A Comprehensive Computational Tool for Performance-driven Reasoning in Floating Building Design and Its Evaluation

Livia Calcagni[1,2], Sridhar Subramani[3,4], Koen Olthuis[3,4]

[1]Sapienza University of Rome, Rome/Italy · livia.calcagni@uniroma1.it
[2]Technische Universität Wien, Vienna/Austria
[3]Technische Universität Delft, Delft/Netherlands
[4]Waterstudio.NL, Delft/Netherlands

Abstract: The emerging non-regulated incursion of urban life into the water realm in waterfront cities could represent a threat to the ecosystem if not adequately led and controlled by an interdisciplinary, sustainable, and data-driven approach. The paper provides a proof of concept for a computational design support system to advance performance-driven reasoning in floating building design. The proposed user interface design platform emphasises a quality score for several design aspects and the logical interaction between different potential choices to identify optimal decisions in the presence of trade-offs between two or more conflicting requirements. The design support system is based on a performance-based design-support framework for floating buildings developed in previous studies. The performance requirements are converted into quantitative, measurable indicators and introduced in a Grasshopper script to integrate and control the design workflow. A custom user interface to control Grasshopper definitions is designed using the Grasshopper plugin Human UI. The tool enables designers to interactively explore and evaluate different design options according to their design objectives. Practitioners have tested the tool by applying it to two existing projects. Its practical implementation provides valuable feedback before building a full-scope system and proves its feasibility and usability regardless of location. Despite being conceived for the design phase of new structures, the paper demonstrates its twofold purpose, as it can also be used for evaluating existing projects or built structures, hence used as a certification system. The potential for upscaling the design support tool to urban scales opens up new possibilities for sustainable and resilient urban development in waterfront cities.

Keywords: Floating buildings, performance-driven design, computational design, design support system, user interface

1 Introduction

Despite being highly exposed and vulnerable to a range of climate and ocean-compounded risks driven by climate change, low-lying delta and waterfront areas are home to a large portion of the global population. Given the physical boundaries of land, the pressures of urbanisation, and the threat of rising sea levels and flooding, water is increasingly emerging as a viable alternative for urban expansion. Floating urban development is gaining increasing attention and is becoming a component of city plans for sustainable development and climate adaptation. This emerging non-regulated incursion of urban life into the water realm could threaten the ecosystem if not adequately led and controlled by an, interdisciplinary, and sustainable design approach. The resilience and sustainability of floating structures are embodied in their ability to withstand climate-driven disasters, in the efficient integration of renewable energy sources, in zero soil consumption, and in their capacity to contribute to the regeneration of the surrounding environment. More specifically, floating urban development can increase river connectivity, enhance habitat biodiversity (DE LIMA et al. 2022), and re-

Journal of Digital Landscape Architecture, 9-2024, pp. 285-299. © Wichmann Verlag, VDE VERLAG GMBH · Berlin · Offenbach. ISBN 978-3-87907-752-6, ISSN 2367-4253, e-ISSN 2511-624X, doi:10.14627/537752027.

duce pressure on sand mining (KOEHNKEN & RINTOUL 2018). However, there is currently no blueprint or best practice guidance for designing floating buildings to help meet and overcome climate-driven challenges and support architects, urban planners, developers, and policymakers. It is mainly because floating buildings are not yet defined as a building typology but are classified as boats or barges (PIATEK 2016). This implies they are not expected to comply with laws and norms that regulate land-based buildings. Widening the building typology to water requires incorporating floating buildings into building codes (WANG 2021).

A performance-based design framework was developed by the author L. Calcagni (2024) and represents a first attempt to address the need for a comprehensive open-structured framework for planning and designing floating buildings. Another essential step in this direction is the report delivered within the Horizon 2020 Research Project Space@Sea (LIN et al. 2019), a catalogue of technical requirements and best practices for the design of floating structures. Within this study, the term performance-based design refers to a design methodology that focuses on achieving specific performance objectives, providing a structured approach to designing floating architecture projects considering the particular needs of a site, their relationships and trade-offs. It is argued that performance-based and performance-driven architectural design differ in that the latter involves computer-aided techniques so that the performance can be used as the criteria to truly drive the design. This paper represents a follow-up of the previous research and marks the shift from a performance-based framework to a performance-driven design support system.

In the last decade, there has been growing recognition of the importance of computational processes – mathematical, algorithmic and knowledge-based – to perform portions of the design, evaluation or construction of interfaces, as they allow the integration of all the parameters affecting the project into a single workflow (CARPO 2012, DADE-ROBERTSON 2013, MENGES & AHLQUIST 2011). Data-driven and integrated analytical computational methods facilitate the production and visualisation of nuanced conditions that develop over time and may underlie complex, multi-faceted design processes (HENSEL & SØRENSEN 2014). Architecture, environment, and inhabitants all perform and interact with one another, yielding perpetually complex behaviour (HENSEL 2010). This highlights how a synergetic understanding and multi-species approach is required to unlock these complex interactions for developing a comprehensive and interdisciplinary design-support tool in such a new and unexplored field.

The paper provides a proof of concept (PoC) for a computational design-support platform to advance performance-driven reasoning in floating building design. The design tool emphasises the logical interaction between the different potential choices to identify different scenarios and consequent optimal decisions in the presence of trade-offs between two or more conflicting requirements. The process of solving one problem in the design process can cause the other functions or performances to worsen. As a result, algorithm thinking is needed to reduce the number of parametric variations a designer may need to explore and enhance the designer's understanding of the interactions between the various performance variables. The tool can be framed within the broader design process, which starts with selecting a location and identifying design objectives.

2 Methodology

The PoC consists of a demo of the platform to demonstrate its feasibility and verify that the above-mentioned performance-based design framework for floating residential buildings has practical potential. The intent is to simplify testing the platform's functionality and receive valuable feedback from its practical implementation before building a full-scope system. In this case, the scope is limited in terms of the number of functionalities implemented in the platform. The main research questions are: To what extent does a performance-driven digital-support system (DSS) for floating residential buildings demonstrate feasibility and practical potential in a limited PoC implementation?

From a methodological point of view, the study was structured into four phases (Fig. 1):

1. **Translation of performance requirements into quantitative, measurable indicators**. This process is carried out for one class of demand and all the related performance requirements. It serves as a paradigmatic example to prove the feasibility of the process that can be replicated for all the other classes. The numerical indicators are identified using recent scientific literature, ISO standards and Eurocodes both for the built environment and the marine industry (shipping and offshore), floating building codes, international protocols and guidelines, and green building certification systems such as LEED, WELL, and BREEAM. The framework and indicators are subjected to validation by scientific experts and practitioners involved in environmental architecture, floating architecture, hydraulic and mechanic marine engineering, and urban ecology.
2. **Definition of a computational algorithm**. The development of this early-stage digital tool involves the creation of a script using the Rhino software and Grasshopper 3D programming language to integrate and control the workflow based on the performance indicators. Integrating performance-driven design support systems (DSS) with custom user interfaces (UI) into design software can significantly benefit early design phases by enabling methodological integration of performance evaluation tools within the design workflow. This approach, similar to the successful implementation in software like Rhino and Grasshopper, empowers designers to make informed decisions based on performance feedback throughout the design process, ultimately leading to improved design outcomes. Using Rhino allows Grasshopper to process data and output scores for each requirement and overall results. Rhino helps visualise data structure, correlations, and trade-offs, integrating a local custom user interface. Setting an evaluation model on a design platform enables constant interaction and improvement of the building's design process at an early stage.
3. **Creation of a custom user interface**. An interactive interface is created using the Grasshopper plugin Human UI. The Human UI interface enables the creation of Grasshopper apps with custom user interfaces for Grasshopper definitions. It allows the creation of tabbed views, dynamic sliders, pulldown menus, checkboxes, 3D viewports, web browsers, and interactive elements that are both functional and aesthetically pleasing. The tabbed view interfaces allow to organise the classes of demand into different sections, making the framework easier to navigate. The dynamic sliders enable the control of the values of Grasshopper parameters in real-time. This allows architects and other stake-

holders involved in the design process of floating buildings to interactively explore different design options and evaluate them according to their overarching design objectives.

4. **Application and validation of the tool on design scenarios**. The tool is used to evaluate existing projects in two different scenarios: one in Europe (Netherlands) meant for long-term residency and one in the United States (Miami) for temporary residential purposes. The tool provides an evaluation of the performance score of the overall design. The tool's output is shared with the architects involved in the design process of the two projects. They are asked to interact with the tool to identify possible improvements based on the performance score, address the trade-off elements, and eventually revisit the design process.

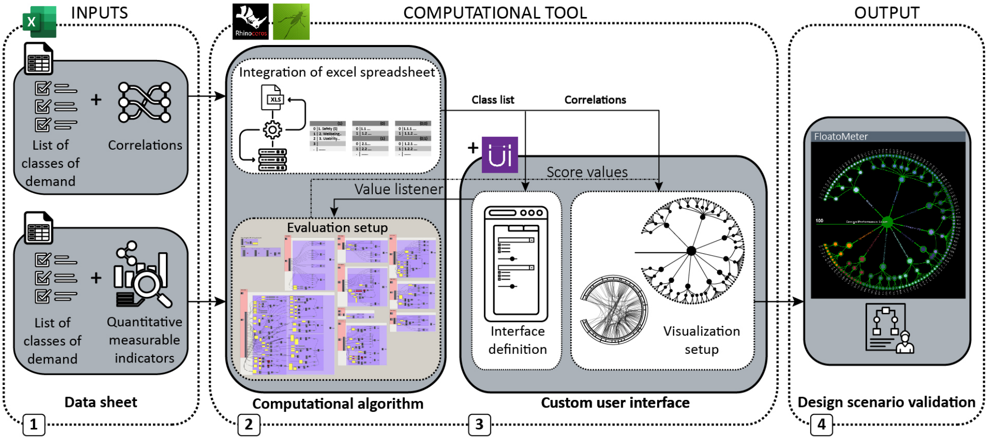


Fig. 1: Computational process behind the creation of the dashboard: (1) conversion into quantifiable indicators (inputs from excel spreadsheet data), computational tool development including (2) grasshopper workflow and (3) custom user interface, and (4) design evaluation (output)

3 Design Support System

3.1 Performance-Based Design-Support Framework

The performance approach behind the framework operates by design objectives, which are the clarification of the end user's needs, evaluated according to environmental, cultural, and economic factors. Classes of demand address design objectives and each class of demand is articulated into specific classes of requirements. Each class of requirements is addressed by specific performance requirements. The trinomial demand-need-requirement aims to ensure the overall quality of architecture, that is, the degree to which a set of intrinsic characteristics meets specific requirements.

The underlying assumption of the framework is that the environment is conceived as a host organism (ecosystem) and the floating facilities (or cities) as grafts. This implies that the design should meet human and ecological needs, adopt a multi-species design approach, and conceive architecture as embedded in the environment, with a view to ecosystem integration. The Performance-based Design-Support Framework (PDSF) classes of demand are the following: safety, wellbeing, usability, management, environmental regeneration, rational use of resources, integrability, buoyancy-stability, and plant system. The requirements are the transposition of a demand at a technical level. Each requirement refers to one or more elements of the building system, to a particular agent that motivates it, and to a particular condition of use. The previous study was limited to identifying and clarifying the set of requirements for each class of demand without defining quantitative indicators for each requirement. The requirements have been validated by expert reviewers from different disciplinary fields through the administration of an evaluative questionnaire and have been published in previous studies (CALCAGNI et al. 2024). Figure 2 shows the classes of demand and classes of requirements that compose the framework and the compatibility, complementarity, influential, and excludability relationships between the performance-requirements.

However, considering that a requirement, by definition, must be quantifiable with a system of parameters or indicators and that the term performance implies a measurable result, the following paragraph demonstrates how each requirement can be assessed and verified through quantifiable parameters.

Each performance requirement has been broken down into specific aspects. For instance, the requirement "Ventilation control" has been articulated into a set of indicators, including air change per hour (ACPH), Filtration MERV (Minimum Efficiency Reporting Value), Ventilation rate (CMH), Presence of natural ventilation in each room, and airspeed both in winter and summer. Baselines and targets (or ranges) have been established for each indicator based on scientific literature and the current state of building indicators in the jurisdiction or industry (standards, norms, and certification systems). While targets represent the desired level of performance that the design seeks to achieve, baselines represent the minimum required to comply with norms or quality standards. Some indicators are expressed as a range, like a temperature range indicator for the requirement "Indoor temperature level and control ($20°C < x < 26°C$), or as a threshold value that defines a condition that must be met for the requirement to be considered fulfilled. An example of a threshold value is the Ventilation rate, expressed as $CMH > 36\ m^3/h$. Instead, some other indicators consist of conditions that must be met and, therefore, are expressed by a binary indicator that denotes whether a specific requirement is met. The value 1 implies the indicator is met, and the 0 means it is not. This binary condition simplifies the evaluation process and reduces the need for subjective interpretations. For instance, one of the indicators of "quality views" is the view of natural elements. In this case, the binary representation of yes/no answers allows for efficient data processing within the tool. These three indicators (target ranges, threshold values, and binary representations) are the most widely used in certification systems that assess the quality of a design as they provide clarity and objectivity, which are crucial for ensuring fair and consistent certifications. With each condition represented by a numerical value, the certification system can easily calculate metrics such as the percentage of applicants meeting specific requirements or the overall design rate. When necessary, conversion factors are developed to translate incompatible measure units into values between 0 and 1.

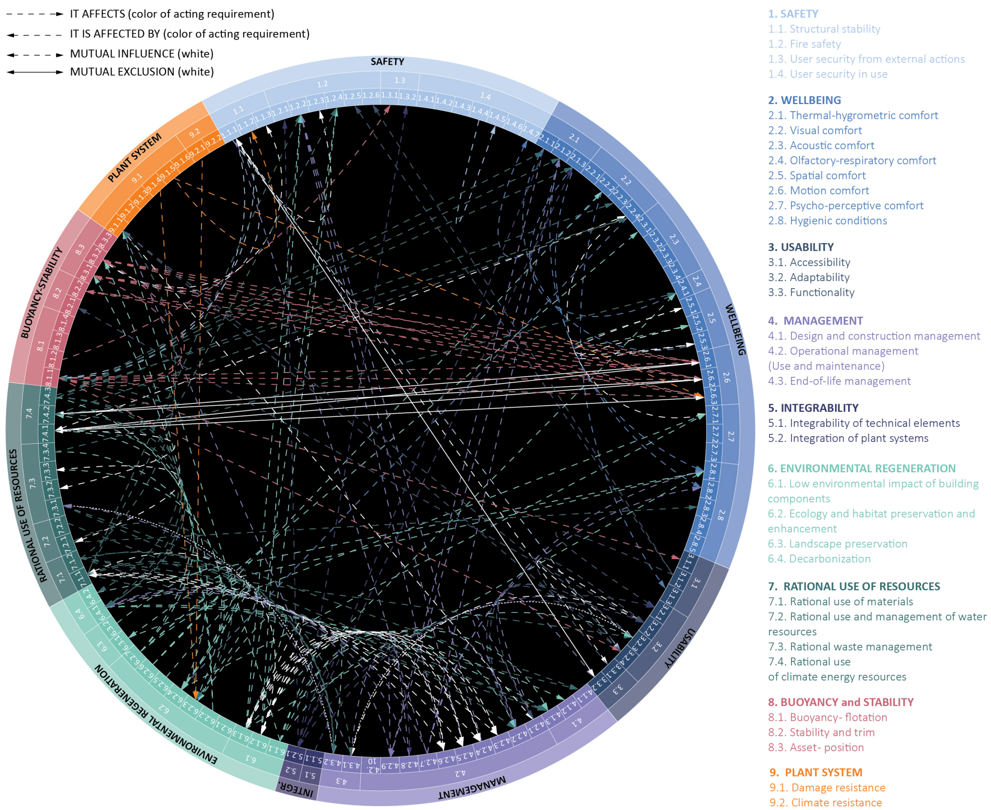


Fig. 2: Correlations and trade-offs between performance-requirements structured in their relevant classes of requirements and classes of demand. Source: Livia Calcagni

3.2 Computational Tool

The Grasshopper script at the core of the digital design support system (DSS) is divided into four parts:

1 – Integration of Microsoft Excel spreadsheet into the Grasshopper file/script. In this part of the definition, the spreadsheet defining the correlations and trade-offs between all the requirements, including mutual influence, one-sided influence (when one requirement affects another one, and consequently the latter is affected by the first one), and correspondence, is accessed within the grasshopper definition. The relations between requirements are transformed into a data tree structure for workflow use. Classes of demands represent a main tree branch, and the classes of requirements and requirements are sub-branches of the data tree. The definition of unique tree paths for each requirement highlights the correlations and displays the data structure for the interface.

2 – Interface definition. The human UI plugin enables the creation and design of the user interface (UI) platform. The tool opens up with a start screen that prompts the user to choose 'Create Project'. Following, a form pops up to input the project name and location coordinates.

In this way, the tool can geo-localize the project on a visual map, input default data (e. g., climate features) in the 'location' tab, and tailor the 'Performance Indicators' sections according to relevant regulations and context features. Once the 'New Project' is created and georeferenced, the interface is structured in four tabs.

'**About**': general information on the tool and instructions on how to use it.

'**Location**' tab (Fig. 3): the user can check and fill in the missing data regarding four fields.

- Space: country, surrounding buildings (average height), utility lines (presence and ease of access), distance to construction or shipyard (any workspace), presence of adequate waterways connected to water plot, distance to the nearest seaport.
- Water: plot size, (main) orientation, wave height, currents, level fluctuations, nautical activities, water temperature, salinity, biochemical water quality, soil consistency, bathymetry, soil displacement and sedimentation.
- Climate: average temperature, annual low extreme, annual high extreme, solar irradiance, mean relative humidity, annual precipitations, number of rainy days per year, average wind speed, and main wind direction.
- Environment: air quality, seismic risk, natural protected areas.

The location-based parameters inform and customize the system, determining the regulations to which the requirements must adhere and adjusting some indicators according to site-specific legal, climate, and environmental conditions. For instance, water conditions in the 'Water' category have a significant impact on the Buoyancy - Stability class of demand.

'**Performance Indicators**' tab (Fig. 4): the interface displays the nine classes of demand with checkboxes for users to select relevant ones for their project. Each class of demand has a dedicated tab with input fields for relevant indicators for each requirement. Units and conversion tools are readily integrated. Each input field collects a value for a measurable variable (indicator) that defines each requirement. The entry variables are represented as range sliders or empty boxes that collect a numeric value with appropriate measure units. Some variables, such as emergency lighting, accept a boolean value via a drop-down checklist. Each input value has a value listener, which feeds the input values into the evaluation system in real time as the user interacts with the tool. As users input data, the circular diagram ('Compliance Tracker') on the top right of the dashboard updates in real-time, showing which baseline requirements are met and which need adjustments. Clicking on a requirement on the checklist opens a pop-up window with justification for non-compliance and suggestions for improvement.

'**Performance Score**' tab (Fig. 5): the scoreboard shows the score for each requirement using intuitive gauges or progress bars (from green to red according to its degree of compliance). The overall performance score is displayed prominently, along with individual scores for each class of requirement and each class of demand.

3 – Evaluation set up. All entered input values from the user interface are fed into this section of the definition to process the evaluation. Based on the type of quantifiable or measurable indicators defined for each requirement, the input variables are normalised into a value range between 1 (optimal) and 0 (inadequate) by remapping the value range, converting them to Boolean values, or checking the conditional value range. The normalisation allows the evaluation of a score for each requirement and the average score for each class of requirements and a class of demand, thus also providing a final performance score for the design.

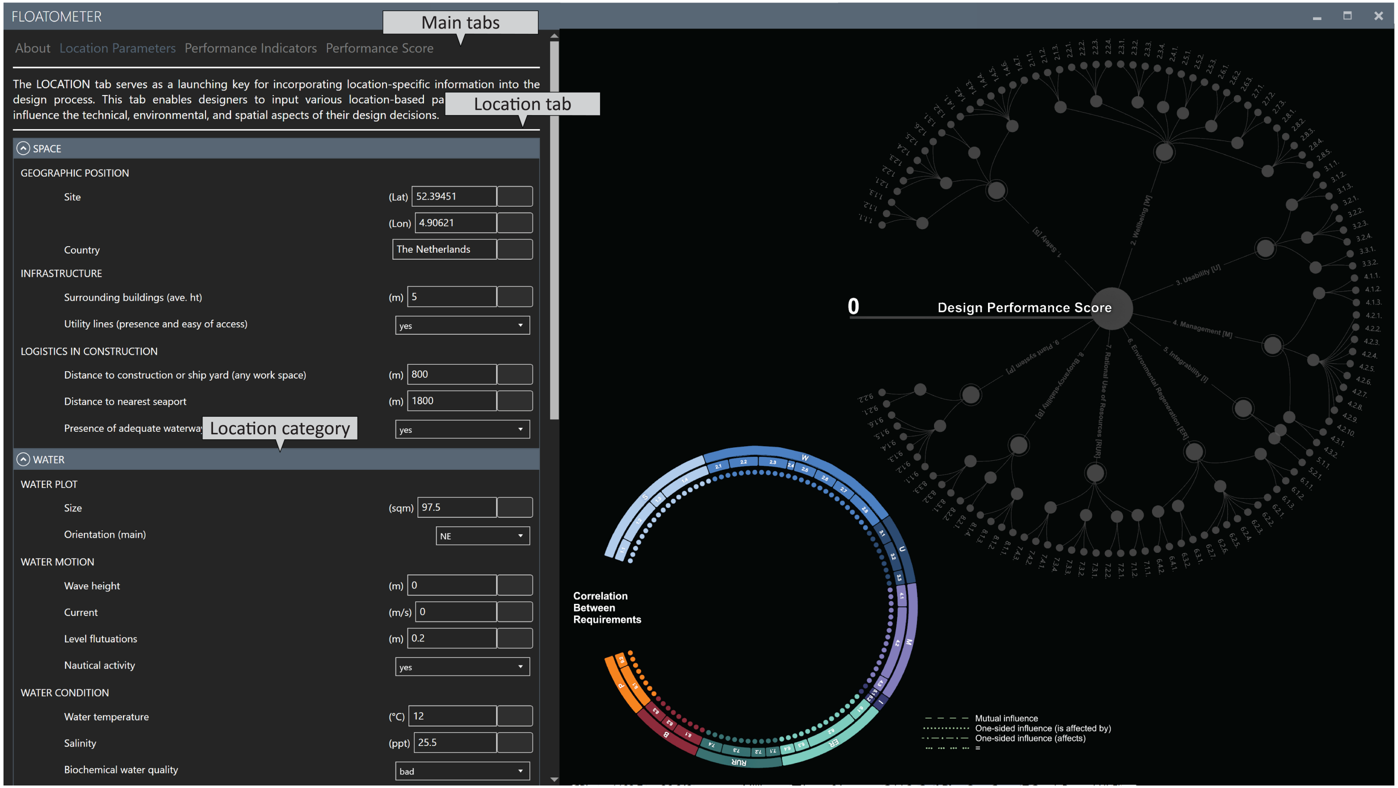
4 – Visualisation set up. The larger circular diagram ('Compliant Tracker') showcases the data structure, with the circle's colour defining each branch of a class of demand. The connecting lines help to visualise the branches of each class of demands. The colour range of the lines from red to green indicates the score of each requirement from inadequate (red) to ideal (green). By clicking on any requirement on the 'Compliance Tracker' it is possible to highlight the other requirements that are connected to it. The total performance score, expressed on the central line, spans from 0 to 100 and results from the average of the normalised values. The smaller circle on the bottom left of the dashboard ('Correlation Tracker') shows the correlations between requirements. Different line types display the type of relation mentioned above. When a checkbox is clicked on the interface, the visualization changes to reflect that specific requirement and displays both diagrams.

3.3 Application and Validation of the Tool

The emphasis on design gradually encompasses a range of specific local conditions and processes across spatial and time scales. Each design approach involves several dynamic processes, each with its duration, velocity, and timeline and specific geological, climate, environmental, cultural, social, and economic conditions. This strengthens the need for a data-driven design process that must be configured case-specifically. For this to be possible, it is necessary to consider the specifically relevant sets of data, their interrelations, and aspects of evaluation that characterise every context. This entails the need to consider location not as a class of demand within the DSS tool but as a starting condition that will affect the design process and the means to address each class of demand. Therefore, the DSS dashboard has an onboarding step that concerns the insertion of location-related information. The "Location" tab (Fig. 3) is a launching first dashboard section for incorporating location-specific information into the design process. This tab enables designers to input various location-based parameters that will influence the technical, environmental, and spatial aspects of their design decisions. The DSS tool must collect sufficient information to personalise the user experience and provide exact ranges and targets calibrated to the specific legal and climate region where the project is located.

The tool was applied to evaluate the quality of two existing projects to determine its applicability, replicability, and effectiveness in aiding the improvement of the design process. Two projects designed and built by the Dutch firm Waterstudio.NL were chosen for two main reasons:

1. Complete data and documentation (direct access to graphic and textual material).
2. Time efficiency: creating a new project from scratch would be time-intensive, allocating valuable resources that could be better used for focused evaluation of the design support tool.



The LOCATION tab serves as a launching key for incorporating location-specific information into the design process. This tab enables designers to input various location-based pa… influence the technical, environmental, and spatial aspects of their design decisions.

SPACE

GEOGRAPHIC POSITION
- Site (Lat) 52.39451
- (Lon) 4.90621
- Country The Netherlands

INFRASTRUCTURE
- Surrounding buildings (ave. ht) (m) 5
- Utility lines (presence and easy of access) yes

LOGISTICS IN CONSTRUCTION
- Distance to construction or ship yard (any work space) (m) 800
- Distance to nearest seaport (m) 1800
- Presence of adequate waterwa… yes

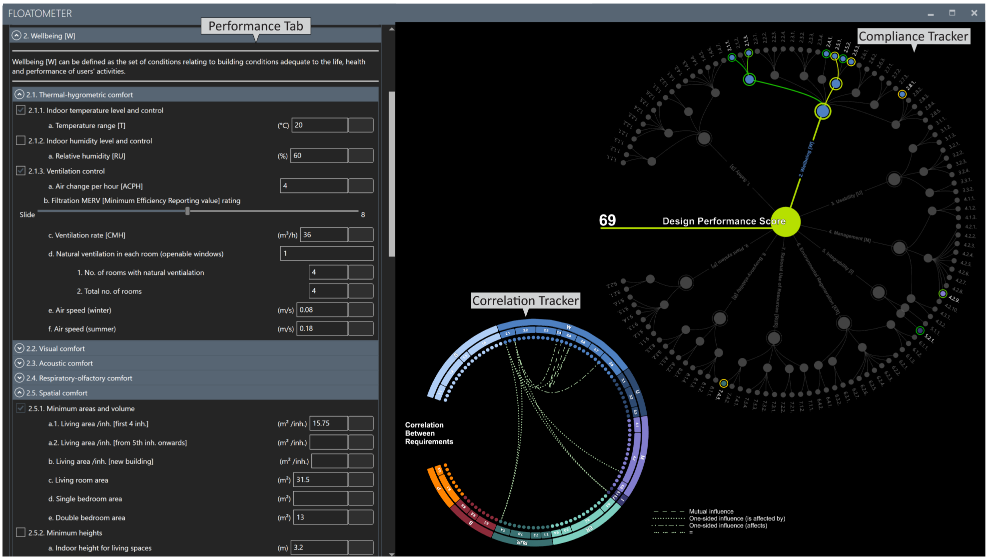
WATER

WATER PLOT
- Size (sqm) 97.5
- Orientation (main) NE

WATER MOTION
- Wave height (m) 0
- Current (m/s) 0
- Level fluctuations (m) 0.2
- Nautical activity yes

WATER CONDITION
- Water temperature (°C) 12
- Salinity (ppt) 25.5
- Biochemical water quality bad

Fig. 3: Interface for project 2. *Schoonship K.13* by Waterstudio.NL in Amsterdam, Netherlands: Location tab filled values – Step 1 (no evaluation process yet).



2. Wellbeing [W]

Wellbeing [W] can be defined as the set of conditions relating to building conditions adequate to the life, health and performance of users' activities.

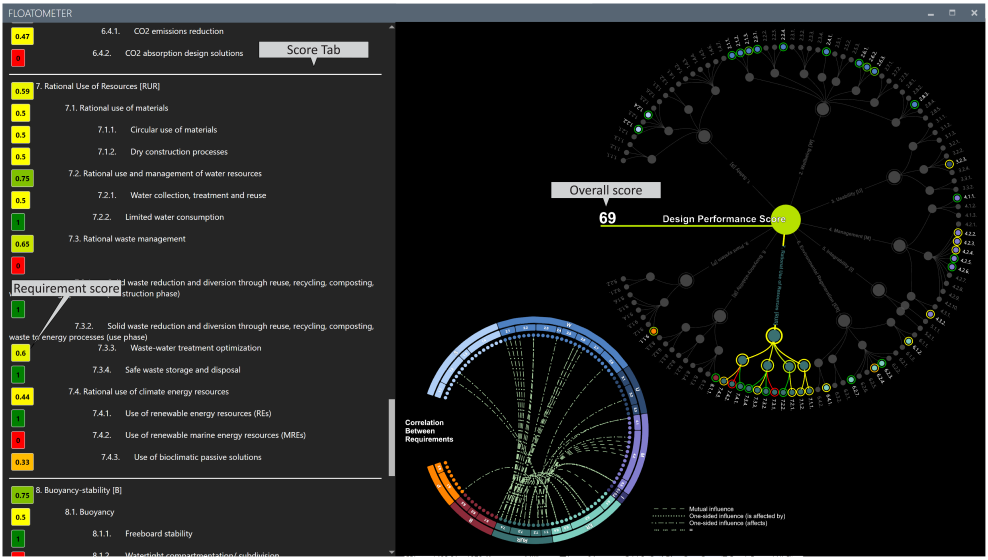
2.1. Thermal-hygrometric comfort

☑ 2.1.1. Indoor temperature level and control
a. Temperature range [T] (°C) 20
☐ 2.1.2. Indoor humidity level and control
a. Relative humidity [RU] (%) 60
☑ 2.1.3. Ventilation control
a. Air change per hour [ACPH] 4
b. Filtration MERV [Minimum Efficiency Reporting value] rating
Slide 8
c. Ventilation rate [CMH] (m³/h) 36
d. Natural ventilation in each room (openable windows) 1
1. No. of rooms with natural ventialation 4
2. Total no. of rooms 4
e. Air speed (winter) (m/s) 0.08
f. Air speed (summer) (m/s) 0.18

2.2. Visual comfort
2.3. Acoustic comfort
2.4. Respiratory-olfactory comfort
2.5. Spatial comfort

☑ 2.5.1. Minimum areas and volume
a.1. Living area /inh. [first 4 inh.] (m² /inh.) 15.75
a.2. Living area /inh. [from 5th inh. onwards] (m² /inh.)
b. Living area /inh. [new building] (m² /inh.)
c. Living room area (m²) 31.5
d. Single bedroom area (m²)
e. Double bedroom area (m²) 13
☐ 2.5.2. Minimum heights
a. Indoor height for living spaces (m) 3.2

Fig. 4: Interface for project 2. *Schoonship K.13* by Waterstudio.NL in Amsterdam, Netherlands: Performance Indicators tab filled in (Step 2)

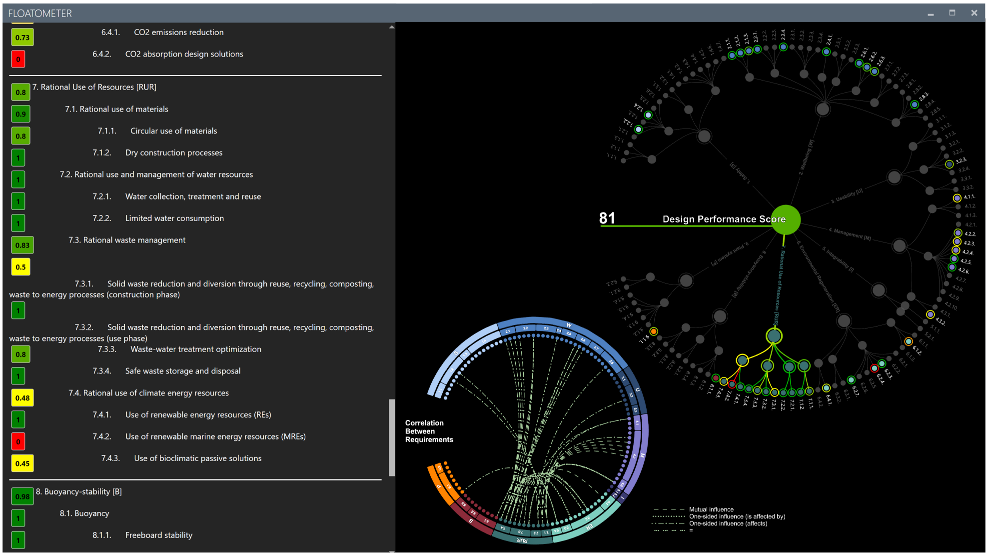


0.47 6.4.1. CO2 emissions reduction
0 6.4.2. CO2 absorption design solutions

0.59 7. Rational Use of Resources [RUR]
0.5 7.1. Rational use of materials
0.5 7.1.1. Circular use of materials
0.5 7.1.2. Dry construction processes
0.75 7.2. Rational use and management of water resources
0.5 7.2.1. Water collection, treatment and reuse
1 7.2.2. Limited water consumption
0.65 7.3. Rational waste management
0
d waste reduction and diversion through reuse, recycling, composting, struction phase)
1 7.3.2. Solid waste reduction and diversion through reuse, recycling, composting, waste to energy processes (use phase)
0.6 7.3.3. Waste-water treatment optimization
1 7.3.4. Safe waste storage and disposal
0.44 7.4. Rational use of climate energy resources
1 7.4.1. Use of renewable energy resources (REs)
0 7.4.2. Use of renewable marine energy resources (MREs)
0.33 7.4.3. Use of bioclimatic passive solutions

0.75 8. Buoyancy-stability [B]
0.5 8.1. Buoyancy
1 8.1.1. Freeboard stability
8.1.2. Watertight compartmentation/ subdivision

Fig. 5: Interface for project 2. *Schoonship K.13* by Waterstudio.NL in Amsterdam, Netherlands: Performance Score tab filled in (Step 3)



0.73 6.4.1. CO2 emissions reduction
0 6.4.2. CO2 absorption design solutions

0.8 7. Rational Use of Resources [RUR]
0.9 7.1. Rational use of materials
0.8 7.1.1. Circular use of materials
1 7.1.2. Dry construction processes
1 7.2. Rational use and management of water resources
1 7.2.1. Water collection, treatment and reuse
1 7.2.2. Limited water consumption
0.83 7.3. Rational waste management
0.5 7.3.1. Solid waste reduction and diversion through reuse, recycling, composting, waste to energy processes (construction phase)
1 7.3.2. Solid waste reduction and diversion through reuse, recycling, composting, waste to energy processes (use phase)
0.8 7.3.3. Waste-water treatment optimization
1 7.3.4. Safe waste storage and disposal
0.48 7.4. Rational use of climate energy resources
1 7.4.1. Use of renewable energy resources (REs)
0 7.4.2. Use of renewable marine energy resources (MREs)
0.45 7.4.3. Use of bioclimatic passive solutions

0.98 8. Buoyancy-stability [B]
1 8.1. Buoyancy
1 8.1.1. Freeboard stability

Fig. 6: Interface for project 2. *Arkup* by Waterstudio.NL in Miami, Florida: Performance Score tab filled in (Step 3)

The first project is Schoonship K.13 (Fig. 7), a floating house part of a large-scale (46 households) floating energy-neutral and circular district in the Johan van Hasseltkanaal in the north of Amsterdam, Netherlands. The other project, meant for temporary residency, is Arkup (Fig. 8), a luxury, self-sufficient floating villa in Miami, Florida. It is classified as a pleasure boat, so legally, it is a boat, but it is habitable like a house.

Fig. 7:
Schoonship K.13 by Waterstudio, Amsterdam (NL). Source: Waterstudio.NL.

Figures 3, 4, and 5 refer to Scoonship K.13 and display different screen captures of the evaluation process within the dashboard: Location section, Performance Indicators value insertion, and Performance score and evaluation. Figure 6 shows the Performance tab score for Arkup. The dashboard with which the designers can interact is on the left and the circular diagrams on the right display in real time the results for each class, all the selected correlations, and the overall performance score of the project.

Fig. 8:
Arkup by Waterstudio in Miami, Florida (USA). Source: Waterstudio.NL.

4 Discussion

Although both Schoonship K.13 and Arkup were revealed to be high-quality projects, with overall scores of 69 and 81, the tool shows how they could undergo further improvements. Unlike Schoonship, Arkup is equipped with smoke detectors, alarms, sprinklers, fire extinguishers, and several collision risk reduction arrangements, including fenders and a watertight compartmentalised pontoon. These provisions increase Arkup's overall score in terms of safety. A watertight subdivided pontoon and sink risk prevention indicators (water leak detectors and bilge pumps) are not present in Schoonship that, on the other hand, is equipped, unlike Arkup, with rubber seals around the mooring poles to reduce squeaking, rubbing, and creaking during vertical movement of the structure due to water level fluctuations. On the other hand, Schoonship, unlike Arkup, implements several passive strategies to improve the building envelope performance and nature-based solutions to promote biodiversity. The dashboard highlights how both projects could improve regarding environment regeneration, especially ecology and habitat preservation and enhancement, CO2 emission and absorption strategies, and landscape preservation, as not all requirements are met within these categories. Moreover, Arkup and Schoonship K.13 could integrate marine-renewable energy production devices to meet the entire energy demand and become entirely self-sufficient. Schoonship K.13 could also integrate a rainwater harvesting system or a purification/desalination unit to increase water availability for domestic use. Both projects could improve within the class rational use of resources by integrating solid waste reduction and diversion through reuse, recycling, composting, and waste-to-energy processes both in the construction and operational phases. Arkup does not meet most of the spatial comfort requirements, and this is because it is registered as a boat rather than as a stationary residential building. Arkup has a very high overall performance score, presumably because of its incredibly high construction cost. Schoonship has lower construction costs but still represents a luxury district that does not provide a solution for affordable housing developments. For this reason, it should be essential to integrate economic considerations within the tool to evaluate the overall performance, considering construction and operational costs.

Two architects from Waterstudio.NL, who were in charge of the design of the two case studies, were asked to use the tool to evaluate their project and identify possible improvements. Feedback on the functionality and potential of the tool was provided through unstructured interviews. This least rigid type of interview, with no predetermined set of questions, was chosen to get a more holistic view of the interviewee's experiences. The evaluation criteria (requirements) and methods categorised in different disciplinary classes were regarded as well-structured and comprehensive. The interviewed architects pointed out that the tool should have an additional instruction manual clarifying each parameter. Overall, they underlined that it is currently more suitable for evaluating the final design rather than for supporting the design process from the very initial phase, as it is hard to fill in all values at an early stage of the design process. They also suggested that public officers could use the tool as a certification/evaluation system to check if projects meet specific standards and criteria. In order to make it work effectively as a design support tool for architects, it should be split into different sections referring to the different design stages and integrate a 3D design model. Therefore, to embrace all stages of the design process, the tool could be split into 3 or more chronological sections that refer to the different design stages. In this case, some detailed requirements that refer to final stages of the project should be removed in the first section for instance.

5 Conclusions

This study aimed to assess the feasibility and practical potential of a performance-driven digital design support system (DSS) for floating residential buildings and to identify critical aspects requiring further refinement or expansion considering the limited PoC implementation. First of all, the development of an algorithm thinking process for one class of demand proves the feasibility of applying the methodology to any other class and the replicability of the whole process on the rest of the framework, notwithstanding the limited PoC implementation. Not only was the potential practical use of the tool confirmed by the interviewed designers, but novel applications also opened up. What emerges from applying the tool to existing designs is that the dashboard can be used both for design and evaluation purposes. In the design phase of new buildings, it acts as a set of guidelines and checklist in the initial stages and as a design support tool in the technical final phases. At the same time, through slight adjustments, the tool could be applied in evaluating existing projects or built structures and used as a certification system to evaluate their overall quality concerning the identified requirements.

Moreover, the development of a performance-driven design support tool for the small (building) scale shows the potential of upscaling the process to a cluster of buildings and eventually to the city scale through the integration of computational aid.

The authors are aware that the limited data available on floating buildings and floating urban development may represent a limit for the research due to the bias that can occur in the selection of indicators carried out by the authors. This limitation can be overcome by designing the tool as a flexible platform that undergoes constant development, thus allowing the inclusion of new metrics to improve multi-criteria optimization and accuracy. Another limit is the application of the tool to an existing project, where the adjustment phase, specifically the feedback loop between the project and the output of the DSS tool, is absent. Nevertheless, our work could be conceived as one of the first iterations of a project that could be improved using the output delivered by the tool for future versions.

Future studies could involve the integration of a 3D model to provide architects and designers with a tool aiding each stage of the design process rather than only design support for evaluation. Additionally, a set of default strategies could be incorporated to guide users in making improvements based on real-time dashboard data. The tool could be implemented as a standalone platform designed for uploading IFC files or as a plug-in for BIM software that already integrates data into the model. This enhanced version would allow users to initiate their design in 3D and receive real-time feedback on the performance of each framework category and each requirement. The tool could be further developed to include affordability information, including average rough costs (price ranges). Providing users with a general estimate of construction costs would enhance the tool's ability to assess proposed designs' economic feasibility and affordability. This feature would also facilitate comparing design scenarios based on cost implications.

References

CARPO, M. (2012), The digital turn in architecture 1992-2012. John Wiley & Sons.

CALCAGNI, L. (2024), A Performance-Based Design Framework for Floating Architecture. Trade-offs and correlations between requirements for multiple criteria decision making optimization. In Proceedings of the World Conference on Floating Solutions.

DADE-ROBERTSON, M. (2013), Architectural user interfaces: Themes, trends and directions in the evolution of architectural design and human computer interaction. International Journal of Architectural Computing, 11 (1), 1-19.

DE LIMA, R. L. P., DE GRAAF-VAN DINTHER, R. E. & BOOGARD, F. C. (2022), Impacts of floating urbanization on water quality and aquatic ecosystems: a study based on in situ data and observations. Journal of Water and Climate Change, 13 (3), 1185-1203.

HABIBI, S. (2015), Floating Building Opportunities for Future Sustainable Development and Energy Efficiency Gains. Journal of Architectural Engineering Technology, 04 (02). doi:10.4172/2168-9717.1000142.

HENSEL, M. U. & SØRENSEN, S. S. (2014), Intersecting knowledge fields and integrating data-driven computational design en route to performance-oriented and intensely local architectures. Footprint, 59-74.

HENSEL, M. (2010), Performance-oriented architecture: towards a biological paradigm for architectural design and the built environment.

KOEHNKEN, L. & RINTOUL, M. (2018), Impacts of sand mining on ecosystem structure. Process and Biodiversity in Rivers, 1-162.

LIN, F.Y., CZAPIEWSKA, K., IORGA, G., TOTOLICI, D. & KONING, J. (2020), A catalogue of technical requirements and best practices for the design- Deliverable. 7.2. Space@Sea.

MENGES, A. & AHLQUIST, S. (2011), Computational design thinking: computation design thinking. John Wiley & Sons.

MOON C. (2014), Three dimensions of sustainability and floating architecture. In International Journal of Sustainable Building Technology and Urban Development, Vol. 5, Issue 2, pp. 123-127). Taylor & Francis. Doi:10.1080/2093761X.2014.908809

NEBULONI, A. & BURATTI, G. (2023), Design by data. From interfaces to responsive architectures. TECHNE, 25, 93-100.

NEBULONI, A. & ROSSI, A. (2018), Codice e progetto. Il Computational Design Tra Architettura, Design, Territorio, Rappresentazione, Strumenti, Materiali e Nuove Tecnologie. Mimesis.

ROTH, S. F., KOLOJEJCHICK, J., GOLDSTEIN, J. (1994), Interactive graphic design using automatic presentation knowledge. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 112-117.

PIATEK, L. (2016), Displacing Architecture? From Floating Houses to Ocean Habitats? Expanding the Building Typology. Education for Research, Research for Creativity, 1, 273-280.

SHEPHERD, M. & BINTA, K. C. (2015), Climate change and African Americans in the USA. Geography Compass, 9 (11), 579-591.

WANG, K. F. (2021), An Investigation into Environment-Floating Buildings in relation to Sustainability Indicators for Planning and Design Process.