

Framing a computational tool for early conceptual structural design

J. Corneliussen, S. Deleuran, P.H. Kirkegaard, and V. Beatini

Aarhus University, Dept. of Civil & Architectural Engineering, Aarhus, 8000, Denmark.

ABSTRACT: The aspiration for integration of architecture and structure toward a tectonic design clashes with the lack of adequate computational design tools and additional costs that it can face when brought to the strategic early stages of the conceptual design phase. Through a double cycle of action research involving professionals, this paper firstly analyses the current conditions of collaboration in the early conceptual design phase, and, on this basis, proposes a framework for a structural computational tool that could be adequate or rather favor the integrated process. Though further research is needed to clarify all aspects of the tool, this paper delineates the kind of information meaningful for the conceptual structural design to be a collaborative venture as well as the media and method of communication between the architect and engineer, eventually proposing a formulation of the shared design space that could support it.

1 A NEED FOR AN INTEGRATED CONCEPTUAL DESIGN PRACTICE

For architecture to become a work of art, it must reach beyond the concrete by harmoniously synthesizing structure and architecture. The underlying integrated design approach should characterize the decision-making from the early stages in the design process because here the design freedom is at its maximum and the cost of design changes at its minimum. Given the highly specialized knowledge required today, such an early integration cannot be conveyed by a single designer. Yet, the computational design process frees the designer from relying solely on his capacity and experience – thereby challenging the conventional design process. Computational design indeed enhances the agility of the designers, easing the possibility of interdisciplinary collaboration in the fast-moving conceptual design phase. Furthermore, is a recipient of data but also a generator of data, and such it can become an active member of the design process. This realizes the aspirational interaction between designers and computational tools qualitatively experimented by Cross (2018). Many tools have been successfully developed aimed at these objectives; however, when the early conceptual phase is addressed, results are not satisfactory (Mora et al. 2008). To favor a deeper collaboration between architecture and structure in conceptual design, this research tackles through an action research method the lack of applied research addressing the early design phase, eventually framing a computational tool for early conceptual structural design.

A discussion on the conceptual design phase clarifies the nature of the problem. Following Schön (1984) and Lawson (2005), design is a reflective practice that, through actions and reflections, outlines the design solution and, at the same time, designates the design problem. Along the process, the designer can apply different methodologies, but according to Cross (2018), who has for a long time investigated the design discipline, three stages of the creation of design may be distinguished, see Figure 1a. The first stage is kick-off, also known as the design brief. This is a mix of relatively stable, implicit problem goals held by the designer, and explicit solution criteria specified by the client's program and relevant regulations. Afterward, the intermediate stage begins, also called the conceptualization phase. Now a possible concretization of the design concept

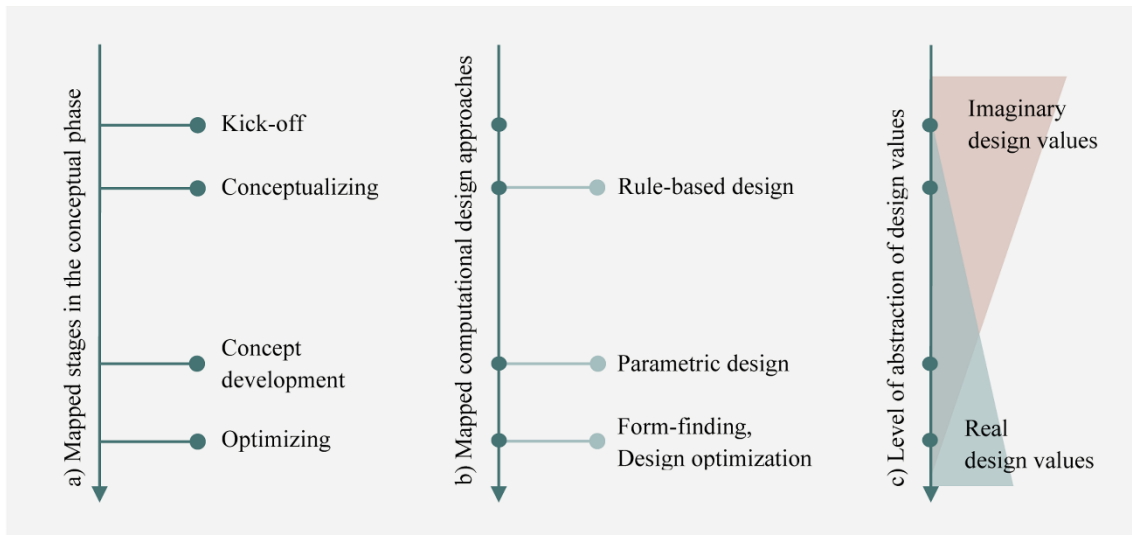


Figure 1. a) b) c) Mapping of the conceptual design phase

is explored into form and space under the aim of framing the problem, and, contextually, matching abstract solution criteria. Despite the strategic importance of the phase, the methods applied by the designer are subjective and often tacit. Eventually, a concept arises when the designer recognizes the principles of knowledge embodied in the solution space that allow him to identify the problem and that will inform the successive design phases. Nowadays, the advancement of computational tools also allows an optimization stage where the developed concept is refined before entering the next phase of the design process.

Rolvink et al. (2014) have searched the constraints posed by each phase of the design process to the usage of such tools, see Figure 1b. Of interest here, they found a difference in the design values used in the conceptualizing and concept development phase, Figure 1c. The activities implemented in the conceptualizing phase use qualitative parameters and imaginary values. Vice versa, the concept development employs also numerical measurement, that is real design values, which will dominate the successive phases. Of the current computational tools, Figure 1b, some are devoted to purely formal design explorations, aimed at exploring and extending the solution space, others optimize the engineering metrics once the form is outlined, and favor the convergence to a design solution. If the tools, Figure 1b, are analyzed based on the type and level of knowledge that they require, Figure 1c, and based on this they are associated with the relevant phase of the design process, Figure 1a, it appears evident that they cannot support the early concept design, because they operate with a different type of knowledge. In other words, even the tools that support divergent processes, which do extend the solution space, do not allow the divergent coding (Cross 2001) that the designer subjectively tests in order to translate abstract requirements into concrete entities because their variables are numerical parameters that univocally refer to certain entities.

To map the recurring behavioral patterns in the early concept design phase that a tool shall confront, this research reports the results from interviews with architectural and structural professionals in the following section. From their critical analysis, in Section 3 a tool storyboarding is implemented, which is likewise subjected to an action research evaluation by a focus group of practitioners. The results and possible future work are discussed in the Conclusions.

2 THEMATIC PATTERNS IN THE CONCEPTUAL DESIGN PHASE

2.1 *Integrated conceptual design as experienced by professionals*

In order to better understand if and how a structural computational tool can foster the collaboration of engineers and architects in the conceptualization activities, this section investigates how the phase is experienced and perceived by architects and engineers who already work with computational tools somewhere along the conceptual design process.

The employed approach is Action Research (Kemmis et al. 2014). Briefly, the approach derives from social science, where it is used to investigate the conditions of communication and collaboration between different groups willing to improve their situation connected to a shared interest. It involves people inside the problem to tackle, here the professionals, and it has been employed here to evaluate the transferability of knowledge between architects and engineers in the conceptual design phase. Kvale (2007) describes interviews as among the most ordinary ways of collecting knowledge of a person's life situation, opinions, and experiences.

Based on the observations reported in the Introduction, professionals working internationally and with a base in Denmark have been selected for individual semi-structured qualitative interviews. According to Braun & Clarke (2006), a low number of interviewees shall be set when the aim is not to confirm the problem statement but, as in this case, to uncover the professional interaction of the interviewees. The interviewees are two architects from 3XN Architects and LINK Architecture and three engineers from Arup, Rambøll, and Søren Jensen respectively, for a total of five persons. An effort has been paid to select high-quality professionals, as previous studies (Cross 2001) also suggest that the quality of the interviewees is more important to understand the practice from the inside.

The data that emerged from the interviews were eventually subjected to a deductive thematic analysis to derive meaningful patterns that can inform a tool. Given the subjective nature of the designers' reality, semi-structured individual interviews have been chosen. The professional was asked to narrate the situation encountered in his/her practice starting from a relevant experience and commenting on whether it is unique or typical on his/her personal experience. If opportune, the interviewer guided the interviewee to uncover the type of structural information exchanged, its timeliness, and the media and method required for the communication. In order to limit bias and misunderstandings, it was emphasized with the professionals that the interviews' outcome was not to be generalized but represented a few designers' subjective opinions, and they were able to reformulate it if they noticed possible misunderstanding in the minutes. The interview guides, different for architects and engineers, as well as the interviews, can be read in (Corneliusen & Deleuran 2021).

2.2 *The space for a conceptual structural design tool*

The responses to the interviews have been investigated to disclose recurring *themes* useful to characterize the requirements that a structural computational tool shall fulfill. Following the thematic analysis procedure (Braun & Clarke 2006), the findings contained in the interviews have been assigned codes and reorganized into thematic areas here under a deductive approach relevant to the research objective. Eventually, to refine and validate the themes, the interviewees have reviewed the themes ensuring that they recognize a meaningful relationship between their original statements and the given themes. Six themes have resulted, whose essence is outlined below.

(1) Current conditions of collaboration in the conceptual phase. The interviews recognize the conceptual design phase as the most challenging to the integration of information primarily because of the very diverse solutions investigated by the architect in a short time.

(2) Industry-related evolution. The lack of early integration causes structural problems, opportunities, and constraints to pass unnoticed. The potential consequent increment of cost cannot be healed in a later integration. The phase deserves more resources to be prioritized.

(3) The characteristic of conceptual design. The interviews remark the subjective and implicit nature of the phase and recognize the architect as the leading actor. The engineer shall be willing to embrace the early design's fuzzy and fast-moving process.

(4) Relevant structural information to be transmitted. The interviewed architects find it difficult to formulate a request for specific structural information at this phase, whereas the engineers point out the evolution of a structural concept expressed by the load path as key information to be passed to the architects.

(5) Media and method of communication. A shared platform is considered essential, which shall allow the engineer to read and use abstract geometries defined by the architect.

(6) Intended user of the conceptual structural design tool. The architects find using a specialistic tool too resource-intensive at this phase; the engineers also remark that specialistic tools require knowledge in judging the tool's results and experience in recognizing possible inconsistencies in the application or limitations of the results.

3 THE ROLE OF A STRUCTURAL CONCEPT TOOL

3.1 *The tool's requirements*

The findings in Section 2 confirm that it is advisable to develop structural computational tools directed to the early conceptual design phase of the design process.

The tool shall share some *requirements* common to computational tools for integrated, explorative design already mentioned in the Introduction, namely:

(a) Favor interactive rather than automated design. The tool shall automate the tedious and repetitive tasks while here making easier and more focused the discussion between the engineer and the architect thereby serving as a communication and collaboration tool.

However, it shall also have unique features that derive from the specificity of the conceptualization phase. These are summarized below.

(b) Serve the communication between the engineer and the architect. The tool shall provide visual input to the integrated process with the same level of abstraction as the input geometry serving the engineer in communicating comprehensible structural input. The tool shall require no more than the abstract, not structurally organized geometrical entities that the architect uses to conceptualize the imaginary design values: the tool shall be in charge to codify them as structural elements; the architect shall not be constrained to use the tool from when he/she starts modeling the concept if he/she wants to use it later.

(c) Serve as a non-intrusive partner to the architect. The tool shall be able to recognize and apply the kind of design activities implemented by the architect while investigating a certain concept. This implies that the tool shall be able to implement all the architect's composition and transformation activities. It should be noted here that, because the tool is intended for sharing meaningful structural knowledge, it is not considered necessary for the tool to try learning the subjective logic applied by the architect in doing such activities.

(d) Serve the engineer in the development of a structural concept. The tool shall speed the engineer in identifying possible load paths and allow him to qualitatively compare alternatives in short cycles of synthesis-analysis-evaluation.

(e) Serve as a creative brainstorming media. The tool shall allow the engineer to dynamically explore structural concepts within and out of the spatial compositions and transformations set by the architect. At best, the tool shall be a creative partner capable of generating divergent solutions that may also provoke the designers' beliefs in a dynamic, yet meaningful way.

3.2 *Formulation of the design space*

As discussed in the Introduction, the existing tools manipulate numerical parameters, whereas a tool for early structural conceptualization shall work with imaginary values implicit into the logic of relationships between geometrical entities. By analyzing requirement (b) of the target tool, it can be noticed that this is indeed the first and foremost difference between the target and the existing ones. Yet, the research reported in Section 2 has enlightened that the major challenge to an integrated early design is not due to the values in use, but to the substantially divergent formal solutions through which they are explored. The structural concept tool shall therefore not be able to identify the imaginary parameters, which stay subjective and context related, but reproduce and favor their formal combinations and compositions.

These activities are the ones that the architect performs to organize spatially formal geometrical entities and associated material qualities in a similar way to the organization of words in a sentence, that is, according to rules. Rule-based design has been tested in architecture and related computational tools in the form of shape grammar design, that is by formulating the rules through formal graphical relationships. Opposite to parametric design, this visual thinking methodology allows exploring a large solution space not codifiable by a limited set of parameters. A rule-based formulation of the design space may be particularly suitable for transferring to engineers the ongoing results of the conceptualization run by the architect because it would constrain the engineers to manipulate the concepts within the same type of exploration used by the architects, and such it would result more agile and relevant. This responds to requirement (c) because it would lighten the time that the architect shall invest trying to explain and make the engineer apply his/her design logic. However, without numerical boundaries, rule-based formulations can contain many purely artificial solutions, that detach architecture from its physical realm. According to requirement (d),

the tool shall instead allow the exploration of a reasonable load path, ensuring a meaningful relationship between the architectural activities and the structural interpretation realized by the tool.

Purup & Petersen (2019) have investigated these activities. They found that, once the design is moved into the digital realm, there is a pattern of correspondences between the kind of transformations and composition explored by the architect and the type of geometrical entities they operate with. For example, the building envelope may be deformed or punctuated by windows, vertical elements are changed in number, position, and scale. Despite the strategies that underpins the design values can vary, some transformations and compositions tend to be actuated simultaneously. For example, if the designer wants to create a dichotomy between a bounding envelope and its internal volumes, he may want to move and scale the volumes while keeping them within the envelope, a set of transformations which Purup & Petersen (2019) call “Tetris-like”.

Since the architectural organism is in fact a combination of structure and architecture, these forms also have a structural role, and therefore, once the hierarchy of the structure from abstract geometric entities has been recreated, it is possible to associate the various structural elements and components specifically with one or more of the transformation rules relevant to the architect. The engineer shall thus become able to provide the architect relevant structural information that fits his design exploration. This would open the possibility to use the tool as a creative brainstorming partner not only able to support but also to provoke the engineer and the architect, fulfilling requirement (e). It should be remarked that the purpose of introducing a new tool is to favor tectonic conceptual design solutions by supporting the engineer in proposing a range of possible structural concepts in harmony with the architectural design. Therefore, it is not considered necessary for the tool to try learning the subjective and strategic thinking that guides the spatial transformations and compositions applied by the architect; machine learning possibilities to understand and reproduce the architect’s thinking are not considered a capability of the tool.

3.3 The design workflow

The tectonic design workflow is reviewed to visualize the potential positioning of the tool, Figure 2. The design is usually initiated in the analogous space, because it allows a more direct setting of the implicit problem goals held by the designer (see the Introduction). From here, an amount A of design solutions is selected for digital modeling, where an expanded design space A+ emerges thanks to the continuation of the exploration through the faster media. Within it, the designer selects a subgroup B of design solutions to be structurally explored. Here, a potentially much larger structurally informed design space B+ emerges. Eventually, the designer(s) select one or more design solutions C which can be further expanded either analogically or digitally (C1+ or C2+) also through parametric tools. For this workflow to be entirely applicable, there must be a dynamic and open coupling of the various tools in the design team. Therefore, for the tool to be effectively agile in processing the data, requirement (a), its input and output should be processed in a Visual Programming Language (VPL) as simple geometric primitives. That is for example what happens within the parametric plug-in Grasshopper and its many sub-plug-in.

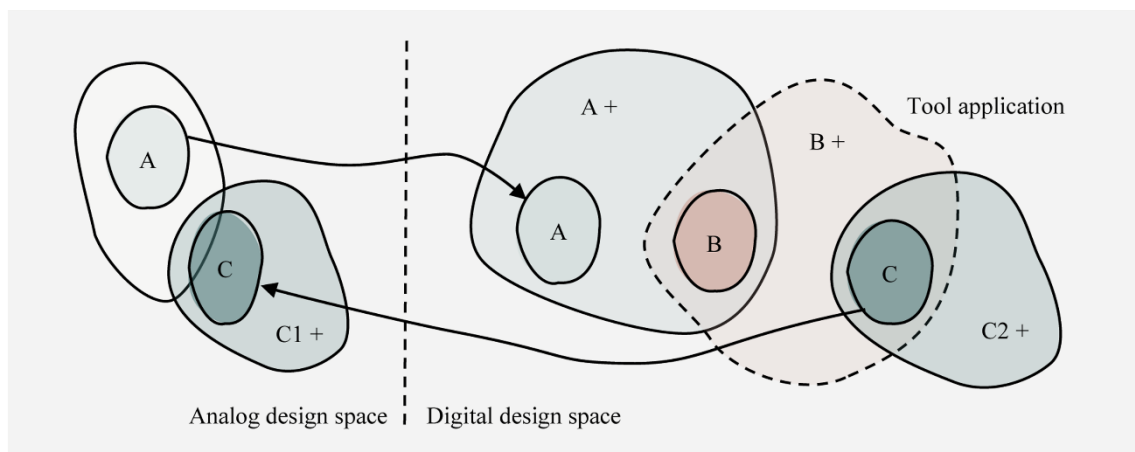


Figure 2. The design space with the influence of a conceptual structural design tool.

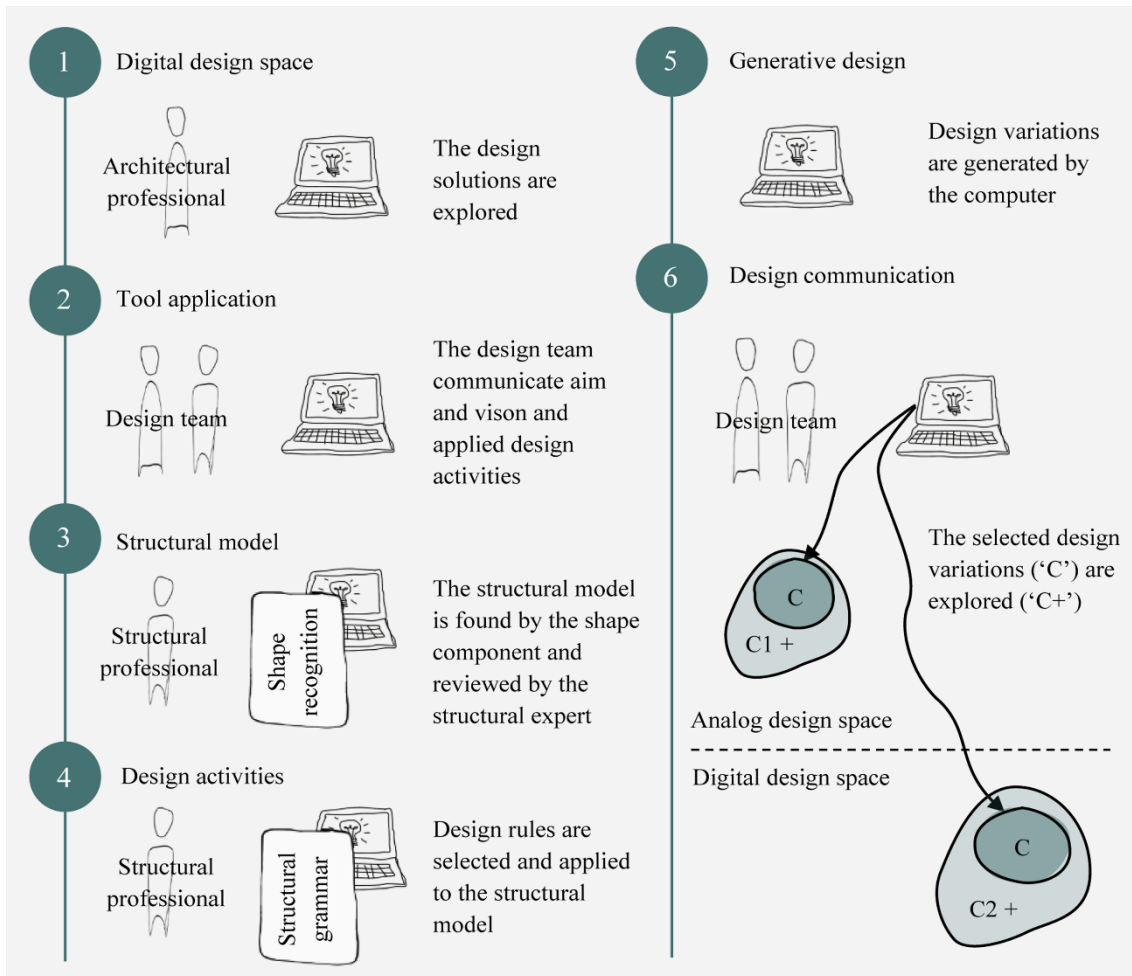


Figure 3. Storyboard prototype of the tool proposal for the conceptual design phase.

4 STORYBOARDING FOR AN INTEGRATED DESIGN TOOL

4.1 *The tool's components*

The tool shall comprise two major components, a Shape and a Structural Grammar component. The Shape component seeks to translate the primitive geometrical entities into structural entities. These shall be organized hierarchically according to their structural role within the hypothetical construction. Mora et al. (2008) have attempted to define structural components and elements by reorganizing the classic technological taxonomy of the architectural organism based on structural hierarchy. This is pertinent to the scope and briefly summarized to clarify the behavior of the shape component. At the higher level, the tool shall recognize structurally independent Volumes, which are the largest entities that bring external actions to the ground; within them, it shall identify possible Sub-Volumes, i.e. volumes not structurally independent, Assemblies, which are not structurally independent entities organized into a surface or a frame, Elements, which comprise minimal structural entities, like beams, columns, plates, and shell panels, and Connections, which define the endpoints of the elements and, by default, shall act here as rigid joints. The tool shall be able to identify step-wise the structural entities based on their positional relationship, orientation in space, and intrinsic geometrical proportion. Once an Element is activated, the component recognizes it as a load-bearing entity. The active connected Elements visually return the potential Load Paths. The user shall decide which Elements and Load Paths thereof to activate. The trajectory of a load path in addition to self-weight shall be initiated by the user by means of vector(s) applied to one or more elements. The tool may support optional servant components, for example, supports shall be better inserted manually, whereas the user may want to control the DOFs of some joints or the ratio of stiffness between elements.

The Structural Grammar component defines rules that can be applied to the identified shapes. These allow the engineer to interact in a fast manner with the architect while proposing solutions pertinent to the formal kind of combination and composition explored by the architect. Each level of structural entity accepts a certain type of transformations and compositions. Volumes can be translated, Sub Volumes can be scaled and translated, optionally while staying within the Volume they belong to; Assemblies can be rotated and bent; Elements can be scaled, translated, and multiplied. In addition, the component suggests to group multiple activities at a higher level based on the common patterns recognized in (Purup & Petersen 2019), see Section 3.2. For example, translation and scaling of Sub Volumes may be made accessible if the “Tetris” design mode is chosen.

4.2 *The tool’s workflow*

The workflow of the tool is visualized in Figure 3 as a storyboard prototype. This is a representation of the tool and a visual walkthrough of its use. The usage of the tool is divided into six steps. (1) The architectural professional explores design solutions within a digital environment. (2) A design solution is explored with the tool. It is important to communicate the design activities, aim and vision between the architectural and structural professionals, to make a correlation between the input and output of the tool. (3) The structural model is generated by the computer through identification of structural entities and reviewed by the structural professional. (4) Structural rules for generating design alternatives are chosen. The rules should follow the current activity of the architectural professional. (5) Design alternatives are generated based on the outputs from the two components. (6) Communication of the design alternatives are made. New ideas inspired by the design communication and the generated output of the computer can afterwards be explored in either an analog or digital design space.

4.3 *Feedback on the conceptual tool by the focus group*

A second action research cycle has been conducted to obtain feedback on the validity of the requirements deduced in Section 3.1 concerning the themes presented in Section 2.2, and to collect different perspectives on how such a kind of tool may be further improved.

As suggested by Kvale (2007), when exploratory studies are conducted in a new domain, an unstructured interview on the subject is very suitable to enrich perspectives. Here, the storyboard prototype has been submitted for a joint discussion to a group of professionals, two architects from LINK Architecture and one engineer from Multiconsult, who are collaboratively working themselves on developing computational tools focusing on early interdisciplinary knowledge.

The focus group has had positive feedback concerning the medium of communication and confirmed the type of knowledge shared to be relevant. Altogether, the Shape and the Structural Grammar components allow the designers to reflect on their work within a common medium as called for by requirement (a). Yet, they expect neither the architect nor the engineer to possess advanced mutual interdisciplinary knowledge, and the level of knowledge sharing allowed by the transformation and composition of design rules on one side, the load path on the other side is thought to be sufficient for the conceptual design phase, as outlined by requirements (b) and (d). The suggested ability of the Shape component to automatically categorize the input geometrical entities in a structural hierarchy makes the time investment for its usage shorter, and such it diminishes the industry-related challenges, also requirement (c).

It shall be remarked that the tool as conceived lacks dedicated structural analysis and visualization components, which also means that the tool is not able to prioritize solutions based on the structural efficiency. This, together with the quality of the knowledge shared, have centralized the interest of the focus group during the discussion. Concerning the intended users, the interviewees strongly remarked that interaction rather than automation happens when the engineer can recognize and communicate weaknesses and strengths that the architect may not be aware of. In the best current conditions of collaboration, an experienced engineer may further communicate and envision challenges, both threads and opportunities, that can provoke better solutions. Whereas the two proposed components do allow automatization of repetitive tasks, the information produced by the tool may appear sterile and more attention shall be paid in framing the tool to facilitate the engineer in communicating the critical aspects of the design, as by requirement (e).

5 CONCLUSIONS

Current computational tools have intrinsic limitations when facing the early conceptualization design phase. Under the aim to foster the integrated tectonic approach within this strategic design phase, the paper has identified through an Action Research analysis the kind of information meaningful to be processed at this phase and has delineated the desirable media and method of communication. Noteworthy, computational tools aimed at this early integrated design shall be perceived and act as communication and collaboration support between the architect and the engineer. Within the design workflow, the tool's input shall be general geometrical entities and its output shall be structurally valid divergent solutions. These shall at best be capable of provoking the users and yet stay within a design solutions space that, in according to the tectonic design, synthesizes structure and architecture. A suitable formulation of the shared design space that could support the collaboration of architectural and engineering professionals have been proposed, and the findings have been implemented in a tool storyboard also subjected to an Action Research analysis. Many aspects of the tool have had positive feedback to ease the difficulties and additional costs of early integration of the professionals. However, the tool shall be further developed concerning dedicated structural analysis and visual communication of critical aspects of the design. This has highlighted once more that the interpretation of output solutions as well as the recognition and communication of structural logic should rely on the structural professionals; if these qualities would be imbedded in the tool itself, they would imply a risk of the tool being a "black box" for architectural professionals.

ACKNOWLEDGEMENTS

The authors would like to thank all the selected professionals that participated in the research: Ronni Lundoff Stenlökke, Kenn Clausen, Martin Romby Hauge, Peter Vejrum, Kristjan Plagborg Nielsen, Jan Buthke, and Marcin Luczowski. Thank you all for your time and highly valuable dedication.

REFERENCES

- Braun, V. & Clarke, V. 2006. Using thematic analysis in psychology. *Qual. Res. Psychol.* 3(2): 77–101.
- Schön, D.A. 1992. *The reflective practitioner: How professionals think in action*. London: Routledge.
- Corneliusen, J. & Deleuran, S. 2021. Integrated Computational Design: A study proposing a new computational tool supporting early conceptual structural design. MS Thesis, Aarhus University.
- Cross, N. 2001. Designerly Ways of Knowing: Design Discipline Versus Design Science. *Des. Issues* 17(3): 49–55.
- Cross, N. 2018. Developing design as a discipline. *Journal of Engineering Design* 29(12): 691-708.
- Kemmis, S., McTaggart, R. & Nixon, R. 2014. *The Action Research Planner: Doing Critical Participatory Action Research*. Singapore: Springer.
- Kvale, S. 2007. *Doing Interviews*. London: Sage Publications.
- Mora, R., Bédard, C. & Rivard, H. 2008. A geometric modelling framework for conceptual structural design from early digital architectural models. *Adv. Eng. Informatics* 22: 254–270.
- Mueller, C.T. 2014. Computational Exploration of the Structural Design Space, Ph.D. Thesis, Massachusetts Institute of Technology.
- Purup, P. B. & Petersen, S. 2020. Requirement analysis for building performance simulation tools conformed to fit design practice. *Autom. Constr.* 116: 1–12.
- Rolvink, A., Mueller, C. & Coenders, J. 2014. State on the Art of Computational Tools for Conceptual Structural Design. *Proc. IASS-SLTE 2014 Symp.*