



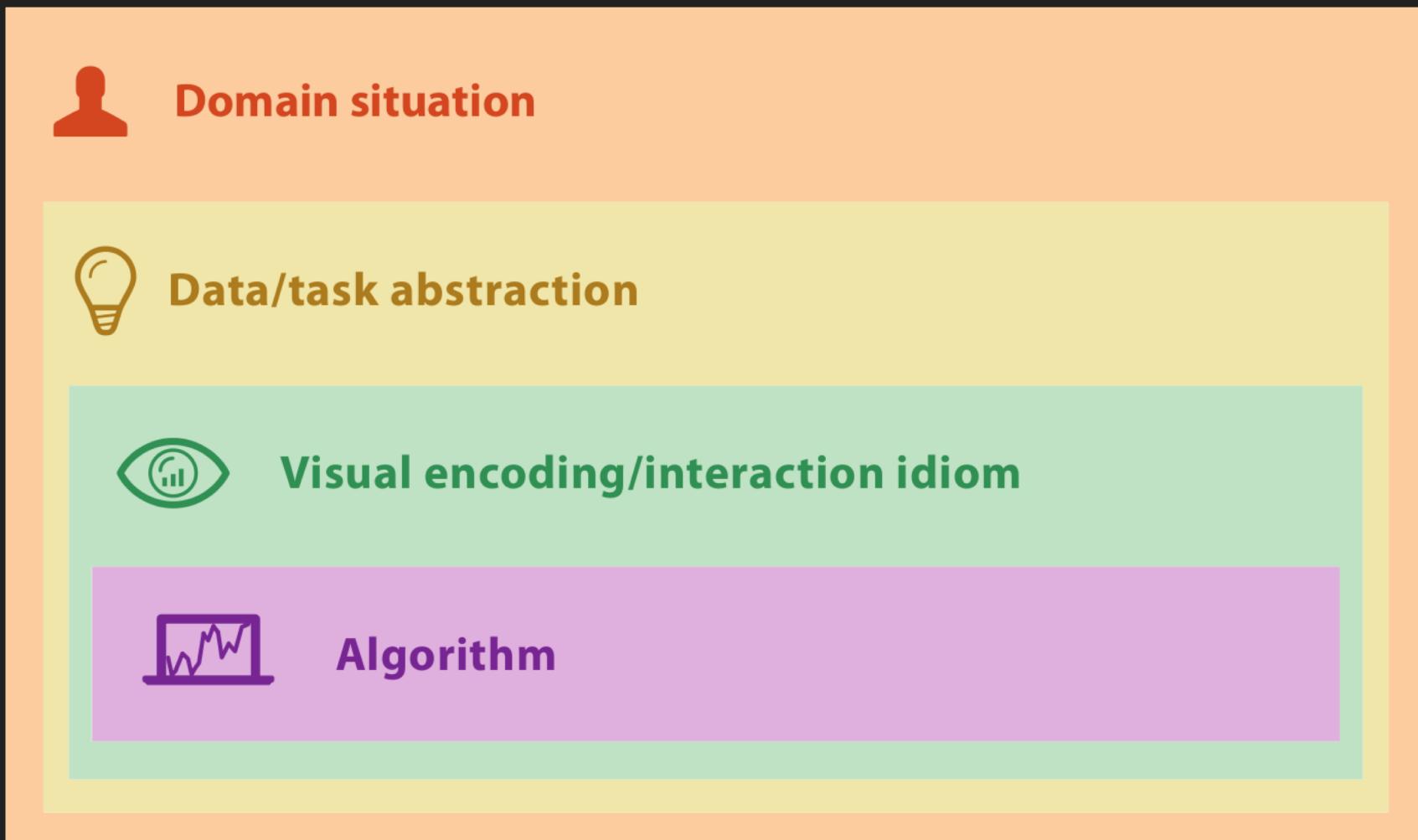
PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Validación

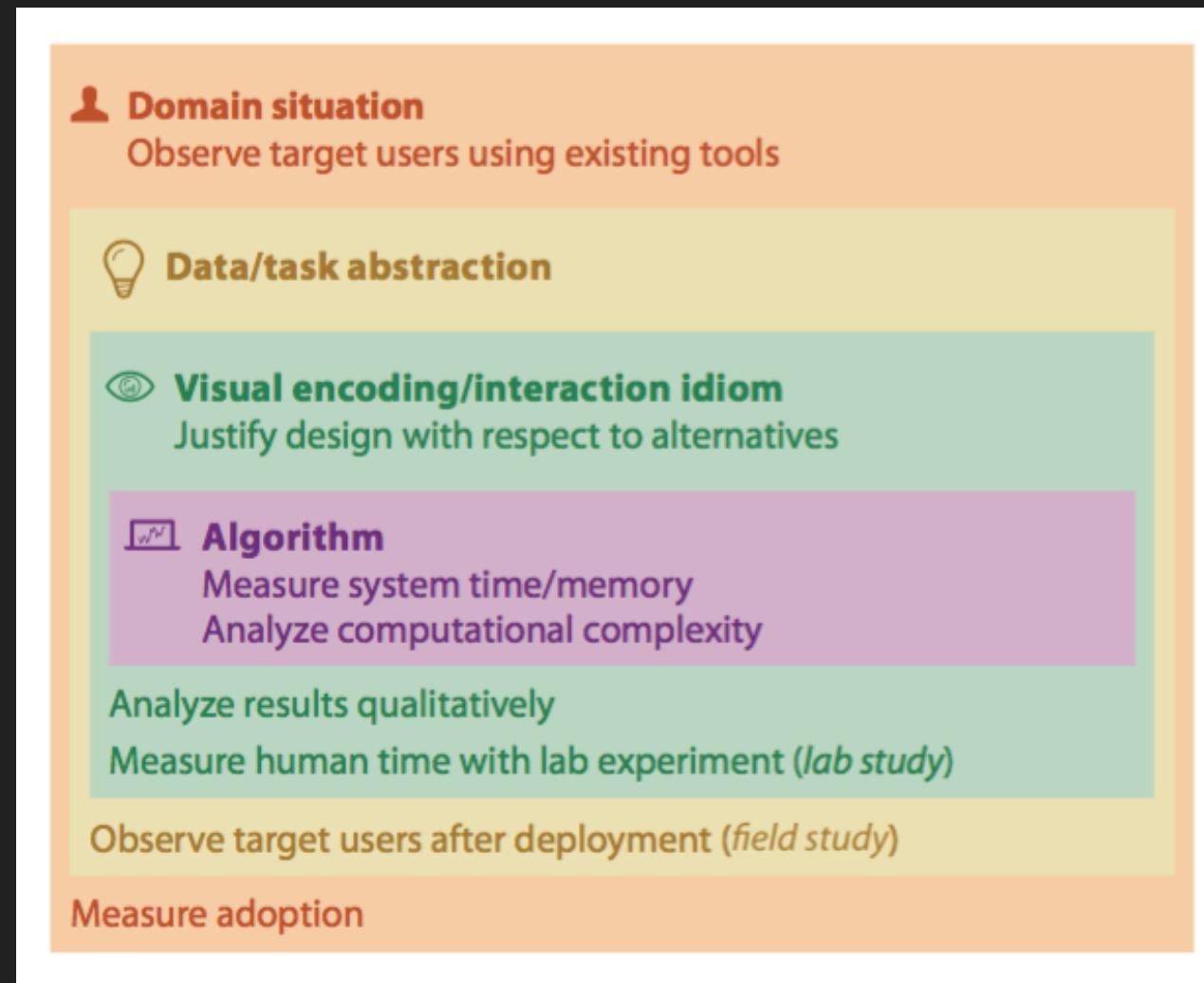
Visualización de Información
IIC2026

Profesor: Denis Parra

Cuatro niveles de validación



Cuatro niveles de validación (detalle)



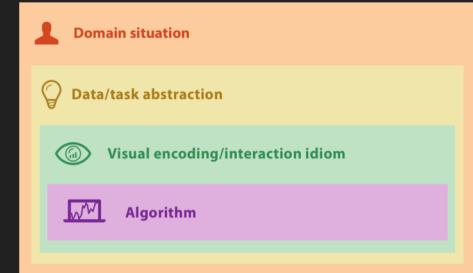
Cuatro niveles de validación

- Este *framework* presenta un modelo de cuatro niveles para validar.
- Estos niveles están anidados:
 - *domain situation*
 - *task and data abstraction (why and what)*,
 - *visual encoding and interaction idiom (how)*,
 - *algorithm.*
- Cada nivel tiene sus amenazas, que debemos ir analizando paso a paso.

¿Por qué validar?

- Como vimos al inicio del curso, validar es importante:
- El *espacio* de combinaciones es gigante.
 - La gran mayoría de las combinaciones produce un diseño ineficiente.
- Validar es un problema engorroso; sin embargo, es valioso pensar en ello al **principio del proceso** de diseño, en vez dejarlo como un *afterthought*.

Un *framework* en cascada



- El nivel de más arriba se consideran los detalles del dominio en particular.
- Luego, en el nivel de abstracción ***what-why***, estos problemas y datos específicos al dominio se mapean en formas independientes al dominio.
- En el siguiente nivel (en ***how***) se validan los métodos que especifican el *visual encoding* y la interacción.
- Finalmente, en el último nivel, se revisan los algoritmos que permiten instanciar computacionalmente estos métodos.

Un *framework* en cascada

- Los niveles se dicen anidados, puesto que el *output* del nivel superior será el *input* del nivel que le sigue. Al resultado de cada nivel, le llamaremos **bloque**.
 - Por ejemplo, una mala decisión en la etapa de abstracción (*what and why*) puede producir una visualización que no resuelve el problema, incluso habiendo hecho un perfecto trabajo para los niveles de más abajo.
- El objetivo de separar la validación en cuatro niveles es **desacoplar** las responsabilidades que tiene cada uno.
- De esta forma, este análisis se puede realizar de forma independiente e iterativa, sin importar del orden de las decisiones de diseño.



Domain situation

Observe target users using existing tools



Data/task abstraction



Visual encoding/interaction idiom

Justify design with respect to alternatives



Algorithm

Measure system time/memory

Analyze computational complexity

Analyze results qualitatively

Measure human time with lab experiment (*lab study*)

Observe target users after deployment (*field study*)

Measure adoption

Domain situation 1/3

- Este bloque describe un dominio específico, con un grupo de usuarios, su dominio de interés, su vocabulario, sus preguntas y sus datos.
 - Ejemplo: un biólogo computacional en el campo de genética comparativa, usando secuencias de datos genéticos, formulando preguntas acerca de la adaptabilidad de especies.
 - Ejemplo: Público general tomando decisiones de salud en la presencia de incertezas.
- Lo importante de este nivel es **entender** las necesidades del usuario.
 - Generalmente, se realizan entrevistas, observaciones, investigaciones acerca del usuario y sus actividades relacionadas con este problema.

Domain situation 2/3

- Desarrollar un **entendimiento claro** de los requerimientos es un problema engorroso para un diseñador (de aquí nace, *user-centered design* o *human-centered design*).
 - El error típico es hacer supuestos simplificadores en vez de *comprometerse* con los usuarios.
 - Ellos (los usuarios) saben que, de alguna forma, deben *ver* sus datos; pero les costará articular sus requerimientos analíticos de una forma precisa.
 - Formular preguntas sobre sus acciones es insuficiente: lo que ellos dicen, pensando en el pasado, entrega una vista incompleta con respecto al análisis de lo que actualmente hacen.

Domain situation 3/3

- El resultado de identificar un bloque de dominio/situación es un conjunto detallado de preguntas o acciones llevadas a cabo por usuarios-objetivo, acerca de una colección de datos heterogénea que es comprendida en detalle.
- Volviendo al ejemplo del biólogo computacional
 - OK: ¿cuál es la densidad de cobertura y dónde están las brechas entre cromosomas?
 - Not OK! ¿cuál es la base genética de una enfermedad? (demasiado general)

Discusión

- Este bloque describe un dominio específico, con un grupo de usuarios, su dominio de interés, su vocabulario, sus preguntas y sus datos.
- Lo importante de este nivel es entender las necesidades del usuario.

TopicPanorama: a Full Picture of Relevant Topics

IEEE VAST 2014

[Shixia Liu](#)¹ [Xiting Wang](#)^{1,2} [Jianfei Chen](#)² [Jun Zhu](#)² [Baining Guo](#)^{1,2}
¹[Microsoft Research Asia](#) ²[Tsinghua University](#)

CNVis: A Web-Based Visual Analytics Tool for Exploring Conference Navigator Data

*Samuel M. Bailey; University of Notre Dame; Notre Dame, IN
Justin A. Wei; University of North Texas; Denton, TX
Chaoli Wang; University of Notre Dame; Notre Dame, IN
Denis Parra; Pontificia Universidad Católica de Chile; Santiago, Chile
Peter Brusilovsky; University of Pittsburgh; Pittsburgh, PA*

 **Domain situation**

Observe target users using existing tools

**Data/task abstraction****Visual encoding/interaction idiom**

Justify design with respect to alternatives

**Algorithm**

Measure system time/memory

Analyze computational complexity

Analyze results qualitatively

Measure human time with lab experiment (*lab study*)

Observe target users after deployment (*field study*)

Measure adoption

Task and data abstraction 1/3

- En este bloque se necesita abstraer el problema a un vocabulario independiente del dominio.
- Esta abstracción nos permitirá notar qué bloques de dominio —que pueden ser descritos con lenguajes distintos— desembocan en razones similares de por qué un usuario necesita la herramienta de visualización.
 - De esta forma, preguntas desde dominios completamente distintos pueden ser mapeados a las mismas *tasks abstractas*: *browsing*, *comparing*, *summarizing*.

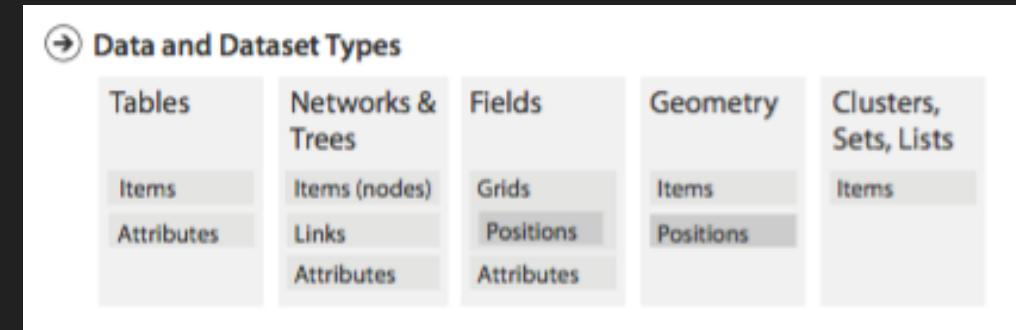
	Target known	Target unknown
Location known	••• <i>Lookup</i>	•• <i>Browse</i>
Location unknown	<i>Locate</i>	<i>Explore</i>

Task and data abstraction 2/3

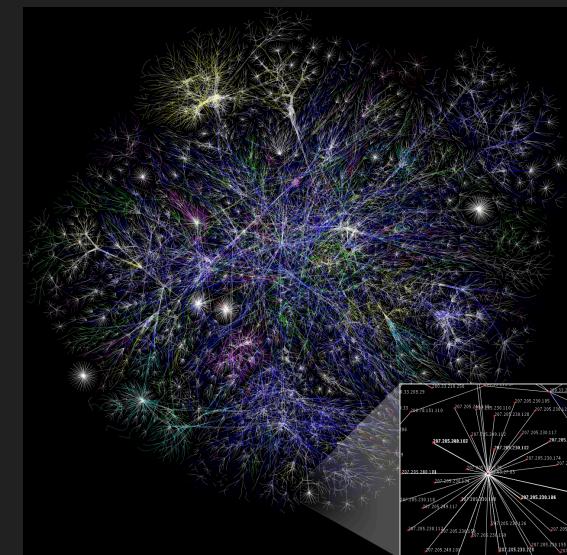
- Por otra parte, los bloques de datos son **diseñados**.
 - Seleccionar un *bloque de datos*, más que un acción de identificación (como lo es con *tasks*), es un paso creativo de diseño.
 - Si bien es posible que los datos sean utilizados exactamente de la misma forma que en la situación del dominio, es probable que estos datos sean transformados.
- El objetivo es determinar qué *data type* puede ayudar a una representación visual que responda a los problemas del usuario.
 - A veces, mantener la forma original del *dataset* hace un buen match con el *visual encoding* que resuelve el problema; sin embargo, transformar a otro *data type* puede proporcionar una mejor solución.

Task and data abstraction 3/3

- Recordemos tipos de datos y datasets



- Ejemplo de mala elección de task/data abstraction: mostrar mapa de sitio web para resolver el problema de “lost in hyperspace”



https://upload.wikimedia.org/wikipedia/commons/d/d2/Internet_map_1024.jpg

Discusión

- ¿ Qué tareas, tipos de datos y de datasets se utilizaron en los papers estudiados ?

TopicPanorama: a Full Picture of Relevant Topics

IEEE VAST 2014

[Shixia Liu](#)¹ [Xiting Wang](#)^{1,2} [Jianfei Chen](#)² [Jun Zhu](#)² [Baining Guo](#)^{1,2}
¹[Microsoft Research Asia](#) ²[Tsinghua University](#)

CNVis: A Web-Based Visual Analytics Tool for Exploring Conference Navigator Data

*Samuel M. Bailey; University of Notre Dame; Notre Dame, IN
Justin A. Wei; University of North Texas; Denton, TX
Chaoli Wang; University of Notre Dame; Notre Dame, IN
Denis Parra; Pontificia Universidad Católica de Chile; Santiago, Chile
Peter Brusilovsky; University of Pittsburgh; Pittsburgh, PA*

 **Domain situation**

Observe target users using existing tools

**Data/task abstraction****Visual encoding/interaction idiom**

Justify design with respect to alternatives

 **Algorithm**

Measure system time/memory

Analyze computational complexity

Analyze results qualitatively

Measure human time with lab experiment (*lab study*)

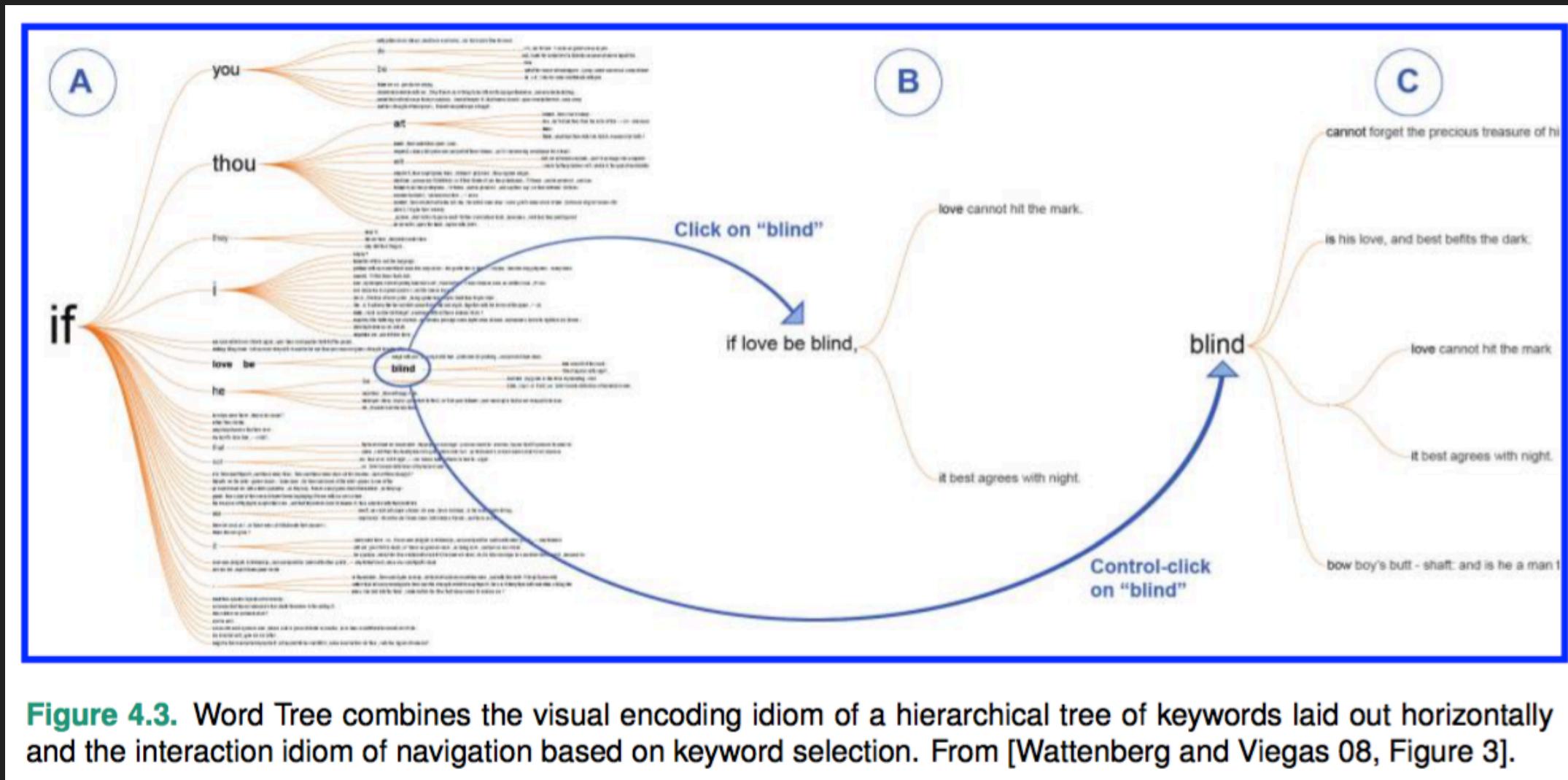
Observe target users after deployment (*field study*)

Measure adoption

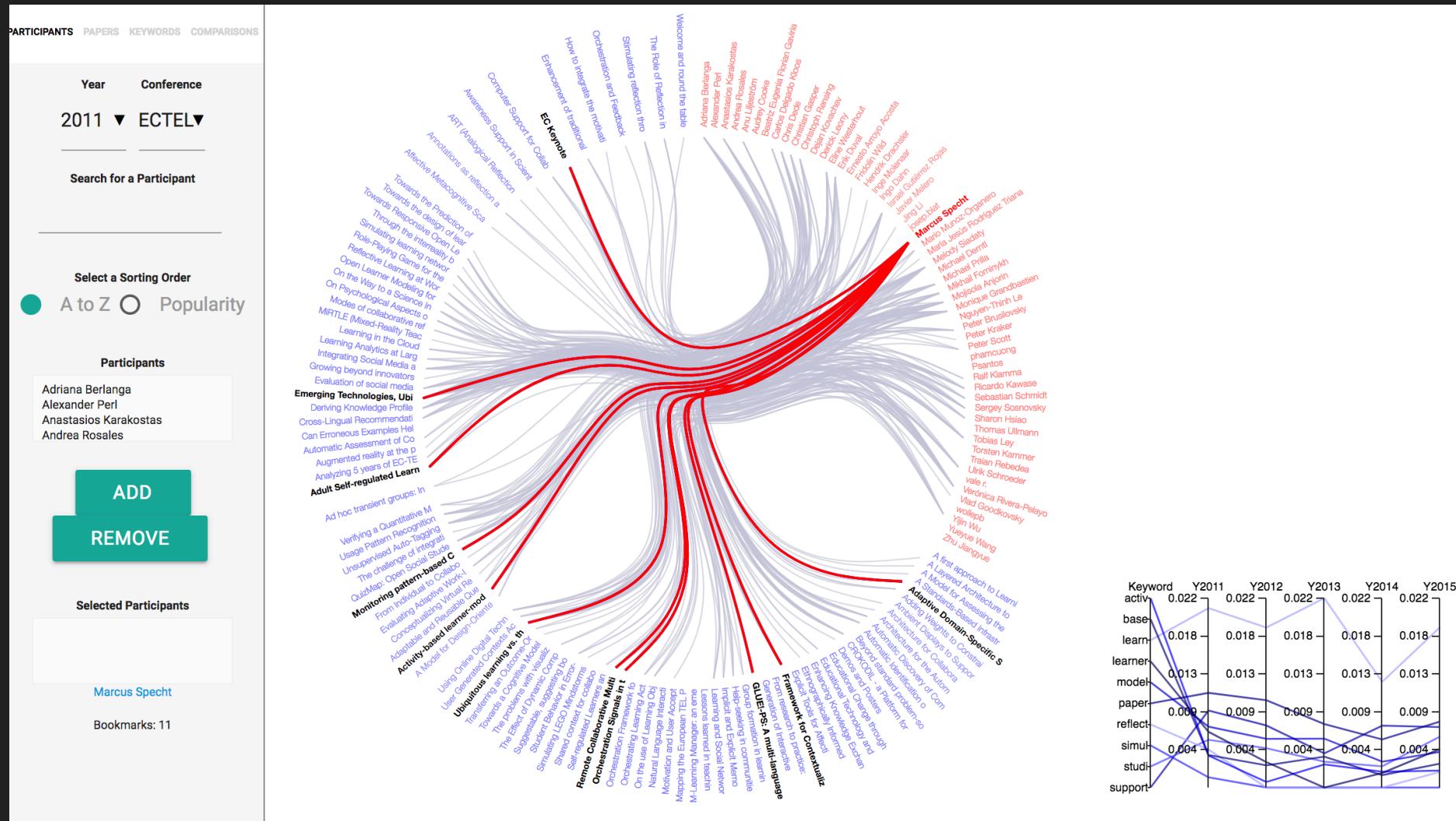
Visual encoding and interaction

- En este nivel se decide la manera específica de cómo crear y manipular la representación visual, según los resultados del bloque anterior
- A cada posible *approach*, le llamaremos *idiom*.
- Acá hay dos asuntos importantes en el diseño de un *idiom*:
 - Por un lado, debemos definir las opciones que nos mostrarán una *imagen de los datos*. Esto sería el *idiom del visual encoding*, que controla exactamente lo que el usuario ve.
 - Por otro lado, tenemos que responder a un conjunto de preguntas que nos permitirán manipular esta representación de forma dinámica. Esto sería el *idiom de la interacción*, que controla cómo los usuarios pueden ir cambiando lo que están viendo.

Word Tree (WattenBerg & Viegas '08)



CNVis (Davis, Huang, Parra, Brusilovsky '17)



Discusión

- ¿Qué visual encodings e interacciones se implementaron para estos papers?

TopicPanorama: a Full Picture of Relevant Topics

IEEE VAST 2014

[Shixia Liu](#)¹ [Xiting Wang](#)^{1,2} [Jianfei Chen](#)² [Jun Zhu](#)² [Baining Guo](#)^{1,2}
¹[Microsoft Research Asia](#) ²[Tsinghua University](#)

CNVis: A Web-Based Visual Analytics Tool for Exploring Conference Navigator Data

*Samuel M. Bailey; University of Notre Dame; Notre Dame, IN
Justin A. Wei; University of North Texas; Denton, TX
Chaoli Wang; University of Notre Dame; Notre Dame, IN
Denis Parra; Pontificia Universidad Católica de Chile; Santiago, Chile
Peter Brusilovsky; University of Pittsburgh; Pittsburgh, PA*

 **Domain situation**

Observe target users using existing tools

**Data/task abstraction****Visual encoding/interaction idiom**

Justify design with respect to alternatives

**Algorithm**

Measure system time/memory

Analyze computational complexity



Analyze results qualitatively

Measure human time with lab experiment (*lab study*)

Observe target users after deployment (*field study*)

Measure adoption

Algorithm

- El último nivel es el algoritmo: la elaboración de un procedimiento detallado que le permite al computador generar el objetivo deseado.
- El objetivo es manejar de forma eficiente los métodos escogidos en el nivel anterior. Estos bloques de algoritmos son también **diseñados**, ya que muchos algoritmos diferentes podrían lograr el mismo *idiom*.
 - Ejemplo: buscar un algoritmo para mostrar eficientemente un modelo en tres dimensiones de una resonancia magnética. Diferentes algoritmos han sido propuestos para alcanzar los requerimientos de este *idiom*: *ray casting*, *splatting*, o *texture mapping*.
- La elección del algoritmo puede depender de la rapidez del cómputo, de la cantidad de memoria utilizada, o si es que el resultado de la imagen es un *match* exacto del *visual encoding* o es simplemente una aproximación.

Ejemplo: algoritmos para dibujar grafos

Graph Drawing by Force-directed Placement

THOMAS M. J. FRUCHTERMAN* AND EDWARD M. REINGOLD
Department of Computer Science, University of Illinois at Urbana-Champaign, 1304 W.
Springfield Avenue, Urbana, IL 61801-2987, U.S.A.

SUMMARY

We present a modification of the spring-embedder model of Eades [*Congresses Numerantium*, 42, 149–160, (1984)] for drawing undirected graphs with straight edges. Our heuristic strives for uniform edge lengths, and we develop it in analogy to forces in natural systems, for a simple, elegant, conceptually-intuitive, and efficient algorithm.

KEY WORDS Graph drawing Force-directed placement Multi-level techniques Simulated annealing

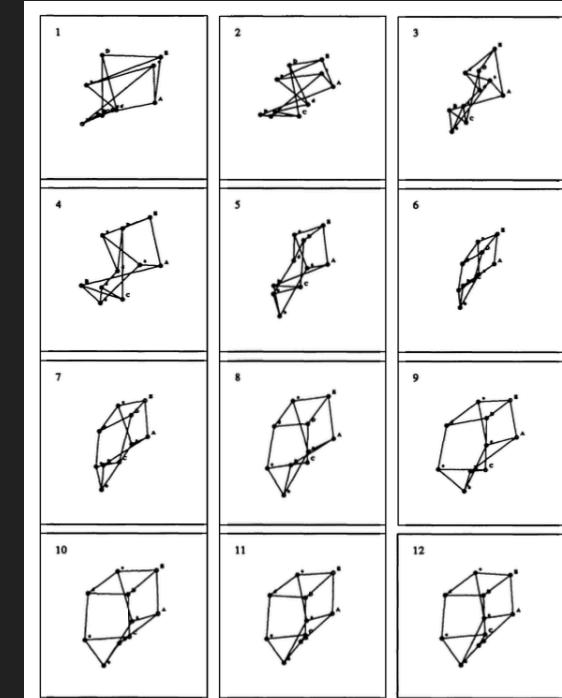
THE GRAPH-DRAWING PROBLEM

A graph $G = (V, E)$ is a set V of vertices and a set E of edges, in which an edge joins a pair of vertices.¹ Normally, graphs are depicted with their vertices as points in a plane and their edges as line or curve segments connecting those points. There are different styles of representation, suited to different types of graphs or different purposes of presentation. We concentrate on the most general class of graphs: undirected graphs, drawn with straight edges. In this paper, we introduce an algorithm that attempts to produce aesthetically-pleasing, two-dimensional pictures of graphs by doing simplified simulations of physical systems.

We are concerned with drawing undirected graphs according to some generally accepted aesthetic criteria:²

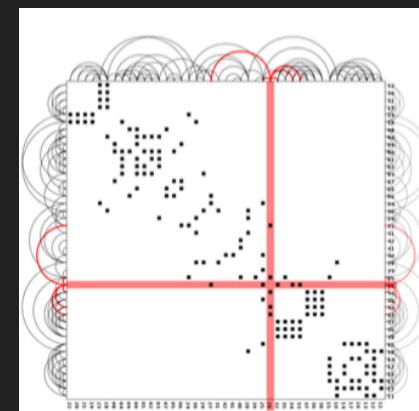
1. Distribute the vertices evenly in the frame.
2. Minimize edge crossings.
3. Make edge lengths uniform.
4. Reflect inherent symmetry.
5. Conform to the frame.

Our algorithm does not explicitly strive for these goals, but does well at distributing vertices evenly, making edge lengths uniform, and reflecting symmetry. Our goals for the implementation are speed and simplicity.



3 Adjacency Matrix Representations

An adjacency matrix (Figure 5, top) contains one row and one column for each node of a network. Given two nodes i and j , the cells located at (i, j) and (j, i) in the matrix contain information about the edge(s) between the two nodes. Typically, each cell contains a boolean value indicating if an edge exists between the two nodes. (In the figures in this article, a true boolean value is shown as a black, filled-in cell.) If the graph is undirected, the matrix is symmetric, i.e., the two cells (i, j) and (j, i) correspond to the same edge. If the graph is directed, however, the matrix is not symmetric.



Visiones: Top-down o Bottom-up

- Proyectos Problem-Driven (top down): Problemas del mundo real donde probablemente algún idiom o grupo de idioms existentes los resuelven.
 - En literatura de visualización se les conoce como estudio de diseño
- Proyectos Technique-Driven (bottom up): diseño e implementación de idioms o algoritmos, independientes de un dominio de conocimiento específico.

Amenazas a la validez 1/2

- Validar la efectividad de un diseño es difícil, pues existen múltiples preguntas.
- Sin embargo, este modelo anidado permite cuestionar nuestro diseño en cada nivel, de forma separada.
- Esto, a su vez, nos permite enfrentar un mar de posibles preguntas e ir respondiéndolas, nivel a nivel.

Amenazas a la validez 2/2

- Las amenazadas pueden ser clasificadas como:
 - Un **problema** incorrecto
 - Una **abstracción** incorrecta
 - Un ***idiom*** incorrecto
 - Un **algoritmo** incorrecto

Amenazas a la validez



Domain situation

You misunderstood their needs



Data/task abstraction

You're showing them the wrong thing



Visual encoding/interaction idiom

The way you show it doesn't work



Algorithm

Your code is too slow

Approaches para validar

 **Threat** Wrong problem

 **Validate** Observe and interview target users

 **Threat** Wrong task/data abstraction

 **Threat** Ineffective encoding/interaction idiom

 **Validate** Justify encoding/interaction design

 **Threat** Slow algorithm

 **Validate** Analyze computational complexity

 **Implement system**

 **Validate** Measure system time/memory

 **Validate** Qualitative/quantitative result image analysis

Test on any users, informal usability study

 **Validate** Lab study, measure human time/errors for task

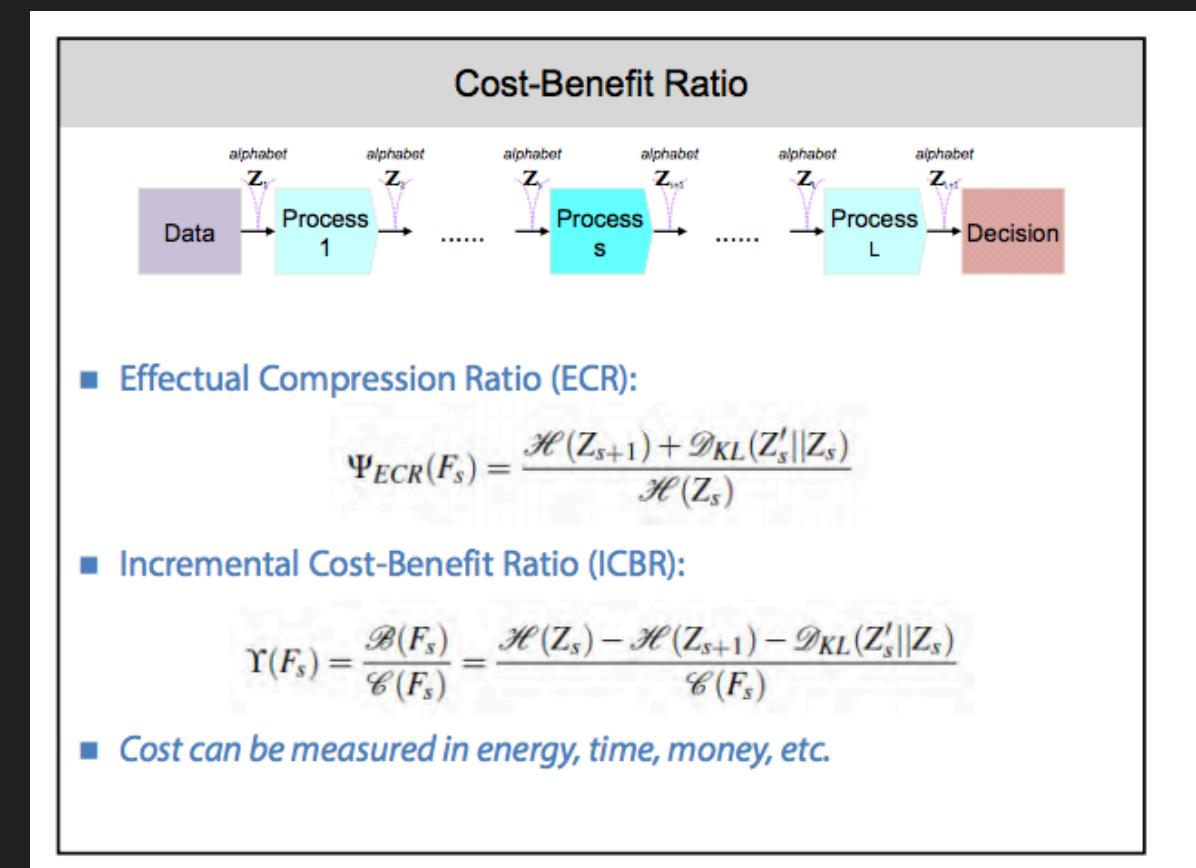
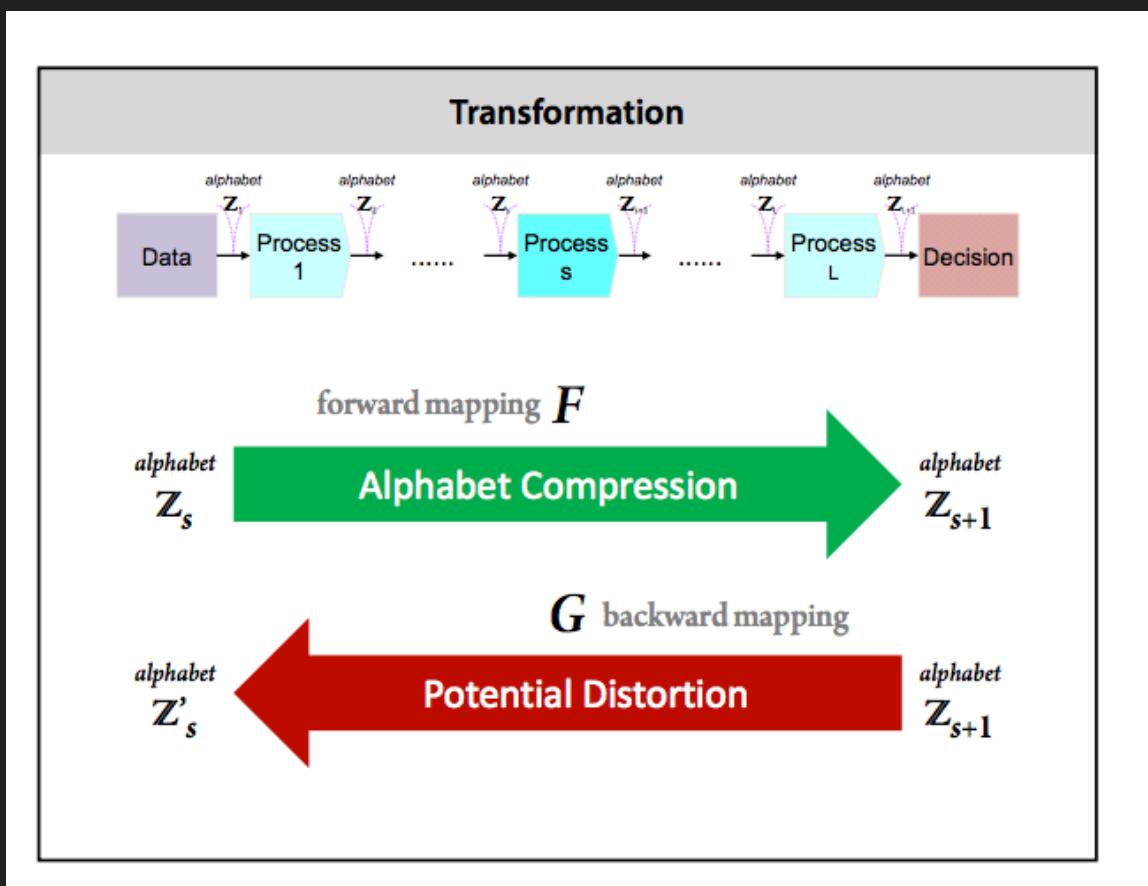
 **Validate** Test on target users, collect anecdotal evidence of utility

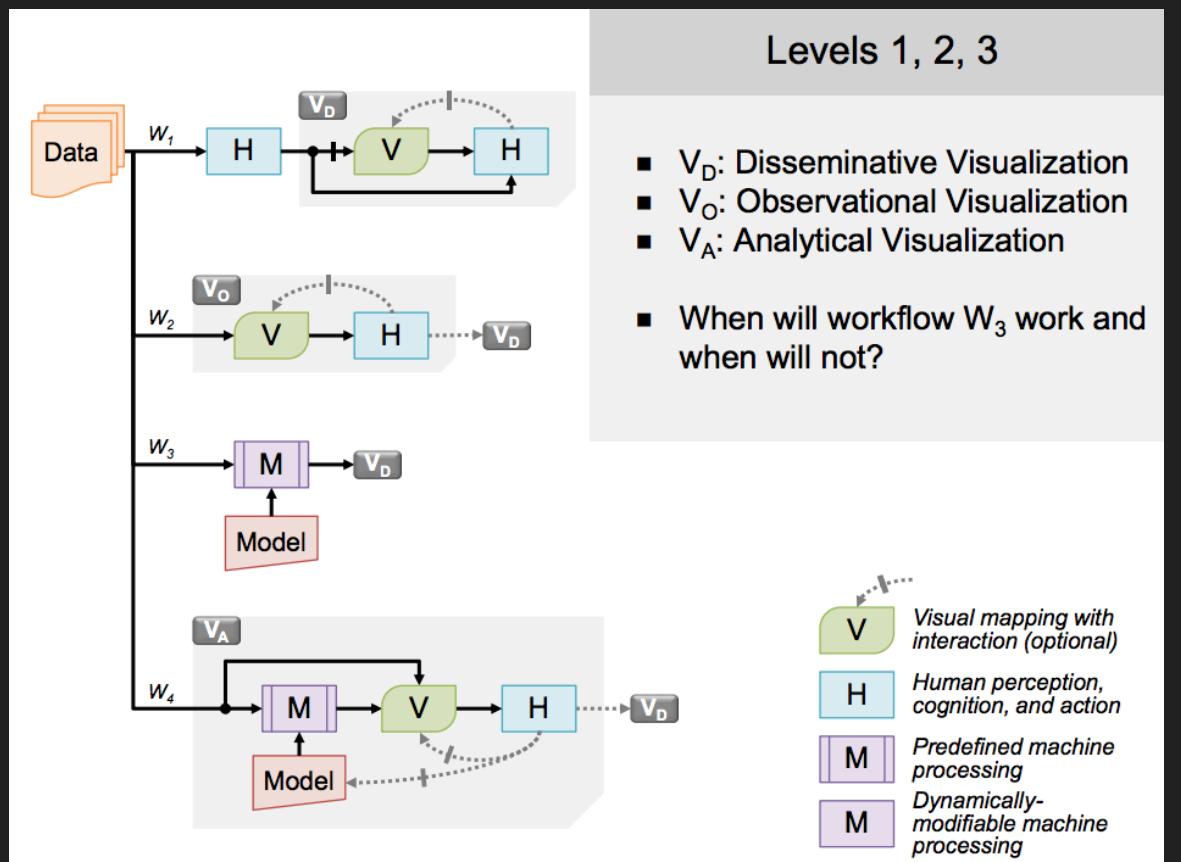
 **Validate** Field study, document human usage of deployed system

 **Validate** Observe adoption rates

Otros frameworks de validación

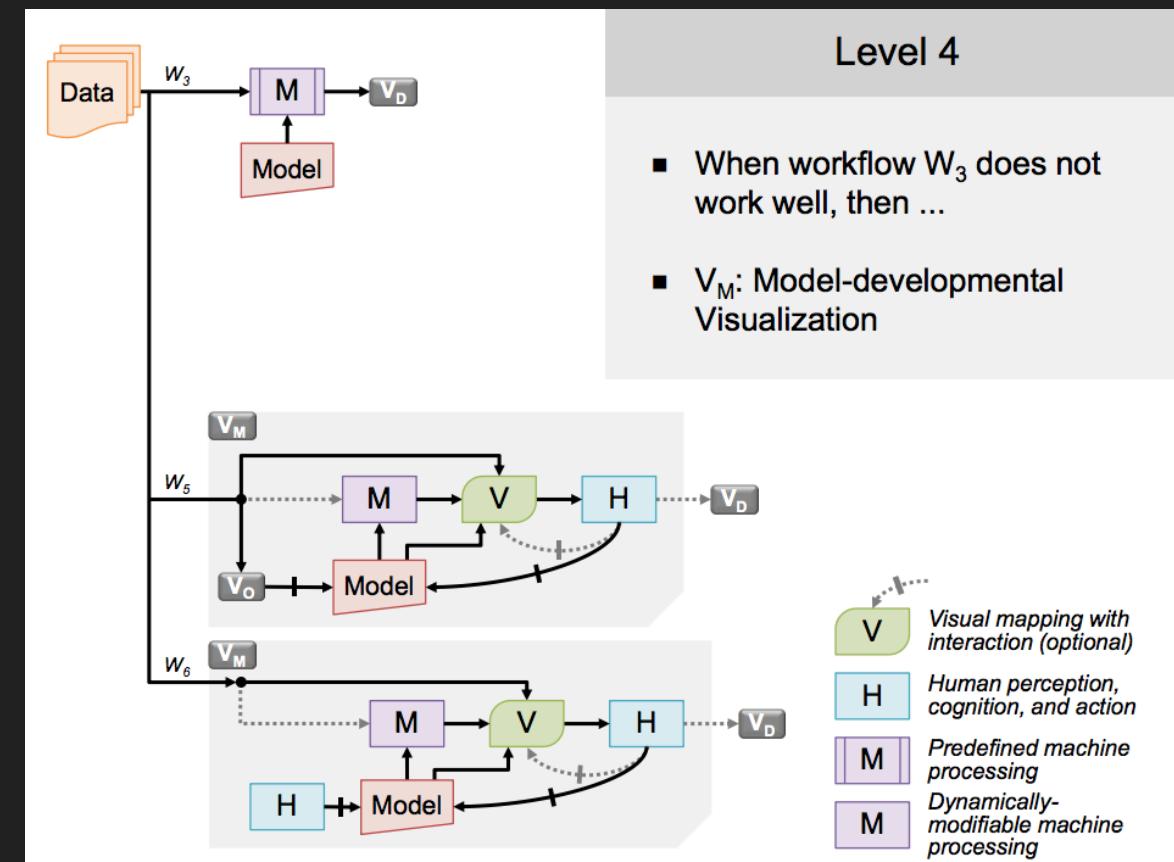
- M. Chen and A. Golan, “**What may visualization processes optimize?**” *IEEE Transactions on Visualization and Computer Graphics*, Accepted in 2015 (available in Xplore).
 - Further details on the cost-benefit metric, and case studies
- M. Chen and H. Jänicke. “**An information-theoretic framework for visualization.**” *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1206–1215, 2010
 - Reasoning about the visualization and interaction in relation to DPI
- T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.
 - Comprehensive coverage of information theory





Levels 1, 2, 3

- **V_D**: Disseminative Visualization
- **V_O**: Observational Visualization
- **V_A**: Analytical Visualization
- When will workflow **W₃** work and when will not?



Level 4

- When workflow **W₃** does not work well, then ...
- **V_M**: Model-developmental Visualization

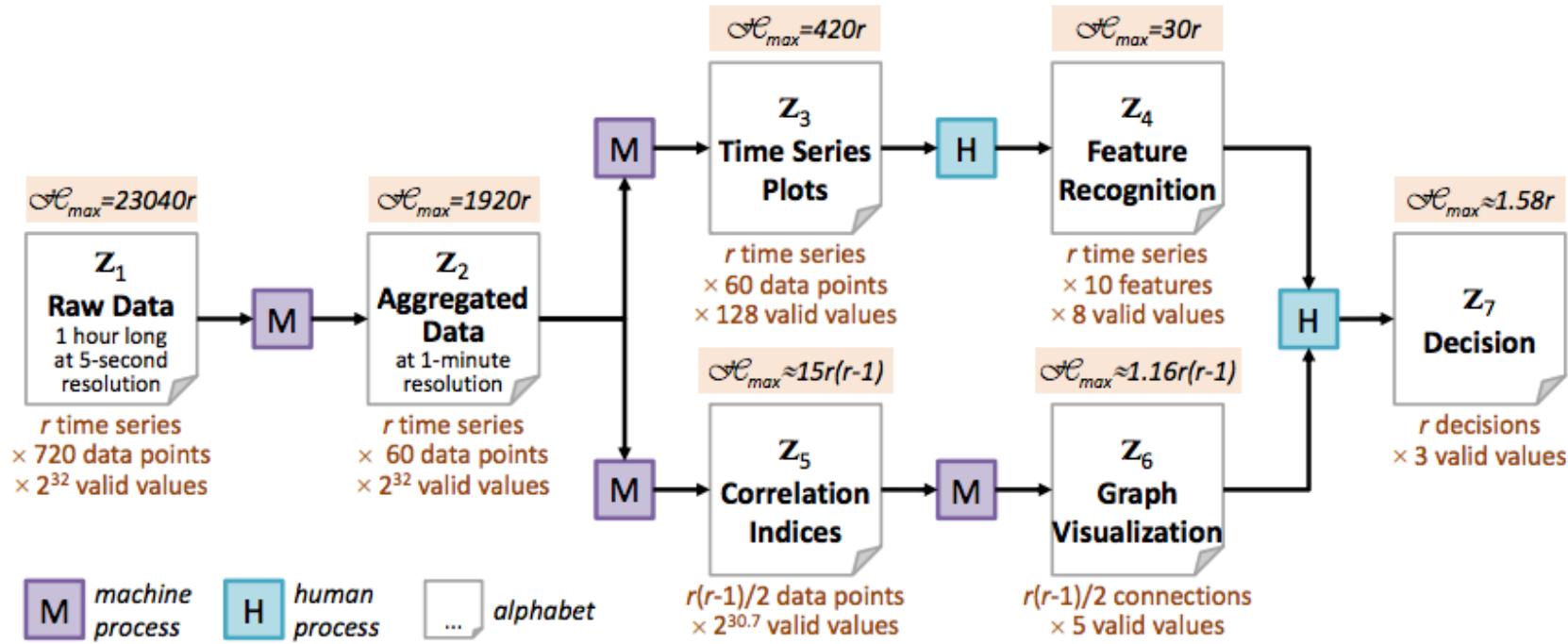


Fig. 3. An example transformation of alphabets during a data analysis and visualization process. From left to right, the initial alphabet corresponds to r time series each capturing a share price at 5 second interval within an hour. For each time series, the 12 data points in every minute are then aggregated into a mean value. The r time series is then visualized as line plots. The analyst identifies various features during the visualization, such as different levels of rise or fall, different intensity, etc. Meanwhile, the analyst computes the correlation indices between each pair of time series and visualize these using, for instance, a circular graph plot, where correlation indices are mapped to five different colors. The analyst finally makes a decision for each of the r shares as to buy, sell or hold. The maximal entropy \mathcal{H}_{MAX} shows a decreasing trend from left to right.