation -> True].

pref2:EquipmentUseReq [spaceType -> teacherOffice, equipment -> workstation, minArea -> 4.5, designation -> True].

4.3.2. Comprehensiveness of the developed ontology

The proposed ontology is comprehensive enough to distinguish all of the 288 space-use types identified in Section 3.2. This can be proven by showing that the proposed ontology can represent the six SUTDs, from which the 288 space-use types are derived. The developed ontology is an extension of the combined existing theories

(Akinci et al. 2002a; Darwiche et al. 1988; Pennanen 2004). Based on these theories, to capture SUTD 1, I divided the <User activity> concept into <Typical activity> and <Atypical activity>. To capture SUTD 2, I divided the <User> concept into <Important user> and <Regular user>. To capture SUTD 3, I gave the <User activity> concept properties of *constraints* and *preferences*, which have an instance of the <Spatial requirements> concept as their values. To capture SUTD 4, I added the *designation* property to the <Spatial requirements> concept. To capture SUTD 5, I divided the <Spatial requirements> concept into <Whole room use requirements> and <Equipment use requirements>. Finally, to capture SUTD 6, I added the *space super type* property to the <Spatial requirements> concept and added the properties representing features of space (e.g., *area and number*) to <Whole room use requirements> and <Equipment use requirements>. Note that the *space super type* property was added to the <Spatial requirements> concept, not to the <Whole room use requirements> concept, because a certain user activity requires occupying a piece of equipment in the specific super type of a space.

4.4. Conclusions about the developed ontology for representing user activities in space-use analysis

I proposed an ontology for representing user activities for use in SUA, which consists of 10 classes (four concepts and their subclasses) and 24 properties and their facets. Based on this ontology, architects can represent a user activity in a <User>, an <Action>, and two <Spatial requirements> instances. This ontology is novel because it allows a computer to distinguish between different space-use types that affect SUA, whereas existing theories of representing user or construction activities (Akinci et al. 2002a; Darwiche et al. 1988; Pennanen 2004) require architects to keep track manually of the space-use type of each activity during the analysis. I validated the proposed ontology by representing 28 user activities identified in three case studies and by determining whether or not it can capture all SUTDs, from which the 288 space-use types are derived.

This work currently has the following limitations and assumptions:

- Since the research focuses on supporting the programming phase of a project, space location, geometry, and aesthetic factors are not captured in this ontology. For use of this method in the design phase, these factors must also be captured in an ontology.
- This ontology does not capture stochastic features (e.g., fluctuating numbers of users) of user activities.
- This ontology does not take into account individual users (e.g., start and end time of an activity of an individual user, and individual requirements regarding selecting space for performing his or her activities).
- One user activity is assumed to have only one action, and there is no sequence between two user activities.
- There are only two importance levels for a user group: important and regular.

To achieve the full potential of automated SUA, these limitations and assumptions will need to be tested and addressed. My ontology can be extended to capture more space-use types and properties in the future. For example, by extending my ontology, architects can represent a user activity to have multiple actions or to have more than two importance levels for a user group. However, these extensions must be made in tandem with the formalization of how the newly captured space-use types and properties affect space-use and its analysis.

5. A method for mapping user activities onto appropriate spaces

This chapter describes a method I formalized to enable a computer to map user activities onto appropriate spaces and to eliminate the human effort of speculating on all possible activity-space pairs during SUA. Appropriate spaces refer to space entities that accommodate a user activity of interest in the occupancy of a building. This method takes space information in the space program and user activity information represented using the ontology described in Chapter 4 as input and yields the activity-space pairs as output.

Figure 13 depicts the three steps of the activity-space mapping method each user activity must take. First, the “choose spatial requirements” step determines the requirements to apply in the subsequent steps from constraints and preferences. Second, the “find spaces” step finds all spaces (whole rooms or equipment) that satisfy chosen spatial requirements of the activity. Third, the “map the activity onto spaces” step generates activity-space pairs for use in computing utilization.

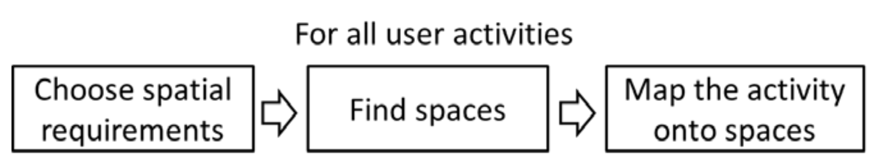

Figure 13. Three steps of the activity-space mapping method: each user activity represented using the ontology described in Chapter 4 must go through these steps.

Below, I elaborate each step of the method using SUTDs defined in Section 3.2. This elaboration is in the form of “IF <conditions> AND <conditions> THEN <actions>” to be used as rules in the KSUA system (Section 5.1 through 5.3). I then validated the developed mapping method by letting five architects with an average of 15.8 years of experience conduct the mapping given information about user activities and spaces. I measured the conformity of the activity-space pairs generated by the method with the pairs generated by the architects. I also explained the method I used to architects and asked them to score the acceptability of the method on a Likert scale of 1 to 5 to avoid the case of high conformity achieved by random chance (Section 5.4). The conformity was 93% over two cases, and the acceptability was 5.0.

5.1. Step 1: choose spatial requirements

In the first step, a computer checks the choices for two SUTDs (SUTD 2 and 3) and determines which requirements to apply in the subsequent steps (Figure 14).

- IF preferences are not identified or are the same as constraints,

  THEN the spatial requirements to apply are constraints regardless of whether or not users are important.

- IF preferences differ from constraints,

  AND users are important,

  THEN the spatial requirements to apply are preferences.

- IF preferences differ from constraints,

  AND users are regular,

  THEN the spatial requirements to apply are constraints.

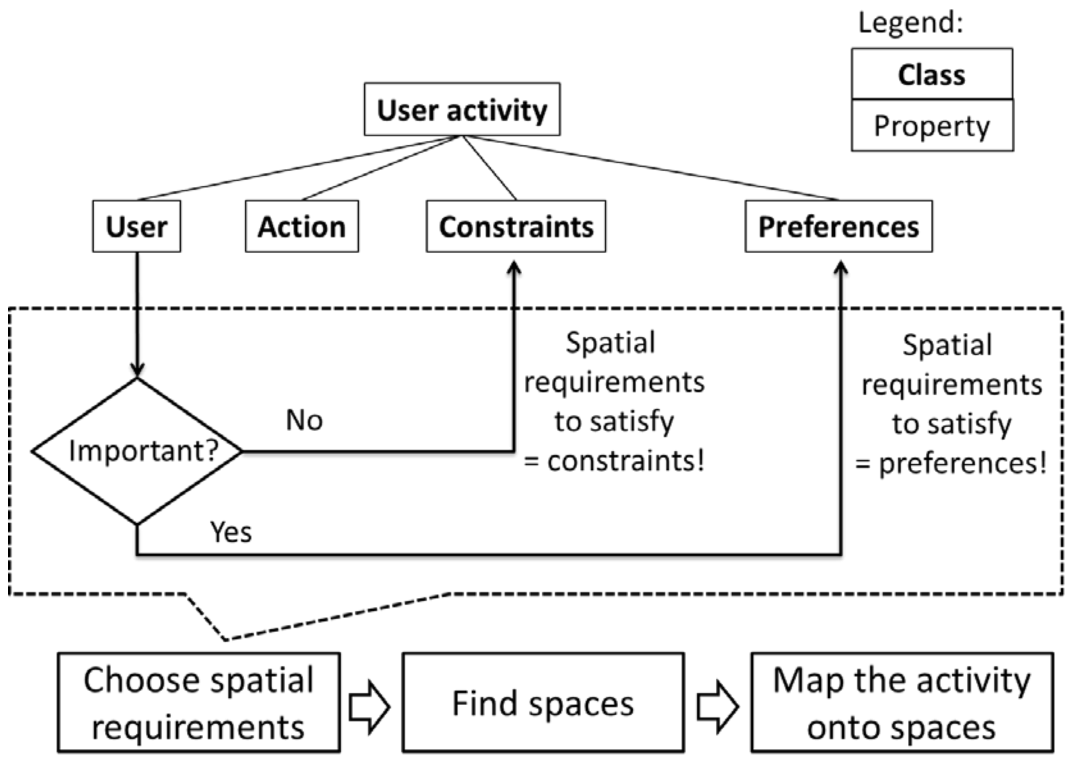

Figure 14. The “choose spatial requirements” step of the activity-space mapping method: if constraints and preferences are different, a computer must choose spatial requirements to satisfy based on the specialization of the <User> concept of the activity.

5.2. Step 2: find spaces

In the second step, the computer checks the choices for two SUTDs (SUTD 5 and 6) and determines which space in the space program satisfies the spatial requirements of the activity (Figure 15).

- IF an activity requires occupying a whole room,

  AND the activity specifies a space type it needs,

  THEN the computer finds all spaces that are not designated and whose type is the same as the specified space type.

- If an activity requires occupying a whole room,

  AND the activity specifies certain features to satisfy,

  THEN the computer finds all spaces that are not designated and that satisfy the specified features.

- IF an activity requires occupying a piece of equipment,

  AND the activity specifies the equipment it needs,

  THEN the computer finds all equipment that is not designated and whose name is the same as that of the specified equipment.

- IF an activity requires occupying a piece of equipment,

  AND the activity specifies certain features to satisfy,

  THEN the computer finds all equipment that is not designated and that satisfies the specified features.

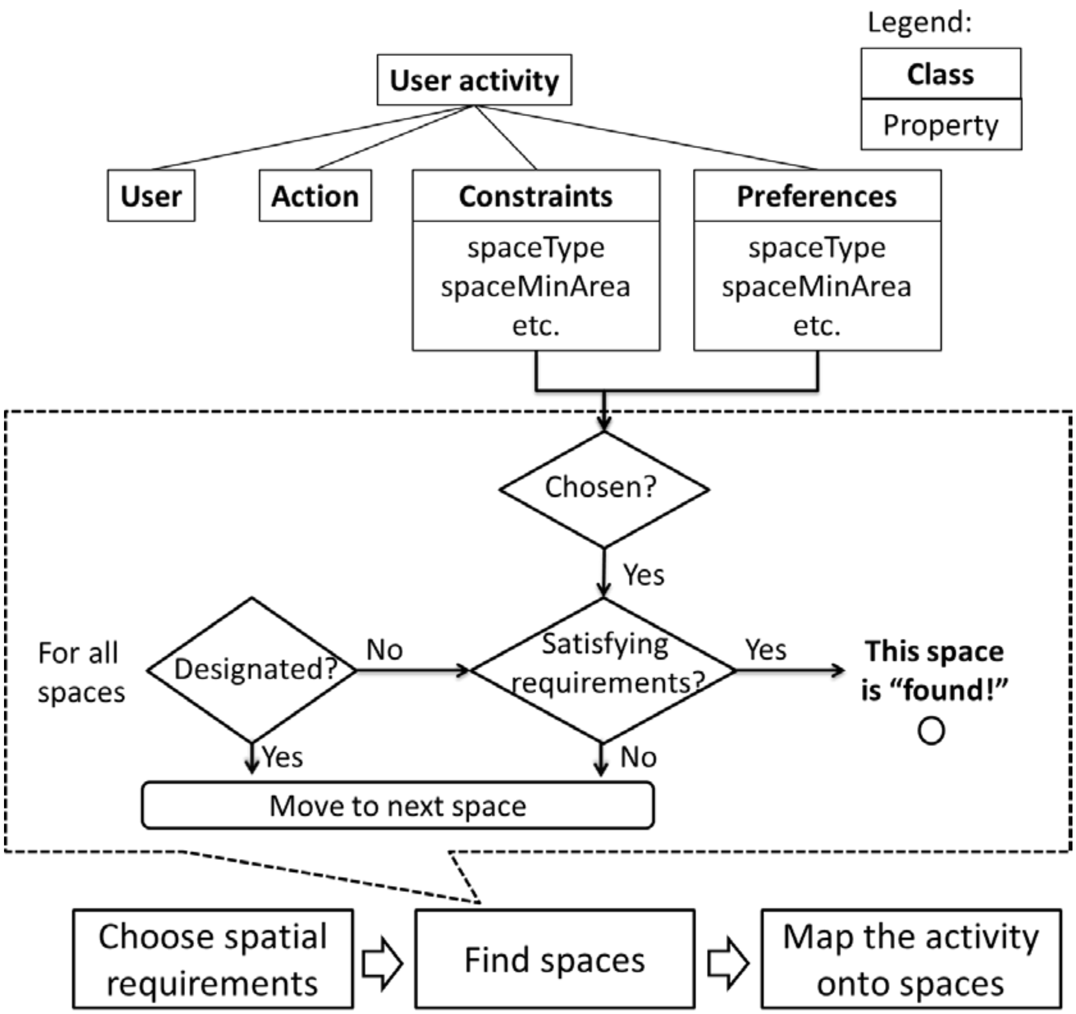

Figure 15. The "find spaces" step of the activity-space mapping method: a computer must find all spaces that are not designated and satisfy chosen spatial requirements of the activity.

5.3. Step 3: map the activity onto spaces

In the third step, the computer checks the choices for two SUTDs (SUTD 4 and 5) and determines how the mapping is completed, to generate activity-space pairs (Figure 16). The number of user groups for the activity (or event quantity) should be calculated to process this step.

- IF an activity requires occupying a whole room,

  AND the activity does not require designated space,

  THEN the computer simply maps the activity onto the found spaces.

- IF an activity requires occupying a whole room,

  AND the activity requires designated space,

  AND the number of spaces found in the previous step is larger than the event quantity,

  THEN the computer divides the found spaces into two entities (i.e., one entity whose number is equal to the event quantity and a second whose number is the remaining number), maps the activity onto the former entity, flags the former entity as "designated," and leaves the latter entity for accommodating other activities.

- IF an activity requires occupying a whole room,

  AND the activity requires designated space,

  AND the number of found spaces is equal to the event quantity,

  THEN the computer maps the activity onto the spaces and flags them as "designated."

- IF an activity requires occupying a whole room,

  AND the activity requires designated space,

  AND the number of found spaces is less than the event quantity,

  THEN the computer maps the activity onto the spaces, flags them as "designated," and records the number of lacking spaces (the event quantity minus the number of spaces) to inform architects about this shortage.

The cases where an activity requires occupying a piece of equipment are similar to the aforementioned cases where an activity requires occupying a whole room except that the mapping is done to both found equipment and spaces that the equipment belongs to.

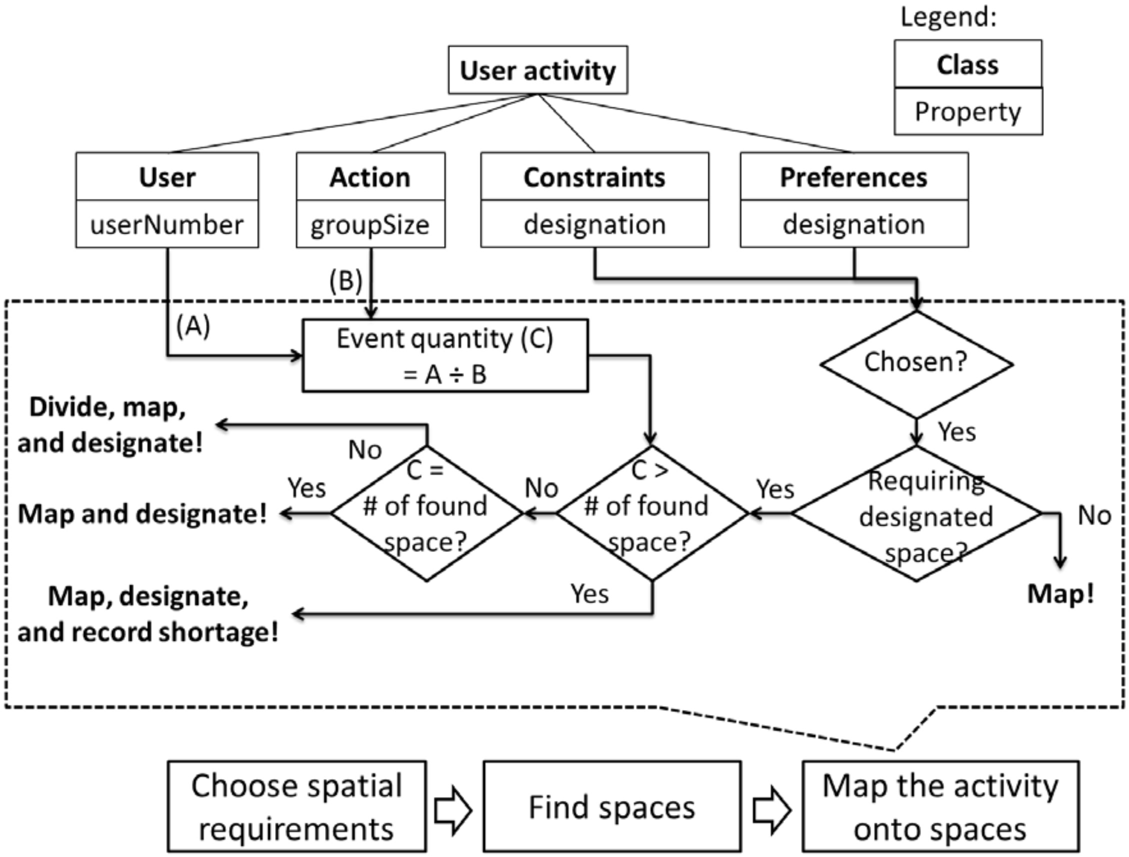

Figure 16. The "map the activity onto spaces" step of the activity-space mapping method: a computer assigns designated space to users of an activity if the activity requires designate space.

5.4. Validation of the developed method

The proposed method must have sufficient conformity and acceptability for use in KSUA, as described in Section 1.7.

5.4.1. Conformity of the developed method

To validate the conformity, I asked five architects with an average of 15.8 years of experience to generate activity-space pairs given information about user activities and spaces. Two architects worked in the United States of America, two worked in Korea, and one worked in England at the time I conducted the test. I used two cases in this test: X publishing company located in Seoul, Korea [5] and Cygnaeus High School located in Jyväskylä, Finland. For the first case, I examined e-mail correspondences and meeting minutes of the publishing company project to identify user activities and spaces of the company. Four user activities and three space types were given to the architects. The second case, the Cygnaeus High School renovation project in Finland, is described by Pennanen (2004). He uses this case to demonstrate the effectiveness of workplace planning. This case is also described in the dissertation of Whelton (2004). Based on these studies, I provided the architects with four user activities and four space types. Conformity was measured by dividing the number of the conforming activity-space pairs by the number of activity-space pairs generated by either the proposed method or the architects.

As a result, 93% of the activity-space pairs generated by either the proposed method or the architects were conforming pairs. This shows that the proposed method has high conformity (Table 8). When asked about the activity-space pairs that were not conforming, an architect answered that she misinterpreted that the “student association club” space in Cygnaeus High School was designated by the “student

[5] The name of the company cannot be mentioned here because of confidentiality constraints.

association meeting" activity although the activity did not require a designated space in the description. The other two architects answered that they thought that each user activity must be mapped onto exactly one space type, which was not always the case.

Table 8. Summary of the conformity test of the activity-space mapping method: the conformity was 93%, implying that the proposed method can emulate the architects' practice of mapping activities onto spaces.

	Experience (years)	The number of the conforming pairs	The number of pairs generated	Conformity
Architect 1	27	8	9	89%
Architect 2	6	9	9	100%
Architect 3	15	8	9	89%
Architect 4	11	9	9	100%
Architect 5	20	8	9	89%
Average	15.8			93%

5.4.2. Acceptability of the developed method

I measured the acceptability of the proposed method to avoid the case of high conformity achieved by random chance. I explained the proposed method to architects after they mapped user activities onto spaces and asked them to score the acceptability of the method from their perspectives. They all agreed that this method is acceptable by giving 5 points on a Likert scale of 1 to 5.

Here are the comments I received from the architects who participated in the test:

- "The activity-space pairs generated by this method can provide a good starting point for discussion between us (architects) and clients. Because client requirements get bigger and bigger during project development, informing clients about where their activities will be occurring, as well as how many people will perform the activities and how often they will do the activities can avoid getting lost during project development.”

- “This method can be a communication tool; architects can immediately show the impact of changes in the space program and user profiles to clients.”
- “It would be better if other features, such as video conferencing tools, aesthetics, natural daylight, and windows, could be better structured to support the mapping. For example, some companies assign a workstation in the frequent flow area to a new employee so that he or she can communicate with other employees frequently. There are also cultural differences in theater chair width in different countries.” [6]

5.5. Conclusions about the method for mapping user activities onto spaces

I proposed a method for mapping user activities onto appropriate spaces, which consists of “choose spatial requirements,” “find spaces,” and “map the activity onto spaces” steps. This method can immediately inform architects about the impact of changes in the space program and user profiles on the activity-space pairs. This method is novel because it formalizes knowledge about how to deal with different space-use types that affect SUA during the mapping, whereas existing user presence models (Ioannidis et al. 2012; Pennanen 2004; Shen et al. 2012; Tabak 2008; Yan and Kalay 2006; Zimmermann 2007) lack this knowledge and, therefore, require architects

[6] He made a good point regarding SUA, but the focus of this research is on the programming phase of office and educational building projects. The factors he mentioned can be formalized and added to the KSUA method in the future when the method is expanded for use in the design phase (Section 8.3).

to map activities onto spaces manually. I validated the proposed method by measuring the conformity of the activity-space pairs generated by the method with the pairs generated manually by five architects in two cases and by asking the architects to score the acceptability of the method.

This work currently has the following limitations and assumptions:

- Since the research focuses on supporting the programming phase of a project, space location, geometry, and aesthetic factors are not taken into account in this method.
- This method assumes that spatial requirements of a user activity must be entirely satisfied by a space to trigger the mapping between the activity and the space.
- Space is designated to user activities, not to users. The concept of designation must be further studied and elaborated in the mapping method.
- User activities are split equally into found spaces because all activity-space pairs are treated equally; there are no different types of activity-space pairs that function differently in the mapping.
- Activities by important users occur only in spaces that satisfy the preferences and do not at all occur in spaces that satisfy the constraints. In the case where preferences of an activity are not a subset of constraints of the activity (e.g., an activity requiring a workstation and, preferably, a meeting room), some portion of the activity could be performed in space that satisfies constraints.

To achieve the full potential of automated SUA, these limitations and assumptions will need to be tested and addressed.

6. Knowledge-based space-use analysis

This chapter describes how the two contributions of this thesis (i.e., an ontology for representing user activities in SUA, explained in Chapter 4, and a method for mapping user activities onto appropriate spaces, explained in Chapter 5) enable the KSUA method to support the programming phase of educational and office buildings. I defined the process of the KSUA method, which consists of four functions, using Integrated Definition for Functional Modeling (IDEF0). I also visualized utilization information, which is produced by a knowledge-based system, to support iterative refinement of the space program and user profiles (Bhatt and Freksa 2010) (Section 6.1). I developed a prototype KSUA system and applied it to selected areas of the Jerry Yang and Akiko Yamazaki Environment and Energy (Y2E2) Building, Stanford University, to demonstrate the effectiveness of the KSUA method (Section 6.2). I tested the prototype system in an existing building to check that the user activities and related information I observed can be represented and analyzed with the KSUA method and to work with realistic information. However, I only defined the space program and user profiles for the programming phase of a project. The goal of this test was not to extend the application of the KSUA method to the occupancy phase of a project, but to demonstrate the effectiveness of the KSUA method in the programming phase. The results were successful: the prototype system predicted utilization based on the given project information and updated utilization in response to two hypothetical but realistic tests that modified the project information.

6.1. Knowledge-based space-use analysis process

The process of the KSUA method has the following four steps: "building the knowledge base," "mapping user activities onto spaces," "computing utilization," and "visualizing the results." Outputs of the last function are used by architects or clients to refine the space program or user profiles, which makes the SUA process iterative. The overall process is shown in Figure 17.

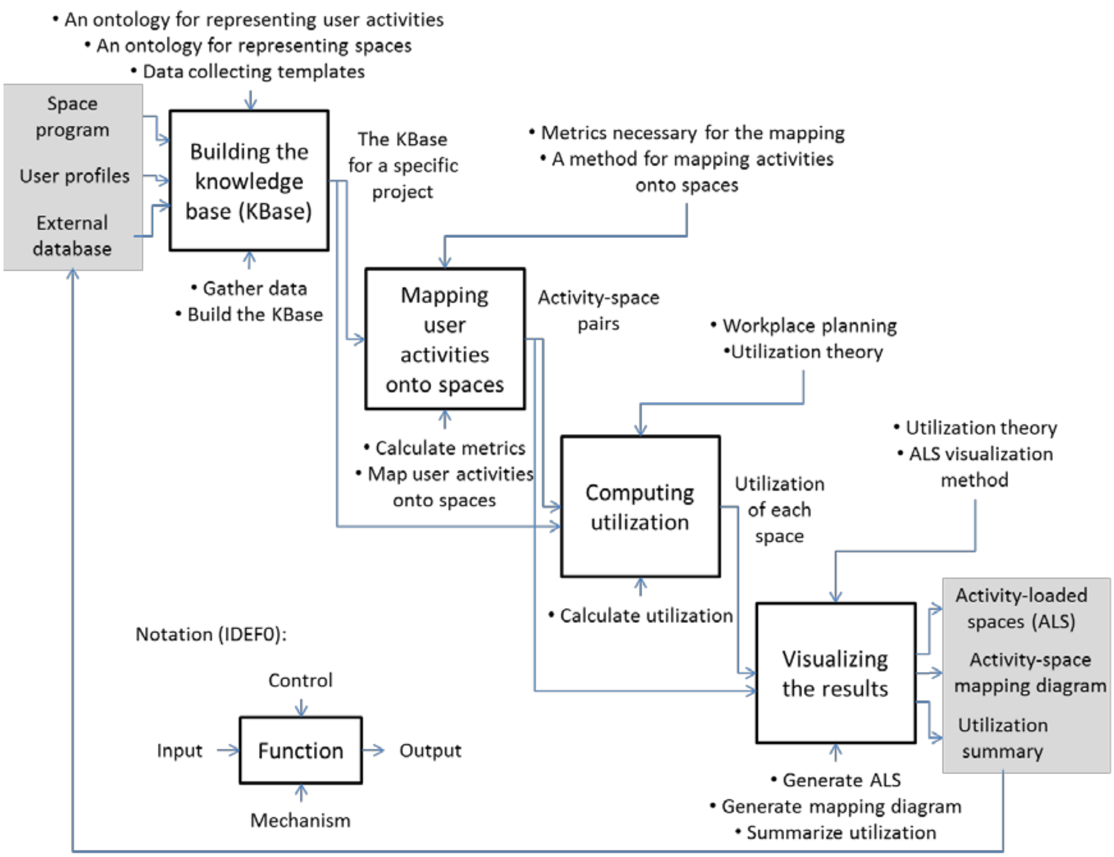

Figure 17. The process of the KSUA method: it has four functions, which support the iterative refinement of the space program and user profiles during project development.

6.1.1. Building the knowledge base

The “building the knowledge base” function takes input from the space program, user profiles, and the external database to provide the knowledge base for a specific project as an output. Ontologies for representing user activities and spaces are needed as a control. I developed an ontology for representing user activities for use in KSUA, as described in Chapter 4. I was able to define the ontology below for representing spaces in frame logic (or F-Logic) relatively easily because the space program is typically in a tabular format (Section 2.4). Unless stated otherwise, the default cardinality of a property is {0:*}, which means that the property can have any number of values or instances (Angele et al. 2009).

Ontology for representing spaces in the KSUA method:

Space :: Entity [area {1:1} *=> float, number {1:1} *=> integer, features *=> Features, openHours {1:1} *=> float, superType {0:1} *=> SpaceSuperType, inaccessible *=> User].

OccupiableSpace :: Space [designated {1:1}*=> boolean].

NonOccupiableSpace :: Space [equipped *=> EquipmentSet].

EquipmentSet :: Entity [equipment {1:1} *=> Equipment, number {1:1} *=> integer, designated {1:1} *=> boolean, features *=> Features, openHours {1:1} *=> float, inaccessible *=> User].

Equipment :: Entity [area {1:1} *=> float].

Data collecting templates, another control, can help architects enter the necessary information even without knowing the ontology for KSUA. Gathering data and building the knowledge base are two mechanisms in this function.

6.1.2. Mapping user activities onto spaces

The "mapping user activities onto spaces" function takes the knowledge base as an input to provide activity-space pairs as its output. A method for mapping user activities onto spaces is one of the contributions I made, which is described in Chapter 5. I also defined the following three metrics for this function:

- *Event quantity* refers to the number of groups for a given activity; it is calculated by dividing the number of users by the size that the activity requires, i.e., group size. The ratio of the activity represents what percentage of users is involved in the activity:

  Event quantity = (the number of users of the activity × the ratio of the activity) ÷ the group size of the action of the activity
- *Load* refers to hours that an activity demands from spaces:

  Load = event quantity of the activity × the frequency of the activity × the duration of the action of the activity
- *Space-use area* refers to the area that a group of users requires for an activity:

  Space-use area = the group size of the action of the activity × space criteria of the action of the activity

These controls are put into the KSUA system in the form of rules.

6.1.3. Computing utilization

The "computing utilization" function takes the knowledge base (i.e., the output of the first function) and activity-space pairs (i.e., the output of the second function) to compute the utilization of each space based on the utilization theory (Section 2.2) and workplace planning theory (Section 2.6). The process is described in detail in Section

2.6. Note that workplace planning theory computes load per space for an activity by dividing the activity load by the number of the space types that accommodate the activity. In this way, the KSUA method spreads the activity load over all spaces accommodating the activity. These controls are put into the KSUA system in the form of rules.

6.1.4. Visualization of space-use analysis results

The "visualizing the results" function takes outputs of "mapping user activities onto spaces" and "computing utilization" functions to provide visualized results of SUA. The utilization theory, one of the controls in this function, was defined in Section 2.2. This function has three outputs: activity-loaded spaces, the activity-space mapping diagram, and the utilization summary.

Since KSUA makes spaces in the space program "activity-loaded," architects can see the visualization of an activity-loaded space easily by selecting the space in the space program (Figure 18). This visualization shows which activities occupy a space (by black area and the name of the activities), how long the activities occupy the space (by loads per space on the x-axis), how much of the space the activities occupy (by space-use area on the y-axis), and how many area-hours of the space cannot be used even if the space is vacant (by gray area).

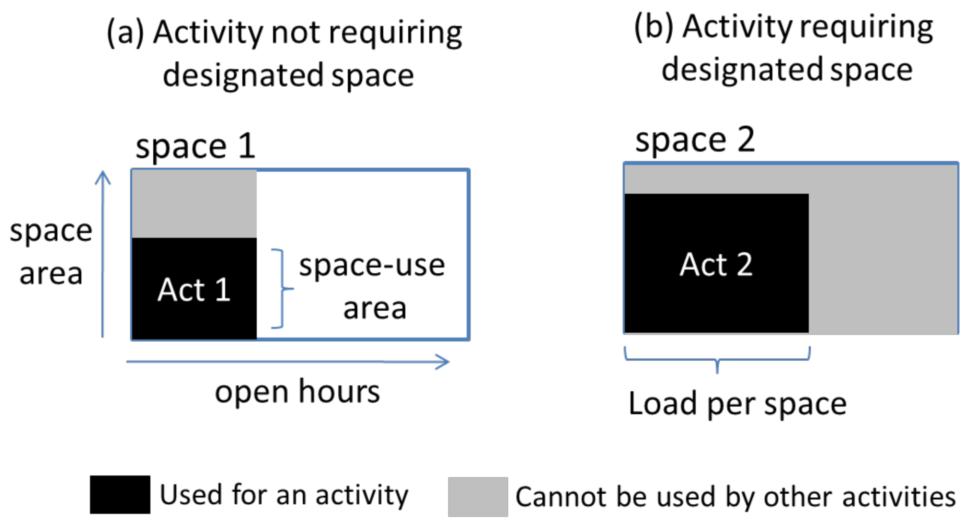

Figure 18. Visualization of activity-loaded space (ALS): (a) represents ALS where activity 1 does not require designated space, and (b) represents ALS where activity 2 requires designated space.

The activity-space mapping diagram illustrates activity-space pairs so that architects can see the mapping results at a glance. The utilization summary allows architects to see and document the utilization of each space by providing a table that lists spaces, their utilizations and the corresponding implication.

6.2. Prototypical implementation

I conducted a trial run in selected areas of the Y2E2 Building, Stanford University (Figure 19) to show the effectiveness of the KSUA method in terms of consistency and quick updating of SUA. I first developed a prototype KSUA system. The project-independent domain knowledge about SUA was represented in F-Logic (Angele et al. 2009; Kifer 2005). This knowledge was embedded in the prototype KSUA system. I also used Flora-2, an inference engine for a knowledge-based system (Yang et al. 2003), and Eclipse Software Development Kit (SDK), a software development environment, in the development of the prototype. I then gathered and represented 13 user activities and nine space types, also in F-Logic, based on observation, hourly measurement of space-use, interviews with users, the BIM of the building, and Stanford University's Space and Furniture Planning Guidelines (2009) and added them to the knowledge base of the prototype. Although I manually visualized the results of the analysis, I specified the data I used that were drawn from the previous functions to show that the visualization can also be automated.

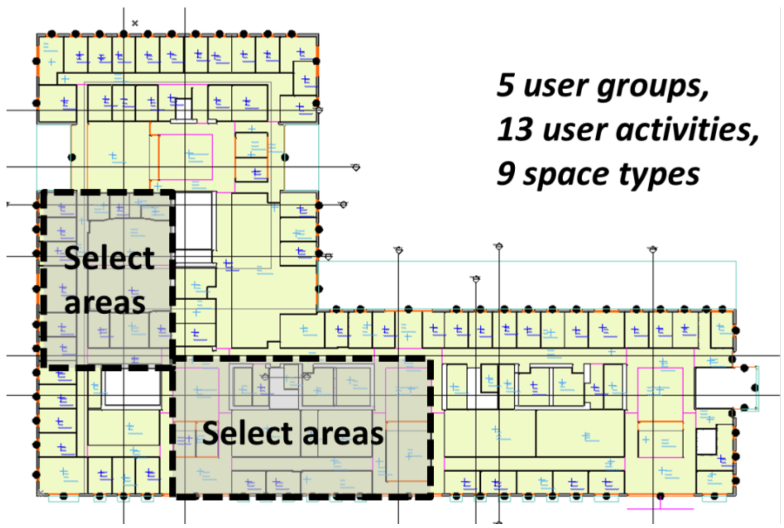

Figure 19. Selected areas in the Y2E2 Building, Stanford University: 13 activities of five different user groups were identified in nice space types in the selected areas.

6.2.1. Building the knowledge base for the prototype system

The domain knowledge about SUA is project-independent and is therefore embedded in the prototype system for multiple projects. I developed 32 rules that represent metrics necessary for mapping, a method for mapping user activities onto spaces, a method for computing utilization, and utilization implication levels in F-Logic (Appendix E). Here are some examples of the rules:

- Metrics necessary for the mapping (Computing load of an activity):

  ?ACT002[load -> ?V002] :- ?ACT002:Activity[evtQty -> ?_EQ002, frequency -> ?_FR002, action -> ?AC002], ?AC002:Action[duration -> ?_DUR002], ?V002 is (?_EQ002 * ?_FR002 * ?_DUR002).

  (To compute the activity load, the prototype KSUA system multiplies the event quantity of the activity, the frequency of the activity, and the duration of the action of the activity.)

- Computing utilization:

  ?SP309[utilization -> ?UTIL309] :- ?SP309:OccupiableSpace[loadInSpace -> ?_VAL309, openHours -> ?_OPEN309], ?UTIL309 is (?_VAL309 / ?_OPEN309).

  (If a space type has the value for total activity loads (loadInSpace), the prototype KSUA system divides the value by open hours of the space type to compute utilization.)

In terms of project-specific information about user activities and spaces, I represented 13 activities of five different user groups and nine space types, as shown in Table 9. I scaled down the number of undergraduate students and graduate students proportionate to their enrollment in the Department of Civil and Environmental Engineering, the department that is located in and uses this building. I counted the number of faculty, researchers, and staff, in the selected areas. I identified user activities observing those areas and also interviewed users to gather information on the spatial requirements of each activity. I also referenced Stanford University's Space and Furniture Planning Guidelines (2009) to identify the spatial requirements of user activities, e.g., minimum size of a private office for faculty.

Table 9. Users, user activities, and spaces in selected areas of Y2E2 Building.

User groups	1. 25 undergraduate students 2. 122 graduate students 3. 5 faculty 4. 12 non-faculty scholars (e.g., visiting scholars and research associates) 5. 6 staff
User activities	1-1. Graduates having classes 1-2. Undergraduates having classes 2-1. Graduates meeting for coursework (computer work) 2-2. Undergraduates meeting for coursework (computer work) 3-1. Graduates meeting for coursework (no computer work) 3-2. Undergraduates meeting for coursework (no computer work) 4. Graduates meeting for research 5-1. Graduates studying individually 5-2. Undergraduates studying individually 6. Faculty working 7. Staff working 8. Non-faculty scholars working 9. Faculty meeting
Space types	1. A computer cluster containing 25 computers (74 m^2) 2. A classroom (74 m^2) 3. Three small conference rooms with a computer (9 m^2) 4. A conference room with a computer (27 m^2) 5. Five private offices (17 m^2) 6. Three shared offices containing 2 workstations (15 m^2) 7. Two cubicle spaces containing 6 workstations (33 m^2) 8. Two small conference rooms (15 m^2) 9. A large conference room (51 m^2)

The knowledge base consists of 64 facts about this building in F-Logic. Here are some examples of the facts describing project-specific information:

- User activities:

  gradsMeetingForResearch : TypicalActivity [user -> grads, action -> haveResearchMeeting, ratio -> 1.0, frequency -> 0.1, constraints -> cons4, preferences -> pref4].

  (Graduate students have a group meeting for discussing their research. The meeting occurs once in two weeks, and 100% of the students perform the meeting.)

- Space type:

  smallConferenceRoom : OccupiableSpace [superType -> conferenceRoom, area -> 15, number -> 2, openHours -> 8.0, features -> quiet, designated -> False, inaccessible -> noOne].

  (Two small conference rooms, the area of which is 15 m^2, are open for eight hours daily. They are in quiet conditions and can be occupied by any group of users who want to use them.)

6.2.2. Analysis results

The prototype system mapped the 13 activities onto the nine spaces automatically, generating 26 activity-space pairs. This automated mapping of user activities onto spaces and utilization computation based on this mapping contribute to the consistency of SUA. That is, utilization is always the same given the same user activities and spaces because the KSUA method formalizes the concepts that are related to three space-use perspectives (space, user, and user activity) and their relationships. Figure 20 shows the activity-space mapping diagram of the Y2E2 Building. There are many-to-many relations between activities and spaces, meaning that each activity occurs at least in one space type and at most in four different space types, and each space type also accommodates at least one activity and at most six different activities. The activity-space mapping diagram visualizes those relations and the utilization implication levels of space types in one figure.

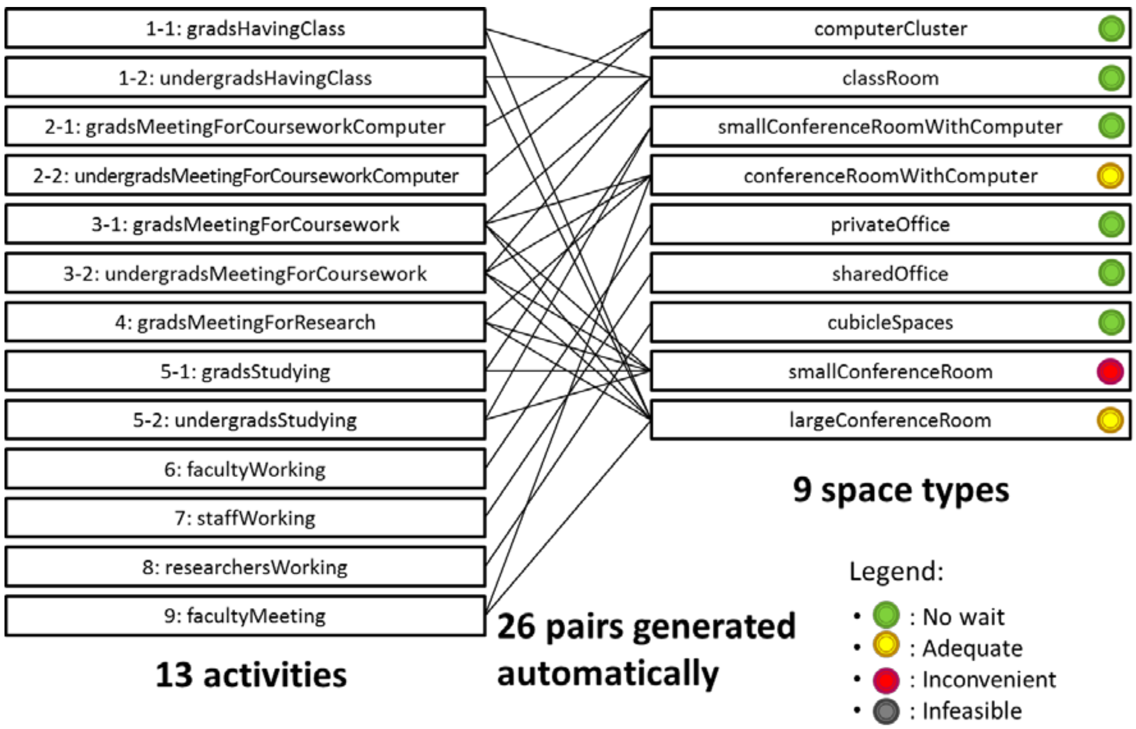

Figure 20. Activity-space mapping diagram of the Y2E2 Building (initial setting): the KSUA method generated 26 activity-space pairs and computed utilization automatically.

Architects can populate an activity-loaded space by selecting a space in the activity-space mapping diagram or in the space program to see the use of the space in detail. For example, Figure 21 represents the activity-loaded space of a conference room with a computer. As shown in the figure, the space accommodates four activities, i.e., faculty meeting for 0.23 hours, graduates meeting for coursework for 1.48 hours, graduates meeting for research for 2.03 hours, and undergraduates meeting for coursework for 0.87 hours. The utilization of this space is 0.58, which implies that those activities may need to be scheduled in this space. The space-use areas those activities require are much less than the area of this space (27 m^2), which means that the area of the space could be reduced without affecting the space-use of those activities.

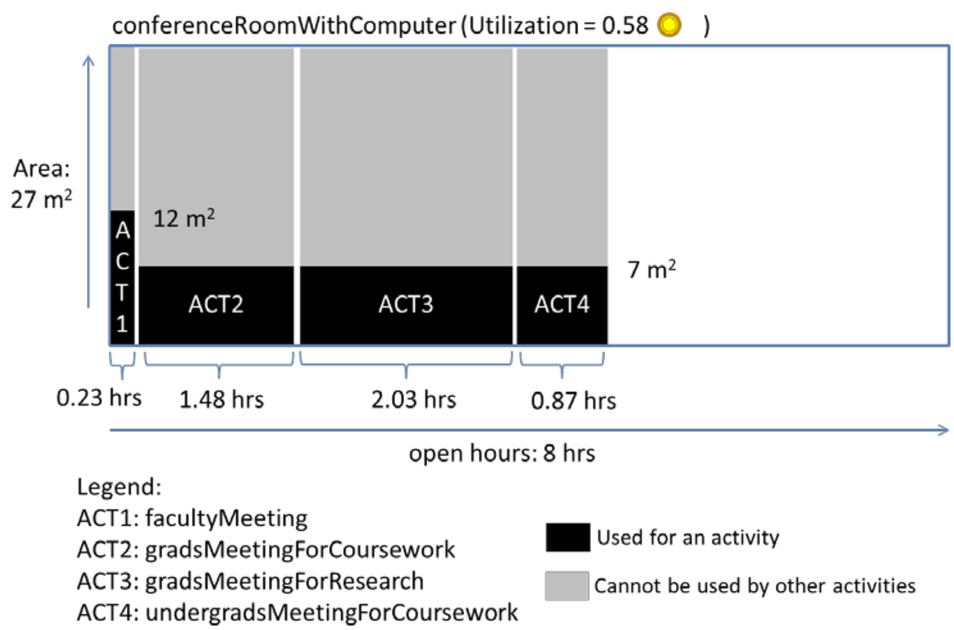

Figure 21. Activity-loaded space of the conference room with a computer: it shows which activities occupy a space, how long the activities occupy the space, and how much of the space the activities occupy.

Utilization of each space is summarized in Table 10. Since the utilization of small conference rooms is 0.99 (i.e., its implication level is classified as “inconvenient”), the space program and/or intended usage of user profiles should be modified by iterative refinement with the assistance of the KSUA method.

Table 10. Utilization summary table of the Y2E2 Building (initial setting): the utilization of small conference rooms is 0.99, which needs to be addressed by iterative refinement.

Space	Utilization	Implication
Computer cluster	0.20	No wait
Classroom	0.46	No wait
Small conference room with a computer	0.44	No wait
Conference room with a computer	0.58	Adequate
Private office	0.40	No wait
Shared office	0.50	No wait
Cubicle space	0.50	No wait
Small conference room	0.99	Inconvenient
Large conference room	0.74	Adequate

6.2.3. Iterative refinement in space and intended usage

In this trial run, I developed two hypothetical but realistic options to respond to the unacceptably high utilization of small conference rooms in the initial setting. The first option is to increase the number of small conference rooms from two to three while maintaining the gross area of this building by reducing the area of a large conference room (51 m^2 to 36 m^2). If the first option does not reduce the utilization of small conference rooms to an acceptable implication level (i.e., “no wait” or “adequate” level), then the second option would be to maintain the first option but to prevent undergraduate students from using small conference rooms and require them to find other spaces for their individual study. Note that the first option is a change in the space program, and the second option is a change in intended usage of user profiles. The KSUA method must be able to update the utilization according to either or both changes.

To test these options, I first changed the knowledge base according to the first option. The prototype KSUA system then updated the activity-space pairs and computed the utilization of spaces automatically. Figure 22 is the activity-space mapping diagram of this scenario, which applies the first option to the initial setting. The system deleted two pairs (pair 1: graduates having classes and a large conference room and pair 2: undergraduates having classes and a large conference room) from the initial setting because those activities require a space that is larger than 37 m^2. These changes in activity-space pairs and the increased number of small conference rooms resulted in changes in the utilization of spaces: (1) the utilization of a classroom changed from “no wait (0.46)” to “adequate (0.58),” (2) the utilization of a large conference room changed from “adequate (0.74)” to “no wait (0.48),” and (3) the utilization of a conference room with a computer changed from “adequate (0.58)” to “no wait (0.48).” Although the utilization of small conference rooms dropped by 0.17, it remained under the “inconvenient” level at 0.82.

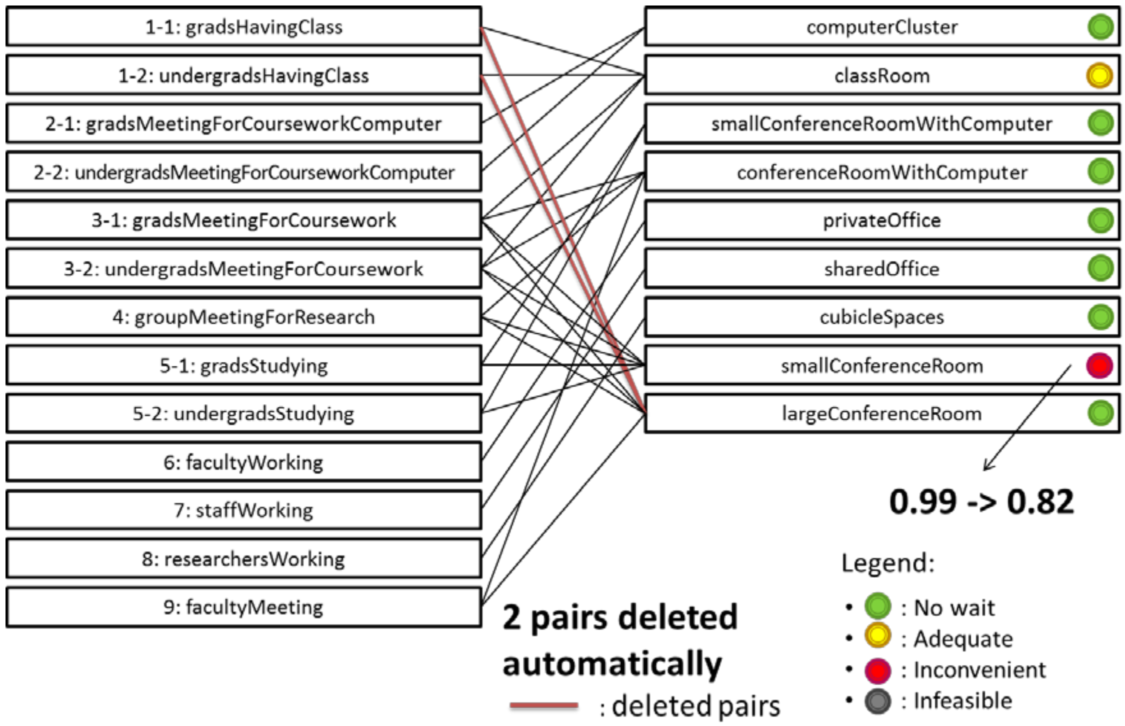

Figure 22. Activity-space mapping diagram of the Y2E2 Building (the first option): two pairs were deleted automatically in response to the changes in space program.

Since, with the first option alone, the utilization of small conference rooms is still unacceptable, I changed the knowledge base according to the second option. As a result, the knowledge system deleted two pairs (pair 1: undergraduates meeting for coursework and small conference rooms and pair 2: undergraduates studying individually and small conference rooms) and added two pairs (pair 1: undergraduates studying individually and conference rooms with a computer and pair 2: undergraduates studying individually and a large conference room), as shown in Figure 23. As a result, the utilization of small conference rooms changed from “inconvenient (0.82)” to “adequate (0.68)” while maintaining utilizations of all other spaces at an acceptable level.

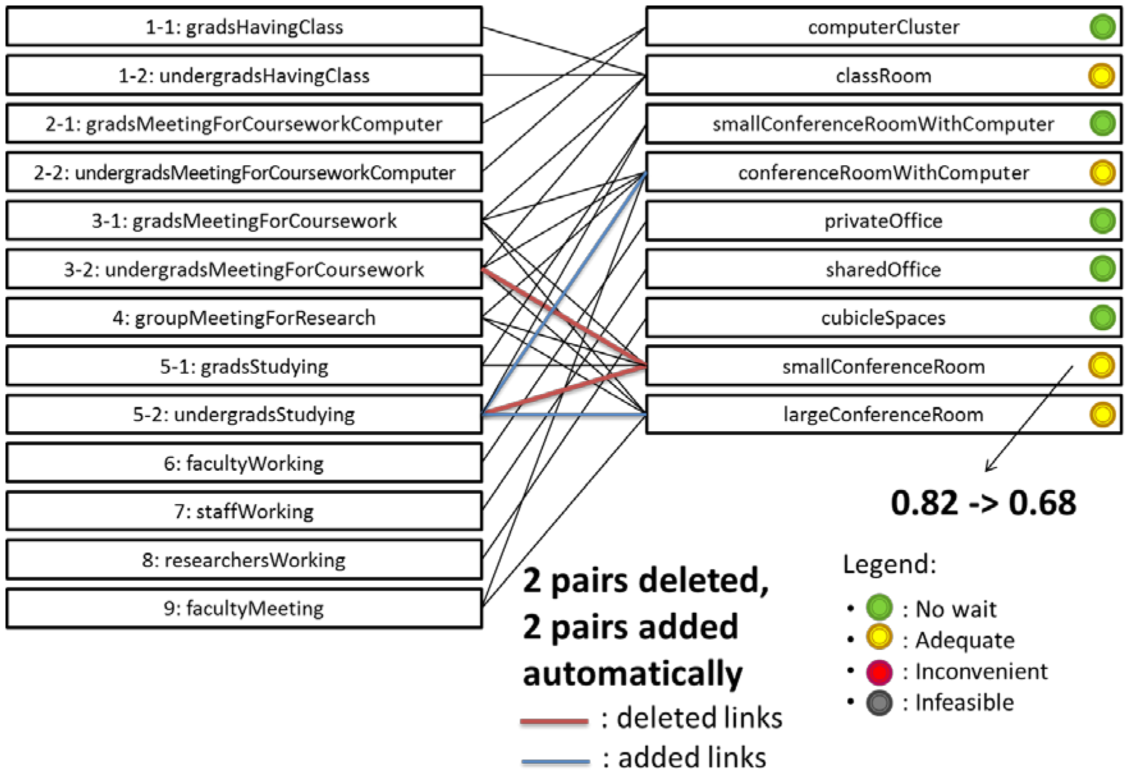

Figure 23. Activity-space mapping diagram of the Y2E2 Building (the second option): two pairs were deleted, and two pairs were added automatically in response to the changes in user profiles.

The iterative refinement process described in this section demonstrates that the KSUA method enables architects to update utilization quickly in response to changes in space program and user profiles. When clients or architects change any space and user activity information that affects space-use, the prototype system can immediately track the changes and update utilization because it has a formalized KSUA process. The summary of the iterative refinement and its impact on space-use is shown in Table 11.

Table 11. Summary of the iterative refinement and its impact on space-use in the Y2E2 Building: the utilization of small conference rooms was reduced from 0.99 to 0.68 by iterative refinement. The KSUA method supports this refinement by providing architects with utilization information quickly and consistently.

Refinement	Initial setting	First option	Second option
Description	N/A	Increasing the number of small conference rooms by reducing the area of a large conference room	Not allowing undergraduate students to use small conference rooms
Impact on activity-space pairs	N/A (26 activity-space pairs)	2 pairs deleted	2 pairs deleted/ 2 pairs added
Space	Utilization	Utilization	Utilization
Computer cluster	0.20	0.20	0.20
Classroom	0.46	0.58	0.67
Small conference room with a computer	0.44	0.37	0.38
Conference room with a computer	0.58	0.48	0.63
Private office	0.40	0.40	0.40
Shared office	0.50	0.50	0.50
Cubicle space	0.50	0.50	0.50
Small conference room	0.99	0.82	0.68
Large conference room	0.74	0.48	0.63

6.3. Conclusions from prototypical implementation

I defined the process of the KSUA method, which consists of the following four functions: “building the knowledge base,” “mapping user activities onto spaces,” “computing utilization,” and “visualizing the results.” I demonstrated the effectiveness of the KSUA method by developing a prototype system and conducting a trial run on selected areas of the Y2E2 Building at Stanford University. The results show that the KSUA method leads to quick updating of utilization with high consistency in SUA. I can also use this prototype KSUA system to quantitatively validate the KSUA method, which is described in Chapter 7. In addition, I developed the KSUA method based on a knowledge-based systems approach, which has generality as one of its innate strengths because the approach develops the right degree of abstraction of domain knowledge (in this research, concepts for SUA and their relationships) and uses the abstraction recursively within the defined world (in this research, office and educational buildings) (Brachman and Levesque 2004).

To enhance the practicality of the KSUA system, it is important to reduce the burden of data entry. Connecting the KSUA method to other computational models could reduce the architects' effort of adding project-specific information in the knowledge base of the system. A BIM, for example, provides a computational representation of space information, which can seamlessly feed into the KSUA method. Having a database of the user profiles that is sortable by various factors, such as project types and regions where a project is conducted, could also reduce architects' effort of gathering the necessary project-specific information. Such a database would be more useful when architects intend to apply SUA in the programming phase of a project, where information about users and their activities is often insufficient for building the knowledge base.

7. Validation test based on the charrette test method

This chapter describes the charrette test I conducted to validate the KSUA method quantitatively. A charrette test is a validation method by which the effectiveness of a computing method can be tested (Clayton et al. 1998). Participants perform a task without the method a researcher developed (i.e., the control group, or the conventional process) and with the method (i.e., the experimental group, or the innovative process), for each trial. Although the charrette test has limitations regarding the impact of familiarity and training, measurement instruments, and arbitrariness of the process, it can increase both reliability and validity of research conclusions over other validation methods that are commonly in computing method research projects, such as worked example, demonstration, and trial (Clayton et al. 1998).

I conducted the test using six Stanford graduate students (as novice architects) and six expert architects on two cases, i.e., X publishing company in Korea and Cygnaeus High Schoool in Finland (Pennanen 2004). Each case had only four user activities and three or four spaces for participants to predict and update utilization of each space within a two-hour time frame. In each case, I let participants initially predict utilization and then update it twice according to changes in the space program or user profiles (Section 7.1 and 7.2). As a result, novice architects could update predictions about space utilization 6.5 times faster with the KSUA method than with the conventional process (manual SUA, as shown in Section 1.4) and do so much more consistently (the standard deviation of predictions across the novice architects was 68% less with the KSUA method). The test also showed that novice architects with the KSUA method outperform expert architects without the method (Section 7.3).

7.1. Overall structure of the validation test

In the charrette test, I regarded manual SUA without the prototype KSUA system as the "conventional process" and SUA with the prototype system as the "innovative process." By letting six Stanford graduate students (as novice architects) conduct the analysis given the same project information, once without the prototype system and once with the system, and comparing the performance of the two groups, I could measure the "power of the KSUA method for novice architects." Although I could not measure the "power of the KSUA method for expert architects" because I did not measure the performance of architects with the prototype system, I measured the "power of experience in SUA" by comparing the performance of graduate students without the prototype system to the performance of six architects without the system. I also measured the "power of the KSUA method over experience" by comparing the performance of graduate students with the prototype system to the performance of expert architects without the system. The average experience of the expert architects was 13.2 years (Figure 24). To claim the generality of the KSUA method, I conducted the test on two cases (one educational building and one office building), each of which with three trials (an initial prediction and two updates), and 12 participants (six Stanford graduate students as novice architects and six expert architects).

Six trials over two cases also helped reduce the effects of learning, which take place during one trial and affect the results in successive trials (Clayton et al. 1998). In addition, variations in time spent for updating and predicted utilization in the first update with the manual SUA were similar to those in the second update, as shown in Figure 6 in Section 1.4.1. This implies that the effects of learning are not significant for conducting the task of SUA.

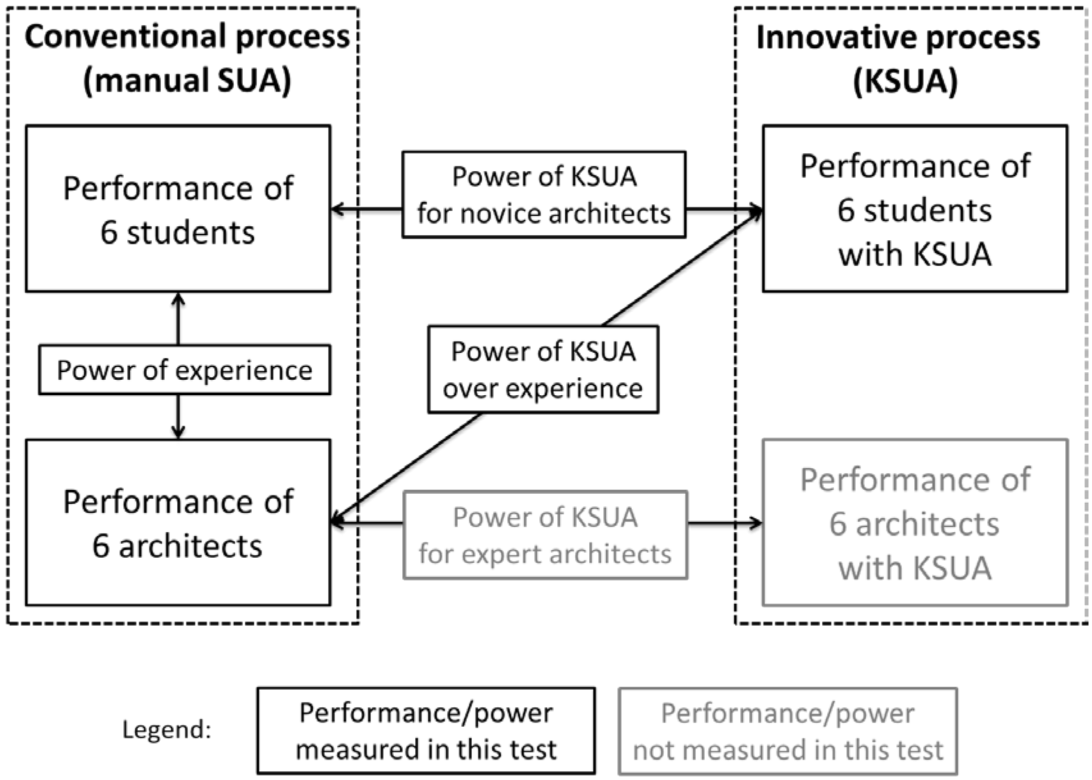

Figure 24. Overall structure of the charrette test: I measured the power of the knowledge-based space-use analysis (KSUA) method for novice architects, power of experience in space-use analysis, and the power of the KSUA method over experience.

7.2. Implementation of the validation test

The validation test was two hours long. Figure 25 depicts the overall process of the test.

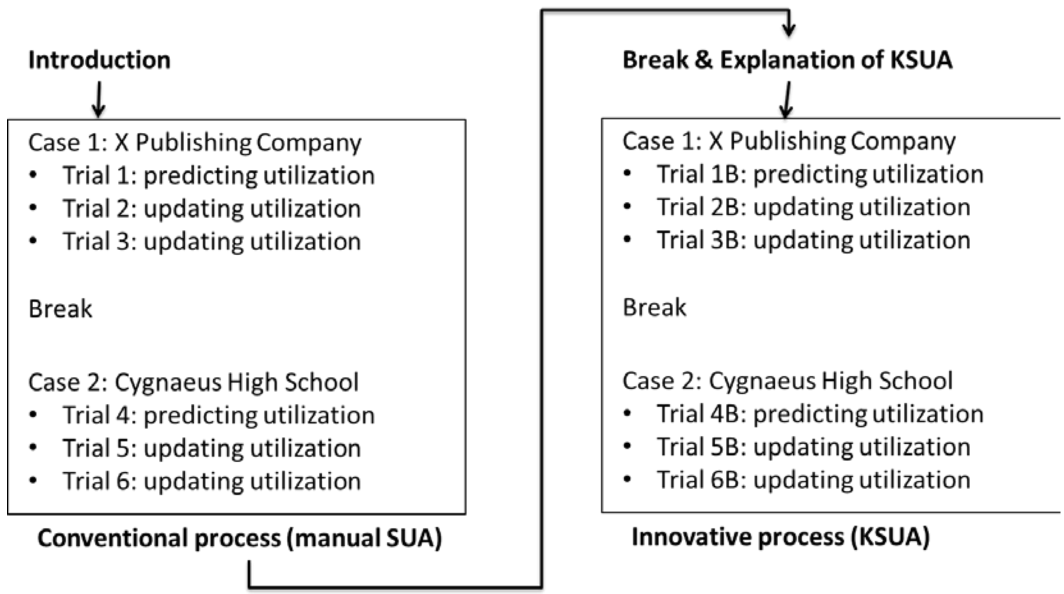

Figure 25. The overall process of the charrette test: each participant went through six trials over two cases twice, once without the KSUA method (manual SUA) and once with the KSUA method.

After introducing the purpose of the test and the concept of space utilization, I made participants predict the utilization of each space based on user profiles that contained four user activities for X publishing company and the space program for the company (Trial 1). When the participants answered about the utilization, I made the participants update the utilization if necessary according to further information I provided. Twice I told them to assume the information about user activities and/or spaces had changed during project development (Trials 2 and 3). After a short break, I made the participant predict and update the utilization for the Cygnaeus High School case, similarly to the X publishing company case (Trials 4, 5, and 6). The participants also recorded the time spent for each trial.

After a break, I introduced the participants to the KSUA method and how user activities and spaces are represented in KSUA. Since the prototype KSUA system I developed (Section 6.2) did not have a user interface, I developed a pseudo user interface using Microsoft Excel with which the participants could provide all necessary information for building the project-specific knowledge base in the KSUA system without knowledge about F-Logic (Figure 26). Thus, in the validation test, I asked the participants to enter the necessary information into the pseudo user interface and update the information according to the provided user profiles and the space program for both cases (Trials 1B through 6B). The participants also recorded time spent for entering and updating the information for each trial.

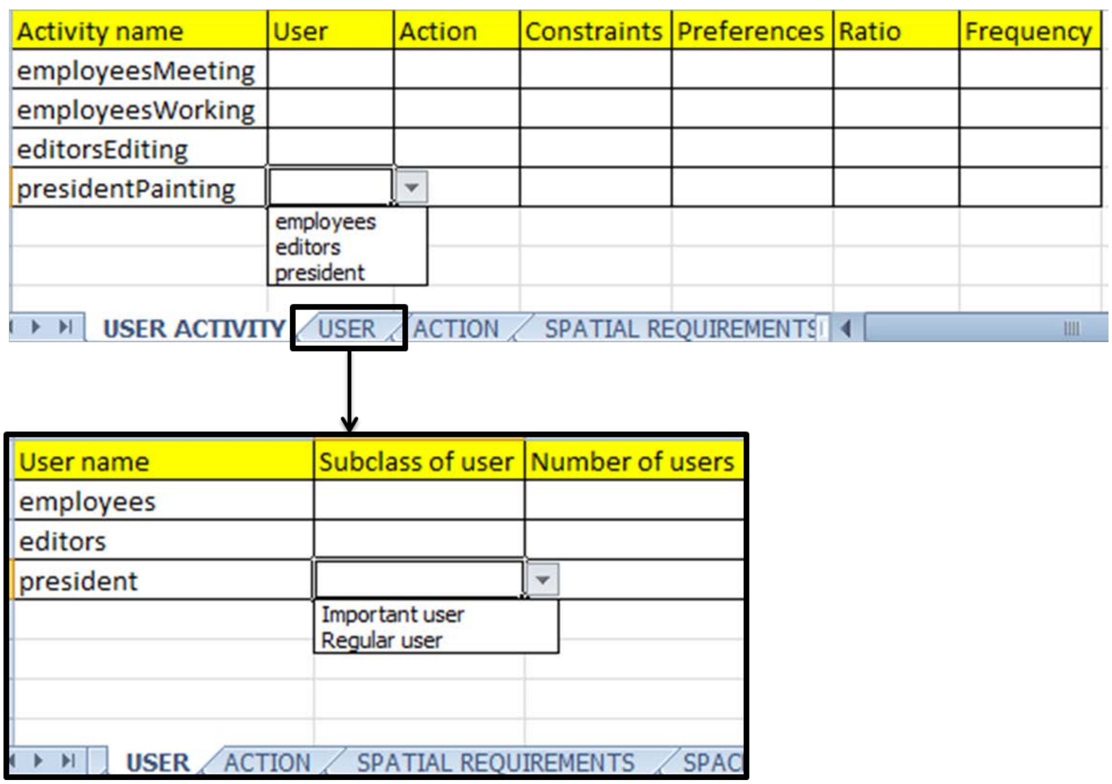

Figure 26. The pseudo user interface for the charrette test: participants of the test entered the information necessary for building the knowledge base into the seven tabs of the interface: user activity, user, action, spatial requirements, space, equipment set, and equipment.

After the session with the participants, I built the knowledge base in F-Logic for each participant based on his or her input to the pseudo user interface. The prototype KSUA system then mapped user activities onto spaces and computed the utilization of each space automatically based on the formalized process I described in Section 6.1. I compared the performance of the conventional process where the participants manually predicted and updated the utilization and the performance of the innovative process where the prototype KSUA system computed the utilization based on the information the participants entered into the pseudo user interface. For the comparison, I did not count the time I spent on building the knowledge base because I assume that the prototype system automatically builds the knowledge base once architects enter information about user activities and spaces into the pseudo user interface.

7.3. Results of the validation test

This section describes the results of the validation test in terms of the speed for updating utilization and the consistency in predicted utilization across architects. I measured time spent for updating utilization (and updating the pseudo user interface in case of the KSUA method) and the ratio of time for initial prediction and time for updating to claim the improved speed for updating by KSUA. I also measured the standard deviation of predictions (initial and updated) across the participants to claim the improved consistency across architects by KSUA. Detailed results from the test are shown in Appendix F.

7.3.1. Improved speed for updating utilization

The KSUA method improved the speed of SUA by eliminating the human effort of speculating on all the possible activity-space pairs and the burden of computation in SUA. The test results show that students (novice architects) spent 85% less time for updating utilization (or 6.5 times faster), from 4.4 minutes without the KSUA method to 0.7 minutes with the KSUA method. Expert architects could update utilization spending only 3.1 minutes (showing the power of experience), but students could update utilization with the KSUA system much faster than the expert architects (showing the power of KSUA over experience) (Figure 27).

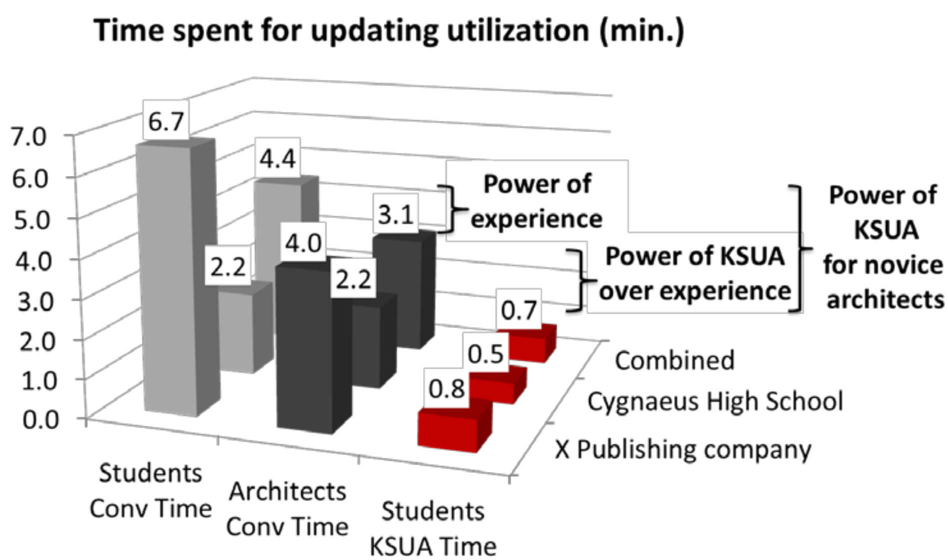

Figure 27. Time spent for updating utilization in the charrette test: students (novice architects) could update predictions about space utilization quickly with the KSUA method.

I conducted the t-test to compare the time spent by students without the KSUA method and the time spent by students with the KSUA method. I found that the critical t value (p = 0.05), which the calculated t value must exceed to claim a significant difference at 95% confidence, was 2.447. Since the t value calculated from the data was 2.836, the two samples are significantly different: there is less than a 5% probability that students can update utilization without the KSUA method as quickly as with the method by chance alone. The P value was 0.0297.

In addition, using the KSUA method, students could spend only 5% of their initial prediction time for each update compared to 24% (for students, or novice architects) or 28% (for expert architects) without the KSUA method. Since information about user activities and spaces changes very frequently during project development, reducing this ratio is important in SUA to enable architects to respond to these changes rapidly. For example, to update 20 changes in the space program or user profiles, the KSUA method requires students to spend only the same amount of time they needed for the initial prediction. In contrast, to respond to the 20 changes, architects without the KSUA method have to spend 4.8 or 5.6 times as much as the amount of time needed for initial prediction (Figure 28).

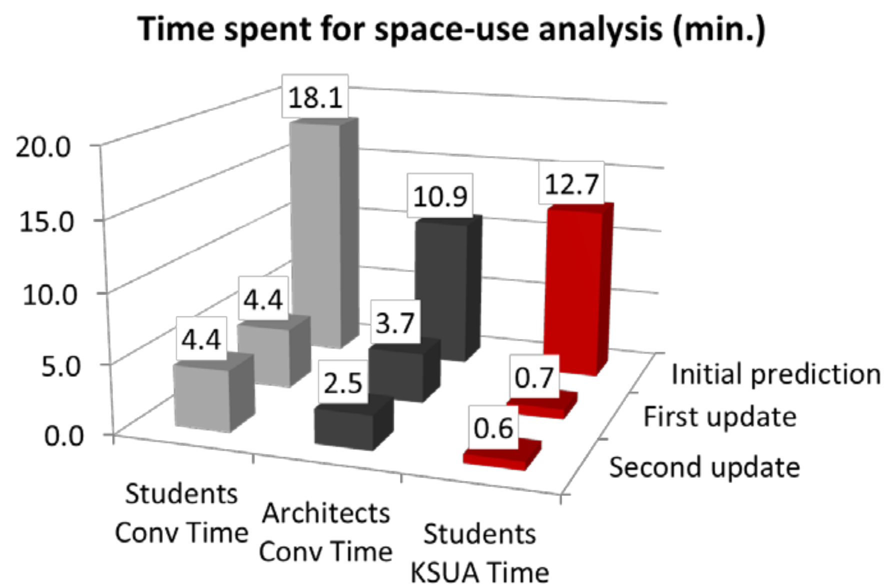

Figure 28. Time spent for space-use analysis in the charrette test: students (novice architects) spent only 5% of their initial prediction time (12.7 minutes) for each update with the KSUA method.

7.3.2. Improved consistency in predicted utilization

The KSUA method improved consistency in predictions across architects by formalizing and integrating concepts that are related to three space-use perspectives (space, user, and user activity). The standard deviation across students (novice architects) was reduced by 68% from 0.24 (manual SUA) to 0.07 (KSUA), which shows that the KSUA method improved consistency in predicted utilization across novice architects. In both cases (i.e., the X publishing company case and the Cygnaeus High School case), the power of experience was shown to be large as the standard deviation across architects (expert architects) was almost half of the standard deviation across students without the KSUA system. However, students with the KSUA system predicted and updated utilization more consistently than architects without the KSUA system, which shows the power of KSUA over experience (Figure 29).

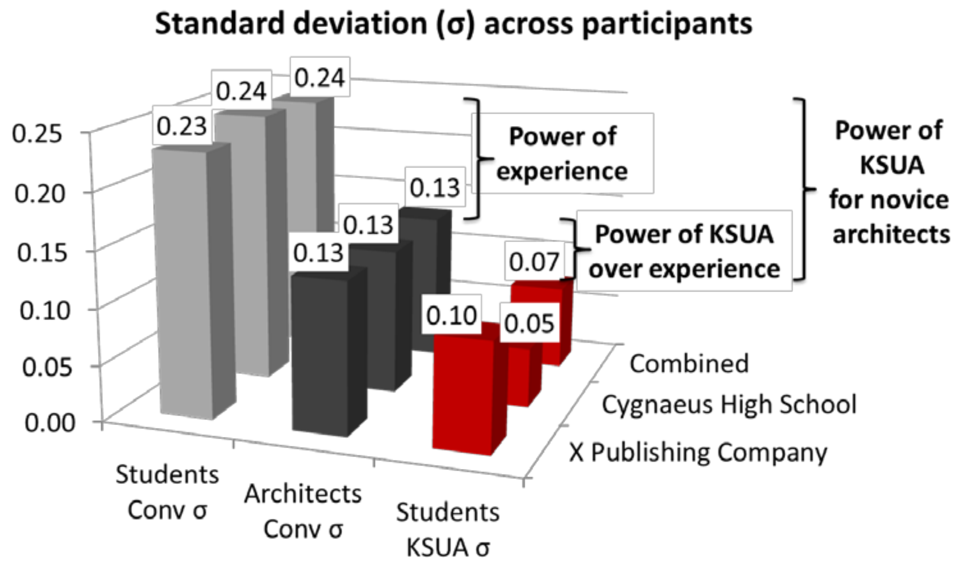

Figure 29. Standard deviation across participants in the charrette test: students (novice architects) could predict space utilization consistently with the KSUA method.

I conducted the t-test to compare the standard deviation of students without the KSUA method with the standard deviation of students with the KSUA method. The resulting critical t value (p = 0.05) was 2.228. Since the t value calculated from the data (5.265) exceeds the critical value, the two samples are significantly different: there is less than a 5% probability that students can predict utilization without the KSUA method as consistently as with the method by chance alone. The P value was 0.000365.

The standard deviation across architects (either novice architects or expert architects) should be 0 (zero) in an automated SUA given the same user activity and space information. The standard deviation of 0.07 across students with the KSUA method is the result of misrepresentation of the user activity or space information by students. Therefore, consistency could be further improved if the prototype KSUA system is equipped with pop-up descriptions regarding the representation of user activities and spaces to support architects' data entry. Connecting the KSUA method to other computational models, e.g., BIM and an external database, could also contribute to the consistency of SUA by semi-automating the representation.

7.4. Conclusions from validation test based on charrette test method

I conducted the charrette test to provide quantitative evidence of power and generality of the KSUA method, which contains the two theoretical contributions, i.e., an ontology for representing user activities for use in SUA and a method for mapping user activities onto spaces to generate activity-space pairs. As a result, six graduate students (novice architects) who participated in the test could update the utilization of each space 6.5 times faster with the KSUA method than with manual SUA. In addition, the standard deviation of predictions across the students with the KSUA method was 68% less than the standard deviation of predictions across the same students without the KSUA method, i.e., the manual SUA. These results provide evidence of improved speed for updating and consistency in predictions in SUA for novice architects. I conducted the test on two cases (one educational building and one office building), each of which had three trials (an initial prediction and two updates), which shows the generality of the KSUA method.

The test results also show the power of experience in SUA and the power of the KSUA method over experience based on the measured performance of six expert architects without the KSUA method and its comparison with the performance of the students. That is, using the manual method, expert architects could conduct SUA faster and more consistently than students (novice architects). However, with the KSUA method, the students could conduct SUA even faster and more consistently than the expert architects, who had 13.2 year experience and did not use the KSUA method.

The power of the KSUA method for expert architects could further be claimed if I could measure their performance with the KSUA method in the future. In addition, with a higher number of participants, I could gather statistically more significant numbers as evidence of improved speed for updating and consistency in predictions by the KSUA method.

8. Conclusions

This section concludes my work during the PhD by describing theoretical contribution I made to the field of performance-based building (Section 8.1), practical significance (Section 8.2), and future research directions that would maximize the potential of this research (Section 8.3).

8.1. Theoretical contributions

Performance-based building theory requires architects to test multiple design options based on the performance criteria in a quick and consistent way and to refine the design to best meet their design intention (Becker 2008). Although space utilization can function as a performance criterion that determines whether or not each space will be properly used, a method that enables architects to be informed about space utilization quickly and consistently is missing. Without such a method, architects cannot systematically monitor and enhance the design intention regarding space-use during project development.

I contribute the KSUA method, which automatically predicts space utilization given user activity and space information, to the field of performance-based building. Prototypical implementation on the Y2E2 Building, Stanford University shows that the KSUA method leads to quick updating of utilization with high consistency in SUA (Section 6.2). In addition, the charrette test data on two cases show that novice architects can update predictions about space utilization 6.5 times faster with the KSUA method than with manual SUA, and do so much more consistently (the standard deviation of predictions across the novice architects was 68% less with the KSUA method). The test data also show that novice architects with the KSUA method outperform expert architects without the method (Section 7).

Based on the evidence, I claim the underlying two theories as specific contributions enabling the KSUA method:

- *An ontology for representing user activities for use in SUA*: this ontology consists of 10 classes (four concepts and their subclasses) and 24 properties and their facets. Based on this ontology, architects can represent a user activity in a <User>, an <Action>, and two <Spatial requirements> instances. This ontology is novel because it allows a computer to distinguish between different space-use types that affect SUA, whereas existing theories of representing user or construction activities (Akinci et al. 2002a; Darwiche et al. 1988; Pennanen 2004) require architects to keep track of the space-use type manually during the analysis. I validated the proposed ontology by representing 28 user activities identified in three case studies and by determining whether or not it can capture all SUTDs, from which 288 space-use types are derived (Section 4).
- *A method for mapping user activities onto appropriate spaces*: this method consists of the "choose spatial requirements," "find spaces," and "map the activity onto spaces" steps. This method can immediately inform architects about the impact of changes in the space program and user profiles on the activity-space pairs. This method is novel because it formalizes the knowledge about how to deal with different space-use types that affect SUA during

mapping, whereas existing user presence models (Ioannidis et al. 2012; Pennanen 2004; Shen et al. 2012; Tabak 2008; Yan and Kalay 2006; Zimmermann 2007) lack this knowledge and, therefore, require architects to map activities onto spaces manually. I validated the proposed method by measuring the conformity of the activity-space pairs generated by the method with the pairs generated manually by five architects in two cases and by asking the architects to score the acceptability of the method (Section 5).

8.2. Practical significance

Decisions about space-use may always be subjective because they heavily depend on the architects' holistic approach to a building. However, having a formal method that incorporates related information into a SUA process (i.e., the KSUA method) is important because this method provides architects with a consistent means of assessing and comparing architects' decisions about space-use by combining the three perspectives of space-use, i.e., the space, user, and activity perspectives.

Thus, deployment of the KSUA method in project development should enable architects to achieve high utilization of buildings without rarely used or crowded spaces. Specifically, in the programming phase, architects can use the KSUA method for planning space-use of a building by iteratively refining the space program and intended usage of the building: if this method indicates that a certain space type will be used rarely during project development, architects can decrease the number of the space type or assign more user activities to the space type to increase the utilization of the space type. If this method indicates a certain space type will be crowded, architects can increase the number of the space type or prevent some user activities from occurring in the space type to reduce the utilization of the space type. When architects modify the space program due to other reasons than space-use, such as changes in project budget and regulations, this method informs architects about the impact of the modification on space-use.

In the design phase, architects can use the KSUA method for monitoring space-use of a building as the design evolves: if the number, area, or other features of a space type in the design differ from those specified in the space program, this method informs architects about the impact of the difference on space-use. Since SUA becomes automated, architects can check the impact of their architectural activities, such as adding a room and changing the area of the room immediately throughout the design phase.

The KSUA method should also be beneficial when adjustments to existing buildings are discussed. The adjustments of a building supported by the KSUA method includes (1) a renovation project, which changes the physical structure (space information) of the building, and (2) changes in the space usage policy, which do not change the physical structure of the building but change the intended usage of the building. After an effort of calibrating the inputs of the KSUA method, which is discussed in Section 8.3.1, architects or facility managers can develop multiple adjustment options and compare these options in terms of space-use.

As a result, buildings are more likely to have the minimum number of spaces that are needed for supporting user activities and the future growth plan of an organization that uses the buildings. Achieving the minimum amount of spaces in a building contributes economic and environmental sustainability because building cost and energy use generally grow when the size of the building increases, as described in Section 1.1.

8.3. Future research directions

During the PhD, I formalized the KSUA method that takes information about user activities and spaces as input and produces predicted utilization of each space as output. Future research directions include (1) measuring and enhancing the accuracy of the KSUA method, (2) reducing architects' burden of building the project-specific knowledge base, (3) supporting decision-making about improving space-use throughout the project, and (4) extending the KSUA method for use in other performance analyses and other building types.

8.3.1. Measuring and enhancing the accuracy of the KSUA method

Although the KSUA method enables architects to predict utilization quickly and consistently, it should do so with sufficient accuracy to be used in practice. Hence, I plan to acquire measured utilization data in several office and educational buildings using occupancy sensors (Dodier et al. 2006) or existing network infrastructure (Melfi et al. 2011). When the measured utilization data of a building are compared to the predicted utilization of the same building, the current accuracy level of the KSUA method can be measured. In an existing building, utilizations would be different for space instances of the same space type. Hence, I plan to measure the accuracy level by comparing predicted utilization of each space type and the average of measured utilization of space instances of the space type. Once the KSUA can predict space utilization at a space instance level (instead of at a space type level), I can then compare predicted utilization and measured utilization for each space instance to measure the accuracy level of the KSUA method.

To enhance the accuracy level, I intend to take my research further in the following three directions: First, I aim to improve the accuracy of SUA by exploring and formalizing more knowledge about space-use, including time-varying and stochastic features of user activities (e.g., fluctuating numbers of users over a time period, randomly and periodically, and the duration of a user activity with a probability distribution) and information about the geometry, location, and aesthetics of spaces. This formalization would reduce the assumptions and limitations, made during the limited scope of the programming phase for educational and office buildings, described in Section 4.4 and Section 5.5. For example, the ontology I developed can be extended to allow architects to represent a user activity to have multiple actions. To make this extension, I would have to explore different types of relationship between the actions (e.g., sequential or parallel) and how these types affect space-use and its analysis. Functional steps of the KSUA method (Section 6.1) should also be further defined to reduce these assumptions and limitations. For example, the “mapping user activities onto spaces” step needs to be elaborated because spatial requirements of a user activity must be entirely satisfied by a space to trigger the mapping in the current version.

Second, the accuracy of SUA can also be enhanced by taking into account individual users and their activities, e.g., start and end time of an activity of an individual user, and the selection of a space by an individual user for performing his or her activities. To do so, activity scheduling models (Tabak 2008; Zimmermann 2007), which generate a sequence of activities for individuals, and pedestrian behavior models (Helbing et al. 2001; Willis et al. 2004; Yan and Kalay 2006), which simulate a movement of individuals as agents, should be integrated into the KSUA method.

Third, I want to develop a method for calibrating inputs of the KSUA method so that measured utilization matches predicted utilization before testing different building renovation plans in existing buildings. The first step to do so would be documentation of approximations, assumptions, and simplifications (Maile et al. 2010) that generate the discrepancy between the two utilizations. For example, the approximation of the duration and the frequency of an activity in user profiles might cause a discrepancy. The assumption that activity load is split equally into spaces that satisfy the spatial requirements might also contribute to a discrepancy. For example, when activity A is accommodated by space A and space B, and activity B is accommodated by only space B, users might prefer using space A for performing activity A to avoid the congestion in space B. In this case, activity A is not split equally into space A and space B although these two spaces satisfy the spatial requirements of activity A. Finally, the simplification of representing different features of space types as the same feature, e.g., a touch screen monitor, a projector system, and a large screen for presentation, all of which are represented as “screenProvided,” can also create a discrepancy. This documentation must be conducted after observations and measurements of space-use for a certain period that covers a cycle of building operations. For example, the space-use of Stanford University’s Y2E2 Building needs to be observed and measured for at least a quarter (i.e., three months) since Stanford has an academic quarter system. These observations and measurements might cost hiring a surveyor or installing occupancy sensors for the period.

8.3.2. Reducing architects' burden of building the knowledge base

To use the KSUA method in practice, architects must provide the information about user activities and spaces required by the method. To instantiate one user activity, architects must determine whether or not the activity is typical. If the activity is deemed a typical activity, architects must further determine the ratio and the frequency of the activity. Architects must also decide on the type of users (important or regular), the type of constraints and preferences (whole room use requirement or equipment use requirement), and many properties, such as the number of users, group size, duration, and space criteria, as described in Section 4.2. Similarly, to instantiate one space, architects must enter the area, number, open hours, super type, inaccessible users, and other features of the space into the KSUA method, as described in Section 6.1.1. Since there are many user activities and spaces to deal with, as described in Section 1.4.1, this burden of building the knowledge base can hinder architects from using the KSUA method in practice.

Therefore, I aim to connect the KSUA method to other computational models to reduce architects' burden of building the project-specific knowledge base, which is the first step of the KSUA method. A Building Information Model (BIM), for example, provides a computational representation of a design, and therefore space information stored in the BIM as properties, such as space type and area, can feed seamlessly into the KSUA method. Thus, I envision SUA software as an add-on to BIM tools, such as Revit Architecture. When architects enter user profiles into this add-on software and click the button “predict utilization,” the software would predict utilization of each space by taking space information from the BIM and activity information from the user profiles. If architects predetermine the policy on utilization of the spaces, the software would report which spaces will be rarely used, properly used, or crowded by comparing the policy on utilization and the predicted utilization. If a method that calibrates inputs of the KSUA method is developed, as described in Section 8.3.1, the software would also calibrate the inputs based on the measured utilization and the predicted utilization for existing buildings.

In tandem with the BIM, I plan to use a declarative spatial reasoning framework (Bhatt et al. 2011, 2012), which formalizes design artifacts and design requirements on three different levels (i.e., conceptual or high-level abstraction, qualitative, and quantitative), to allow architects to formalize the knowledge base as a high-level abstraction and to allow computers to reason about the knowledge base on a quantitative level, which is connected to the BIM. For example, when architects declare that an activity occurs in a space that is “close to teachers’ offices” in a high school, the KSUA method should be able to find the spaces in the BIM without human intervention. If architects move the teachers’ office in the BIM during design, the KSUA method should be able to change the design artifacts that are “close to teachers’ offices” to support SUA.

In addition, developing a database of the knowledge base that is sortable by various factors, such as project types and regions where projects are conducted, can contribute to reducing the architects’ effort of providing the KSUA method with information about user activities when it is difficult to develop project-specific user profiles or when architects want to be informed about space-use even before they develop project-specific user profiles. To do so, user profiles in existing buildings should be developed, represented using the ontology described in Section 4, and stored in a database. Similarities and differences in user profiles stored in a database should be analyzed to categorize the project types, regions, and other important factors in terms of space-use.

8.3.3. Supporting decision-making about improving space-use

I would like to develop methods that support the decision-making of architects and facility managers about improving space-use throughout the project. My PhD work can support such decision-making by predicting and visualizing utilization quickly and consistently. I plan to further support the decision-making in the following ways: (1) optimizing the numbers of space types in the space program, (2) developing a method for controlling the space supply based on information about user activities, and (3) providing additional metrics about space-use.

First, once all activity-space pairs are identified given information about activities and space types, as described in Chapter 5, the numbers of space types (i.e., decision variables) can be optimized to satisfy certain project goals (i.e., goal variables). I aim to define the goal variables (e.g., minimizing the average utilization of space types or minimizing the largest utilization value among space types) and constraints (e.g., the range of utilization that is allowed or the maximum total square footage of a project) in support of this optimization. Based on optimization theory and these definitions, I would like to develop a tool that would inform architects about the optimum number of each space type whenever the architects change user profiles or the space program during project development.

Second, I aim to develop a method that would enable facility managers to control the space supply in an existing building based on information about user activities. For example, if predicted utilization of three meeting rooms in a company is 0.33 for a certain period of time (e.g., for the next week), the company can close one meeting room for the period. Such a change would increase the utilization to 0.50, but these meeting rooms are still at the “no wait” level according to Table 1. If this type of information can be provided for all space types, the company could close a zone or a whole floor without affecting user activities for the period. The developed method should also be able to show impacts of various ways of controlling the space supply (e.g., closing spaces, making spaces inaccessible to certain user groups, allocating specific user activities to certain spaces) on space-use. Therefore, I plan to formalize various ways of controlling the space supply and develop an algorithm that determines the best space supply mode(s) for a certain period of time, e.g., a space supply mode that minimizes the number of used zones without affecting user activities.

Third, adding additional quantitative metrics about space-use other than utilization can help architects determine the goodness of a space program or a design in terms of space-use. For example, it would be helpful to inform architects about the number of preferences of regular users that are violated in a specific scenario (a set of user activities and space types). To do so, for all activity-space pairs generated by the KSUA method, a computer must determine whether or not a space type that accommodates a user activity of regular users satisfies the preferences of the activity.

If the space type does not satisfy the preferences, the computer must increase "the number of violated preferences" by one and record the violation issues. For example, in a trial run described in Section 6.2, the KSUA method generated 26 activity-space pairs, all of which were performed by regular users. Among the pairs, the "classroom" space type violates the preferences of the "graduates having classes" activity because the preferences require the space type to have a screen ("screenProvided"). Similarly, the "conference room with a computer" space type violates the preferences of the "faculty meeting" activity because the preferences require the space type to be equal to or larger than 37 m^2 while the area of the space type is 27 m^2. The number of violated preferences in the trial run is 15 out of 26 activity-space pairs.

8.3.4. Extending the KSUA method for use in other performance analyses and other building types

Another future research direction is to extend the KSUA method for use in other performance analyses in various areas. Space-use interacts with other building performance areas, such as energy use, indoor air quality, users' work productivity, and profitability. For example, an occupation of a space by users for an activity affects energy use and the indoor air quality of the space. When the indoor air quality declines, it harms work productivity of the users, and users leave the space to perform the activity in another space. Hence, these performance areas should be brought together to enable architects to assess their design more accurately, systematically, and holistically in light of these dynamics.

One research should be linking the KSUA method with energy simulation tools, such as Riuska (http://www.granlund.fi/en/software/riuska/) and EnergyPlus (http://apps1.eere.energy.gov/buildings/energyplus/energyplus_about.cfm). These tools predict energy use based on design information, generally represented in a BIM, and user presence information, generally represented in diversity profiles (Bourgeois et al. 2006; Mahdavi and Pröglhöf 2009). Although diversity profiles are widely used because of their simplicity, they do not provide project-specific information about user presence. Therefore, predicted energy use of buildings can vary "by more than 150% from the lower to the higher values established by experts as representative of typical user behavior" (Haymaker and Clevenger 2006). Thus, to enhance the project-specificity of the energy simulation, many researchers have modeled an activity schedule and integrated it with the design information to provide project-specific user presence information over time (Ioannidis et al. 2012; Shen et al. 2012; Tabak 2008). Researchers have also modeled user control behavior, such as lighting and window control, in interaction with buildings (Bourgeois et al. 2006; Mahdavi et al. 2008; Yu et al. 2011; Yun and Steemers 2008). Although these two efforts contribute greatly to energy simulation, architects still have difficulties in updating user presence when they change the design or activity schedules because they still have to manually map user activities onto spaces. I envision an energy simulation tool that is integrated with an automated activity-space mapping method. This integration would enable architects to quickly see the impact on energy use whenever they change the design and the activity schedule.

Similarly, "the degree of user satisfaction" or "the congestion index" could also be computed and provided to architects and other stakeholders to support their decision-making during project development. To do so, these concepts (user satisfaction and congestion) should be defined in the context of space-use, and their relationships with space utilization should be explored and formalized.

In addition, I aim to extend the KSUA method to other types of buildings where users select spaces for their activities autonomously, and relations between user activities and spaces are complex, such as hospitals, religious buildings, and transportation buildings. To do so, characteristics of user activities in these types of buildings in terms of SUA must be further explored and formalized for the KSUA method. I also plan to define user profiles of these types of buildings in support of architects' developing user profiles for SUA. There are some building types, however, for which the KSUA method seems less useful. For example, in industrial buildings, such as plants and factories, employees' movements can be planned rather than predicted, which is similar to the situation of construction space planning described in Section 2.2. Thus, plant layout and material-handling theories (Meyers 1993; Levary and Kalchik 1985) might be more effective for planning industrial buildings than the KSUA method. The KSUA method does also not provide much benefit to residential buildings, such as apartments and dormitories, because there are not many multi-multi user-space relationships that affect space usage negatively. The KSUA method seems also less applicable for planning open and flexible office space, where space configurations continuously change as employees participate in various projects over the course of a year.

8.4. Summary of the conclusion

During the PhD, I contributed the KSUA method, which automatically predicts space utilization given user activity and space information, to the field of performance-based building design. To enable this method, I developed an ontology for representing user activities for use in SUA and formalized a method for mapping user activities onto appropriate spaces. As a result, by integrating the space, user, and activity perspectives, the KSUA method enables architects to modify one perspective and be informed about the impact of the modification on the other two perspectives, quickly and consistently. Specifically, architects can modify the space perspective by changing the space program and policy on utilization of each space in the space program. Architects can modify the user perspective by changing the type of users (important or regular) and the number of users. Most importantly, architects can modify the activity perspectives by changing space-use type and many values, such as the frequency and duration of activities. The KSUA method can track and analyze all of these changes quickly and consistently in support of architects' iterative refinement of spaces and their intended usage.

For this research, I focused only on office and educational buildings and limited the scope to the programming phase of a project. There are also many assumptions I made during the research, as described in Sections 4.4 and 5.5. To address these limitations and assumptions and to achieve the full potential of automated SUA, I defined the following three future research directions: first, I plan to measure the accuracy level of the KSUA method and continuously enhance the accuracy level. To do so, I aim to explore and formalize more knowledge about space-use, model individual users and their activities, and develop a method for calibrating the inputs to the KSUA method. Second, I plan to reduce architects' burden of building the knowledge base, which might hinder the deployment of the method. To do so, I aim to connect the KSUA method to other computational models, such as BIM and a declarative spatial reasoning framework, and develop a database of the knowledge base that supports the formalization of user activities for the KSUA method. Third, I plan to further support the decision-making of architects and facility managers about improving space-use throughout the project. Specifically, I aim to optimize the numbers of space types in the space program, develop a method for controlling the space supply based on information about user activities, and provide additional metrics about space-use other than utilization. Fourth, I plan to extend the KSUA method for use in other performance analyses because many building performance areas, such as energy use and indoor air quality, interact with space-use. One research I aim to perform in this direction is to connect the KSUA method with energy simulation tools so that architects quickly see the impact on energy use whenever they change the design and the activity schedule. I also plan to extend the KSUA method to other types of buildings, such as hospitals, religious buildings, and transportation buildings.

This research and extensions of the research in aforementioned directions could make buildings more economically and environmentally sustainable without harming the performance of user activities. The KSUA method could be used to manage space-use of buildings throughout a building's lifecycle—it can be used for planning space-use of a building in the programming phase, for monitoring space-use as a design evolves in the design phase, and for improving space-use as information about actual use becomes available in the occupancy phase.

Appendix A. Glossary

Activity load: Hours that an activity demands from spaces (Pennanen 2004).

Activity-space pair: A pair of a user activity and a space that accommodates this activity; it can be generated by the activity-space mapping method, which are represented in a set of rules in knowledge-based space-use analysis.

Activity-space mapping method: A representation of the knowledge about how to map user activities onto appropriate spaces for use in space-use analysis (SUA) during project development; it eliminates human efforts to speculate on all possible pairs between user activities and spaces during SUA.

Assumed usage: A set of assumptions about the usage of spaces in a building, which function as external factors that architects cannot change during project development; it includes the expected user groups, their numbers and expected activities, and generally acceptable spatial requirements of these activities.

Charrette test: A validation method by which the effectiveness of a computing method can be tested; participants perform a task without the method a researcher developed (i.e., the control group, or the conventional process) and with the method (i.e., the experimental group, or the innovative process), for each trial. Although the charrette test has limitations regarding the impact of familiarity and training, measurement instruments, and arbitrariness of the processes, it can increase both reliability and validity of research conclusions over other validation methods (e.g., worked example, demonstration, trial) that are commonly used for computing method research projects (Clayton et al. 1998).

Crowded space: A space that is occupied by building users more frequently than the policy on utilization of this space specifies.

Event quantity: The number of user groups for a given activity; it is calculated by dividing the number of users by the group size of the activity, which is the number of people that the activity requires to have.

Facts: "A set of raw observations, symbols, or statement" (Akerkar and Sajja 2010), represented in formal languages so that a knowledge-based system (KBS) can interpret them for itself.

Guidelines approach: A method in which architects pick parameters that are related to the area or the number of a space type and values for these parameters to define the area or the number in a building; the values are usually based on personal experience, surveys, analytical techniques, or well-established standards. It is also called as a "choose-and-use method" (Erhan 2003).

Intended usage: The space usage policy that reflects a client's or an architect's intention about how spaces should be used in a building. Although both intended usage and assumed usage are important sources of user profiles, intended usage functions as a set of parameters that is changeable during project development. In contrast, assumed usage is not changeable during project development. Examples of intended usage include keeping a user group from using a space and allowing a user group to use spaces that satisfy the group's preferences for the group's activities.

Knowledge-based space-use analysis (KSUA) method: Space-use analysis method that is built based on a knowledge-based systems approach; A KSUA system has the following four components: the knowledge base, inference engine, working memory, and interface. The knowledge base contains (1) the project-specific information about user activities and spaces, i.e., information to be processed, (2) project-independent domain knowledge about space-use analysis, including how to map user activities onto spaces and how to compute utilization of each space, and (3) ontology (meta-knowledge), which provides a scheme in which project-specific information and the domain knowledge is represented in forms of facts and rules.

Knowledge-based system (KBS): A system that uses one or more experts' knowledge to solve problems in a specific domain; "it can understand the information being processed and make a decision based on it, whereas traditional computer systems do not know or understand the information they process" (Akerkar and Sajja 2010).

Measured utilization: Space utilization that is measured by observation, survey, or occupancy sensors in the occupancy phase.

Model-driven (engineering): "A software development methodology which focuses on creating and exploiting domain models (i.e., abstract representation of the knowledge and activities that govern a particular application domain) rather than on the algorithmic concepts" (Wikipedia, as of Feb. 8, 2013); according to model-driven engineering, "a model is considered effective when the model makes sense from the perspective of users that are familiar with the domain, and when the model can serve as a basis for implementing software systems" (Wikipedia, as of Feb. 8, 2013).

Ontology: "A scheme in which project-specific information and domain knowledge about space-use analysis is represented" (Akerkar and Sajja 2010); ontology enables humans to clearly express project-specific information and enables computer models to interpret the information for their own purposes (Wang et al. 2011).

Performance-based building (PBB): “A building market environment in which all stakeholders involved in the building process recognize the need to ensure long-term performance-in-use of buildings as an explicit target” (Becker 2008); to achieve performance-based building design, the following concepts must be defined: user needs, performance requirements and criteria, generalized loads, and performance assessment method.

Policy on utilization: Utilization of a space that is deemed to be properly used; a client often determines the policy on utilization (or target utilization) taking into account its business strategy and expected future growth (Cherry 1999; Pennanen 2004). In knowledge-based space-use analysis, the policy on utilization of a space has a utilization implication level as its value.

Predicted utilization: Space utilization that is predicted by space-use analysis during project development.

Programming phase: The second phase of the facility production process, positioned after strategic planning and before design (Preiser and Vischer 2005); some researchers (Hansen and Vanegas 2003; Kamara et al. 1999) see programming as an overlapping work process throughout the strategic planning phase and the early design phase. In the programming phase, architects develop user profiles and the space program.

Project development: Programming and design phases where architects develop a space program or a design that satisfies user profiles and other project requirements.

Properly used space: A space that is occupied by building users with a desired frequency, which is represented by the policy on utilization.

Rarely used space: A space that is occupied by building users less frequently than the policy on utilization of this space specifies.

Relation schema: A relation name and a list of its attributes; it describes the structure of data or information that is shown in a two-dimensional table called a relation (Ullman and Widom 2007).

Rules (production rules): "A two-part structure comprising an antecedent set of conditions and a consequent set of actions" (Brachman and Levesque 2004); the general form of rules is IF <conditions> AND <conditions> THEN <actions>.

Space program: A document that lists the number and the area of each space type. Other features that need to be fulfilled by a design, including adjacency, specific equipment, and location, can also be included in a text format under the name of "other," "etc.," or "note."

Space-use: The assessment regarding how a space in a building will be used; although this term contains various meanings, in this research, it only concerns how much a space will be used.

Space-use analysis (SUA): A method that predicts how much each space in a building will be used, i.e., utilization, to inform architects about which space will be rarely used, properly used, or crowded. It takes user profiles and a space program or design as input to produce predicted utilization of each space and its implication level as output. The prediction includes initial prediction of utilization and updating of utilization when input of space-use analysis changes.

Space type: A classification of spaces (e.g., office, meeting room, storage room) that have the same required area, super-type, and other features, which can be captured in a space program; Kiviniemi (2005) calls space type “space program instance (SPI)” to distinguish this, which is used in a requirements model, from space instance (e.g., meeting room A204, A205), which is used in a design model.

Space-use type differentiator (SUTD): An entity that distinguishes between different space-use types of user activity in terms of space-use analysis; a space-use type is explained as a set of choices for each SUTD. There are 288 space-use types that architects must distinguish and treat differently during SUA.

Space utilization: Utilization and space utilization are interchangeable in this research unless other definitions are provided.

Spatial requirements: Properties of a space that a user activity requires for occupying the space. A user activity is mapped onto a space when all spatial requirements of the activity are met by the features of the space. It is different from “space requirements” used in Kiviniemi’s (2005) requirements model, which defines requirements of a space that must be met by the design based on implicit consideration of user activities in the space.

Total activity loads (or total loads): Time amount that refers to how long a space is used by user activities; for example, if a space is used for two hours for activity A and two hours for activity B, the total load of this space is four hours.

User profiles: A document that describes the types of users, the number of users, what they do (i.e., their activities), and their functional needs (i.e., spatial requirements) (Cherry 1999). User profiles include assumed usage and intended usage of a building.

Utilization: "The usage rate of a space" (Pennanen 2004); for example, 50% (or 0.50) utilization means that the space is used for four hours in an eight-hour day.

Utilization implication level: The distinction of what a certain range of utilization implies.

Appendix B. Expert survey for defining space-use types

I conducted an expert survey through an online survey website called SurveyMonkey (http://www.surveymonkey.com/) to identify important characteristics of user activities (Section 3.1). These characteristics describe the contexts where different space-use types need to be treated differently during SUA.

Appendix B1: Survey questionnaires

Introduction and personal information

Hello, my name is Tae Wan, and I am a Ph.D. student at Stanford University. I am developing a tool that automatically maps user activities onto spaces in a building to predict space-use of the building by its users (how much each space will be used by user activities).

Here I need your kind help to find important characteristics of user activities regarding the prediction of space-use. This survey has ONLY 10 QUESTIONS. All information gathered will be used only in academic purposes.

If you have any questions or comments on my research, please contact me at tkim08@stanford.edu. Thank you very much for your precious time and effort in responding to this survey.

Best Regards,

Tae Wan Kim

1. About how many years have you been working in the construction industry?

____

2. Of these years in the construction industry, about how many years have you been working as a planner/architect/construction manager?

____

3. Do you agree that predicting/analyzing space-use by building users (i.e., how much each space will be used by building users) is important to a building planning/design?

( ) 1: Strongly disagree

( ) 2: Disagree

( ) 3: Neither agree nor disagree

( ) 4: Agree

( ) 5: Strongly agree

4. Please describe your method of predicting the space-use by users when you are planning/designing a building or reviewing someone's plan/design.

____

Importance of identified characteristics of user activities

I have identified these characteristics of user activities based on literature review, case observations, and interviews with architects. Please help me validate these characteristics. How important do you think each of these characteristics is when it comes to predicting space-use of a building?

5. Do you agree that in a building, "some users require having a better space than minimum requirements for their activities, (e.g., a president of a company wants to have a bigger and quieter room than the minimum condition for his work)?”

( ) 1: Strongly disagree

( ) 2: Disagree

( ) 3: Neither agree nor disagree

( ) 4: Agree

( ) 5: Strongly agree

6. Do you agree that in a building, "some activities require having a designated space, e.g., a professor's office?"

( ) 1: Strongly disagree

( ) 2: Disagree

( ) 3: Neither agree nor disagree

( ) 4: Agree

( ) 5: Strongly agree

7. Do you agree that in a building, "some activities require occupying the whole room (e.g., a meeting activity requires a whole conference room) while others need part of a room (e.g., regular work of an employee requires occupation of only one workstation in the office space)?”

( ) 1: Strongly disagree

( ) 2: Disagree

( ) 3: Neither agree nor disagree

( ) 4: Agree

( ) 5: Strongly agree

8. Do you agree that in a building, "some activities are conducted in a specifically named space (e.g., main auditorium) while others are conducted in any spaces with certain conditions (e.g., a room with a table for more than 6 people)?"

( ) 1: Strongly disagree

( ) 2: Disagree

( ) 3: Neither agree nor disagree

( ) 4: Agree

( ) 5: Strongly agree

9. Do you agree that in a building, "some activities are atypical activities that are not conducted on a regular basis, e.g., commencement?"

( ) 1: Strongly disagree

( ) 2: Disagree

( ) 3: Neither agree nor disagree

( ) 4: Agree

( ) 5: Strongly agree

10. (Optional) Do you see any other characteristics of user activities that might impact users' space-use (space selection, space requirements, etc.)? Please name them if you have any.

____

Appendix B2: Answers from experts

Personal information

Expert ID	Industry experience (yr.)	SUA experience (yr.)	Importance of SUA (1-5)	Current practice
1	40	24	5	Starting with user interviews (their needs, views, expectations, etc.) and analyzing their current spaces and usage, then synthesizing the findings a spatial program.
2	23	23	4	Using a design experience and other examples.
3	10	10	4	Reviewing plans and checking furniture and equipment plans, location of doors, windows, etc.
4	15	9	5	Estimating the number of users who are most likely to use the space and their average usage time for each space.
5	6	6	5	
6	6	6	5	
7	12	6	4	Understanding client expectations for particular departments or areas. Referencing graphic standards and other hallmark documents, which have predictive/prescriptive information. In early concept design we rarely are considering a specific number of persons.
8	8	6	4	
9	7	6	5	Using minimum space requirement for each use, design guidelines, and industry standards.
10	5	5	4	
11	5	5	5	Using Autodesk Revit Architecture.

12	5	5	3
13	8	4	5
14	4	4	5
Average	11.0	8.5	4.5

Importance of identified characteristics of user activities

Expert ID	C1 (question 5)	C2 (question 6)	C3 (question 7)	C4 (question 8)	C5 (question 9)
1	4	5	5	5	5
2	-	-	-	-	-
3	4	4	5	5	5
4	2	5	5	5	5
5	-	-	-	-	-
6	4	4	5	4	4
7	4	4	3	4	4
8	-	-	-	-	-
9	5	5	5	4	5
10	4	4	4	4	5
11	4	4	4	4	4
12	3	4	4	4	4
13	5	5	5	5	5
14	5	5	5	5	5
Average	4.0	4.5	4.5	4.5	4.6

Other characteristics of user activities (question 10):

- Expert 3: “Building spaces are also greatly affected by the equipment they need to have to serve their purpose (e.g. a CT scan room). Locations and swings of doors and windows also affect building space use (e.g. you wouldn't be able to put a desk by a door that opens towards it, or a cabinet that blocks a window).”

- Expert 4: “1. Training activities, specifically hands-on training, require special room with training tools (e.g., computers). 2. Some activities (e.g., design coordination using VDC) require room with multimedia settings. 3. Organizational culture of a company has strong impact on space requirements. I have seen that an organization with strong hierarchical structure requires allocation of bigger and quieter space for high people. Conversely, an organization with flat structure, where there is no official title within the company and every employee is considered as equally important, does not require special space for the people with more important responsibilities.”
- Expert 7: “Working with hazardous chemicals or materials (ventilation, adequate exit and life safety equipment required). Heavy computer user (ability to control lighting and to avoid direct light/sunlight)”

Appendix C. Ontology for representing project information in knowledge-based space-use analysis

This schema is written in F-Logic.

Entity::Thing.

Event::Thing.

User::Entity [number {1:1} *=> integer].

ImportantUser::User.

RegularUser::User.

Action::Event [groupSize {1:1} *=> integer, duration {1:1} *=> float, spaceCriteria {1:1} *=> float].

SpatialReq::Entity[spaceType *=> Space, superType *=> SpaceSuperType, designation {1:1} *=> boolean].

WholeRoomUseReq::SpatialReq[number {0:1} *=> integer, minArea {0:1} *=> float, otherFeatures *=> Features].

EquipmentUseReq::SpatialReq[equipment *=> Equipment, number {0:1} *=> integer, minArea {0:1} *=> float, otherFeatures *=> Features].

Activity::Event[user {1:1} *=> User, action {1:1} *=> Action, constraints {1:1} *=> SpatialReq, preferences {1:1} *=> SpatialReq].

TypicalActivity::Activity[ratio {1:1} *=> float, frequency {1:1} *=> float].

AtypicalActivity::Activity.

Appendix C. Ontology for representing project information in knowledge-based space-use analysis

Space::Entity[area {1:1} *=> float, number {1:1} *=> integer, features *=> Features, openHours {1:1} *=> float, superType {0:1} *=> SpaceSuperType, inaccessible *=> User].

OccupiableSpace::Space[designated {1:1} *=> boolean].

NonOccupiableSpace::Space[equipped *=> EquipmentSet].

Equipment::Entity[area {1:1} *=> float].

EquipmentSet::Entity[equipment {1:1} *=> Equipment, number {1:1} *=> integer, designated {1:1} *=> boolean, features *=> Features, openHours {1:1} *=> float, inaccessible *=> User].

Features::Entity.

SpaceSuperType::Entity.

Status::Entity.

noWait:Status.

adequate:Status.

inconvenient:Status.

infeasible:Status.

anySpace:Space.

noType:SpaceSuperType.

noFeature:Features.

noOne:User.

Appendix D. Represented project information in case studies

D1, D2, and D3 were used in validating the developed ontology, D1 was used in the prototypical implementation, D2 and D4 were used in the charrette test; therefore, D1, D2, and D4 have both user activities and spaces, and D3 has only user activities. The information is written in F-Logic.

Appendix D1: Represented project information in the Y2E2 Building case

// Users

undergrads:RegularUser[number -> 25].

grads:RegularUser[number -> 122].

faculty:RegularUser[number -> 5].

researchers:RegularUser[number -> 12]. // visiting scholars and RAs

staff:RegularUser[number -> 6]. // administrative and academic staff

// Activity 1-1, 1-2: students' having a class

gradsHavingClass:TypicalActivity[user -> grads, action -> haveClass, ratio -> 0.7, frequency -> 0.2, constraints -> cons1, preferences -> pref1].

undergradsHavingClass:TypicalActivity[user -> undergrads, action -> haveClass, ratio -> 1.0, frequency -> 0.4, constraints -> cons1, preferences -> pref1].

haveClass:Action[groupSize -> 20, duration -> 2.0, spaceCriteria -> 1.7].

cons1:WholeRoomUseReq[spaceType -> anySpace, superType -> noType, minArea -> 37, otherFeatures -> noFeature, designation -> False].

pref1:WholeRoomUseReq[spaceType -> anySpace, superType -> noType, minArea -> 37, otherFeatures -> screenProvided, designation -> False].
screenProvided:Features.

// Activity 2-1, 2-2: students' having a group meeting for a class with computers
gradsMeetingForClassWithComputers:TypicalActivity[user -> grads, action -> haveMeetingWithComputers, ratio -> 0.7, frequency -> 0.4, constraints -> cons2, preferences -> pref2].
undergradsMeetingForClassWithComputers:TypicalActivity[user -> undergrads, action -> haveMeetingWithComputers, ratio -> 1.0, frequency -> 0.2, constraints -> cons2, preferences -> pref2].
haveMeetingWithComputers:Action[groupSize -> 3, duration -> 3, spaceCriteria -> 2.4].
cons2:EquipmentUseReq[spaceType -> computerCluster, equipment -> computer, number -> 1, designation -> False].
pref2:WholeRoomUseReq[spaceType -> anySpace, superType -> noType, minArea -> 14, otherFeatures -> computerProvided, designation -> False].
computerProvided:Features.

// Activity 3-1, 3-2: students' having a group meeting for a class without computers
gradsMeetingForClass:TypicalActivity[user -> grads, action -> haveMeeting, ratio -> 0.7, frequency -> 0.2, constraints -> cons3, preferences -> pref3].

undergradsMeetingForClass:TypicalActivity[user -> undergrads, action -> haveMeeting, ratio -> 1.0, frequency -> 0.4, constraints -> cons3, preferences -> pref3].

haveMeeting:Action[groupSize -> 3, duration -> 1.3, spaceCriteria -> 2.4].

cons3:WholeRoomUseReq[spaceType -> anySpace, superType -> noType, minArea -> 14, otherFeatures -> noFeature, designation -> False].

pref3:WholeRoomUseReq[spaceType -> anySpace, superType -> noType, minArea -> 14, otherFeatures -> computerProvided, designation -> False].

// Activity 4: graduate students' having a group meeting for their research

groupMeetingForResearch:TypicalActivity[user -> grads, action -> haveResearchMeeting, ratio -> 1.0, frequency -> 0.1, constraints -> cons4, preferences -> pref4].

haveResearchMeeting:Action[groupSize -> 3, duration -> 2, spaceCriteria -> 2.4].

cons4:WholeRoomUseReq[spaceType -> anySpace, superType -> conferenceRoom, minArea -> 14, otherFeatures -> noFeature, designation -> False].

pref4:WholeRoomUseReq[spaceType -> anySpace, superType -> conferenceRoom, minArea -> 17, otherFeatures -> computerProvided, designation -> False].

// Activity 5-1, 5-2: students' individual study

gradsStudying:TypicalActivity[user -> grads, action -> study, ratio -> 0.5, frequency -> 0.25, constraints -> cons5, preferences -> pref5].

undergradsStudying:TypicalActivity[user -> undergrads, action -> study, ratio -> 0.5, frequency -> 0.2, constraints -> cons5, preferences -> pref5].

study:Action[groupSize -> 1, duration -> 1.0, spaceCriteria -> 2.4].

cons5:WholeRoomUseReq[spaceType -> anySpace, superType -> conferenceRoom, minArea -> 4.6, otherFeatures -> quiet, designation -> False].

//cons5b:WholeRoomUseReq[spaceType -> anySpace, superType -> conferenceRoom, minArea -> 4.6, otherFeatures -> noFeature, designation -> False].

pref5:WholeRoomUseReq[spaceType -> anySpace, superType -> conferenceRoom, minArea -> 4.6, otherFeatures -> quiet, designation -> False].

quiet:Features.

// Activity 6: faculty's regular work

facultyWorking:TypicalActivity[user -> faculty, action -> doRegularWork, ratio -> 1.0, frequency -> 0.8, constraints -> cons6, preferences -> pref6].

doRegularWork:Action[groupSize -> 1, duration -> 4, spaceCriteria -> 15].

cons6:WholeRoomUseReq[spaceType -> privateOffice, minArea -> 15, otherFeatures -> noFeature, designation -> True].

pref6:WholeRoomUseReq[spaceType -> privateOffice, minArea -> 15, otherFeatures -> noFeature, designation -> True].

// Activity 7: staff's regular work

staffWorking:TypicalActivity[user -> staff, action -> doRegularWork, ratio -> 1.0, frequency -> 1.0, constraints -> cons7, preferences -> pref7].

doRegularWork:Action[groupSize -> 1, duration -> 4, spaceCriteria -> 7.4].

cons7:EquipmentUseReq[spaceType -> sharedOffice, equipment -> workStation, designation -> True].

pref7:WholeRoomUseReq[spaceType -> privateOffice, minArea -> 15, otherFeatures -> noFeature, designation -> True].

// Activity 8: researchers' regular work

researchersWorking:TypicalActivity[user -> researchers, action -> doRegularResearch, ratio -> 1.0, frequency -> 1.0, constraints -> cons8, preferences -> pref8].

doRegularResearch:Action[groupSize -> 1, duration -> 4, spaceCriteria -> 7.4].

cons8:EquipmentUseReq[spaceType -> openOffice, equipment -> workStation, designation -> True].

pref8:EquipmentUseReq[spaceType -> sharedOffice, equipment -> workStation, designation -> True].

// Activity 9: faculty meeting

facultyMeeting:TypicalActivity[user -> faculty, action -> haveMeetingLong, ratio -> 1.0, frequency -> 0.15, constraints -> cons9, preferences -> pref9].

haveMeetingLong:Action[groupSize -> 5, duration -> 3, spaceCriteria -> 2.4].

cons9:WholeRoomUseReq[spaceType -> anySpace, superType -> conferenceRoom, minArea -> 18.5, conditions -> noFeature, designation -> False].

pref9:WholeRoomUseReq[spaceType -> anySpace, superType -> conferenceRoom, minArea -> 37, conditions -> noFeature, designation -> False].

Appendix D. Represented project information in case studies

// Spaces

computerCluster:NonOccupiableSpace[area -> 74, number -> 1, equipped -> computerSet, openHours -> 8.0]. // 292E

computerSet:EquipmentSet[equipment -> computer, number -> 25, openHours -> 8.0, designated -> False, inaccessible -> noOne].

computer:Equipment[area -> 1.9].

classRoom:OccupiableSpace[superType -> classRoom, area -> 74, number -> 1, openHours -> 8.0, designated -> False, inaccessible -> noOne]. // 292A

smallConferenceRoomWithComputer:OccupiableSpace[superType -> conferenceRoom, area -> 9, number -> 3, openHours -> 8.0, features -> {computerProvided, quiet}, designated -> False, inaccessible -> noOne]. // 292F, 292G, 292H

conferenceRoomWithComputer:OccupiableSpace[superType -> conferenceRoom, area -> 27, number -> 1, openHours -> 8.0, features -> computerProvided, designated -> False, inaccessible -> noOne]. // 292C

privateOffice:OccupiableSpace[area -> 17, number -> 5, openHours -> 8.0, designated -> False, inaccessible -> noOne]. // 297, 295, 293, 289, 287, 292B

sharedOffice:NonOccupiableSpace[area -> 15, number -> 3, equipped -> workStationOfTwo, openHours -> 8.0]. // 275B, 275A, 228

workStationOfTwo:EquipmentSet[equipment -> workStation, number -> 2, openHours -> 8.0, designated -> False, inaccessible -> noOne].

workStation:Equipment[area -> 5].

openOffice:NonOccupiableSpace[area -> 33, number -> 2, equipped -> workStationOfSix, openHours -> 8.0]. // 291, 274

workStationOfSix:EquipmentSet[equipment -> workStation, number -> 6, openHours -> 8.0, designated -> False, inaccessible -> noOne].

smallConferenceRoom:OccupiableSpace[superType -> conferenceRoom, area -> 15, number -> 2, openHours -> 8.0, features -> quiet, designated -> False, inaccessible -> noOne]. // 278A, 278B

largeConferenceRoom:OccupiableSpace[superType -> conferenceRoom, area -> 51, number -> 1, openHours -> 8.0, features -> screenProvided, designated -> False, inaccessible -> noOne]. // 270

conferenceRoom:SpaceSuperType.

classRoom: SpaceSuperType.

Appendix D2: Represented project information in the Cygnaeus High School case

// Users: teachers, students, graduating students, student association members

teachers:RegularUser[number -> 70].

students:RegularUser[number -> 650].

graduatingStudents:RegularUser[number -> 217].

studentAssociationMembers:RegularUser[number -> 10].

// Activity 1: teachers' regular work

teacherWorking:TypicalActivity[user -> teachers, action -> doRegularWork, ratio -> 1.0, frequency -> 1.0, constraints -> cons2, preferences -> pref2].

doRegularWork:Action[groupSize -> 1, duration -> 1.5, spaceCriteria -> 2.7].

cons2:EquipmentUseReq[spaceType -> office, equipment -> workstationForTeachers, minArea -> 3.0, otherFeatures -> noFeature, designation -> True].

pref2:EquipmentUseReq[spaceType -> office, equipment -> workstationForTeachers, minArea -> 4.5, otherFeatures -> noFeature, designation -> True].

// Activity 2: students' waiting for meeting with teachers

waitingForMeetingWithTeachers:TypicalActivity[user -> students, action -> waitingForMeeting, ratio -> 1.0, frequency -> 0.1, constraints -> cons3, preferences -> pref3].

waitingForMeeting:Action[groupSize -> 1, duration -> 0.5, spaceCriteria -> 1.3].

cons3:EquipmentUseReq[spaceType -> anySpace, equipment -> chair, minArea -> 2.0, otherFeatures -> closeToOffice, designation -> False].

pref3:EquipmentUseReq[spaceType -> anySpace, equipment -> chair, minArea -> 4.0, otherFeatures -> closeToOffice, designation -> False].

closeToOffice:Features.

// Activity 3: students meeting

studentsMeeting:TypicalActivity[user -> students, action -> normalMeeting, ratio -> 1.0, frequency -> 0.1, constraints -> cons4, preferences -> pref4].

normalMeeting:Action[groupSize -> 4, duration -> 1.0, spaceCriteria -> 3.0].

cons4:WholeRoomUseReq[spaceType -> anySpace, number -> 1, minArea -> 15.0, superType -> meetingRoom, otherFeatures -> noFeature, designation -> False].

pref4:WholeRoomUseReq[spaceType -> anySpace, number -> 1, minArea -> 20.0, superType -> meetingRoom, otherFeatures -> quiet, designation -> False].

quiet:Features.

// Activity 4: student association meeting

studentAssociationMeeting:TypicalActivity[user -> studentAssociationMembers, action -> longMeeting, ratio -> 1.0, frequency -> 0.2, constraints -> cons5, preferences -> pref5].

longMeeting:Action[groupSize -> 4, duration -> 2.0, spaceCriteria -> 3.0].

cons5:WholeRoomUseReq[spaceType -> anySpace, number -> 1, minArea -> 30.0, superType -> meetingRoom, otherFeatures -> noFeature, designation -> False].

pref5:WholeRoomUseReq[spaceType -> anySpace, number -> 1, minArea -> 30.0, superType -> meetingRoom, otherFeatures -> quiet, designation -> False].

// Activity 5: students' final exam before graduation – used in testing the developed ontology, but not used in the charrette test

finalExam:AtypicalActivity[user -> graduatingStudents, action -> doExam, constraints -> cons1, preferences -> pref1].

doExam:Action[groupSize -> 220, duration -> 2.0, spaceCriteria -> 1.0].

cons1:WholeRoomUseReq[spaceType -> auditorium, number -> 1, minArea -> 240.0, otherFeatures -> noFeature, designation -> False]. // original plan for accommoding "finalExam" activity

pref1:WholeRoomUseReq[spaceType -> auditorium, number -> 1, minArea -> 300.0, otherFeatures -> noFeature, designation -> False]. // original plan for accommoding "finalExam" activity

// Spaces

office:NonOccupiableSpace[area -> 230.4, number -> 1, features -> noFeature, openHours -> 8.0, superType -> noType, inaccessible -> noOne, equipped -> wsSet1].

wsSet1:EquipmentSet[equipment -> workstationForTeachers, number -> 70, features -> noFeature, openHours -> 8.0, inaccessible -> noOne, designated -> False].

workstationForTeachers:Equipment[area -> 3].

waitingArea:NonOccupiableSpace[area -> 50.0, number -> 1, features -> closeToOffice, openHours -> 8.0, superType -> noType, inaccessible -> noOne, equipped -> chairSet1].

chairSet1:EquipmentSet[equipment -> chair, number -> 20, features -> closeToOffice, openHours -> 8.0, inaccessible -> noOne, designated -> False].

chair:Equipment[area -> 2.3].

smallMeetingRoom:OccupiableSpace[area -> 15.0, number -> 3, features -> noFeature, openHours -> 8.0, superType -> meetingRoom, inaccessible -> noOne, designated -> False].

meetingRoom:SpaceSuperType.

studentAssociationClub:OccupiableSpace[area -> 31.0, number -> 1, features -> noFeature, openHours -> 8.0, superType -> meetingRoom, inaccessible -> noOne, designated -> False].

Appendix D3: Represented project information in the X Construction Company case

// Users: male employees, female employees, employees, new employees

maleEmployees:RegularUser[number -> 120]. // for headquarters

femaleEmployees:RegularUser[number -> 30]. // for headquarters

employees:RegularUser[number -> 150]. // for headquarters

newEmployees:RegularUser[number -> 10]. // on average

// Activity 1-1, 1-2: employees' meeting activity (small)

mEmployeesSmallMeeting:TypicalActivity[user -> maleEmployees, action -> haveSmallMeeting, ratio -> 1.0, frequency -> 1.0, constraints -> cons1, preferences -> pref1].

fEmployeesSmallMeeting:TypicalActivity[user -> femaleEmployees, action -> haveSmallMeeting, ratio -> 1.0, frequency -> 0.5, constraints -> cons1, preferences -> pref1].

haveSmallMeeting:Action[groupSize -> 5, duration -> 1.0, spaceCriteria -> 2].

cons1:WholeRoomUseReq[spaceType -> anySpace, superType -> meetingRoom, minArea -> 10, otherFeatures -> noFeature, designation -> False].

pref1:WholeRoomUseReq[spaceType -> anySpace, superType -> meetingRoom, minArea -> 12, otherFeatures -> quiet, designation -> False].

quiet:Features.

// Activity 2-1, 2-2: employees' meeting activity (large) mEmployeesLargeMeeting:TypicalActivity[user -> maleEmployees, action -> haveLargeMeeting, ratio -> 1.0, frequency -> 0.1, constraints -> cons2, preferences -> pref2].

fEmployeesLargeMeeting:TypicalActivity[user -> femaleEmployees, action -> haveLargeMeeting, ratio -> 1.0, frequency -> 0.05, constraints -> cons2, preferences -> pref2].

haveLargeMeeting:Action[groupSize -> 20, duration -> 2.0, spaceCriteria -> 1.7].

cons2:WholeRoomUseReq[spaceType -> anySpace, superType -> meetingRoom, minArea -> 40, otherFeatures -> noFeature, designation -> False].

pref2:WholeRoomUseReq[spaceType -> anySpace, superType -> meetingRoom, minArea -> 40, otherFeatures -> {quiet, screenProvided}, designation -> False].

screenProvided:Features.

// Activity 3: regular work by employees

employeesRegularWork:TypicalActivity[user -> employees, action -> doRegularWork, ratio -> 1.0, frequency -> 1.0, constraints -> cons3, preferences -> pref3].

doRegularWork:Action[groupSize -> 1, duration -> 6.0, spaceCriteria -> 3].

cons3:EquipmentUseReq[spaceType -> office, equipment -> workstation, minArea -> 3.0, otherFeatures -> noFeature, designation -> True].

pref3:EquipmentUseReq[spaceType -> office, equipment -> workstation, minArea -> 4.5, otherFeatures -> noFeature, designation -> True].

// Activity 4-1, 4-2: employees' rest

mEmployeesRest:TypicalActivity[user -> maleEmployees, action -> takeARest, ratio -> 1.0, frequency -> 3, constraints -> cons4, preferences -> pref4].

fEmployeesRest:TypicalActivity[user -> femaleEmployees, action -> takeARest, ratio -> 1.0, frequency -> 5, constraints -> cons4, preferences -> pref4].

takeARest:Action[groupSize -> 2, duration -> 0.25, spaceCriteria -> 5].

cons4:EquipmentUseReq[spaceType -> restArea, equipment -> table, minArea -> 8.0, otherFeatures -> noFeature, designation -> False].

pref4:WholeRoomUseReq[spaceType -> roomForRest, superType -> noType, minArea -> 20, otherFeatures -> noFeature, designation -> False].

// Activity 5-1, 5-2: employees’ using a restroom

mEmployeesRestroom:TypicalActivity[user -> maleEmployees, action -> useARestroom, ratio -> 1.0, frequency -> 3, constraints -> cons5, preferences -> pref5].

fEmployeesRestroom:TypicalActivity[user -> femaleEmployees, action -> useARestroom, ratio -> 1.0, frequency -> 4, constraints -> cons5, preferences -> pref5].

useARestroom:Action[groupSize -> 1, duration -> 0.1, spaceCriteria -> 2.7].

cons5:EquipmentUseReq[spaceType -> restRoom, equipment -> toilet, minArea -> 2.7, otherFeatures -> noFeature, designation -> False].

pref5:EquipmentUseReq[spaceType -> restRoom, equipment -> toilet, minArea -> 2.7, otherFeatures -> noFeature, designation -> False].

// Activity 6: training new employees

trainingNewEmployees:AtypicalActivity[user -> newEmployees, action -> trainEmployees, constraints -> cons6, preferences -> pref6].

trainEmployees:Action[groupSize -> 5, duration -> 4.0, spaceCriteria -> 2.0].

cons6:WholeRoomUseReq[spaceType -> anySpace, superType -> meetingRoom, number -> 1, minArea -> 25.0, otherFeatures -> screenProvided, designation -> False].

pref6:WholeRoomUseReq[spaceType -> anySpace, superType -> meetingRoom, number -> 1, minArea -> 25.0, otherFeatures -> screenProvided, designation -> False].

Appendix D4: Represented project information in the X Publishing Company case

// Users: employees, editors, president of the company

employees:RegularUser[number -> 20].

editors:ImportantUser[number -> 5].

president:ImportantUser[number -> 1].

// Activity 1: employees' meeting

employeesMeeting:TypicalActivity[user -> employees, action -> haveMeeting, ratio -> 1.0, frequency -> 2.0, constraints -> cons1, preferences -> pref1].

haveMeeting:Action[groupSize -> 4, duration -> 1.0, spaceCriteria -> 3.5].

cons1:WholeRoomUseReq[spaceType -> meetingRoom, minArea -> 15, otherFeatures -> noFeature, designation -> False].

pref1:WholeRoomUseReq[spaceType -> meetingRoom, minArea -> 15, otherFeatures -> noConditions, designation -> False].

// Activity 2: employees' regular work

employeesWorking:TypicalActivity[user -> employees, action -> doRegularWork, ratio -> 1.0, frequency -> 1.0, constraints -> cons2, preferences -> pref2].

doRegularWork:Action[groupSize -> 1, duration -> 5.5, spaceCriteria -> 6].

cons2:EquipmentUseReq[spaceType-> officeArea, equipment -> workStation, number -> 1, designation -> False].

pref2:EquipmentUseReq[space-> officeArea, equipment -> workStation, number -> 1, designation -> False].

// Activity 3: editing

editorsEditing:TypicalActivity[user -> editors, action -> editBook, ratio -> 1.0, frequency -> 1.0, constraints -> cons3, preferences -> pref3].

editBook:Action[groupSize -> 1, duration -> 2.0, spaceCriteria -> 4.5].

cons3:EquipmentUseReq[spaceType -> officeArea, equipment -> workStation, number -> 1, designation -> False].

pref3:WholeRoomUseReq[spaceType -> anySpace, superType -> noType, minArea -> 15, otherFeatures -> quiet, designation -> False].

quiet:Features.

// Activity 4: president's painting as her hobby

presidentPainting:TypicalActivity[user -> president, action -> paint, ratio -> 1.0, frequency -> 0.2, constraints -> cons4, preferences -> pref4].

paint:Action[groupSize -> 1, duration -> 4.0, spaceCriteria -> 30].

cons4:WholeRoomUseReq[spaceType -> artRoom, minArea -> 30, otherFeatures -> noFeature, designation -> True].

pref4:WholeRoomUseReq[spaceType -> artRoom, minArea -> 40, otherFeatures -> noFeature, designation -> True].

// Spaces

meetingRoom:OccupiableSpace[area -> 20, number -> 3, features -> quiet, openHours -> 8.0, designated -> False, inaccessible -> noOne].

officeArea:NonOccupiableSpace[area -> 175, number -> 1, equipped -> eqSet1, openHours -> 8.0].

eqSet1:EquipmentSet[equipment -> workStation, number -> 25, openHours -> 8.0, designated -> False].

workStation:Equipment[area -> 7].

artRoom:OccupiableSpace[area -> 40, number -> 1, features -> quiet, openHours -> 8.0, designated -> False].

Appendix E: Rules in prototype knowledge-based space-use analysis system

Rules defined here do not cover all features of spaces but are applicable for the prototypical implementation (Section 6.2) and the charrette test (Section 7). These rules are written in F-Logic.

```
// Metrics for the mapping
?ACT001[evtQty -> ?V001] :-
        ?ACT001:Activity[user -> ?U001, action -> ?AC001, ratio -> ?_RA001],
        ?U001:User[number -> ?_NUM001],
        ?AC001:Action[groupSize -> ?_SIZE001],
        ?V001 is (?_NUM001 * ?_RA001 / ?_SIZE001).

?ACT002[load -> ?V002] :-
        ?ACT002:Activity[evtQty -> ?_EQ002, frequency -> ?_FR002, action -
> ?AC002],
        ?AC002:Action[duration -> ?_DUR002],
        ?V002 is (?_EQ002 * ?_FR002 * ?_DUR002).

?ACT003[useArea -> ?V003] :-
        ?ACT003:Activity[action -> ?AC003],
        ?AC003:Action[groupSize -> ?_SIZE003, spaceCriteria -> ?_SC003],
        ?V003 is (?_SIZE003 * ?_SC003).
```

Appendix E. Rules in prototype knowledge-based space-use analysis system

// Rules for mapping activities onto spaces - designation

?SP101[activity -> ?ACT101, designated -> True] :- // Requirements have a space name and minimum area of the space.

((?ACT101:TypicalActivity[user -> ?U101, constraints -> ?CONS101, evtQty -> ?EQTY101],

?U101:RegularUser,

?CONS101:WholeRoomUseReq[spaceType -> ?SP101, designation -> True, minArea -> ?_MINAREA101]) or

(?ACT101:TypicalActivity[user -> ?U101, preferences -> ?PREF101, evtQty -> ?EQTY101],

?U101:ImportantUser,

?PREF101:WholeRoomUseReq[spaceType -> ?SP101, designation -> True, minArea -> ?_MINAREA101])),

?SP101:OccupiableSpace[designated -> False, number -> ?NUM101, area -> ?_AREA101],

?_VAL101 is (?NUM101 - ?EQTY101),

?_VAL101 =:= 0,

(?_AREA101 >= ?_MINAREA101),

not ?SP101:OccupiableSpace[inaccessible -> ?U101].

?EQSET151[activity -> ?ACT151, designated -> True] :- // Requirements have equipment name of a specific space type.

((?ACT151:TypicalActivity[user -> ?U151, constraints -> ?CONS151, evtQty -> ?EQTY151],

?U151:RegularUser,

?CONS151:EquipmentUseReq[spaceType -> ?SP151, equipment -> ?EQ151, designation -> True]) or

(?ACT151:TypicalActivity[user -> ?U151, preferences -> ?PREF151, evtQty -> ?EQTY151],

?U151:ImportantUser,

?PREF151:EquipmentUseReq[spaceType -> ?SP151, equipment -> ?EQ151, designation -> True])),

?SP151:NonOccupiableSpace[equipped -> ?EQSET151, number -> ?_SNUM151],

?EQSET151:EquipmentSet[equipment -> ?EQ151, number -> ?_NUM151, designated -> False],

?_VAL151 is ((?_NUM151 * ?_SNUM151) - ?EQTY151),

(?_VAL151 =:= 0),

not ?EQSET151:EquipmentSet[inaccessible -> ?U151].

?EQSET153[activity -> ?ACT153, lack -> ?VAL153, designated -> True] :- // Requirements have equipment name of a specific space type.

((?ACT153:TypicalActivity[user -> ?U153, constraints -> ?CONS153, evtQty -> ?EQTY153],

?U153:RegularUser,

?CONS153:EquipmentUseReq[spaceType -> ?SP153, equipment -> ?EQ153, designation -> True]) or

(?ACT153:TypicalActivity[user -> ?U153, preferences -> ?PREF153, evtQty -> ?EQTY153],

?U153:ImportantUser,

?PREF153:EquipmentUseReq[spaceType -> ?SP153, equipment -> ?EQ153, designation -> True])),

?SP153:NonOccupiableSpace[equipped -> ?EQSET153, number -> ?_SNUM153],

?EQSET153:EquipmentSet[equipment -> ?EQ153, number -> ?_NUM153, designated -> False],

?VAL153 is (?EQTY153 - (?_NUM153 * ?_SNUM153)),

(?VAL153 > 0),

not ?EQSET153:EquipmentSet[inaccessible -> ?U153].

// Rules for mapping activities onto spaces - non-designation

?SP201[activity -> ?ACT201] :- // Requirements have a space type and minimum area of the type.

((?ACT201:TypicalActivity[user -> ?U201, constraints -> ?CONS201],

?U201:RegularUser,

?CONS201:WholeRoomUseReq[spaceType -> ?SP201, designation -> False, minArea -> ?_MINAREA201, otherFeatures -> noFeature]) or

(?ACT201:TypicalActivity[user -> ?U201, preferences -> ?PREF201],

?U201:ImportantUser,

?PREF201:WholeRoomUseReq[spaceType -> ?SP201, designation -> False, minArea -> ?_MINAREA201, otherFeatures -> noFeature])),

?SP201:OccupiableSpace[designated -> False, area -> ?_AREA201],

not ?SP201:OccupiableSpace[designated -> True],

(?_AREA201 >= ?_MINAREA201),

not ?SP201:OccupiableSpace[inaccessible -> ?U201].

?SP202[activity -> ?ACT202] :- // Requirements have minimum area and other features of any space.

((?ACT202:TypicalActivity[user -> ?U202, constraints -> ?CONS202],

?U202:RegularUser,

?CONS202:WholeRoomUseReq[spaceType -> anySpace, designation -> False, superType -> noType, otherFeatures -> ?COND202, minArea -> ?_MINAREA202])

or

(?ACT202:TypicalActivity[user -> ?U202, preferences -> ?PREF202],

?U202:ImportantUser,

?PREF202:WholeRoomUseReq[spaceType -> anySpace, designation -> False, superType -> noType, otherFeatures -> ?COND202, minArea -> ?_MINAREA202])),

?COND202:Features,

?SP202:OccupiableSpace[designated -> False, area -> ?_AREA202, features -> ?COND202],

not ?SP202:OccupiableSpace[designated -> True],

(?_AREA202 >= ?_MINAREA202),

not ?SP202:OccupiableSpace[inaccessible -> ?U202].

?SP204[activity -> ?ACT204] :- // Requirements have a space type, other features, and minimum area of the space.

((?ACT204:TypicalActivity[user -> ?U204, constraints -> ?CONS204],

?U204:RegularUser,

?CONS204:WholeRoomUseReq[spaceType -> ?SP204, designation -> False, otherFeatures -> ?COND204, minArea -> ?_MINAREA204]) or

(?ACT204:TypicalActivity[user -> ?U204, preferences -> ?PREF204],

?U204:ImportantUser,

?PREF204:WholeRoomUseReq[spaceType -> ?SP204, designation -> False, otherFeatures -> ?COND204, minArea -> ?_MINAREA204])),

?COND204:Features,

?SP204:OccupiableSpace[designated -> False, features -> ?COND204, area -> ?_AREA204],

not ?SP204:OccupiableSpace[designated -> True],

(?_AREA204 >= ?_MINAREA204),

not ?SP204:OccupiableSpace[inaccessible -> ?U204].

?SP205[activity -> ?ACT205] :- // Requirements have minimum area of any space.

((?ACT205:TypicalActivity[user -> ?U205, constraints -> ?CONS205],

?U205:RegularUser,

?CONS205:WholeRoomUseReq[spaceType -> anySpace, superType -> noType, otherFeatures -> noFeature, designation -> False, minArea -> ?_MINAREA205]) or

(?ACT205:TypicalActivity[user -> ?U205, preferences -> ?PREF205],

?U205:ImportantUser,

?PREF205:WholeRoomUseReq[spaceType -> anySpace, superType -> noType, otherFeatures -> noFeature, designation -> False, minArea -> ?_MINAREA205])),

?SP205:OccupiableSpace[designated -> False, area -> ?_AREA205],

not ?SP205:OccupiableSpace[designated -> True],

(?_AREA205 >= ?_MINAREA205),

not ?SP205:OccupiableSpace[inaccessible -> ?U205].

?SP206[activity -> ?ACT206] :- // Requirements have a space super type, minimum area and other features.

((?ACT206:TypicalActivity[user -> ?U206, constraints -> ?CONS206],

?U206:RegularUser,

?CONS206:WholeRoomUseReq[spaceType -> anySpace, designation -> False, superType -> ?STYPE206, otherFeatures -> ?COND206, minArea -> ?_MINAREA206]) or

(?ACT206:TypicalActivity[user -> ?U206, preferences -> ?PREF206],

?U206:ImportantUser,

?PREF206:WholeRoomUseReq[spaceType -> anySpace, designation -> False, superType -> ?STYPE206, otherFeatures -> ?COND206, minArea -> ?_MINAREA206])),

?STYPE206:SpaceSuperType,

?COND206:Features,

?SP206:OccupiableSpace[designated -> False, superType -> ?STYPE206, area -> ?_AREA206, otherFeatures -> ?COND206],

not ?SP206:OccupiableSpace[designated -> True],

(?_AREA206 >= ?_MINAREA206),

not ?SP206:OccupiableSpace[inaccessible -> ?U206].

?SP208[activity -> ?ACT208] :- // Requirements have a space type and minimum area.

((?ACT208:TypicalActivity[user -> ?U208, constraints -> ?CONS208],

?U208:RegularUser,

?CONS208:WholeRoomUseReq[spaceType -> anySpace, designation -> False, superType -> ?STYPE208, otherFeatures -> noFeature, minArea -> ?_MINAREA208])

or

(?ACT208:TypicalActivity[user -> ?U208, preferences -> ?PREF208],

?U208:ImportantUser,

?PREF208:WholeRoomUseReq[spaceType -> anySpace, designation -> False, superType -> ?STYPE208, otherFeatures -> noFeature, minArea -> ?_MINAREA208])),

?STYPE208:SpaceSuperType,

?SP208:OccupiableSpace[designated -> False, superType -> ?STYPE208, area -> ?_AREA208],

not ?SP208:OccupiableSpace[designated -> True],

(?_AREA208 >= ?_MINAREA208),

not ?SP208:OccupiableSpace[inaccessible -> ?U208].

?EQSET251[activity -> ?ACT251] :- // Requirements have equipment name of a specific space type.

((?ACT251:TypicalActivity[user -> ?U251, constraints -> ?CONS251],

?U251:RegularUser,

?CONS251:EquipmentUseReq[spaceType -> ?SP251, equipment -> ?EQ251, designation -> False]) or

(?ACT251:TypicalActivity[user -> ?U251, preferences -> ?PREF251],

?U251:ImportantUser,

?PREF251:EquipmentUseReq[spaceType -> ?SP251, equipment -> ?EQ251, designation -> False])),

?SP251:NonOccupiableSpace[equipped -> ?EQSET251],

?EQSET251:EquipmentSet[equipment -> ?EQ251, designated -> False],

not ?EQSET251:EquipmentSet[designated -> True],

not ?EQSET251:EquipmentSet[inaccessible -> ?U251].

?SP252[activity -> ?ACT252] :- // When a piece of equipment accommodates an activity, space that contains the equipment also accommodates the activity.

?ACT252:TypicalActivity,

?SP252:NonOccupiableSpace[equipped -> ?EQSET252],

?EQSET252:EquipmentSet[activity -> ?ACT252].

// Rules for utilization computation

?ACT302[sumNumberOfRooms -> ?SUM302] :- // the sum of the number of space types that accommodates an activity

?SUM302 is sum{?NUM302 [?ACT302]

| ?ACT302:TypicalActivity, ?SP302:OccupiableSpace[number -> ?NUM302, activity -> ?ACT302]}.

?ACT303[loadPerSpace -> ?LPS303] :- // load of an activity per space

?ACT303:TypicalActivity[load -> ?_LOAD303, sumNumberOfRooms -> ?_SUM303],

?LPS303 is (?_LOAD303 / ?_SUM303).

?ACT305[sumNumberOfEquipment -> ?SUM305] :- // the sum of the number of equipment pieces that accommodates an activity

?SUM305 is sum{?VAL305 [?ACT305]

| ?ACT305:TypicalActivity, ?SP305:NonOccupiableSpace[activity -> ?ACT305, number -> ?_SNUM305, equipped -> ?EQSET305], ?EQSET305:EquipmentSet[number -> ?_NUM305], ?VAL305 is (?_SNUM305 * ?_NUM305)}.

Appendix E. Rules in prototype knowledge-based space-use analysis system

```
?ACT306[loadPerEquipment -> ?LPE306] :- // load of an activity per equipment
        ?ACT306:TypicalActivity[load -> ?_LOAD306, sumNumberOfEquipment -
> ?_SUM306],
        ?LPE306 is (?_LOAD306 / ?_SUM306).

?SP307[loadInSpace -> ?V307] :- // total activity loads of a space type
        ?V307 is sum{?LPS307 [?SP307] | ?SP307:OccupiableSpace[activity -
> ?ACT307], ?ACT307:TypicalActivity[loadPerSpace -> ?LPS307]}.

?EQSET308[loadInEquipment -> ?V308] :- // total activity loads of an equipment set
        ?V308 is sum{?LPE308 [?EQSET308]
| ?ACT308:TypicalActivity[loadPerEquipment -
> ?LPE308], ?EQSET308:EquipmentSet[activity -> ?ACT308]}.

?SP309[utilization -> ?UTIL309] :- // utilization of a space type
        ?SP309:OccupiableSpace[loadInSpace -> ?_VAL309, openHours -
> ?_OPEN309],
        ?UTIL309 is (?_VAL309 / ?_OPEN309).

?EQSET310[utilization -> ?UTIL310] :- // utilization of an equipment set
        ?EQSET310:EquipmentSet[loadInEquipment -> ?_VAL310, openHours -
> ?_OPEN310],
```

```
    ?UTIL310 is (?_VAL310 / ?_OPEN310).


// Rules for determining a utilization implication level
?SP401[status -> noWait] :-
    ?SP401:OccupiableSpace[designated -> False, utilization -> ?UTIL401],
    not ?SP401:OccupiableSpace[designated -> True],
    (?UTIL401 =< 0.5).

?SP402[status -> adequate] :-
    ?SP402:OccupiableSpace[designated -> False, utilization -> ?UTIL402],
    not ?SP402:OccupiableSpace[designated -> True],
    (?UTIL402 =< 0.75 and ?UTIL402 > 0.5).

?SP403[status -> inconvenient] :-
    ?SP403:OccupiableSpace[designated -> False, utilization -> ?UTIL403],
    not ?SP403:OccupiableSpace[designated -> True],
    (?UTIL403 =< 1.0 and ?UTIL403 > 0.75).

?SP404[status -> infeasible] :-
    ?SP404:OccupiableSpace[utilization -> ?UTIL404],
    (?UTIL404 > 1.0).

?SP405[status -> noWait] :-
```

```
    ?SP405:OccupiableSpace[designated -> True, utilization -> ?UTIL405],
    (?UTIL405 =< 1.0).

?EQSET406[status -> noWait] :-
    ?EQSET406:EquipmentSet[designated -> False, utilization -> ?UTIL406],
    not ?EQSET406:EquipmentSet[designated -> True],
    (?UTIL406 =< 0.5).

?EQSET407[status -> adequate] :-
    ?EQSET407:EquipmentSet[designated -> False, utilization -> ?UTIL407],
    not ?EQSET407:EquipmentSet[designated -> True],
    (?UTIL407 =< 0.75 and ?UTIL407 > 0.5).

?EQSET408[status -> inconvenient] :-
    ?EQSET408:EquipmentSet[designated -> False, utilization -> ?UTIL408],
    not ?EQSET408:EquipmentSet[designated -> True],
    (?UTIL408 =< 1.0 and ?UTIL408 > 0.75).

?EQSET409[status -> infeasible] :-
    ?EQSET409:EquipmentSet[utilization -> ?UTIL409],
    (?UTIL409 > 1.0).

?EQSET410[status -> noWait] :-
```

?EQSET410:EquipmentSet[designated -> True, utilization -> ?UTIL410],

(?UTIL410 =< 1.0).

Appendix F: Detailed results of the charrette test

Appendix F1: the X Publishing Company Case (Case 1)

Trial 1: initial prediction

This case has been developed based on the observation of a publishing company's building project. Based on the following user and space information, please predict the utilization of each space to your best knowledge. Do not assume anything other than the provided information.

Space:

Space type	Super type	Area (m^2)	Number	Features
Meeting room	N/A	20	3	quiet
Workstation (office)	N/A	7 (175)	25 (1)	N/A
Art room	N/A	40	1	quiet

User:

There are 20 employees and five of them are editors. There is also one president of this company.

Employees of this company have a meeting twice a day on average. Four employees participate in a meeting on average, and they need to use a meeting room that is equal or larger than 15 m^2 for this activity. Average meeting lasts one hour. Each employee also does a regular work at a workstation in the office. The workstations are not designated for this activity, i.e., employees can use any empty workstations for doing this activity. The company assumes that employees spend 5.5 hours per day doing regular work.

Editors edit a book once a day. It is an individual work and takes two hours on average. This activity requires a workstation in the office, and preferably a whole room with quiet conditions. The company would like to satisfy the preference (i.e., a whole room with quiet conditions) because editors are key employees in the publishing industry.

The president of this company requires space for her painting, which is her hobby. She paints once a week, and each occurrence lasts four hours. This activity requires an art room of at least 30 m^2, but preferably it requires 40 m^2. No condition is needed for this room, but the president wants the room designated solely for her.

Trial 2: first update

Because of the limited space budget, the company has determined to reduce the number of meeting rooms from 3 to 2. To reduce the impact of this decision on space-use of meeting rooms, the company also allows the art room to be used for activities other than the painting activity, i.e., the painting activity does not require having a designated space any more. Please update the utilization of each space according to this change.

Trial 3: second update

On top of the trial 2, this company would no longer provide editors with space that satisfies the preferences of them, but provide space that satisfies the constraints of them, i.e., a workstation in the office. Please update the utilization of each space according to this change.

Trial 1B: initial prediction using the prototype KSUA system

Based on the described ontology, please open Microsoft Excel file named “KSUA for Session 1” and complete the tables that describe user, user activity, and space information in pages 2-3 (information in Trial 1).

Be sure to complete all of the following worksheets in the file:

- User activity
- User
- Action
- Spatial requirements
- Space
- Equipment set
- Equipment

Upon your completion, save the file as “First Name_Trial 1_KSUA for Session 1.”

Trial 2B: first update using the prototype KSUA system

Please update your “First Name_Trial 1_KSUA for Session 1” file based on the changes described in page 4 (Trial 2).

Please mark where you updated with RED COLOR!

Be sure to update any necessary information of the following worksheets:

- User activity

- User
- Action
- Spatial requirements
- Space
- Equipment set
- Equipment

Upon your completion, save the file as "First Name_Trial 2_KSUA for Session 1."

Trial 3B: second update using the prototype KSUA system

Please update your "First Name_Trial 2_KSUA for Session 1" file based on the changes described in page 5 (Trial 3).

Please mark where you updated with RED COLOR!

Be sure to update any necessary information of the following worksheets:

- User activity
- User
- Action
- Spatial requirements
- Space
- Equipment set
- Equipment

Upon your completion, save the file as "First Name_Trial 3_KSUA for Session 1."

Appendix F. Detailed results of the charrette test

Performance of students, conventional process (manual SUA)

Trial	Student ID	Predicted utilization Meeting room	Workstation (office)	Art room	Time spent (min.)
1	1	0.83	0.55	0.10	31
	2	1.67	0.69	0.10	28
	3	0.83	0.55	0.06	10
	4	0.44	0.50	0.07	15
	5	0.48	0.55	0.10	27
	6	0.42	0.36	0.10	34
2	1	1.00	0.55	0.60	3
	2	1.37	0.69	1.00	4
	3	1.04	0.55	0.48	9
	4	0.44	0.50	0.51	5
	5	0.75	0.55	0.10	7
	6	0.42	0.11	0.35	10
3	1	0.38	0.60	0.60	14
	2	1.25	0.75	0.10	3
	3	0.42	0.60	0.49	10
	4	0.02	0.69	0.09	4
	5	0.13	0.60	0.10	6
	6	0.42	0.30	0.10	5

Performance of expert architects, conventional process (manual SUA)

Trial	Architect ID	Predicted utilization Meeting room	Workstation (office)	Art room	Time spent (min.)
1	1	0.33	0.69	0.35	5
	2	0.33	0.55	0.10	5
	3	0.43	0.55	0.10	20
	4	0.43	0.69	0.10	9
	5	0.43	0.41	0.10	10
	6	0.10	0.55	0.03	20
2	1	0.50	0.69	0.35	5
	2	0.92	0.55	0.52	3
	3	0.65	0.55	0.10	5
	4	0.65	0.69	0.10	3
	5	0.65	0.41	0.10	2

	6	0.11	0.55	0.11	10
3	1	0.50	0.89	0.10	5
	2	0.50	0.60	0.10	1
	3	0.03	0.80	0.10	5
	4	0.03	0.69	0.10	2
	5	0.05	0.46	0.10	2
	6	0.03	0.60	0.03	5

Performance of students, innovative process (the KSUA method)

Trial	Student ID	Predicted utilization			Time spent (min.)
		Meeting room	Workstation (office)	Art room	
1B	1	0.83	0.55	0.10	14
	2	0.83	0.55	0.10	15
	3	0.42	0.55	0.10	10
	4	0.52	0.55	0.07	12
	5	0.83	0.55	0.10	15
	6	0.83	0.55	0.10	12
2B	1	1.04	0.55	0.52	0.5
	2	1.04	0.55	0.52	1
	3	1.25	0.55	0.10	1
	4	0.78	0.55	0.07	1
	5	1.04	0.55	0.52	1
	6	1.04	0.55	0.52	1
3B	1	0.63	0.60	0.10	0.5
	2	0.63	0.60	0.10	0.5
	3	1.25	0.55	0.10	1
	4	0.16	0.60	0.07	0.33
	5	0.63	0.60	0.10	1
	6	0.63	0.60	0.10	1

Appendix F2: the Cygnaeus High School Case (Case 2)

Trial 4: initial prediction

This case has been developed based on the description of a high school project in Finland. Based on the following user and space information, please predict the utilization of each space to your best knowledge. Do not assume anything other than the provided information.

Space:

Space type	Super type	Area (m^2)	Number	Features
Workstation (teachers' office)	N/A	3 (230.4)	70 (1)	N/A
Chairs (waiting area)	N/A	2.3 (50)	20 (1)	close to teachers' office
Small meeting room	Meeting room	15	3	N/A
Student association club	Meeting room	31	1	N/A

User:

There are 70 teachers in this high school. They perform regular work in a workstation in the teachers' office. Teachers have their own workstations for this activity. The size of the workstation needs to be equal or larger than 3 m^2, preferably equal or larger than 4.5 m^2. Each teacher uses the workstation 1.5 hours per day on average.

There are 650 students in this school. Each year, about a third of students (i.e., 217 students) are graduating.

There is also a need for a chair where students wait for meeting with teachers. The chairs need to be situated close to the teachers' office. Thus, this activity requires occupying a chair (that is equal or larger than 2 m^2) in any room that is close to the office. This activity will be occurred once for 10 days, and average waiting time is assumed to be 0.5 hours.

Students also have a meeting the average group size of which is four. Any meeting room-type space that is equal or larger than 15 m^2 can accommodate this activity. Students prefer to perform this activity in a quiet room that is equal or larger than 20 m^2. Each student has a meeting once per 10 days on average, and the meeting lasts one hour per occurrence.

This school also has the student association. 10 students work as the student association members. The student association has also a meeting activity. This meeting is more frequent than regular students' meeting: each association member has this activity once per five days on average. This meeting continues two hours per occurrence, and the average group size is four. The student association requires a meeting room type space that can hold ten students at the same time (equal or larger than 30 m^2) for this activity.

Trial 5: first update

This school has determined to double the number of the student association members (from 10 to 20). Please update the utilization of each space according to this change.

Trial 6: second update

On top of the trial 5, this school has changed its space use policy and has prevented (regular) students from using the student association club. Please update the utilization of each space according to this change.

Trial 4B: initial prediction using the prototype KSUA system

Based on the described ontology, please open Microsoft Excel file named "KSUA for Session 2" and complete the tables that describe user, user activity, and space information in pages 8-9 (information in Trial 4).

Be sure to complete all of the following worksheets in the file:

- User activity
- User
- Action
- Spatial requirements
- Space
- Equipment set
- Equipment

Upon your completion, save the file as "First Name_Trial 4_KSUA for Session 2."

Trial 5B: first update using the prototype KSUA system

Please update your "First Name_Trial 4_KSUA for Session 2" file based on the changes described in page 10 (Trial 5).

Please mark where you updated with RED COLOR!

Be sure to update any necessary information of the following worksheets:

- User activity
- User
- Action

- Spatial requirements
- Space
- Equipment set
- Equipment

Upon your completion, save the file as “First Name_Trial 5_KSUA for Session 2.”

Trial 6B: second update using the prototype KSUA system

Please update your “First Name_Trial 5_KSUA for Session 2” file based on the changes described in page 11 (Trial 6).

Please mark where you updated with RED COLOR!

Be sure to update any necessary information of the following worksheets:

- User activity
- User
- Action
- Spatial requirements
- Space
- Equipment set
- Equipment

Upon your completion, save the file as “First Name_Trial 6_KSUA for Session 2.”

Appendix F. Detailed results of the charrette test

Performance of students, conventional process (manual SUA)

Trial	Student ID	Predicted utilization Workstation (teachers' office)	Chair (waiting area)	Small meeting room	Student association club	Time spent (min.)
4	1	0.19	0.20	0.68	0.50	17
	2	0.19	0.33	1.13	0.75	14
	3	0.19	0.20	0.51	0.63	10
	4	0.19	0.20	0.68	3.25	11
	5	0.19	0.51	0.39	1.00	9
	6	0.19	0.20	0.51	0.13	11
5	1	0.19	0.20	0.68	1.00	1
	2	0.19	0.33	1.23	0.80	6
	3	0.19	0.20	0.51	0.76	1
	4	0.19	0.20	0.68	1.63	1
	5	0.19	0.51	0.22	1.00	4
	6	0.19	0.20	0.51	0.26	2
6	1	0.19	0.20	0.68	1.00	1
	2	0.19	0.33	1.33	0.50	1
	3	0.19	0.20	0.68	0.25	1
	4	0.19	0.20	0.51	0.51	5
	5	0.30	0.51	1.02	0.20	1
	6	0.19	0.20	0.75	0.18	2

Performance of expert architects, conventional process (manual SUA)

Trial	Architect ID	Predicted utilization Workstation (teachers' office)	Chair (waiting area)	Small meeting room	Student association club	Time spent (min.)
4	1	0.19	0.20	0.05	0.55	5
	2	0.19	0.20	0.68	0.36	7
	3	0.19	0.20	0.67	0.13	25
	4	0.19	0.20	0.68	0.05	5
	5	0.19	0.20	0.68	0.15	5
	6	0.19	0.20	0.68	0.13	15
5	1	0.19	0.20	0.05	1.05	5

	2	0.19	0.20	0.68	0.38	2
	3	0.19	0.20	0.67	0.25	2
	4	0.19	0.20	0.68	0.05	1
	5	0.19	0.20	0.68	0.25	1
	6	0.19	0.20	0.68	0.25	5
6	1	0.19	0.20	0.05	1.00	5
	2	0.19	0.20	0.90	0.04	1
	3	0.19	0.20	0.67	0.25	1
	4	0.19	0.20	0.68	0.05	1
	5	0.19	0.20	0.68	0.25	1
	6	0.19	0.20	0.68	0.25	1

Performance of students, innovative process (the KSUA method)

Trial	Student ID	Predicted utilization				Time spent (min.)
		Workstation (teachers' office)	Chair (waiting area)	Small meeting room	Student association club	
4B	1	0.19	0.20	0.51	0.63	15
	2	0.19	0.20	0.51	0.63	9
	3	0.19	0.20	0.68	0.13	10
	4	0.19	0.20	0.68	0.13	13
	5	0.19	0.20	0.51	0.63	18
	6	0.19	0.20	0.51	0.63	9
5B	1	0.19	0.20	0.51	0.76	0.5
	2	0.19	0.20	0.51	0.76	0.33
	3	0.19	0.20	0.68	0.25	0.5
	4	0.19	0.20	0.68	0.25	0.5
	5	0.19	0.20	0.51	0.76	1
	6	0.19	0.20	0.51	0.76	0.5
6B	1	0.19	0.20	0.68	0.25	0.5
	2	0.19	0.20	0.68	0.25	0.25
	3	0.19	0.20	0.68	0.25	0.5
	4	0.19	0.20	0.68	0.25	0.5
	5	0.19	0.20	0.68	0.25	1
	6	0.19	0.20	0.68	0.25	0.5

Bibliography

Aalami, F., Levitt, R., and Fischer, M. (1998). "A Customizable Representation for Construction Method Models." *CIFE Working Paper 051, Stanford University*, http://cife.stanford.edu/sites/default/files/WP051.pdf.

Akbas, R., Clevenger, C., and Haymaker, J. (2007). "Temporal Visualization of Building Occupancy Phase." *Proceedings of the 2007 ASCE International Workshop on Computing in Civil Engineering*, Pittsburgh, USA, 208-215.

Akerkar, R., and Sajja, P. (2010). *Knowledge-Based Systems*. Jones and Bartlett, Sudbury, USA.

Akinci, B., Fischer, M., Kunz, J., and Levitt, R. (2002a). "Representing Work Spaces Generically in Construction Method Models." *Journal of Construction Engineering and Management*, 128(4), 296–305.

Akinci, B., Fischer, M., Levitt, R., and Carlson, R. (2002b). "Formalization and Automation of Time-Space Conflict Analysis." *Journal of Computing in Civil Engineering*, 16(2), 124-134.

Angele, J., Kifer, M., and Lausen, G. (2009). "Ontologies in F-Logic." *Handbook on Ontologies, International Handbooks on Information Systems*, S. Staab and R. Studer, eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 45–70.

Becker, R. (2008). "Fundamentals of Performance-Based Building Design." *Building Simulation*, Springer, 1(4), 356–371.

Bhatt, M., and Freksa, C. (2010). "Spatial Computing for Design: An Artificial Intelligence Perspective." *NSF International Workshop on Studying Visual and Spatial Reasoning for Design Creativity*, http://cindy.informatik.uni-bremen.de/cosy/staff/bhatt/seer/Bhatt-Freksa-SDC-10.pdf.

Bhatt, M., Hois, J., and Kutz, O. (2012). "Ontological Modelling of Form and Function for Architectural Design." *Applied Ontology Journal*, 7(3), 1–32.

Bhatt, M., Lee, J., and Schultz, C. (2011). "CLP (QS): A Declarative Spatial Reasoning Framework." *Proceedings of the 10th International Conference on Spatial Information Theory (COSIT 11), LNCS*, Springer, Belfast, Maine, 210–230.

Bourgeois, D., Reinhart, C., and Macdonald, I. (2006). "Adding Advanced Behavioural Models in Whole Building Energy Simulation: A Study on the Total

Energy Impact of Manual and Automated Lighting Control." *Energy and Buildings*, 38(7), 814–823.

Brachman, R., and Levesque, H. (2004). *Knowledge Representation and Reasoning*. Morgan Kaufmann.

Building Smart Alliance. (2013). *Building Programming Information Exchange*, http://www.buildingsmartalliance.org/index.php/projects/activeprojects/31.

Carrara, G. (1994). "Knowledge-Based Computational Support for Architectural Design." *Automation in Construction*, 1(2), 111–175.

Chau, K., Anson, M., and Zhang, J. (2005). "4D Dynamic Construction Management and Visualization Software: 1. Development." *Automation in Construction*, 14(4), 512–524.

Cherry, E. (1999). *Programming for Design: from Theory to Practice*. John Wiley & Sons, Inc., USA.

Chinowsky, P. (1991). "The CADDIE Project: Applying Knowledge-Based Paradigms to Architectural Layout Generation." *CIFE Technical Report 054, Stanford University*, http://cife.stanford.edu/sites/default/files/TR054.pdf.

Clayton, M., Kunz, J., and Fischer, M. (1998). "The Charrette Test Method." *CIFE Technical Report 120, Stanford University*, http://cife.stanford.edu/sites/default/files/TR120.pdf.

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R., and Stafford, J. (2011). *Documenting Software Architectures: Views and Beyond*. Pearson Education, Inc., Boston, USA.

Corrado, C., and Mattey, J. (1997). "Capacity Utilization." *Journal of Economic Perspectives*, 11(1), 151–167.

Darwiche, A., Levitt, R., and Hayes-Roth, B. (1988). "OARPLAN: Generating Project Plans by Reasoning about Objects, Actions and Resources." *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, 2, 169–181.

Dell'Isola, M. (2002). *Architect's Essentials of Cost Management*. John Wiley & Sons, Inc., New York, USA.

Dijkstra, J., and Timmermans, H. (2002). "Towards a Multi-Agent Model for Visualizing Simulated User Behavior to Support the Assessment of Design Performance." *Automation in Construction*, 11(2), 135–145.

Dong, B., and Andrews, B. (2009). "Sensor-Based Occupancy Beahvioral Pattern Recognition for Energy and Comfort Management in Intelligent Buildings." *11th International IBPSA Conference.* Glasgow, Scotland, 1444-1451.

Van den Dobbelsteen, A., and Wilde, S. (2004). "Space Use Optimisation and Sustainability-Environmental Assessment of Space Use Concepts." *Journal of Environmental Management*, 73(2), 81–89.

Dodier, R., Henze, G., Tiller, D., and Guo, X. (2006). "Building Occupancy Detection through Sensor Belief Networks." *Energy and Buildings*, 38(9), 1033–1043.

Emmer, R., Tauck, C., Wilkinson, S., and Moore, R. (1993). "Marketing Hotels using Global Distribution Systems." *The Cornell Hotel and Restaurant Administration Quarterly*, 34(6), 80–89.

Erhan, H. (2003). *Interactive Computational Support for Modeling and Generating Building Design Requirements*. Doctoral Dissertation, School of Architecture, Carnegie Mellon University, Pittsburgh, USA, http://www.cmu.edu/architecture/research/grad_work/2003_phdcd_erhan_halil.pdf.

Fischer, M. (2006). "Formalizing Construction Knowledge for Concurrent Performance-based Design." *Intelligent Computing in Engineering and Architecture, Lecture Notes in Artificial Intelligence 4200*, I. F. C. Smith, ed., Springer, 186–205.

Flager, F., Adya, A., and Haymaker, J. (2009). "AEC Multidisciplinary Design Optimization: Impact of High Performance Computing." *CIFE Technical Report 188, Stanford University*, http://cife.stanford.edu/sites/default/files/TR188.pdf.

Gajjar, H. and Adil, G. (2011). "Heuristics for Retail Shelf Space Allocation Problem with Linear Profit Function." *International Journal of Retail & Distribution Management*, 39(2), 144–155.

Gibson, V. (2000). "Evaluating Office Space Needs and Choices." *Research Report for MWB Business Exchange, University of Reading*, http://www.corporateso.com/PDFDocuments/EvaluatingOfficeSpaceNeedsChoices.pdf.

GSA. (2007). *GSA Building Information Modeling Guide Series: 02 – GSA BIM Guide for Spatial Program Validation*, General Services Administration, USA, http://www.gsa.gov/graphics/pbs/BIM_Guide_Series_02_v096.pdf.

Guo, S. (2002). "Identification and Resolution of Work Space Conflicts in Building Construction." *Journal of Construction Engineering and Management*, 287–295.

Hansen, K., and Vanegas, J. (2003). “Improving Design Quality through Briefing Automation.” *Building Research & Information*, 31(5), 379–386.

Haymaker, J., and Clevenger, C. (2006). “The Impact of the Building Occupant on Energy Modeling Simulations.” *Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada*, 1–10.

Helbing, D., Farkas, I., and Molnar, P. (2001). “Simulation of Pedestrian Crowds in Normal and Evacuation Situations.” *Pedestrian and Evacuation Dynamics*, M. Schreckenberg and S. D. Sharma, eds., Springer-Verlag, Berlin, 21–58.

IFMA. (2009). *Strategic Facility Planning: A White Paper*, International Facilty Management Association, USA, http://www.ifma.org/docs/knowledge-base/sfp_whitepaper.pdf.

Ioannidis, D., Tzovaras, D., and Malavazos, C. (2012). “Occupancy and Business Modelling.” *The European Conference of Product and Process Modelling (ECPPM) Conference, July 25-28, 2012*, Reykjavik, Iceland, http://www.adapt4ee.eu/adapt4ee/files/document/resources/Occupancy_and_Business_Modelling_for_eeBDM%20at%20ECPPM.pdf.

ECCJ. (2010). *Energy Conservation for Office Buildings*. The Energy Conservation Center, Japan (ECCJ), http://www.asiaeec-col.eccj.or.jp/brochure/pdf/office_building.pdf.

Kamara, J., Anumba, C., and Evbuomwan, N. (1999). “Client Requirements Processing in Construction: A New Approach Using QFD.” *Journal of Architectural Engineering*, ASCE, 5(1), 8–15.

Kifer, M. (2005). “Rules and Ontologies in F-Logic.” *Reasoning Web, Lecture Notes in Computer Science*, Springer, Berlin Heidelberg, 3564, 22-34.

Kiviniemi, A. (2005). “Requirements Management Interface to Building Product Models.” *CIFE Technical Report 161, Stanford University*, http://cife.stanford.edu/sites/default/files/TR161.pdf.

Levary, R., and Kalchik, S. (1985). “Facilities Layout—A Survey of Solution Procedures.” *Computers and Industial Engineering*, Elsevier, 9(2), 141-148.

Mahdavi, A., Mohammadi, A., Kabir, E., and Lambeva, L. (2008). “Shading and Lighting Operation in Office Buildings in Austria: A Study of User Control Behavior.” *Building Simulation*, 1(2), 111–117.

Mahdavi, A., and Pröglhöf, C. (2009). “Toward Empirically-Based Models of People’s Presence and Actions in Buildings.” *Eleventh International IBPSA Conference, July 27-30, 2009*, Glasgow, Scotland, 537–544.

Maile, T., Fischer, M., Haymaker, J., and Bazjanac, V. (2010). “Formalizing Approximations, Assumptions, and Simplifications to Document Limitations in Building Energy Performance Simulation.” *CIFE Working Paper 126, Stanford University*, http://cife.stanford.edu/sites/default/files/WP126.pdf.

Melfi, R., Rosenblum, B., Nordman, B., and Christensen, K. (2011). “Measuring Building Occupancy using Existing Network Infrastructure.” *International Green Computing Conference and Workshops (IGCC), July 25-28, 2011*, IEEE, Orlando, Florida, USA, 1–8.

Meyers, F. (1993). *Plant Layout and Material Handling*. Prentice Hall. New Jersey, USA.

Mourgues, C., Fischer, M., and Kunz, J. (2012). “Method to Produce Field Instructions from Product and Process Models for Cast-In-Place Concrete Operations.” *Automation in Construction*, 22, 233–246.

Noy, N., and McGuinness, D. (2001). “Ontology Development 101: A Guide to Creating Your First Ontology.” *Stanford Knowledge Systems Laboratory Technical Report*, KSL-01-05, http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf.

Pan, X., Han, C., Dauber, K., and Law, K. (2007). “A Multi-Agent Based Framework for the Simulation of Human and Social Behaviors During Emergency Evacuations.” *Ai & Society*, 22(2), 113–132.

Peňa, W., and Parshall, S. (2001). *Problem Seeking: An Architectural Programming Primer*. John Wiley & Sons, Inc., New York, USA.

Pendlebury, D. (1990). “Design for Working.” *American Bar Association Journal*, 76(April), 98.

Pennanen, A. (2004). *User Activity Based Workspace Definition as an Instrument for Workplace Management in Multi-User Organizations*. Doctoral Dissertation, Department of Architecture, Tampere University of Technology, Tampere, Finland, https://www.haahtela.fi/images/publications/pdf/Workspace_Planning.pdf.

Pérez-Lombard, L., Ortiz, J., and Pout, C. (2008). “A Review on Buildings Energy Consumption Information.” *Energy and Buildings*, 40(3), 394–398.

Preiser, W., Rabinowitz, H., and White, E. (1988). *Post-Occupancy Evaluation*. Van Nostrand Reinhold, New York, USA.

Preiser, W., and Vischer, J. (2005). *Assessing Building Performance*. Elsevier Butterworth-Heinemann.

Ritzman, L., Bradford, J., Jacobs, R., Rttzman, L., and Bradfordf, J. (1979). "A Multiple Objective Approach to Space Planning for Academic Facilities." *Management Science*, 25(9), 895–906.

Sharp, T. (1996). "Energy Benchmarking In Commercial Office Buildings." *ACEEE 1996 Summer Study on Energy Efficiency in Buildings*, 321–329.

Shen, W., Shen, Q., and Sun, Q. (2012). "Building Information Modeling-Based User Activity Simulation and Evaluation Method for Improving Designer–User Communications." *Automation in Construction*, 21, 148–160.

StanfordUniversity. (2009). *Stanford University Space and Furniture Planning Guidelines.* Department of Capital Planning and Space Management, Stanford University, http://lbre.stanford.edu/cap_plan/sites/all/lbre-shared/files/docs_public/DCPSM_SpaceandFurniturePlanningGuidelines_v3_April_2009.pdf.

Staub-french, S., Fischer, M., Kunz, J., and Paulson, B. (2003). "An Ontology for Relating Features with Activities to Calculate Costs." *Journal of Computing in Civil Engineering*, 17(4), 243–254.

Tabak, V. (2008). *User Simulation of Space Utilisation*. Doctoral Dissertation, Design Systems Group, Eindhoven University, Eindhoven, Netherlands, http://ykje.ca/Human%20Behavior%20Modelling.pdf.

Thabet, W., and Beliveau, Y. (1994). "Modeling Work Space to Schedule Repetitive Floors in Multistory Buildings." *Journal of Construction Engineering and Management*, 120(1), 96–116.

Ullman, J., and Widom, J. (2007). *A First Course in Database Systems*. Prentice Hall, Upper Saddle River, USA.

Vischer, J. (2008). "Towards a User-Centred Theory of the Built Environment." *Building Research & Information*, 36(3), 231–240.

Wang, H., Boukamp, F., and Elghamrawy, T. (2011). "Ontology-Based Approach to Context Representation and Reasoning for Managing Context-Sensitive Construction Information." *Journal of Computing in Civil Engineering*, 25(5), 331–346.

Whelton, M. (2004). *The Development of Purpose in the Project Definition Phase of Construction Projects- Implications for Project Management*. Doctoral Dissertation, Department of Civil and Environmental Engineering, University of California at Berkeley, Berkeley, USA, http://www.leanconstruction.org/media/docs/WheltonMichaelPhD2004.pdf.

Whyte, J., and Gann, D. M. (2001). “Closing the Loop between Design and Use: Post-Occupancy Evaluation.” *Building Research & Information*, 29(6), 460–462.

Willis, A., Gjersoe, N., Havard, C., Kerridge, J., and Kukla, R. (2004). “Human Movement Behaviour in Urban Spaces: Implications for the Design and Modelling of Effective Pedestrian Environments.” *Environment and Planning B: Planning and Design*, 31(6), 805–828.

Yan, W., and Kalay, Y. (2006). “Geometric, Cognitive, and Behavioral Modeling of Environmental Users.” *Design Computing and Cognition*, 61–79.

Yang, G., Kifer, M., and Zhao, C. (2003). “Flora-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web.” *Second International Conference on Ontologies*, *Databases and Applications of Semantics (ODBASE)*, Catania, Sicily, Italy, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.140.7283.

Yu, A., Shen, Q., Kelly, J., Hunter, K., and Ann, T. (2008). “Comparative Study of the Variables in Construction Project Briefing/Architectural Programming.” *Journal of Construction Engineering and Management*, 134(2), 122–138.

Yu, A., Shen, Q., Kelly, J., and Hunter, K. (2007). “An Empirical Study of the Variables Affecting Construction Project Briefing/Architectural Programming.” *International Journal of Project Management*, 25(2), 198–212.

Yu, Z., Fung, B., Haghighat, F., Yoshino, H., and Morofsky, E. (2011). “A Systematic Procedure to Study the Influence of Occupant Behavior on Building Energy Consumption.” *Energy and Buildings*, 43(6), 1409–1417.

Yun, G. and Steemers, K. (2008). “Time-Dependent Occupant Behaviour Models of Window Control in Summer.” *Building and Environment*, 43(9), 1471–1482.

Zimmerman, A., and Martin, M. (2001). “Post-Occupancy Evaluation: Benefits and Barriers.” *Building Research and Information*, 29(2), 168–174.

Zimmermann, G. (2007). “Modeling and Simulation of Individual User Behavior for Building Performance Predictions.” *Proceedings of the 2007 Summer Computer Simulation Conference*, Society for Computer Simulation International, San Diego, California, 913–920.