

---

# Estructura d'un disseny generatiu

---

PID\_00267103

David Casacuberta

---

Temps mínim de dedicació recomanat: 2 hores

---



**David Casacuberta**

Com a professor de Filosofia de la ciència a la Universitat Autònoma de Barcelona (UAB), la seva línia de recerca actual són els impactes socials i cognitius de les TIC, tema sobre el qual ha publicat diversos llibres i articles.

Actualment és membre del Grup de Treball d'Ètica, Seguretat i Regulació de Bioinformàtica Barcelona i investigador del Grup d'Estudis Humanístics en Ciència i Tecnologia (GEHUCT). També és codirector del màster de Disseny i direcció de projectes per a Internet d'Elisava i participa com a professor en diversos postgraus de gestió cultural, teoria de l'art contemporani i disseny de tecnologies digitals.

Ha rebut el premi Eusebi Colomer de la Fundació Epsom al millor assaig sobre els aspectes socials, antropològics, filosòfics o ètics relacionats amb la nova societat tecnològica amb el seu llibre *Creació col·lectiva*. També ha guanyat el premi Ingenio 400, organitzat pel Ministeri de Cultura i la Societat Estatal de Comemoracions Culturals, al millor projecte de net.art amb la seva obra *X-Reloaded* (en col·laboració amb Marco Bellonzi).

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Quelic Berga (2019)

Primera edició: setembre 2019

Autoria: David Casacuberta

Llicència CC BY-NC-ND d'aquesta edició, FUOC, 2019

Av. Tibidabo, 39-43, 08035 Barcelona

Realització editorial: FUOC



Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>

# Índex

<b>1. Pensar el teu projecte de disseny generatiu.....</b>	<b>5</b>
<b>2. L'ontologia d'un disseny generatiu.....</b>	<b>8</b>
2.1. Fix enfront de variable .....	8
2.2. Determinista i seqüencial .....	9
2.3. Seqüencialitat enfront d'aleatorietat .....	11
<b>3. Dades externes.....</b>	<b>13</b>
3.1. Bases de dades .....	13
3.2. Processament de dades i graus de generalització .....	14
3.3. Interactivitat i personalització .....	15
3.4. Dades en viu. Animacions .....	16
<b>4. Hibridació.....</b>	<b>18</b>
<b>Bibliografia.....</b>	<b>19</b>



## 1. Pensar el teu projecte de disseny generatiu

Tal com hem vist en el mòdul «En què consisteix el disseny generatiu», hi ha moltes formes de desenvolupar un disseny generatiu. Ni tan sols estem obligats a utilitzar tecnologies digitals per a dur-lo a terme. Disposem de diversos paradigmes que hem apuntat en l'esmentat mòdul: ser un grafista computacional com Maeda, utilitzar algorismes de *deep learning* com les *Generative Adversarial Networks* o bé algorismes genètics en els quals juguem a ser la naturalesa i decidir quin disseny serà el que finalment sobrevisqui.

És important que entenguem el canvi conceptual que implica moure'ns en el disseny generatiu. En fer-ho, passem de veure el disseny gràfic com un procés manual, d'artesanía, a un procés intel·lectual, de conceptualitzar un problema, desenvolupar un model, executar-lo, valorar-lo, revisar les nostres apreciacions, modificar el model i tornar-lo a provar.

Si ens llancem directament a l'editor de codi i comencem a col·locar-hi instruccions «a veure què passa», els resultats seran probablement decebedors i inútils en un projecte seriós. Abans de programar és necessari planificar molt bé on volem arribar i quins procediments utilitzarem per a arribar a aquests objectius. Parafraçant Humpty Dumpty d'*Alícia a través del mirall*: només quan no t'importi on vols arribar tant et serà el camí que triïs.

Aquesta planificació ha d'organitzar-se des de tres fases diferenciades:

- En una primera fase analitzarem el problema que volem resoldre, ja sigui un objectiu propi o l'encàrrec d'un client. Descriurem els diferents condicionants associats al projecte i com interactuen entre si, establint quines interaccions ajuden a resoldre el problema, quines el dificulten i quines semblen irrelevantes per als nostres objectius. Una vegada entès el context, procedirem a buscar una solució factible al problema plantejat a partir de la informació que hem recopilat.
- En una segona fase definirem una estructura formal que caracteritzi la solució generativa que estem buscant, especificant els diferents objectes gràfics que formaran part de la solució i les regles que els connecten entre si a l'estil de l'exemple que vam veure a l'apartat 1.2. del mòdul «En què consisteix el disseny generatiu».
- En una tercera fase testarem el disseny gràfic que hem ideat a l'apartat anterior, per veure si aconsegueix els efectes esperats o no. Si veiem que encara queden coses pendents que no funcionen, tornarem a la segona o a la primera fase, depenent de si la falta d'adequació és una qüestió de l'estructura del nostre disseny generatiu (i llavors tornariem a la fase 2) o

si no hem tingut en compte condicionants rellevants (i llavors hauríem de tornar a la fase 1). Quan sentim que la solució realment és eficient i resol el problema que volíem solucionar, farem els últims ajustos al nostre model i en farem la implementació gràfica final. Aquest tipus de procés rep el nom d'iteració.

Tal com argumenten autors com Nigel Cross, en una acció dissenyada, problema i solució estan íntimament connectats. És el que en la literatura de teoria del disseny es descriu com un *wicked problem*: un problema que en realitat no està ben definit, i en el qual, inevitablement, en el procés d'oferir una solució el problema es reinterpreta i es reescriu, fins a crear-se un interessant bucle d'interconnexions entre problema i solució.

Dit d'una altra forma, quan dissenyem no hem de pensar que hi ha una solució única que és la correcta. En el procés de buscar la solució, examinarem el problema des d'altres perspectives, i és fàcil que ens replantegem el problema original i li fem la volta. En replantejar el problema, haurem de tornar a pensar la solució. I el procés pot anar-se repetint...

Una enginyera s'enfronta normalment a problemes ben definits del tipus «construeix un mecanisme de seguretat per a un ascensor que sigui capaç de resistir tantes tones de pes i faci que l'ascensor baixi a una velocitat màxima de tants metres per minut». Gràcies a això pot oferir una solució factible que s'adapta científicament a tots els constreyniments. En canvi, un problema en disseny gràfic del tipus «Dissenya un cartell que faci que la gent tingui interès a comprar el meu llibre» és molt més obert: haurem de saber de què va el llibre, a quins públics va dirigit, on es pensa distribuir els cartells, etc. Cada possible solució gràfica que proposem tindrà més efecte en un entorn que un altre, i cridarà més l'atenció a un col·lectiu que a un altre, per la qual cosa segons avança el procés de disseny la pregunta s'anirà transformant. Si decidim fer un disseny generatiu, el nostre cartell probablement atregui més gent jove, curiosa, que busca coses noves, de manera que la pregunta original es transformarà i, en lloc de preguntar-nos en general sobre com generar interès en «la gent», tindrem possibles nous problemes com «dissenya un cartell generatiu que cridi l'atenció a la gent jove que busca novetats però al mateix temps no desanimi persones de gustos més clàssics». Però també puc generar un altre tipus de situació com «dissenya un cartell generatiu que resulti tan cridaner i xocant que generi un gran interès entre el públic més radical encara que provoqui rebuig en persones de gustos clàssics».

En aquest mòdul ens centrarem en la segona fase d'aquest procés: com cal pensar l'estructura formal d'un projecte generatiu, previ a implementar-lo en un llenguatge o entorn de programació específics. Les altres dues fases queden fora dels objectius d'aquest mòdul.

De fet, la primera fase se centra en els mecanismes creatius del professional del grafisme, i seria complex intentar desenvolupar una teoria sistemàtica sobre com analitzar el problema i generar una solució. Cada dissenyadora i dissenyador tenen el seu propi sistema: n'hi ha de més analítics i n'hi ha de més intuïtius. La forma en què es processa un problema de disseny dependrà també de l'experiència del professional: una persona principiant pot dedicar setmanes a reflexionar sobre la millor tipografia per a un logo, mentre que a un professional se li apareix la solució en pocs segons.

La tercera fase té diferents exemplificacions en funció del tipus d'encàrrec que ens facin i el model de disseny que tinguem al cap. Un grafista clàssic amb l'encàrrec de solucionar qüestions de comunicació gràfica amb un client concret podrà despatxar la fase d'avaluació amb unes poques trobades amb el client. Una dissenyadora interessada en el *design thinking*, la cocreació o el disseny inclusiu haurà d'assegurar mecanismes de *feedback* directes amb el públic molt més reglamentats i sistemàtics.

## 2. L'ontologia d'un disseny generatiu

Una vegada tenim una possible solució, encara que sigui un esbós, en un diagrama de flux, en pseudocodi, o en la nostra forma favorita de representar la solució, hem d'establir de forma més exhaustiva quins elements formaran part de la nostra solució gràfica i de quina manera estaran organitzats. En l'argot de les ciències de la computació aquest procés es coneix com a ontologia.

Originalment, l'ontologia era una disciplina filosòfica eminentment abstracta, que s'ocupava d'establir què és l'ésser. La part més fosca i esotèrica de la metafísica. Afortunadament per a nosaltres, el terme ha anat evolucionant i ara significa bàsicament establir quins objectes formen part d'un sistema, quines propietats poden tenir i com s'interrelacionen uns objectes amb uns altres. En l'exemple d'algorisme generatiu analògic per a construir un logo de la UOC vam veure un exemple molt simple d'ontologia.

L'ontologia en un sentit informàtic és una ciència complexa, que intenta inventariar les propietats de diferents objectes i les seves relacions per a tasques com ensenyar una intel·ligència artificial. Hi ha ontologies molt completes que, per exemple, permeten organitzar les informacions que un troba a la Viquipèdia en format classificació, de manera que un pot fer cerques del tipus: troba'm tots els científics nascuts al segle XX a Suïssa un divendres.

A primera vista pot sonar pura màstica: una ontologia de tota la Viquipèdia? Però en realitat són sistemes que poden consultar-se de forma senzilla quan un aprèn a programar. Potser un dia, si aneu avançant en el desenvolupament de projectes de disseny generatiu, us plantegeu conèixer com accedir a aquest tipus d'ontologies per a desenvolupar projectes realment ambiciosos. Però ara no us preocupeu: la idea que treballarem aquí d'ontologia és totalment interna, inclourà solament els objectes que formin part directa del vostre projecte gràfic i seran molt fàcils de desenvolupar.

### 2.1. Fix enfront de variable

Tret que el nostre pla sigui generar atzar absolut en els nostres gràfics –una cosa que finalment resultarà avorrit–, l'ontologia del nostre projecte haurà d'incloure elements fixos per a fer de contrapart als elements variables.



Aquí es troba precisament el poder dels gràfics generatius: poder observar l'aleatorietat dins d'una certa estructura que li dona sentit. O, vist des de l'altre costat, disposar d'una estructura parcialment estable però que s'adapta i transforma en funció de com canvia l'entorn.

Així, la primera tasca serà especificar quins elements es donaran de manera fixa i quins seran variables. Aquests elements invariants són molt importants, ja que són els que determinen una estructura bàsica de relacions que donaran sentit als elements variables.

Imagineu un client que us demana un logo per a la seva empresa, GeneTex. El client té una ment oberta en general, però hi ha alguna cosa en la qual no està disposat a cedir: l'estructura bàsica del logo ha de ser un cercle violeta amb el nom de l'empresa dins. La mida i la tipografia del nom podran variar, però dins del cercle hi ha d'haver només el nom de l'empresa i el color d'aquest cercle ha de ser violeta.

Així, l'ontologia del nostre disseny gràfic seria més o menys així:

- Contenedor:
  - Forma: cercle (invariable)
  - Color: violeta (invariable)
  - Mida: variable
- Contingut:
  - Text: GeneTex (invariable)
  - Tipografia: Time News Roman, Verdana, Baskerville, Futura, etc. (variable)
  - Mida de la lletra: de 10 a 24 punts (variable)

## 2.2. Determinista i seqüencial

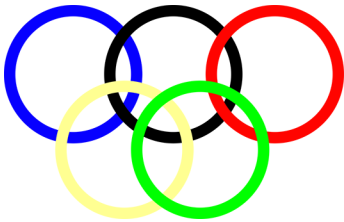
El disseny generatiu, en el fons, és com un art en el qual cal calibrar un equilibri entre els elements que ja venen prefixats per la grafista i els que volem que siguin variables.

Si ho deixem tot a mans de l'atzar, el resultat final acabarà sent indistingible de soroll pur i dur. Si tots els passos estan determinats per endavant, el resultat acabarà sent equivalent al disseny convencional, amb el qual perdem la capa generativa.

No hi ha cap problema a desenvolupar un disseny computacional plenament determinista, en el qual hem decidit per endavant de quina manera es comportaran els elements i som capaços de visualitzar *a priori* l'aspecte que tindria el nostre objecte gràfic final. Imaginem que ens encarreguen un redisseny de les anelles olímpiques, el logo de les Olimpíades. Volem fer un disseny molt precís, en el qual les distàncies de les anelles siguin perfectament equidistants, de manera que en lloc de dibuixar el logo a mà ho fem en un llenguatge de programació gràfica en el qual iterem una anella d'una mida fixa, al qual fem prendre cinc posicions específiques, calculades. Cada cercle tindrà un color específic, assignat pels codis Pantone.

Aquest disseny seria generatiu en el sentit que ha estat generat a través d'instruccions en lloc d'haver-lo traçat a mà, però seria un disseny fix: cada vegada que executéssim l'algorisme el resultat seria exactament igual. Si n'imprimíssim dues-centes còpies, totes serien idèntiques, de manera que perdria algunes de les característiques que fan que el disseny generatiu es diferenciï del clàssic.

Figura 1. Les anelles olímpiques versió determinista. El programa sempre generaria la mateixa imatge



Però insisteixo, no hi ha cap problema a fer-ho així. Cada problema té diverses solucions i si la grafista creu que la millor solució és un procés determinista, tirem endavant. De fet, John Maeda té dissenys professionals resolts amb programació en els quals el resultat final és totalment determinista: cada objecte gràfic que forma el cartell està predefinit de forma exacta per una sèrie de fórmules matemàtiques. Al contrari del nostre insuls exemple de les anelles olímpiques, aquests projectes de Maeda són rellevants perquè explota la capacitat de la programació per a dur a terme un disseny que a mà resultaria impossible, com omplir la pantalla amb la paraula *helvètica* en diferents posicions, molt a prop les unes de les altres, comportant-se com a megapíxels que dibuixen una A de la tipografia Helvètica.

Quan generem grafismes com els de Maeda, estem aprofitant la seqüencialitat dels llenguatges de programació. Si volem omplir un DIN A3 de petits cercles fins a generar una malla compacta de fons per a un cartell, si ho fem amb un programa d'edició gràfica comercial ens haurem de passar una estona ben avorrida tallant i enganxant cercles. En canvi, amb un llenguatge de programació com Processing ho solucionarem en dues línies. Crearem un bucle que farà que el programa dibuixi, per exemple, un cercle de 2 mm de diàmetre

cada mm, fins a arribar al final del full. En uns segons tindrem solucionat un problema francament tediós que hauria consumit una bona estona del nostre temps preciós.

Gràcies a la seqüencialitat també podem crear variacions progressives d'un disseny base. Així, podem transformar el concepte de producció seriada i, en comptes de generar diverses còpies del mateix disseny, anar generant diferents variacions, de manera que no hi hagi dos dissenys iguals en la sèrie. Per exemple, podem anar variant de forma progressiva el color d'un element o la seva posició al llarg de totes les generacions d'un cartell.

### 2.3. Seqüencialitat enfront d'aleatorietat

Un disseny seqüencial segueix sent determinista i cada vegada que executem el nostre miniprograma que omple de cercles un full sortirà exactament la mateixa imatge. Però no hi ha cap necessitat de quedar-se aquí. Com expliquem en el mòdul «En què consisteix el disseny generatiu», hi ha formes matemàtiques de generar processos pseudoaleatoris en un ordinador, de manera que podem incloure atzar en la seqüencialitat i generar diversos resultats gràfics segons quins nombres aleatoris sorgeixin.

Tornem al nostre exemple de redissenyar les anelles olímpiques. L'ontologia original que definiria el projecte seria la següent:

- Nombre d'anelles 5
- Mida de cada anella: radi de 50 píxels
- Colors i coordenades de cada anella: (100, 100) blava; (140, 100) negra; (180, 100) vermella; (120,130) groga; (150,130) verda (com que totes són anelles amb la mateixa mida, n'hi ha prou indicant la posició del centre d'aquesta per situar-ho en un plànol)

Però aquesta vegada el Comitè Olímpic ens deixa més llibertat i decidim jugar amb l'aleatorietat. Així, parametrizem el nostre logo i incloem en la nostra ontologia els paràmetres següents.

- Nombre de cercles: entre 5 i 10
- Mida de cada cercle: variarà entre 10 i 70 píxels
- Colors: completament aleatoris
- Posició: un nombre aleatori tant per a X com per a Y entre 100 i 300

Amb aquesta ontologia tan senzilla el nostre disseny ha deixat ja de ser determinista i té davant seu milions de diferents combinacions. Cada vegada que executem el nostre programa, d'unes escasses línies tindrem una imatge diferent. Que al Comitè Olímpic li agradin les nostres propostes és una altra cosa...

Figura 2. Quatre exemples de la quantitat d'objectes gràfics diferents que genera aquest simple exercici d'afegir aleatorietat a les anelles olímpiques



### 3. Dades externes

Fins aquí hem vist una descripció del disseny generatiu on treballem amb objectes virtuals generats pel mateix llenguatge de programació, però no hi ha res que ens impedeixi afegir-hi informació externa. Bàsicament, podem dividir el tipus de dades externes que inclourem en el nostre disseny generatiu en tres grans blocs: bases de dades, dades en directe i interacció amb els usuaris.

En el primer cas, disposem d'una sèrie de dades organitzades en una base de dades que el nostre programa llegirà i parametritzarà seguint una sèrie d'informacions.

En el segon cas, un sensor com una càmera o un termòmetre digital rebrà dades directament i les reexpedirà al nostre algorisme, o bé el sensor reexpedirà les dades a un servidor en línia i el nostre programa les anirà llegint directament en temps real, a mesura que els sensors les vagin situant en el servidor.

L'última opció fa referència a sistemes interactius en els quals el resultat final variarà en funció de les accions de l'usuari i seria com un subconjunt molt especial i important de les dades en directe.

#### 3.1. Bases de dades

A més dels elements que el nostre programa pugui generar de manera automàtica, podem incloure bases de dades de diferents tipus d'objectes digitals que poden ser inclosos de forma paramètrica en el nostre grafisme generatiu.

Vegem-ne un exemple senzill. Suposem que hem de fer el cartell per a un festival a la nostra ciutat on tocaran deu grups musicals diferents, i tenim tres fotografies en bona resolució de cada grup. En lloc de fer un únic cartell amb una foto en petit de cada grup, o escollir el grup més famós i posar una foto seva, preferim fer un disseny generatiu i combinar les fotografies amb certa aleatorietat. Així, definim per exemple un sistema de cartell on hi haurà una sèrie d'elements fixos que són el nom del festival, el lloc on tindrà lloc, les dates, etc. Això formarà una estructura fixa sobre la qual acoblar elements variables i aleatoris. En concret, posarem dues fotografies de dos grups, amb el nom de cada grup a baix. Quins seran aquests dos grups s'escollirà a l'atzar, així com quina fotografia de cada grup s'usarà. L'únic criteri que afegim a la selecció és que els dos grups siguin diferents. Això ens permet crear més de set-cents possibles cartells diferents.

No obstant això, és fàcil que vulguem aprofitar les possibilitats que un llenguatge de programació ens ofereix i jugar més amb aquestes fotos. Entrem així en la idea de processar les dades. Per a això, utilitzem la informació que ens ofereix la base de dades i la utilitzem per a remodelar el nostre sistema gràfic.

Imaginem que en la base de dades que inclou la fotografia i el nom del grup hi ha a més una indicació de l'estil que aquest grup musical practica. Podem llavors decidir un criteri extra en la nostra ontologia: que els dos grups que comparteixen cartell siguin d'estils molt diferents, per tal de crear una contraposició conceptual. Evidentment, un llenguatge de programació no sap res sobre estils musicals (ni tan sols l'increïble Processing!), de manera que hauríem d'incloure-hi unes línies de programació per a establir quins estils són similars i quins no.

### **3.2. Processament de dades i graus de generalització**

Quantes línies hauríem d'incloure al nostre programa per a decidir si dos estils musicals són similars o no i com hauria de desenvolupar-se el tema és una qüestió important. De fons, hi ha un problema bàsic del disseny generatiu: el grau de generalització que donarem al nostre algorisme. Volem fer un algorisme que resolgui un encàrrec concret? Busquem més aviat un sistema que permeti resoldre tota una família de problemes similars? O tenim al cap un programa màgic que solucioni de manera genèrica tota una tipologia?

No és una decisió fàcil, i dependrà de tot tipus de factors. La nostra personalitat és un element rellevant: Som persones sistemàtiques que busquem enquadrar un problema en una estructura tipus? O preferim solucionar cada problema al vol? Un altre factor clau és com és de comú el problema: ens trobarem de forma repetida amb aquest problema? O és una situació que no tornarà a passar d'aquí a un bon grapat d'anys?

Tornem a l'exemple del festival. Si aquesta és la primera vegada que fem el cartell per al festival i no queda clar si ens tornaran a contractar per a fer-lo –o si el festival es tornarà a fer– el més fàcil és no complicar-se la vida i incloure a mà quatre regles que ens indiqui que el Death Metal i el Grindcore són molt similars, però que difereixen molt del Reggae. Total, són deu grups, deu estils a tot estirar...

Imaginem ara que hi ha un compromís clar amb l'organització de poder dedicar-nos diversos anys a treballar amb ells, que a més d'aquest festival en faran d'altres de semblants, que la seva intenció és cada vegada incloure més grups en els seus festivals i que els ha encantat la idea de posar dues fotos de dos grups molt diferents. En aquest cas, segurament és una bona opció treballar una mica més en el programa, buscar informació sobre estils musicals i deixar

un algorisme robust que pugui decidir de manera automàtica quins estils musicals són similars i quins no. Així, la propera vegada que hàgiu de desenvolupar un cartell per a aquest client la decisió ja estarà automatitzada.

Finalment, imaginem ara una professional del grafisme famosa pels seus cartells generatius. Rep una gran quantitat d'encàrrecs i no vol rebutjar-ne cap. És fàcil que acabi creant la seva pròpia aplicació generadora de cartells on el sistema llegirà de forma automàtica text, dates, imatges, etc. i generi cartells en un estil de disseny evolutiu en el qual la dissenyadora l'única cosa que hagi de fer és clicar les opcions que li semblen més prometedores fins a tenir el cartell perfecte per a aquest client que té tanta pressa i necessita el cartell per a ahir.

D'altra banda, la decisió de generalitzar pot ser que no estigui associada a l'encàrrec pròpiament dit, sinó al nostre desig d'aprendre. Tornant a l'exemple del cartell de festival de música, encara que no tenim gens clar que els organitzadors del festival ens acabin contractant, podem decidir igualment fer un programa molt generalista, capaç de produir molts cartells diferents de festivals i concerts perquè volem aprendre a programar millor i la millor manera de fer-ho és davant d'un encàrrec concret.

### 3.3. Interactivitat i personalització

Fins ara hem estat parlant de sistemes generatius totalment autònoms. Creem l'algorisme i, una vegada l'executem, el sistema processa les dades de manera unilateral i ens dona un resultat final. Però un disseny generatiu no ha de quedar-se pas aquí.

Una de les raons per les quals el disseny generatiu és tan important és perquè facilita la revisió dels dissenys en temps real per part del grafista i/o del públic al qual va dirigit.

Podem distingir, així, dos tipus d'interactivitat: interactivitat en el procés de creació i interactivitat en el procés d'execució.

La interactivitat en el procés de creació és la que hem comentat a l'apartat 2.5 del mòdul «En què consisteix el disseny generatiu» en parlar de disseny evolutiu. El procés de creació del cartell no és autònom, sinó que l'algorisme va generant diferents opcions, que el grafista o el públic accepta o rebutja i a poc a poc es va generant un objecte gràfic. És a dir, el cartell, portada de llibre, etc. han estat creats de forma interactiva.

Clarament, aquesta és una versió molt limitada d'interactivitat, associada a objectes estàtics que ja no poden canviar, com ara cartells o portades de llibres en paper. Però res no impedeix imaginar objectes gràfics interactius, que canvien al llarg del temps en funció de les decisions que prenen els usuaris que

interactuen amb aquests. Formalment, però, els dos processos són molt similars: quan generem un *website* l'aspecte gràfic del qual sigui personalitzable per cada usuari o creem una aplicació interactiva per la qual un surfista empedreit pot personalitzar-se la planxa al seu gust abans d'encarregar-la, haurem de desenvolupar una ontologia en la qual especifiquem quins elements són fixos i quins són variables, quines opcions oferim a la usuària que siguin raonables i no causin problemes posteriors, i a través de quins canals aquesta usuària podrà fer arribar les seves decisions a l'algorisme.

De fet, aquesta és una altra de les grans revolucions que ens porta el disseny generatiu, i que solament ara comencem a albirar: la separació entre objecte estàtic enfront d'objecte interactiu no té cap sentit en un entorn digital. Si les portades dels nostres llibres electrònics són sempre iguals no obeeix a cap criteri tècnic o estètic, és el simple costum d'associar les propietats d'un objecte físic a un objecte virtual per a facilitar la nostra interacció amb aquest. En aquests temps d'impressió a demanda o de la fabricació d'objectes per mitjà del *crowdsourcing*, no hi ha cap raó per la qual totes les còpies d'un llibre físic hagin de tenir la mateixa portada, o que unes sabates solament es puguin trobar en negre i marró. Les raons són finalment protegir la capacitat de produir barat i a gran escala per part de grans multinacionals. Si el disseny generatiu et crida l'atenció, ara és un bon moment per a ficar-s'hi, ja que queda molt de terreny per explorar i disposem d'eines cada vegada més accessibles i que ens brinden moltes possibilitats.

### 3.4. Dades en viu. Animacions

Si el nostre projecte inclou dades en viu, podem plantejar-lo de dues formes bàsiques: d'una banda podem simplement considerar el temps com una dimensió extra i generar un objecte gràfic concret cada  $X$  temps. Imaginem que un equip de geòlegs ens contracta per a ajudar-los a establir la freqüència de terratrèmols en diferents països. La idea és rebre dades de diferents sismògrafs mundials i generar cada hora un document imprimible que consistirà bàsicament en un mapamundi on un codi de colors indicarà la freqüència relativa de terratrèmols a cada país. Si vermell és gran activitat sísmica i violeta nul·la activitat sísmica, cada hora tindrem una sèrie d'informacions distribuïdes de forma determinista en un DIN A4 com ara dia, hora, etc. i un mapamundi en el qual a través d'un sistema de processament de dades cada país prendrà una gamma del vermell si en aquell moment hi ha gran activitat sísmica i s'aproparà a tons blavosos si és una zona tranquil·la.

Amb aquesta forma de treballar les dades, en el fons el que estem fent és reduir aquest nou cas a la situació anterior. En realitat, el que fem és produir un gràfic generatiu a partir d'una base de dades. L'única diferència és que la base de dades canvia cada hora.



És una decisió possible i en aquest cas adequada per als interessos dels científics que ens han contractat. Però tenint dades que estan canviant contínuament és una pena no capturar aquest canvi de manera contínua. Necessitem, doncs, una animació.

Entrem, per tant, plenament en un entorn digital en el qual la nostra ontologia inclou un objecte que és contínuament variable segons el temps. Podem usar aquestes dades per a mostrar-les directament, com en el cas d'una càmera, transduir aquestes dades a una altra magnitud, com en l'exemple anterior de convertir l'activitat sísmica en una gamma de colors, o podem processar aquestes dades en temps real creuant-les amb altres informacions en viu o de bases de dades, com quan un navegador ens indica quina ruta per a tornar a casa amb cotxe és la menys transitada a partir de la informació que rep en temps real sobre la mobilitat a les carreteres properes.

Una altra possibilitat és establir un tipus de transducció menys cognitiva i més artística, com utilitzar les dades de temperatura que rebem de diferents països per a crear imatges expressionistes abstractes a l'estil d'un Motherwell o un Rothko. O, com la ballarina cïborg Moon Ribas, que converteix en moviments de dansa les dades de sismògrafs que rep a través d'un sensor en els seus turmells.

De nou, l'organització del nostre sistema serà la mateixa: tenim una sèrie d'elements fixos i uns altres de variables. En alguns casos la variació es generarà a través d'un procés aleatori. En uns altres es llegiran dades d'una base de dades; però una part central de l'objecte gràfic interactiu seran dades en temps real, ja siguin internes o externes, que s'organitzaran en algun tipus d'animació. Entenent aquí «animació» en un sentit ampli, que simplement recull la idea d'elements visuals que es van transformant al llarg del temps i aquestes transformacions poden contemplar-se en temps real.

## 4. Híbridació

Hem parlat aquí de diferents objectes gràfics produïts de manera generativa, seguint un principi de classificació, buscant tipus purs. Però res no ens impedeix jugar promíscuament amb aquests tipus i generar tota mena d'objectes i aplicacions híbrids.

Així, en lloc de representar fidelment les dades de la temperatura exterior en una animació podem utilitzar les variacions en temperatura, com a nombres aleatoris que canviïn l'aspecte d'un bàner cada hora. O utilitzar cent fotos d'una càmera digital a internet d'un interval de 24 hores per a fer un cartell clàssic que reflexiona sobre el pas del temps. Res no ens impedeix combinar informació en bases dades amb els moviments d'un usuari davant d'una càmera de vídeo per a inventar un nou tipus de videojoc o que un cartell digital penjat a internet mostri un paisatge nocturn o diürn en funció de la zona horària en la qual es troba la persona que el visita.

De la mateixa forma, en aquesta època d'impressió sota demanda i producció digital no hi ha raons de pes per les quals els objectes gràfics físics no puguin presentar diferents aspectes. Res no impedeix tenir 125 cartells diferents d'una exposició, o que cada vegada que s'imprimeix una novel·la la portada tingui un joc de figures geomètriques d'estil mitjà diferent. Aquests mateixos principis els podem traslladar al disseny de moda, de producte, d'interfícies o de serveis.

I les raons no han de ser purament estètiques o de disseny d'innovar. Hi ha també raons ètiques i socials per a la pluralitat en el disseny gràfic. Imagina la interfície d'un sistema de comerç electrònic en el qual en lloc de tenir un sol sistema d'interacció tenim en compte la diversitat sensorial i neuronal, i així tenim un sistema d'interacció pensat especialment per a les persones cegues o amb problemes a l'hora de distingir colors, un cartell pensat especialment per a persones amb síndrome d'Asperger o amb dificultats cognitives. Els temps d'«una sola talla va bé per a tothom» han passat a la història. Gràcies al disseny generatiu disposem ara d'eines individualitzades per a polvoritzar distincions i crear tot tipus de sistemes híbrids.

## Bibliografia

**Cross, Nigel** (2001). «Designerly ways of knowing: design discipline versus design science». *Design Issues* (vol. 3, núm. 17 pàg. 49-55). <<https://oro.open.ac.uk/3281/1/Designerly-DisciplinevScience.pdf>>

**Engelhard, Yuri** (2002). *The language of graphics Institute for Logic, Language and Computation*. Universitat d'Àmsterdam.

**Fry, Ben** (2007). *Visualizing Data*. O'Reilly.

**Kress, Gunther; Van Leeuwen, Theo** (1996). *Reading Images. The Grammar of Visual Design*. Routledge.

**Stuikys, Vytautas; Damaševičius, Robertas** (2008). «Design of ontology-based generative components using enriched feature diagrams and meta-programming». *Information Technology and Control*(vol. 37, núm. 4, pàg. 301-310). <<https://www.semanticscholar.org/paper/DESIGN-OF-ONTOLOGY-BASED-GENERATIVE-COMPONENTS-AND-Stuikys/5cbb085827a71f516e2e27b090adeef38b0a586b>>

**Wilkinson, Leland** (1999). *The Grammar of Graphics*. Springer.

