BARCODE HOUSING SYSTEM
Integrating floor plan layout generation processes within an open and collaborative system to design and build customized housing

LEANDRO MADRAZO, ÁLVARO SICILIA, MAR GONZÁLEZ, ANGEL MARTIN COJO
ARC Enginyeria i Arquitectura La Salle
Universitat Ramon Llull, Barcelona, Spain

ABSTRACT: The goal of the project has been to design and implement an ICT environment which facilitates the interaction of the different actors (architects, builders, manufacturers, occupants) involved in the design, construction and use of affordable housing built with industrialized methods. The interwoven working environments which form the structure of the system enable the actors to carry out their activities in a synchronous and asynchronous manner. As well as providing a structure that supports collaboration, the system automatically generates housing units and buildings.

KEYWORDS: Design thinking, knowledge based design, project management, collaboration and communication

RÉSUMÉ : L’objectif de ce projet a été de concevoir et d’implémenter un environnement de TIC qui facilite les interactions des différents acteurs (architectes, constructeurs, manufacturiers, usagers) impliqués dans la conception, la construction et l’usage des logements abordables construits par des techniques industrialisées. Les environnements de travail entrelacés qui forment la structure du système permettent aux acteurs de réaliser leurs activités de manière synchrone ou asynchrone. Le système assure une structure de support à la collaboration et génère automatiquement des unités de logements et des bâtiments.

MOTS-CLÉS : Réflexion en conception, conception basée sur les connaissances, gestion du projet, collaboration et communication

T. Tidafi and T. Dorta (eds)
Joining Languages, Cultures and Visions: CAADFutures 2009
© PUM, 2009
1. INTRODUCTION
The use of information and communication technologies has become as a decisive transforming factor in the housing industry, by providing computerized environments that support the design and construction of housing suited to the social, economic and technological needs of our time. Integrated, modular environments distributed throughout Internet could contribute to creating more effective organizational structures which help to overcome the insufficiencies caused by the perennial fragmentation of the building and construction industry (Brown et al. 1996). Such environments could facilitate the emergence and consolidation of novel forms of collaboration made by the different actors (architects, engineers, clients, contractors, and project managers) participating in the life cycle of a building, from design to occupation and even demolition.

The purpose of this research work has been the creation of an Internet-based environment which facilitates the interaction of the different actors (architects, builders, manufacturers, occupants, facilities managers) involved in the design, construction and use of affordable housing built with industrialized methods. This project began in 2002, when we started to develop a rule-based system to automatically generate housing units and buildings resulting from their aggregation, by means of user-friendly interfaces. This system, which was completed in 2005, was stand-alone and limited to single users. We then proceeded to develop a new environment to overcome these limitations, facilitating the participation and interaction of different agents in the process of designing, building and using the housing generated with the support of the system (Figure 1). It is a modular environment which supports the decision-making process in an open and distributed fashion, allowing inputs from different users at any stage of the process.

FIGURE 1. ACTORS AND ACTIONS TAKING PLACE IN THE ENVIRONMENT.
2. WORKING SPACES

Barcode housing system consists of interwoven working spaces in which the different actors (architects, developers, manufacturers, occupants) can participate in a synchronous and asynchronous manner throughout the whole process of design and construction of housing units and of the buildings resulting from their aggregations.

The work spaces and their functionalities are the following:

- **Project development.** In this working space, developers, architects and building managers specify site properties (area, size), number and type of housing units, building and planning regulations (building volumes and height), and environmental conditions (climate, orientation). Alternative solutions of buildings (massing, location) can then be explored for the given site conditions and brief.

- **Housing unit layouts.** In this space, architects select a set of units that will be later used in the generation of a building. The units have been generated by the system in batch processing, and are stored in the system database. The selection becomes a “discovery” process as the architect finds the housing layouts while navigating through the space of solutions which the system has generated in previous project developments. In the case of an adequate layout not being found from the pool of the existing solutions, the architect can request the generative system to create alternative layouts that conform to the desired criteria (surface, number of rooms, number of bathrooms, an open or closed kitchen, etc.). The new solutions are stored in the database, thus enhancing the previously existing pool of solutions.

- **Housing unit configuration.** Occupants describe their housing program (family members, usage of spaces, lifestyles…) working with user-friendly interfaces that represent housing units and layout in a graphic language that can be understood by lay people (schematic plans, photographs depicting activities in spaces, bubble diagrams). The system returns the housing units which best most closely correspond to the criteria defined by the users and they select those that most adequately meet their needs. Then, the selected units are used in the generative process that creates the housing block. Once the housing units have been assembled, there is a process by which occupants and architects collaborate using a 3D environment to define the arrangement of a living unit (finishing, partitions, and furniture).

- **Housing units assembly.** In this environment, the architect defines the design criteria for the assembly of living units, for instance: degree of compactness of the housing block; degree of optimization of building services; minimum distances to access cores (staircases, elevators); material of the structural skeleton, and so on. Once the design values are set, a generative process creates the solutions that satisfy these criteria.

- **Building components catalogue.** A XML-based product modeling catalogue enables manufacturers to introduce descriptions of their products which will
be then selected by the team in charge of the project development. Based on this selection, the future occupant chooses the components (doors, windows, partitions) and inserts them in the 3D representation of a dwelling.

It is not necessary to proceed along the working spaces in a linear fashion in order to interact with the system. Rather, the only prerequisite is to start the Project Development workspace, where the characteristics of the program are described (brief, number of dwellings, site) and users are registered. All other spaces can be activated later at any moment during the project lifecycle. For instance, a project development could start in a traditional way selecting a site and specifying an architectural brief to make a building design. But it could also start in other ways. A manufacturer could input its building products into the system so that an architect can then search for suitable layouts to design a building with those components. In this way, the system can support non-linear design and construction processes by exploiting the potential of ICT technologies to transform established design and construction processes (Kalay 2004).

Describing all the workspaces and their corresponding activities is beyond the scope of this paper. Therefore, in the following sections we will focus on explaining only the processes that are part of the Housing unit layout working space.

3. GENERATING VERSUS SEARCHING

Automatic generation of floor plan layouts is amongst the earliest applications of computers to architectural design. Developments of early applications started at the end of the 1960s, and the underlying issues regarding spatial representation, generative procedures, and spatial allocation techniques were identified and summarized more than thirty years ago (Mitchell 1977).

As Steadman observed, in automatic floor-plan generation two basic approaches exist: to generate one or few plans that satisfy a set of specified constraints, or to produce all the possible plans which cover all the requirements (Steadman 1983). In this system, we have adopted the second approach: housing layouts are produced in a batch processing and stored in the system’s database. This way, we avoid the high computational cost of generating a possible solution, thus speeding up the search process. The architect will then start searching for housing units that fit the program requirements of a project development. If the adequate designs are not found, a generative process can be triggered – the same application that populates the database with housing designs in batch mode– to create new designs.

Nowadays, the possibility to carry out design processes on the Internet provides an opportunity for users to participate in the design of mass customized housing (Chien and Shih 2000; Gerzso 2001; Huang and Krawczyk 2007). This requires the design of environments which support the interaction with
users by means of appropriate interfaces while, at the same time, take advantage of the capacity of computer programs to generate and evaluate design solutions.

4. SPATIAL STRUCTURE OF THE HOUSING UNITS

More often than not, applications developed to apply computer technology to housing design, rely on an existing building systems (prefabricated, modular) from which assembly rules are derived and implemented in a computer program. In this way, as Cross (1976) pointed out, adhering to a building system helped to constrain the infinite freedom of traditional building, while also making the design manageable to the machine (Benrós et al 2007). In other cases, design rules have been derived from an existing body of architectural examples, so that a computer program generates designs based on existing ones using shape grammars (Koning and Eizenberg 1981; Duarte 2003) and production rules (Gerzso 2000).

In our case, we have simultaneously developed both an architectural system and a computer system. Housing units are based on a spatial structure made of horizontal stripes placed over a modular grid (Figure 2). A similar division of floor plans in stripes was already considered by Habraken in his seminal work (Habraken 1972). Vertically, the floor space is divided into bars which also adhere to the module of the grid. At the intersection of a horizontal stripe and a vertical bar lies a cell, the minimum spatial unit with an associated function. Some of the attributes of a cell (function, maximum and minimum dimensions) are determined by its location within the structure of horizontal stripes and vertical bars. For example, cells located in zone 2 have the function of living room or kitchen, and their depth is proportional to the depth of the housing unit.

Cells can be of two kinds: server (those that contain a building service, like kitchen or bathroom, and/or a structural component) which are located on the Z1, Z3, and Z5 areas, and served (those without any servitude) placed on the Z2 and Z4 areas. Rooms are the result of joining the cells belonging to a zone. A group of cells makes a room with a specific function and appropriate dimensions (bathroom, kitchen, bedrooms).

The vertical dimensions of the units can vary between 8 and 12 m. to enable the creation of housing blocks which can fit in different sites and comply with building regulations (Figure 3). For the smaller (8-meter high) housing unit, it is possible to do away the central area. Maximum and minimum dimensions are established for each of the stripes. For instance, the intermediate areas Z2 and Z4 can vary from 2.4 to 3.6 meters. These thresholds guarantee that rooms placed in them will have the adequate dimensions.
Housing units created by the BARCODE system are characterized by different kinds of flexibility: multiple layouts can be created for a given area (flexibility in creation); a given housing layout can be transformed by changing the partitions (flexibility in usage); and a built flat can expand or reduce its area by adding or giving away vertical bands (flexibility in transformation). The three kinds of flexibility are interwoven and rely on the same principle of spatial ordering based on horizontal stripes and vertical bars. In this way, each housing unit can be configured differently during the design process and can adapt itself to the different necessities of each user on its completion.

5. GENERATIVE PROCESS OF HOUSING UNITS

A graph represents the spatial structure of a housing unit in terms of relations between cells (Figure 4). A cell is defined by three attributes: privacy (public
or private), size (depending on the adopted module) and type (bath, kitchen, entrance, corridor, balcony, sleeping room, and living room). The edges connecting the cells are labeled “accessibility” and “visibility”. Accessibility indicates whether two adjacent cells can be accessed from one to another. Visibility specifies whether cells are separated by an opaque (wall, door), transparent (glass, empty pass) or semitransparent (shelve) division. Thus, a relationship of one cell to another can be characterized by a combination of these two values. For example, a relation having accessibility 0% and visibility 0% could represent a wall or a dense shelf.

**Figure 4. Graph representing a housing unit.**

Creating a valid housing unit from the graph has a huge computational cost. To minimize this expense, constraints to validate the solutions have been implemented outside the graph. These constraints have been modeled after a tree-data structure and defined as combination of spatial attributes and Boolean operators. For example, the type of entrance to the flat—through a walkway or though an entrance hall— is specified in a branch of the constraint tree in this way: 1 Z1_init WITH (LEFT (null XOR Z1_corridor) OR RIGHT (null XOR Z1_corridor) and 0 Z3_init) XOR (1 Z3_init and 0 Z1_init), where 1 Z1_init stands for the entry cell in zone one, null represents the boundary of the unit, Z1_corridor stands for the walkway, and Z3_init is the entry cell in zone three. Other characteristics of the living units are encapsulated in a list containing the rules applied to cell grouping to make rooms: minimum and maximum square meters and aspect ratio (proportions between length and width).

By separating topological conditions—represented in the graph—from functional and dimensional requirements—represented in the constraint tree and in the list of aggregation rules—the generative process offers a high degree of flexibility at a minimum computational cost (Figure 5). Furthermore, because constraints and aggregation rules are isolated from the graph they can be more
easily edited. This way, different space solutions can be explored without having to modify the graph itself.

**FIGURE 5. INPUT-OUTPUT OF THE GENERATIVE PROCESS.**

The process to generate floor-plans consists of four stages, each one operating on a representation with a different level of detail, starting with the dimensionless plan and concluding with scaled housing layouts (Figure 6). This way it is possible to capture the evolutionary nature of a design process, where each decision depends on those taken in a previous stage (Ruch 1978).

**FIGURE 6. THE FOUR STAGES OF THE GENERATIVE PROCESS OF A LIVING UNIT.**

### 5.1. Dimensionless phase

All possible and valid combinations of cells within the graph are found at this phase. Valid housing layouts are those that fulfill all the constraints on the Boolean-tree. The algorithm uses backtracking techniques to join the nodes of the graph in order to create a dimensionless layout. To speed up execution, the process uses the constrain tree to prune the search space’s branches that have invalid partial solutions. To ensure the reliability of the process, we have used iterative techniques instead of the recursive ones that could cause stack overflow problems. Since it is a batch process, the execution can be stopped at any moment, storing the status of the work done to resume it later on.
Solutions generated in this phase have the lowest level of definition: only the space allocation—topological information—is contemplated. The dimensionless solutions generated in this stage are stored in a database because the different phases of the generative process are executed asynchronously and, therefore, it is necessary to store partial solutions as intermediate memory.

5.2. Spatial phase

The goal of this part of the process is to group cells into rooms. First of all, the minimum functional spaces of the previously generated dimensionless plan are specified. Afterwards, an iterative algorithm using flooding techniques generates all possible spatial distributions. The adequacy of the rooms (size, proportion) is checked against the aggregation rules. Because constraints and aggregations rules are checked along the generative process, each solution found is a potentially proper solution.

Restrictions applied on the flooding algorithm are provided by the graph labels. Two cells can be joined to make a room if they have a connecting edge with a label of 100% accessibility and 100% visibility. This way, even cells with different functions could be combined. For example, kitchen and the living room cells could belong to the same space.

Applying a conventional technique to prune the search space was rather difficult, and we had to use one based on heuristics. At a certain point in the search, we first check whether any other search process has reached this point, because the strict conditions imposed by the aggregation rules will bring the process to the same solution as previously found.

5.3. Circulation phase

Starting with a room layout created in the spatial phase, the process proceeds to resolve the boundaries between rooms. In the dimensionless stage it was not possible to set room boundaries, since at that stage there no rooms had been defined. The aim of this phase is to obtain the best circulation scheme having the shortest distance between spaces and the shortest distance to the front door (shortest path problem) using a best first search algorithm to achieve this.

At first, private spaces are connected to public ones (corridors, walkways, staircases). Then, the path-finding process has to avoid generating a solution where a private room—such as a water-closet—can be used as circulation between rooms. At the end of the process, access to each room is granted with the minimum number of doors.

Providing a unique solution for each spatial layout is not as limited as it might seem, since future occupants will have the opportunity to customize the interior space by selecting different types of partitions (sliding doors, opaque or transparent walls) in the Housing Unit Configuration work space.
5.4. Design phase

In the final phase, all the information generated in the previous phases is assembled in order to create a floor-plan depicted according to architectural conventions: different walls (exterior, interior) are drawn with their thickness; doors and windows are inserted; and furniture elements drawn. In order to generate a 3D model of the housing unit or to export it to another program, geometric information is stored in different data structures. For instance, to export the housing units to Autodesk Revit it is required a tree data-structure in which walls and partitions have doors and windows as children.

The final result of the process is a dimensioned housing layout, which complies with the architectural rules embedded in the system. At the time of writing, the system has created over 10,000 different housing units. These units stored in the database are used to create a building in the Housing Units Assembly work space.

5.5. User's profiles and roles

Typically, the user of the Housing Unit Layout workspace is an architect whose role may be two-folded: as a system designer and as a project designer. As a system designer, the architect defines the characteristics of the architectural system providing the information which is stored in graphs, constraint trees and aggregation rules. Besides, the architect can generate housing layouts and evaluate the outcomes produced by the system (Figure 7).

*Figure 7. Interfaces used by the system designer to generate new housing layouts.*
The architect can also interact with the system as a project designer, who is in charge of starting and supervising the generative processes. For example, an architect can select a given housing unit schema—graph and constraint list—and specify the characteristics of the units which will be used to create a building (dimensions and areas on layout zones, access type, minimum and maximum square meters on each space, cancelation rules on some graph cells for a certain generation).

6. CLUSTERING

The generative process we have described generates housing layouts in batch processing which are stored in the system database. When a project development is initiated, the architect first searches in the solutions already stored in the database. In order to make this search manageable, we used clustering techniques to retrieve sets of solutions which respond to multiple criteria.

The architectural attributes extracted in the last stage of the generative process are used to cluster housing units. The clusters help the project architect to select a pool of housing units to be used for a project development. Two clustering algorithms have been applied to group housing units: k-means (Arthur et al. 2007) and Self-Organizing Maps (Kohonen 1995).

The aim of the k-means algorithm is to cluster n objects into k partitions minimizing the squared-error function (McQueen 1967). The input of this process is the object list (housing units) and the number of partitions desired. The algorithm is strict, meaning that each input object can belong only to a unique cluster.

SOM, or Self-Organizing Maps, is a technique for data visualization that reduces data dimensions applying neural networks (Figure 8). The inputs of the process are the network size (arbitrarily fixed) and the list of objects. The outputs are a Kohonen map and a similarities map. The advantage of this method is that it offers the possibility to visually interpret the results. On the other hand, its disadvantage is the high computational cost that it requires.

**Figure 8. Clusters generated with a SOM technique.**
Both techniques work properly insofar they are able to cluster housing units automatically, without the supervision of a user. At the time of writing, the 10,000 housing units generated by the previously described process have been gathered in 39 clusters.

7. SEARCHING SOLUTIONS

Through the searching process the work spaces Housing Units Layouts, Project Development and Housing Unit Configuration become intertwined. In the Project Development work space a project architect sets the conditions of the new building delimiting the scope of housing units which are most appropriate for the project development. Then, the project architect uses the searching process to create a subset of housing units that meet the project specifications. Later on in the collaborative design process, as future residents describe the characteristics of their dwellings in the Housing Unit Configuration workspace, they will choose their living units within the subset previously selected by the architect.

The search space is populated by the outcomes of the generative process running in batch mode and complemented by the clustering process. The inputs of the search process are the characteristics that housing units must have and a list of quantify modifiers (which are used as weight system) set by the user. The output is a list of housing units ordered by their degree of relevance. The calculation of the relevance value is based on vector space models of information retrieval literature (Singhal 2001).

The project architect uses an interface (Figure 9) to clusters of housing units. At the outset, the basic attributes of housing units can be used as queries, for example: surface equal to 70m², number of rooms to 2, and one room with water-closet integrated. Also, the user can then take advantage of the clusters automatically generated by the system to navigate in a homogeneous subset of the solution space. However, it is known that clustering techniques are not perfect and they can bring about messy groups conflating different dimensions (Inanc 2000). Tagging can be used to compensate these limitations (Quintarelli et al. 2007). The user-architect or future occupant uses tags to describe the housing units with their own words. This way, a knowledge base is collaboratively created as result of users interacting with the system. Tags can also be used as inputs of the search process and their quantify modifiers have more weight than the housing units attributes.

The searching process is also used in another context: in the Housing Unit Configuration work space the future inhabitant of the dwelling describes the housing program. The system gets the characteristics of the dwelling (number of rooms, dimensions or connections between spaces) from the future occupant and uses these attributes as input of the search process. In this case, the ranges
of the quantify modifiers are dynamically set for each kind of attribute. The solutions that are found are returned to the user for him or her to choose the three units that best suits the program requirements. This selection is used by the process that assembles the units in an apartment building.

**Figure 9. Interface to navigate in the solution space.**

8. CONCLUSIONS

To create an open and participative environment to support design and construction of flexible housing we have adopted an strategy consisting in breaking down building components, processes and activities, to facilitate the access of the different actors. The spatial structure of the housing units and buildings has been unfastened from the building system, so that the same design can be materialized in a different way. The generative process of the apartment building is handled in two separate work spaces: one to generate and store the floor-plans, and a second one to assemble the housing units in a block. In turn, the generative process of the housing units is split in four stages, each one of them dealing with specific design criteria. Finally, the user’s activity is distributed throughout the different workspaces, where users can intervene asynchronously, working individually or in collaboration with others user’s. In this way, we can provide an integrated environment that facilitates parallel forms of collaboration among the different actors (designer-manufacturer-contractor-
client) thus breaking away from the linear processes (from designer to contractor, from contractor to manufacturer) which still characterize the design and construction of a building worldwide.

ACKNOWLEDGEMENTS

This research project has been carried out with the support of the grant BIA2005-08707-C02-01 from the Spanish National R+D+i 2005-2008.

REFERENCES


