INTEGRATION, RELATION, AND RECREATION: RETHINKING ARCHITECTURAL PROGRAMMING

Bramasta Putra Redyantanu*
*) Department of Architecture, Petra Christian University, Surabaya, Indonesia
e-mail: bramasta@petra.ac.id

ABSTRACT

This study rethinks architectural programming as an innovative process, emphasizing the integration of contextual elements, the relational blending of multiple functions, and the adaptive reuse of existing spaces as recreation over time. Moving beyond the traditional understanding of programming as problem-finding-solving, this research redefines it as a dynamic framework bridging theoretical insights and practical design applications. Through a qualitative case study methodology, the paper examines three landmark projects with similar functions by Bjarke Ingels Group—8 House, The Mountain, and Urban Rigger—to explore the transformative potential of programming. The analysis positions programming as an integrative tool that aligns spatial, functional, and contextual dynamics to address both immediate and future architectural challenges. Program mixing with relational functions is conceptualized as a strategic approach that harmonizes diverse functions within a single design, fostering innovative and hybrid solutions. Adaptive reuse, reframed as a multi-time design response, focuses on revitalizing existing structures to meet evolving societal and environmental needs. These case studies illustrate how programming establishes a dynamic framework that enables architects to creatively reimagine constraints as opportunities. By emphasizing the principles of design analysis—realization, organization, and integration—this research contributes to the discourse on programming as a catalyst for architectural innovation and transformation. It proposes a shift in practice that highlights programming as a generative and responsive framework, inspiring a rethinking of architectural methodologies in the face of contemporary challenges.

Keywords: *Agenda, Context, Design, Housing, Program*

INTRODUCTION

This study rethinks architectural programming as an innovative process, emphasizing the integration of contextual elements, the relational blending of multiple functions, and the adaptive reuse of existing spaces as recreation over time. Architecture transcends its traditional role as merely addressing needs, evolving into a medium for generating and advancing knowledge. Powers (2007) highlights

design as a form of intellectual inquiry, emphasizing its capacity to produce knowledge and interdisciplinary exploration. Till (2012) reinforces this perspective by defining building construction as a research-driven process that merges experimentation, critical inquiry, and tangible realization. Verbeke (2013) complements these ideas by demonstrating how design through research enables architects to deeply explore conceptual and methodological challenges, fostering both innovation and intellectual growth.

Programming serves as the critical link between theory and practice, creating a pathway for integrating abstract ideas into practical applications. Cherry (1998, 2009) defines programming as a systematic process that aligns theoretical insights with functional realities, ensuring designs resonate with broader cultural and conceptual contexts. Furthermore, Plowright (2014) and Robinson & Weeks (1983) conceptualize programming as a dynamic design tool that guides the organization, integration, and realization of architectural components. By embracing programming as both a practical approach and a conceptual methodology, architects can navigate design complexities with adaptability and clarity.

The evolution of design and programming into tools for knowledge production underscores their transformative potential within architectural practice. Agrest (1977) challenges traditional notions of "design non-design," advocating for intentionality and specificity in architectural processes. This perspective invites architects to critically engage with interdisciplinary dialogues, redefining programming and design as generative frameworks that extend beyond solving immediate problems. Through this lens, architecture becomes a dynamic platform for integrating diverse contexts, blending functions, and adapting spaces across time, fostering meaningful innovation and addressing broader societal challenges.

THEORY / RESEARCH METHODS

Program as a Source of Unity

The concept of rethinking the program in architecture invites a broader exploration of its multidimensional aspects, shifting beyond its conventional utilitarian role. Summerson (1957) emphasizes the program as a "source of unity," underscoring its foundational importance in interlinking spatial dimensions, relational dynamics, and physical conditions within design. Similarly, Zevi (1957) reinforces this notion, describing the program as a cohesive framework that harmonizes various design elements. Gropius (1965) expands on this perspective, highlighting the intrinsic connections between architectural elements and functions, advocating for an integrated approach to programming. These views collectively position the program as a mediator that balances spatiality, functionality, and relational integrity.

In recent decades, architectural programming has evolved to encompass new paradigms, embracing innovation and contextual responsiveness. Vidler (1996, 2003) advocates for programming that explores the potential of materials, structures, social and cultural dynamics, and formal expression—reimagining it as both a creative and reflective process. Vidler's framework emphasizes the transformation of

raw data into meaningful forms, integrating environmental concerns, technological advancements, and formal inventions to expand the scope of programming in architecture.

Sanoff (2016) extends the discourse by positioning programming as more than problem-finding-solving, conceptualizing it as a means to develop sophisticated models that bridge theoretical insight with practical application. Programming, in this view, plays a critical role in reshaping architectural discourse, fostering holistic and adaptive design practices capable of addressing contemporary challenges and aspirations. By incorporating these diverse perspectives, programming emerges as a pivotal instrument for innovation and interdisciplinary exploration, redefining its role in the architectural design process.

Rethinking Programming in the Complexity of Contemporary Design

Programming through complexity in architecture reflects the evolution of design methodologies shaped by advanced computational tools, digital methods, and emerging technologies. Burry (2011) underscores the importance of scripting, computational approaches, and fabrication techniques, enabling architects to manage intricate design processes effectively. The integration of parametric and generative systems, as discussed by Burry (2011) and Coates (2010), facilitates simulation-based exploration of complex forms and adaptive designs. By utilizing algorithmic techniques, as highlighted by Caetano et al. (2020), architects can generate sophisticated designs that respond dynamically to environmental conditions and user interactions. Within this framework, programming emerges not only as a problem-solving tool but as a transformative process for fostering innovation and creative engagement.

The adoption of digital methodologies has expanded opportunities for sustainable practices and interdisciplinary collaboration. Jin & Tu (2024) emphasize how digital programming intersects with social service and education, enabling architects to address broader societal challenges. Building Information Modeling (BIM), as explored by Barekati et al. (2015), provides a comprehensive framework for managing complex projects by aligning architectural designs with construction data, performance analytics, and environmental considerations. Vesselov & Davis (2019) argue that digital data plays a vital role in enriching programming by enhancing precision, adaptability, and contextual responsiveness. These advancements contribute to a holistic approach that balances technological innovation with ethical and environmental concerns.

Programming in architecture is further enriched by systems and methodologies that analyze extensive datasets to propose optimal solutions tailored to project-specific needs. Chaillou (2022) and Steenson (2022) emphasize how advanced computational techniques enable architects to address both present and future user requirements, aligning with Hershberger's (2015) advocacy for user-centric programming. Furthermore, Anders (2003) and Oxman (2012) highlight programming's capacity to integrate multiple aspects—ranging from material selection to spatial organization—into cohesive and innovative designs. Computational processes, combined with generative methodologies, position

programming as an iterative, data-informed approach that redefines traditional boundaries in architectural practice.

Programming through complexity aligns closely with the proposed dynamic framework of integration (multi-context), relation (multi-function), and recreation (multi-time). Integration addresses the alignment of diverse contextual layers within architectural design, ensuring responsiveness to environmental, cultural, and societal factors. Relation fosters the coexistence and blending of multiple functions within cohesive designs, providing flexibility and synergy between spatial elements. Recreation emphasizes the adaptive reuse of spaces over time, addressing evolving needs while maintaining spatial relevance. This triadic framework offers architects a roadmap to navigate complex design challenges effectively, fostering adaptability, resilience, and innovation within the architectural process.

Context and Issue as Programming Dynamic Expansion

The evolving context and challenges in architecture demand a shift from conventional problem-solving approaches to innovative design propositions that respond to present and future possibilities. Clark (2009) and Jones (1992) emphasize the need to consider the "future state of situations" in architectural design, urging adaptability and foresight to address environmental, social, and technological changes. This perspective transcends static solutions by encouraging architects to envision adaptable and resilient frameworks capable of responding to evolving dynamics. In a "rules-type world," as described by Schön (1988), the integration of structured design principles with creative interpretation enables architects to engage with complexity and unlock new potential within the design process.

Alexander's (1964) concept of achieving a "good fit" in architecture highlights the importance of harmonizing user needs, contextual constraints, and spatial coherence within the design program. Hershberger (2015) expands on this by advocating for designs that address both present and future user requirements, emphasizing foresight and adaptability. These principles align seamlessly with the integration, relation, and recreation framework. Integration ensures harmony between user needs and contextual dynamics; relation fosters innovative program mixing within design; and recreation frames adaptability to address societal and environmental shifts over time. By embedding these dimensions within programming, architects can develop solutions that transcend functionality and engage with broader societal and ecological considerations.

Architectural programming thus evolves from problem-solving to proposition-driven innovation, transforming the design process into an opportunity for creative inquiry and adaptation. Future-oriented perspectives ensure architectural solutions remain relevant in a rapidly changing world, empowering architects to redefine boundaries and explore dynamic methodologies. This study reflects on programming as a dynamic platform for integration, relation, and recreation, rooted in the relevance of issues and context as foundational pillars for innovation. Through an examination of three case studies based on innovative programming by a single architect, it expands perspectives on program, programming, and reprogramming in architecture.

Method

This study employs a qualitative case study approach (Groat and Wang, 2013; Creswell, 2018) to examine the concept of program, programming, and reprogramming in architecture. By utilizing a theoretical lens of program in architecture (Summerson, 1957; Vidler, 2003; Koolhaas *et al.*, 2006; McMorrough, 2006), the research analyzes the program not only as a set of functional requirements but as a dynamic and integrative tool that bridges design theory and practice. Three architectural projects by Bjarke Ingels Group (BIG)—8 House, The Mountain, and Urban Rigger—are selected as the objects of study. These projects serve as mediums through which to explore how programming can transcend traditional problem-solving to foster innovation and adaptability in architectural design.

The analysis focuses on each project's contextual relevance and its alignment with innovative programming strategies (Lucas, 2016). The study investigates how these works reflect the integration of user needs, spatial relationships, and contextual challenges, utilizing programming as an analytical and generative framework. The research critically examines how BIG's design approaches demonstrate the interplay of program, programming, and reprogramming, offering insights into the evolution of architectural methodologies. Through these case studies, the research aims to contribute to the discourse on programming as a tool for expanding design possibilities and responding to contemporary architectural challenges.

The Framework

The framework of programming in architecture, based on the principles of realization, organization, and integration, serves as an analytical tool for rethinking the architectural program as a dynamic mechanism for expanding design possibilities. Koolhaas (2006) argues that programming functions more as an agenda rather than a neutral process, inherently shaped by preconceptions that challenge traditional norms and foster innovative solutions. The realization phase involves defining key project elements such as spatial needs and functional objectives, with McMorrough (2006) emphasizing the importance of brief designation and precise tabulation of quantities to establish clarity in complex architectural projects. By integrating reorganization and re-evaluation, programming evolves into a flexible and adaptive process, enabling architects to navigate uncertainty and explore new methodologies. The organization phase expands programming by synthesizing spatial relationships, user interactions, and contextual factors, with Koolhaas (2006) framing it as an opportunity to generate form through the mixing of spaces, unexpected configurations, and the intersection of spatial envelopes with movement vectors. This transforms programming into an innovative tool that encourages experimentation with unconventional layouts and hybrid typologies. The final phase, integration, ensures coherence between realization and organization, combining these elements into a unified framework that responds to dynamic design demands. McMorrough (2006) supports re-evaluation advocating for continuous refinement to align programs with evolving project requirements. Together, realization, organization, and integration establish a transformative framework that transcends

problem-solving, unlocking new possibilities within architectural programming. Figure 1 illustrates this conceptual framework, serving as an analytical tool for the case studies.

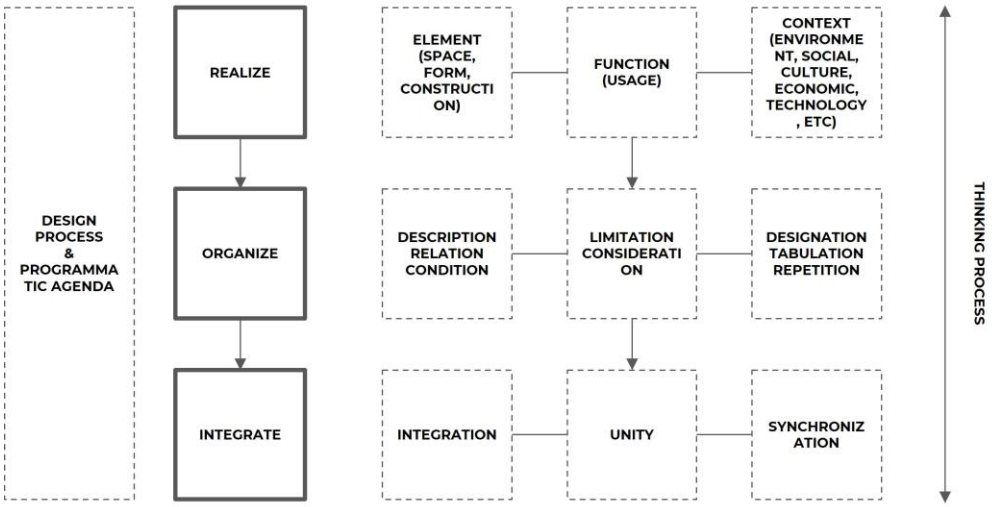

Figure 1. Basic Thinking of Programming in Architecture

Creative programming in architecture opens pathways for adaptive responses that transcend conventional practices, enabling the expansion of program to reprogramming as a transformative process. Program, traditionally viewed as a static list of spatial requirements, becomes a dynamic framework when enriched by programming—a process that organizes, integrates, and synthesizes diverse elements such as user needs, contextual conditions, and functional goals. Reprogramming further extends this notion by re-evaluating and redefining established programs in response to changing circumstances, offering opportunities for innovation and transformation. Zhuang et al. (2023) highlight the role of adaptive reuse and evaluation in reprogramming, emphasizing how architectural projects can evolve in response to new contexts and demands. The iterative nature of programming also allows for feedback loops, as noted by Andreu & Oreszczyn (2004), ensuring that design processes are continuously refined to address emergent challenges and opportunities.

Adaptive programming fosters resilience and versatility, expanding the scope of program, programming, and reprogramming to accommodate dynamic design possibilities. Andaloro et al. (2022), Redyantanu (2023), and Vidler (2003) emphasize the importance of adaptive programming in responding to social, cultural, everydayness and environmental conditions. This approach not only addresses functional needs but also incorporates broader considerations such as sustainability and innovation. Reprogramming enhances these adaptive capabilities by fostering creative exploration and aligning projects with evolving user requirements and technological advancements. Together, programming—through the integration of multi-context, the relation of multi-function, and the recreation and adaptation of

spaces over time—redefines architectural practices, transforming them into interconnected processes that merge creativity with structured methodologies. This expanded framework equips architects with new tools to push the boundaries of design, bridge theoretical insights with practical applications, and reshape the discipline to effectively respond to contemporary challenges.

RESULTS AND DISCUSSION

8 House: Architectural Programming as Multi-Contextual Integration

The 8 House by Bjarke Ingels Group (BIG) exemplifies how architectural program can serve as a dialogue with the surrounding context, integrating urban, social, and environmental dimensions into its design (*8 House / BIG | ArchDaily*, no date). Situated on the edge of Copenhagen, the 8 House reimagines the traditional urban block by stacking residential, commercial, and office spaces into horizontal layers, creating a vibrant, mixed-use community. This approach reflects a deep engagement with the site's context, as the design not only accommodates diverse functions but also fosters interaction between suburban and urban lifestyles. The continuous promenade and cycling path that ascend to the 10th floor symbolize this dialogue, connecting the building's internal program with the broader urban fabric.

The building's form and spatial organization further illustrate how programming can respond to environmental and social conditions. The figure-eight shape of the 8 House allows for optimal daylight access and views, while the sloping green roofs reduce the urban heat island effect and enhance ecological sustainability. These design elements demonstrate how programming extends beyond functional requirements to address environmental concerns and create meaningful connections with the natural surroundings. By integrating green spaces and communal areas, the 8 House fosters a sense of community and interaction, aligning the architectural program with the social dynamics of its context.

Finally, the 8 House showcases how program in architecture can transcend problem-solving to become a tool for innovation and contextual dialogue. By blending diverse functions, responding to environmental challenges, and fostering social interaction, the project redefines the role of programming as a mediator between architecture and its surroundings. This approach not only enriches the design process but also highlights the potential of program to create spaces that are both functional and deeply connected to their context. Figure 2 illustrates the elements of the program in the design process, structured within the framework of realization, organization, and integration of the design.

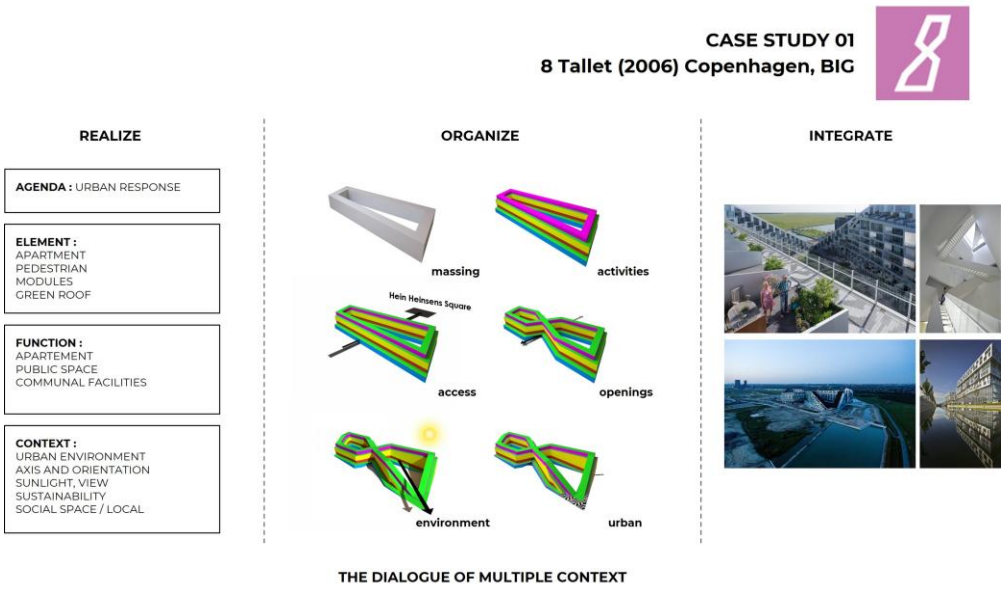

Figure 2. 8 House Program in Design Framework
Source: Edited from Archdaily

The 8 House by Bjarke Ingels Group (BIG) serves as a compelling example of program realization in architecture, emphasizing the integration of elements, functions, and contextual dynamics. The program realizes key elements such as residential, commercial, and office spaces, layered horizontally to create a mixed-use community. These layers address diverse functions, blending private living spaces with public facilities, enabling seamless interaction among the building's inhabitants. Contextually, the 8 House embraces its location at the edge of Copenhagen, responding to the suburban-urban transition by incorporating communal green spaces and a continuous promenade that connects the building with its surroundings. The figure-eight form further maximizes daylight access, enhancing livability and sustainability within the design.

The 8 House also excels in organizing its program by engaging in a dialogue with the surrounding multiple urban contexts and achieving harmonious integration within the design. The sloping green roofs not only merge the architecture with the natural environment but also encourage social interaction through accessible communal spaces. The cycling and walking path, which ascends to the 10th floor, acts as a dynamic spatial element that integrates movement vectors and spatial envelopes, demonstrating the building's innovative organizational strategy. Through this approach, the 8 House exemplifies the synthesis of functional elements, spatial relationships, and contextual responsiveness, showcasing programming as a tool for achieving both practical and visionary architectural outcomes. The integration of these aspects highlights the project's success in transforming programmatic requirements into a cohesive and meaningful design.

The Mountain: Architectural Programming as Hybrid Multi-Function Mixing

The Mountain Dwellings by Bjarke Ingels Group (BIG) exemplify programming as a strategy for mixing diverse functions, seamlessly integrating parking and residential spaces into a cohesive architectural solution (*Mountain Dwellings / PLOT = BIG + JDS | ArchDaily*, no date). Located in Copenhagen, the project combines 2/3 parking and 1/3 residential functions, merging them into a single structure rather than separating them into distinct buildings. This innovative approach transforms the parking area into a base for terraced housing, creating a symbiotic relationship between the two programs. The design ensures that the parking area remains functional and accessible, while the residential units above benefit from sunlight, fresh air, and panoramic views, demonstrating how programming can harmonize seemingly disparate functions.

The project's design further highlights the creative potential of programming in generating unexpected spatial configurations. The terraced housing units cascade down from the 11th floor to the street level, resembling a suburban hillside within an urban context. Each apartment features a private roof garden, blurring the boundaries between indoor and outdoor spaces and fostering a sense of community among residents. Meanwhile, the parking area below is designed with perforated aluminum facades that allow natural light and ventilation while forming a striking visual representation of Mount Everest. This interplay of form and function illustrates how programming can create dynamic relationships between different spatial elements, enhancing both usability and aesthetic appeal.

Finally, the Mountain Dwellings demonstrate how programming can expand architectural possibilities by integrating diverse functions into a unified design. By combining parking and residential spaces in a single structure, the project redefines the role of programming as a tool for innovation and adaptability. This approach not only addresses practical requirements but also creates a vibrant and sustainable living environment that bridges suburban and urban lifestyles. The Mountain Dwellings stand as a testament to the transformative potential of programming in architecture, showcasing how mixed-use strategies can lead to groundbreaking design solutions. Figure 3 illustrates the elements of the program in the design process, structured within the framework of realization, organization, and integration of the design.

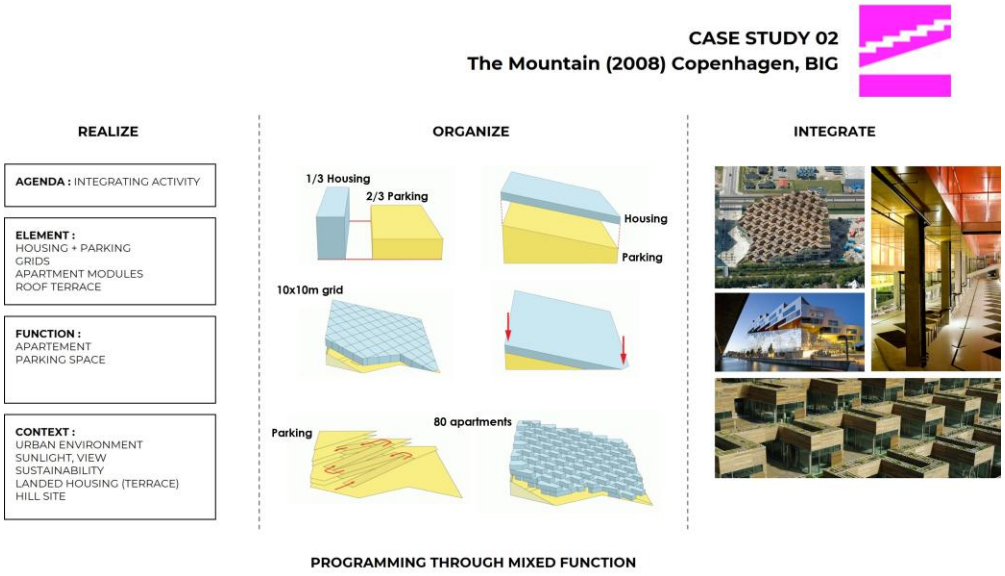

Figure 3. The Mountain Program in Design Framework
Source: Edited from Archdaily

The Mountain Dwellings by Bjarke Ingels Group (BIG) exemplify the realization of an architectural program through a creative synthesis of elements, functions, and contextual considerations. The project combines two distinct yet complementary programs: parking and residential spaces. The parking occupies the lower two-thirds of the structure, providing functional support for urban mobility, while the upper third comprises terraced residential units. Contextually, the project responds to its urban setting in Copenhagen by accommodating high-density parking needs and creating livable housing units with optimal sunlight, fresh air, and green spaces. This approach showcases the realization of a program that balances practical urban requirements with innovative living solutions.

In terms of organization, the Mountain Dwellings reimagine the relationship between these functions by mixing and integrating them into a cohesive design. The parking levels form the foundation of the structure, with perforated aluminum facades featuring an artistic representation of Mount Everest, allowing for natural light and ventilation. Above, the residential units are arranged in a cascading form, each with a private roof garden, creating a suburban feel within an urban environment. This integration blurs the boundaries between function and aesthetics, as the design harmonizes utilitarian parking requirements with the livability of the dwellings. By skillfully organizing and integrating these multi-functional elements, the Mountain Dwellings demonstrate the potential of programming to expand architectural possibilities, offering a dynamic and innovative response to urban challenges.

Urban Rigger: Architectural Programming as the Recreation of Multi-Possibility Adaptation

The Urban Rigger by Bjarke Ingels Group (BIG) exemplifies reprogramming as a means of exploring new possibilities in architecture, particularly through the innovative use of containers and a floating context for dwelling (*Urban Rigger / BIG | ArchDaily*, no date). Designed to address the growing demand for student housing in Copenhagen, the project reimagines the standard shipping container as a modular and flexible building block. By stacking nine containers in a circular arrangement, the design creates twelve studio residences surrounding a central winter garden, which serves as a communal space for social interaction. This approach demonstrates how reprogramming can transform an industrial object into a sustainable and adaptable housing solution, responding to the challenges of limited urban space and affordability.

The floating context of the Urban Rigger further expands the potential of reprogramming by introducing a new typology for urban living. Situated in Copenhagen's harbor, the buoyant structure utilizes underutilized water spaces to provide affordable housing while maintaining proximity to the city center. This innovative strategy not only addresses spatial constraints but also integrates environmental considerations, such as energy efficiency and reduced carbon footprints. The floating design allows for replication in other harbor cities, showcasing the versatility of reprogramming as a tool for addressing global housing challenges. By adapting to the unique conditions of its context, the Urban Rigger exemplifies how reprogramming can create resilient and forward-thinking architectural solutions.

Finally, the Urban Rigger highlights the transformative potential of reprogramming in architecture, redefining the relationship between form, function, and context. By leveraging the modularity of containers and the flexibility of floating structures, the project demonstrates how reprogramming can expand the boundaries of traditional design methodologies. This approach not only fulfills immediate housing needs but also introduces a scalable and sustainable model for future urban development. The Urban Rigger stands as a testament to the power of reprogramming to innovate and adapt, offering new possibilities for dwelling in an increasingly complex and constrained urban landscape. Figure 4 illustrates the elements of the program in the design process, structured within the framework of realization, organization, and integration of the design.

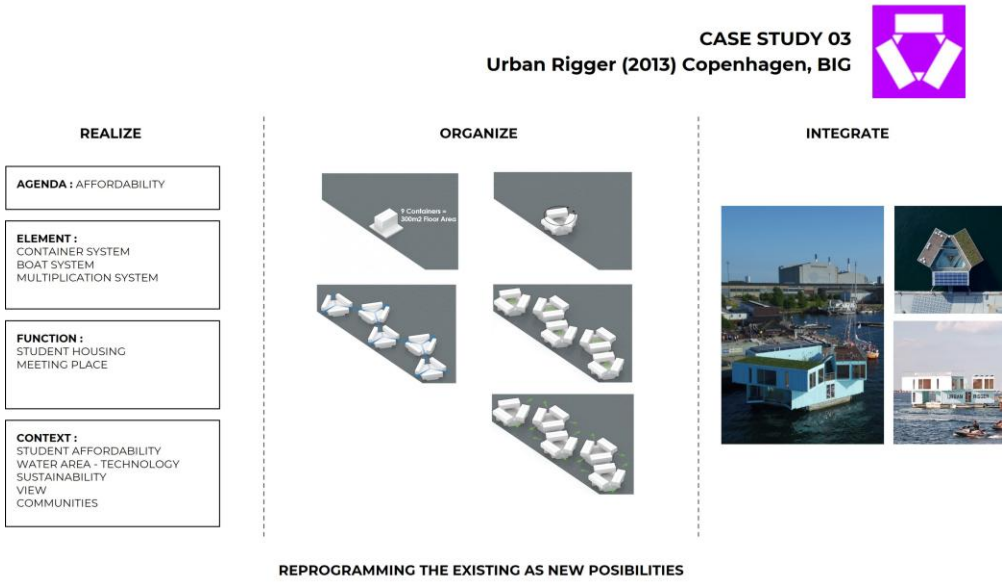

Figure 4. Urban Rigger Program in Design Framework
Source: Edited from Archdaily

The Urban Rigger by Bjarke Ingels Group (BIG) illustrates the realization of an architectural program by rethinking elements, functions, and context to address the growing need for sustainable and affordable housing. The project utilizes shipping containers as modular elements, repurposing them into functional studio residences. Each of the nine containers is configured to create twelve housing units arranged around a central winter garden, which serves as a communal space for social interaction. Contextually, the floating design leverages underutilized water spaces in Copenhagen's harbor, providing proximity to urban amenities while addressing spatial constraints. This innovative approach highlights the realization of a program that balances modularity and adaptability within an environmentally sensitive urban framework.

In terms of organization, the Urban Rigger transforms the conventional use of shipping containers into a cohesive design integrated within a floating context. The containers are arranged in a circular configuration, creating a self-contained housing solution that embraces modular reuse. This arrangement ensures efficient spatial organization while fostering a community-oriented environment through shared spaces. The floating context introduces adaptability to the design, allowing the structure to respond to urban density challenges while utilizing waterborne infrastructure. By successfully organizing and integrating these elements, the Urban Rigger exemplifies programming as a tool for creating sustainability in multi-time and forward-thinking housing solutions that redefine the boundaries of architectural design.

Dynamic Programming in Architecture: Integration, Relation, and Recreation

Architectural programming, while traditionally focused on function and spatial needs, has evolved into a dynamic framework that adapts to contextual, functional, and temporal dimensions. Bjarke Ingels Group (BIG) exemplifies this evolution through three distinct projects—*8 House (8 Tallet)*, *The Mountain*, and *Urban Rigger*—each serving as a dwelling while employing unique programming strategies. Despite their shared residential function, these projects demonstrate how programming can integrate urban conditions, relate multiple functions within a hybrid typology, and recreate structures to respond to evolving spatial demands over time. Through this lens, programming becomes a transformative tool, shaping architecture beyond static problem-solving into an adaptive and generative process.

8 House embodies integration (multi-context) by merging diverse urban functions and promoting interaction between residential, commercial, and public spaces. Its open circulation and diagonal boulevard create a micro-city, seamlessly integrating multiple urban layers into a cohesive form. The programming strategy goes beyond conventional residential design by embedding commercial activity, communal spaces, and a dynamic relationship with its surroundings. This integration allows architecture to engage with evolving urban contexts, ensuring adaptability and long-term relevance.

In contrast, *The Mountain* exemplifies relation (multi-function) by blending disparate functions within a hybrid spatial configuration. Rather than segregating residential and parking facilities, the project interweaves them into a cascading landscape where homes are built atop terraced parking structures, creating a synergy between utility and livability. This approach shifts programming from compartmentalization to relational fluidity, enabling functions to coexist in an interconnected manner that enhances spatial efficiency and user experience.

Finally, *Urban Rigger* embodies recreation (multi-time) by repurposing shipping containers into floating modular housing. This project reflects programming's capacity to embrace adaptive reuse as a response to shifting spatial needs and environmental challenges. By transforming temporary structures into permanent, flexible dwellings, *Urban Rigger* illustrates how programming can extend beyond immediate problem-solving to future-oriented spatial strategies. It reimagines architecture as an evolving entity, continuously adapting to new demands and possibilities. Figure 5 illustrates a comparative analysis of dynamic programming processes within similar functions.

Programming	8 Tallet	The Mountain	Urban Rigger
Location (Year)	Copenhagen (2006)	Copenhagen (2008)	Copenhagen (2013)
Function	Apartment + Public Facilities	Apartment + Parking Building	Apartemen + Community
Concept / Agenda	Public Building : Urban Response	Integrated Building : Integrating Activity	Portability & Multiplication : Adaptive Reuse
Context	Urbanscape, View, Communities, Sustainability	Integrating Parking, Modularity, Terrace System	Multiplicability, Adaptability, Network Container System, Student Affordability
Organization	Deformation building based on urban context	Integration on mixing two elements	Reuse and linking with portability
Unity	Form (Urban Response) - Space (Communal Space) - Context (Public Building)	Form (Mountain Like) - Space (Integrated Parking) - Context (Terrace & Parking)	Form (Container Unit) - Space (Comunal Network) - Context (Modularity)

Figure 5. Programming Dynamic Comparison within Similar Function

Together, these projects reinforce the concept of dynamic programming as an evolving architectural practice—one that integrates multi-contextual environments, relates multiple functions within adaptable frameworks, and recreates spatial solutions to respond to shifting conditions. By embracing programming as an iterative and transformative tool, architects can move beyond prescriptive design processes and engage with complexity, innovation, and resilience. This perspective redefines programming not merely as a method of organization but as a catalyst for architectural evolution, fostering meaningful responses to contemporary urban and social challenges. Figure 6 illustrates the relationship between program, programming, and reprogramming in the design process, highlighting their progression from problem definition to structured integration and adaptive possibilities.

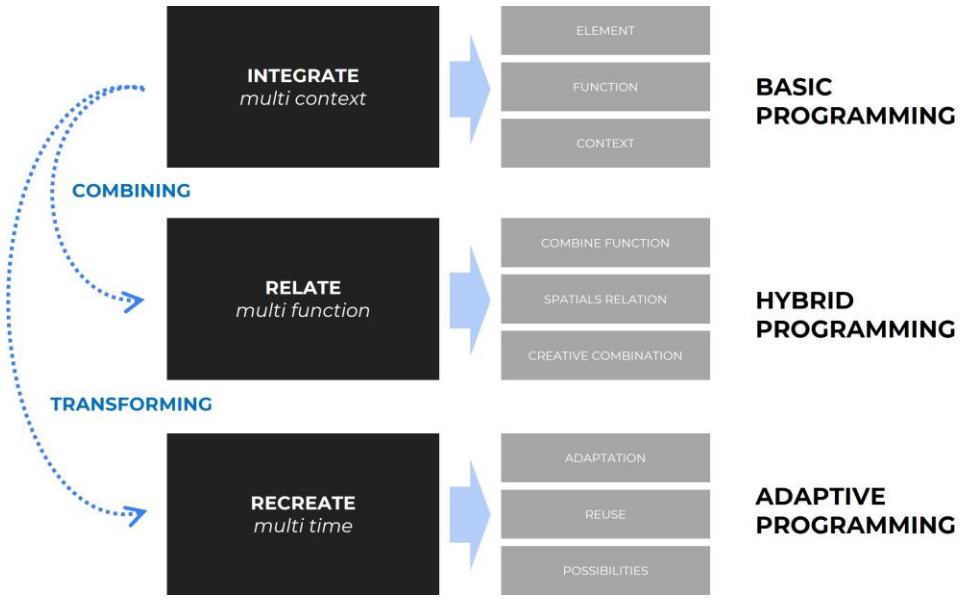

Figure 6. Transformative Dynamic Framework of Programming

CONCLUSIONS

The concepts of program, programming, and reprogramming redefine architectural processes by establishing a dynamic framework to address challenges and opportunities. Within this framework, programming withing multi context integration serves as the foundational layer, defining key elements, functions, and contexts to address specific needs. Combined hybrid program relates and advances this foundation by integrating diverse elements into cohesive and adaptive designs. Recreation through multi time further expands the framework, introducing transformative possibilities by re-evaluating and adapting established structures to respond to evolving contexts and challenges. Together, these three dimensions illustrate the progression of architectural design from foundational problem definition to dynamic integration and innovative possibilities.

However, this framework is not without limitations. Addressing unpredictable variables—such as shifting user needs, environmental constraints, and resource availability—requires navigating complexities and interdisciplinary collaboration. The iterative nature of reprogramming, while transformative, often depends on sufficient time, budget, and technical feasibility, which can present practical constraints. Nonetheless, these challenges highlight areas for further refinement and exploration.

The dynamic framework of integration (multi-context), relation (multi-function), and recreation (multi-time) presented in this study underscores programming's transformative potential as a tool for innovation in architecture. Architects can utilize integration to harmonize diverse spatial and contextual dynamics, relation to blend multiple functions within cohesive designs, and

recreation to adapt and revitalize spaces across time. This triadic approach encourages proactive responses to pressing contemporary challenges, including urban density, sustainability, and technological advancements.

By adopting this framework, architects can move beyond conventional methodologies, fostering adaptable, resilient, and modular solutions. The study ultimately positions programming not merely as a technical process but as a holistic and generative framework. It weaves elements, functions, and contexts into a dynamic and adaptive design process, unlocking new possibilities for architectural innovation. This perspective invites architects to reimagine programming as both a responsive and transformative force, addressing immediate needs while anticipating future opportunities in an ever-evolving world.

REFERENCES

8 House / BIG | ArchDaily (no date). Available at: https://www.archdaily.com/83307/8-house-big (Accessed: 31 March 2025).

Agrest, D. (1977) 'Design versus non-design', *Communications*, 27(1), pp. 79–102. Available at: https://doi.org/10.3406/comm.1977.1410.

Alexander, C. (1964) 'Notes on the Synthesis of Form', in.

Andaloro, Bianca *et al.* (2022) 'Adaptive public spaces'.

Anders, P. (2003) 'Towards Comprehensive Space: A context for the programming/design of cybrids'.

Andreu, I.C. and Oreszczyn, T. (2004) 'Architects need environmental feedback', *Building Research & Information*, 32(4), pp. 313–328. Available at: https://doi.org/10.1080/09613210410001679857.

Barekati, E., Clayton, M.J. and Yan, W. (2015) 'A BIM-Compatible Schema for Architectural Programming Information', in, pp. 311–328. Available at: https://doi.org/10.1007/978-3-662-47386-3_17.

Burry, M. (2011) *Scripting cultures: Architectural design and programming*. John Wiley \& Sons.

Caetano, I., Santos, L. and Leitão, A. (2020) 'Computational design in architecture: Defining parametric, generative, and algorithmic design', *Frontiers of Architectural Research*, 9(2), pp. 287–300. Available at: https://doi.org/10.1016/j.foar.2019.12.008.

Chaillou, S. (2022) *Artificial intelligence and architecture: from research to practice*. Birkhäuser.

Cherry, E. (1998) *Programming for design: From theory to practice*. John Wiley \& Sons.

Cherry, E. and Petronis, J. (2009) 'Architectural programming', *Whole building design guide* [Preprint].

Clark, H. (2009) *Design Studies: A Reader*. Oxford New York: Berg.

Coates, P. (2010) *Programming. architecture*. Routledge.

Creswell, J. (2018) *Qualitative inquiry & research design : choosing among five approaches*. Thousand Oaks, California: SAGE.

Groat, L. and Wang, D. (2013) *Architectural Research Methods*.

Gropius, W. (1965) *The new architecture and the Bauhaus*. London: Faber and Faber.

Hershberger, R. (2015) *Architectural programming and predesign manager*. Routledge.

Jin, S. and Tu, H. (2024) 'Current status and research progress in architectural programming: A comparative analysis between China and other countries', *Frontiers of Architectural Research* [Preprint], (xxxx). Available at: https://doi.org/10.1016/j.foar.2024.07.010.

Jones, J.C. (1992) *Design methods*. John Wiley \& Sons.

Koolhaas, R. *et al.* (2006) '2 Architects 10 Questions on Program', *PRAXIS: Journal of Writing+ Building*, (8), pp. 6–15.

Lucas, R. (2016) *Research methods for architecture*. Hachette UK.

McMorrough, J. (2006) 'Notes on the Adaptive Re-use of Program', *Praxis: Journal of Writing+ Building*, (8), pp. 102–110.

Mountain Dwellings / PLOT = BIG + JDS | ArchDaily (no date). Available at: https://www.archdaily.com/15022/mountain-dwellings-big (Accessed: 31 March 2025).

Oxman, N. (2012) 'Programming matter', *Architectural Design*, 82(2), pp. 88–95.

Plowright, P.D. (2014) *Revealing architectural design: Methods, frameworks and tools*, *Revealing Architectural Design: Methods, Frameworks and Tools*. Available at: https://doi.org/10.4324/9781315852454.

Powers, M. (2007) 'Toward a discipline-dependent scholarship', *Journal of Architectural Education*, 61(1), pp. 15–18. Available at: https://doi.org/10.1111/j.1531-314X.2007.00122.x.

Redyantanu, B.P. (2023) 'Fleksibilitas Dan Adaptabilitas Ruang Domestik Berbasis Arsitektur Keseharian', *Idealog: Ide dan Dialog Desain Indonesia*, 8(1), p. 14.

Robinson, J.W. and Weeks, J.S. (1983) 'Programming as design', *Journal of Architectural Education*, 37(2), pp. 5–11.

Sanoff, H. (2016) *Methods of Architectural Programming (Routledge Revivals)*. Routledge. Available at: https://doi.org/10.4324/9781315541877.

Schön, D.A. (1988) 'Designing: Rules, types and words', *Design Studies*, 9(3), pp. 181–190. Available at: https://doi.org/10.1016/0142-694X(88)90047-6.

Steenson, M.W. (2022) *Architectural intelligence: How designers and architects created the digital landscape*. mit Press.

Summerson, J. (1957) 'The case for a theory of modern architecture', *RIBA Journal*, 64(8), pp. 307–313.

Till, J. (2012) 'Is doing architecture doing research?', *4IAU 4ª Jornadas Internacionales sobre Investigación …*, pp. 1–9.

Urban Rigger / BIG | ArchDaily (no date). Available at: https://www.archdaily.com/796551/urban-rigger-big (Accessed: 31 March 2025).

Verbeke, J. (2013) 'This is research by design', in *Design research in architecture*. Routledge, pp. 137–160.

Vesselov, S. and Davis, T. (2019) *Building Design Systems*. Springer.

Vidler, A. (1996) 'The Third Typology', in *Theorizing A New Agenda for*

Architecture: An Anthology of Architectural Theory 1965-1995. Princeton Architectural Press, p. 258. Available at: https://books.google.co.id/books?id=kXa5xjnHB5QC.

Vidler, A. (2003) 'Toward a theory of the architectural program', *October*, (106), pp. 59–74.

Zevi, B., Gendel, M. and Barry, J.A. (1957) 'Architecture as Space. How to look at Architecture', *Journal of Aesthetics and Art Criticism*, 16(2).

Zhuang, W. *et al.* (2023) 'Research on the frontier trend of architectural programming and post-occupancy evaluation intelligent technology', *SCIENTIA SINICA Technologica*, 53(5), pp. 704–712. Available at: https://doi.org/10.1360/SST-2022-0370.