NETWORKING KNOWLEDGE AND EXPERIENCE:

An Instrumental System for the Personal Development of Individual Designers

by

Bowen Lu

Master of Science in Architecture Design and Research
University of Michigan, 2017

Bachelor of Architecture
Illinois Institute of Technology, 2015

Submitted to the Department of Architecture
in partial fulfillment of the requirements for the degree

of

Master of Science in Architecture Studies

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Signature of Author: ____

Department of Architecture
May 14, 2021

Certified by: ____

George Stiny
Professor of Design and Computation, Department of Architecture
Thesis Supervisor

Accepted by: ____

Leslie K. Norford
Professor of Building Technology
Chair, Department Committee on Graduate Students

Thesis Committee

Thesis Advisor

George Stiny
Professor of Design and Computation
Department of Architecture

Thesis Reader

Paul Keel
Research Scientist
Department of Electrical Engineering and Computer Science

Networking Knowledge and Experience:
An Instrumental System for the Personal Development of Individual Designers

by

Bowen Lu

Submitted to the Department of Architecture on May 14, 2021
in partial fulfillment of the requirements for the degree of

Master of Science in Architecture Studies

Abstract

This research describes a new design thinking technology, which draws knowledge and experience from the learning and ideation of individual designers, and makes this accumulation accessible for future use and inspiration.

Pursuing novelty and diversity, designers are trained through a wide spectrum of different disciplines, requiring a tremendous amount of explicit knowledge and implicit experience. As a result, designers must go beyond the apprentice-based practice long promoted by design education to embrace a more personal exploration of design ideas. However, while a technology surge of computer-aided design (CAD) increases productivity, it limits our imagination to a predefined structure and framework. A technology that facilitates knowledge accumulation and open-ended design ideation is required, especially in long term for individual designers.

In this thesis, I propose a new theory of design ideation representation that integrates combinatory systems of knowledge engineering that extract and simulate symbolic knowledge for decision-making as well as constructive systems of visual calculating that prioritize human visual perception for ambiguous and unrestricted imagination. Based on the integrated theory, I develop a software prototype that augments designers to acquire and take control of their knowledge and experience to generate new, diverse, and creative ideas. I also demonstrate and analyze a constructed knowledge network by the software system and how the system is used in the design process.

This research contributes to a new direction of design technology for ideation as a counterpart of computer-aided design (CAD) technology for productivity. The software prototype inspires new design tools for creative design thinking. It takes one more step towards a promising future of augmented intelligence where more powerful human-computer integration can be actualized.

Thesis Supervisor: George Stiny

Title: Professor of Design and Computation

Acknowledgments

First, I would like to thank Professor George Stiny for being my advisor and offering critical insights into my initial research of knowledge. I was introduced to a new territory of visual experience, and this thesis would have been totally different and lost such diversity without those opportunities. I would like to thank Paul Keel for being my thesis reader and always supporting my new ideas of software. This research would not have been productive without your practical suggestions and experiences.

Furthermore, I would like to thank Professor Lawrence Sass for his continuous support in my studies during the SMArchS program. This research would not happen without all the conversations and independent projects we worked on together. I would like to thank Professor Terry Knight for her insightful guidance at the early stage of my research. The inquiry class and our discussions contribute so much to this thesis.

I would like to thank Alex, Athina, Alessandra, Diego, Sandy, and all my Computation Group colleagues who gave me valuable advice and kind support. It is a great pleasure working with you and I eagerly expect that we can get back and work together in the future.

Finally, I would like to thank my family for their close attention, direction, and encouragement in my studies. You helped me overcome so many difficulties in my work and life, and I owe you all.

Table of Contents

List of Figures

Figure 2-3, 2-4, 2-5, 2-6 are used under permission granted through a license signed by MIT with Elsevier and Springer Nature for access to the journals at MIT. More details on:
https://libraries.mit.edu/scholarly/publishing/using-published-figures/

Chapter 1

Introduction

1.1 Design, Creativity, and Computation

When we talk about design, it always refers to innovation, novelty, and creativity. Industrial designers create fresh user experiences with novel products, and architectural designers create mindful living experiences with unprecedented spaces. It is now universally acknowledged that design is an intellectual work that requires intensive thinking processes, and design as a third culture was firstly separated from the other two cultures, namely science and humanity by Archer [1]. Historically, design had not yet become a discipline but instead a set of skills inherited by generations of artisans in art, handicraft, and architecture. Since the 1960s, design has been extensively studied as a unique methodology and thinking [2]. Though being extremely diverse, design still pivots on the value of human intelligence, and the acquisition of design abilities still originates from apprentice-based education and practice.

Nowadays, computational tools greatly impact the way design projects are produced. Including *Computer-aided Design (CAD)* and *Building Information Modeling (BIM)*, technology provides a significant improvement of efficiency and productivity but is also criticized as it deteriorates creativity [3]. It is difficult because computer technology as an invention of science has a fundamentally different structure from design. Since Ivan Sutherland's *SketchPad* [4], CAD tools follow a strict hierarchical structure of digital objects, while thoughts in a designer's mind can be way more complicated and implicit. This contrast of "clean technology" and "messy talk" [3] demonstrates the main obstacle for computers to facilitate the creative design process. Is it possible to have an ideation tool, as the counterpart of production systems like CAD and BIM, that contributes to design abilities and creativity? If so, then what is the appropriate relationship between computers and human designers in this increasingly complicated world?

1
Archer, B. (1979). Design as a discipline. *Design studies*, *1*(1), 17-20.

2
Cross, N. (2007). 'Forty years of design research'. *Design studies*, *1*(28), 1-4.

3
Dossick C. S. & Neff G. (2011). 'Messy talk and clean technology: communication, problem-solving and collaboration using Building Information Modelling'. *Engineering Project Organization Journal*, 1:2, 83-93.

4
Sutherland, I. (1975). 'Structure in drawings and the hidden-surface problem'. *Reflections on computer aids to design and architecture*, 73-77. Petrocelli/Charter, New York.

1.2 Vision

Answering those questions is significant for further understanding computational tools for design purposes.

Firstly, it requires an investigation of design discipline and its relationship with science and humanity (arts). How do designers acquire the abilities to design? Are they *educated* or *coached*? There is always a contradiction between scientific and artistic perspectives of design.

In fields where design is studied as a scientific method, the ability to design is described as a combinatory system of elements that generates ideas as organizing "building blocks" [5]. Those elements are regarded as reusable design knowledge that can be extracted from expert designers, educated to novice designers, and ideally utilized by computer systems. The design process then becomes a solution-oriented procedure where the results can be measured and qualified for the satisfaction of design problems [5].

Among areas where design is regarded as an artistic behavior, design abilities come from designers' communication with a set of constructive and intuitive processes, which are usually uncertain and implicit, also known as "reflection-in-action" [6]. Through these practices, designers cannot solely apply prior knowledge but have to also rely on the subconscious experience that is already fused into their personalities. This human exclusive experience is not educated but coached through reflections, which cannot be easily realized within a logical framework of science.

This dichotomy of design leads to a variety of theories and applications. Even though design is already recognized as a unique way of thinking [7][8], it absorbs knowledge from science and experience from art, becoming a distinctive "way of knowing." Designers are likely to pursue a collaboration of the three cultures in order to develop their own abilities of design.

Secondly, research on computational and digital design needs to be studied in order to find the appropriate way to facilitate creative design with computers.

Knowledge has been broadly studied by computer and cognitive science in the field of *knowledge representation (KR),* and knowledge-based systems are developed to simulate human-level reasoning [9].

5
Simon, H. A. (1969). *The sciences of the artificial*. Cambridge: MIT Press.

6
Schön, D. A. (1987). *Educating the reflective practitioner: toward a new design for teaching and learning in the professions.* San Francisco: Jossey-Bass.

7
Cross, N. (1982). 'Designerly ways of knowing'. *Design studies*, *3*(4), 221-227.

8
Lawson, B., & Loke, S. M. (1997). Computers, words and pictures. *Design studies*, *18*(2), 171-183.

9
Davis, R., Shrobe, H., & Szolovits, P. (1993). 'What is a Knowledge Representation? ' *AI Magazine*, 14(1):17-33.

Adopting KR research, design knowledge is also represented by design systems that aim at improving design education [10][11].

Experience can be represented in different ways, among which visual material is the most adopted format by designers. Other than drawings and models produced by CAD systems, the digital sketch has more potential to cover designers' experience along with design processes. The sketches designers made during conceptual design phases reveal accumulated design abilities and can be formalized by visual calculating theory [12].

The difference between symbolic calculating of knowledge and visual calculating of experience needs to be considered. Knowledge may constrain design possibilities, and at the same time experience may obscure design trajectories. In order to promote design creativity, computers are likely to handle knowledge and experience in novel approaches.

1.3 Hypothesis

A computational tool for ideation, therefore, needs to collaborate with both knowledge and experience, providing opportunities for transformation between the two. In this thesis, I propose to research representations for both design knowledge and experience as explicit and implicit network models and provide a mechanism to unite them. Based on this theory of representation, I also propose the development of an instrumental software system that facilitates the accumulation of knowledge and experience. The goal is to provide computational augmentations that contribute to individual designer's development of design abilities.

1.4 Methods

In developing the theoretical model of representation, a network of explicit knowledge based on KR theories and cognitive architecture is the foundation structure. Another network of implicit experience in the form of connected digital sketches is built on top of the structured knowledge. The union of the two networks becomes the representation of designers' personal accumulation, providing intuitive abilities to transform between knowledge and experience.

10
Heylighen, A., & Neuckermans, H. (2000). 'DYNAMO: A Dynamic Architectural Memory On-line'. *Journal of Educational Technology & Society*, *3*(2), 86-95.

11
Oxman, R. (2004). 'Think-maps: teaching design thinking in design education'. *Design studies. 25*(1), 63-91.

12
Stiny, G. (2006). *Shape: talking about seeing and doing.* MIT Press.

The instrumental system implementation considers the workflow of input-storage-retrieval that is highly integrated into the design and learning process. All collected knowledge and experience in various formats, including textual, pictorial, and sketch information, need to be formulated in a designated database. The software interface needs to provide real-time access to new and learned content, which is powered by the background logic of searching and retrieving. Completing this workflow, designers integrate this computational augmentation into the construction of their own design abilities.

1.5 Expected Results and Contributions

A demonstration of using this system will be made through learning from different types of material. The system is expected to host explicit knowledge beyond designers' mnemonic capacity and visual experience in subconscious memory. The acquired knowledge and experience will be analyzed and used in design tasks. The results may vary depending on specific design styles as designers ideate differently, which will provide further direction on how to improve the theory and system.

This research provides a comprehensive solution of connecting explicit knowledge and implicit experience as a computational tool that focuses on individual designers' development of design abilities. It contributes to a new approach of design education and practice augmented by computers in this digital age.

Chapter 2

Background

2.1 Design as a Discipline

The rise of design as a discipline started in the 1960s when designers and researchers began to relate design with science and rationality, trying to find design methods that can lead to creative [2]. Design becomes a problem that needs systematic solutions as Simon [5] established the early design science. Design knowledge becomes the "building blocks" in a combinatory system where ideas can be assembled along with the design process.

By the 1970s, the design method of logic and optimization was questioned by many of its early advocators. Design problems were found as ill-defined problems that cannot simply fit entirely in a scientific framework but can only be satisfied by appropriate solutions [2]. The problem-solving strategies of designers and scientists were compared to demonstrate that designers adopt a satisfactory approach rather than a systematic one [13].

The correlation between design and cognition began to emerge in the 1980s. Design thinking [14][15] or designerly ways of knowing [1][7] reinforced the uniqueness of design discipline. Designers use nonverbal and cognitive models [16] [17] [18] to acquire and apply design abilities in their education and practice. Donald Schön [6] brought the idea of reflective practice that connects design to the intuitive and implicit view of arts that introduces uncertainty and ambiguity. Design problems are more likely to be set rather than solved in this reflective process.

Those theories do not replace each other but rather contribute to different respects in design research. Design knowledge and methods are widely used in engineering, and design cognition and reflective practice are common in art and architecture studios. Depending on specific design projects, both knowledge and experience can be applied

13
Lawson, B. (1979) 'Cognitive strategies in architectural design'. *Ergonomics.* Vol 22, No 1, 59-68.

14
Lawson, B. (1980) *How Designers Think: The Design Process Demystified.* Architectural Press, Oxford, Elsevier.

15
Rowe, P (1987) *Design Thinking* MIT Press, Cambridge, MA.

16
Archer, B (January 1980). 'The mind's eye: not so much seeing as thinking' *Designer*. 8-9 33.

17
Ferguson, E S (1977) 'The mind's eye: non-verbal thought in technology' *Science* Vo1197 No 4306 32.

18
Cross, A (1980) 'An introduction to non-verbal aspects of thought' *Design Educ. Res. Note 5* Design Discipline, The Open University, Milton Keynes, Bucks, UK.

in various ways. It is necessary to investigate both of them separately and look for affiliations to construct a more universal framework.

2.2 Knowledge Representation

2.2.1 What is Knowledge?

It is popular to give a philosophical definition of knowledge as the "justified true belief," and it could be more plural. People believe that knowledge is valuable and requires efforts and resources to earn. Different from information and data, knowledge is about generalization that can be used to predict future events. Concepts that give imaginary names to physical things become useful and efficient.

It has a variety of classifications to indicate how we think and perform differently. There are abstract *conceptual knowledge* and concrete *episodic knowledge* correspond to our semantic and episodic memory [19]. Declarative knowledge and procedural knowledge refer to specific things and particular tasks. Ryle [20] also terms them as *knowing-that* and *knowing-how*, as one for systematic descriptions and the other for sequential skills.

Knowledge, especially explicit knowledge, has its designated structure. There are many cognitive structures developed by philosophers and psychologists in history to describe acquired knowledge. For example, the concept of *schema* that has a great impact on cognitive science and artificial intelligence research. It originated from Plato's ideal type, and Kent's description of "structures which organize our world" [21], and is adopted to create theories including *script* [22] and *frame* [23][24]. Either interpreted verbally or as a mental image, *schemata* provide knowledge structure to our internal thoughts and the external physical world.

Knowledge also has strong affiliations with languages, which are the vehicle of our thoughts and communication. Spoken and written material is the primary format of explicit knowledge dissemination. Similarly, languages are also structured syntactically to convey different meanings in combinatory ways [25]. Based on language models, thoughts and objects are named as words and are organized into sentences. But not all language instances carry knowledge, and the same knowledge can be interpreted in different forms and languages without a change in its meaning. Knowledge can be represented and articulated beyond the scene of languages.

19
Tulving, E. (1985). 'How many memory systems are there? '. *American Psychologist*, *40*(4), 385–398.

20
Ryle, G. (2009). *The concept of mind*. Routledge

21
Kant, I. (1929) *Critique of pure reason*. Smith, N. Kemp, Trans.

22
Schank, R. C., and R. P. Abelson. (1977). *Scripts, Plans, Goals, and Understanding: An Enquiry into Human Knowledge Structures*. Erlbaum.

23
Minsky, M. (1975), 'A framework for representing knowledge', P H Winston Ed. *The psychology of computer vision,* McGraw-Hill, New York, NY.

24
Minsky, M. (2007). *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster.

25
Chomsky, N. (2002) *Syntactic structures*. De Gruyter Mouton.

2.2.2 Knowledge Representation

It is easier to solve a problem once the right representation is found [26]. As part of artificial intelligence research, *knowledge representation (KR)* focuses on understanding human knowledge and enabling computers to utilize the knowledge. It originated from fields including logical formalism, psychological behaviorism, biology, and management. The primary perspectives of KR are ontological commitment and logical reasoning [9].

Mostly known as a philosophical term, *ontology* is a metaphysics used to describe existence or what things really are. In the KR context, it is used as the structural representation that describes knowledge affiliations or what knowledge is really about. A graph-based model as the aggregation of nodes and edges is widely used, and linguistic concepts are mostly adopted to assign meanings to the structure. One of the most significant representations is the semantic network [27] that links meaningful concepts within a knowledge structure.

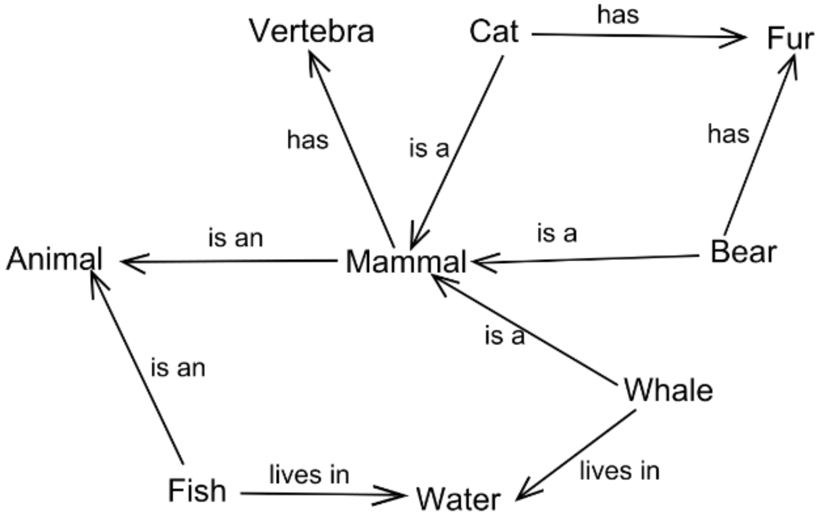


Figure 2-1: Illustration of a semantic network.
Accessed on May 12, 2021, copyright released into the public domain
https://commons.wikimedia.org/wiki/File:Semantic_Net.svg

The logic reasoning component of KR is responsible for making decisions based on acquired knowledge. Started from the 1990s, artificial intelligence researchers built expert systems that consist of a knowledge base and an inference engine to mimic human expert activities. Applications of KR are invented ever since, including *CYC* [28] on common sense reasoning and *NELL* [29] in computer language

26
Winston, P. H. (1993). *Artificial Intelligence, Third Edition*. Addison-Wesley.

9
Davis, R., Shrobe, H., & Szolovits, P. (1993). 'What is a Knowledge Representation? ' *AI Magazine*, 14(1):17-33.

27
Quillan, M. R. (1966). *Semantic memory*. Bolt Beranek and Newman Inc Cambridge MA.

28
Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., & Shepherd, M. (1990). 'Cyc: toward programs with common sense'. *Communications of the ACM*, *33*(8), 30-49.

29
Mitchell, T., & Fredkin, E. (2014). 'Never ending language learning'. In *Big Data (Big Data), 2014 IEEE International Conference.* 1-1.

learning. There are also public semantic web projects such as *DBpedia*, *WikiData*, and Google's *Knowledge Graph* that focus on the distribution of formalized knowledge over the Internet.

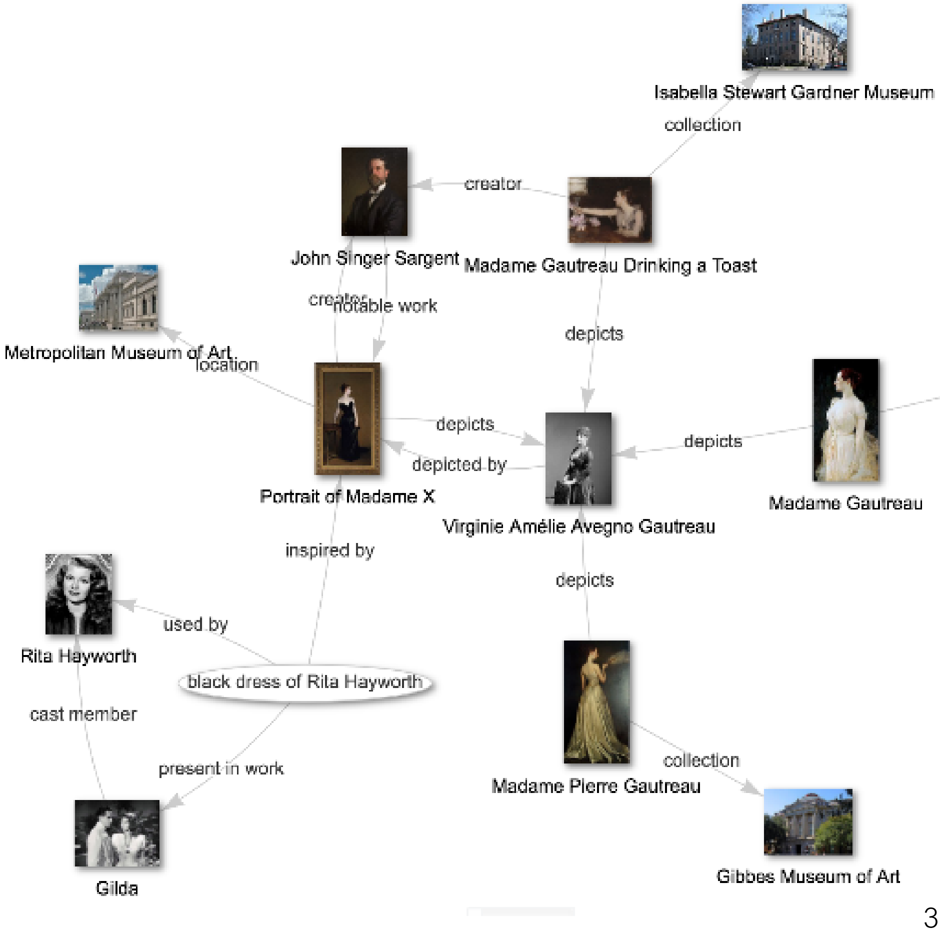


Figure 2-2: Illustration of a knowledge graph.
Accessed on May 12, 2021, licensed under the Creative Commons https://commons.wikimedia.org/wiki/File:Wikidata-knowledge-graph-madame-x-2019.png

2.2.3 What is Design Knowledge?

The concepts of design knowledge and ideation are mentioned mostly in engineering disciplines where problem-solving and optimization purposes are prioritized. Being more open-ended, knowledge in architectural design can also be formalized as organized structures [30][31]. Researchers also developed paradigms to capture essential elements of design, such as *ICF (Issue–Concept–Form)* framework [32] and *FBS (Function-Behavior-Structure)* ontology [33]. The concepts generated in design processes contribute to both passive knowledge acquisition and active knowing in the design studio [34], and this

[30] Mitchell, W. (1990). *The logic of architecture: design, computation and cognition*. MIT Press.

[31] Lawson, B. (1994). *Design in mind.* Oxford: England: Butterworth Architecture.

[32] Oxman, R. E. (1994). 'Precedents in design: a computational model for the organization of precedent knowledge'. *Design studies*, *15*(2), 141-157.

cognitive design learning process includes representation, reasoning, and knowledge structure [35]. Design knowledge, at least the explicit part, can then be formulated with KR models, and a variety of design systems have been developed to support designers and design students in the perspective of knowledge.

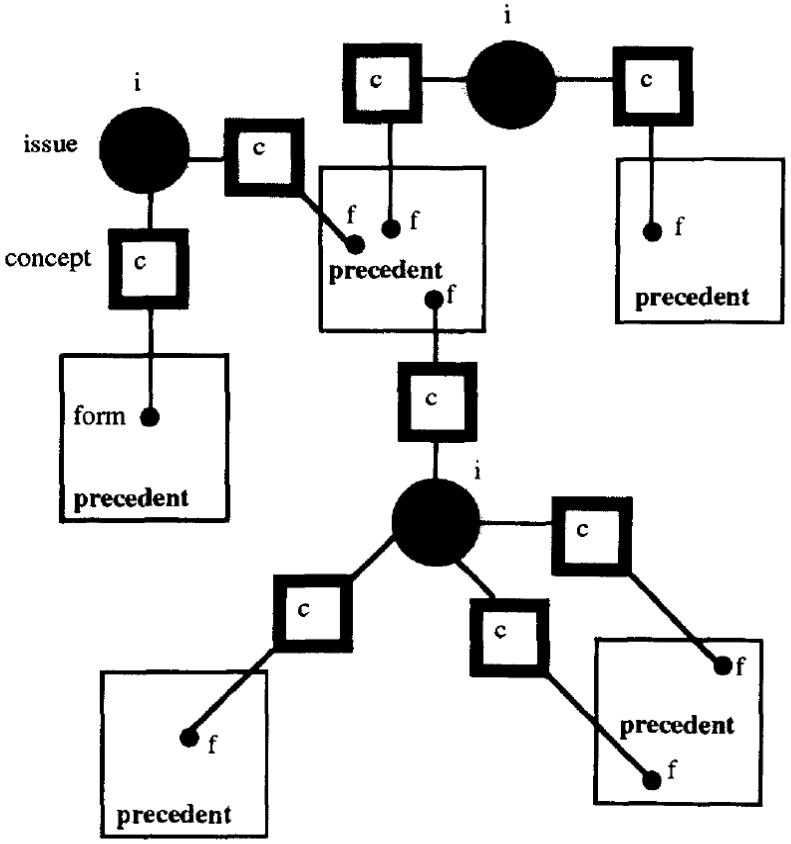


Figure 2-3: Oxman's Structure of memory network [32], using the ICF framework

2.2.4 Design Knowledge Representation Systems

Architectural design firms pay great attention to keeping their organizational documentation of precedent projects and works, where BIM and CAD tools help with maintaining a platform of knowledge sharing [36]. Employing KR theories and applications, more design systems have been developed to transfer knowledge into the personal accumulation of designers and students.

33
Gero, J. S., & Kannengiesser, U. (2014). 'The function-behaviour-structure ontology of design'. In *An anthology of theories and models of design.* 263-283. Springer, London.

34
Heylighen, A., Bouwen, J., & Neuckermans, H. (1998). 'Walking on a thin line—Between passive knowledge and active knowing of components and concepts in architectural design'. *Design Studies, 20*, 211-235.

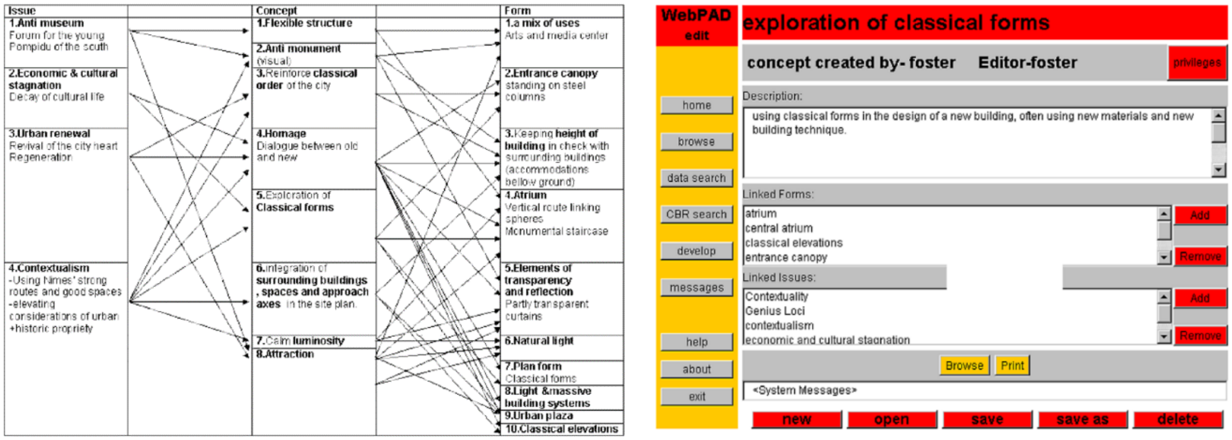
35
Oxman, R. (2001). 'The mind in design: a conceptual framework for cognition in design education'. In *Design knowing and learning: Cognition in design education.* 269-295. Elsevier Science.

36
Miller, H. (2018). 'Cultivating Next-GEN Designers-The Systematic Transfer of Knowledge'. In *Proceedings of the 36th eCAADe Conference.* 25-34.

DYNAMO (Dynamic Architectural Memory On-line) [10] provides an online platform with linked design cases and labels for the exchange of knowledge between designers at different levels. Designers and students can thus learn not only from design cases but the shared characteristics through connections that they have never experienced before. Another project *Building Stories* [37] instead provides labeled stories with metadata rather than direct case information for better storytelling.

37
Martin, W. M., Heylighen, A., & Cavallin, H. (2003). 'Building² Stories. A hermeneutic approach to studying design practice'. In *Proceedings of the 5th European Academy of Design Conference*.

More research projects utilize the semantic web, a larger-scale projection of the semantic network on the Internet, to bring more details to their knowledge bases as *linked open data (LOD)* and support designers' reasoning through linkages. [38]



exploration of classical forms

concept created by- foster Editor-foster

Description:
using classical forms in the design of a new building, often using new materials and new building technique.

Linked Forms:
atrium
central atrium
classical elevations
entrance canopy

Linked Issues:
Contextuality
Genius Loci
contextualism
economic and cultural stagnation

<System Messages>

Figure 2-4: Oxman's Think-map using ICF relations (left) and the WebPAD software interface (right) [11]

In addition to a data structure that hosts precedent design knowledge, the network structure can be used as a pedagogical tool through concept mapping. *Think-map* [11] along with its software implementation *Web-Pad* utilizes concept map to formulate and teach explicit domain knowledge of design. Students construct a think-map collaboratively using concepts learned from precedent design cases with relations defined by an *ICF (Issue–Concept–Form)* framework. The constructed case-base can then be developed, browsed, and searched using the Web-Pad tool.

38
Pauwels, P., De Meyer, R., & Van Campenhout, J. (2011). 'Extending the design process into the knowledge of the world'. In *14th International Conference on Computer Aided Architectural Design Research in Asia (CAADFutures-2011)*. 203-216. Les éditions de l'université de Liège.

2.2.5 Evaluation of Knowledge Systems

The variety of knowledge-based design systems utilize the powerful network model to represent design knowledge on computational platforms to facilitate design processes. However, most systems are web-based public projects that aim at transferring novice designers into more professional designers with complementary pieces of knowledge within a centralized framework. Feeding with design cases, designers can acquire information and exploit existing design techniques but may also be assimilated with similar ideas. More space for designers' unique personal explorations and achievements could be offered.

For easier construction, most systems utilize a set of design paradigms, such as ICF. Those paradigms are efficient to use but can also constrain novel representation of design knowledge into designated structures. The balance of standardization and improvisation should also be considered.

Moreover, the software implementations are mostly based on a database inquiry system that responds to users' search queries. This operational style is appropriate for looking up potential answers to a problem but not for an ongoing design process that may wander in different directions.

Utilizing knowledge is significant in design, but it is not the complete story. Other than the explicit part that can be articulated, the implicit and subconscious components deserve more exploration, which is covered in the following discussion about *experience*.

2.3 Experience Representation

2.3.1 What Is Experience?

Different from the definition of knowledge, which can be abstract but still explainable, there is not a formal way to describe design experience as it embeds in behaviors and embodies as artifacts. The experience discussed in this thesis is also termed implicit knowledge, which has significant differences from the explicit and structured knowledge mentioned in previous sections. Researchers have tried to incorporate engineering implicit knowledge into structures like graph models, but need to reduce its dependency on personal capability [39], which is against the purposes of facilitating individual designers in this thesis. The implicit experience or knowledge needs to be considered separately from the explicit knowledge.

The well-known argument of *reflective practice* can help with understanding how experience shapes designers' abilities over time. According to Schön [6], it is the conversation between practitioners and material that contributes to the development of design abilities. This

39
Iwasaki, K., Kuriyama, Y., Kondoh, S., & Shirayori, A. (2018). 'Structuring engineers' implicit knowledge of forming process design by using a graph model'. *Procedia Cirp*, *67*, 563-568.

6
Schön, D. A. (1987). *Educating the reflective practitioner: toward a new design for teaching and learning in the professions.* San Francisco: Jossey-Bass.

process is implicit and intuitive, and the acquired experience is uncertain and unstructured. The reflection from every single practice is built into designers' subconscious memory that cannot be retrieved in explicit ways, but can only emerge with unexpected intuitions.

There can be a broader range of experience to study than knowledge. A thought experiment is a thinking experience, and glass blowing can be a making experience. There are difficulties in working with these two types of experience as the former requires explicit language structure to articulate, and the latter involves significantly complicated physical interactions with the external objects and environment. To address the uniqueness of experience and its representation, visual experience including drawing and sketching is noteworthy. Visual experience can be captured graphically within the scale and complexity of an individual designer, and it is compatible with computational methods of implementation and analysis. Most importantly, the visual experience can be further formulated with the theoretical background of visual calculating and shape grammar.

2.3.2 Constructive Visual Experience and Shape Grammar

As a formalism theory of design, shape grammar [40] was proposed to work with shapes made of straight lines in 2-dimensional and 3-dimensional spaces. By applying different rules that substitute or transfer shapes, the visual design process can be interpreted with a new kind of visual calculation. Different from symbolic calculating of logic, which is employed by computer systems, visual calculating is about aesthetics and plurality. Human eyes or visual recognition and intuition are emphasized as the power of imagination that enables designers to see things in different ways and in whichever ways they want [12]. This process is described as the "embed-fuse cycle" that perpetuates discovery and rediscovery of creative thoughts.

Shape grammar provides a representation of visual experience. By seeing through their eyes without explicit knowledge or prior memory, designers do visual calculations that develop ideas with a variety of rules in a constructive way. Explicit structures are regarded as the evanescent record of activities and become unnecessary to this ongoing process [12]. Each calculation that applies a rule on a shape contributes to the reflection that accumulatively develops a designer's personal ability. This ambiguous and open-ended representation corresponds to the nature of implicit experience.

40
Stiny, G. (1980). 'Introduction to shape and shape grammars'. *Environment and planning B: planning and design*, *7*(3), 343-351.

12
Stiny, G. (2006). *Shape: talking about seeing and doing.* MIT Press.

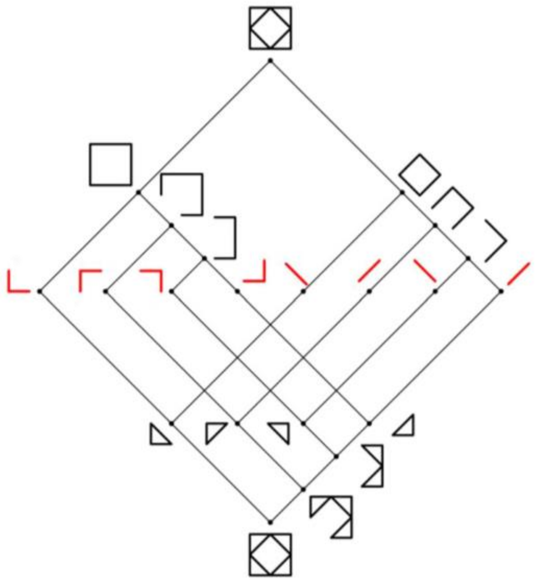


Figure 2-5: Stiny's graph of different descriptions of the same shape [41]

The theory of shape grammar has the expandability that more things and properties can be applied [42], this research focuses on the visual experience of sketches with 1-dimensional lines on 2-dimensional spaces, which refers to Alberti's dimension of architectural design.

2.3.3 Making Visual Experience with Sketch

Sketching has always been one of the primary creative tools in early conceptual design. It is a visual experience-making process that conducts a reflective conversation with materials in a seeing-moving-seeing structure of interaction [43]. Different aspects including designers' working memory, imagery reinterpretation, and mental synthesis are studied [44]. Sketching is also believed as a thinking process more than a record or memory aid [45].

As a constructive way of creative thinking, sketching and the resulting sketches correspond to the ongoing design process and the result of visual calculating. They are also described as *seeing-as* and *seeing-that* [46]. Goldschmidt continued to break the sketching process into smaller moves and related arguments to analyze designers' visual reasoning and concluded that creative production comes from "a special systematic, causal relationship … induced by sketching." From a shape grammar perspective, the power of sketching relates to the ability to see different things and selecting different rules to apply. Each move a

41
Stiny, G. (2015). 'The critic as artist: Oscar Wilde's prolegomena to shape grammars'. *Nexus Network Journal*, *17*(3), 723-758.

42
Knight, T. (2015). Shapes and other things. *Nexus Network Journal*, *17*(3), 963-980.

43
Schön, D. A., & Wiggins, G. (1992). Kinds of seeing in designing. *Creativity and Innovation Management*, *1*(2), 68-74.

44
Purcell, A. T., & Gero, J. S. (1998). Drawings and the design process: A review of protocol studies in design and other disciplines and related research in cognitive psychology. *Design studies*, *19*(4), 389-430.

45
Smithers, T. (2001). 'Is sketching an aid to memory or a kind of thinking'. *Visual and Spatial Reasoning in Design II. eds JS Gero, B. Tversky, T. Purcell. Key Centre Sydney*, 165-176.

46
Goldschmidt, G. (1991). 'The dialectics of sketching'. *Creativity research journal*, *4*(2), 123-143.

designer sketches comes from the implicit experience accumulated over time and reflects new intuitions for the future.

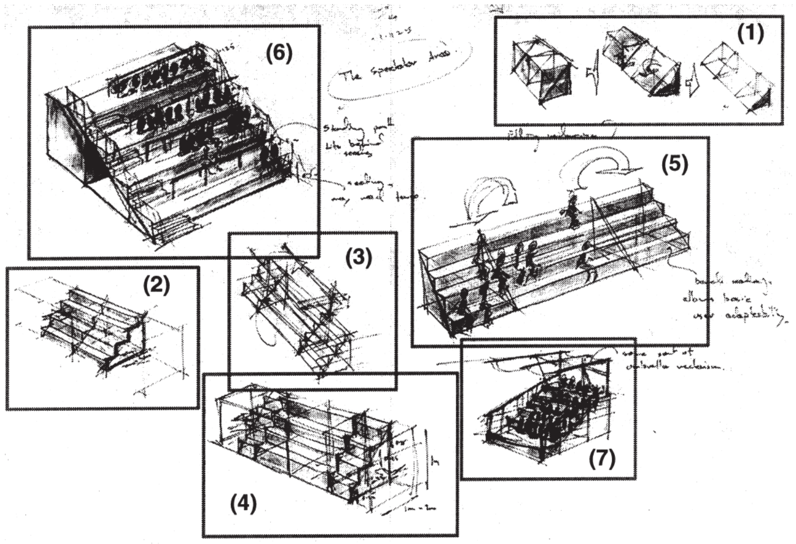


Figure 2-6: Using concept sketches to track design progress [47]

Sketching is also frequently compared with other design tools, especially the CAD system. It is argued that paper-based sketches only have dirty marks on sheets, while computer-based systems have the inherent structures of digital objects [48]. The same as explicit knowledge discussed earlier, structures provide efficiency and productivity when things can be reused and calculated symbolically. However, limitations are set when all the building blocks are constrained within predefined structures and can only be reorganized in combinatory ways. Without the structure, sketching adopts unrestricted visual calculating and releases whatever designers can see with their eyes.

Researchers also studied digital sketching tools as an alternative to hand sketching in the design process. Similarly, computers can make dirty marks on the screen. Though all the data is still stored in structured manners, the display and visual experience are generally acceptable. Information from digital sketches can be converted into verbal clues, drawings, and 3D models [49]. Applications like digital sketch modeling also help with transiting the design process between conceptual hand sketching and CAD modeling production [50]. Digital sketching has unique contributions to visual experience making.

47
Rodgers, P. A., Green, G., & Mcgown, A. (2000). Using concept sketches to track design progress. *Design studies*, *21*(5), 451-464.

48
Sutherland, I. (1975). 'Structure in drawings and the hidden-surface problem'. *Reflections on computer aids to design and architecture*, 73-77. Petrocelli/Charter, New York.

49
Do, E. Y. L. (2005). Design sketches and sketch design tools. *Knowledge-Based Systems*, *18*(8), 383-405.

50
Ranscombe, C., Zhang, W., Rodda, J., & Mathias, D. (2019, July). 'Digital Sketch Modelling: Proposing a Hybrid Visualisation Tool Combining Affordances of Sketching and CAD'. In *Proceedings of the Design Society: International Conference on Engineering Design* (Vol. 1, No. 1, pp. 309-318). Cambridge University Press.

2.3.4 Evaluation of Experience

The tacit experience offers a flexible explanation of design development through constructive practices. However, that flexibility can also make experience difficult to grasp. Because experience is unstructured and ambiguous, designers have no direct access to it and have to wait until it emerges from the intuitive and subconscious mind. There is no way to manage experience, and management is even regarded as inappropriate.

In visual calculating, designers apply different rules as they see, and the most frequently asked question is: what rules should I use next? The answer is "Use any rule(s) you want, whenever you want to." [51] Rules are forged into designers' personalities through experiences and are applied subconsciously without anything priori. Keeping creativity free and unrestricted, experience also makes it hard to articulate and disseminate. In solving satisfactory design problems, experience may not easily lead to design solutions. And in setting open-ended design explorations, experience may still be influenced by factors such as situated context, and designers may not be able to adapt accordingly.

Sketching as a representation of visual experience is one of the most compatible forms for computer applications. However, digital sketching that simply mimics hand sketching has limitations in making experience with computers. The direct transformation from sketches to other formats of information could also be a constraint. Since hand sketching supports the calculation of what can be physically seen and cognitively perceived in designers' working memory, digital sketching should focus on the continuous process of creation where the capacity of visual calculating can take advantage of symbolic organizations. That is to say, the unstructured quality of sketching should be captured and improved by computational tools.

As individual designers, we accumulate both knowledge and experience from a variety of sources in both combinatory and constructive ways. It is rarely the case that we utilize pure knowledge or experience. Is it possible to admire both the logic of knowledge and the intuition of experience, and how can they be connected to develop our unique design personalities?

51
Stiny, G. (2010). 'What Rule(s) Should I Use?'. *Nexus Network Journal.* 13, 15–47

2.4 Making a United Framework

Other than Simon's "building blocks" and Schön's reflective practice, the dichotomy of knowledge and experience widely presents in different theories. Generally, there are symbolic calculating with knowledge and visual calculating with experience, which corresponds to set grammar and shape grammar [52]. Referring to Owen Barfield's theory of figuration [53], shapes as experience are the *original participation* that can be changed afterward, and symbols as knowledge are the *final participation* that stays fixated and unchanged. There are also three types of creativity [54][55], where the combinational creativity utilizes explicit knowledge, and exploratory and transformational creativity relates more to implicit experience construction.

It is then natural to think about joining the two systems together into a united framework of knowledge and experience, and it is possible. Shape grammar already offers a perspective that symbolic calculating is a special case of visual calculating, thus Turing machines are a special case of shape grammars and Chomsky's combinatory atoms of languages are a special case of myriad shapes we see [12]. That is also saying knowledge is a special case of experience. Design originates from the accumulation of implicit experience and finally settles with reusable explicit knowledge. It is always possible to summarize rules applied in the design process retrospectively at the end [56], but the plural constructive procedures that often happen in parallel from the beginning of design complete the whole story.

More than the transformation from experience to knowledge, more directions of the interaction can be explored, especially from knowledge back to experience. It requires the united framework to effectively manipulate both knowledge and experience in a period of time that is long enough to actualize a series of transformations. The framework reflects not only educated expertise, but also continuous active learning that altogether contributes to an individual designer's personal development of design abilities.

---

[52] Stiny, G. (1982). 'Spatial relations and grammars'. *Environment and Planning B: Planning and Design*, *9*(1), 113-114.

[53] Barfield, O. (1965). *Saving the appearances: A study in idolatry*. Harcourt, Brace & World.

[54] Boden, M. A. (1990) *The Creative Mind: Myths and Mechanisms.* London: Abacus.

[55] Boden, M. A. (1994) 'What is creativity?' In M. A. Boden (ed.) *Dimensions of Creativity*. Cambridge, MA: MIT Press

[56] Kotsopoulos, S. (2007, November). Design Concepts in Architecture: the Porosity Paradigm. In *SWW 2.0*.

Chapter 3

A Theory Representing Knowledge and Experience

3.1 What is the Right Representation at All?

Both knowledge and experience systems have been developed in separate approaches by different disciplines. But in order to create a united framework especially for individual designers, there are series of aspects to consider.

3.1.1 Complexity

A majority of knowledge-based systems adopt the semantic network structure, which is originated from the *graph* model. Consisting of nodes and edges, *graph* is a mathematical structure of connected objects. Nodes represent entities, and edges describe relationships between nodes. Edges can be directed or undirected to meet different circumstances. As a fundamental model, *graph* is widely used in constructing different things from simple state machines to complicated semantic webs. Once meanings are applied on *graph* as a data structure, it turns to a representation, such as a semantic network where each node has a meaning and each edge describes a relation. The network model can offer different levels of complexity depending on its end-user, which can be either computers or humans.

In the knowledge representation fields where the purpose is to simulate human reasoning with computers, the corresponding network structure can be an aggregation of simple relationships on a large scale, for example in achieving common-sense reasoning. Those applications are made to automate repetitive tasks that can be easily accomplished by human intelligence, but at a higher speed and especially with a larger amount of data. In the design discipline, human intelligence is still the most valuable to solve ill-defined design problems, and this computer-oriented network structure has limited influence on complicated and realistic design processes.

There are also human-oriented tools based on network models, and the most common examples are bubble diagrams and hierarchical mind maps. Based on cognitive map [57] and concept map [58], these tools are able to handle more complicated organizational and casual relationships that only human readers can recognize. However, they are usually kept as independent clusters on smaller scales so that the cognitive burden to human readers can be well controlled. With constraints on mnemonic capacity, only a limited number of nodes and edges can be processed at the same time. Designers usually employ these tools at the early stage when only a few concepts need to be managed, for example, the spatial relations between functions. As the design project proceeds and absorbs more information, making a network model becomes cognitively expensive and inefficient to designers.

The complexity of how many nodes and edges to be involved is important in designing an appropriate network representation. Oriented to individual designers, the representation needs to be readable and comprehensive to humans as well as expandable and developable to computer systems for long-term accumulation. Human cognition and computer structure need to find a balance on representing knowledge and experience.

3.1.2 Compatibility

Even though explicit knowledge can be adequately represented by an appropriate structure, it is still questionable if implicit experience can or should be included in any kind of computer structure. Does it make sense to host unstructured and ambiguous experience in a determinant graph model anyway? Given the visual experience and sketching as an example, a designer's sketchbook is a linear system of sheets with each sheet contains a couple of sketches. It is by nature a record of ideas but indeed a trace of thinking. Designers sketch as thoughts come to them without constantly referring back to previous sketches. The source of such creativity is not any explicit system, but designers' subconscious accumulation.

In representing experience with computers, the symbolic structure is required, but it is possible to emphasize the experience-making process more than the preserved data. Different from explicit knowledge where nodes and their connections are specifically defined, the computer structure representing experience should focus on reflections that are

57
Huff, A. S. (Ed.). (1990). *Mapping strategic thought*. John Wiley & Sons Incorporated

58
Novak, J. D. (1991). 'Clarify with concept maps'. *The Science Teacher* Vol 58 No 7 45-49.

later turned into designers' own intuition. In accumulating visual experience of sketches, the priority is to enable a streamlined open-ended drawing process that leads to more creative thoughts rather than to save and retrieve precedent sketches. Computational tools should be the aid of consolidating memory, and it is important to model the ambiguity and open-endedness of implicit experience within the symbolic structure.

3.1.3 Transformation

A representation with appropriate complexity and compatibility also needs to allow transformation between knowledge and experience so that designers' development can be fully supported. Educated knowledge or expertise is essential, but can lead to blind rule-following [59]. The openness of experience can be used to introduce more possibilities to the determinant structure. Similarly, implicit experience is not completely unguided, but follows individual designers' professional intuition. Explicit knowledge can potentially activate more experience and create unexpected associations [11].

Knowledge and experience are not separated but different cases of the same thing, and it depends on how human designers perceive as well as how data is stored in computers. The united representation needs to handle transformations between the structured and the unstructured, most commonly the passively acquired textual information and the actively produced visual artifacts. By chaining Barfield's *final participation* of knowledge back to the *original participation* of experience, a cyclic procedure is created to power continuous accumulation and development of design abilities.

3.1.4 Computability

Similar to transformation, both symbolic calculating of computers and visual calculating of human designers should be employed in the united representation. Computers are responsible for circumstances where explicit knowledge is involved, and human designers should utilize visual calculating while making experience. The united representation needs to support and react to both the special and general cases of calculating that correspond to computers and human designers. So that implicit experience may take advantage of computer structure, and

59
Boden, M. A. (2001) 'Creativity and knowledge'. Craft, A., Jeffrey, B., & Leibling, M. (Eds.). (2001). *Creativity in education*. A&C Black. 98.

11
Oxman, R. (2004). 'Think-maps: teaching design thinking in design education'. *Design studies. 25*(1), 63-91.

explicit knowledge can go beyond determinant constraints with human improvisation.

The strategy to create a united representation in this thesis is to start with a knowledge network based on existing theories and expectations, and then construct an experience network on top of it. By integrating both network models, a compatible network based on computational implementations can be created to allow association and transformation of both knowledge and experience.

3.2 Knowledge Network

Creating a network of knowledge for designers has been a continuous motivation in my research and pursuit of more competitive design. Explicit knowledge is understandable and manageable. Introducing new knowledge to a combinatory system is guaranteed to add more possibilities to the pool or space of potential solutions. Imagining designers are able to wire their own memory to a growing knowledge network and have permanent accessibility to whatever they learned, a boost of creative ideas is promised. Such creativity, as defined in this thesis, is combinatory and has an inherent structure to join things together. As determinant as science, things are represented as what they really are. Though it can potentially limit creativity from going unexpected ways, the knowledge network is still a well-established representation as the foundation of expertise.

Balancing between large-scale networks for computers and mapping tools for humans, the goal is to create a representation of explicit knowledge that is compatible with computer operation and intuitive enough for human manipulation. Concept map, semantic network, and ontology all suggest the graph-based network model with nodes and edges an appropriate prototype to be further developed. As the basic elements of a network, properties of nodes and edges need to be specially considered.

3.2.1 Nodes as Knowledge Entities

Nodes are the elements that represent static entities in network models. They are the circles in bubble diagrams indicating different spaces and functions, as well as nouns in semantic networks symbolizing distinct objects. Nodes are the independent unique representation of different knowledge.

In simple network models such as mind maps, nodes are uniform holders of their own content. As networks grow in scale, nodes can be conceptually classified as different types and therefore provide extra information. A network of building elements can describe all the components and relations involved in the design process, and classifying those elements into categories such as interior and exterior space makes the entire system more manageable. This categorization can be achieved by assigning different attributes to each node. However, the categories specified to the network do not have an upper limit thus can lead to over-complexity. In extreme cases, labeling a node with too many properties or having each node falls into a different category does not contribute to an efficient representation. It is important to define properties that are universally compatible and do not scale upon the growth of knowledge.

The idea of knowledge classification helps with such a definition. Cognitively, we can tell if things are abstract or concrete, which relates to conceptual and episodic knowledge. Conceptual knowledge corresponds to semantic memory [27], which is used in managing symbols, concepts, and relations in languages [60]. Conceptual knowledge is usually verbal and describes generalized statements. Episodic knowledge corresponds to episodic memory, which handles temporal-spatial relations among chronological events [59]. Episodic knowledge portraits series of specific information not limited to verbal, pictorial, and auditory material. For example, the concept of *chair* is the knowledge that symbolizes a wide range of things to sit on, and *Barcelona Chair* is a specific episode or instance that relates to the concept of *chair*. It is argued that *Barcelona Chair* can also be regarded as a concept due to its popularity, and it depends on the context, or what the creator of the network believes. It also reveals the fact that there exists transformation between conceptual and episodic knowledge depending on their importance to the knowledge user.

The advantage of using the dichotomy of conceptual and episodic knowledge for node types is to cover as much design knowledge as possible with a simple and intuitive classification. Different from exhaustive labeling systems where different tags of attributes are used to identify each element, nodes are labels or tags, and nodes are used to describe other nodes. The growth of knowledge contributes to not only the increased number of nodes, but also more approaches to manage and associate. Concepts can be first learned and followed by

27
Quillan, M. R. (1966). *Semantic memory*. Bolt Beranek and Newman Inc Cambridge MA.

60
Tulving, E. (1972). 'Episodic and Semantic Memory'. In E. Tulving & W. Donaldson (Ed.) *Organization of Memory*. 381-403. Academic Press.

complementary episodic details, or specific cases can be captured in advance and eventually lead to the construction of a conceptual framework. Individual designers therefore are able to aggregate knowledge in a variety of paths over time.

3.2.2 Connections as Knowledge Flows

Describing relationships between nodes, edges are the other basic elements of a network model. Considering the simplicity of representation and future database design, connections instead of edges are used as attributes of nodes in this research. It allows multiple types of connections to existing between a pair of nodes since the simplest undirected edges are not able to describe complex relations.

The bare connection between two nodes can be either directed or undirected to represent a single or bi-directional relationship. In more complicated networks, connections also have their own properties. For example, a semantic network employs *is_a* and *part_of* connections to construct meaningful structures. As more complex meanings are involved, more connection types are likely introduced to articulate a variety of relations, which can potentially over-complicate the entire network. Covering as many relations with fewer types of connections is essential for human readers.

One knowledge classification that can be used to differentiate connections is declarative and procedural knowledge. Sometimes regarded as the combination of semantic and episodic memory [19], declarative knowledge represents the symbolic storage of facts, events, and their associations [61]. Those facts and events are usually described in verbal languages, and since languages have their inherent syntactical structure, declarative knowledge can also be recursively composed or decomposed into larger or smaller elements. Similar to the *part_of* connections in a semantic network, declarative knowledge depicts a spatial relationship of containing, such that a word is a part of a phrase and a concept is a part of another concept. Also known as the knowledge of skills, procedural knowledge is believed to express only through direct performance [19], which in this thesis corresponds to implicit knowledge or experience. In order to represent and connect to the explicit knowledge system, procedural knowledge can also symbolize the temporal relationship of causality, such that an event leads to another event and an action followed by another action.

19
Tulving, E. (1985). 'How many memory systems are there? '. *American Psychologist*, *40*(4), 385–398.

61
Broadbent, D. (1989). 'Lasting representations and temporary processes'. In H.L. Roediger, III & F.I.M. Craik (Eds.), *Varieties of memory and consciousness*: *Essays in honor of Endel Tulving.* 211–227. Hillsdale, NJ: Erlbaum.

The duality of declarative and procedural knowledge corresponds to Ryle’s *knowing-that* and *knowing-how*, and it covers a wide range of different connections in a knowledge network. Both types of connections are directed to clarify the flow of space and time. For example, going from concept *furniture* to *chair* to *armchair* is a declarative process in the positive direction where each step makes the system more explicit. In addition, the rudimentary undirected connection, which is named neutral connection, can be used as a complement to cover any relation that is not yet declarative or procedural. Altogether, the network with three types of connections is more powerful than the semantic network and causal map combined in representing associated knowledge.

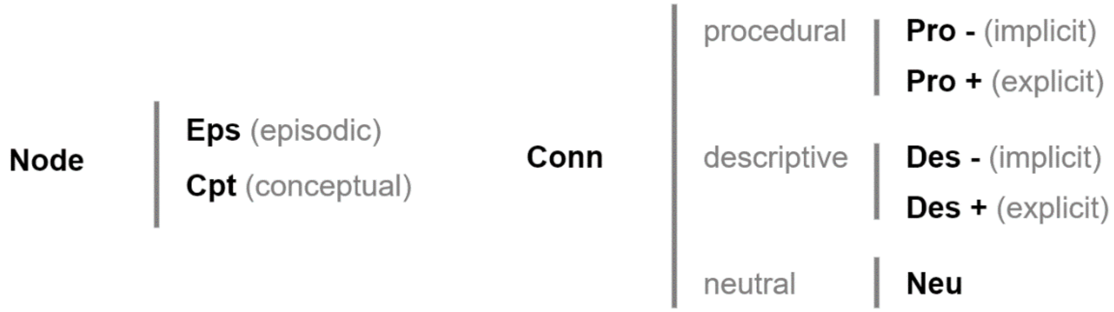


Figure 3-1: Different types of nodes and connections

3.2.3 Building a Knowledge Network

With multiple types of nodes and connections, the network of explicit knowledge is more than a graph or a semantic model. Designed for accumulating human knowledge over time rather than fetching a large amount of data for computer use, the knowledge network utilizes reliable and efficient symbolic structures to augment human knowledge accumulation. Based on human memory systems, the network provides an intuitive representation of learned knowledge so that designers are able to incorporate the network-building process into their learning and practicing.

Different from strict knowledge representation used for determinant inference, the knowledge network is flexible to human intelligence and allows ambiguity as a temporarily unresolved state within a dynamic, always-in-progress accumulating process. It is an external memory and cognition of learned knowledge as well as a computational interface of knowledge utilization.

It can be arguable that if the personal knowledge network is necessary given the fact that large public knowledge providers such as Google and Wikipedia have already linked the world's information together. The biggest problem of these public networks is not accessibility and diversity but selectivity and priority of knowledge that individuals can recall at no cost. Personal knowledge network allows systematic and accelerated retrieval of learned knowledge with its comprehensive association that public networks do not provide. The customization that fits individual interests and development is also valuable for designers.

3.3 Experience Network

The purpose of the experience network is to support designers' accumulation of implicit experience and skills rather than articulate specific content and relations. The representation should offer flexibility rather than constraint. Symbolic calculation and computer structure can be used to facilitate experience making, and designers' cognitive development should be particularly emphasized.

Different from explicit knowledge, giving implicit experience a structure can be tricky. Though it is difficult to explain how designers accumulate experience over time, the network model can still be a promising estimation. Procedural knowledge and *knowing-how* already suggest that experience is chronologically distinguishable, which means the experience network can be an aggregation of chains of subdivided and related actions. The real problem is that how explicit should the network be so that it can be represented computationally and at the same time has the minimal limitation on open-ended creativity.

Giving the visual experience as an example, a sketch can be regarded as a piece of drawing at its final state, or it can be decomposed into strokes that altogether constitute the drawing. A network of finished sketches works as a digital alternative to physical sketchbooks, which only indicates the *final participation* of knowledge instead of experience making. However, a network of decomposed moves of a sketch can be over-detailed and inefficient to grasp since each stroke contributes little to the development of thoughts. Designers sketch as their ideas are formed in an artistic way, and it is important to consider what to keep and how to influence the future experience-making process.

3.3.1 Sketches as Encapsulated Experience in Nodes

Sketching is a thinking process, and the representation of visual experience needs to reflect how thoughts are constructed and accumulated. Goldschmidt [46] described design *moves* as the elementary coherent operations in design supported by *arguments* which are “the smallest sensible statements”. The design process as sketching can then be regarded as architectural reasoning with a series of moves and arguments. However, this process indicates an explicit system in a combinatory viewpoint and the retrospective arguments may not clarify the visual experience when the sketch is first constructed. Goldschmidt [45] concluded that it is *interactive imagery*, a continuous visual dialectic between *seeing-that* and *seeing-as* that inspires new thoughts through sketching. Visual calculating and shape grammar [12] make it even clearer by introducing rules as the representation of seeing and doing. When applying rules, designers create new experiences from one idea to another.

Focusing on rules, experiences before and after a rule is applied can be differentiated, and those experiences are represented as sketches along the design process. Even rules can be represented as sketches as well. In the experience network, sketches become the nodes that can be associated to describe design processes. Considering an implicit representation system, sketches can also be the aggregation of applying multiple rules depending on designers’ choices. There are sketches of the start and end states of the design process as well as intermediate steps of applying different rules and thoughts.

46
Goldschmidt, G. (1991). 'The dialectics of sketching'. *Creativity research journal*, *4*(2), 123-143.

12
Stiny, G. (2006). *Shape: talking about seeing and doing.* MIT Press.

3.3.2 Visual Experience Making

As an ongoing process, the visual experience of sketching can be represented as an expanding network. Each node corresponds to a sketch in the process, and the temporal causal relationship between nodes can be captured by procedural connection.

Different from the explicit knowledge network where nodes and connections are elaborately specified by designers, the construction of the experience network has minimized distraction and automatically collects information from the sketching process. A new node of sketch will be created whenever a rule is applied or a move is made by the designer, and a new connection will be built from the old node to the new one. A series of nodes will chain up and constitute the path of visual thinking, and plurality is also supported when multiple ideas branch from an old sketch. A network of experience can then be constructed while sketching, thinking, and designing. Stiny [62] put design as “an element in an n-ary relation among drawings, other kinds of descriptions, …” The relational network of sketching represents the design process in a constructive way.

3.3.3 Building a Subconscious Experience Network

The experience network preserves design thinking process as connected sketches, but it can be arguable if such an explicit record of implicit experience can help with future design rather than constrain it. The network can temporarily store the sketch data when a designer firstly creates it, and the question is whether and how this data should be kept for future retrieval.

Experience is implicit because it is difficult to specify and articulate, and the human brain runs on a mechanism remains unknown to invoke precedent experience. Since simply giving a definite structure may hurt creativity, the experience network itself can be unspecific and implicit. Different from the conscious knowledge of text, the subconscious experience of sketch cannot be accessed through active search operation, but can be passively referred to as alternative thoughts and mnemonic reinforcement. The purpose is to facilitate designers’ consolidation of experience with more exposure to precedent sketches, so that they are able to incorporate and recall more rules in future design. Such a latent network works as a computational approximation of the implicit experience mechanism employed by human designers.

3.4 Integrated Network

Having the representation of both knowledge and experience, it is possible to create an integrated network by transforming between final and original participation, symbolic and visual calculating. The transformation from explicit knowledge to implicit experience represents the creation of new thoughts as sketches from designers’ own inspirations of textual and pictorial information. The transformation from implicit experience to explicit knowledge represents the generalization of ideas embedded in sketches. Being able to handle both knowledge and experience, the network can be used to

[62] Stiny, G. (1990). 'What is a design?'. *Environment and Planning B: Planning and Design*, 17(1):97 103.

facilitate the development of design abilities. More accumulated knowledge contributes to more robust combinatory ideation, and more associative experience leads to more streamlined constructive creativity. The representation corresponds to design as the combination of drawings and symbols [61], and indicates a possible computational augmentation to design and creativity.

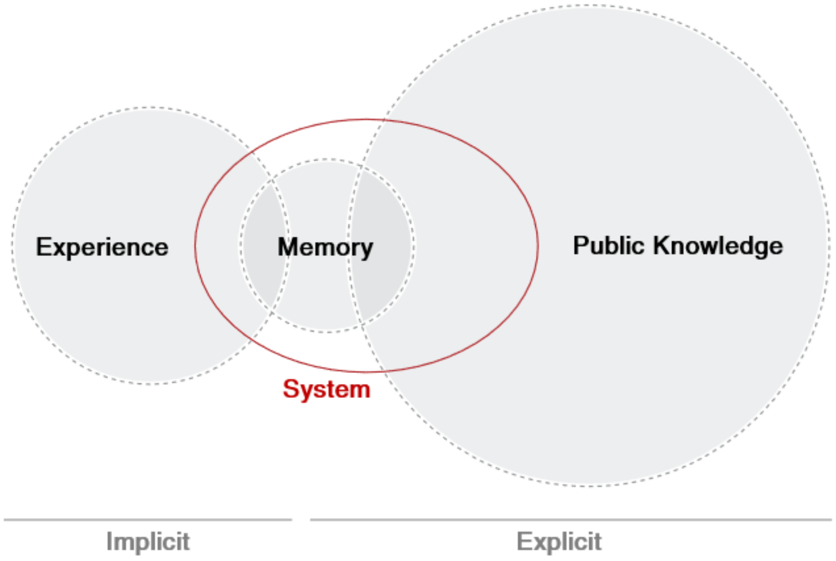


Figure 3-2: The proposed system covers both knowledge and experience

Chapter 4

An Instrumental System for Designers

4.1 Why Need a System?

In order to implement such an integrated network of experience and knowledge, a designated computational system need to be designed and evaluated. It is necessary to prove that there are no existing tools and systems that can support the theoretical representation of knowledge and experience.

4.1.1 Information Externalization

Historically, we have invented innumerous different tools, and one of the most critical inventions is the writing system that separates humans from other species. Because of the technology of externalizing thoughts from the brain, human ancestors were able to accumulate knowledge beyond their mnemonic capacity and pass it to future generations. Encoding information in paper-based or electronic material, the symbolic system of writing still benefits us today and creates a world filled with all kinds of knowledge. In order to acquire and utilize the knowledge, individual of us not only use our internal memory, but also construct personal externalization of knowledge by creating physical records or taking notes as artificial memory. This note-taking behavior can further expand into a conversation or reflection to oneself, repetitively, as described as *hypomnema* by Plato and *personal writing* by Foucault [63]. Either writing on a piece of paper or typing with a laptop, we take notes as a learning aid that customizes external knowledge for personal understanding.

4.1.2 Limitations of Our Current Tools

Usually, the notes are organized so that the embedded knowledge can be recalled and used in the future (the use of notes without organizing

[63] Foucault, M. (2005). *The hermeneutics of the subject: Lectures at the Collège de France 1981--1982* (Vol. 6). Macmillan.

is considered as experience, such as napkin sketch by designers), and the structure of notes is highly influenced by the materiality of media. A piece of paper contains paragraphs of text, and so do digital pages or canvas.

Paper-based or digital tools range from organizational knowledge bases to personal wiki systems all adopt an article-based format to convey ideas, which are comprehensive but not flexible enough to host network models. Writing as a linear aggregation of content in different languages can be further decomposed into concepts with related meanings.

There are also index card writing systems such as Zettelkästen (slip box) invented by German sociologist Niklas Luhmann that break down passages into smaller pieces as cards and focus on building relations between cards. Those systems describe an organization of episodes that are connected procedurally, and it can still be improved when a layer of concepts is introduced to provide extra associativity.

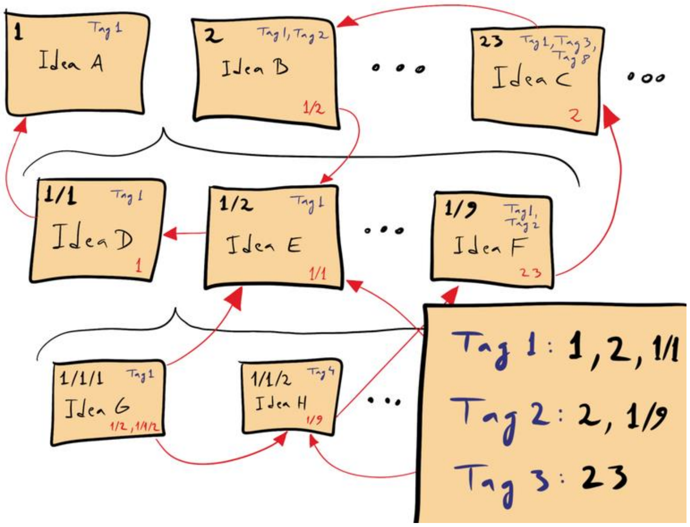


Figure 4-1: Zettelkästen paper schematic relations.
Accessed on May 12, 2021, licensed under the Creative Commons
https://commons.wikimedia.org/wiki/File:Zettelkasten_paper_schematic.png

Mindmap and other concept mapping tools have a similar appearance with network models, but are usually on a smaller scale so that users can fully navigate through and absorb all the content. Each mind map is treated as a complete drawing with limited further updates and an isolated piece with a lack of interconnection between maps. The network representation is more like an aggregation of mind maps with more specific attributes and is constantly changing so that each concept can be regularly reviewed.

4.1.3 Limitations of Public Knowledge

It can be arguable that the Internet services have already provided us a web of information and to build a personal one is unnecessary. The problem of public networks is not the quantity but the priority and efficiency. Those public networks are centralized and fixated, which require individual contributors to come to a single agreement. However, each individual utilizes knowledge in a unique way, and personal networks can be shaped according to different users. The knowledge included in a personal network is more valuable because it is filtered and processed intellectually by its user. Not all knowledge can be found from public sources – instead, it usually comes from exclusive channels. Personal networks balance those sources by making customized copies of known information and filling the gaps that public networks cannot reach.

4.1.4 System Goal

It indicates that a computational tool for constructing a knowledge network at the personal level is still to be developed. Furthermore, the incorporation of implicit experience adds new possibilities to a novel system. The development of design abilities or design education is usually a studio-based apprenticeship-oriented process that involves mostly subjective hands-on activities. There is barely a tool that can offer such an experience other than the activity itself, even a comprehensive knowledge network. To facilitate experience-making is to preserve it as much as possible in a compatible approach for maximized performance. For visual experience, a digital sketch may not be able to realize the same level of flexibility as a hand sketch, but it can preserve the thinking process of sketching in terms of visual calculating as well as employ network structures as potential resources and connections to explicit knowledge.

The system tool is integrative and instrumental as it prioritizes human participation as the process and designers' development of abilities as the result.

4.2 Workflow

As the augmentation for designers, a computational system utilizes its strength to extend human limits, including finite mnemonic capacity, limited attention, and restricted ability to respond quickly. In the world of explicit knowledge where rigorous logic and symbolic calculating outperform human efficiency and productivity, the system can support designers with a natural and reliable workflow of input, storage, and retrieval, as the extension of their own cognitive abilities. With diverse and accessible knowledge, designers can not only think in combinatory ways, but also inspire more experience-making activities constructively.

The input-storage-retrieval workflow contributes to a cycle of learning and utilizing in designers' personal development by maintaining a dynamic network of connected knowledge and experience. Acquiring various forms of information from all kinds of sources, the input process is open and inclusive to absorb any learned knowledge and sensed experience at any time. Those acquired can be securely stored and particularly wired up in a precise yet flexible network. Upon request, they can be responsively retrieved to solve problems or unexpectedly brought up to trigger new thoughts. The workflow is a union of both human and computer for the purpose of more intellectual and creative design.

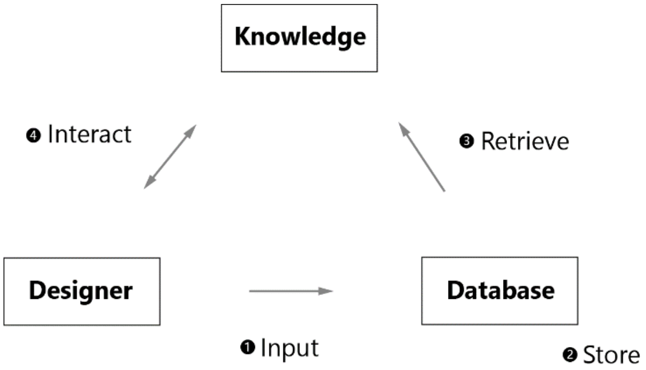


Figure 4-2: The input-store-retrieve workflow that enables interaction

4.2.1 Input: Instantaneous, Ambient, and Integrated

The *input* process is the portal of all explicit knowledge. Different from writing or typing with a separate medium, *input* should be an instantaneous process with the least effort, as if using one's own memory. The valuable knowledge and experience may appear at any moment in any context, and capturing them with minimized distraction is essential. It also suggests that *input* is an ambient process that can be accessed from anywhere alongside the ongoing focus or action. Designers learn a wide range of knowledge from a variety of sources and come up with different ideas while interacting with different forms of information. The input process can handle textual and pictorial knowledge as well as sketching experience as nodes in constructing the network.

4.2.2 Storage: Network Representation and External Memory

Corresponding to the representation theory, storage contains all collected nodes of knowledge and experience, including their content, attributes, and associations. When precision and fidelity of knowledge are required, symbolic calculating in terms of computer structure can provide reliable supports in constructing an external memory. Such memory is not the alternative to human memory and instead an augmentation. Improved performance of human-computer collaboration requires both types of memory to work together and constantly transfer from one to the other. Storage also takes the burden of keeping some implicit experience such as sketching in order to reflect back to designers in the future.

4.2.3 Retrieval: Responsive, On-demand, and Unexpected

The retrieval process can be responsive as soon as a signal of inadequate information arouses. Utilizing the network structure, there should be multiple paths to retrieve each node and more directions from the node to explore even further. The computational system can perform more retrievals than humans at the same time and works as external working memory. The retrieval process can not only respond to active inquiries from designers, but also provide passive reminds and unexpected inspiration.

4.3 Software Structure

The implementation of the workflow and system primarily consists of the interface that communicates with designers, the database that stores all information, and the background logic modules that realize different functions. The implementation in this research is a prototype software written in Python with SQLite database and wxPython module for graphic user interface (GUI).

As a stand-alone multi-function program, the software consists of different modules including libraries for functional programming that handle basic calculation and language processing, as well as object-oriented GUI modules for each interface. User configuration is kept in separate files for easier customization, and an independent file system is maintained in collaboration with the database. A hierarchy is described below to draw a big picture of the software structure.

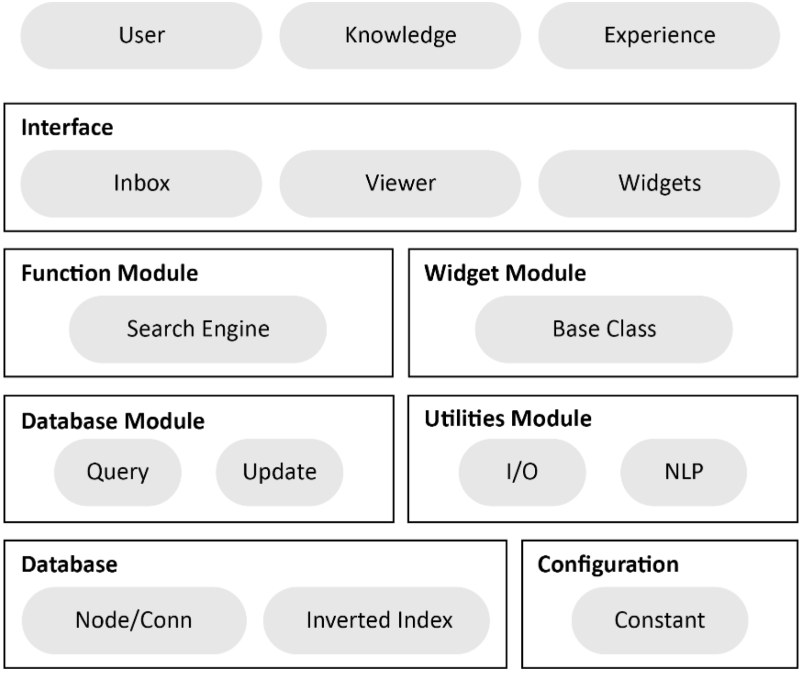


Figure 4-3: Software architecture diagram with modules

launch module is the top-level and entry point where the software initiates and all other modules are referred. Constructing a GUI application that always listens to user input and events, the root class sets up global settings and registers all GUI interfaces.

config module reads the user-customized configuration file in JSON format and converts it into objects that can be directly referred to by other modules. It includes class-wise graphic information such as dimension, color, and font for each interface, as well as predefined directory and command.

The *lib* package includes modules that backup essential calculations.

utilities module is a collection of functions that handle system-level operations such as I/O (input and output), data structure, customized computation, as well as third-party powered operations such as image rendering and language processing.

database module is a collection of functions that make queries and updates on the database, including all nodes, node attributes, and connections. It is also a *graph database* implementation on top of a conventional *relational database*. Other database-dependent functions are also included such as search engine. The database file stores all information of the network, including nodes, connections as well as keyword indexing and statistic information.

functions module consists of higher-level functions that can be called directly from interfaces or the operating system, including interface utilities and file manipulation.

The *gui* package includes modules that construct interface widgets.

widget module defines all basic and advanced interface classes with inheritance. Those classes are used to construct top-level interfaces such as *inbox* and *floater*.

inbox module constructs the *inbox* interface, which is the portal of all user input. It also includes related classes such as *tag* and *suggestion* that aid user input experience.

viewer module constructs the top-level *viewer* framework, which hosts sub-level interfaces such as *floater* that displays retrieved information from the database. A series of nested interfaces that derived from *floater* are also defined such as *inventory* and *entry*.

There are also directories maintained by the software to access external resources.

The *asset* directory includes graphic and multimedia resources for the interfaces.

The *rsc* directory includes cache and backup files maintained by the system.

With those modules, the software is able to construct and maintain a network of knowledge and experience as a separate but integrated augmentation with designers' own knowledge. The software structure also allows rapid development of new functionalities along with the research.

The following sections explain in detail how each module is designed and actuated.

4.4 Database Design

Related to how information is stored and retrieved, the network of knowledge and experience resides as organized data in the database. The design of the database defines the basic structure of operations and future functions derive from it. This research employs SQLite database, which is a relational database on local storage and is widely used for small and mobile projects. Oriented to individual designers' accumulation over time, the system does not need to face any high concurrency environment, which is common in web-based platforms. The storage occupied can be maintained on a relatively small scale in the local environment. There are also potential benefits to look into the graph-based database in terms of increased efficiency in searching algorithms.

4.4.1 Database Structure

In the network, nodes are the basic entry of information. Each node representing knowledge or experience contains not only its content but all other different attributes that make the entire network function. Relational databases depend on a data structure called *table* to record each entry and its attributes by rows and columns. This database contains four tables to store node information, including:

Node Table, contains basic information of the content, node type, and time stamp. Content is the textual information describing each node, node type declares conceptual or episodic nodes, and time stamp records the Unix time when the node is created. Each node has a unique id number so it can be referred to by other nodes without undesired ambiguity. A customized node label is found helpful by giving extra priority especially for sorting. Node source stores the affiliation with an external file system so resources can be called as necessary. Each entry also contains a field called operation, which can contain a series of customized commands and make each node programmable. The data type of each attribute can be integer, text string and JSON string depends on its corresponding function.

Connection Table, contains all the associations with other nodes. Sharing the same id number, each node has ten different attributes to describe its connectivity. Those attributes come from the knowledge network theory and include *neutral*, *descriptive*, and *procedural* connection types. In describing special and temporal relationships, *descriptive* and *procedural* connections have *positive* and *negative* directions indicating the flow of knowledge. Considering each type can be applied to both conceptual and episodic nodes, a total of ten different connection types are formed. The id numbers of connected nodes are stored in each connection type as a textual string and can be retrieved to read the connections.

The ten connection types are also called unitary connection types that describe a single kind of connection. More meanings can be constructed when multiple unitary connections are detected for the same node id. For example, if two nodes are each other's *procedural* connection in both directions, they form an *equivalent* relationship in terms of one leads to the other and vice versa. These kinds of combinatory connection types are able to convey more complex relationships.

Significant Dimension Table, contains statistic information of connections that can be used to measure the importance of a node. It also includes counters that record how many times the node is referred in search and other operations, and can be used as a factor of sorting.

Keyword Table, contains the reverse-indexed content in respect to their id numbers and is the basic mechanism of a text-based search engine. With specific input of content, related nodes can be found to fulfilled search inquiries from users. It also supports a suggestion function that helps with easier input of tags. Statistic information of search is also stored in the table.

With all information stored in the database, real-time read and write operations can be made by the *database* module to fulfill other advanced functions.

4.4.2 Conscious Knowledge Network

Connected nodes aggregate and become a knowledge network. With Each node and connection made by a specific designer for a particular purpose, such a network represents the personal understanding of things and their relations. Because nodes are created as inseparable containers of meanings, the network is also a combinatory system. As nodes are putting together and apart, the system describes what things really are in the particular designer's perspective.

In the database, such a network is maintained as a collection of rows that contain node information. Tied through their content in the

Keyword Table, all those nodes can be retrieved by calling their content. Starting from any node, more nodes can be reached through its connections in different levels for different purposes, either by direct browsing or searching algorithms. This network is said to be conscious as it embeds explicit human decisions of input.

4.4.3 Subconscious Experience Network

In order to cover implicit experience and express as many human exclusive abilities as possible, the network can be made subconscious by manipulating nodes of experience. Those nodes, user-created sketches as in this thesis, contain visual data instead of verbal information and cannot be accessed through active searching. The connections between them are not constructed through explicit operation either, but are captured through the drawing process when a new copy is made or a rule is applied.

The experience network of sketches is stored in the database as well and it indeed has an explicit structure so that a symbolic computer system can manipulate it. The implicity comes from two sources. Firstly, the nature of visual experience guarantees it cannot be represented in explicit ways. A sketch can only connect to verbal information but cannot be fully transcribed into a verbal form. So that at the *final participation* circumstance, a designer can only see and imagine the content of a sketch by linking concepts to it. Any other experience prior to this finalization can be completely implicit to human designers in the design process without any predominating concepts. Secondly, the system renders implicity by limiting access to certain nodes in order to prevent cognitive routine or fixation on them. In other words, the network is symbolic preservation of the subconscious aspects of the design process, and some of the experience of sketching is intentionally obscured to reduce the constraint of ongoing creativity.

The database manages pieces of knowledge and experience in symbolic ways so that the non-symbolic abilities of human designers are prioritized. In order to connect computer mechanisms and human cognition, an interface for communicating and instantiating knowledge is critical.

4.5 Interface Design

Bridging human thoughts and computer data, the interface system is responsible for the constant input and retrieval of knowledge, especially for intensive design work where ideas usually flow all over the place. In order to fulfill the workflow and integrate it into the design process, the system is designed to challenge the interaction manner of standard applications. Those features include:

The interface is *ambient* rather than interruptive. Most modern applications are contained inside a single window or frame, which is good for concentrating on a specific task but can be distractive when switching between multiple applications is needed to complete a workflow. On the contrary, this interface weakens the presence of a clearly defined window that occupies a large screen area by having multiple smaller widgets floating on top of any other applications. Those ambient widgets provide a versatile representation of learned knowledge and allow a continuous process of learning and design.

The interface is *responsive* rather than *redundant*. Applications with a complicated structure always have a longer path to direct users to the wanted functionality. In order to integrate deeply into the ongoing design process, the interface responds instantaneously to the users whenever an idea comes to mind and completes the input and retrieval workflow with minimal steps and time. The interface provides a streamlined experience that human designers and augmenting systems collaborate on the knowledge network.

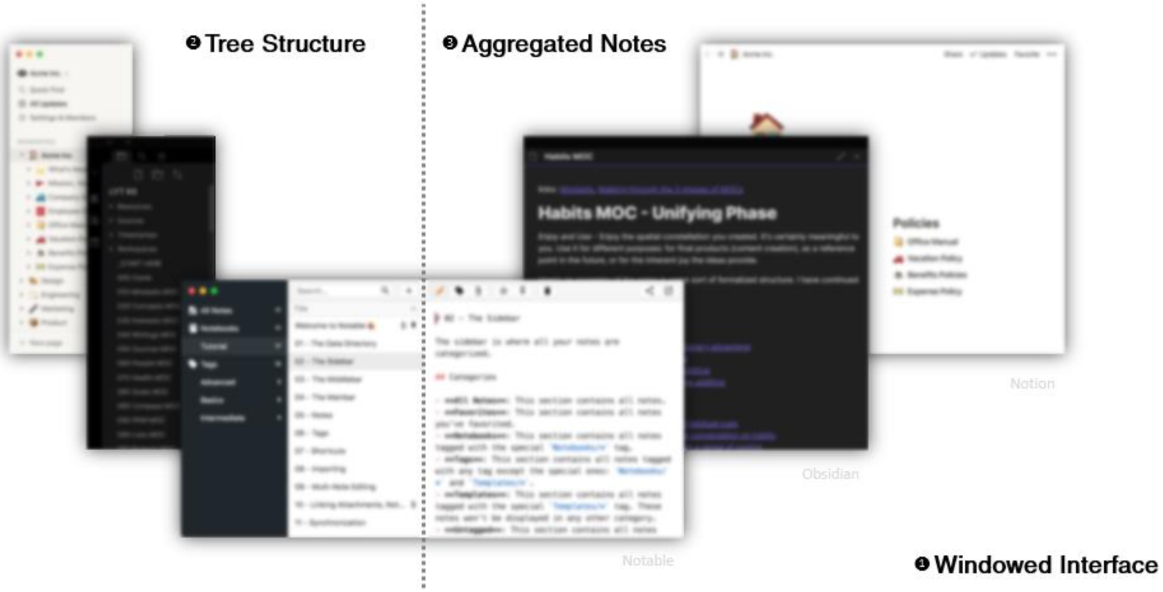


Figure 4-4: Classic interfaces with side panel and tree structure logic

The interface is a representation of a knowledge network as well as a manageable graphic interactive system. Most present applications employ a tree-based structure with a hierarchy of elements. It has advantages in managing a relatively small cluster of data but becomes difficult to navigate through a large number of entries in the scale of all learned knowledge. This interface is designed to dynamically present a knowledge network from any nodes. It also provides the flexibility to create all kinds of data structures, including graphs, trees, and lists. The interface aims to share the user's cognitive burden of using knowledge.

Currently, the interface system includes a set of fundamental components. They are:

inbox for initiative input and query,

viewer for display and further update of retrieved knowledge, and

other *widgets* to support different forms of manipulation including image rendering and sketching.

The *sketch widget* is particularly important as it enables visual ambiguity and the capture of design experience beyond explicit knowledge. Those interactive components together connect human designers to computational augmentation and expand their abilities to learn and create.

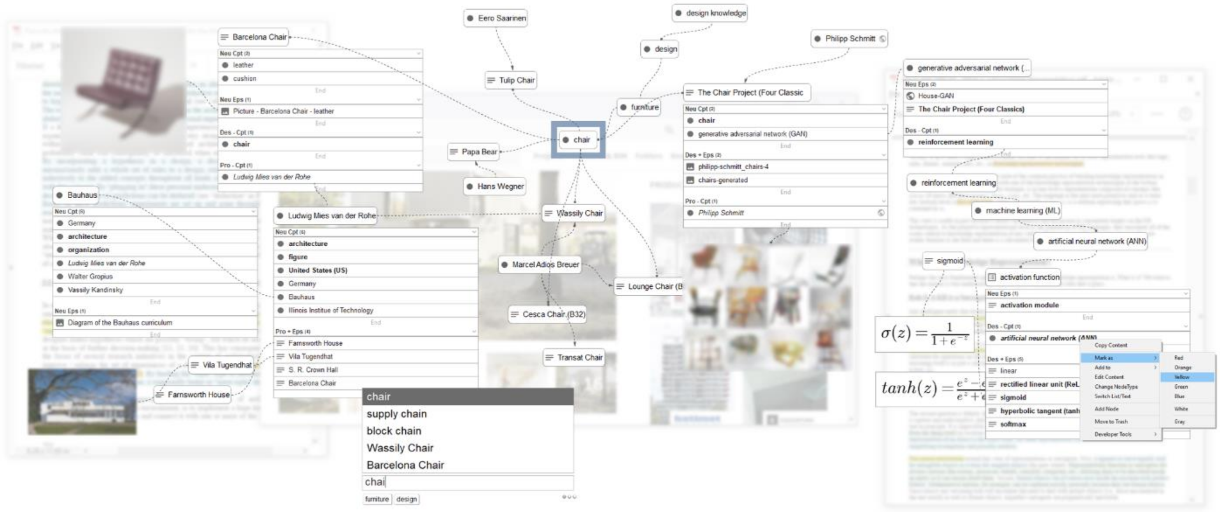


Figure 4-5: Interface illustration of inbox, viewer, and widgets

4.5.1 Inbox: Portal of Input and Inquiry

The *inbox* interface is the portal for initiating the input and query when a user tries to contribute content to the system or looks up for learned knowledge that is already maintained in the system. More than a simple input area for typing, the interface is also responsible for giving feedback to the user's input and providing automated aids to accelerate the input process. This idea comes from the quick search box which is included in many software for convenient querying purposes and is expanded in this system to efficiently handle all kinds of inputs. The interface is independent of other applications on the screen so that the user can access it at any time. It includes three different parts: *input frame*, *suggestion frame*, and *tag frame*.

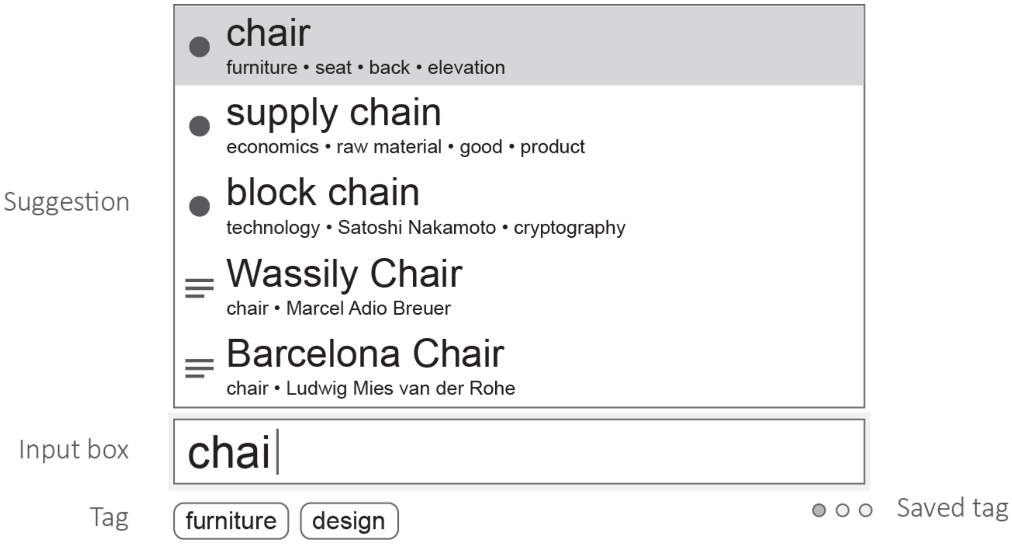


Figure 4-6: The inbox interface with input box, suggestions, and tags

As the primary part of the interface, the *input frame* is a floating window with text entry functions that appears on the screen. Following the ideas of an ambient and responsive interaction, the input frame is placed at the peripheral area and can be shown and hidden as needed.

As with other search boxes, the *input frame* responds to the user's input by one character or letter to another and shows suggestions as the software searches through the database to match proper records (more details in the search engine session). Those records, as rows of node information, are displayed in the *suggestion frame* and can be selected and picked out by the user. The picked entries, along with all the information, appear as tags in the *tag frame* and can be used as parameters when searching through the database or creating new nodes.

In making a search query, the user can quickly access prior knowledge that has been indexed by using a combination of tags and keywords. In creating a node record, content and connection information can be passed as text and tags to quickly build up the network.

The interface is designed to accept different formats of information and work with the computer file system so that a universal top layer of information input can be created to manage knowledge from different sources at the same place. That is to say without dozens of windows and pages that each contains isolated information, the input interface is the only entrance to the united knowledge network where things are wired up. There are also functions such as batch mode and drag-and-drop to streamline the input experience with fewer restrictions.

4.5.2 Viewer: Extension of Memory and Thinking

The *viewer* interface is designed to display and edit retrieved knowledge. It utilizes a collection of different node symbolizers called *floaters*. Namely, those elements float on top of other applications on the screen and can be freely moved and grouped. They are created from inbox query or spawn from the traversal of connected nodes. Each floater representing a node can also show its connected nodes in an appended component called *inventory* as an expandable list. Each type of connection is display in a separate *pane*, in which nodes are represented as a list of *entries.* This separation makes it easier to make sense of different connected nodes in neutral, descriptive, or procedural relationships. For example, *Des + Eps* symbolizes the episodic nodes the floater connects in descriptive relation on positive direction (or simply, nodes it contains) and can be used to describe the instances of a concept, such as cases of a design style. Each floater or entry contains a collection of attributes for the node it represents, including types of the node, associated source information, and programmable command. By creating and browsing through different floaters, the thinking process of designers can be supported by reliable and responsive interaction with learned knowledge.

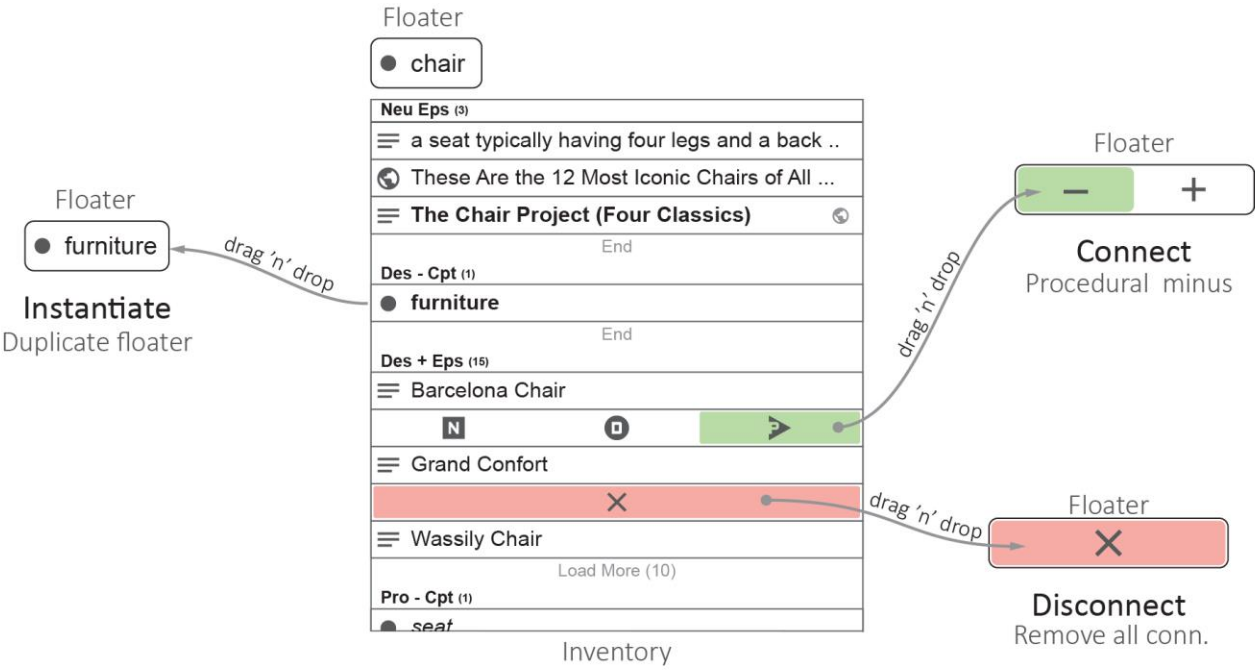


Figure 4-7: The viewer interface with inventory and connecting functions

Dislike standard applications which are contained within a single rectangular window, floaters greatly increase the density of information on the screen and create a virtual working memory that shares the cognitive burden of thinking. Each *floater* and *entry* represent a thought or a piece of learned knowledge that is valuable for the personal accumulation of designers. At any moment and regardless of the running applications on the screen, learned knowledge can be summoned from the system without interrupting current operation, as if retrieved from biological memory. Each floating node can be freely placed on the screen as a hook of memory and attention when an exceeding amount of information is involved in a thinking process. Previously acquired design knowledge can then be efficiently utilized.

The viewer also provides further editing abilities to dynamically update the knowledge network. Attributes of nodes such as content and customized labels can be modified in place, and different types of connections can be easily updated through simple drag-and-drop operation between *floaters* and *entries*. There is no barrier that can stop the system from growing when new thoughts and different opinions emerge, and the knowledge network should reflect the personal development of a designer through learning and experiencing.

4.5.3 Widgets: Making More Experiences

Supporting more formats other than verbal and symbolic information, different *widgets* are designed to represent visual knowledge and experience. There are two types of *widgets*:

The *preview widget* is used to display static pictorial knowledge, such as images and mathematical formulas. When visual details are necessary, previously-stored images can be easily retrieved from associated nodes. Similar to *floaters*, the *widgets* provide an independent layer of information on top of other applications. Designers are able to juxtapose visual information with embedded knowledge on the screen to hook memory and facilitate ideation.

The *sketch widget* is used to support the active experience of sketching. Inspired by a shape grammar exercise using tracing paper, the *widget* is designed as pieces of translucent digital canvas. Designers who tend to express their ideas visually can sketch and apply visual calculations on the *widgets*. Even though based on the symbolic computer structure, those widgets simulate unrestricted experience-making process using real pen and paper. Different from object-oriented CAD drawings with innate structure, those digital sketches prioritize what designers can see and are truly "making dirty marks on the" *screen*. [12]

In order to streamline the sketching experience and provide extra flexibility, some novel functions are designed in addition to basic drawing tools. A sketch contains real strokes (black) and imaginary strokes (red) called *comment*, which is used to distinguish what a designer sees and perceives. This feature is useful when a shape grammar rule is applied to update the given sketch so that a continuous process of visual calculation can be modeled. There are tools named *wand* and *dye* which can conveniently turn strokes and pixels into *comments*, as well as functions like *impress* to transmit sketch content across different canvases.

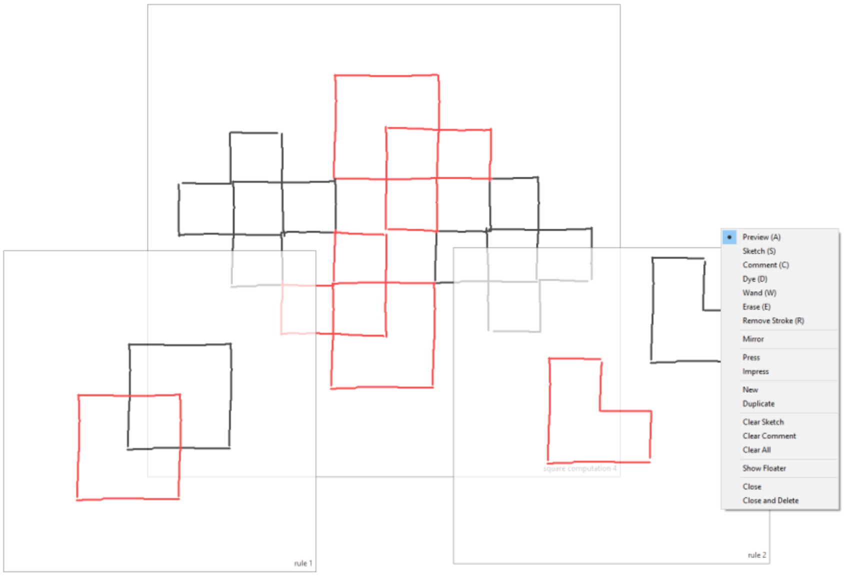


Figure 4-8: Doing visual calculating using sketch widgets

A simple demonstration of visual calculating shows the process using multiple *sketch widgets* altogether. In shape grammar, the initial shape can be recursively updated using different rules, and the process exclusively depends on the human perception that embeds and fuses whatever shape they choose to see. The widgets containing shapes and rules can be modified, scaled, rotated, and duplicated to fulfill design purposes. Through drawing and tracing over sketches and images, creative ideas evolve with both human memory and computer storage.

Taking advantage of the software system, the sketch widget becomes a more powerful tool to express and accumulate visual material. Most importantly, it connects explicit knowledge and implicit experience by triggering the cycle of *participation*, where final knowledge and original experience can be transformed to each other with human perception. It introduces ambiguity to symbolic concepts and embodiment to tacit imagination. All happened within a unified network.

The collaboration of interfaces contributes to the diverse usage of the system. Being able to represent both explicit knowledge and implicit experience empowers designers with extended abilities to learn and think.

4.6 Logic Design

In order to complete the input-store-retrieve workflow, the system has core functions such as a search engine to coordinate learned knowledge in the database and real-time requests from the interfaces. Those functions express the logic of metabolism that keeps the system running and growing. There are four especially unique types of logic in the system.

4.6.1 Accessing Knowledge with Search engine

Similar to recall a piece of memory, the system supports inquiries to previously learned knowledge. Different from myriad public knowledge on the Internet, personal knowledge and experience is more relevant and can be easily accumulated through the system. Designers are able to take advantage of the computational power of machines and search through their ever-expanding network of knowledge at any time.

The search engine function is based on the network structure and works in a straightforward way. When a node is created, verbal information is parsed into meaningful pieces, which are then related back to the node so that it can be reached by keywords in the future. This is a common technique named *inverted index* in common search engine applications. The system not only looks for keyword records but also finds existing nodes as the result, which usually contain more relevant information. The fuzzy search function allows quick access to specific nodes with little input information. It also relies on the attributes of nodes, such as node type and created time, to precisely locate demanded results.

The system is unique as it utilizes the network structure of connections, which enables deeper investigation beyond the information on the surface. While the neutral connections are generic and undirected, the descriptive and procedural connections clearly model the flow of spatial and causal relationships. It is then possible to traverse across the network from one node to another with relatively small computational resources while still being meaningful. When a series of concepts are chained, the end node can be reached from all precedent nodes as the connections transmit recursively. For example, a specific node of “Barcelona Chair” has a series of upstream descriptive concepts: chair, furniture, interior design, architecture. As the concept used for search gets broader, more ambiguity is introduced. More than in-depth structure, the node also expands in width with concepts: *modernism*, *Mies van der Rohe*, etc. Those concepts can be used to regulate search results and discover more relevant nodes. It becomes even more effective when multiple chains are networked and their ends are merged.

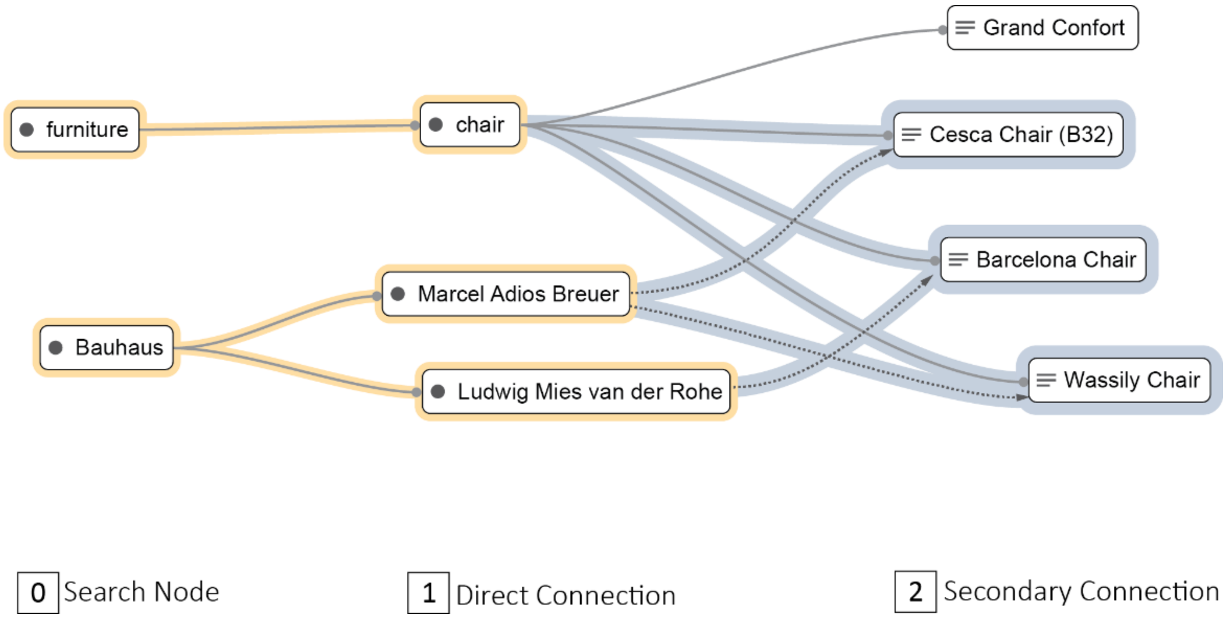


Figure 4-9: Traverse through secondary connections

4.6.2 Making Experience with Active Sketching

As one of the important aspects in modeling the active visual thinking process, the system handles designers' sketches naturally. The visual experience of sketching as a thinking device contributes to a great part of the implicit experience. The purpose of the system is more than keeping the experience as an accessible digital archive, but as a handy tool that supports and streamlines the experience-making process.

Using the *sketch widgets* as pieces of digital tracing paper, each piece is considered a unique episodic node that contains visual information. It can be a drawing, a scribble, or a shape grammar rule that will later be applied and impressed to perform visual calculations. As designers sketch continuously with multiple *widgets*, a subconscious network of experience is dynamically created. Specifically, when visual content transmits from one sketch to another as they are traced, a procedural connection (Pro + Eps) representing a temporal relationship is created. This function helps with clarifying the steps or moves of visual idea creation when a later review of thoughts is necessary. Moreover, the network of sketches is said to be subconscious so that designers are able to draw freely without worrying too much about preservation and storage. At any time when a clear idea surfaces from a pool of sketches, it can be associated back with other nodes and made accessible to verbal inquiry. During this process, the original participation of implicit experience is also finalized as explicit knowledge.

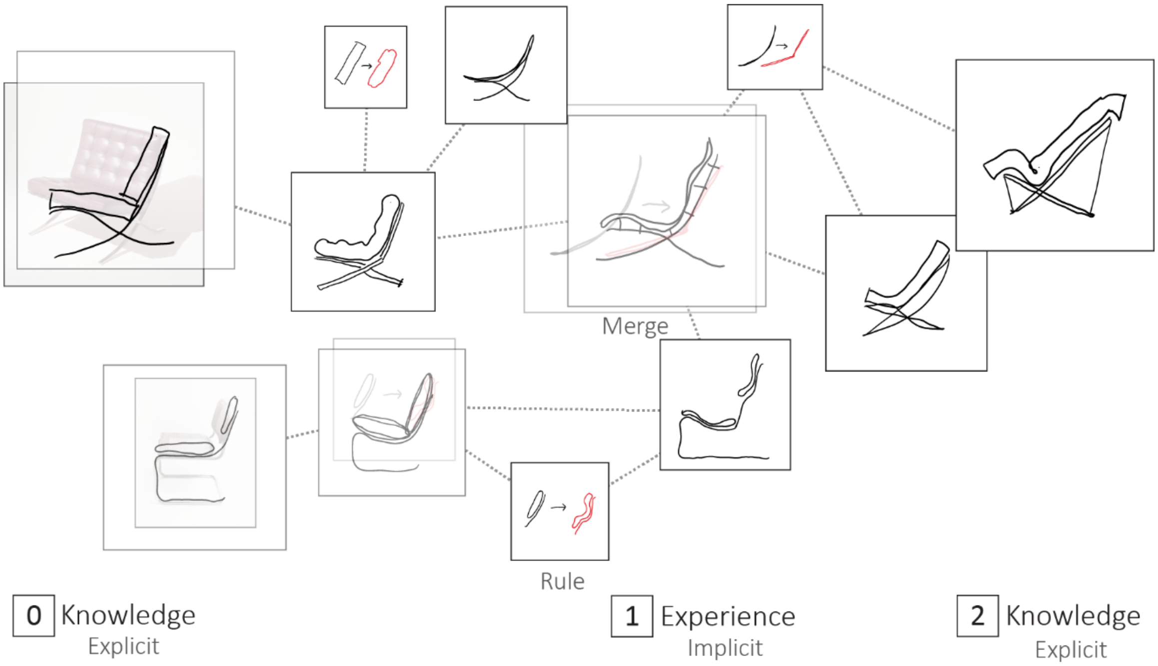


Figure 4-10: Designing of chairs using sketch widgets

4.6.3 Knowledge Backflow: The Popping Mechanism

Maintaining a network of personal accumulation can do more than actively searching for indexed results. While other tools such as notebooks and note-taking software mostly remain static, the system is able to kinetically "pop" learned knowledge back for memorization and inspiration, and designers can passively receive and re-evaluate previous knowledge and experience.

This popping mechanism is based on the *significance dimension* information in the database that records how many times a certain node is searched and popped. When a node is popped, it simply adds 1 to its value. Initially, the system tries to equalize the pop value of all nodes to ensure an overall understanding of all knowledge. This pattern can be customized according to different levels of understanding for individual designers. For example, a simple node with a deeper understanding can be postponed to the future by adding a large number to its pop value, and vice versa.

The system pops nodes on-demand or on a regular basis to ensure that constant reviewing becomes a habit. While human memory can be unreliable and computer memory is usually inflexible, the system keeps dynamical flow across the biological and digital networks of knowledge and experience. As a result, designers are able to include more raw material in creating new ideas with an ever-refreshing brain, both cognitively and digitally.

4.6.4 Versatile Nodes with Programmable Operations

Nodes as a representation of concepts have more potential to inspire network-based applications. It is made possible when each node becomes programmable, as a subcomponent of a greater program, by appending operational information as *operations* to it in the database.

Similar to designing a programming language, the *operations* are instructions that conditionally direct behaviors of nodes. All modules with access to the database can be referred in those *operations*, and the system can automatically update the network in response to user intentions. For example, one common use of *operation* is to defer a currently irrelevant node to the future. A node is firstly disconnected from other nodes so that it becomes invisible with minimal distraction. Then a deferring operation is added, which includes a condition, usually a timestamp of a future moment, and a command, which reconnects the node to its dependent nodes and resumes its accessibility. A separate timer will constantly check for the right time to trigger the command so that the node reappears at the end of the deferment.

The possibility of *operation* is open-ended as it establishes a protocol to computationally manipulate designers' accumulated knowledge and experience. Since sophisticated automation can sometimes be regarded as intelligent, the system can then easily provide efficient cognitive aids customized by each individual. It is also promising to imagine that computers can autonomously construct knowledge networks for future retrieval by human users.

4.7 Creating A Personal System

All components together make a powerful and extendable system rooted in an individual’s knowledge and experience. Incorporating technology into the thinking process, designers create their personal systems to learn effectively and ideate creatively. Notice the keywords are *instrumental* and *augmented*, which means the system never attempts to replace or compete with human intelligence, but rather prioritizes it with computational supports. Aiming at long-term accumulation, designers and their systems are able to cooperate and deliver design results in broader vision and higher quality.

Chapter 5

Demonstration and Evaluation

5.1 Demonstrating the Network

Learning and practicing to become a designer is a long journey, and the development of design abilities derives from day-to-day accumulation. Since this research was a personal project back in 2017, I started to develop a very early version of the system to collect knowledge from my own studies. Most of the progress of the current version are made since the summer of 2020, and since then I began to use the system as the primary tool to manage my own knowledge and experience. More functions were designed along the way to make it a useful tool and thinking augmentation for designers.

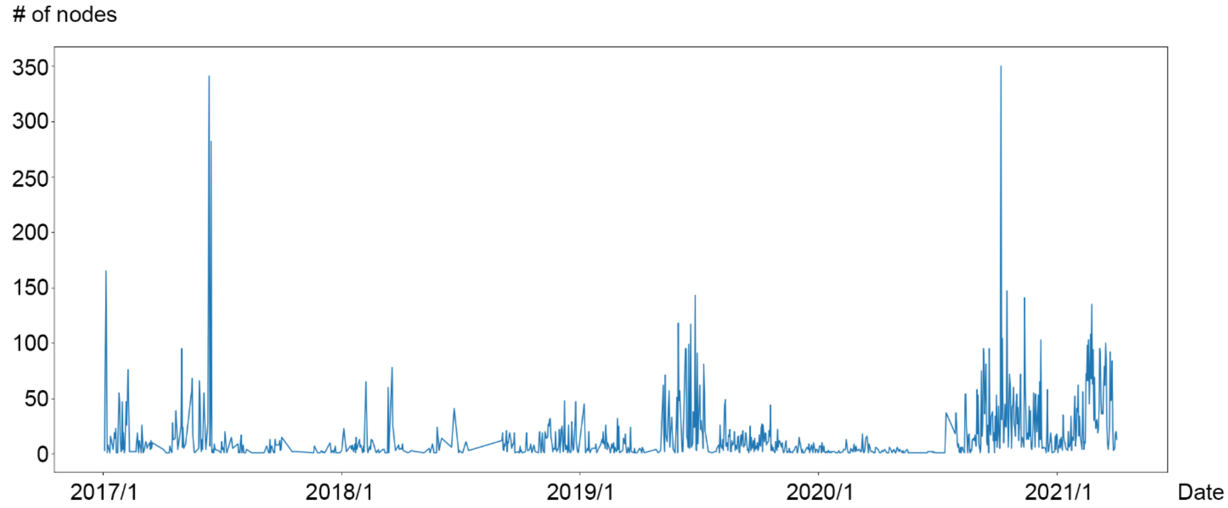


Figure 5-1: A timeline of daily node creation since 2017

Upon until April 2021, I have accumulated over 15,000 nodes each represents a piece of knowledge or experience. Though the natural approach of inquiry depends on real-time events and the system should definitely follow an on-demand manner, it is still valuable to look at the entire network as a big picture. A visualization of all the 15,000 nodes and more than 50,000 connections between them is created by the *Graphia* software.

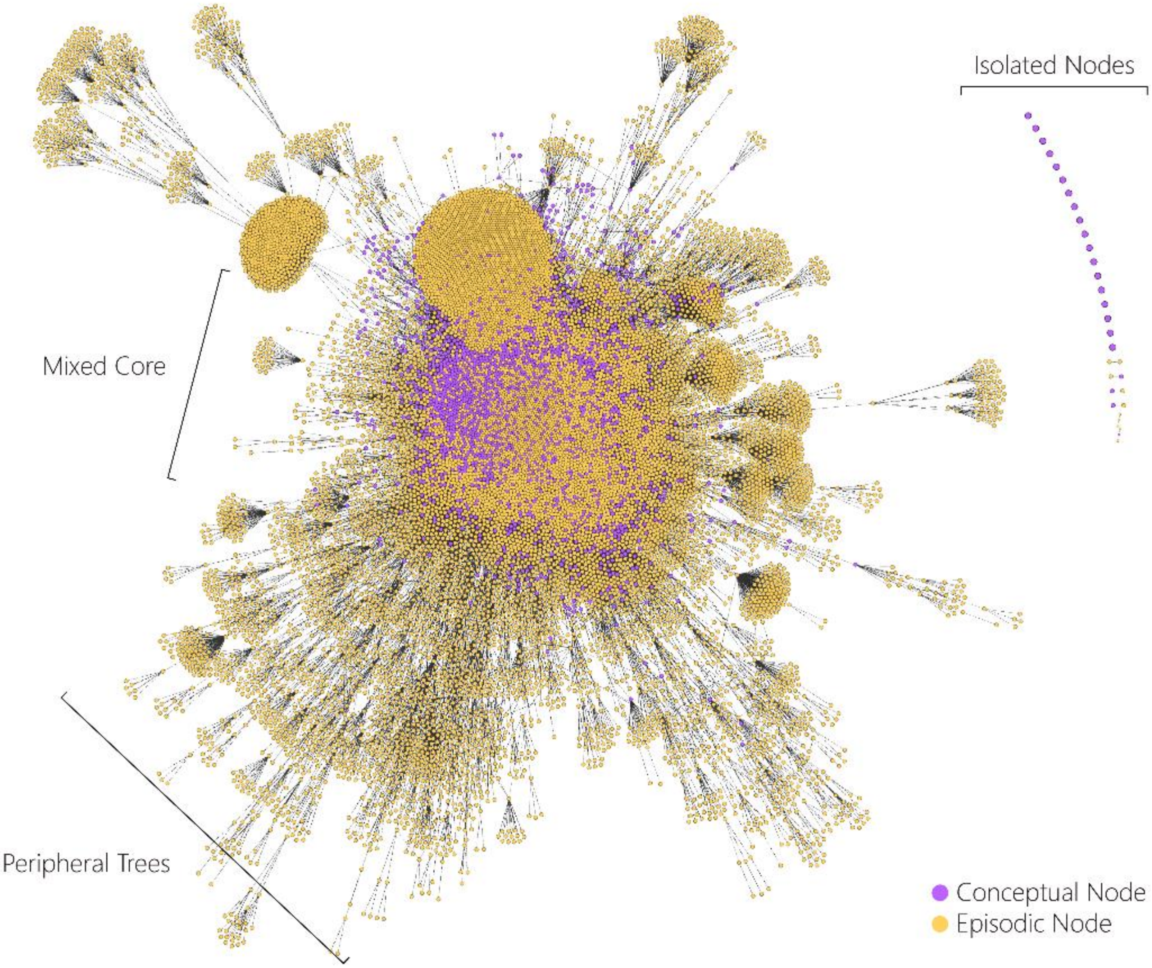


Figure 5-2: Personally accumulated knowledge and experience in a network of more than 15,000 nodes, including Conceptual nodes (purple, 1794), Episodic nodes (yellow, 13026), and 10 types of connections (50382).

The network shows conceptual nodes and episodic nodes in different colors to demonstrate the relationship between important concepts and their instances. Apparently, there exists a core of mixed types of nodes as they are condensed by common connections. Purple conceptual nodes are clustered as schemata and separated by their details in yellow. It is also noticeable that various tree structures present at the periphery are due to hierarchic formats of knowledge such as book and article. This is a temporary ongoing state, and they will be eventually devoured into the center core as more connections are made with different concepts.

The visualization demonstrates that the network of knowledge and experience can be complicated and such sophisticated structure can be constructed accumulatively through the system. The database is compatible with hosting human knowledge, the interfaces are intuitive, and the logic behind the system allows efficient input and retrieval with large-scale networks. Beyond other types of linear and hierarchic management systems such as notebooks and note-taking software, the system is an indivisible part of a designer’s personal development.

5.2 Demonstrating the Design Process

When the system is used in the design process, multiple areas of the knowledge network can be activated to facilitate idea creation. The following example shows a learning and designing procedure inspired by precedent projects and theory framework, where knowledge is collected and retrieved to generate new thoughts. In this case, a local network of architects and styles helps designers with the organization of their symbolic explicit knowledge so that anything can be reached in the thinking process. This can be accomplished by using the searching algorithm of the system. Then abstract knowledge is extracted from examples and episodes of writings and images, and another local network of design techniques and features is constructed. Finally, multiple smaller networks of different fields can be retrieved on-demand for reference, and designers can sketch, trace, and do visual calculations based on all the knowledge made accessible by the system.

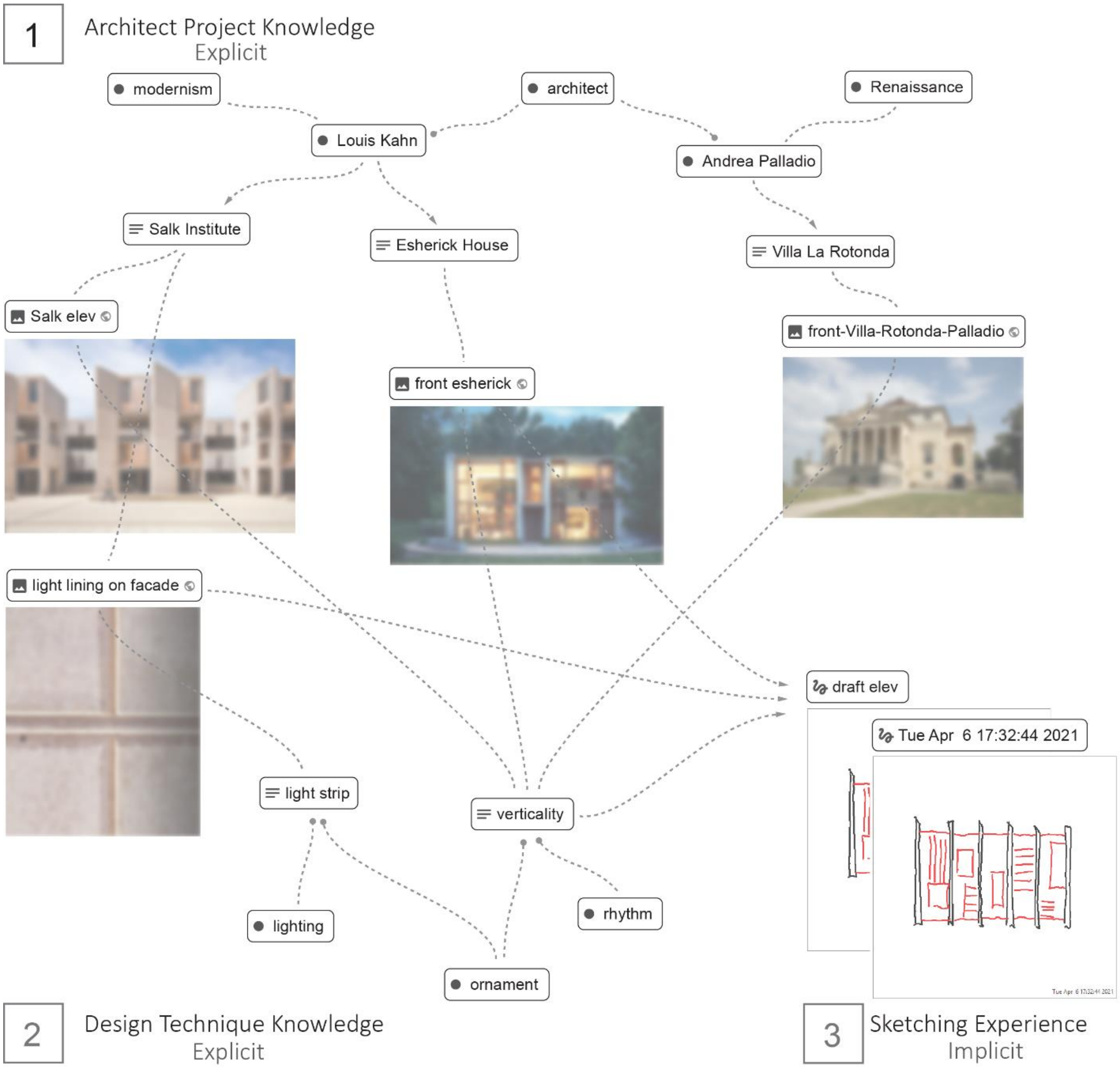


Figure 5-3: Use knowledge and experience in the design process

The demonstrated process only recovers a small piece of design activities, and the same transformation among knowledge and experience can be performed anywhere in the design process for as many times as needed. Developing and utilizing personal networks makes individual designers more competitive and productive in design thinking and ideation, as they take the advantage of technology augmentation in learning and practicing.

5.3 Evaluation and Contribution

Surveys within the architecture group suggests that different designers may have their distinct understanding of knowledge and experience, and a variety of strategies can be applied to fulfill their personal demands. Inefficient hand sketches can also be less restrictive, and well-documented digital projects can also be poorly reused. One of the key values of the instrumental system is to build a powerful yet unique representation of both verbal knowledge and visual experience for each individual designer. It is highly customizable to adapt to all kinds of design purposes. The system also actively maintains a dynamic equilibrium between human perception and computer automation, which contributes to a growing integration of augmented intelligence.

Filling the gap between design and technology in thinking and ideation sectors, the instrumental system suggests a new perspective looking at the continuous development of individual designers. When CAD and BIM applications focus on efficiency and productivity, the system prioritizes designers' personal perception and provides reliable supports to human imagination and creativity. Though the prototype software can fulfill most of the functions and can also be developed rapidly, the future of the system is a formal public product. Long-term user research and feedback are critical to further refine the system. It also has the potential to establish communities and an organizational ecosystem where knowledge and experience can be shared between individuals for educational and practical purposes.

Chapter 6

Conclusion

This research thesis starts with the question of how technology can support creative design. It covers an investigation of precedent literature and current tools and a proposed theory of network representing designers' accumulation. Based on studies of explicit knowledge representation which describe a symbolic and combinatoric worldview, the skeleton of the network theory is established to effectively organize knowledge. Introducing implicit experience and research on visual calculation, the limitation of predefined structure is released and a unified network model is created to better represent the design process.

The thesis also includes a developed software system that implements the personal network with fully functional storage, interface, and operational logic. Guided by the accomplishment of theories, the system is designed to fulfill a complete workflow of learning and designing for individual designers. Lower-level mechanisms and higher-level applications are discussed in detail to illustrate how different functions are determined and assembled together.

In the end, a demonstration of an established network indicates the system is able to grow with the user as a computational augmentation of memory and perception. The thesis concludes that the system can facilitate the personal development of individual designers and thus improve ideation and creativity with the help of novel design technology. Still being a conceptual prototype system, future user research in a long term and larger scale is necessary to turn this project into a real product. It is promising to discover more potential functionalities that can benefit designers and the design discipline.

Bibliography

[1] Archer, B. (1979). Design as a discipline. *Design studies*, *1*(1), 17-20.

[2] Cross, N. (2007). 'Forty years of design research'. *Design studies*, *1*(28), 1-4.

[3] Dossick C. S. & Neff G. (2011). 'Messy talk and clean technology: communication, problem-solving and collaboration using Building Information Modelling'. *Engineering Project Organization Journal,* 1:2, 83-93

[4] Sutherland, I. E. (1964). 'Sketchpad a Man-Machine Graphical Communication System'. *SIMULATION*, *2*(5), R-3-R-20.

[5] Simon, H. A. (1969). *The sciences of the artificial*. Cambridge: MIT Press.

[6] Schön, D. A. (1987). *Educating the reflective practitioner: toward a new design for teaching and learning in the professions.* San Francisco: Jossey-Bass.

[7] Cross, N. (1982). 'Designerly ways of knowing'. *Design studies*, *3*(4), 221-227.

[8] Lawson, B., & Loke, S. M. (1997). Computers, words and pictures. *Design studies*, *18*(2), 171-183.

[9] Davis, R., Shrobe, H., & Szolovits, P. (1993). 'What is a Knowledge Representation? ' *AI Magazine*, 14(1):17-33.

[10] Heylighen, A., & Neuckermans, H. (2000). 'DYNAMO: A Dynamic Architectural Memory On-line'. *Journal of Educational Technology & Society*, *3*(2), 86-95.

[11] Oxman, R. (2004). 'Think-maps: teaching design thinking in design education'. *Design studies. 25*(1), 63-91.

[12] Stiny, G. (2006). *Shape: talking about seeing and doing.* MIT Press.

[13] Lawson, B. (1979) 'Cognitive strategies in architectural design'. *Ergonomics.* Vol 22, No 1, 59-68.

[14] Lawson, B. (1980) *How Designers Think: The Design Process Demystified.* Architectural Press, Oxford, Elsevier

[15] Rowe, P (1987) *Design Thinking* MIT Press, Cambridge, MA

[16] Archer, B (1980) 'The mind's eye: not so much seeing as thinking' *Designer*. 8-9 33.

[17] Ferguson, E S (1977) 'The mind's eye: non-verbal thought in technology' *Science* Vo1197 No 4306 32.

[18] Cross, A (1980) 'An introduction to non-verbal aspects of thought' *Design Educ. Res. Note 5* Design Discipline, The Open University, Milton Keynes, Bucks, UK

[19] Tulving, E. (1985). 'How many memory systems are there? '. *American Psychologist*, *40*(4), 385–398.

[20] Ryle, G. (2009). *The concept of mind*. Routledge.

[21] Kant, I. (1929) *Critique of pure reason*. Smith, N. Kemp, Trans.

[22] Schank, R. C., and R. P. Abelson. (1977). *Scripts, Plans, Goals, and Understanding: An Enquiry into Human Knowledge Structures*. Erlbaum.

[23] Minsky, M. (1975), 'A framework for representing knowledge', P H Winston Ed. *The psychology of computer vision*, McGraw-Hill, New York, NY.

[24] Minsky, M. (2007). *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster.

[25] Chomsky, N. (2002) *Syntactic structures*. De Gruyter Mouton.

[26] Winston, P. H. (1993). *Artificial Intelligence, Third Edition*. Addison-Wesley.

[27] Quillan, M. R. (1966). *Semantic memory*. Bolt Beranek and Newman Inc Cambridge MA.

[28] Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., & Shepherd, M. (1990). 'Cyc: toward programs with common sense'. *Communications of the ACM*, *33*(8), 30-49.

[29] Mitchell, T., & Fredkin, E. (2014). 'Never ending language learning'. In *Big Data (Big Data), 2014 IEEE International Conference*. 1-1.

[30] Mitchell, W. (1990). *The logic of architecture: design, computation and cognition*. MIT Press.

[31] Lawson, B. (1994). *Design in mind.* Oxford: England: Butterworth Architecture.

[32] Oxman, R. E. (1994). 'Precedents in design: a computational model for the organization of precedent knowledge'. *Design studies*, *15*(2), 141-157.

[33] Gero, J. S., & Kannengiesser, U. (2014). 'The function-behaviour-structure ontology of design'. In *An anthology of theories and models of design.* 263-283. Springer, London.

[34] Heylighen, A., Bouwen, J., & Neuckermans, H. (1998). 'Walking on a thin line—Between passive knowledge and active knowing of components and concepts in architectural design'. *Design Studies, 20,* 211-235.

[35] Oxman, R. (2001). 'The mind in design: a conceptual framework for cognition in design education'. In *Design knowing and learning: Cognition in design education.* 269-295. Elsevier Science.

[36] Miller, H. (2018). 'Cultivating Next-GEN Designers-The Systematic Transfer of Knowledge'. In *Proceedings of the 36th eCAADe Conference.* 25-34.

[37] Martin, W. M., Heylighen, A., & Cavallin, H. (2003). 'Building² Stories. A hermeneutic approach to studying design practice'. In *Proceedings of the 5th European Academy of Design Conference*.

[38] Pauwels, P., De Meyer, R., & Van Campenhout, J. (2011). 'Extending the design process into the knowledge of the world'. In *14th International Conference on Computer Aided Architectural Design Research in Asia (CAADFutures-2011).* 203-216. Les éditions de l'université de Liège.

[39] Iwasaki, K., Kuriyama, Y., Kondoh, S., & Shirayori, A. (2018). 'Structuring engineers' implicit knowledge of forming process design by using a graph model'. *Procedia Cirp, 67,* 563-568.

[40] Stiny, G. (1980). 'Introduction to shape and shape grammars'. *Environment and planning B: planning and design*, *7*(3), 343-351.

[41] Stiny, G. (2015). 'The critic as artist: Oscar Wilde's prolegomena to shape grammars'. *Nexus Network Journal*, *17*(3), 723-758.

[42] Knight, T. (2015). Shapes and other things. *Nexus Network Journal*, *17*(3), 963-980.

[43] Schön, D. A., & Wiggins, G. (1992). Kinds of seeing in designing. *Creativity and Innovation Management*, *1*(2), 68-74.

[44] Purcell, A. T., & Gero, J. S. (1998). Drawings and the design process: A review of protocol studies in design and other disciplines and related research in cognitive psychology. *Design studies*, *19*(4), 389-430.

[45] Smithers, T. (2001). 'Is sketching an aid to memory or a kind of thinking'. *Visual and Spatial Reasoning in Design II. eds JS Gero, B. Tversky, T. Purcell. Key Centre Sydney*, 165-176.

[46] Goldschmidt, G. (1991). 'The dialectics of sketching'. *Creativity research journal*, *4*(2), 123-143.

[47] Rodgers, P. A., Green, G., & Mcgown, A. (2000). Using concept sketches to track design progress. *Design studies*, *21*(5), 451-464.

[48] Sutherland, I. (1975). 'Structure in drawings and the hidden-surface problem'. *Reflections on computer aids to design and architecture*, 73-77. Petrocelli/Charter, New York.

[49] Do, E. Y. L. (2005). Design sketches and sketch design tools. *Knowledge-Based Systems*, *18*(8), 383-405.

[50] Ranscombe, C., Zhang, W., Rodda, J., & Mathias, D. (2019, July). 'Digital Sketch Modelling: Proposing a Hybrid Visualisation Tool Combining Affordances of Sketching and CAD'. In *Proceedings of the Design Society: International Conference on Engineering Design* (Vol. 1, No. 1, pp. 309-318). Cambridge University Press.

[51] Stiny, G. (2010). 'What Rule(s) Should I Use?'. *Nexus Network Journal*. 13**,** 15–47

[52] Stiny, G. (1982). 'Spatial relations and grammars'. *Environment and Planning B: Planning and Design*, *9*(1), 113-114.

[53] Barfield, O. (1965). *Saving the appearances: A study in idolatry*. Harcourt, Brace & World.

[54] Boden, M. A. (1990) *The Creative Mind: Myths and Mechanisms.* London: Abacus.

[55] Boden, M. A. (1994) 'What is creativity?' In M. A. Boden (ed.) *Dimensions of Creativity*. Cambridge, MA: MIT Press

[56] Kotsopoulos, S. (2007, November). Design Concepts in Architecture: the Porosity Paradigm. In *SWW 2.0*.

[57] Huff, A. S. (Ed.). (1990). *Mapping strategic thought*. John Wiley & Sons Incorporated.

[58] Novak, J. D. (1991). 'Clarify with concept maps'. *The Science Teacher* Vol 58 No 7 45-49.

[59] Boden, M. A. (2001) 'Creativity and knowledge'. Craft, A., Jeffrey, B., & Leibling, M. (Eds.). (2001). *Creativity in education*. A&C Black. 98.

[60] Tulving, E. (1972). 'Episodic and Semantic Memory'. In E. Tulving & W. Donaldson (Ed.) *Organization of Memory*. 381-403. Academic Press.

[61] Broadbent, D. (1989). 'Lasting representations and temporary processes'. In H.L. Roediger, III & F.I.M. Craik (Eds.), *Varieties of memory and consciousness*: *Essays in honor of Endel Tulving.* 211–227. Hillsdale, NJ: Erlbaum.

[62] Stiny, G. (1990). 'What is a design?'. *Environment and Planning B: Planning and Design*, 17(1):97 103.

[63] Foucault, M. (2005). *The hermeneutics of the subject: Lectures at the Collège de France 1981--1982* (Vol. 6). Macmillan.