A Constraint-Based Building Bulk Design Support

Dirk Donath and Luis Felipe González
Chair Computer Science in Architecture
Bauhaus-Universitaet Weimar
caad@archit.uni-weimar.de

We introduce the BDS (Building Bulk Design Support) as an architecture practice-oriented implementation strategy of constraint-based design methods to support early phases of participatory planning. This ongoing research examines optimization problems of site coverage and building massing, according to a problem-solving approach based on the constraint satisfaction problems (CSP) paradigm. The case study is focused on Latin American self-management housing programs, testing the Chilean standard low-cost housing and planning regulations effective today. The BDS constitutes a novel approach on ICT support for the jointly collaborative involvement of lower-income households and NGOs into new ways of low-cost housing production in developing countries.

Introduction

Self-management housing programs have proved cost-effectiveness in reducing the housing shortage and improving the quality of low-cost housing in many developing countries by optimizing the use of public and private resources and expanding the civil rights of the poorest households. In the LAC region, planning consumes one-eighth of the total construction costs spent in housing [Tapia 02]. Thus, local capacities for self-management are mainly complemented by participatory planning procedures whereby dwellers themselves agree on choices about what habitation conditions suit their needs best and how to manage available resources, in order to gain access to home ownership or upgrade their current dwelling.

The current tendency in Chile - pioneer country in developing innovative housing policies - shows the progressive replacement of state's mass production of low-cost dwellings by a complete shift onto a competitive project-based funding scheme to supporting small self-organized groups of households that are technically supported by diverse NGOs. Project goals may include from the purchase to the production of new tenements. In fact, 85% of all funded projects correspond to housing construction on new land subdivisions [Nieto 05]. Most of these new tenements belong to some type of progressive-development housing program by which dwellers are on their own in charge of enlarging a small (from 6 to 30sqm) starter dwelling over time. Especially in Chile, progressive-development housing programs have turned into the only affordable housing alternative for the poorest quintile of the population. Indeed, 59,868 housing units of this type were finished within the last fifteen years [Minvu 05] and 215,000 new units are planned to be finished for 2010 (EL SUR, April 26th, 2006). Until 2001, technical assistance was allocated at a rate of 4 experts per 400 households [Minvu 04], today, NGOs are supporting individual groups of maximal 60 households per project [Cepal 06].

In this highly competitive housing subsidy scheme, efficiency regarding time and cost of planning directly affects dwellers' quality of life, especially when lacking adequate shelter during the project development time. Besides the shortage of economic resources, the next largest impediment to reduce effectively the project development time is first, a large number of not very intelligible normative constraints and second, the challenge of optimizing the inhabitable space upon very small sites of usually 60sqm surface. Furthermore, geometric reasoning and decision-making processes carried out at early phases of participatory planning may demand great effort in terms of communication and coordination between planners and dwellers. CAD-systems, traditionally used in the practice of architecture, are unable to support exploring different valid solution alternatives in a reasonable time.

Objectives

The general goal of our ongoing research is to improve efficiency in solving complex design problems that planners as well clients have to face at early phases of any planning process. The specific goal regarding this article aims at the implementation of a constraint-based planning methodology to support designing the optimum volume and shape (bulk) for a building upon a given site.

The structure of an architectural problem can be completely specified by a set of relevant constraints [Gross 86]. In this sense, a valid architectural solution is the result of the instantiation of a space configuration that satisfies a prescribed set of planning-relevant constraints. There are different types of normative constraints that have to be taken into account right from the beginning of every planning process. Satisfying the type of constraints on the bulk of a building is subject to the accomplishment of a class of complex-solving tasks, mainly because almost all bulk constraints are related to one another with distinct a degree of dependence. Thus, in order to create normatively valid solutions for the building bulk, architects have to deal with a complex system of non-redundant combination of different planning and zoning regulations that are applicable to the project program and its insertion context.

Unfortunately, conventional planning methods demand a large domain-specific knowledge base and certain dexterity to achieve the efficiency required in participatory planning. A customary planning procedure may consist of the following general steps:

1. *Create an instance of the eventual architectural solution (or part of it) by assigning a value to each configuration variable that specifies that instance in the three-dimensional space unambiguously.*

2. *Loop until every assigned value are labeled VALID:*

(a) *Verify whether the assigned value violates any normative constraint.*

i. *If the assigned value does not violate any normative constraint, label it as VALID.*

ii. *Otherwise, assign a new value and return to step (a).*

3. *Evaluate the valid solution instance according to optimization criteria and not formalizable constraints (e.g. on aesthetics):*

i. *If the valid solution does satisfy most of the optimization criteria and not formalizable constraints, quit.*

ii. *Otherwise, return to step 1.*

Eventually, step 3 may precede step 2. However, the exhaustive loop of validation (here, step 2) may consume a lot of time depending on the geometric complexity of the proposed solution, the type and amount of normative constraints to be satisfied, and the dexterity of the planner her/himself. In participatory planning, availability of time and resources to search for additional solution alternatives is rare.

We propose to describe the architectural design problem in terms of a set of geometric entities (shortened, geoms) and a set of relevant constraints. First, each geom represents a single spatial component of the design problem, which may be as well a building compartiment as a norm, like e.g. a rear setback requirement. Second, each constraint specifies how subsets of geoms should interact with each other. Strictly speaking, each constraint delimits the set of possible values that may be assigned to each configuration variable (real-valued parameters of shape and location) of a geom involved in the problem. Kramer (1994) calls the collection of geoms and constraints, the constraint system. Tentatively, our constraint-based planning procedure shall consist of the following general steps:

1. *Initialize the constraint system (prescribed according to the specific type of design problem)*

2. *Loop until every geometric entity you want to instantiate is specified in the three-dimensional space unambiguously in terms of its configuration variables.*

(a) *Assign a value to that configuration variable you want to edit.*

(b) *Evaluate the valid solution instance according to optimization criteria and not formalizable constraints (e.g. on aesthetics):*

i. *If the valid solution does satisfy most of the optimization criteria and not formalizable constraints, quit.*

ii. *Otherwise, return to step (a).*

In a computer aided constraint-based planning procedure, the validation loop itself shall be carried out internally by general deductive mechanisms the very constraint system specifies. The effect of an interactive constraint-based planning procedure may be perceived at user level as some already assigned values change automatically, i.e. they will adjust themselves to the new input values, due to the interdependence between subsets of their configuration variables. Likewise, some new input values will be unable to be increased unless others are decreased. Notice, that a computer implementation of a constraint-satisfaction approach in the planning process does not necessarily imply full automated search procedures, as may be the case of a different type of software applications called *constraint solvers* which are not considered in this article.

Currently, the lack of formal representations of elementary architectural design problems hinders an easier adaptation of a large number of useful problem paradigms like the very CSP. Likewise, the lack of architectural practice-oriented implementation strategies to allow the use of modern problem-solving techniques makes an ICT support rather difficult. Hence, the BDS strategy makes use of a CSP-based formal representation to describe the bulk design problem, allowing the development of a new highly interactive planning support tool by which semi-automatic procedures to search for additional valid solution alternatives may be carried out in real time. The variety of dwellings we want to be capable of designing with this tool includes detached, semi-detached, and attached buildings of up to three stories, upon convex rectangular site surfaces.

Methodology

In order to set up a constraint system that meets customary requirements of the abovementioned field of application, we determined which are the elementary spatial components and planning constraints involved in the bulk design problem. The great advantage of using the CSP paradigm in solving architectural design problems is that the standard representation of any state of the problem reveals the structure of the problem itself (See Fig. 1. Allowed position for freestanding volumes).

First, we differentiate three classes of spatial components: (A) the site, (B) the building and (C) the setback requirements. The building itself is made up of two types of spatial components: (B1) freestanding volumes and (B2) semi-detached volumes. The analysis on local regulations and housing typologies determined a set of twenty-five different spatial components to be the minimum number required to describe the bulk design problem graphically. We modeled each spatial component as a rigid body which is specified unambiguously in a fixed xyz coordinate frame by means of three translational and three dimensional degrees of freedom. Notice that for each spatial component of the problem, there is a geom representing it in the constraint system.

Second, we set up a taxonomy of all elementary constraints involved in the bulk design problem, determining four different classes:

Normative constraints are prescribed by planning and zoning regulations. Such constraints may directly specify a fixed value like e.g. building height = 7m; or a set of geometric and topological relationships between configuration variables of different geoms, e.g. *maximum allowable length of a zero-lot-line façade = 40% of the entire length of the side lot line.*

Intrinsic constraints are exclusively determined by geom's own geometric nature. Such constraints specify sets of possible values that may be assigned to a geom, according to its semantic definition. For instance, a setback requirement is graphically represented as a rigid body with which building freestanding volumes are supposed to collide. So, each setback volume has a fixed position by default, adjacent to its respective site borderline.

Interaction constraints are automatically

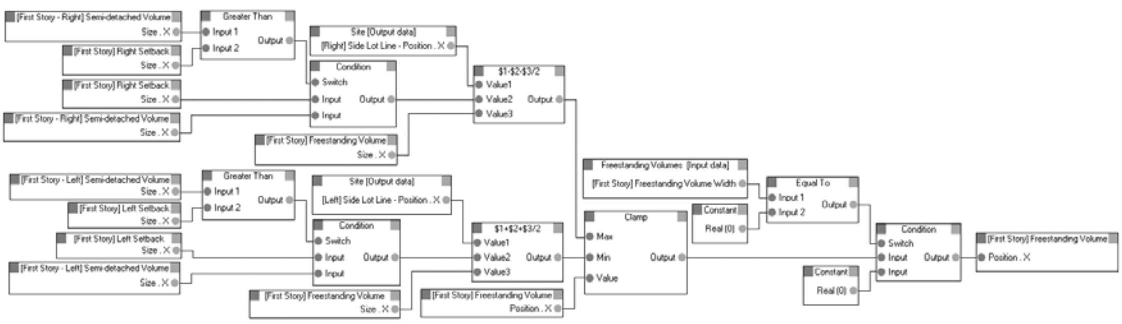

Figura 1

deduced by the value changes of geom's configuration variables. For instance, if the height of the first story changes, then the vertical position of the second story must also change. Although, such constraints provide consistency to the three-dimensional representation model, they also demand an exhaustive programming process.

User constraints are exclusively determined by the system user (dwellers and planners). Dimensional and translational values may be changed by means of text-based input or drag-&-drop techniques, which may be carried out directly on the editor's screen. Not to confuse the effect of user constraints with the fact that user may also change values concerning normative constraints, e.g. lot coverage coefficient. User constraints exclusively concern the design intentions and the very act of designing.

Third, we select six fundamental planning regulations of the Chilean building code:

Site coverage coefficient is a nonnegative real value that multiplied by the total site area, determines the maximum ground floor area of the building allowed upon that said site.

Buildable coefficient is a nonnegative real value that multiplied by the total site area, determines the maximum gross floor area of the building allowed upon that said site.

Setback requirement determines the minimum horizontal distance allowed between a determined lot line and the nearest point of a building or any projection thereof. There are different setback requirements for three height intervals ($h \leq 3.5m$; $3.5m < h \leq 7.0m$; $7.0m < h$), depending on whether the involved façade has fenestrations or not.

Story height determines the minimum allowable vertical distance between the finished floor of a room and its ceiling.

Building height determines the maximum vertical distance allowed between the natural ground level and the highest point of the building.

Zero-lot-line requirement determines the maximum allowable length and height of those vertical structures of the building that rest directly on a lot line.

Fourth, we set up a constraint system by using *Cinema 4D's XPresso*® node editor. This developer

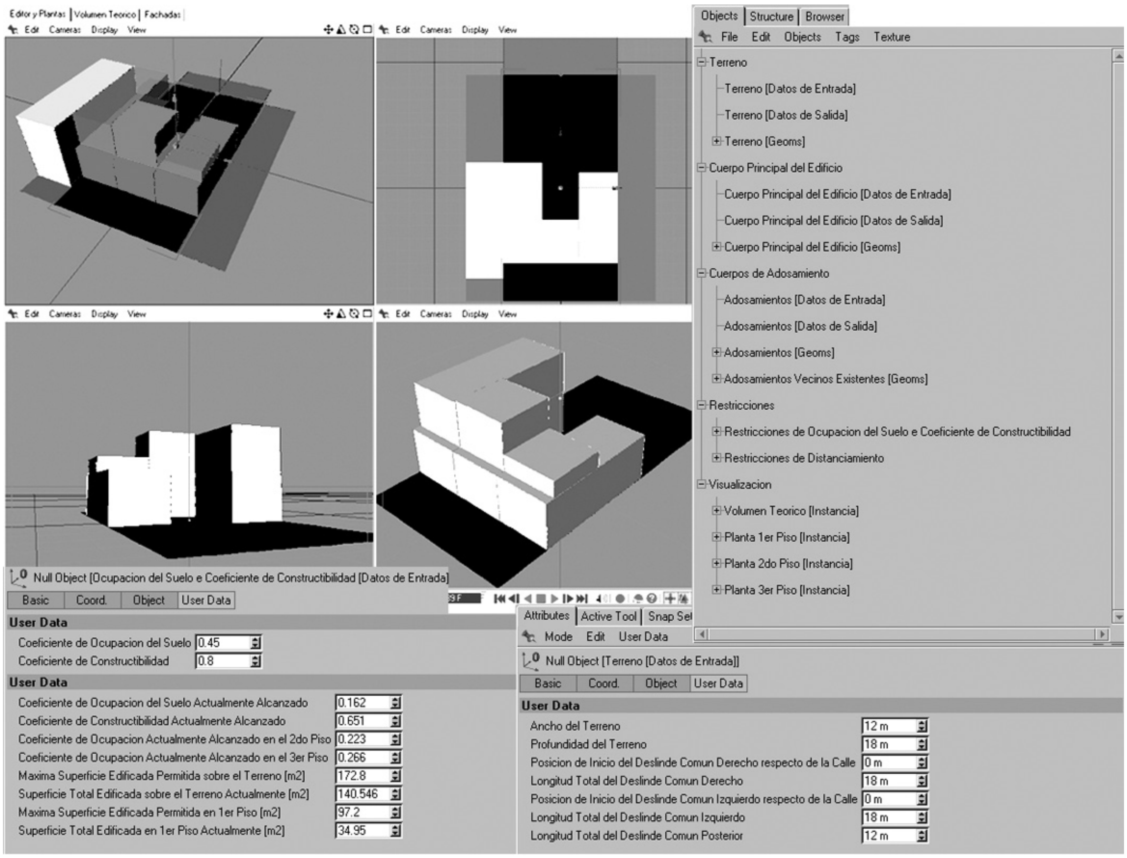

Figura 2

tool allowed us to set up the data structure for the bulk design problem in form of a large-scale ER diagram, which is directly compiled by that very software. A prototypical implementation of the BDS, has proved to be a highly efficient, interactive modeling and evaluation tool to generate in real time valid solutions for the building bulk design problem concerning the standard characteristics of low-cost housing and sites applied in Chile (See Fig. 2. GUI of BDS).

Conclusions

The formal representation of elementary architectural design problems allows both a better understanding on its specific solving requirements and the development of suitable support technologies. The customary error in CAAD research has been to attempt full automation of the planning process or part of it. A CSP-based taxonomy of different design problems in conjunction with a certain automation degree of geometric reasoning mechanisms to support participatory planning activities, may let dwellers and planners to refocus their attention on the decision making rather than on the tedious task of satisfying normative constraints. Besides, the definition of a constraint system type needs to be carried out only once in order to solve all design problems of its type. Consequently, different types of constraint systems to represent respective types of design problems may be standardized. Finally, a constraint-based participatory planning process may provide lower-income households with an advantageous tool to explore additional design solution alternatives, in less time and with less effort than applying conventional methods of architectural planning.

References

Cepal. 2006. Fondo Solidario de Vivienda. Comisión Económica para América Latina y el Caribe, Experiencias emblemáticas para la superación de la pobreza y precariedad urbana en América Latina y el Caribe.

Gross, M. 1986. Design as Exploring Constraints. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge.

Kramer, G. 1994. A Geometric Constraint Engine. In Constraint-Based Reasoning: Special Issues of Artificial Intelligence, Freuder, E. and Mackworth, A. (eds.), The MIT Press, ISBN 0262560755, pp. 327-360.

Minvu. 2005. Viviendas Terminadas y Subsidios Pagados, 1990-2005. Ministerio de Vivienda y Urbanismo. www.minvu.cl

Minvu. 2004. Programa de Asistencia Técnica Sectorial. Ministerio de Vivienda y Urbanismo. Resolución No. 533 de 1997, JTF/AHM/CMR, 31.07.2004.

Nieto, M. 2005. Intervention by the IV Jornada de Vivienda Social: Síntesis Panel No. 3, Impacto de la Política Habitacional, Santiago de Chile.

Tapia, R. and Mesías, R. 2002. Hábitat Social Progresivo, Vivienda y Urbanización. Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo – CYTED, ISBN 9568263004, Santiago.

Keywords:
constraint-based design, participatory planning, low-cost housing.