14 del 2 al 5 de Diciembre de 2008
CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA

44 ANIVERSARIO cujae 2008

Congreso SIGraDi Cuba 2008
Gráfica Digital
Integración y Desarrollo
La Habana 1 al 5 Diciembre 2008

Enseñando diseño generativo: una experiencia didáctica

Gabriela Celani
celani@fec.unicamp.br
Universidad Estatal de Campinas, Brasil

Resumen. **Este trabajo describe el seminario de Diseño Generativo ofrecido en el programa de posgrado en el área de Arquitectura y Construcción en la Universidad de Campinas en el primer semestre de 2008. El curso tuvo como objetivo introducir el tema de los métodos sistemáticos de generación de alternativas de diseño, como base para el desarrollo de sistemas de CAAD. El método pedagógico del curso estuvo compuesto por presentaciones teóricas, lecturas, discusiones, y ejercicios prácticos. Los sistemas generativos más conocidos fueron categorizados en dos grandes grupos: el de combinaciones de elementos (como la enumeración sistemática de combinaciones y los algoritmos genéticos) y el de los sistemas basados en reglas (como los fractales y la gramática de formas). Además de constituirse en una oportunidad para proponer una categorización de los principales sistemas generativos, el curso también permitió que se hiciera una reflexión sobre la importancia de la introducción de los sistemas generativos y de las técnicas de programación en los cursos de pre-grado y pos-grado.**

Palabras Claves. Sistemas generativos de diseño; base teórica del CAAD.

I. INTRODUCCIÓN

Este trabajo describe una experiencia de seminario de posgrado con duración de un semestre y carga horaria de 45 horas, que tuvo como alumnos diez arquitectos candidatos al título de *master* en Ingeniería Civil - área de Arquitectura y Construcción - en la Facultad de Ingeniería Civil, Arquitectura y Urbanismo de la Universidade Estadual de Campinas, Brasil.

El seminario fue inspirado por un trabajo de Fischer y Herr [2], en el que los autores discuten que se necesita enseñar para que se pueda introducir efectivamente el CAD generativo en los currículos de diseño. Los autores afirman que la cuestión no está en "qué generar", sino en "como generar", y proponen que se incluyan temas teóricos como sistemas emergentes, auto-organización, gramáticas generativas, generación algorítmica, diseño evolucionário y diseño paramétrico. En lo que se refiere a la tecnología, Fischer y Herr [2] reconocen la importancia de la introducción de técnicas de programación, pero afirman que también es posible enseñar diseño generativo a personas que no saben programar. En este seminario aproximadamente la mitad de los alumnos ya había tomado clases de programación el semestre anterior (VBA script para AutoCAD), mientras que la otra mitad no tenía ningún conocimiento de programación.

II. MÉTODO PEDAGÓGICO

El método pedagógico del curso estuvo compuesto por presentaciones teóricas, lecturas de textos, discusiones, y ejercicios prácticos. La bibliografía del seminario incluya textos de las décadas de 70, 80 y 90, lo que reflete la intención de introducir el tema bajo una perspectiva histórica.

El texto "*The theorectical foundations of computer-aided architectural design*", de Mitchell [1] fue presentado como primera lectura y su discusión sirvió como base para la comprensión de los objetivos de los sistemas generativos y sus aplicaciones en el desarrollo de sistemas CAD. El diseño generativo fue presentado como una manera sistemática de generar alternativas de diseño. El concepto espacio de búsqueda (*search space*) presentado por Mitchell fue fundamental para la comprensión del distinto alcance de los diferentes métodos discutidos a lo largo del curso (Figura 1). Inicialmente, se imaginó que la dinámica del seminario sería basada apenas en lecturas y discusiones. Sin embargo, las lecturas les parecieron demasiado abstractas a los alumnos, que sentían falta de ejemplos prácticos de los sistemas generativos sobre los que estaban leyendo.

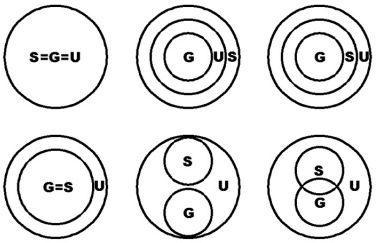


Figura 1: El espacio de búsqueda: soluciones potenciales (S), soluciones generadas (S) y soluciones ideales (G), según Mitchell [1].

Por ese motivo se trató de cambiar la metodología: se pasó a intercalar las clases teóricas con clases de estudio, en las que se proponía un ejercicio de diseño en grupo con el uso del sistema generativo que se estaba estudiando.

III. SISTEMAS GENERATIVOS: DOS GRUPOS

Los sistemas generativos fueron presentados en dos grandes categorías principales: la de intercambios de elementos y la de sistemas basados en reglas. Las lecturas sugeridas sobre cada tema coinciden con las referencias bibliográficas presentadas abajo, en cada sección.

A. *Intercámbio de elementos*

La combinación de elementos intercambiables es la manera más manera más simple de generar alternativas sistemáticamente. El estudio de este tipo de sistema generativo empezó con una revisión de los fundamentos matemáticos de la análisis combinatoria [3]. Fueron presentados los métodos para cálculo del número de alternativas posibles según el número de elementos intercambiables y el número de alternativas posibles para cada uno de ellos. Fueron presentados los siguientes ejemplos de sistemas generativos basados en técnicas de combinatoria: uso de matrices para intercambiar elementos constructivos matrices [4, capítulo 4 – *Matrices and vectors*] cuadros morfológicos [6] (Figura 2), y sistemas paramétricos [5]. En el caso de la substitución de parámetros es necesario primeramente hacer una descripción paramétrica de lo que se quiere generar. En seguida, los parámetros son substituidos por un número de valores posibles, y combinados de todas las maneras posibles.

	Alternativas				
Elementos	1	2	3	4	5
1. Esquadria	Alumínio	Aço	Madeira		
2. Vidro	Vidro simples	Vidro duplo			
3. Sistema de abertura	Vertical de correr	Horizontal de correr	Horizontal de abrir	Fixo	Pivotante
4. Sistema de privacidade	Cortina com trilho	Cortina rolô	Vidro espelhado		
5. Sistema de proteção contra o sol	Venezianas de abrir	Venezianas de correr	Painéis de fechamento	Brise-soleil	

Figura 2: Ejemplo de cuadro morfológico, basado en [6].

En seguida, fue introducido el tema de los grafos [7]. Fueron presentadas nociones de conectividad, completud y planaridad de los grafos (Figura 3).

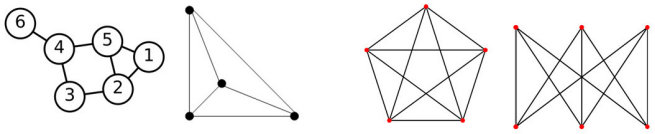
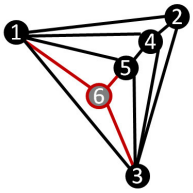


Figura 3: Grafos planares (izq.) y no planares (der.).

Fueron también presentadas las matrices de adyacencia y como se determina si su grafo correspondiente es plano o no (Figura 4). Solo es posible convertir un grafo en un plano arquitectónico si el grafo es plano.



0	1	2	3	4	5	6
1	0	1	1	1	1	1
2	1	0	1	1	0	0
3	1	1	0	1	1	1
4	1	1	1	0	1	0
5	1	0	1	1	0	1
6	1	0	1	0	1	0

G=grafo
v=número de vértices
a=número de arestas
(para v ≥ 3)
se a<= 3*v - 6
então G é planar

Figura 4: Grafo (izq.), matriz de adyacencia (centro) y cálculo de la planaridad del grafo (der.).

El ejercicio práctico con grafos consistió en desarrollar primeramente una matriz de adyacencias y un grafo a partir del plano de un edificio existente. En seguida, cada grupo entregó su grafo a otro grupo, para que estos crearan un nuevo plano arquitectónico a partir del grafo que habían recibido. Como se podría suponer, los planos generados a partir de los grafos eran todos distintos de los planos originales (Figura 5).

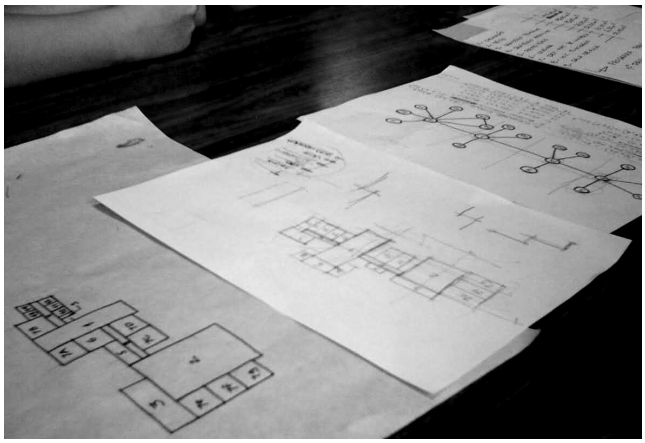


Figura 5: Dos planos diferentes resultantes de un mismo grafo.

El texto de Liggett [8] fue sugerido como lectura por demonstrar que los grafos son todavía un eficiente método de representación abstracta y generación de planos. Sin embargo, como advertido por Steadman [7], la generación sistemática de planos posibles para un mismo grafo crece de manera impresionante a partir de los grafos con 7 nudos, lo que hace con que ese método solo sea práctico para planos con un número reducido de espacios. El tema de los algoritmos genéticos fue introducido en el curso en ese momento, precisamente como una manera de solucionar el problema del exceso de alternativas.

Una vez que se obtienen todas las variaciones posibles de una determinada situación, es necesario evaluarlas para decidir cuál es la mejor desde uno o más puntos de vista. A este proceso se llama optimización. Algunas veces el número de combinaciones posibles es tan grande que resulta imposible generar y además evaluar y seleccionar la mejor de ellas. Los algoritmos genéticos (AG) permiten que se explore el espacio de búsqueda en profundidad, pero sin la necesidad de generar todas las alternativas posibles. Aún que este método no garantice necesariamente la solución ideal, permite que se llegue a una solución satisfactoria y muy probablemente próxima de lo ideal. Los AG están inspirados en la combinación de genes presente en todos los seres vivos.

La literatura sobre las aplicaciones de los AG sea muy vasta, pero se buscó en este curso presentar lecturas sobre trabajos específicos de aplicaciones de los AG en el planeamiento espacial – por eso se utilizó el trabajo de Thorsten y Gero [9], que incluye a la vez detalles sobre *cross-over*, mutación y hasta genes dominantes y recesivos.

En el ejercicio de AG se pidió a los alumnos que identificaran una situación de diseño en la que habían diversos parámetros para optimizar. Los grupos deberían especificar un código genético de su diseño con la secuencia de estos parámetros y realizar las principales etapas de un AG: (1) inicialización - la creación aleatoria de la primera generación, (2) evaluación - la selección de los individuos más adaptados de esta generación, (3) cruzamiento – la recombinación de los genes de cada pareja, y la repetición de los pasos 2 y 3 hasta llegar a una buena solución.

Los diferentes grupos desarrollaron AG´s para distintas situaciones de layout con multi-criterios, como la organización de las mesas en un restauran (Figura 6), de sillas en un auditorio y de plazas en un aparcamiento. Los estudiantes se quedaron sorpresas con los resultados obtenidos, según ellos con parámetros muy diferentes de lo que ellos intentarían utilizar.

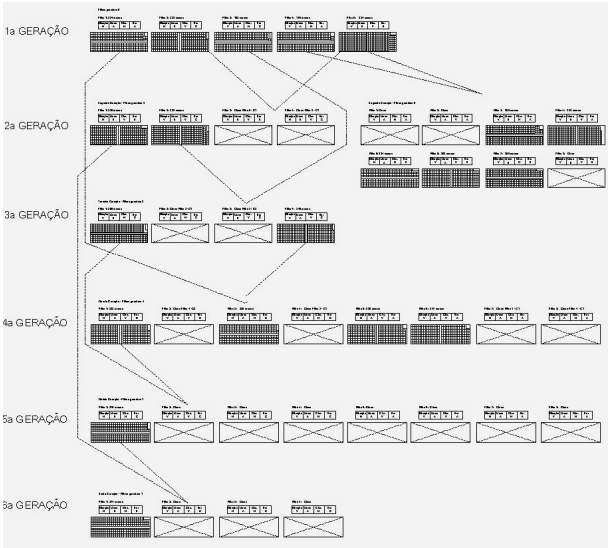


Figura 6: AG desarrollado para el layout de mesas en un restaurán, con los siguientes criterios: dos tipos de mesas, dos orientaciones posibles de las mesas, y cinco maneras de distribuir la circulación (Rafael Mateus, Sérgio Righetto y Max Andrade).

B. Sistemas basados en reglas

Los sistemas basados en reglas permiten producir un número prácticamente ilimitado de alternativas a partir de un número finito de reglas. En este caso las variables de la combinación son el orden de la aplicación de las reglas, la forma a la que se aplican, y el número de veces que se aplican a cada forma. Cuando se utilizan sistemas basados en reglas el número de posibilidades es infinito. Sin embargo, estos sistemas no tienen como objetivo generar todas las alternativas posibles, sino explorar distintas soluciones posibles de manera heurística y no determinística.

El ejemplo más conocido de sistema basado en reglas es el diseño simétrico. En este sistema las reglas utilizadas son las de reflexión, rotación y translación. Estas reglas añaden a la forma original otra forma igual, pero con posición, orientación o quirilidad diferentes de la original. March y Steadman [4] han descrito los 17 grupos de simetría en el plano. Sus ejemplos demuestran la obtención de patrones variados y complejos a partir de figuras muy sencillas, según las reglas de transformación que se apliquen. Como ejercicio sobre el tema de la simetría se puede utilizar cualquier sistema CAD, con las funciones de reflexión y matrices rectangulares y polares.

Otro ejemplo muy típico de sistema basado en reglas son los fractales [10]. La principal diferencia entre las reglas de simetría y de los fractales es que las primeras hacen adiciones, mientras que las segundas hacen substituciones. Además, diferentemente de lo que ocurre en la simetría, las reglas de los fractales no son genéricas, sino específicas para una determinada forma, pero se pueden aplicar en cualquier escala.

Un ejemplo de aplicación de fractales en un estudio de arquitectura ha sido descrito por Chris Yessios [11]. Las reglas fractales también pueden ser utilizadas para generar complejidad a partir de formas simples. El uso de reglas resulta en inteligibilidad, porque permite que se note una lógica subyacente. Como ejercicio sobre el tema de los fractales se utilizó un script desarrollado por la autora, que permite crear fractales en AutoCAD (Figura 7).

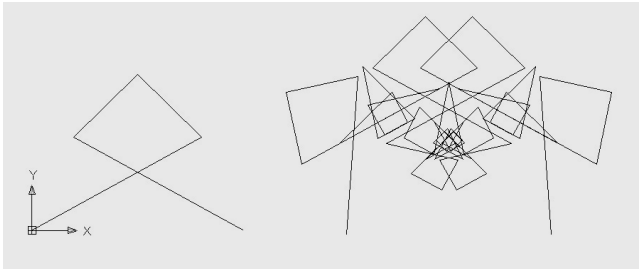


Figura 7: Ejemplo de un fractal generado con el script para AutoCAD utilizado en el curso.

Otro ejemplo de uso de reglas para la generación de alternativas es la gramática de la forma. El tema es introducido de manera completa y muy didáctica por Knight [12]. La diferencia entre los fractales y la gramática de la forma está en que en el primer se hace una aplicación sistemática y repetida

de reglas, mientras que en la segunda se elige una regla específica a cada etapa, de manera heurística. Como resultado, el diseñador tiene mayor control sobre el proceso. Además, con la gramática de la forma es posible reconocer formas emergentes y aplicar reglas sobre esas formas.

Como ejercicio para esta parte del curso, se pidió que los estudiantes desarrollaran gramáticas de la forma originales o basadas en el análisis de algún lenguaje arquitectónico existente. La Figura 8 presenta una gramática desarrollada a partir del análisis del conjunto de viviendas protegidas São Francisco, en São Paulo. Los alumnos fueron capaces de comprender la lógica compositiva de los bloques de habitación y de establecer reglas que permitían recrear casa iguales a las existentes, además de otras totalmente nuevas.

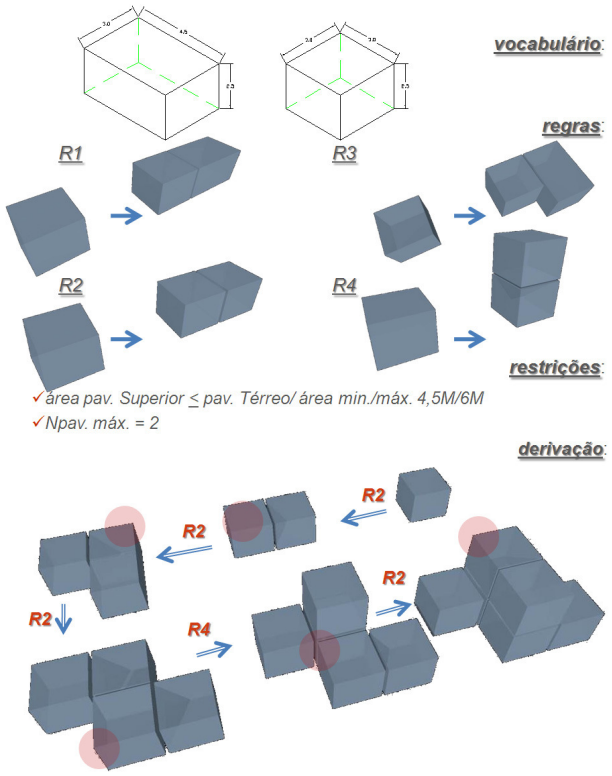


Figura 8: Gramática desarrollada a partir del análisis del conjunto de viviendas protegidas São Francisco, en São Paulo (Max Andrade).

C. *Combinaciones de los dos sistemas*

Es posible combinar los sistemas basados en intercambios y los basados en reglas. Gero y Kazakov [13] presentan un ejemplo en el que el código "genético" de los planos de un edificio son determinados por la secuencia de reglas que se deben aplicar para obtenerlos.

Por falta de tiempo, no se llegó a pedir que los estudiantes desarrollaran un sistema como este, pero eso sería interesante, como manera de dejar más clara la distinción entre los dos tipos de sistemas generativos.

IV. RESULTADOS

En la discusión final del curso los alumnos dijeron estar muy satisfechos. Algunos de ellos llegaron a sugerir que la la materia fuera adaptada como materia obligatoria para el curso de pregrado en Arquitectura y Urbanismo en la FEC-UNICAMP, reconociendo la importancia de enseñar estrategias de generación de alternativas a jóvenes arquitectos. Sin embargo, también fueron presentadas algunas críticas y sugerencias que serán incorporadas en los próximos cursos.

La principal crítica estuvo relacionada a la fecha de publicación de los textos utilizados. El texto introductorio de la materia, por ejemplo, un trabajo de William Mitchell publicado en el año 1975 [1], fue considerado demasiado antiguo y supuestamente desactualizado. Sin embargo, el texto es importante porque es unos de los primeros a proponer el estudio de los sistemas generativos como base teórica para el CAD. Sin embargo, es importante buscar referencias actuales que puedan ser presentadas juntamente con textos más antiguos, que demuestran que el diseño generativo tiene su lugar en la tiene su historia de la teoría de la arquitectura. El texto de Fischer y Herr, del 2001 [2], es un buen ejemplo de texto introductorio más reciente que se puede presentar junto con el de Mitchell [1]. El texto de Ligget [8], del 2000, también es excelente en ese sentido, porque además presenta el tema del uso de los grafos en en planeamiento espacial desde una perspectiva histórica.

Otro resultado interesante fue que algunos alumnos que ya habían hecho una materia de programación pidieron para volverla a hacer (lo que infelizmente no es posible en nuestra facultad). Cerca de la mitad de los alumnos ya había tomado una clase de scripts para CAD, mientras los demás no tenían ningún conocimiento de programación. Los alumnos que sabían programas dijeron que ahora veían otra aplicación para la programación, mucho más útil. Los alumnos que todavía no sabían programar, por otro lado, tuvieron más dificultad en comprender los conceptos de diseño generativo, sobre todo las discusiones sobre substituciones de parámetros, y pasaron a interesarse por programación. De hecho, muchos de ellos se matricularon en la materia de programación el siguiente semestre.

La principal sugerencia hecha por los alumnos fue que se incluyeran más ejercicios prácticos a lo largo del curso, como manera de consolidar mejor los conceptos teóricos.

V. CONCLUSIONES Y TRABAJOS FUTUROS

Durante el seminario ha sido posible concluir, como sugerido por Fischer y Herr [2], que el diseño generativo es una importante herramienta para la formación del arquitecto, y que el conocimiento básico de programación es importante para la comprensión de las teorías de diseño generativo. Lo ideal sería llevar paralelamente el curso de introducción a la programación y el de diseño generativo. Esto permitiría el desarrollo de más actividades prácticas directamente relacionadas a la teoría, lo que sedimentaria el aprendizaje y le daría más significado.

Además de estas conclusiones, el seminario permitió que se iniciara una interesante reflexión sobre como agrupar los sistemas generativos. Aparentemente, es posible separarlos, según su esencia, en apenas dos grandes grupos: los basados en substituciones y los basados en reglas. Espera-se llevar adelante esta discusión, iniciada simplemente por la necesidad de estructuración del curso, con otros investigadores del área.

RECONOCIMIENTOS

La autora agradece a FAPESP (Fundación de Amparo a la Investigación del Estado de São Paulo) por el soporte financiero a esta investigación, y a todos los alumnos que participaron del seminario por su dedicación y entusiasmo durante todo el semestre.

REFERENCIAS

[1] W. J. Mitchell, "The theoretical foundation of computer-aided architectural design", Environment and Planning B, vol. 2, pp. 127-150, 1975.
[2] T. Fischer and C. M. Herr, "Teaching Generative Design" , in Proc. 4th International Generative Art Conference, Milano, 2001.
[3] J. P. O. Santos, M. P. Mello and I. T. C. Murari, Introdução à Análise Combinatória. São Paulo: Ed. Ciência Moderna, 2008.
[4] L. March and P. Stadman, The geometry of environment - an introduction to spatial organization in design. Cambridge, MA: The MIT Press, 1974.
[5] L. N. Avdot´in, "The use of computing techniques in the design of residential buildings", Environment and Planning B, vol. 7, pp. 47-86, 1980.
[6] F. Zwicky, "The morphological approach to discovery, invention, research and construction", in New Methods of Thought and Procedure, F. Zwicky and A. Wilson, Eds. Berlin: Springer, 1967, pp. 273–297.
[7] P. Steadman, "Graph-theoretic representation of architectural arrangement", in The Architecture of Form, L. March, Ed. London: Cambridge Univ. Press, 1976, pp. 94-115.
[8] R. S. Liggett, "Automated facilities layout: past, present and future", Automation in Construction Automation in Construction, Vol. 9, no. 2, pp. 197-215, March 2000.
[9] S. Thorsten and J. Gero, "Dominant and recessive genes in evolutionary systems applied to spatial reasoning", in Proc. Advanced Topics in Artificial Intelligence, Perth, 1997.
[10] M. Batty and P. Longley, Fractal Cities. London: Academic Press, 1994.
[11] C. Yessios, "A fractal studio", in Proc. Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), Raleigh, 1987, pp. 169-181.
[12] T. Knight, "Shape Grammars in Education and Practice: History and Prospects", International Journal of Design Computing, vol. 2, 2000.
[13] J. Gero and V. Kazakov, "Evolving building blocks for design using genetic engineering: a formal approach", in Proc. Advances in Formal Design Methods for CAD, London, 1995, pp. 31-50.

Gabriela Celani nació en São Paulo, Brasil, en el año 1967. Estudió Arquitectura y Urbanismo en la Universidad de São Paulo (USP). Hizo el master en la misma universidad (1997), y el Ph.D. en *Design & Computation* en el Massachusetts Institute of Technology (2002) bajo la orientación de William Mitchell y Terry Knight. Es actualmente profesora adjunta en la Universidade Estatadual de Campinas (UNICAMP), donde coordina el Laboratório de Automación y Prototipage para Arquitectura y Construcción (LAPAC). Página web: www.fec.unicamp.br/ ~lapac. Dirección electrónica celani@ fec.unicamp.br. Dirección: Av. Albert Einstein 951, CEP13083-852, Campinas, SP, Brasil.