

Tecnologías disruptivas: programación y fabricación en Latinoamérica

Disruptive Technologies: Scripting and Digital Fabrication in Latin America

Pablo C. Herrera

Universidad Peruana de Ciencias Aplicadas, Perú

✉ pablo@espaciosdigitales.org

<http://espaciosdigitales.org/>

ABSTRACT

Since 2008 the preference for using different programming methods (Rhinoscript) had been analyzed using blogs. Searching for answers to explain the negative tendency of this year (from 48,063 to 16,332), a second repository was created (Grasshopper) featuring interactive methods and techniques. It has been discovered that of the five geographic regions analyzed Latin America is the only one that preferred the interactive interface (18% over programming). This shows that we are still keeping a strong dependency on the use of stable and safe technologies over disruptive ones that proved to be more efficient in design and fabrication.

KEYWORDS: digital fabrication, scripting, architectural education, rhinoscripting, grasshopper

En nuestra vida social y cultura contemporánea empleamos como medio de aprendizaje, descubrimiento y creación el *software* y el computador. En conjunto, un metamedio que contiene medios existentes y por existir (Manovich, 2008); pero mientras los usamos, ambos llegan rápidamente a ser obsoletos, debido a la constante actualización entre una versión y otra (Stevens, 1997). Bajo el mismo patrón de evolución, y durante casi diez años, las reglas y convenciones de un *software* entre plataformas DOS y Windows 7 (ambas del mismo fabricante) fueron transformadas y presentadas a través de una interfaz que nos obliga a depender de los cambios y actualizaciones sin pensar en otras posibilidades que solo adaptarse.

Dentro de los procesos estables de implementación, Terzidis (2003) identificó dos conceptos: *computerization* y *computation*. El primero se refiere a la automatización, mecanización y digitación de acciones manuales y electrónicas (Manovich, 2008). *Computation*, sin embargo, se refiere a una exploración indeterminada y que está en continuo proceso. Terzidis sostiene que *computation* prolonga el conocimiento humano, porque se trata de racionalización, razonamiento, lógica, algoritmo, deducción, inducción, extrapolación, exploración y estimación.

En sus tantas implicancias involucra la estructura mental, conocimiento, solución de problemas y reglas basadas en la inteligencia. Acciones que difícilmente se implementarían con

una interfaz interactiva, porque ella utiliza reglas propuestas por un programador. Al contrario, como evidenciaron Terzidis (2003), Herrera (2009) y Bueno (2008), *computation* puede ser explorada usando la programación con usuario final (comúnmente llamado guiones o *scripts*).

De manera general, Terzidis (2003) también sostuvo que mantenemos un periodo de *computerization*, el mismo que atravesaron las modernas técnicas vanguardistas de nuestra cultura (Manovich, 2008). Acciones que hemos realizado suponiendo que son la única posibilidad para usar un *software* (Kesson, 1995), porque los fabricantes nos presentan una interfaz muy amigable y a la cual nos hemos acostumbrado a usar o esperamos que llegue cada año.

En esta investigación se pone en perspectiva la experiencia digital latinoamericana de los últimos cuatro años bajo el concepto de *disrupción*, buscando patrones de preferencia para el tipo de tecnología que se ha usado en talleres de arquitectura latinoamericanos. Del conjunto de aplicaciones informáticas existentes, se eligió *Rhinoceros* (en adelante Rhino), fabricado por McNeel, que delimita el estudio a técnicas propias de la *computerization* (representada por la interfaz de un *software* interactivo) y de la *computation* (que utiliza la programación del tipo VBScript en su modalidad *Rhinoscripting* [RS]), ambas incluidas en Rhino. El objetivo es encontrar evidencia que

permita comparar preferencias de uso entre un proceso y otro bajo el concepto de disrupción.

Este último es el efecto inesperado por el que una tecnología desplaza a otra en menor tiempo y con una mejor *performance* (Christensen, Horn y Johnson, 2008). El uso de la interfaz interactiva supone aprender convenciones de representación que han dominado la presentación de una idea bajo las mismas reglas unidireccionales de dibujo de hace alrededor de 5.000 años que, utilizando un *software*, siguen representando una posibilidad solucionar un problema por vez y esto es lo que hace Rhino al usar técnicas basadas en íconos.

La documentación de casos de *computerization* en talleres de arquitectura y diseño se ha presentado desde las primeras conferencias de Sigradi (1997), pero experiencias tan lejanas en el tiempo no representan la estabilidad de una tecnología, sino el inicio de ella. Sigradi realizó su primera publicación temática con la selección de 21 experiencias pedagógicas en universidades de la región, bajo el criterio de innovadoras en talleres de arquitectura y diseño (Rodríguez, 2006). En ese contexto, cada autor tuvo la posibilidad de actualizar y ampliar su experiencia de acuerdo con nuevas investigaciones. Estas revelaron que el uso de convenciones y técnicas para representar y diseñar con una interfaz eran estables, y del total, ninguno de los talleres proyectuales priorizó la programación como estrategia.

En general, con la implementación de la programación se identifican combinaciones de variables y se evidencia una amplitud de posibilidades, así como tantos parámetros y criterios pensemos en utilizar, especialmente cuando tenemos referentes biológicos o complejos.

De este modo, al empezar la implementación de RS en el 2006 en Latinoamérica, se propuso que los participantes solucionaran un problema de diseño pensando cómo actuarían sin utilizar un *software*, es decir, no pensar en técnicas, sino en identificar soluciones sin dependencia directamente del *software*. En todas las experiencias se detectó una fuerte dependencia de la interfaz interactiva, que limitaba incluso el

recordar reglas básicas de geometría (Herrera, 2008). Esto era un claro efecto de *computerization* y la automatización de tareas. Sin embargo, en las conclusiones de todos los talleres, el uso de RS se impuso sobre la interfaz, porque les permitía encontrar más alternativas, una vez definido el diseño que se iba a explorar.

La confrontación entre el medio interactivo y la programación demostró que cada una contiene técnicas que exigen diferentes grados de racionalidad. Al utilizar una interfaz interactiva, estamos limitados a una solución, sin que podamos modificar la secuencia de la técnica o del proceso, lo cual limita más aún los cambios, si el modelo no se rige por reglas paramétricas. Por el contrario, con la programación tenemos que pensar en matematizar y geometrizar el problema y regresar a lo básico de la arquitectura.

Con sus tantas ventajas, la programación a nivel de usuario final no ha logrado ser disruptiva en arquitectura a lo largo de su historia. Arnold (1984) explicó que antes de la interfaz la programación era parte de las oficinas de arquitectura, donde el fabricante de *software* escribía rutinas específicas según el problema por resolver, utilizando principalmente *scripts*. De esta manera, el *software* de comandos y luego interactivo desplazó la programación en la década de 1980.

A principios del siglo XXI (Fig. 1), la programación reapareció en investigaciones de maestría y doctorado en MIT y Harvard, principalmente, alentadas primero por William J. Mitchell y continuadas por Lawrence Sass y Kostas Terzidis, enfocadas en resolver problemas de fabricación (panelización, ensamble, fractura de modelo, racionalización, seccionado, teseleado, entre otras). A falta de máquinas de fabricación en escuelas de Latinoamérica, la implementación de programación a nivel de usuario final se limitó a abordar la solución a problemas de formas complejas.

En esta investigación se sostiene que los talleres de RS implementados en Latinoamérica (Herrera, 2007 y 2008) se propusieron como una experiencia innovadora en el mismo contexto que aparece la publicación de Sigradi (Rodríguez, 2006). Este

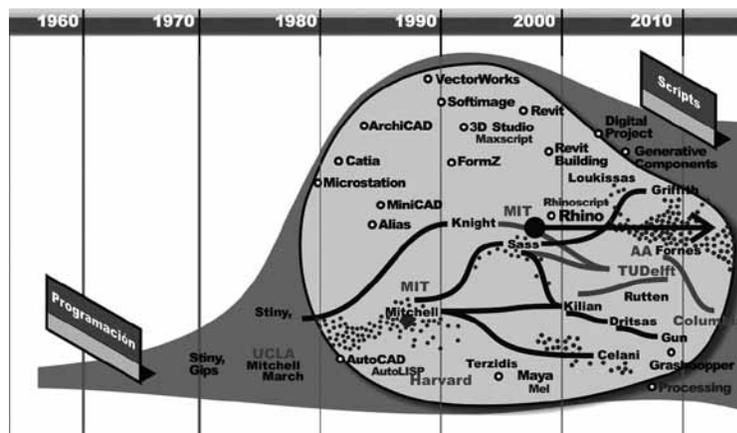


Figura 1. Histórico comparativo del software interactivo y la programación

proyecto representó una experiencia pedagógica que la llevó incluso a ser parte de la III Architectural Beijing Biennale, en el 2008. A pesar de estos resultados, hoy se detecta que esta implementación priorizó la tecnología sobre el individuo. Esta es una falla que Christensen (2008) detectó cuando una tecnología aparecía como disruptiva, pues demuestra que bajo esa prioridad la implementación fracasaba y seguía su ritmo estable, es decir, el aprendizaje bajo patrones pedagógicos conocidos.

Según Senske (2005), la velocidad de implementación tecnológica superó a la pedagógica, y para facilitar el autoaprendizaje se propuso mantener un conjunto de recursos y métodos a través de un blog, para facilitar el acceso a nueva información sobre patrones, usos y técnicas que iban apareciendo (Herrera, 2009).

Entre el 2008 y el 2010, se registraron en el blog de RS (<http://rhinoscriptingresources.blogspot.com/>) un total de 102.500 visitas de diferentes regiones del mundo. Los resultados de preferencias fueron variables y se agruparon en tres categorías: (1) aquellos con una preferencia ascendente (Estados Unidos, Canadá, Centroamérica y Asia), (2) aquellos con una preferencia descendente (Europa y Suramérica) y (3) aquellos con una preferencia estable (Fig. 2).

Entre el 2008 y el 2009, el total mundial de visitas tuvo un crecimiento del 2,77%; mientras del 2009 al 2010 se observó un déficit del -60,44%, ello a pesar de que la cantidad de blogueros entre el 2006 y el 2009 sumaron casi 50 y produjeron más de 250 enlaces a más de 20 métodos de RS (Herrera, 2009).

Rhino convive con diferentes técnicas para abordar un problema de representación, diseño o fabricación. *Explicit History* (2007) es una de ellas. Desarrollado por David Rutten, se incluyó en el 2008 en Rhino (versión 4.0) con el nombre de Grasshopper (GH), para facilitar la construcción de relaciones lógicas y matemáticas en función de un árbol que asocia variables hacia componentes, parametrizando el problema de diseño y facilitando a los usuarios un mayor control sobre cada objeto, todo ello interactivamente.

Así es como Rhino resume tres maneras de resolver un problema: (1) utilizando la interfaz de usuario con base en técnicas interactivas para dibujar una propuesta, (2) utilizar componentes y variables para controlar la historia del objeto con GH y (3) escribir línea por línea el problema, utilizando un vocabulario y una gramática a través de RS. Esto hace a Rhino particularmente potente, porque en general los fabricantes potencian una o dos técnicas por aplicación, no las tres al mismo tiempo.

Luego de implementar una lista de recursos en internet para GH (<http://grasshopperresources.blogspot.com/>), se comparó con el blog de RS y ello permitió valorar el conjunto de preferencias entre ambas técnicas nuevamente por área geográfica. Para el caso de Latinoamérica, se observa una disminución en la preferencia de RS entre el 2008 y el 2009 (Fig. 3).

Al comparar el universo de visitas entre RS y GH, se aprecia que del total mundial, Latinoamérica (RS: 6,7%; GH: 24,7%) marca una preferencia en el uso de GH del 18% sobre RS, y Estados Unidos/Canadá, una situación inversa (RS: 27,8%; GH: 11,2%). Así mismo, hay grupos que mantienen una preferencia equilibrada entre ambas técnicas: Europa (RS: 43,3%; GH: 41%), Asia (RS: 16,6%; GH: 15,3%) y Australia (RS: 2,8%; GH: 2%). Sin embargo, este resultado se puede interpretar de otra manera si tomamos en cuenta la cantidad de visitas para el mismo lapso. El promedio de visitas mensuales al blog de RS fue de 1.332, y de 346, para GH.

Aunque los resultados geográficos muestran una preferencia por GH, la cantidad de visitas anuales fue de solo el 25% del promedio anual de usuarios de RS, que en 12 meses tuvo un total de 16.332 visitas sobre los 4.059 de GH. Este no es un fenómeno aislado. Si se revisan los archivos del blog de Marc Fornes (<http://tvmny.blogspot.com/>), que sirve de portafolio de trabajos y talleres hechos con RS, entre el 2008 y el 2009 tuvo déficit de -5,83% y del 2009 al 2010, de -10,77%, pero con un promedio de alrededor de 140.000 visitas anuales.

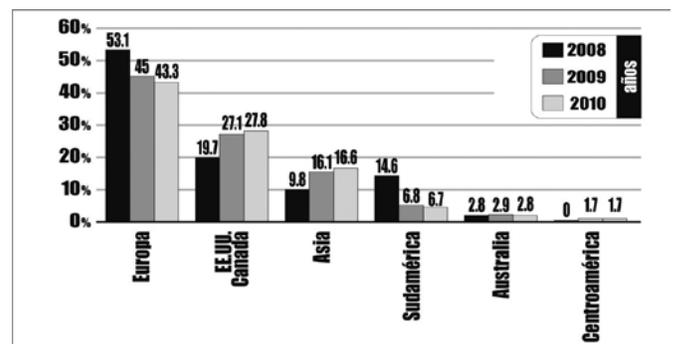


Figura 2. Usuarios de métodos de Rhinoscripting por área geográfica (2008-2010)

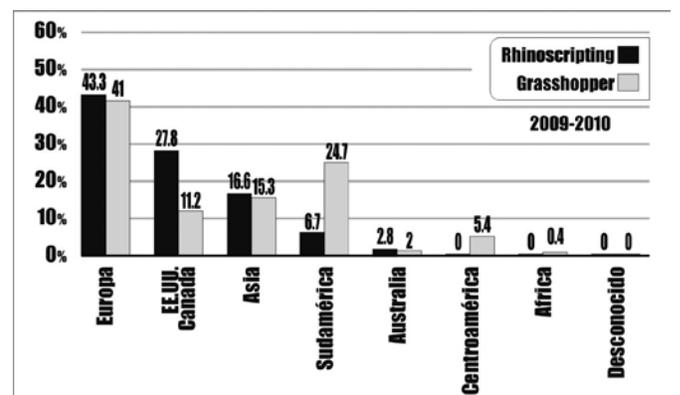


Figura 3. Comparativo de visitas entre Grasshopper y Rhinoscripting

Conclusiones

Chermayeff y Alexander sostuvieron que “la máquina es un complemento y no un sustituto del talento creativo, que mientras esta no pueda inventar, puede explorar relaciones muy rápida y sistemáticamente de acuerdo a reglas preestablecidas, funcionando como una extensión natural de la habilidad analítica del hombre” (1963, p. 166). La formalidad de procesos de implementación en arquitectura no tiene el mismo ritmo que el del hemisferio norte del planeta; ninguna máquina para fabricar estuvo presente de manera sistematizada en el proceso de esta implementación, principalmente por presupuesto o porque en las universidades donde se realizó la experiencia recién evaluaban la adquisición de un laboratorio de fabricación digital y el proyecto se convirtió en un taller de exploración de forma y espacio, lo cual no permitió la continuidad del trabajo con programación o *scripts* de manera sostenible para el total de alumnos que participaron en estos talleres.

Aquellos alumnos que siguen utilizando técnicas de programación lo hacen porque integraron máquinas después de los talleres. Aunque existen en Latinoamérica universidades con maquinaria para este proceso, se prefiere mantener el aprendizaje de técnicas interactivas sobre la programación y se elige la estabilidad sobre lo disruptivo. Luego de concluir que la programación es una aliada importante en el proceso de diseño, la siguiente implementación potenciará la personalización de estas técnicas para fabricar, más que solo visualizar o representar el proyecto.

Referencias

- Arnold, E. (1984). *Computer aided design in Europe*. Brighton: Sussex European Research Centre.
- Bueno, E. (2008). *El scripting como estrategia de diseño: una experiencia pedagógica*. Documento procedente de Sigradi, La Habana, Cuba. Recuperado el 12 de julio de 2010, de http://cumincades.scix.net/data/works/att/sigradi2008_100.content.pdf.
- Chermayeff, S. y Alexander, C. (1963). *Community and privacy: Toward a new architecture of humanism*. New York: Doubleday.
- Christensen, C., Horn, M. y Johnson, C. (2008). *Disrupting class: How disruptive innovation will change the way the world learns*. New York: McGraw Hill.
- Herrera, P. (2007). *Solución de problemas relacionados al diseño de superficies complejas: experiencia de programación en la educación del arquitecto*. Documento procedente de Sigradi, México. Recuperado el 12 de mayo de 2010, de http://cumincades.scix.net/data/works/att/sigradi2007_af15.content.pdf.
- Herrera, P. (2008). *VBScript en la educación del arquitecto: estrategias y métodos durante y después de la implementación*. Documento procedente de Sigradi, La Habana, Cuba. Recuperado el 12 de julio de 2010, de http://cumincades.scix.net/data/works/att/sigradi2008_165.content.pdf.
- _____(2009). *Patrones y convenciones en el uso de rhinoscripting*. Documento procedente de Sigradi, São Paulo, Brasil. Recuperado el 11 de junio de 2010, de http://cumincades.scix.net/data/works/att/sigradi2009_1085.content.pdf.
- Kesson, M. (1995). *An introduction to 3D computer graphics*. Recuperado el 15 de enero de 2006, de http://old.siggraph.org/education/materials/renderman/pdf_tutorial/rman_booklet.pdf.
- Manovich, L. (2008). *Software takes command*. Recuperado el 14 de mayo de 2009, de <http://softwarestudies.com/softbook>.
- Rodríguez, D. (2006). Sobre usos, prácticas y estrategias en talleres de arquitectura y diseño en entornos virtuales: reseña de casos en la región latinoamericana de los últimos 10 años y tendencias posibles. En D. Rodríguez (Ed.), *Experiencia digital* (pp. 23-33). Buenos Aires: Editorial de la Universidad del Mar del Plata.
- Senske, N. (2005). *Fear of code: An approach to integrating computation with architectural design*. Recuperado el 26 de enero de 2007, de <http://dspace.mit.edu/bitstream/handle/1721.1/33032/62100250-MIT.pdf?sequence=2>.
- Stevens, G. (1997, September). Reflections of an Apostate CAD Teacher. *JAE*, 51 (1), 78-80.
- Terzidis, K. (2003). *Expressive form*. Londres: Spon Press.