

ANDRÉS CAVIERES*

Diseño paramétrico orientado a la *performance*

Parametric design performance oriented

<RESUMEN>

Este artículo presenta el concepto *performance* de edificios y cómo las tecnologías CAD/BIM pueden soportar el desarrollo de diseños orientados a su optimización. Se discute técnicas de modelamiento paramétrico que pueden ayudar a construir procesos colaborativos a partir de la explicitación de reglas de diseño, del análisis de alternativas y la selección de soluciones.

<ABSTRACT>

This article presents the concept of building performance and shows how BIM models can support design processes that focus on performance optimization. Parametric modeling is discussed as a key technology for such optimization since it allows for the computational generation of design alternatives. The potentials for collaborative and multidisciplinary design promoted by domain expertise embedded in parametric models, is also reviewed in the context of sustainable design, structural analysis and fabrication.

<PALABRAS CLAVE>

PERFORMANCE / BIM / DISEÑO PARAMÉTRICO / PROGRAMACIÓN / PROCESOS

<KEY WORDS>

PERFORMANCE / BIM / PARAMETRIC DESIGN / PROGRAMMING / PROCESS

Introducción

John Frazer menciona en su artículo «*The generation of Virtual Prototypes for Performance Optimization*» que durante los años ochenta tenía por costumbre terminar sus presentaciones en conferencias con una diapositiva que contenía la siguiente pregunta:

«¿Puede alguien recordar acaso qué había en el diseño que necesitara de asistencia antes que tuviéramos acceso al Diseño Asistido por Computador?»¹.

Esta pregunta se mantiene especialmente pertinente más de veinte años después. En el mismo artículo Frazer nos entrega su respuesta, la cual es dada bajo la perspectiva de los proyectos de geometría compleja que podemos observar en muchas de las

publicaciones de arquitectura contemporánea. Su respuesta se desarrolla más o menos así:

«Me pregunto cuál es el propósito o la necesidad de desarrollar proyectos basados en geometrías avanzadas y complejas. ¿Para qué? ¿Por qué? ¿Para el beneficio de quién? ¿Para resolver qué problema? ¿Es acaso para resolver problemas de constructibilidad? ¿Para mejorar la industria de la construcción? ¿Mejorar el ambiente? ¿Energía? ¿Performance? ¿O para producir clones de Gehry? ¿O su propósito es el lucro de los fabricantes de software? Ahora mi respuesta es que el interés no está en las geometrías complejas en sí. No son ellas las interesantes, sino que lo son los procesos subyacentes por los cuales generamos la geometría avanzada así como las estructuras de datos que las posibilitan lo que es realmente significativo»².

* Arquitecto de la Universidad de Chile. Estudiante del Doctorado en Diseño Computacional, *Georgia Institute of Technology*.

¹ Frazer, J. «The generation of virtual prototypes for performance optimization». En: Oosterhuis, K., Feireiss L. *The architecture co-laboratory: Game set and match II - On computer games, advanced geometries and digital technologies*. Rotterdam: Episode Publishers, 2006.

² Frazer, J. *Op. cit.*

El presente artículo busca entregar algunas luces respecto a dichos procesos subyacentes y describir, en parte, la dimensión computacional que soporta dichos procesos. La perspectiva adoptada sin embargo no refiere exclusivamente al uso de geometrías avanzadas o complejas en si, sino más bien se centra en una definición más general de la noción de diseño paramétrico, y de su rol en el contexto de la optimización de la *performance* de los edificios.

Contexto general

Diseño basado en performance

La arquitectura enfrenta actualmente nuevos desafíos, especialmente cuando es entendida en su dimensión productiva junto a las áreas de la ingeniería y de la construcción. Esta triada, conocida en algunos países como la industria AIC (o en sus siglas en inglés AEC), está siendo presionada para que mejore radicalmente los estándares de calidad de sus productos, particularmente desde el punto de vista de la *performance* energética y medio ambiental.

Las preocupaciones iniciales surgidas hace algunos años por la sustentabilidad de los diseños están dando lugar a políticas cada vez más agresivas que se traducen en normativas y exigencias³ más altas respecto a la *performance* de los edificios durante el ciclo completo de su vida útil, es decir, desde su etapa de construcción, pasando por el impacto que se pueda producir durante sus años de uso, hasta las consecuencias ambientales que implicaría su demolición y desmantelamiento.

De esta forma varias metodologías de administración de ciclo de vida de edificios⁴ están siendo incorporadas gradualmente a esta industria, caracterizada hasta ahora por ser una de las más ineficientes y contaminantes del mundo⁵. El objetivo de estas metodologías es la optimización de la *performance* de los proyectos con relación a una serie de aspectos, entre los que últimamente destacan los relativos al rendimiento energético y el impacto medio ambiental. Sin embargo la

noción de *performance* de edificios abarca una amplia gama de otras dimensiones que son susceptibles de análisis y evaluación.

Aspectos cuantitativos como comportamiento estructural, térmico, acústico y lumínico, calidad del aire interior y tasas de ventilación, figuran como algunos de los problemas sobre los que se han determinado métricas e indicadores precisos de *performance*⁶ mientras que ciertas dimensiones de orden cualitativo del espacio que afectan la percepción y el comportamiento de los habitantes (tales como la relación entre campos visuales y patrones de ocupación de espacios, grado de interacción espacial entre ocupantes de un edificio, entre otros) también están siendo definidos y aplicados de forma concreta.

De hecho es posible establecer mediciones sobre cada aspecto tangible u observable de un espacio o edificio sobre los que se pueden determinar estándares de calidad. Aunque para muchos pueda resultar discutible evaluar un proyecto de arquitectura sobre la base de su adecuación con estándares objetivos, es un hecho cada vez más evidente que criterios intuitivos basados en las preferencias personales o en la experiencia profesional del diseñador se están volviendo insuficientes para validar decisiones de diseño. Solamente la definición de métricas e indicadores de *performance* claros pueden ayudar a constituir un criterio común para prever y controlar el comportamiento futuro de las obras de arquitectura.

Procesos y estructuras de datos

Tecnologías BIM

Hoy en día las herramientas más importantes para el diseño, construcción y administración del ciclo de vida de los proyectos de arquitectura son las aplicaciones que conforman el modelo de información de edificios BIM. Este modelo se fundamenta en un conjunto de tecnologías y procesos asociados que constituyen un entorno de diseño integrado⁷. Entre las primeras ventajas

que han contribuido a la creciente adopción de modelos BIM esta la facilidad y eficiencia con que se pueden extraer información constructiva actualizada y consistente a lo largo del proceso de desarrollo de un proyecto. Esta capacidad contribuye a reducir significativamente errores de coordinación entre sistemas facilitando la comunicación entre diferentes especialidades, de este modo el proceso de ejecución de obra se ve mejorada y el modelo sirve posteriormente como fuente de referencia para la administración de las instalaciones del edificio ya construido.

Sin embargo una de las potencialidades más interesantes de las tecnologías BIM está en el hecho que el modelo integrado de información puede utilizarse como un prototipo virtual sobre el que se pueden efectuar una serie de análisis y experimentos predictivos. Sobre este prototipo virtual, equipos multidisciplinarios de diseñadores, ingenieros, fabricantes y consultores pueden simular aspectos concernientes a sus disciplinas de forma concurrente, y resolver colaborativamente conflictos existentes en situaciones de borde o de interferencia entre subsistemas. Esta aproximación permite la resolución temprana de problemas a través de un proceso de desarrollo sistémico del proyecto.

El potencial de beneficios ofrecidos por los modelos BIM, incluyendo el desarrollo concurrente y colaborativo se apoya en tres tecnologías fundamentales: 1) una base de datos relacional que contiene toda la información del edificio virtual, 2) un formato neutral de intercambio de información (IFC o XML) que permite la interoperatividad entre aplicaciones y 3) un sistema de modelamiento paramétrico que define y controla las relaciones geométrico-espaciales fundamentales del modelo tridimensional.

Es importante destacar que el modelamiento paramétrico más bien refiere a una forma o metodología de modelamiento que a una tecnología propiamente tal. Muchas tecnologías han sido desarrolladas para facilitar la parametrización de los modelos geométricos, pero en esencia, cualquier

³ En este sentido el sistema de certificación norteamericano LEED está siendo usado como pauta de referencia global para la elaboración de normas y requerimientos orientados al diseño y construcción de edificios sustentables.

⁴ Eastman, C. *Building product models*. Nueva York: CRC Press, 1999.

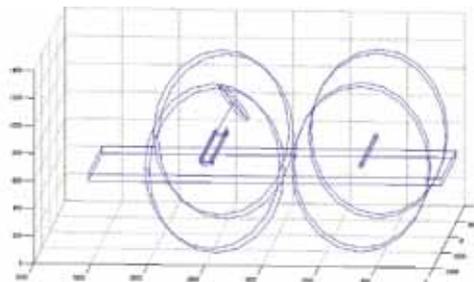
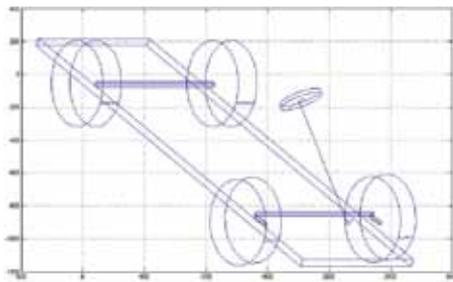
El concepto de administración de ciclo de vida de edificios está basado en el principio de PLM (*Product Lifecycle Management*) proveniente del área de diseño y desarrollo de productos manufacturados.

⁵ La industria de la construcción es responsable del 30% al 40% del consumo energético del mundo. Sólo en Estados Unidos ella es responsable del 30% de la producción de gases invernadero y el 30% de los desechos producidos (<http://www.usgbc.org>).

⁶ Augenbroe, G. *Advanced building simulation*. Nueva York: Spon Press, 2004.

⁷ Eastman, C., Teicholz, P., Sacks, R., Liston, K. *BIM Handbook: A guide to building modeling for owners, managers, designers, engineers and contractors*. Wiley, 2008.

Figura 1: Modelo paramétrico conceptual de un automóvil usando MATLAB. Variaciones entre los diferentes modelos se deben a la concatenación de matrices que definen transformaciones geométricas básicas (traslación, rotación y escalamiento).



```
% chassis from parametric cube
chScale = scale3D (A, B, C);
chRot = rotate3Dz (degree_to_radian (0.0));
chOr = rotate3Dx (degree_to_radian (0.0));
chTrans = translate3D (0.0, 0.0, G);
chassis_Orient = g * chTrans * chRot;
chassis = chassis_Orient * chOr * chScale * cube;

% steering rack
f_rackScale = scale3D (S, Ss, Ss);
f_rackTrans = translate3D (0.0, (B-E)-100, (W*0.2)+ Ks);
f_rack_Orient = chassis_Orient * f_rackTrans;
steerRack = f_rack_Orient * f_rackScale * cube
```

software de modelamiento tridimensional dotado de interfaz de programación puede ser utilizado paramétricamente (Figura 1).

El principio básico de un modelo paramétrico yace en que sus relaciones geométricas dependen de valores numéricos o expresiones lógicas de tipo condicional que controlan la forma, las dimensiones y el posicionamiento de los objetos. por lo tanto, es posible conceptualizar la parametrización de la geometría tanto en términos de su representación físico-material (muros, cubiertas, fachadas) como en su calidad de representación espacial (vacíos, perforaciones, espacios habitables, etc.). Las relaciones entre forma y espacio pueden ser exploradas a partir de definiciones concertadas entre sistemas materiales y espaciales de tal modo que se producen importantes consecuencias al diseño como proceso colaborativo y multidisciplinario.

Diseño paramétrico

Según Robert Aish, el diseño trata fundamentalmente del establecimiento de relaciones. Muchas de esas relaciones son de naturaleza geométrica o pueden ser expresadas en términos geométricos⁸. El establecimiento de estas relaciones se fundamenta siempre en la aplicación de algún sistema de reglas y restricciones que provienen tanto del conocimiento como de las intenciones del diseñador con respecto a lo diseñado. Dado que las intenciones siempre emanan de lo que se sabe de un problema, la parametrización de relaciones geométricas

implicaría un modo de representación de reglas que adquiere la forma de un modelo programado computacionalmente⁹.

Aunque toda forma de representación geométrica es un tipo de representación del conocimiento, tanto del problema como de las posibles soluciones de diseño¹⁰, dicho conocimiento se utiliza por lo general de manera implícita detrás de la geometría. Esta situación es característica de los modelos analógicos convencionales y de modelos digitales no paramétricos, donde muchas de las reglas que definen las relaciones geométricas existen solamente en la cabeza del diseñador. En cambio, en un modelo paramétrico, este orden es invertido: Las reglas, y por lo tanto el conocimiento existente acerca de las reglas se vuelve explícito y legible en primera instancia mientras que la geometría puede estar implícita¹¹.

Las consecuencias prácticas de esta inversión son básicamente tres. La primera de ellas es que al hacer explícito el conocimiento de diseño, que antes se encontraba escondido detrás de la representación geométrica del modelo, se le hace accesible para el resto del equipo de diseño. La existencia de conocimiento explícito de reglas de diseño en términos de relaciones paramétricas pone a disposición de otras personas no sólo la problemática modelada, sino también la intencionalidad del diseño y las posibles maneras de implementarla¹².

Un segundo aspecto importante es la posibilidad que las reglas que controlan

la forma provengan de los requerimientos técnicos de alguna especialidad y no sólo del conocimiento e intención del arquitecto, buscando así la optimización de ciertas performances del proyecto. Ejemplos básicos pero ilustrativos de ello son la optimización de la forma de una fachada con el fin de mejorar las condiciones de iluminación o ventilación natural de un edificio. Requerimientos respecto a la dimensión de los elementos estructurales también pueden ser incorporados de manera temprana.

Así el conocimiento acerca de las condiciones ideales de ventilación u otro requerimiento de tipo técnico puede ser examinado y validado por un especialista a través del uso de un software de análisis apropiado. El modelo paramétrico puede ser sometido rápida y eficientemente a un experimento virtual de simulación con el fin de analizar su rendimiento de acuerdo a indicadores de performance previamente definidos¹³ (Augenbroe, 2004). Esto nos lleva a la tercera ventaja del modelamiento paramétrico, que consiste en la posibilidad de generar múltiples variaciones y someter a evaluación diversas alternativas de diseño.

De este modo la toma de decisión se vuelve un proceso mucho más sistematizado, objetivo y colaborativo, a través del análisis comparativo de diferentes soluciones posibles. Por otro lado el repertorio y la capacidad expresiva de las formas diseñadas puede ser ampliada considerablemente por las nuevas posibilidades de experimentación geométrica.

⁸ Aish, R. «Exploring the Analogy that Parametric Design is a Game». En: Oosterhuis, K., Feireiss L. *The architecture co-laboratory: Game set and match II - On computer games, advanced geometries and digital technologies*. Rotterdam: Episode Publishers, 2006.

⁹ La noción de «programado» puede incluir o no la programación con lenguajes computacionales.

¹⁰ Lyon, E. «Architecture and autopoiesis: Notes on design theory, methods, and instruments». *International Journal of Architectural Computing* 2005.

¹¹ Katz, N. «Parametric Modeling in AutoCAD». *AECbytes* (http://aecbytes.com/viewpoint/2007/issue_32.html).

¹² Aish, R. *Op. cit.*

¹³ Augenbroe, G. *Op. cit.*

Replicación, propagación y adaptatividad

Infelizmente los actuales sistemas comerciales BIM ofrecen al diseñador muy pocas libertades de exploración formal en etapas conceptuales así como también de generación de alternativas de diseño durante etapas más avanzadas de anteproyecto. Una de las razones de ello se encuentra en la escasez de métodos que estos productos ofrecen para parametrizar los modelos de acuerdo a las necesidades del usuario.

Por lo general la parametrización se limita a la posibilidad de cambiar valores de componentes preestablecidos en las librerías del *software*. Una serie de elementos convencionales tales como muros, columnas, puertas, ventanas, escaleras, entre otros, se encuentran disponibles para ser insertados en el proyecto, siendo básicamente sus dimensiones las únicas susceptibles de cambio paramétrico.

La «inteligencia» paramétrica de estos componentes proviene de su definición semántica, es decir, de la manera como son estructurados los datos que la describen y el modo que su comportamiento es programado¹⁴. Aunque esta «inteligencia» ha ido aumentando últimamente, al punto que permite obtener cierto grado de adaptatividad automática con relación a cambios efectuados en el diseño, el carácter convencional de las reglas incrustadas en ellos y la dificultad para poder cambiarlas impone serias restricciones a los procesos que buscan diseños más innovadores.

Al obligar al arquitecto a elegir desde un comienzo componentes paramétricos predefinidos los modeladores BIM tienden a forzar una aproximación al diseño a través de estrategias de tipo *bottom-up*¹⁵, propias de la disciplina de la ingeniería. El principal problema con las metodologías de diseño tipo *bottom-up* es que tratan de simplificar el problema ofreciendo soluciones parciales que responden bien a nivel local, pero que carecen de capacidades responsivas frente a cambios globales. De esta manera se restringe seriamente el espectro de posibles soluciones a un problema de arquitectura, induciendo al arquitecto a desarrollar proyectos altamente

estereotipados, al verse obligado a adoptar las convenciones definidas de antemano por el *software*.

Dado que el diseño arquitectónico se apoya generalmente en el uso de grillas o trazados reguladores que controlan la posición y las dimensiones de los elementos sería deseable entonces que los elementos tuviesen mayor «inteligencia» frente a cambios generados a nivel global. Por lo tanto se hace necesario desarrollar herramientas de diseño que enfatizen estrategias de tipo *top-down*, más propias del diseño arquitectónico, compensando los riesgos del modelo *bottom-up* presente en los sistemas BIM actuales.

Caso de estudio: Muro cortina paramétrica

Para resolver las deficiencias antes descritas se ha venido desarrollando tecnologías que buscan ampliar la capacidad de configuración y responsividad de los componentes paramétricos de modo que estrategias de tipo *top-down* y *bottom-up* puedan coexistir. En este sentido la adopción explícita del paradigma de programación orientado a objetos y la no definición de semánticas convencionales destacan como dos de las alternativas más claras para alcanzar estos objetivos. El poder acceder directamente a la estructura de datos de los objetos modelados permite crear componentes altamente «customizados» que puedan responder a tres requerimientos fundamentales:

1. Replicación: Esta es la capacidad de un componente o colección de ellos para ser posicionados en el espacio de manera jerárquica con relación a otros (estrategia *top-down*).
2. Propagación de cambios: Una vez que los componentes son replicados, éstos deben ser responsivos a la propagación de cambios globales efectuados en el nivel del ensamblaje mayor o esqueleto de control (estrategia *top-down*).
3. Adaptatividad: Los componentes debieran exhibir responsividad frente a cambios locales. Satisfacción de condiciones de

adyacencia, coplanaridad, sobreposición, pivote, etc., constituyen típicos requerimientos existentes a nivel local de ensamblajes (estrategia *bottom-up*).

En la búsqueda de un *software* de modelamiento que soportara las tres capacidades antes descritas se estudiaron diversas alternativas. Durante este proceso se contemplaron opciones de modeladores genéricos como *Rhino* y *Maya* así como modeladores paramétricos avanzados como *Catia* o *SolidWorks*. Después de una revisión bibliográfica, del estudio de casos y de consultas a usuarios expertos se llegó a la conclusión que ninguno de ellos ofrece las tres condiciones de manera simultánea y directa. Aunque sin duda el desarrollo de modelos paramétricos a través de programación de *scripts* es desde luego posible en herramientas como *Rhino* o *Maya*, sus estructuras de datos no resultan adecuadas para la representación de un modelo de edificios¹⁶. Por otro lado *software* como *Catia* y su variante arquitectónica *Digital Projects*, aunque soportan las modalidades 2) Propagación de cambios y 3) Adaptatividad de manera eficiente, requieren en cambio de habilidades muy avanzadas de programación para satisfacer el punto 1) Replicación.

Se eligió entonces el modelador paramétrico *GenerativeComponents* (GC)¹⁷. El problema de diseño se trató de un sistema de muro cortina que tuviera cierto grado de variabilidad geométrica de modo que se pudiera experimentar con los tres requerimientos usando estrategias mixtas *top-down* y *bottom-up*.

El muro cortina se definió entonces como un ensamblaje de elementos de madera laminada y perfiles metálicos que debiera componer la fachada principal de un edificio académico de 4 pisos ubicado en un terreno frente al lago Mendota, en la Universidad de Madison en Wisconsin. Este proyecto fue parte de un taller colaborativo a distancia donde estudiantes de posgrado de arquitectura, construcción e ingeniería civil debían producir proyectos sustentables de manera concurrente. Para ello se desarrolló un modelo BIM que sirvió de base para todas las tareas de análisis y evaluación desde etapas tempranas del diseño.

¹⁴ Guzdial, M., Ericson, B. *Problem solving with data structures: A multimedia approach*. College of Computing / GVI Georgia Institute of Technology, 2007.

¹⁵ Para una revisión más detallada acerca de estrategias de diseño *bottom-up* y *top-down* ver Zeid I., 2005.

¹⁶ Cabe mencionar que estos modeladores no soportan de manera consistente la representación de sólidos (*Boundary Representation* o *B-Reps*), tecnología considerada clave para una descripción más apropiada de un modelo de edificio. Su principal finalidad es la visualización de geometrías, ya sea para la industria del cine y del video juego. Por esta razón no poseen de manera nativa tecnologías de base de datos ni comparten el principio de interoperatividad.

¹⁷ GC es un sistema de diseño paramétrico y asociativo desarrollado por Robert Aish, director de investigación de Bentley Systems. Se trata de un nivel complementario a la plataforma *Microstation*.

Así, las condiciones climáticas, de viento e iluminación, así como la topografía del lugar fueron tomadas en cuenta desde el comienzo para la propuesta. Con la idea de alcanzar la certificación LEED en la categoría «Silver» se eligió por una estructura completa de madera, dado que se trata de un material local que implica bajos niveles de energía para su producción y transporte (*embodied energy*). Un papel clave dentro de la filosofía de diseño adoptada lo tendría la fachada principal, alineada según modelos de simulación desarrollados en *EcoTect* con las condiciones más favorables de ventilación e iluminación natural (Figura 2).

En la etapa conceptual de su desarrollo se la definió como una piel translúcida que debía integrar visualmente el edificio con el paisaje boscoso del terreno. Así, ideas de ramificación arbórea, profundidad, luz y sombra fueron tomadas como imágenes conceptuales que debían apoyar la función de un elemento destinado a maximizar el confort térmico y lumínico interior.

Para ello se estudiaron sistemas constructivo-estructurales basado tanto en módulo discretos que pudiesen ofrecer un alto grado de prefabricación como sistemas de líneas continuas que ofreciesen mayor estabilidad estructural. Posteriormente se procedió a identificar los principios o patrones geométricos de articulación entre elementos, centrándonos en las reglas y restricciones necesarias para la parametrización de estas relaciones¹⁸. Mientras que en el sistema continuo la atención se centró exclusivamente en las fijaciones ubicadas en las intersecciones, en el sistema discreto los esfuerzos se concentraron en la parametrización del módulo en sí (Figuras 3 y 4).

La construcción de los componentes en el nivel rótula (intersección) o de módulo claramente obedece a una estrategia *bottom-up* de suma de subensamblajes menores para conformar un ensamblaje mayor. Pero aquí, a diferencia de los componentes típicamente disponibles en un modelador BIM, su configuración fue hecha completamente por los diseñadores de acuerdo a las necesidades surgidas durante el proceso, y no como imposición externa. De esta forma los requerimientos de responsividad local son satisfechos (adaptatividad).

Una vez insertos en el esqueleto que define la superficie del muro cortina (superficie B-Spline

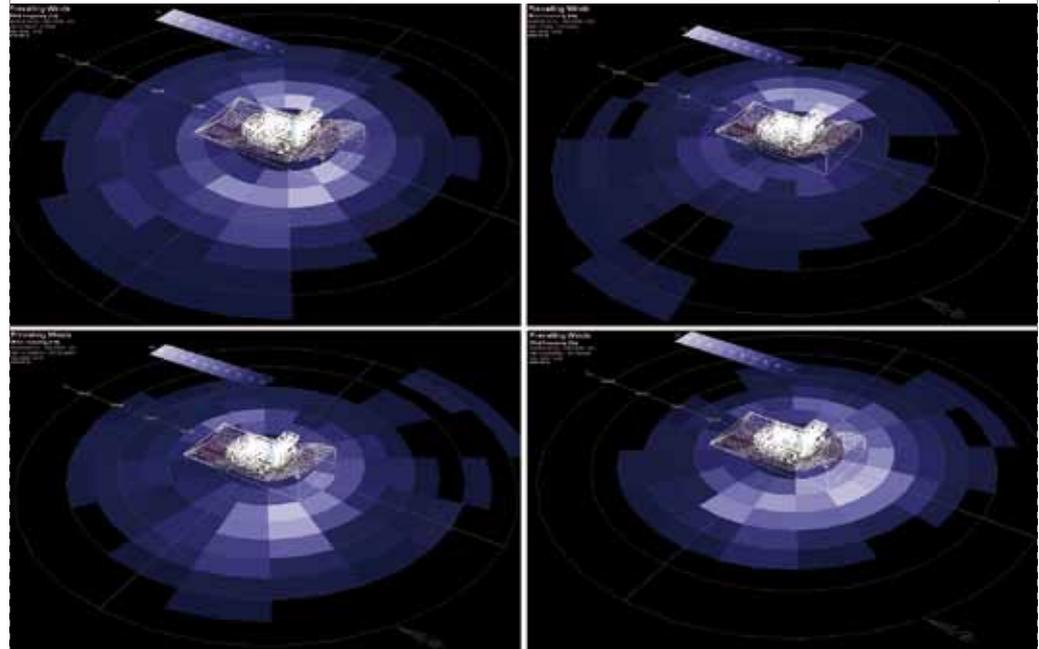


Figura 2: Modelo de análisis de vientos predominantes en las cuatro estaciones sobre sitio ubicado en Madison, Wisconsin, EE.UU. Proyecto colaborativo desarrollado a distancia por Jordan Read (constructor civil), Susanne Nikulla (ingeniero estructural), Jixin Zhang (ingeniero estructural), Eliel de la Cruz (arquitecto) y Andrés Cavieres (arquitecto). Departamento de Ingeniería Estructural, Universidad de Stanford.

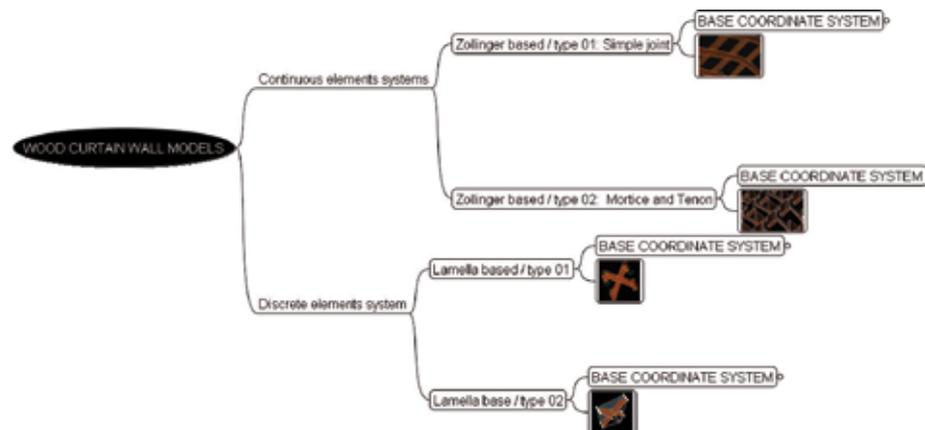


Figura 3: Diseño paramétrico de muro cortina en madera laminada. Elección de la configuración constructiva-estructural entre un sistema de elementos continuos (Zollinger) y un sistema basado en modulo discretos (Lamella).

¹⁸ Woodbury, K., Aish. *Some patterns for parametric modeling*. Bentley Systems, 2005.

de grado 2), dichos componentes responden automáticamente a cambios globales del mismo (por ejemplo la deformación a partir de la manipulación de su polígono de control a la manera de un títere). La jerarquía de relaciones que se logra entre componentes y

subcomponentes producto de la aplicación combinada de estrategias *top-down* y *bottom-up* se representa claramente en la forma de un grafo de dependencias que el *software* construye automáticamente mientras el diseñador modela.

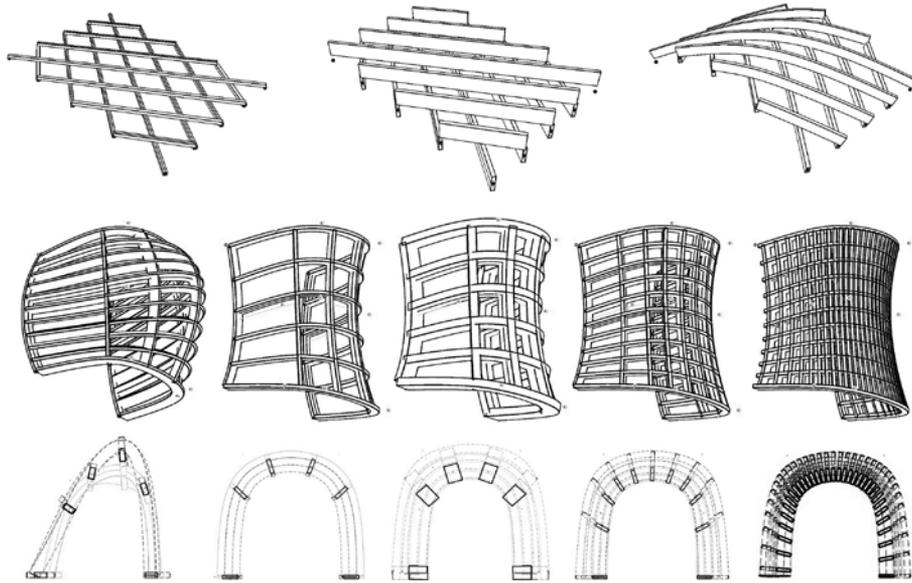


Figura 4: Modelos de estudio desarrollados en *GenerativeComponents*. Perfiles continuos se mantienen perpendiculares a la dirección principal de la fachada. La atención se centra en los componentes de unión entre perfiles.

Aunque esta capacidad de GC para soportar un control *top-down* sin tener que recurrir a la programación dura es única entre los modeladores paramétricos analizados, el potencial máximo de flexibilidad y responsividad de los componentes se logra a través de una interfaz de programación que da acceso directo a la estructura de datos. De esta forma GC es capaz de representar las geometrías modeladas y las relaciones incrustadas entre ellas de tres formas diferentes (Figuras 5 a 8).

Trabajo futuro

Aunque *GenerativeComponents* se encuentra aún en etapa de desarrollo, creemos que reúne características muy interesantes que pueden guiar el desarrollo de futuras herramientas de diseño. La posibilidad de modelar con acceso inmediato a la estructura de datos de los objetos, y poder contar con diversas formas de representación / manipulación de las relaciones constituyen sin duda un aporte a la cognición del proceso, a la implementación de métodos para la optimización de la *performance* y la reusabilidad de las soluciones encontradas.

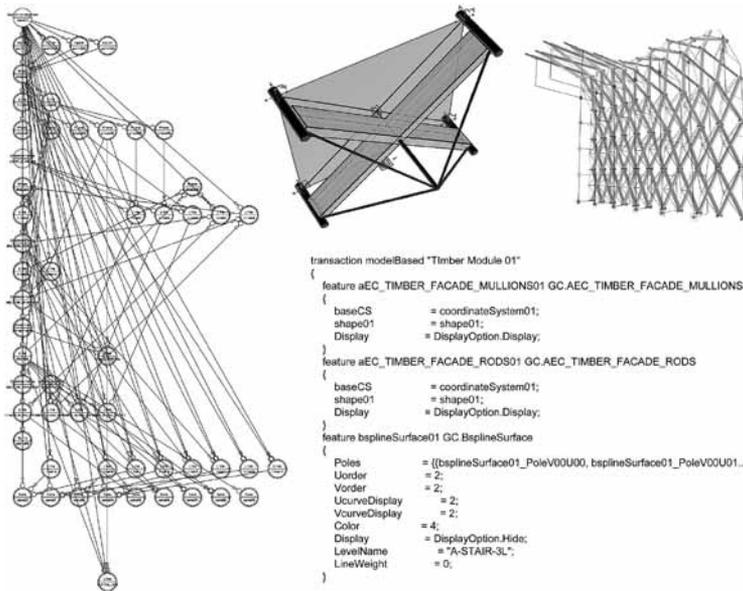


Figura 5: Definición de componente modular que cumple con los tres requisitos. Comportamiento responsivo logrado en base a la utilización de métodos disponibles en la estructura de datos de los subcomponentes geométricos (punto, línea, plano y dirección). En la figura se denota las tres formas simultáneas que GC permite la representación de relaciones: grafo de dependencias dirigido acíclico, modelo geométrico típico y *scripting*.

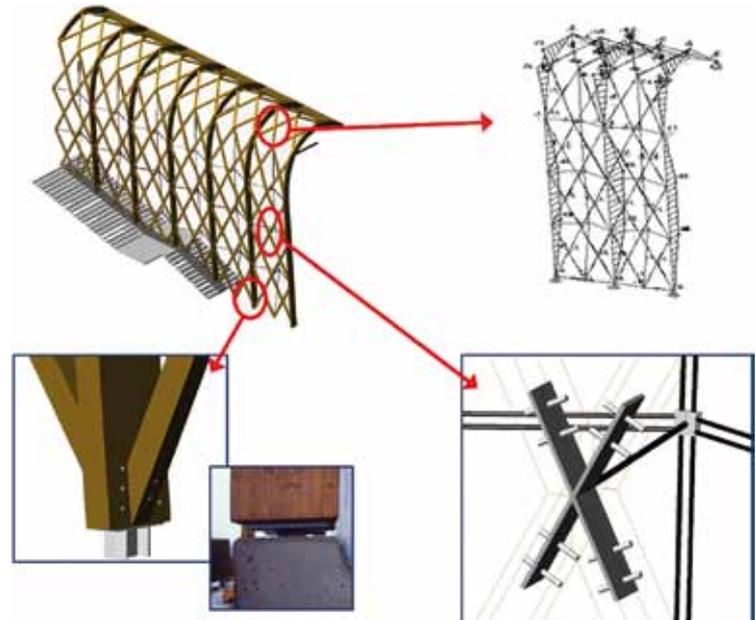


Figura 6: Análisis estructural de muro cortina en madera laminada efectuado en *Rstab3D*. Momentos y presión de viento (Susanne Nikulla).

Uno de los desafíos futuros de esta investigación será la implementación de reglas orientadas a la optimización de componentes arquitectónicos según indicadores de performance y su relación con procesos de manufactura digital. La incorporación a través de la programación de requerimientos técnicos provenientes de otras especialidades, la generación de alternativas y su validación por medio de herramientas de análisis constituyen las principales etapas a seguir.

Comentarios finales

Consideramos que las herramientas computacionales son solamente medios. Sin embargo su relevancia estriba precisamente en este hecho. Según Lyon¹⁹, solo en la apropiada comprensión del rol fundamental que poseen los medios convertidos en instrumentos cognitivos yace la clave del diseño asistido por computador. La estructura de datos y procesos computacionales subyacentes mencionados por Frazer²⁰ ofrecen la oportunidad no sólo de hacer explícito el conocimiento de las reglas que controlan el diseño, sino que también permite analizar y validar dicho conocimiento.

Por otro lado, en la medida que aumente la preocupación por la situación medio ambiental y crezca la conciencia de responsabilidad que le cabe a la industria AIC, es muy probable que soluciones intuitivas y basadas en la experiencia de los diseñadores no sean más suficientes para dar respuestas adecuadas a la complejidad de los parámetros en juego.

En la importancia de implementar un nuevo grado de control y predictibilidad del comportamiento futuro del diseño yace la oportunidad de recuperar la posición de liderazgo del arquitecto dentro de los procesos de toma de decisión. Dado que la complejidad que implica el nuevo escenario orientado a la *performance* sólo puede ser solventado de forma multidisciplinaria, la objetividad técnica y científica del discurso arquitectónico se vuelve la condición fundamental para el trabajo colaborativo efectivo. Esta condición se vuelve particularmente cierta cuando la colaboración se ve mediada por instrumentos de diseño y análisis sobre los que deben regir lenguajes, metodologías y protocolos comunes.

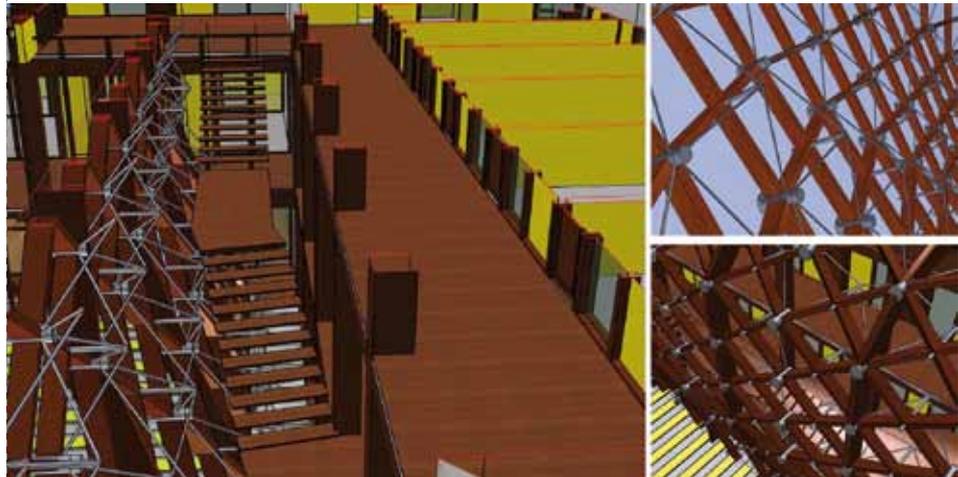


Figura 7: Corte perspectivado de modelo BIM. Solución preliminar de módulo propuesto: Ensamblaje de piezas de madera laminada y tensores de acero (Read, Nikulla, Zhang, de la Cruz y Cavieres).

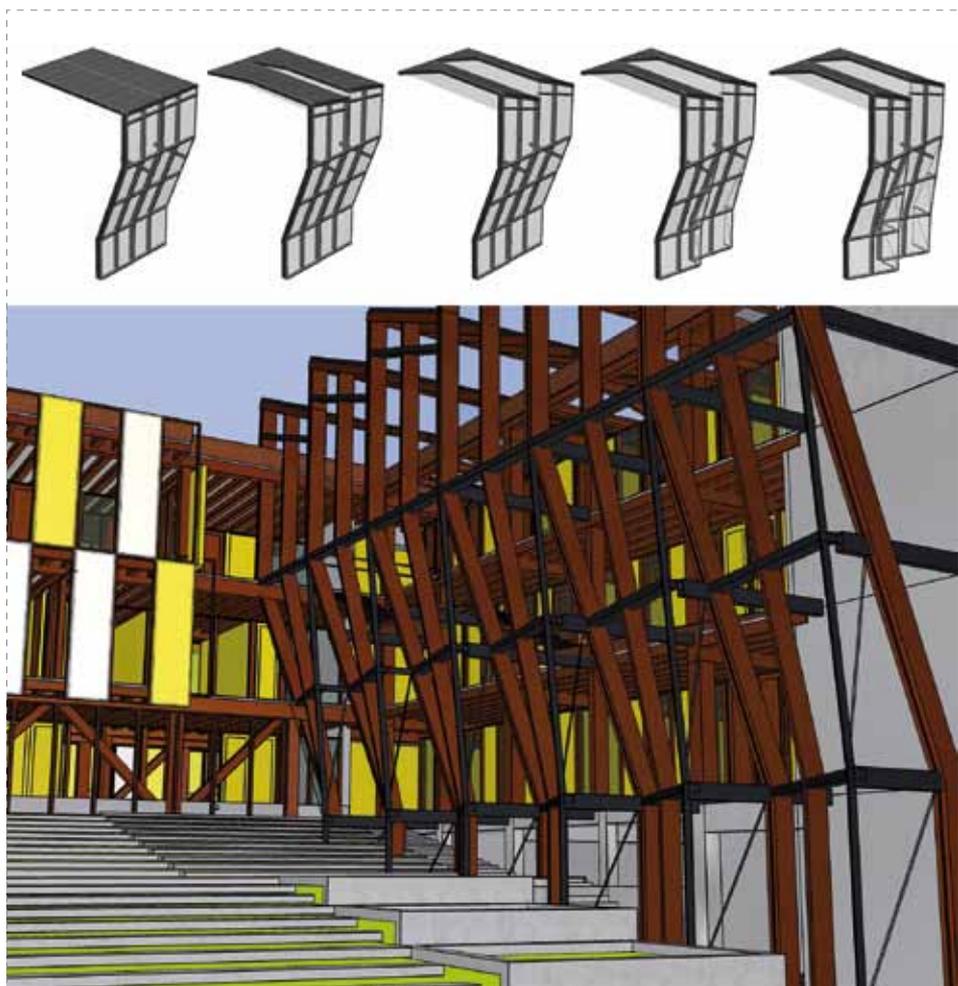


Figura 8: Implementación *top-down*. Pliegues en la superficie de control como solución para mejorar las condiciones de ventilación e iluminación natural. Los subcomponentes lineales se adaptan automáticamente desde los cambios producidos en el nivel superior (Read, Nikulla, Zhang, de la Cruz y Cavieres).

¹⁹ Lyon, E. *Op. cit.*

²⁰ Frazer, J. *Op. cit.*