THE FORMS OF BUREAUCRACY AT CENTRE BEAUBOURG

Ewan Branda[1]
[1]Woodbury University School of Architecture, Los Angeles/Burbank, California

ABSTRACT: The 1970 international competition for the Centre Beaubourg (later renamed the Centre Georges Pompidou) proposed a new cultural institution for the information age: a museum and library conceived as a giant computer. The competition brief represented this enormous cultural information processing system through a program comprising meticulously tabulated requirements, diagrams of spatial relationships, and specifications for all technical aspects of the building's performance. At Beaubourg, rational programming was applied for the first time to an elite cultural building. This paper examines the visual and notational languages of programming used at Beaubourg to control the performance of this information machine and to model the complex exchanges upon which the new institution's metabolic processes were based. Borrowing the conceptual frameworks, rhetoric, and notational conventions from the new discipline of systems engineering, the programming team created novel graphs, topological diagrams, and flow diagrams that formed a new zone for architectural creativity, and in so doing challenges the possibility of a clean bifurcation in the early 1970s between the formal avant garde and an emerging positivist technocracy.

KEYWORDS: Centre Pompidou, architectural programming, technocracy, software studies

Few buildings are as closely associated with the names of their architects as the Centre Pompidou. Although Piano and Rogers were relatively unknown in 1971 when they were announced as the winners of the international design competition, they quickly rose to fame through the countless articles, interviews, and media appearances in which they explained and defended the strange object slowly rising over the Plateau Beaubourg. Today, their names are synonymous with that building, while the building, more than almost any other of its day, stands as testimony to its authors' clarity of vision and their heroic commitment to its realization. They were the first starchitects of the media age.

Shortly before the building's completion, however, the historian and critic Reyner Banham pointed out that the building's success as a realized vision was due not only to the determination of its architects and engineers but also to the "less public but far longer sustained managerial determination of its programmaticien (so much more than a mere 'manager'!) François Lombard" (Banham 1976, 211). Who was this technician who worked behind the scenes to produce one of the 20th century's most recognizable and notorious buildings? Initially, hired to write the brief for the 1970 competition, Lombard ran the Programmation et études team that oversaw all of the project's functional requirements from conception through completion. Relentlessly demanding of the architects and wielding formidable administrative techniques powered by the new systems thinking, Programmation was the invisible co-author of the building we see today. This paper examines its methods. In particular, it considers the languages of form inherent in its managerial techniques, arguing that they signal an attitude to design that denies the false opposition between performance and form.

THE BRIEF

Any architect who entered the 1970 Centre Beaubourg competition would have been surprised by the sheer weight of the package arriving in the mail. The competition brief presented a detailed program with meticulously tabulated requirements, diagrams of spatial relationships, and specifications for all technical aspects of the building's performance, from lighting and acoustics to computers and networks. In contrast, the brief for the Sydney Opera House competition, launched fifteen years earlier, offered mainly site photographs and competition regulations, with the building program sketched out in two short pages in the Appendix. While the differences between the Beaubourg brief and its predecessors unquestionably reflected the technocratic methods of its authors, they also point to a more general disciplinary shift in which systems methods such as programming were added to the architect's professional toolkit for dealing with increasingly complex buildings. It was not until 1959, four years after the Sydney competition, that Peña and Caudill's seminal article on programming appeared in Architectural Record (Peña and Caudill, 1959). In 1966, an AIA report announced that "one of the scientific techniques for problem solving appearing on the horizon is that of systems analysis or systems development" (Wheeler 1966), and by the end of the decade, systems were everywhere, and one could not visit an architecture school or art museum without being reminded of the fact.

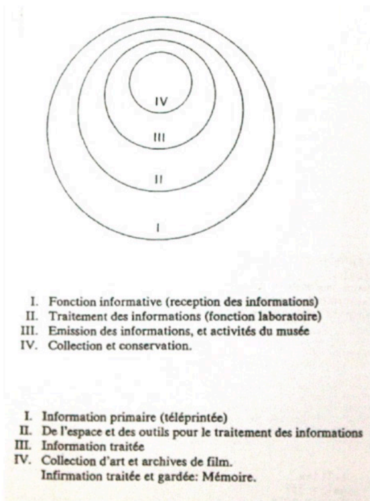
If Programmation believed such methods were the only hope of realizing this complex project and meeting its unreasonable deadline, Piano and Rogers held an equally firm suspicion of those methods, particularly when they were wielded by a state bureaucracy. Despite the ubiquity of systems discourse, the use of techniques normally applied to anonymous building types such as hospitals, mass housing, and schools in the design of an elite cultural building was entirely new. To flaunt them in a high-profile international design competition bordered on heresy. The brief overreached, and its “highly formalized, super rationalized” approach, as Rogers put it, flew in the face of the humanist, emancipatory, even libertarian, values latent in the early years of his practice (Rawstorne, 408). The brief raised another more insidious and troubling problem. Such an overbearing managerial approach surely warned competition entrants of potential future difficulties in its tacit challenge to the individual agency of the architect. Architectural postmodernism was already sensitive to the incursions of methods from the social sciences, and even Piano and Rogers, who unlike their neo-avant-garde colleagues were committed to a democratic and pragmatically-minded architecture, valued the architect’s hallowed position as chief organizer.

After a week of deliberations, they decided to enter the competition. Reflecting back on the project, they considered the collaboration with Lombard’s team productive. Not everyone agreed. Shortly after the opening in 1977, Rogers’s mentor Alan Colquhoun wrote a critical review that revived many of the architects’ early concerns. Colquhoun attacked client, programmer, and architect alike for complicity in an insincere rationalism. He argued that technocracy and its sublimation in the radically open plan and the trope of flexibility demonstrated an abdication of the architect’s fundamental responsibility — the articulation of content through a language of form, as exemplified, say, in the Beaux-arts plan. Even earlier functionalist architects, Colquhoun argued, were “less concerned with creating a rational architecture than they were with creating the symbolism for a new social and cultural order” (Colquhoun 1977). What Colquhoun could not see was that this rationalist technocracy offered a new type of architectural creativity, with techniques drawn from a range of outside influences such as computation and systems engineering, whose attitudes to architectural form suggested ways the discipline might navigate the unknown territory of the 1970s. In other words, the open plan did not eliminate the logic of the Beaux-arts plan but merely demonstrated its transposition to new practices.

Programmation embodied French technocracy. Lombard had studied engineering at the École Centrale des Arts et Manufactures in Paris, and after finishing graduate studies in engineering at Berkeley returned to work for the Architecture division of the Ministry of Cultural Affairs. There, he applied new techniques of programming to the planning of cultural amenities for De Gaulle’s Villes Nouvelles on the outskirts of Paris. Pompidou decided to run the project directly out of his offices, bypassing the Ministry under whose responsibilities a cultural building of this scope would normally fall. Lombard left to join the Établissement publique, the autonomous public agency created to execute the project, and was charged with writing the brief and running the Programmation et études group. To start, Lombard recruited Patrick O’Byrne from a small engineering firm in Montréal whose research into the programming and fabrication of school buildings was funded by the Ford Foundation’s influential Educational Facilities Laboratory (EFL). Where Lombard had experience with programming at the Ministry of Cultural Affairs, O’Byrne brought experience in what in EFL circles was known as the Performance Concept, in which programming was part of a broader performance-based approach to industrialized building.

The somewhat clandestine nature of Programmation’s operation meant that few funds were available from the Ministry, and so the team worked improvisationally in somewhat impoverished conditions. Despite this, the competition program was detailed beyond anyone’s expectations. In late 1970 the funds for the competition were released, the brief adopted by presidential decree, and the competition announced that November. The brief was much more than a specification for a library or a museum. Reading it carefully it becomes clear that the myriad activities whose performance is described in such detail constituted a meta-program for a new institutional type—a meticulously engineered information machine in which visitors no longer went to a museum to merely view artworks nor visited a library simply to read books but rather were users of a more general apparatus of cultural exchange, education, self-improvement, and discovery powered by the flow of information. In short, the brief described a cultural center conceived as a giant information processing system. This was no mere metaphor. Earlier that year, three members of Pompidou’s client team participated in a series of Unesco workshops where they developed a conceptual diagram of museum conceived as an information system consisting of four concentric rings corresponding to four different levels of information processing (Fig. 1).

In the brief’s main diagram, the concentric rings of the Unesco diagram exploded into a constellation of activities linked by interfaces and interactions (Fig. 2). This schema was provocatively non-hierarchical: one would find the relatively tiny documentation and research center, or even parking, given as much attention as the library or museum, which the brief shows not central as one would expect but pushed to the margins. In their place at the heart of the building the diagram showed a network of minor activities such as temporary exhibitions, documentation and research, reception, and a range of new experimental galleries and resources. These activities were the interfaces between public and the flow of objects and documents, and between public and the circulation of experts, interpreters, and reference workers. Within this environment, links were more important than the elements they connected. Television monitors, satellite broadcasting, computer terminals, signage, and circulation systems were the adhesive. As with all information utopias—from Le Corbusier’s Mundaneum to Malraux’s Museum Without Walls—more was more, and so functional efficiency was made secondary to the proliferation of information: visitors were to pass through the public reception spaces not simply on their way in and out of the building but as often as possible and in so doing continually and accidentally encounter and produce constantly renewing information. The brief thus asked not for the mitigation of complexity but rather for its exploitation and the amplification of its effects.



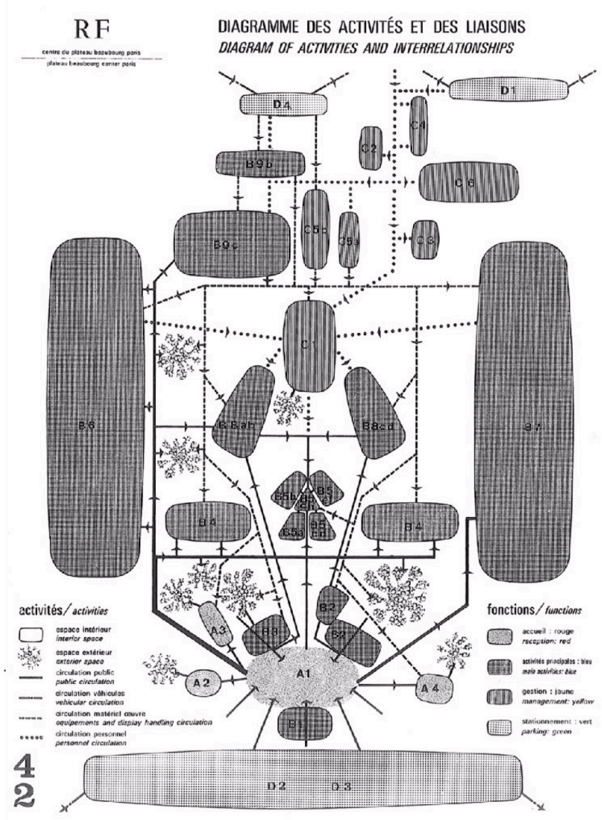


Figure 1: Unesco diagram: Museum as Information System. Source: (Museum 1970)
Figure 2: Competition brief program diagram. Source: (Competition Brief 1970)

THE PROGRAMME SPECIFIQUE

Once the brief was completed and distributed, Programmation immediately started work on the Programme Spécifique, working in parallel with the competing architects as they developed their competition entries. By the autumn of 1971, when Piano and Rogers started work on the definitive scheme, Lombard's team already had produced two large volumes of detailed diagrams and tables that made the competition brief look schematic and rudimentary in comparison. The brief was by definition a starting point for design work. In contrast, the Programme Spécifique can be seen as an architectural work in itself in that, like the competition entries, it responded the brief by giving form to its speculative visions.

It did so through innovative uses of graphic representation. Early programming researchers had established the centrality of flow charts and spatial relations diagrams in the pre-design toolkit. For user-needs specialists and users alike, the diagram was democratizing: it played a central role, both rhetorical and practical, in bridging the needs of non-expert users and the abstractions of architectural representation. The canonical early sources on programming, however, show little commitment to representational innovation. Diagrams were often limited to aphoristic commentary on design preconditions or the design process itself. In some cases, diagrams merely restated tabular data. For most researchers, the diagram was simply one of a set of practical tools available for pre-design work. Horowitz, for example, mentions only briefly the graphic diagram as one of three options for describing spatial requirements, the other two being verbal description and a two-dimensional matrix of spaces (Horowitz 1967). Early programming researchers introduced the main formalism upon which these diagrams were based. The spatial adjacency diagram was a variant of the network graph in which spaces were represented by nodes scaled to show required floor area, with edges scaled to represent desired distances between spaces (Fig. 3). This type of diagram merged the geometric logic of the architectural plan with the more spatially abstract topological graph, which had been introduced into architectural discourse by Alexander in 1964 and was commonplace by the early 1970s.

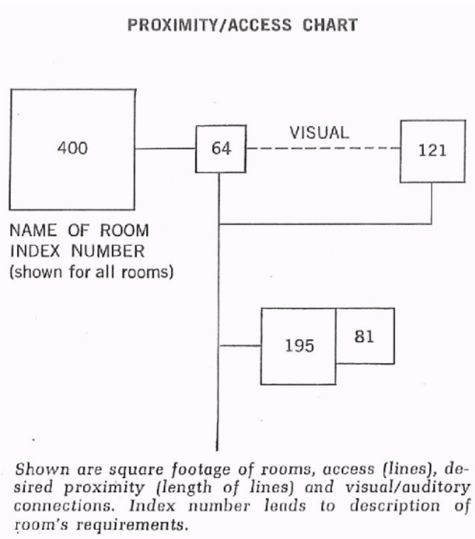
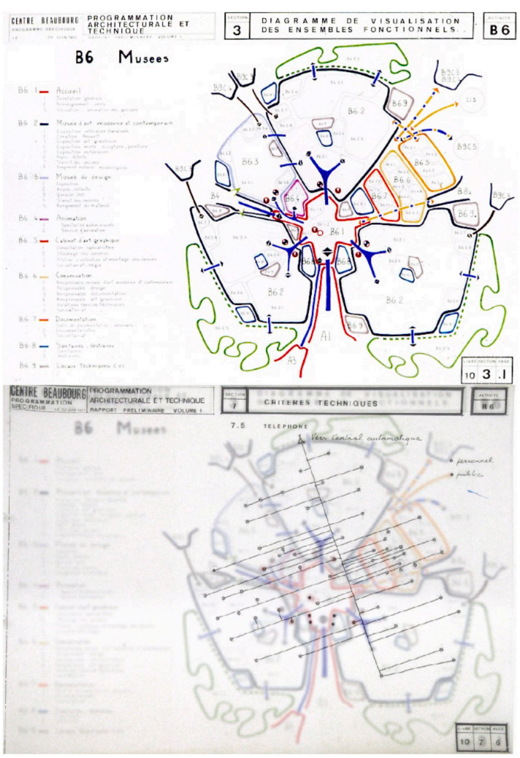


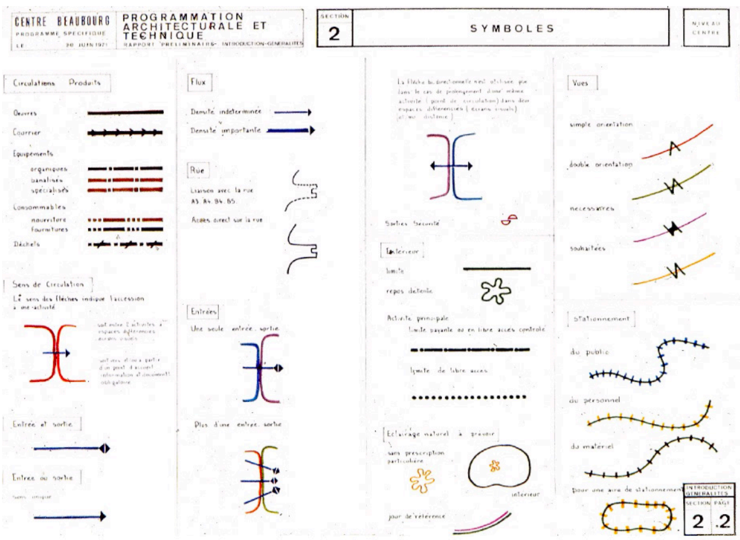
Figure 3: Spatial relationship diagram with weighted nodes and edges. Source: (Horowitz 1967, 96)

The Programme Spécifique exuberantly explored the space between the hard-headed spatial relationship diagram and speculative architectural representation. Programmation launched a two-pronged graphic attack on the problem. On the one hand, a spatial approach addressed the relationships between functional elements through an expanded syntax of network graphs and spatial-relation diagrams. On the other, a temporal analysis examined the flows of objects and information through those same functional elements. In the former, Lombard and his team transformed the basic bubble diagram of the competition brief into a graphic language with far greater representational capabilities. For each functional category, the program dedicated a single page showing the principle functions and the qualities and attributes of interactions between them—desirable and undesirable views, sound isolation, interfaces between people, goods, documents, and relationships to outdoor spaces (Fig. 4). Homologous systems such as telephones and computers were shown using overlays drawn on translucent vellum (Fig. 5).



Figures 4-5: Spatial relation diagram and overlay, Programme spécifique. Source: (Archives CFP, 2007)

The notations shown in the bubble diagrams were based on a predefined taxonomy of material and immaterial attributes of interfaces (Fig. 6). This taxonomy represented the invisible protocols by which administrative or social behavior coagulated, as well as the reification of those protocols through visual, acoustic, and physical boundaries of varying degrees of permeability. Boundaries were classified by the type of agents they mediated (people and machines, people and artworks, documents and archives), the specific quality or parameter being controlled or exchanged (natural light, views, access privileges), and the degree to which the interaction was required or desired, and the degree of allowed overlap (adjacent, partially overlapping, embedded). Potential movement across each boundary was marked with attributes such as the reciprocity of exchange (non-directional, unidirectional, bidirectional), the actors and objects involved (people, information, artworks, food, cars, natural light, views), and various other criteria (unrestricted, restricted, high-density, low-density).



Figures 6: Taxonomy of symbols, Programme spécifique. Source: (Archives CFP, 2007

In what ways did these diagrams prefigure the form of the building? Researchers at the time had struggled with the complex relationship between the topological graph and the architectural plan (Cousin 1970; March and Steadman 1971). Where the graph showed an edge linking two nodes, the plan showed the common limit surface between the two spaces running 90 degrees to that edge (Fig. 7). Although the graph at first appeared planimetric, it differed from the plan in that it described the quality and attributes of the spatial boundary as it is transgressed, not the geometry of the boundary itself. In other words, the graph might describe the qualities of the wall's performance in separating two spaces rather than the wall's qualities as an object as such. Thus, in the Programmation diagrams, the curvilinear shape of the boundary surfaces was intended merely as a tactic of evasion to discourage the architects from reading them as plans per se. The precise shapes themselves had an indeterminate effect on architectural form, since both rectangular shapes and curvilinear shapes do not affect the logic of the network graph's topology. More important was the quality of the boundary as a protocol of exchange.

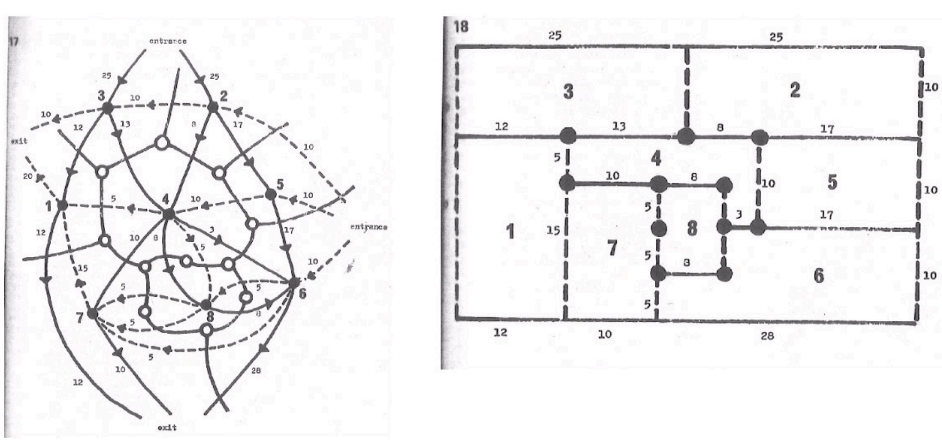


Figure 7: Topological graph versus floor plan. Source: (Cousin 1970, 493)

As topological models, then, the bubble diagrams are not meant to be read as plans. For example, in a given diagram, two lobes making up one activity might very well represent distinct spaces on different floors in the final building, in which case the arrows connecting them would correspond not to doors or corridors but rather to vertical circulation. Nor do the curved boundary lines necessarily represent walls. An interface between storage and museum gallery, for example, might in the most straightforward case be reified as a wall, but it also might be plausibly translated, as it was in the final building, into a jukebox-like machine by which visitors could call up a picture on demand from an overhead storage system (Fig. 8).

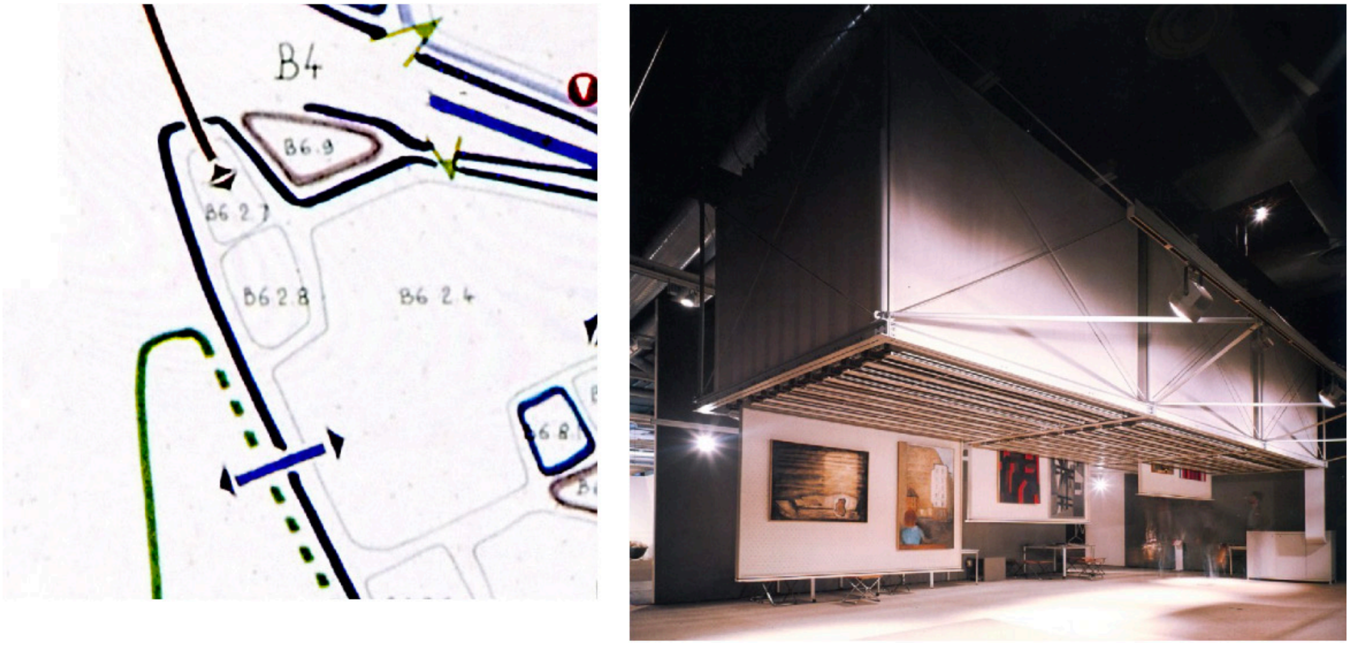


Figure 8: Kinakotheque system for viewing and storage of paintings (B6.2.8 in plan). Source: (Archives CFP, 2007)

One of Programmation's main doctrines stated that a solution to a functional problem might be found in any combination of spatial organization, personnel, furniture, and equipment. In this way, the tendency for these bubble diagrams to communicate a parti, in the traditional typological sense, requires us to understand that, as functional types, they might be instantiated through any combination of walls, floors, materials, machines, and workflows. Thus, a diagram showing the library as having two lobes organized around a central spine, or showing the museum as having three lobes organized around central core, represents a type that is both formal and performative.

The second type of diagram in the Programme Spécifique concerned the flows of people, objects, and messages through the various functional units (Fig. 9). A detailed examination of these diagrams is beyond the scope of this paper. Briefly, these diagrams are temporal where the bubble diagrams are spatial. They comprehensively track the flow of visitors, artworks, books and journals, equipment, mail, even garbage. Here, everything is rendered equivalent as it is subject to the same analytical and organizational regime.

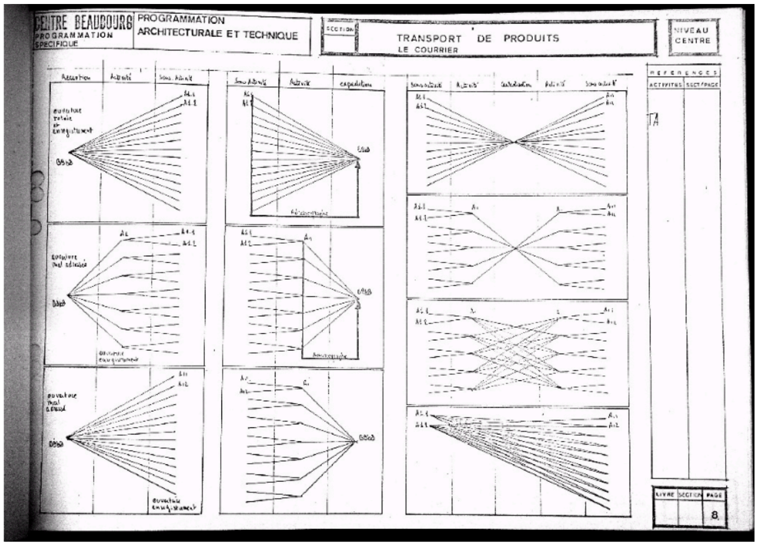
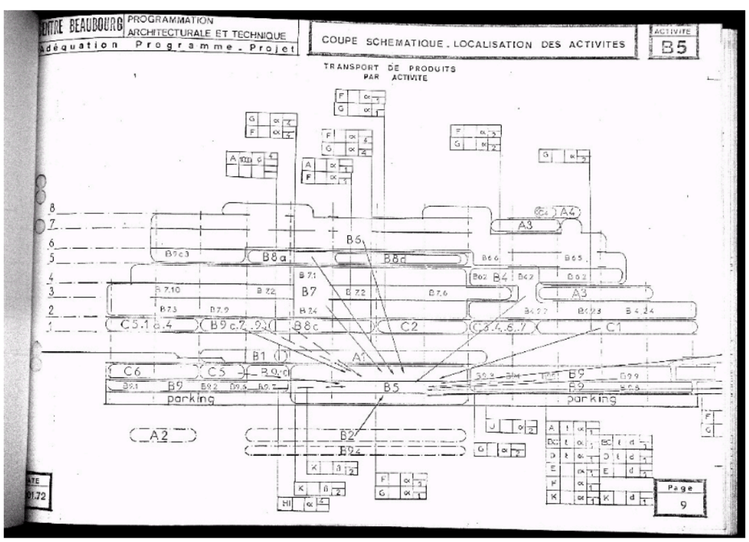


Figure 9: Flow diagrams, Programme spécifique. Source: (Archives CFP, 2007)

As Piano and Rogers developed the winning competition entry into the definitive scheme, the complex relationship between program diagram and architectural form played out in a meticulous process of matching requirements to design. A third set of diagrams notated this process. In the sections, we fully understand the range possible modes of reification of the abstract bubble diagrams (Fig. 10). Here, no deterministic relationship between functional activity and architectural form governs the end result. Instead, activities are fitted iteratively into the generic frame as if solving a puzzle, and the resulting organization clearly breaks any causal relationship between architectural form and functional signification since activities are straightforwardly accommodated following a logic of pure opportunism and optimization.



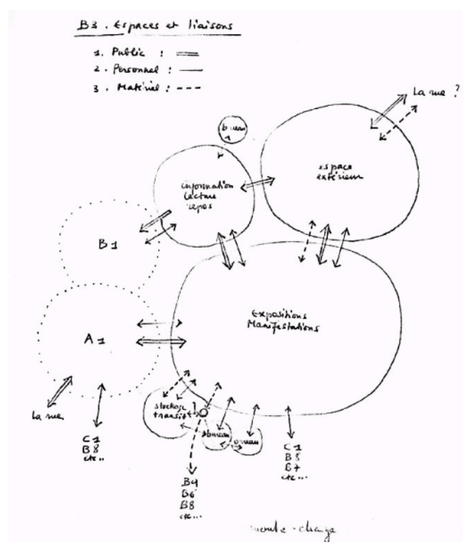


Figure 10: Space allocation sections, Adéquation Programme-Projet. Source: (Archives CFP, 2007)
Figure 11: Needs diagram, by Germain Viatte, head of the department of contemporary art. Source: (Archives CFP, 2007)

The architectural approach of Piano and Rogers made this relatively easy. With Team 4 and Norman Foster, Rogers had advanced what Bryan Appleyard described as "an increasingly available, serviceable and reticent architecture" (Appleyard 1986, 159). This ethos was sympathetic to Lombard's intention that the program be a living document that doesn't merely operate at the start of a project but overlaps with the entirety of the building's design, construction, delivery, and occupancy. Earlier programmers emphasized pre-design: Peña and Caudill called the program a "prelude" to design, and Horowitz described the program as something handed over to the architect at the end of analysis (Peña and Caudill 1953; Horowitz 1967). In contrast, Programmation insisted on an ongoing working relationship with the architects throughout the life of the implementation, and with the users after the building was handed over. The Programme Spécifique had already demonstrated that its graphic language could serve as a visual lingua franca in discussions between programmer and user in the discovery of new requirements. Heads of departments, from curators of contemporary art to managers of loading docks, were encouraged to use it to document their needs (Fig. 8). Its conventions even served as a graphic tool in the rubric for evaluating the submitted competition entries. But shortly before the building's opening, Lombard wrote to his superiors asking for continued involvement after building operations started. "Putting a building into service is not the end of programming," he implored, and complained that the only place one could find the Programme Spécifique was on the shelves of the library's archives (Lombard 1976).

CONCLUSION

Among the risks of any pictorial representation in systems-based design practice was that it encouraged the use of prior forms in problem solving. Indeed, this concern was not restricted to graphical representation. Even in the program text, Lombard rejected such seemingly innocuous terms as "library," "civic center," or even "city" lest they impose prejudice upon the discovery of an innovative architectural outcome. In much the same way, the topological graphic approach rejected appeals to both intuition and geometric types. Despite the deliberate naïveté of the program's visual language, its diagrams thus shared with computable mathematical graphs the potential of revealing unforeseen solutions—either latent or synthesized anew—by freeing architectural problem-solving from the grip of geometric thinking and prior form.

It was precisely this rejection of prior forms in architectural problem solving that had bothered Alan Colquhoun when he wrote his critical review of the building after it opened. In his essay "Typology and Design Method" of a decade earlier, Colquhoun attacked the new rational design methods for their assumption that it was desirable, or even possible, to sweep away all preconceptions, biases, and prior solutions in the search for solutions to complex problems. He pointed out that if one looked at examples of systems design in particularly complex domains such as aeronautics engineering one immediately saw that prior forms offered the only way to mitigate intractably complex problems. Colquhoun acknowledged that "[t]he characteristic of our age is change," but went on to argue that "it is precisely because this is so that it is necessary to investigate the part which modifications of type-solutions play in relation to problems and solutions which are without precedent in any received tradition" (Colquhoun 1969).

Colquhoun may have been right. The 1970s saw the birth of software engineering as a discipline, and as computers became more powerful and their uses more wide-ranging, the complexity curve of its design problems quickly steepened. In response, engineers proposed a formal object-oriented method to replace the improvisational approaches on which these practices had until then survived. By the end of the decade, the shift from procedural thinking to object-oriented thinking in the design of software systems had taken hold, fully arriving in the 1980s with design patterns, borrowed directly from Christopher Alexander's Pattern Language, a work that in itself signaled the architect's own object-oriented turn. With design patterns, the engineer started with the identification of prior models, arrangements, and behaviors. In France, methods such as MERISE and RACINE offered a particularly managerial vision of object modeling for system design. In the evolving domains of systems design, complexity had reached a point where, as Colquhoun had anticipated back in 1969, the only reasonable approach to its mitigation was to turn to the collective wisdom of prior solutions expressed through articulate formalisms.

What Colquhoun later failed to see when he wrote his critical review of the Pompidou was that architecture's will-to-form, which he accused Piano and Rogers of denying, had migrated to the programmer along with its implicit object-oriented thinking, while the architectural reductivism he criticized in the building constituted an equally willful architectural response to that migration. If the original competition entry showed, in good 1960s fashion, a mechanistic and largely symbolic apparatus of clip-on devices and moveable floors, the final building offered a precisely delimited monolith with no moving parts, a "smooth monument," as its engineer Ted Happold called it (Happold 1977). The famous escalator is most indicative of this shift: in early versions of the project it appears as an articulated clip-on mechanism, while in later ones it tautly traverses the west façade in a continuous, abstract diagonal that would later inspire Jean Widmer's design for the institution's logo. It would be a mistake to attribute this formal trajectory exclusively to value engineering. Instead, it must be seen as a wordless denunciation of the imagistic approaches of earlier technological utopians like Friedman, Archigram, and the Metabolists who sought direct formal correspondences between architecture and new information technologies. In the resulting empty space, Programmation embraced architecture's form-giving practices and their fascination with pattern, symmetries, typology, and even the pure pleasure of the drawing.

REFERENCES

Appleyard, Bryan. 1986. *Richard Rogers: A Biography.* London: Faber.

Banham, Reyner. 1976. *Megastructure*. New York: Harper and Row.

Colquhoun, Alan. 1969. "Typology and Design Method." *Perspecta* 12: 71–74.

Colquhoun, Alan. 1977. “Critique.” *Architectural Design* 77 (2): unpaginated.

Concours International D’idées À Un Degré (*Competition Brief*). 1970. Paris: France, Ministère d’état chargé des affaires culturelles.

Cousin, Jean. 1970. “Topological Organization of Architectural Space.” *Architectural Design*, October, 491–93.

Happold, Ted. 1977. “Beaubourg: Architecture or Engineering.” *Architectural Design* 77 (2): unpaginated.

Horowitz, Harold. 1967. “The Program’s the Thing.” *AIA Journal*, May, 94–100.

Lombard, François. 1976. “Ultime reflexion d’un programmateur.” Archives CGP, 2005100/6.

March, Lionel, and Philip Steadman. 1971. *The Geometry of Environment.* London: RIBA Publications.

Peña, William M., and William W. Caudill. 1959. “Architectural Analysis: Prelude to Good Design.” *Architectural Record*, May 1959.

Rawstorne, Peter. “Piano & Rogers: Centre Beaubourg.” *Architectural Design* 42, July, 407-410.

Wheeler, Clarence Herbert. 1966. *Emerging Techniques of Architectural Practice.* American Institute of Architects.